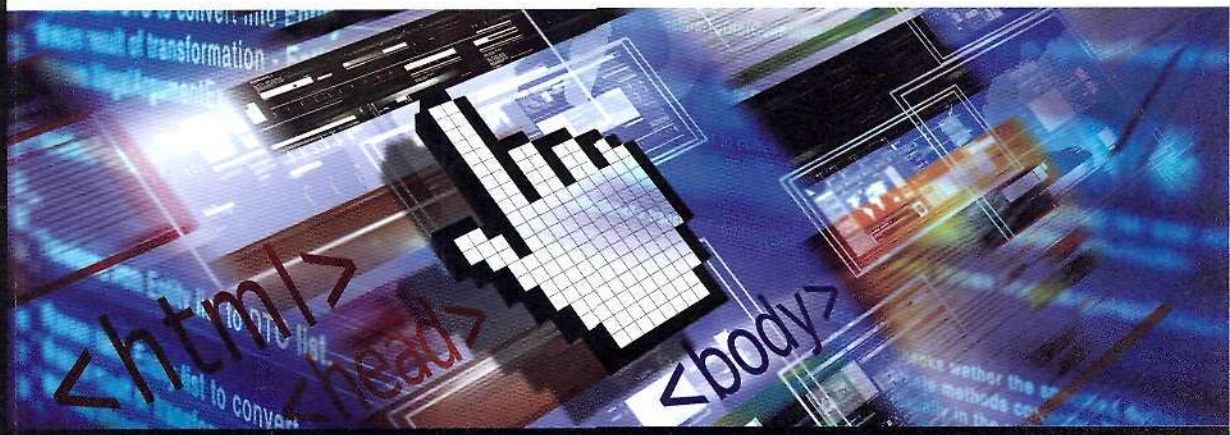
DO NOWEJ PODSTAWY PROGRAMOWEJ

Część 1

Kwalifikacja E.14
Tworzenie stron internetowych



Podręcznik do nauki zawodu **technik informatyk**

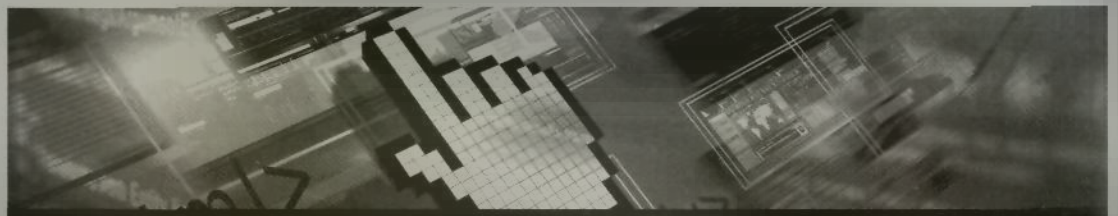
Jolanta Pokorska

Helion Edukacja

DO NOWEJ PODSTAWY PROGRAMOWEJ

Część 1

Kwalifikacja E.14
Tworzenie stron internetowych



Podręcznik do nauki zawodu
technik informatyk

Jolanta Pokorska

Podręcznik dopuszczony do użytku szkolnego przez ministra właściwego do spraw oświaty i wychowania i wpisany do wykazu podręczników przeznaczonych do kształcenia w zawodzie technik informatyk, na podstawie opinii rzeczoznawców: mgr Marii Dziurzyńskiej-Ścibior, mgr Elżbiety Leszczyńskiej, dr. inż. Stanisława Szabłowskiego.

Nazwa kwalifikacji: Kwalifikacja E-14. Część 1. Tworzenie stron internetowych.
Typ szkoły: technikum, szkoła policealna, kurs kwalifikacyjny.
Rok dopuszczenia 2013.
Numer ewidencyjny w wykazie: 13/2013

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiejkolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Redaktorzy prowadzący: Marcin Borecki
Projekt okładki: Maciej Pasek

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *http://helion.pl* (księgarnia internetowa, katalog książek)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
http://helion.pl/user/opinie?e14tei
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-246-5102-3

Copyright © Helion 2013

Printed in Poland.

Spis treści

Wstęp 7

Część 1. Tworzenie stron internetowych 11

Rozdział 1. Język HTML 13
1.1. Wprowadzenie 13
1.2. Języki HTML, XML, XHTML 15
1.3. Kodowanie polskich znaków 20
1.4. Edytory tekstu 21
1.5. Struktura strony WWW 23
1.6. Tworzenie i formatowanie dokumentu HTML 27

Rozdział 2. Kaskadowe arkusze stylów (CSS) 76
2.1. Wstawianie stylów 77
2.2. Składnia języka CSS 84
2.3. Selektory 87
2.4. Właściwości elementów 99
2.5. Model blokowy CSS 112
2.6. Inne elementy 122
2.7. Menu w języku CSS 139
2.8. Zasady projektowania szablonu strony internetowej 145

Rozdział 3. Edytory WYSIWYG 160
3.1. Edytory stron WWW 160
3.2. Adobe Dreamweaver 162

Rozdział 4. Zasady projektowania strony internetowej 185
4.1. Projektowanie stron 185

Rozdział 5. Grafika na stronie internetowej 192
5.1. Rodzaje grafiki 192
5.2. Sposoby zapisu obrazu cyfrowego 193
5.3. Metody pozyskiwania obrazów 195
5.4. Modele barw 196
5.5. Formaty plików graficznych 199
5.6. Ogólne zasady tworzenia grafiki na potrzeby strony internetowej 200
5.7. Modyfikowanie grafiki na potrzeby strony internetowej 201
5.8. Tworzenie grafiki na potrzeby strony internetowej 208
5.9. Tło strony internetowej 214
5.10. Ikona i logo jako elementy strony internetowej 220
5.11. Nawigacja 222
5.12. Mapowanie obrazu 224
5.13. Zapisywanie obrazów na potrzeby witryny internetowej 227

Rozdział 6. Graficzny projekt strony internetowej 232
6.1. Opracowanie projektu 232
6.2. Projektowanie strony 234
6.3. Dobór palety barw 238

Rozdział 7. Animacja na stronie internetowej 245
7.1. Metody tworzenia animacji 245
7.2. Tworzenie animacji na potrzeby strony internetowej 246

Rozdział 8. Dźwięk i wideo 271
8.1. Dźwięk 271
8.2. Edytowanie i obróbka dźwięku 274
8.3. Pliki wideo 281
8.4. Planowanie filmu 283
8.5. Tworzenie filmu w programie Adobe Premiere 284

Rozdział 9. Testowanie i publikowanie strony 293
9.1. Sprawdzanie poprawności strony 293
9.2. Walidacja strony 294
9.3. Testowanie strony w przeglądarkach 297
9.4. Publikowanie strony internetowej 298
9.5. Testy szybkości wczytywania strony 303
9.6. Testy funkcjonalności strony internetowej 306
9.7. Optymalizacja strony internetowej 308
9.8. Pozycjonowanie strony 308

Skorowidz 313

Wstęp

Podręcznik *Tworzenie stron internetowych* jest pierwszym z grupy podręczników przygotowanych dla kwalifikacji *E.14. Tworzenie aplikacji internetowych i baz danych oraz administrowanie bazami*. Kwalifikacja ta wraz z kwalifikacjami *E.12. Montaż i eksploatacja komputerów osobistych oraz urządzeń peryferyjnych* i *E.13. Projektowanie lokalnych sieci komputerowych i administrowanie sieciami* zawiera efekty kształcenia, których osiągnięcie jest niezbędne do uzyskania dyplomu potwierdzającego kwalifikacje w zawodzie technik informatyk.

Zawarte w podręczniku *Tworzenie stron internetowych* treści zostały oparte na podstawie programowej kształcenia w zawodzie technik informatyk 351203 wprowadzonej rozporządzeniem Ministra Edukacji Narodowej z dnia 7 lutego 2012 r. w sprawie podstawy programowej kształcenia w zawodach (Dz. U. 2012 nr 34 poz. 184).

Obejmują one zagadnienia teoretyczne prowadzące do uzyskania wymienionych w podstawie programowej efektów kształcenia, projekty różnych zadań oraz ich realizację praktyczną. Tak skonstruowany podręcznik pomaga uczniowi w zdobywaniu wymaganej wiedzy oraz umożliwia mu samodzielne poszerzanie umiejętności.

Podręcznik składa się z dziewięciu rozdziałów. Ich budowa pozwala na realizację treści programowych w sposób wybrany przez nauczyciela.

Rozdział 1., „Język HTML", zawiera omówienie zagadnień związanych z tworzeniem stron internetowych w języku HTML. Dotyczy efektów związanych z posługiwaniem się hipertekstowymi językami znaczników oraz efektów związanych z tworzeniem stron internetowych za pomocą tych języków. Efektami tymi są: rozpoznawanie pojęć HTML, XML i XHTML, stosowanie znaczników edycji tekstu, tabel, różnego rodzaju list, organizacji strony, definiowanie hiperłącza i mapy odsyłaczy, stosowanie znaczników osadzania obrazów, identyfikowanie elementów strony internetowej, planowanie rozmieszczenia elementów na stronie, wstawianie obiektów zewnętrznych takich jak obraz i dźwięk.

Rozdział 2., „Kaskadowe arkusze stylów (CSS)", zawiera omówienie zagadnień związanych z użyciem kaskadowych arkuszy stylów do tworzenia układu graficznego strony internetowej. Dotyczy efektów związanych z tworzeniem oraz wykorzystywaniem kaskadowych arkuszy stylów do opisu formy prezentacji strony WWW. Efektami tymi są: definiowanie ogólnej struktury arkusza CSS, tworzenie różnych arkuszy stylów, identyfikowanie i stosowanie atrybutów stylów do formatowania obiektów na stronie internetowej oraz identyfikowanie i stosowanie atrybutów definiujących opis formy prezentacji strony internetowej.

Rozdział 3., „Edytory WYSIWYG", zawiera omówienie zagadnień związanych z wykorzystaniem edytorów typu WYSIWYG do tworzenia stron internetowych. Dotyczy

efektów związanych z rozpoznawaniem funkcji edytorów spełniających założenia WYSIWYG oraz tworzeniem stron WWW za pomocą takich edytorów. Efektami tymi są: definiowanie założeń edytora WYSIWYG, rozróżnianie edytorów dostępnych na rynku, korzystanie z funkcji edytora WYSIWYG do tworzenia witryn.

Rozdział 4., „Zasady projektowania strony internetowej", zawiera omówienie zagadnień związanych z architekturą strony internetowej oraz zasadami jej projektowania. Dotyczy efektów związanych z projektowaniem struktury witryny WWW oraz wykonaniem strony internetowej zgodnie z projektem. Efektami tymi są: identyfikowanie i stosowanie zasad projektowania witryn internetowych, identyfikowanie i stosowanie zasad rozmieszczania elementów witryn oraz analizowanie projektu strony lub witryny internetowej.

Rozdział 5., „Grafika na stronie internetowej", zawiera omówienie zagadnień związanych z tworzeniem, modyfikowaniem i umieszczaniem grafiki na stronie internetowej. Dotyczy efektów związanych ze stosowaniem różnych modeli barw, z przestrzeganiem zasad cyfrowego zapisu obrazu, z tworzeniem grafiki statycznej, ze zmianą atrybutów obiektów graficznych i modyfikowaniem obiektów graficznych, a także z przetwarzaniem i przygotowaniem elementów graficznych do publikacji w internecie. Efektami tymi są: definiowanie kolorów za pomocą nazwy, kodu szesnastkowego, RGB; definiowanie pojęć „grafika rastrowa" i „grafika wektorowa", identyfikowanie formatów plików graficznych, tworzenie tekstur, ikon i obrazów statycznych według projektu, tworzenie przycisków i menu na potrzeby stron internetowych, zmienianie atrybutów obiektów graficznych, modyfikowanie obiektów graficznych, dobieranie formatów plików graficznych do publikacji w internecie.

Rozdział 6., „Graficzny projekt strony internetowej", zawiera omówienie zagadnień związanych z tworzeniem projektu graficznego strony WWW. Dotyczy efektów związanych z projektowaniem struktury witryny internetowej oraz z wykonaniem jej projektu graficznego. Efektami tymi są: projektowanie struktury witryny internetowej, dobieranie obrazów i elementów graficznych do koncepcji witryny, projektowanie witryny internetowej z wykorzystaniem różnych technik, stosowanie zasad doboru palety barw do projektowanych witryn.

Rozdział 7., „Animacja na stronie internetowej", zawiera omówienie zagadnień związanych z tworzeniem animacji na potrzeby strony WWW. Dotyczy efektów związanych z tworzeniem animacji jako elementów stron internetowych. Efektami tymi są: tworzenie animacji, a także animowanych przycisków i menu na potrzeby stron internetowych.

Rozdział 8., „Dźwięk i wideo", zawiera omówienie zagadnień związanych z edytowaniem i obróbką dźwięku oraz planowaniem i montażem filmu na potrzeby stron internetowych. Dotyczy efektów związanych z przetwarzaniem i przygotowaniem obrazu i dźwięku do publikacji w internecie oraz z przestrzeganiem zasad komputerowego przetwarzania obrazu i dźwięku. Efektami tymi są: dobieranie formatów plików multimedialnych do publikacji w sieci, modyfikowanie obrazów i dźwięków na potrzeby publikacji internetowych, identyfikowanie i przestrzeganie zasad komputerowego przetwarzania obrazu i dźwięku.

Rozdział 9., „Testowanie i publikowanie strony", zawiera omówienie zagadnień związanych z walidacją, testowaniem i publikowaniem witryny internetowej. Dotyczy efektów związanych ze stosowaniem reguł walidacji stron WWW oraz testowaniem i publikowaniem witryny internetowej. Efektami tymi są: definiowanie procesu walidacji strony, wykorzystywanie narzędzi do testowania stron internetowych, analizowanie wyników walidacji stron, testowanie stron internetowych w różnych przeglądarkach, publikowanie stron i witryn na serwerach WWW, korzystanie z klientów FTP.

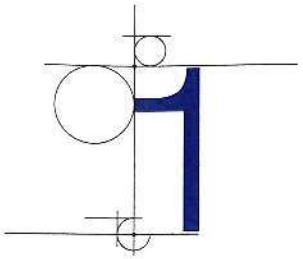
Część I

Tworzenie stron internetowych

Sieć WWW może być przerażającym miejscem, zwłaszcza jeśli nigdy nie mieliśmy odwagi, by spojrzeć na kod HTML typowej strony WWW. Kod HTML jest zapełniony ukrytymi polami, szalonymi identyfikatorami, tajemniczymi instrukcjami JavaScript i znacznikami HTML wyglądającymi jak po wyjęciu z miksera. Jest to cena za próbę ukrycia przed nami bezstanowej natury WWW.

Jeśli jednak opanujemy fundamentalne technologie WWW, to otrzymamy czysty, czytelny kod HTML, opatrzony stylami CSS i dynamicznymi zachowaniami zapewnianymi przez łatwy do zrozumienia język JavaScript.

James Kovacs



Język HTML

1.1. Wprowadzenie

Pierwsze strony internetowe pisane w języku HTML były bardzo ograniczone i zawierały jedynie podstawową strukturę w postaci nagłówków i paragrafów. Dawały także możliwość używania odnośników (linków). Klikając odnośniki, internauci mogli łatwo poruszać się między różnymi stronami.

W miarę rozwoju internetu język HTML zmieniał się, zapewniając większą złożoność i elastyczność. Pozwalał na dodawanie do stron różnego rodzaju obiektów, np. obrazów, tabel. Tabele zaczęto wykorzystywać jako elementy struktury i wyglądu strony. Następnym krokiem było wprowadzenie kaskadowych arkuszy stylów (CSS), co pozwoliło na oddzielenie warstwy strukturalnej i treści strony od warstwy prezentacyjnej.

Obecnie standardem są strony bogate w grafikę, zdjęcia cyfrowe, animacje, elementy interaktywne oraz w mniej czy bardziej profesjonalne filmy wideo.

Każda ze stron dostępnych w internecie jest zwykłym dokumentem tekstowym, który można stworzyć przy użyciu dowolnego edytora tekstu. Aby się przekonać, że strona internetowa to rzeczywiście dokument tekstowy, wystarczy w dowolnej przeglądarce internetowej otworzyć wybraną stronę, a następnie kliknąć prawym przyciskiem myszy i wybrać z menu opcję *Pokaż źródło*. Spowoduje to wyświetlenie strony w postaci dokumentu tekstowego. Teksty, które są dostępne na stronie internetowej, pojawią się otoczone znacznikami (tagami), czyli elementami języka **HTML** (ang. *HyperText Markup Language*), za pomocą którego tworzy się strony WWW.

Dokumenty HTML to pliki tekstowe, które zawierają proste polecenia dotyczące składu tekstu, a także dynamiczne odnośniki do innych dokumentów oraz obiekty postscriptowe, preformatowane i multimedialne (animacje, dźwięki, sekwencje wideo). Dokumenty te są przesyłane w sieci w postaci źródłowej — dopiero przeglądarka internetowa dostępna na komputerze użytkownika interpretuje kod strony. Dzięki temu możliwe jest uniezależnienie się od systemu operacyjnego i komputera, na którym wyświetlana jest strona.

Dokumenty HTML można przygotowywać, korzystając z dowolnego edytora tekstu.

Edytory możemy podzielić następująco:

- Edytory tekstowe — umożliwiają bezpośrednią pracę z kodem HTML. Często są wyposażone w dodatkowe funkcje, np.: automatyczne wstawianie zamykającego znacznika, sprawdzanie składni języka HTML, proste dodawanie atrybutów, łatwe tworzenie formularzy, tabel i ramek.
- Edytory graficzne (wizualne) — umożliwiają pracę w trybie WYSIWYG (ang. *What You See Is What You Get*). Pozwalają na tworzenie stron bez wpisywania kodu źródłowego.

1.1.1. Podstawowe pojęcia

Strona internetowa

Dokument utworzony w języku HTML lub XHTML, zapisany w pliku i umieszczony na serwerze, odczytywany na komputerze użytkownika za pomocą przeglądarki internetowej.

Witryna internetowa

Zbiór stron internetowych powiązanych tematycznie i umieszczonych na jednym serwerze, stanowiący rodzaj serwisu informacyjnego.

Portal internetowy

Serwis informacyjny zawierający informacje na różne tematy, zwykle wyposażony w dodatkowe funkcje takie jak: poczta elektroniczna, katalogi stron internetowych, wyszukiwarki, czat, pobieranie plików. Przykładami portali internetowych są Onet i Wirtualna Polska.

Wortal internetowy

Portal wertykalny (ang. *vertical portal*), czyli wyspecjalizowany, zawierający informacje z jednej dziedziny, tematycznie ze sobą powiązane. Nazwa „wortal" się nie przyjęła.

Serwer internetowy

Określenia tego używa się w odniesieniu do komputera podłączonego do internetu i świadczącego różne usługi w internecie oraz w odniesieniu do oprogramowania uruchamianego na tym komputerze. Przykłady: serwery FTP, DNS, HTTP.

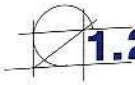
Statyczna strona internetowa

Strony statyczne to takie, których zawartość i wygląd nie zmieniają się przy każdym odwołaniu do nich. Aby dokonać zmian na stronie, programista musi zmienić jej kod.

Dynamiczna strona internetowa

Strony dynamiczne są generowane na bieżąco przez serwer HTTP na podstawie zmiennych i parametrów przekazanych przez przeglądarkę internetową. Takie strony zmieniają zawartość i wygląd w zależności od interakcji z użytkownikiem.

Zmiany zawartości strony mogą odbywać się w dwojaki sposób. Można dokonać tych zmian po stronie użytkownika, na jego komputerze (ang. *client-side*), przy użyciu języków skryptowych takich jak JavaScript, JScript, ActionScript, dokonujących zmian bezpośrednio na obiektach. Drugim sposobem jest kompilacja dynamicznej zawartości na serwerze (ang. *server-side*) przy użyciu języków programowania takich jak PHP, ASP, Perl, Java (JSP), Python.



1.2. Języki HTML, XML, XHTML

1.2.1. HTML (ang. HyperText Markup Language)

Podstawowym językiem projektowania stron internetowych jest HTML. Język ten rozwijał się intensywnie, aż stał się potężnym narzędziem do tworzenia stron WWW.

Coraz większą popularność zdobywa HTML w wersji 5., która wykorzystując standardową składnię języka, daje nowe możliwości. Aby zrozumieć nowe funkcje HTML 5, trzeba jednak poznać podstawowe funkcje HTML.

Podstawowym elementem języka HTML jest znacznik. Znaczniki są poleceniami umieszczonymi w nawiasach ostrokątnych, np.: `<body>`, `<head>`. Informują one przeglądarkę o wyglądzie otwieranej strony oraz o strukturze umieszczonego na niej tekstu. Nazwy poleceń mogą być pisane małymi lub wielkimi literami, nie ma to znaczenia. Wyróżnia się znaczniki otwierające, np. `<b>`, i zamykające, np. `</b>`. Znacznik zamykający zawsze rozpoczyna się znakiem / i ma taką samą nazwę jak znacznik otwierający. Pomiędzy tymi znacznikami może zostać umieszczony tekst, którego znaczniki dotyczą, np. wpisanie kodu `<b>Formatowanie tekstu</b>` spowoduje pogrubienie tekstu umieszczonego między znacznikami. Prawie wszystkie znaczniki otwierające trzeba zamknąć, stosując znaczniki zamykające. Wyjątki to np.:

- `<br>` — znacznik łamania wiersza,
- `<hr>` — znacznik linii poziomej,
- `<li>` — znacznik pozycji listy.

Znaczniki mają dodatkowe elementy zwane atrybutami, które definiują ich działanie. Jeden znacznik może posiadać wiele atrybutów. Atrybuty mogą przyjmować różne wartości nazywane parametrami. Parametrem może być np. wielkość, rodzaj lub kolor czcionki. Wartość atrybutu powinna zostać zapisana w cudzysłowie, ale język HTML dopuszcza zapisywanie wartości atrybutu bez cudzysłowu:

```
<nazwa_znacznika atrybut=wartość> zawartość </nazwa_znacznika>
```

lub

```
<nazwa_znacznika atrybut="wartość"> zawartość </nazwa_znacznika>
```

W języku HTML znaczniki mogą być zagnieżdżane, czyli umieszczane w innych znacznikach. Jednak nie wszystkie znaczniki mogą zawierać w sobie inne znaczniki.

Przykład 1.1

```
<p align="center"><font color="red"><i>Architektura stron<b> WWW</b></
i></font></p>
```

Znaczniki powinny być zamykane w kolejności odwrotnej do tej, w jakiej były otwierane. Najpierw powinien zostać zamknięty znacznik, który został otwarty jako ostatni, na końcu znacznik, który został otwarty jako pierwszy. Kolejność wypisywania znaczników i atrybutów jest dowolna.

W dokumencie tworzonym w języku HTML powinna być wstawiona deklaracja typu dokumentu (ang. *Document Type Definition* — DTD). Określa ona, jaka wersja języka HTML jest stosowana na stronie.

Przykład 1.2

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/
html4/strict.dtd">
```

1.2.2. XML (ang. Extensible Markup Language)

XML jest metajęzykiem opartym na zwykłym tekście, przeznaczonym do definiowania języków znakowania. Jest to coś więcej niż tekstowy format do opisywania dokumentów. XML daje dostęp do wielu technologii konstruowania, przetwarzania i przeszukiwania danych. Został stworzony w celu definiowania nowych formatów dokumentów w internecie. Dostarcza mechanizmów opisywania struktury dokumentu za pomocą znaczników.

XML jest powszechnie używany do opisu nowych formatów dokumentów na potrzeby internetu oraz do opisu danych posiadających wyraźną lub słabą wewnętrzną strukturę (arkuszy kalkulacyjnych, protokołów sieciowych, plików konfiguracyjnych).

Ze względu na to, że składnia XML jest oparta na zwykłym tekście, dokumenty XML mogą być czytane i edytowane przy użyciu najprostszych edytorów tekstu.

Język XML nie ma ustalonego słownika, dlatego można tworzyć słowniki przeznaczone dla określonych aplikacji lub branż. Opracowano już słowniki XML dla różnych gałęzi przemysłu i wielu zastosowań (np. dla dokumentów internetowych opracowano standard XHTML, a dla protokołów sieciowych opracowano standard SOAP, ang. *Simple Object Access Protocol* — protokół wywoływania zdalnego dostępu do obiektów). Aplikacje wykorzystujące formaty XML są odporniejsze na zmiany w strukturze danych niż aplikacje oparte na innych formatach.

XML nie jest związany z żadnym językiem programowania ani z żadnym systemem operacyjnym. Dzięki temu dokumenty można tworzyć i edytować, korzystając z różnych języków programowania, a XML może współpracować z różnymi systemami operacyjnymi.

Zapisane w języku XML dokumenty biznesowe i bazy danych mogą być udostępniane jako źródło wymiany informacji pomiędzy firmami lub są formą prezentacji danych wszędzie tam, gdzie wymagana jest współpraca różnych platform.

Składnia XML 1.0

Specyfikacja XML 1.0 określa tekstowy format do opisu danych przy użyciu składni podobnej do składni języka HTML.

Dokument jest zbudowany z elementów, a każdy element składa się ze znacznika otwierającego (np. `<table>`), znacznika zamykającego (np. `</table>`) oraz z informacji znajdujących się pomiędzy tymi znacznikami. Elementy mogą być opatrywane atrybutami zawierającymi metadane opisujące je oraz ich zawartość.

1.2.3. XHTML (ang. Extensible HyperText Markup Language)

Język XHTML to język, który dostosowuje zasady języka HTML 4 do specyfikacji języka XML. Opiera się na specyfikacji języka HTML 4.01, ale równocześnie wprowadza obostrzenia wynikające ze składni języka XML. Dokumenty pisane w tym języku są zgodne ze składnią XML, dlatego mogą być przeglądane, edytowane i sprawdzane za pomocą standardowych narzędzi XML. Strony pisane w XHTML są wyświetlane przez przeglądarki obsługujące zarówno język HTML, jak i XHTML.

Do dokumentu utworzonego w języku XHTML musi zostać wstawiona deklaracja typu dokumentu (DTD). Określa ona, jaka wersja języka HTML jest stosowana na stronie.

W XHTML 1.0 można wykorzystywać jedną z trzech wersji tej deklaracji.

Wersja Strict (ścisła) zawiera znaczniki i atrybuty, które nie są zdeprecjonowane i nie pojawiają się w dokumentach z ramkami; nie obejmuje również atrybutu `target=" "` oraz znacznika `<iframe>`.

Postać deklaracji dla wersji Strict:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.
w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Wersja Transitional (przejściowa) zawiera wszystkie elementy Strict, elementy zdeprecjonowane oraz znacznik `<iframe>` i atrybut `target=" "`.

Postać deklaracji dla wersji Transitional:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://
www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Wersja Frameset (ramkowa) zawiera wszystkie elementy Transitional oraz elementy odnoszące się do ramek (`frame`, `frameset`, `noframes`).

Postać deklaracji dla wersji Frameset:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.
w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

W wersji XHTML 1.1 stosowana jest tylko jedna wersja deklaracji:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1 //EN" "http://www.w3.org/TR/
xhtml11/DTD/xhtml11.dtd">
```

Deklaracja XML zarówno dla wersji XHTML 1.0, jak i dla wersji XHTML 1.1 ma postać:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Użycie jej nie jest wymagane, ale zaleca się umieszczanie tej deklaracji szczególnie wtedy, gdy kodowanie znaków jest inne niż UTF-8.



UWAGA

Znaczniki i atrybuty zdeprecjonowane (dotyczy to języków HTML i XHTML) to elementy, które są uznawane za przestarzałe. Zaleca się nieużywanie ich i korzystanie z nowszych technik, np. z arkuszy stylów CSS. Zdeprecjonowane znaczniki i atrybuty są dozwolone w wersji Transitional języków HTML 4.01 i XHTML 1.0, ale nie są dozwolone w wersji Strict.

Przykład 1.3
Oto przykład dokumentu XHTML z dołączoną deklaracją XML:

```
<?xml version="1.0" encoding="ISO-8859-2"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">
  <head>
    <title>Tytuł strony</title>
  </head>
  <body>
    Moja strona internetowa
  </body>
</html>
```



UWAGA

Niektóre przeglądarki nie rozpoznają deklaracji XML i mogą wyświetlać dokument w sposób inny od oczekiwanego.

1.2.4. Różnice między językami HTML 4 a XHTML 1.0

Język XHTML 1.0 wprowadza następujące obostrzenia w stosunku do języka HTML 4:

- Cały dokument musi być poprawnym kodem w języku XML.
- Znaczniki muszą być zamykane w kolejności odwrotnej do tej, w której były otwierane.
- Nazwy znaczników i atrybutów muszą być pisane małymi literami.

- Znaczniki zamykające są obowiązkowe.
- Wartości atrybutów muszą być ujęte w cudzysłowy.
- Nie można pomijać atrybutów logicznych.
- Elementy puste muszą mieć znacznik zamykający lub ich znacznik otwierający musi się kończyć sekwencją />, np.
.
- Wewnętrzny skrypt lub arkusz stylów może zawierać znaki <, >, &. Procesor XML potraktowałby je jako znaczniki, dlatego należy umieszczać te znaki w dokumencie w następujący sposób:

```
<script type="text/javascript">
<![CDATA[zawartość skryptu]]>
</script>
```

lub

```
<style type="text/css">
/* <![CDATA[  */ zawartość arkusza CSS/* ]]> */
</style>
```



UWAGA

Nie wszystkie przeglądarki obsługują podane metody, dlatego w celu zapewnienia zgodności z językiem XHTML skrypty i arkusze stylów zawierające znaki <, >, &, [,], -- należy umieszczać w zewnętrznym pliku.

1.2.5. HTML 5

Język HTML 5 jest rozwinięciem języków HTML 4 i XHTML 1. Jest on kompatybilny z poprzednimi wersjami języka HTML. W nowej wersji dopracowano specyfikację języka HTML 4 (przede wszystkim w zakresie obsługi błędów przez przeglądarki) oraz usprawniono narzędzia tworzenia serwisów i aplikacji internetowych. Pojawiły się nowe znaczniki i typy wejść oraz nowe atrybuty elementów formularzy. Wprowadzono obsługę wideo i dźwięku. Język HTML 5 został skonstruowany tak, że starsze przeglądarki bez problemu mogą ignorować jego nowe polecenia. Brak wsparcia ze strony przeglądarek internetowych i brak kompatybilności z wcześniejszymi wersjami spowodował odwrót od XHTML, więc wielkość liter w znacznikach i atrybutach nie ma znaczenia; również domykanie wszystkich znaczników nie jest konieczne. Nie ma znaczników i atrybutów deprecjonowanych, ale niektóre elementy są klasyfikowane jako przestarzałe i nie zaleca się ich używania. Mimo to wiele dokumentów pisanych w HTML zachowuje składnię typową dla XHTML ze względu na jasno określone reguły tego języka.

Standard HTML 5 upraszcza zapis deklaracji kilku elementów.

Deklaracja typu dokumentu ma postać:

```
<!DOCTYPE HTML>
```

Sposób kodowania znaków ma postać:

```
<meta charset="UTF-8"/>
```

Dołączenie arkusza stylów nie wymaga atrybutu type:

```
<link rel="stylesheet" href="file.css">
```

deklaracja JavaScript również nie wymaga atrybutu type:

```
<script src="file.js"></script>
```

1.3. Kodowanie polskich znaków

Język HTML umożliwia obsługę znaków różnych języków narodowych, w tym znaków języka polskiego. Aby było to możliwe, musi obsługiwać standardy kodowania tych znaków.

Pierwszym standardem, który obsługiwał polskie litery, był standard ISO 8859. Jest on zgodny z podstawowym standardem ASCII, ale poza zakresem zgodnym z ASCII występują konflikty między znakami reprezentującymi każdy bajt. Obecnie istnieje 15 wariantów tego standardu. Niektóre wersje Microsoft Windows przy kodowaniu polskich znaków korzystają ze standardu Windows-1250, który jest podobny do standardu ISO 8859-2 (występują niewielkie różnice, ale obydwa standardy zachowują zgodność z ASCII).

Powstały jeszcze inne standardy kodowania znaków narodowych, np. znaków koreańskich, japońskich, cyrylicy. Dodatkowo różne systemy operacyjne wykorzystywały inne systemy kodowania dla tych samych języków. Stosowanie różnych sposobów kodowania oraz często występujący brak zgodności standardów z ASCII uniemożliwiały komunikację.

Aby temu zaradzić, opracowano nowy sposób kodowania znaków — Unicode — który dla 17 zbiorów 16-bitowych pozwala zapisać 1 114 112 znaków. Jeden zbiór 16-bitowy zawiera 65 536 znaków, czyli więcej, niż może wykorzystać przeciętny użytkownik.

Najpopularniejsze wersje Unicode to UTF (ang. *Unicode Transformation Format*) i UCS (ang. *Universal Character Set*).

UTF-8 jest najpopularniejszą wersją Unicode ze względu na swoją zgodność z kodowaniem 8-bitowym. Wykorzystuje kodowanie ze zmienną długością znaku. Oznacza to, że każdy znak może być opisany przez 1 – 4 bajtów. Pierwszy bajt używany jest do kodowania ASCII, natomiast pozostałe bajty są wykorzystywane w zależności od stosowanego alfabetu.

UTF-8 to standard, który pozwala na pracę w środowisku wielojęzycznym i jest najprostszym sposobem przesyłania znaków spoza ASCII przez internet.

Obecnie w języku HTML powszechnie używa się kodowania UTF-8, co rozwiązuje problem wyświetlania różnych międzynarodowych znaków diakrytycznych. Również edytory tekstu zapisują dokumenty z wykorzystaniem Unicode.

W nagłówku dokumentu HTML można określić sposób kodowania znaków, definiując znacznik <meta>, np.:

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

lub

```
<meta charset="UTF-8">
```

lub

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

Można również stosować kodowanie ISO albo Windows:

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
```

lub

```
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1250">
```

1.4. Edytory tekstu

Do pisania kodu strony internetowej wystarczy najprostszy edytor tekstu, należy jednak wówczas pamiętać o kontrolowaniu składni języka HTML. W tworzonym dokumencie istotne znaczenie ma każdy nawias czy cudzysłów. Ich brak uniemożliwi przeglądarce prawidłowe zinterpretowanie zapisanego kodu. Aby ułatwić i przyspieszyć pracę z dokumentem, można skorzystać z wyspecjalizowanych edytorów tekstu, które mają wbudowane różne narzędzia, np.: autouzupełnianie, podpowiedzi, schematy kolorowania, pomoc kontekstową.

Najprostszym i zawsze dostępnym w systemie Windows edytorem tekstu jest Notatnik.

Podobny edytor tekstu dla systemu Linux to **Pico**, a dla systemu Macintosh — **Text**.

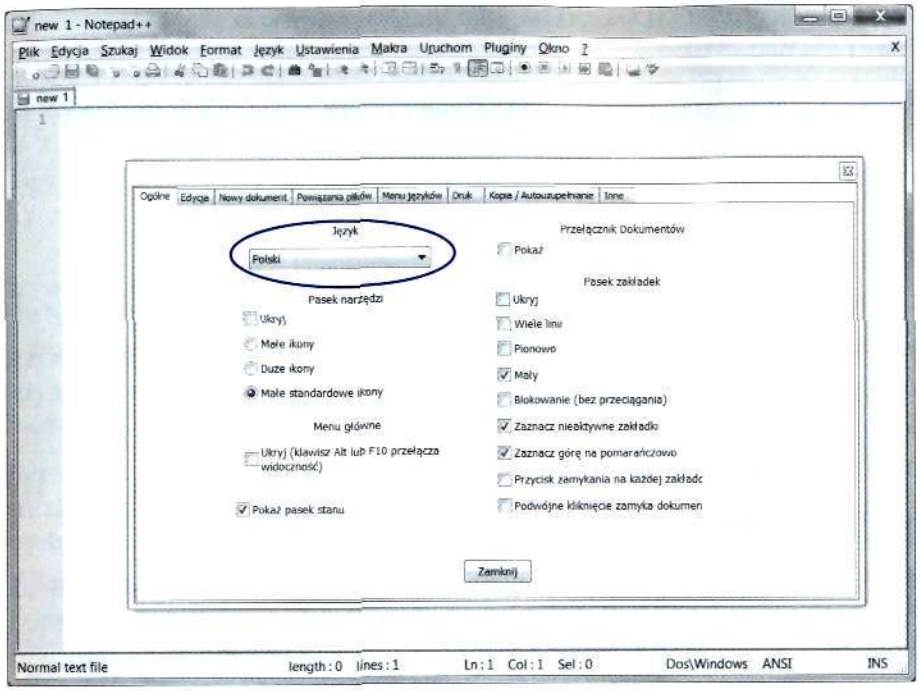
Inny edytor tekstu to **Notepad++**. Jest to bardzo dobry, bezpłatny edytor kodów źródłowych, dostępny w języku polskim i dający możliwość kodowania polskich znaków.

Prosty edytor tekstu jest najlepszym narzędziem do tworzenia dokumentów w językach HTML i CSS, ponieważ wprowadzony przez projektanta kod nie zostanie automatycznie zmieniony ani sformatowany.

Do testowania działania kodu można używać dowolnej przeglądarki internetowej, należy jedynie pamiętać o jej uaktualnieniu.

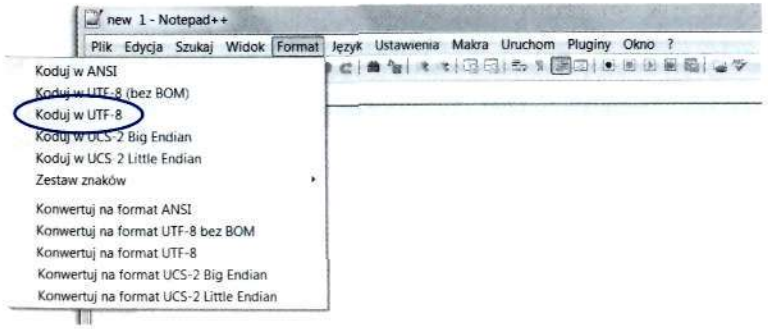
1.4.1. Notepad++

Plik instalacyjny programu Notepad++ można pobrać ze strony *http://notepad-plus-plus.org/*. Po zainstalowaniu programu należy ustawić jego polską wersję językową, wybierając *Settings/Preferences* i w zakładce *General* wskazując język Polski (rysunek 1.1).



Rysunek 1.1. Ustawienie polskiej wersji językowej w edytorze Notepad++

Aby polskie znaki były prawidłowo interpretowane, należy ustawić odpowiedni format kodowania. W tym celu trzeba w menu *Format* wybrać z dostępnej listy żądaną opcję. Najlepiej wybrać opcję *Koduj w UTF-8* (rysunek 1.2). Jest to standardowy i zalecany format kodowania polskich znaków.



Rysunek 1.2. Ustawienie opcji kodowania znaków

Zapisywanie plików

Pliki zawierające dokumenty HTML powinny być zapisywane z rozszerzeniem *.html* lub *.htm*.

Plik o nazwie *index.html* (lub *index.htm*) domyślnie jest traktowany jako strona startowa dla danej witryny. Jeśli plik o takiej nazwie znajduje się w folderze z witryną, to właśnie

jego zawartość zostanie automatycznie wyświetlona po wywołaniu jej adresu. Z tego powodu należy stronę startową witryny zawsze nazywać *index.html* (lub *index.htm*) i umieszczać ją w katalogu głównym konta. W nazwie nie można używać wielkich liter.

UWAGA

Jeżeli w folderze z witryną znajdują się dwa pliki — jeden o nazwie *index.html*, drugi o nazwie *index.htm* — to dla domyślnej konfiguracji serwera HTTP plik *index.html* ma pierwszeństwo wyświetlania przed plikiem *index.htm*, a więc będzie on pełnił funkcję pliku startowego.

1.5. Struktura strony WWW

Każdy dokument HTML składa się ze znaczników, które mówią przeglądarce, w jaki sposób ma wyświetlać teksty, obrazy, animacje, filmy. Znaczniki są słowami lub znakami umieszczonymi wewnątrz nawiasów < i >. Elementów umieszczonych poza nawiasami przeglądarka nie potrafi zinterpretować (są to te elementy, które będą wyświetlane na ekranie). Znaczniki są poleceniami wysyłanymi do przeglądarki i składają się z dwóch części: polecenia otwierającego `<nazwa polecenia>` i polecenia zamykającego `</nazwa polecenia>`, a pomiędzy nimi znajduje się odpowiednia treść.

Przykład 1.4

```
<b> Moja pierwsza strona </b>
```

Wpisanie podanego fragmentu kodu spowoduje pogrubienie tekstu zawartego między znacznikami i .

Znaczniki mogą zawierać atrybuty.

Przykład 1.5

```
<BODY BGCOLOR="#00ccff">
Moja pierwsza strona
</BODY>
```

Po wpisaniu podanego fragmentu kodu zostanie ustawiony kolor tła strony.

Każdy dokument tekstowy powinien zaczynać się od znacznika <html> i kończyć się znacznikiem </html>. Znaczniki te informują przeglądarkę, że wszystko zawarte między nimi ma być interpretowane jako strona internetowa. Oprócz tego każda strona internetowa zawiera nagłówek (<head> i </head>) oraz ciało (<body> i </body>), w którym zawarta jest treść dokumentu. W sekcji <head> powinien pojawić się tytuł strony umieszczony w znacznikach <title> i </title>. Przed sekcją <html> powinien zostać umieszczony jeszcze znacznik zawierający informacje na temat standardu języka, na podstawie którego tworzymy stronę internetową, a w sekcji <head> powinien się znajdować znacznik <meta>, który przekaże do przeglądarki informację, że to, co ona odczyta z sekcji <body>, będzie dokumentem HTML, oraz że kodowanie znaków zostanie sformatowane w standardzie UTF-8 (dzięki temu, gdy pojawią się polskie litery, przeglądarka prawidłowo je zinterpretuje).

Przykład 1.6

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">
  <head>
    <title>
      ...
    </title>
    ...
  </head>
  <body>
    ...
  </body>
</html>
```

Podany przykład pokazuje elementy, z których składa się każda strona internetowa. Część nagłówkową, by nie tracić czasu na jej tworzenie, można przepisać z istniejącego dokumentu HTML. Należy jednak wiedzieć, jak jest zbudowana każda strona internetowa i jak zinterpretować treść dokumentu tekstowego. Po sprawdzeniu kodu tworzonej strony można dokonać modyfikacji jej zawartości i ręcznie dopisać nowe fragmenty.

Przykład 1.7

```
<!DOCTYPE html>
<html lang="pl-PL">
<head>
<meta charset="UTF-8">
<title>Tytuł strony</title>
</head>
<body>
Treść strony
</body>
</html>
```

Podany wyżej przykład pokazuje szablon dokumentu w HTML5.

1.5.1. Nagłówek

Nagłówek dokumentu jest zawarty między znacznikami `<head>` i `</head>`. W nim znajduje się informacja o dokumencie HTML.

Najważniejsze polecenia umieszczane w nagłówku to znacznik `<title>` określający tytuł dokumentu oraz znacznik `<meta>`, który przechowuje informacje o dokumencie i umożliwia wyszukiwarkom indeksowanie stron.

Znacznik `<title>` służy do wyświetlania tytułu strony w zakładce przeglądarki (rysunek 1.3) i ma postać:

```
<title>Tytuł strony</title>
```

Rysunek 1.3. Tytuł strony wyświetlany na karcie przeglądarki



Znaczniki `<meta>` informują, kto jest autorem strony, kiedy strona została utworzona, jakie zastosowano kodowanie znaków, a ponadto decydują o miejscu strony w wynikach wyszukiwania. Występują z trzema atrybutami: `http-equiv`, `name`, `content`, które mogą przyjmować różne wartości.

UWAGA

Atrybuty `http-equiv` i `name` nie mogą występować w tym samym znaczniku.

Atrybut `http-equiv` dostarcza przeglądarce internetowej informacji na temat otwieranej strony i steruje jej zachowaniem. Może przyjmować następujące wartości:

- `Content-Type` — określa sposób kodowania znaków.
- `Content-Language` — określa język zawartości dokumentu.
- `Reply-to` — podaje adres e-mail do kontaktowania się z autorem.

Atrybut `name` pozwala umieszczać niezbędne dla wyszukiwarek informacje dotyczące strony internetowej. Może przyjmować następujące wartości:

- `Title`, `page-topic` — ustawiają tytuł i opis strony.
- `Creation_Date` — ustawia datę utworzenia strony.
- `Keywords` — definiuje słowa kluczowe, które opisują zawartość strony.
- `Description` — definiuje opis strony widoczny w wynikach wyszukiwania.
- `Robots` — określa, czy dana strona powinna być indeksowana.

Przykład 1.8

```
<meta http-equiv="refresh" content="x">
```

Takie użycie znacznika spowoduje automatyczne odświeżanie dokumentu co x sekund.

```
<meta http-equiv="refresh" content="x; url=adres">
```

Tak użyty znacznik spowoduje przekierowanie użytkownika po upływie x sekund do podanego adresu URL.

```
<meta http-equiv="reply-to" content="adres e-mail">
```

Tak użyty znacznik podaje adres e-mail autora strony.

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

Tak użyty znacznik podaje sposób kodowania znaków.

```
<meta http-equiv="Content-Language" content="pl">
```

Tak użyty znacznik definiuje język zawartości dokumentu.

```
<meta name="title" content="tytuł strony">
```

Tak użyty znacznik definiuje tytuł strony lub jej opis.

```
<meta name="Keywords" content="aplikacje internetowe, serwery WWW">
```

Tak użyty znacznik definiuje słowa kluczowe, które opisują zawartość strony.

Zadanie 1.1

Przeanalizuj podany niżej fragment kodu dokumentu HTML. Jakie parametry definiują poszczególne znaczniki?

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://
www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="title" content="Aplikacje internetowe - wszystko
o aplikacjach internetowych" />
<meta name="robots" content="index, follow" />
<meta http-equiv="Content-Language" content="pl" />
<meta name="description" content="To najlepsza strona o aplikacjach
internetowych" />
<meta name="keywords" content="PHP, HTML, XML, JavaScript" />
<meta name="rating" content="General" />
<meta http-equiv="Reply-to" content="inform@gmail.com" />
<meta name="revisit-after" content="7 days" />
<meta http-equiv="refresh" content="25" />
</head>
<body>
</body>
</html>
```

1.5.2. Ciało dokumentu

Ciało dokumentu to jego część zawarta między znacznikami `<body>` i `</body>`. Zawiera ono właściwą treść dokumentu pokazywaną użytkownikowi przez przeglądarkę. Znacznik `<body>` ma wiele atrybutów, które definiują wygląd dokumentu.

Zmianę koloru tła strony można uzyskać, definiując atrybut `bgcolor`.

```
<body bgcolor="kolor">
```


Przykład 1.9

```
<html>
<head>
</head>
<body bgcolor="#FFCCFF">
Moja strona internetowa
</body>
</html>
```

Za pomocą atrybutu `background` można wybrać obraz lub zdjęcie, które stanie się tłem strony.

```
<body background="obraz.jpg">
```

Przykład 1.10

```
<html>
<head>
</head>
<body background="obraz.jpg">
Moja strona internetowa
</body>
</html>
```

1.6. Tworzenie i formatowanie dokumentu HTML

Dokument HTML najczęściej zawiera elementy różnego rodzaju. Każdy z nich może być w dowolny sposób formatowany. Formatowanie elementów dokumentu wpływa na wygląd projektowanej strony internetowej.

1.6.1. Formatowanie tekstu

Podstawowymi poleceniami języka HTML są znaczniki formatowania tekstu (określania kroju, wielkości, koloru i stylu czcionki).

Zmiana rodzaju czcionki

Zmianę rodzaju czcionki umożliwia znacznik `<font>`.

```
<font face="nazwa_czcionki">
```

Przykład 1.11

```
<font face="Times New Roman">Formatowanie dokumentu HTML</font>
```

UWAGA

Znacznika `<font>` można używać również do zmiany koloru i rozmiaru tekstu.

Przykład 1.12

```
<font size="12" face="Times New Roman" color="red">Formatowanie dokumentu
HTML</font>
```

Pogrubienie i kursywa

Za pogrubienie czcionki odpowiada znacznik <b>.

Przykład 1.13

```
<b> Architektura stron WWW</b>
```

Znacznik <i> służy do wypisania tekstu kursywą.

Przykład 1.14

```
<i> Architektura stron WWW</i>
```

Zmiana rozmiaru czcionki

Rozmiar czcionki całego tekstu zmienia znacznik <basefont>, np. <basefont size ="n">, gdzie *n* jest liczbą z przedziału 1 – 7 (wartość domyślna wynosi 3).

Przykład 1.15

```
<basefont size ="6">
```

UWAGA

Znacznik <basefont> nie jest obsługiwany w HTML5.

Aby zmienić rozmiar fragmentu tekstu, możemy zastosować dwie metody. Pierwsza to użycie znacznika <font size="n"> i podanie określonej wielkości tekstu. Druga to określenie, czy wybrany fragment będzie większy, czy mniejszy od pozostałego tekstu. W drugiej metodzie wykorzystywane są znaczniki <big> i <small>.

Przykład 1.16

```
Wprowadzenie do <big>języka</big> HTML
Wprowadzenie do <small>języka</small> HTML
```

Wynik działania fragmentów kodu, podanych w przykładach 1.11-1.16, pokazuje rysunek 1.4.



Rysunek 1.4. Wynik formatowania tekstu w kodzie HTML

Kolor tekstu

Aby zmienić kolor całego tekstu, należy w znaczniku <body> użyć atrybutu text.

```
<body text="kolor">
```

Przykład 1.17

```
<body text="green">
```

Do zmiany koloru fragmentu tekstu służy znacznik <font> z atrybutem color.

```
<font color="kolor">
```

Przykład 1.18

```
<font color="blue">Tekst wyświetlony na niebiesko</font>
<font color="#cfcfcf">Kolorowy tekst</font>
```

UWAGA

Znacznik <font color="kolor"> ma wyższy priorytet niż znacznik <body text="kolor">.

Indeks dolny i indeks górny

Do definiowania indeksu górnego używamy znacznika <sup>, a do wprowadzenia indeksu dolnego — znacznika <sub>.

Przykład 1.19

```
Parametr X<sub>2</sub>
Powierzchnia wynosi 20 m<sup>2</sup>
```

Przekreślenia i podkreślenia tekstu

Znacznik <strike> spowoduje przekreślenie tekstu, a znacznik <u> jego podkreślenie.

Przykład 1.20

```
<strike>Czy tekst zostanie przekreślony?</strike>
<u>Czy tekst zostanie podkreślony?</u>
```

Pozioma linia

Do narysowania i wyświetlenia na ekranie poziomej linii używamy znacznika <hr>. Używając odpowiednich atrybutów, możemy określić grubość, długość lub kolor rysowanej linii.

Zwykła linia:

```
<hr />
```

Linia o grubości 20 pikseli:

```
<hr size="20" />
```

Linia o długości 200 pikseli:

```
<hr width="200" />
```

Linia w kolorze zielonym:

```
<hr color="green" />
```

Przykład 1.21

```
<hr size="15" width="300" color="green" align="center" />
```

Znaki specjalne

Ciąg spacji umieszczony w tekście jest zawsze interpretowany przez przeglądarkę jako jedna spacja. Dlatego w celu wyświetlenia dłuższych odstępów należy użyć znaków specjalnych: ` `.

Przykład 1.22

```
<b><u><i> Komputer         to mój przyjaciel</i>
</u></b>
```

Jeżeli na stronie internetowej będą wyświetlane znaki specjalne, powinny one zostać wpisane w projekcie strony jako nazwy lub numery znaków. Wpisanie znaku bezpośrednio z klawiatury może zostać błędnie zinterpretowane przez przeglądarkę. Najważniejsze znaki specjalne to: <, > oraz &. Przeglądarka odczyta je jako kod języka HTML. Nazwy popularnych znaków specjalnych zostały podane w tabeli 1.1.

Tabela 1.1. Tabela znaków specjalnych

Znak specjalny	Nazwa	Opis
&	`&`	znak „and”
<	`<`	znak mniejszości
>	`>`	znak większości
§	`§`	znak sekcji lub paragrafu
™	`™`	zastrzeżony znak towarowy
©	`©`	znak praw autorskich
®	`®`	zastrzeżony znak towarowy
€	`€`	znak „euro”

Przykład 1.23

```
<b><u><i><font size="4"> Zastrzeżony znak towarowy może mieć symbol
&trade; lub &reg;</font></i></u></b>
```

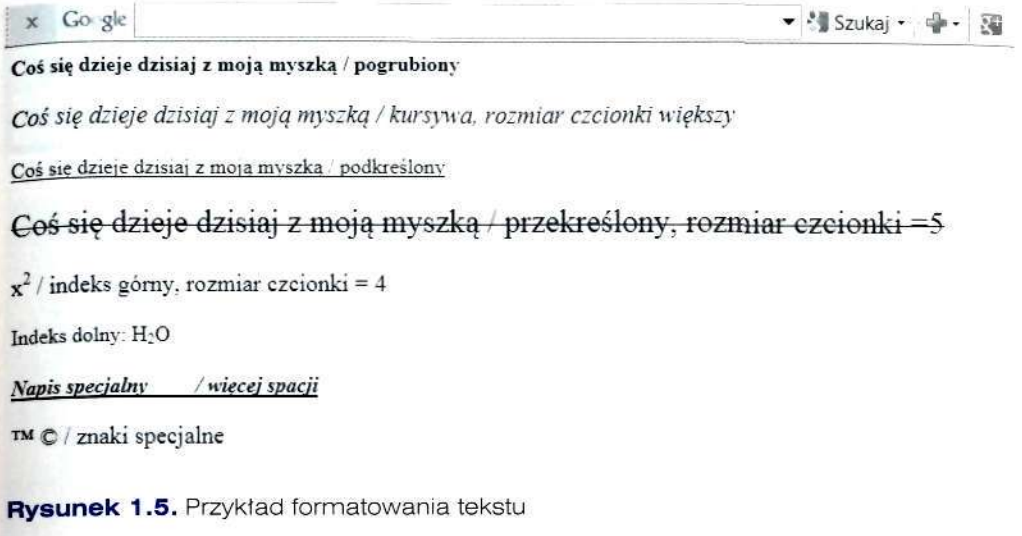
Komentarze

W pracy z dokumentem HTML powinny być używane komentarze, które ułatwią analizowanie jego kodu. Komentarze są wyświetlane tylko w kodzie dokumentu i nie są widoczne dla użytkownika strony internetowej. Umieszcza się je w następujący sposób:

```
<!--Komentarz do mojej strony internetowej-->
```

Zadanie 1.2

Przy użyciu języka HTML napisz kod programu, który wyświetli w przeglądarce internetowej podany poniżej tekst. Wyświetlony tekst powinien zostać sformatowany w taki sposób, aby wyglądem nie różnił się od widocznego na rysunku 1.5.



Rysunek 1.5. Przykład formatowania tekstu

1.6.2. Układ strony dokumentu

Akapit

Tekst umieszczony na stronie może być grupowany w akapity. Akapit powinien zostać otoczony znacznikami i . Akapity oznaczone w ten sposób będą oddzielane w tekście pustym wierszem. Zapisywanie tekstu w postaci akapitów jest wygodną metodą formatowania jego fragmentów. Wystarczy zdefiniować atrybuty znacznika , żeby cały tekst zapisany jako akapit został odpowiednio sformatowany.

Przykład 1.24

```
<p> W pracy z dokumentem HTML powinny być używane komentarze.</p>
<p> Komentarze nie są widoczne dla użytkownika.</p>
```

Tekst zapisany w akapicie może być na stronie różnie wyrównywany. Sposób wyrównania w akapicie określamy, używając atrybutu `align`.

Wyrównanie do lewej (stosowane domyślnie):

```
<p align="left"> W pracy z dokumentem HTML powinny być używane
komentarze. Komentarze nie są widoczne dla użytkownika.</p>
```

Wyrównanie do prawej:

```
<p align="right"> W pracy z dokumentem HTML powinny być używane
komentarze. Komentarze nie są widoczne dla użytkownika.</p>
```

Wyśrodkowanie:

```
<p align="center"> W pracy z dokumentem HTML powinny być używane
komentarze. Komentarze nie są widoczne dla użytkownika.</p>
```

Wyjustowanie:

```
<p align="justify"> W pracy z dokumentem HTML powinny być używane
komentarze. Komentarze nie są widoczne dla użytkownika.</p>.
```

UWAGA

Atrybut `align` jest nadal akceptowany przez przeglądarki, ale zgodnie ze specyfikacją HTML 4.01 zaleca się stosowanie odpowiednich poleceń stylów CSS.

Nagłówki

Język HTML umożliwia tworzenie nagłówków w sześciu rozmiarach. Do definiowania nagłówka używa się znacznika `<hx>`, gdzie *x* może przyjmować wartości z zakresu 1 – 6.

Przykład 1.25

```
<h1> Język HTML</h1>
<h2> Język HTML</h2>
<h3> Język HTML</h3>
<h4> Język HTML</h4>
```

Aby zachować spójność dokumentu przy dzieleniu tekstu na rozdziały, podrozdziały i punkty, należy stosować nagłówki w odpowiedniej kolejności, np. nagłówek `h2` powinien zostać poprzedzony nagłówkiem `h1`, a nagłówek `h3` — nagłówkiem `h2`.

Wyśrodkowanie tekstu na stronie

Do wyśrodkowania tekstu służy znacznik `<center>`.

Przykład 1.26

```
<center>Tekst na środku strony</center>
```

Nowa linia

Aby wymusić wyświetlanie tekstu w nowej linii, należy użyć znacznika `<br>`.

Przykład 1.27

```
Język HTML <br> Język HTML <br> Język HTML <br> Język HTML
```

Znacznik łamania wiersza jako jeden z nielicznych nie wymaga znacznika zamykającego.

Interlinia

Do określenia odstępu między kolejnymi liniami tekstu używa się znacznika `<spacer>` w postaci `<spacer size="n">`, gdzie *n* oznacza wielkość odstępu między liniami podaną w pikselach.

Przykład 1.28

```
<spacer size="15">
```

W znaczniku `<spacer>` można użyć atrybutu `type="vertical"`, który spowoduje automatyczne łamanie linii.

Przykład 1.29

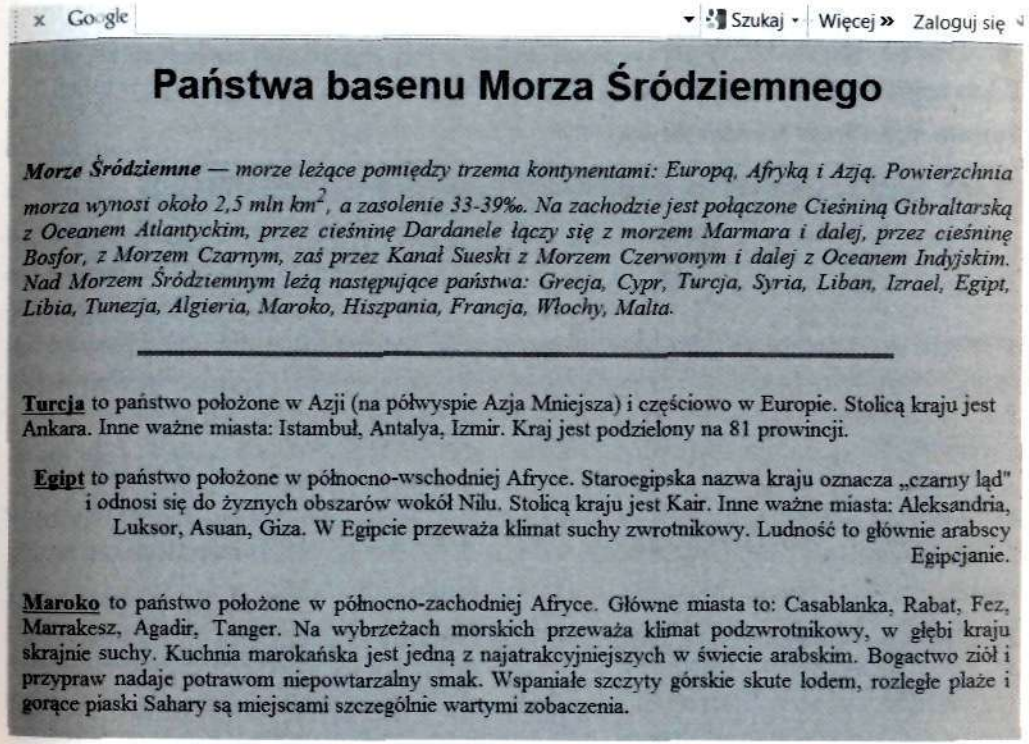
```
<spacer type="vertical" size="15">
```

UWAGA

Nie wszystkie przeglądarki internetowe poprawnie interpretują znacznik `<spacer>`.

Zadanie 1.3

Za pomocą języka HTML zaprojektuj swoją pierwszą stronę internetową. Strona ma zawierać podany poniżej tekst. Wyświetlony w przeglądarce internetowej tekst nie powinien różnić się wyglądem od widocznego na rysunku 1.6.



Rysunek 1.6. Formatowanie tekstu na stronie internetowej

1.6.3. Kolory

Niektóre polecenia języka HTML zawierają definicję kolorów, np.

```
<font color="red"> ... </font>
```

lub

```
<body bgcolor="#ffccff"> ... </body>
```

Wartość dla atrybutu `color` może być podawana na trzy sposoby. Można stosować słowną angielską nazwę koloru (`blue`, `red`) albo wprowadzić odpowiednik koloru zapisany w kodzie heksadecymalnym (`#ffcc00`, `#aa2237`) lub w kodzie RGB (`rgb(120,12,27)`, `rgb(39,21,110)`).

Znak `#` przed kodem koloru oznacza, że kolor jest definiowany właśnie w kodzie heksadecymalnym. Kod musi składać się z sześciu cyfr, nawet jeśli są to same zera. Im większa jest wartość liczbowa kodu, tym jaśniejszy kolor.

W modelu RGB kod koloru zawiera trzy składowe zapisane w postaci liczb oddzielonych przecinkiem. Napis `rgb` przed kodem koloru oznacza definiowanie koloru w kodzie RGB.

Poza kolorami zapisywanymi w systemie szesnastkowym można stosować nazwy kolorów. Lista dostępnych nazw jest ograniczona. W języku HTML zdefiniowano ich 16. Czasami stosuje się rozszerzoną paletę kolorów nazwanych, ale wykracza ona poza specyfikację języka HTML.

Paleta kolorów zgodna ze specyfikacją języka HTML i XHTML została podana w tabeli 1.2.

Tabela 1.2. Paleta kolorów języka HTML

Kolor	Nazwa	Kod HEX	Kod RGB
czarny	`black`	`#000000`	`rgb(0,0,0)`
srebrny	`silver`	`#c0c0c0`	`rgb(192,192,192)`
szary	`gray`	`#808080`	`rgb(128,128,128)`
biały	`white`	`#ffffff`	`rgb(255,255,255)`
kasztanowy	`maroon`	`#800000`	`rgb(128,0,0)`
czerwony	`red`	`#ff0000`	`rgb(255,0,0)`
purpurowy	`purple`	`#800080`	`rgb(128,0,128)`
fuksja	`fuchsia`	`#ff00ff`	`rgb(255,0,255)`
zielony	`green`	`#008000`	`rgb(0,128,0)`
limonkowy	`lime`	`#00ff00`	`rgb(0,255,0)`
oliwkowy	`olive`	`#808000`	`rgb(128,128,0)`
żółty	`yellow`	`#ffff00`	`rgb(255,255,0)`
granatowy	`navy`	`#000080`	`rgb(0,0,128)`
niebieski	`blue`	`#0000ff`	`rgb(0,0,255)`
morski	`teal`	`#008080`	`rgb(0,128,128)`
akwamaryna	`aqua`	`#00ffff`	`rgb(0,255,255)`

Więcej informacji o barwach można znaleźć na stronie *http://www.barwy.net/*.

W HTML5 wiele znaczników odpowiedzialnych za prezentację zostało uznane za przestarzałe. Mimo, że są one nadal używane w wielu istniejących stronach internetowych, powinny z czasem zostać zastąpione przez style CSS. Zdarza się, że strony tworzone przy użyciu graficznych edytorów WYSIWYG zawierają znaczniki i atrybuty uznane w specyfikacji HTML5 za zdeprecjonowane. Do takich znaczników należą:

- basefont,
- big,

- center,
- font,
- strike,
- u.

Oprócz znaczników zostały również unieważnione niektóre atrybuty. Dotyczy to atrybutów związanych z prezentacją treści, takich jak:

- align,
- bgcolor,
- valign,
- hspace,
- vspace.

Zadanie 1.4
Ile maksymalnie kolorów można uzyskać, stosując do definiowania koloru kod heksadecymalny?

1.6.4. Listy

W języku HTML można definiować listy numerowane, listy punktowane, listy zagnieżdżone oraz listy definicji.

Listy numerowane
Do zdefiniowania listy numerowanej stosowany jest znacznik `<ol>`, natomiast elementy listy są wprowadzane za pomocą znacznika `<li>`. Znacznik `<ol>` może zawierać atrybut `type="x"`, określający sposób wyliczania listy, oraz atrybut `start="n"`, określający wartość rozpoczynającą wyliczanie. Znaczniki `<li>` mogą zawierać atrybut `type="x"`, określający sposób wyliczania danego elementu listy, oraz atrybut `value="n"`, określający wartość, od której rozpocznie się wyliczanie elementów.

Ustawienie atrybutów `type` i `value` dla znacznika `<li>` unieważnia atrybuty `type` i `start` ustawione dla znacznika `<ol>`.

Lista według dużych liter:

```
<ol type="A">
```

Lista według małych liter:

```
<ol type="a">
```

Lista według cyfr rzymskich:

```
<ol type="I">
```

Lista według cyfr arabskich:

```
<ol type="1">
```

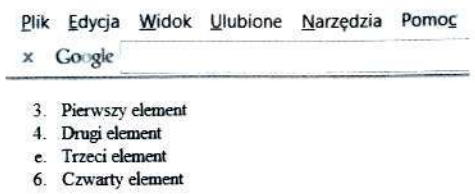
Przykład 1.30

```
<ol type="1" start="3">
   <li> Pierwszy element
```

```
    <li> Drugi element
    <li type="a"> Trzeci element
    <li> Czwarty element
</ol>
```

Wynik interpretacji kodu został pokazany na rysunku 1.7.

Rysunek 1.7. Lista numerowana



Listy punktowane

Listy punktowane tworzy się przy użyciu znacznika `<ul>`, a elementy listy, podobnie jak w liście numerowanej, są tworzone za pomocą znacznika `<li>`. Rodzaj punktora jest określany atrybutem `type="kształt"`, który może przyjmować wartości:

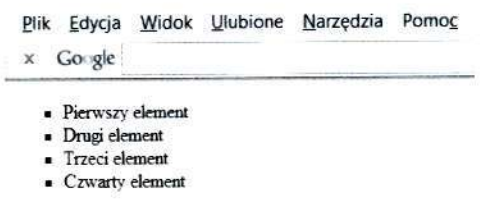
- `disc` — wypełnione koło,
- `square` — kwadrat,
- `circle` — niewypełnione koło.

Przykład 1.31

```
<ul type="square" >
   <li> Pierwszy element
   <li> Drugi element
   <li> Trzeci element
   <li> Czwarty element
</ul>
```

Wynik interpretacji kodu został pokazany na rysunku 1.8.

Rysunek 1.8. Lista punktowana. Typ punktora — kwadrat

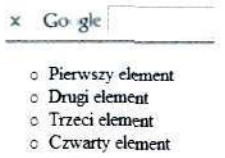


Przykład 1.32

```
<ul type="circle">
<li> Pierwszy element
<li> Drugi element
<li> Trzeci element
<li> Czwarty element
</ul>
```


Wynik interpretacji kodu został pokazany na rysunku 1.9.

Rysunek 1.9. Lista punktowana. Typ punktora — niewypełnione koło



Listy zagnieżdżone

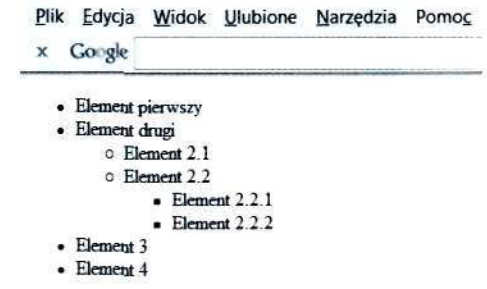
Zagnieżdżanie list to umieszczanie jednej listy wewnątrz innej. Takie listy tworzy się najczęściej jako listy wielopoziomowe. Przy zagnieżdżaniu można stosować kilka poziomów, jednak każdy kolejny poziom musi zawierać własną definicję listy.

Przykład 1.33

```
<ul><!--Lista wielopoziomowa - pierwszy poziom-->
  <li>Element pierwszy
  <li>Element drugi
  <ul><!--Lista wielopoziomowa - drugi poziom-->
    <li>Element 2.1
    <li>Element 2.2
    <ul><!--Lista wielopoziomowa - trzeci poziom-->
      <li>Element 2.2.1
      <li>Element 2.2.2
    </ul><!--Koniec listy na trzecim poziomie-->
  </ul><!--Koniec listy na drugim poziomie-->
  <li>Element 3
  <li>Element 4
</ul>
```

Wynik interpretacji kodu został pokazany na rysunku 1.10.

Rysunek 1.10. Lista zagnieżdżona



Listy definicji

Listy definicji mogą być przydatne do tworzenia słowników, wyjaśnień i opisów różnych terminów. Tworzy się je przy użyciu znaczników `<dl>`, `<dt>` i `<dd>`.

```
<dl>
   <dt> Pierwszy termin
      <dd> Opis pierwszego terminu
```

```
    <dt> Drugi termin
        <dd> Opis drugiego terminu
</dl>
```

Przykład 1.34

```
<dl>
    <dt> Znaczniki HTML
        <dd> Znaczniki w HTML są poleceniami pisanymi pomiędzy znakami &lt;
&gt;.
    <dt> Atrybuty
        <dd> Atrybuty definiują parametry znaczników. Pozwalają doprecyzować
ich działanie.<br/>Można wpisać wiele atrybutów do jednego znacznika,
pisząc je jeden po drugim oddzielone spacją.
</dl>
```

Wynik interpretacji kodu został pokazany na rysunku 1.11.

Plik Edycja Widok Ulubione Narzędzia Pomoc

x Google Szukaj

Znaczniki HTML

Znaczniki w HTML są poleceniami pisanymi pomiędzy znakami <>.

Atrybuty

Atrybuty definiują parametry znaczników. Pozwalają doprecyzować ich działanie.
Można wpisać wiele atrybutów do jednego znacznika, pisząc je jeden po drugim oddzielone spacją.

Rysunek 1.11. Lista definicji

Zadanie 1.5

1. Zaprojektowaną w zadaniu 1.3 stronę internetową zmodyfikuj, aby wyglądała tak jak na rysunku 1.12.

Rysunek 1.12. Tekst sformatowany z użyciem list

x Google Szukaj Więcej» Zaloguj się

Państwa basenu Morza Śródziemnego

Morze Śródziemne — morze leżące pomiędzy trzema kontynentami: Europą, Afryką i Azją. Powierzchnia morza wynosi około 2,5 mln km², a zasolenie 33-39‰. Na zachodzie jest połączone Cieśniną Gibraltarską z Oceanem Atlantyckim, przez cieśninę Dardanele łączy się z morzem Marmara i dalej, przez cieśninę Bosfor, z Morzem Czarnym, zaś przez Kanał Sueski z Morzem Czerwonym i dalej z Oceanem Indyjskim. Nad Morzem Śródziemnym leżą między innymi:

- *Grecja,*
- *Cypr,*
- *Turcja,*
- *Izrael,*
- *Egipt,*
- *Tunezja,*
- *Maroko,*
- *Hiszpania.*

- **Turcja** to państwo położone w Azji (na półwyspie Azja Mniejsza) i częściowo w Europie. Stolicą kraju jest Ankara. Inne ważne miasta: Istambuł, Antalya, Izmir. Kraj jest podzielony na 81 prowincji.
- **Egipt** to państwo położone w północno-wschodniej Afryce. Staroegipska nazwa kraju oznacza „czarny ląd" i odnosi się do żyznych obszarów wokół Nilu. Stolicą kraju jest Kair. Inne ważne miasta: Aleksandria, Luksor, Asuan, Giza. W Egipcie przeważa klimat suchy zwrotnikowy. Ludność to głównie arabscy Egipcjanie.
- **Maroko** to państwo położone w północno-zachodniej Afryce. Główne miasta to: Casablanka, Rabat, Fez, Marrakesz, Agadir, Tanger. Na wybrzeżach morskich przeważa klimat podzwrotnikowy, w głębi kraju skrajnie suchy. Kuchnia marokańska jest jedną z najstrakcyjniejszych w świecie arabskim. Bogactwo ziół i przypraw nadaje potrawom niepowtarzalny smak. Wspaniałe szczyty górskie skute lodem, rozległe plaże i gorące piaski Sahary są miejscami szczególnie wartymi zobaczenia.

2. Zmień wygląd swojej strony. Przy opisywaniu państw użyj *Listy definicji*.

1.6.5. Tabele

Przez długi czas do projektowania układu strony internetowej stosowano tabele. Poprzez zaprojektowanie odpowiedniego układu tabeli otrzymywało się stały układ strony. Każda komórka tabeli zawierała określone dane. Mogły to być tekst, grafika, formularz lub inny obiekt. Obecnie coraz rzadziej stosuje się tę metodę.

Tabele składają się z wierszy. Każdy wiersz ma określoną liczbę komórek, dlatego tworząc tabelę w języku HTML, należy nie tylko zdefiniować ją samą za pomocą znacznika `<table>`, ale także zdefiniować każdy jej wiersz przy użyciu znacznika `<tr>` oraz każdą komórkę w wierszu przy użyciu znacznika `<td>`.

Przykład 1.35

```
<table>
<tr><td>komórka 1-1</td><td>komórka 1-2</td><td>komórka 1-3</td></tr>
<tr><td>komórka 2-1</td><td>komórka 2-2</td><td>komórka 2-3</td></tr>
<tr><td>komórka 3-1</td><td>komórka 3-2</td><td>komórka 3-3</td></tr>
<tr><td>komórka 4-1</td><td>komórka 4-2</td><td>komórka 4-3</td></tr>
</table>
```

W wyniku zastosowanych poleceń powstała tabela składająca się z trzech kolumn i czterech wierszy.

Rozmiar tabeli i komórki

Jeżeli nie zdefiniujemy szerokości tabeli, przeglądarka automatycznie dostosuje jej szerokość do zawartości. Używając atrybutu `width="n"`, możemy sami zdefiniować szerokość zarówno tabeli, jak i poszczególnych komórek. Atrybut ten można umieścić w znaczniku `<table>` lub `<td>`.

Przykład 1.36

```
<table border= "6" bordercolor="blue" width="400">
<tr><td>komórka 1-1</td><td>komórka 1-2</td><td>komórka 1-3</td></tr>
<tr><td>komórka 2-1</td><td>komórka 2-2</td><td>komórka 2-3</td></tr>
<tr><td>komórka 3-1</td><td>komórka 3-2</td><td>komórka 3-3</td></tr>
<tr><td>komórka 4-1</td><td>komórka 4-2</td><td>komórka 4-3</td></tr>
</table>
```

Wynik interpretacji kodu został pokazany na rysunku 1.13.

Rysunek 1.13. Definiowanie tabeli

x Google

komórka 1-1	komórka 1-2	komórka 1-3
komórka 2-1	komórka 2-2	komórka 2-3
komórka 3-1	komórka 3-2	komórka 3-3
komórka 4-1	komórka 4-2	komórka 4-3

Szerokość komórki lub tabeli może zostać podana jako wartość procentowa szerokości całej tabeli (dla komórki) lub jako wartość procentowa szerokości okna przeglądarki (dla tabeli). W tym drugim przypadku zmiana szerokości okna przeglądarki prowadzi do zmiany szerokości tabeli.

Używając atrybutu `height="n"`, można zdefiniować wysokość tabeli lub wysokość komórek. Podobnie jak atrybut `width`, można go stosować w znacznikach `<table>` i `<td>`.

Tabela otoczona tekstem

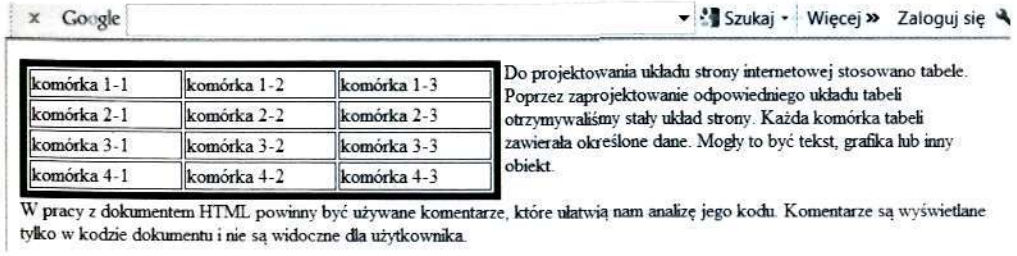
Tabela może zostać otoczona tekstem, który został zapisany poniżej niej. Aby otoczyć tabelę tekstem, należy dla znacznika `<table>` zdefiniować atrybut `align`. Atrybut ten może przyjmować następujące wartości:

- `left` — tabela zostanie wyrównana do lewej strony, a tekst otoczy ją z prawej.
- `right` — tabela zostanie wyrównana do prawej strony, a tekst otoczy ją z lewej.
- `all` — tekst zostanie wyświetlony wokół tabeli.

Przykład 1.37

```
<table align="left" border= "6" bordercolor="blue" width="400">
<tr><td>komórka 1-1</td><td>komórka 1-2</td><td>komórka 1-3</td></tr>
<tr><td>komórka 2-1</td><td>komórka 2-2</td><td>komórka 2-3</td></tr>
<tr><td>komórka 3-1</td><td>komórka 3-2</td><td>komórka 3-3</td></tr>
<tr><td>komórka 4-1</td><td>komórka 4-2</td><td>komórka 4-3</td></tr>
</table>
Do projektowania układu strony internetowej stosowano tabele. Poprzez
zaprojektowanie odpowiedniego układu tabeli otrzymywaliśmy stały układ
strony. Każda komórka tabeli zawierała określone dane. Mogły to być
tekst, grafika lub inny obiekt.
<br clear="left">
W pracy z dokumentem HTML powinny być używane komentarze, które ułatwią
nam analizę jego kodu. Komentarze są wyświetlane tylko w kodzie dokumentu
i nie są widoczne dla użytkownika.
```

Wynik interpretacji kodu został pokazany na rysunku 1.14.



Rysunek 1.14. Tabela otoczona tekstem

W przykładzie należy zwrócić uwagę na linię kodu `<br clear="left">`, która powoduje przerwanie wyświetlania tekstu wokół tabeli. Gdy zmieni się wartość atrybutu `align` w znaczniku `<table>`, również wartość atrybutu `clear` w znaczniku `<br>` musi zostać zmodyfikowana.

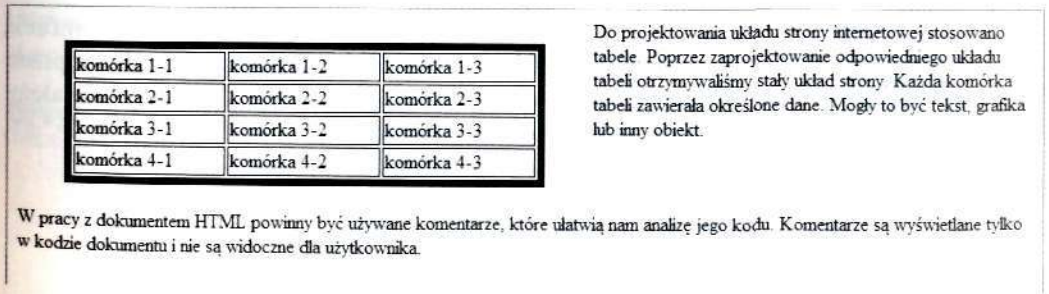
Wolny obszar wokół tabeli

Tabelę można oddzielić od innych elementów strony, dodając wokół niej wolną przestrzeń. Do dodania wolnego miejsca z prawej lub z lewej strony tabeli służy umieszczony w znaczniku `<table>` atrybut `hspace="x"`, gdzie x to szerokość dodanego obszaru w pikselach. Wolne miejsce u góry lub u dołu tabeli uzyskamy, stosując w znaczniku `<table>` atrybut `vspace="y"`. Wartość y to wysokość wolnego miejsca w pikselach.

Przykład 1.38

```
<table align="left" border="6" bordercolor="blue" width="400" hspace="40"
vspace="20">
<tr><td>komórka 1-1</td><td>komórka 1-2</td><td>komórka 1-3</td></tr>
<tr><td>komórka 2-1</td><td>komórka 2-2</td><td>komórka 2-3</td></tr>
<tr><td>komórka 3-1</td><td>komórka 3-2</td><td>komórka 3-3</td></tr>
<tr><td>komórka 4-1</td><td>komórka 4-2</td><td>komórka 4-3</td></tr>
</table>
Do projektowania układu strony internetowej stosowano tabele. Poprzez
zaprojektowanie odpowiedniego układu tabeli otrzymywaliśmy stały układ
strony. Każda komórka tabeli zawierała określone dane. Mogły to być
tekst, grafika lub inny obiekt.
<br clear="left">
W pracy z dokumentem HTML powinny być używane komentarze, które ułatwią
nam analizę jego kodu. Komentarze są wyświetlane tylko w kodzie dokumentu
i nie są widoczne dla użytkownika.
```

Wynik interpretacji kodu został pokazany na rysunku 1.15.

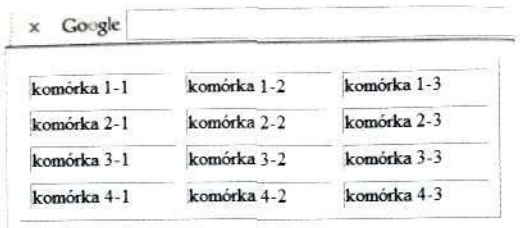


Rysunek 1.15. Wolny obszar wokół tabeli

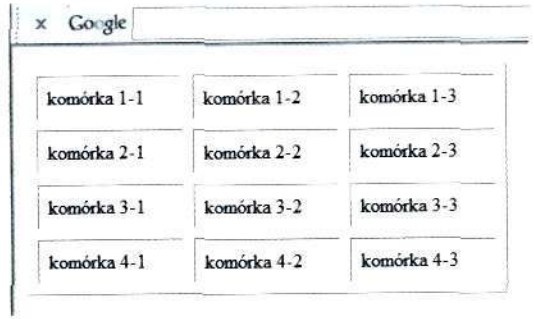
Odstępy między komórkami

Odstępy między komórkami tworzącymi tabelę uzyskamy, stosując dla znacznika `<table>` atrybut `cellspacing="n"` (rysunek 1.16), natomiast atrybut `cellpadding="n"` ustawi w komórkach tabeli wewnętrzne marginesy (rysunek 1.17).

Rysunek 1.16.
Definiowanie odstępów między komórkami



Rysunek 1.17.
Definiowanie wewnętrznych marginesów dla komórek



Zadanie 1.6

Zmodyfikuj kod strony z przykładu 1.36 (rysunek 1.13) tak, aby otrzymać tabelę widoczną na rysunku 1.16. Ponownie zmień kod, aby uzyskać tabelę pokazaną na rysunku 1.17.

Łączenie komórek

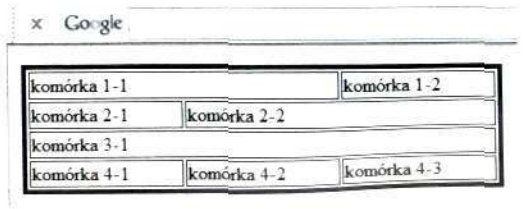
W języku HTML oprócz tabel, które mają w każdym wierszu taką samą liczbę komórek, można tworzyć tabele z różną liczbą komórek w wierszu. Efekt ten uzyskujemy poprzez łączenie komórek z sąsiednich kolumn. Aby połączyć kilka komórek w wierszu, należy w znaczniku `<td>` opisującym komórkę tabeli zdefiniować atrybut `colspan="n"`, gdzie `n` określa, ile sąsiednich komórek zostanie połączonych.

Przykład 1.39

```
<table border="4" bordercolor="blue" width="400">
<tr><td colspan="2">komórka 1-1</td><td>komórka 1-2</td></tr>
<tr><td>komórka 2-1</td><td colspan="2">komórka 2-2</td></tr>
<tr><td colspan="3">komórka 3-1</td></tr>
<tr><td>komórka 4-1</td><td>komórka 4-2</td><td>komórka 4-3</td></tr>
</table>
```

Wynik interpretacji kodu został pokazany na rysunku 1.18.

Rysunek 1.18.
Wynik łączenia komórek w wierszach



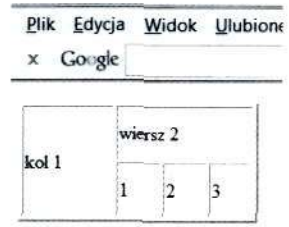
Łączenie komórek tabeli może odbywać się również w kolumnach. Łączone są komórki tej samej kolumny znajdujące się w sąsiednich wierszach. Do takiego łączenia komórek służy definiowany w znaczniku `<td>` atrybut `rowspan="n"`.

Przykład 1.40

```
<table border="2" width="200" height="100">
<tr><td rowspan="2">kol 1</td><td colspan="3">wiersz 2</td></tr>
<tr><td>1</td><td>2</td><td>3</td></tr>
</table>
```

Wynik interpretacji kodu został pokazany na rysunku 1.19.

Rysunek 1.19.
Wynik łączenia komórek w kolumnach

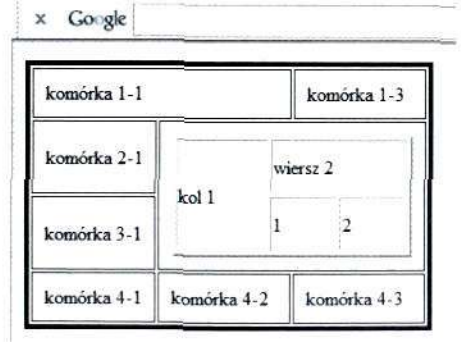


Istnieje również możliwość tworzenia tabel zagnieżdżonych, czyli umieszczania wewnątrz komórki tabeli innej tabeli. Przy takim działaniu należy zachować szczególną ostrożność i podczas wpisywania znaczników pilnować, aby każdemu znacznikowi otwierającemu odpowiadał znacznik zamykający.

Zadanie 1.7

Używając kodu języka HTML, zaprojektuj tabelę, która będzie wyświetlana w przeglądarce dokładnie w taki sposób jak tabela pokazana na rysunku 1.20.

Rysunek 1.20.
Tabele zagnieżdżone



Wyrównywanie zawartości komórek

Zawartość komórki tabeli domyślnie jest wyrównywana w poziomie do lewej krawędzi, a w pionie do środka. Definiowanie własnego wyrównania zawartości komórki odbywa się za pomocą atrybutów `align` (w poziomie) i `valign` (w pionie).

`<td align="kierunek">` — parametr *kierunek* może przyjmować wartości: `left`, `center`, `right`.

`<td valign="kierunek">` — parametr *kierunek* może przyjmować wartości: `top`, `middle`, `bottom`.

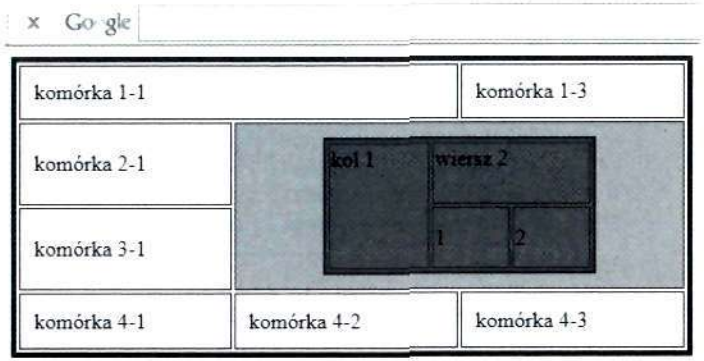
Kolor tła komórki
Do ustawienia koloru tła komórki (lub kilku komórek) służy atrybut `bgcolor` znacznika `<td>`.

Przykład 1.41

```
<td bgcolor="#ff66ff">Jaki to kolor?</td>
```

Zadanie 1.8
Zmodyfikuj kod programu z zadania 1.7 w taki sposób, aby tabela wyświetlana w przeglądarce wyglądała dokładnie tak jak tabela pokazana na rysunku 1.21.

Rysunek 1.21.
Formatowanie tabeli



1.6.6. Bloki

Aby wydzielić większy fragment tekstu, możemy za pomocą znacznika `<div>` zdefiniować blok.

Przykład 1.42

```
<div>Aby zachować spójność dokumentu przy dzieleniu tekstu na rozdziały,
podrozdziały i punkty, należy nagłówki stosować w odpowiedniej
kolejności.</div>
```

Podobnie jak w przypadku akapitu, tekst umieszczony w bloku można formatować; np. za pomocą atrybutu `align` tekst zostanie odpowiednio wyrównany.

Przykład 1.43

```
<div align="center">Aby zachować spójność dokumentu przy dzieleniu
tekstu na rozdziały, podrozdziały i punkty, należy stosować nagłówki
w odpowiedniej kolejności.</div>
```

1.6.7. Grafika

Obecnie każda strona internetowa zawiera elementy graficzne. Najczęściej są one zapisane w formatach GIF, JPEG i PNG.

Format GIF
GIF to format 8-bitowy pozwalający na zapisanie maksymalnie 256 kolorów. Najlepiej nadaje się do zapisywania elementów graficznych zawierających duże obszary o tym

samym kolorze i niewielkiej liczbie szczegółów i dlatego jest wykorzystywany przy projektowaniu nagłówków, przycisków, pasków i prostej grafiki. Główną zaletą tego formatu jest niewielki rozmiar plików. GIF ma również zaletę bardzo przydatną w projektowaniu stron internetowych, mianowicie umożliwia tworzenie grafik transparentnych (przezroczystych). Potrzebne jest to przy tworzeniu grafiki, której tło należy ukryć (np. przycisku o owalnym kształcie). W tym przypadku wystarczy uczynić transparentnym tło grafiki. Kolejna właściwość formatu GIF to możliwość stosowania przeplotu. Oznacza ona, że obraz jest wczytywany fragmentami. Początkowo wczytywana jest co czwarta linia obrazu, co powoduje, że od razu widoczny jest jego zarys. Kolejne przejścia to odczyt kolejnych linii — obraz stopniowo nabiera właściwego wyglądu. Daje to wrażenie szybszego wczytywania strony.

W formacie GIF można również zastosować mechanizm redukcji palety kolorów. Redukcja palety kolorów polega na zmniejszeniu liczby barw, które zapisujemy w pliku. Decyzja o liczbie kolorów w palecie należy do użytkownika i jest zależna od liczby kolorów wykorzystywanych w tworzonym obrazie.

Format JPG
W formacie JPG najpierw tworzony jest obraz czarno-biały, a informacje o kolorach są zapisywane osobno. Na powstałym obrazie jest przeprowadzana analiza dotycząca barw, a kolory, które są niewidoczne dla oka, zostają usunięte. W formacie JPG można regulować stopień kompresji, co powoduje zmniejszenie rozmiaru pliku, ale może prowadzić do pogorszenia jakości obrazu. Ponieważ w tym formacie zapisywana jest pełna, 24-bitowa paleta kolorów, JPG nadaje się doskonale do zapisywania fotografii.

Format PNG
Format PNG pozwala na zapisanie 48-bitowej palety kolorów i 16-bitowej palety odcieni szarości. Umożliwia tworzenie grafik transparentnych i ma efektywny i bezstratny mechanizm kompresji.

Wstawianie grafiki
Do umieszczania obrazów na stronie internetowej służy znacznik `<img>` z atrybutem `src="lokalizacja"`, za pomocą którego definiujemy nazwę i położenie pliku z grafiką.

Przykład 1.44

```
<img src="C:\obrazek.gif">
```

Jeżeli obrazek znajduje się w tej samej lokalizacji co strona, wystarczy podać nazwę pliku; jeżeli obrazek znajduje się w innej lokalizacji, należy podać pełny adres URL.

Stosując dla znacznika `<img>` atrybuty `height="x"` i `width="y"`, można określić wysokość i szerokość obrazka, zaś atrybut `border="n"` odpowiada za grubość obramowania.

Przykład 1.45

```
<img src="C:\obrazek.gif" width="180" height="220" border="2">
```

Podobnie jak w tabelach, używając atrybutów `hspace="x"` i `vspace="y"`, można oddzielić grafikę od innych elementów strony.

Przykład 1.46

```
<img src="C:\obrazek.gif" width="180" height="220" border="2" hspace="40"
vspace="20">
```

Jeśli obrazek z jakiegoś powodu nie jest wyświetlany na stronie (brak obrazka w lokalizacji, przeglądarka nie obsługuje grafiki, wyświetlanie obrazów zostało wyłączone), zamiast grafiki mogą zostać wyświetlone teksty zastępcze. Do projektowania tekstów zastępczych służy atrybut `alt="tekst zastępczy"`.

Przykład 1.47

```
<img src="C:\obrazek.gif" width="180" height="220" border="2" hspace="40"
vspace="20" alt="tekst zastępujący obrazek">
```

W przypadku obrazów o dużych rozmiarach istnieje możliwość utworzenia ich miniatur. Pozwoli to na szybkie wyświetlenie zawartości, szczególnie gdy nie korzystamy z szybkich łączy internetowych.

Przykład 1.48

```
<a href="obrazek.gif"> <img src="min_obrazek.gif"></a>
```

Znacznik `<a>` tworzy łącze do adresu URL podanego jako wartość atrybutu href. Wewnątrz znacznika `<a>` został umieszczony znacznik `<img>`, który wyświetli miniaturę obrazka na stronie. Kliknięcie widocznej na stronie miniatury obrazka spowoduje przejście do strony o podanym adresie URL.

Grafika i tekst

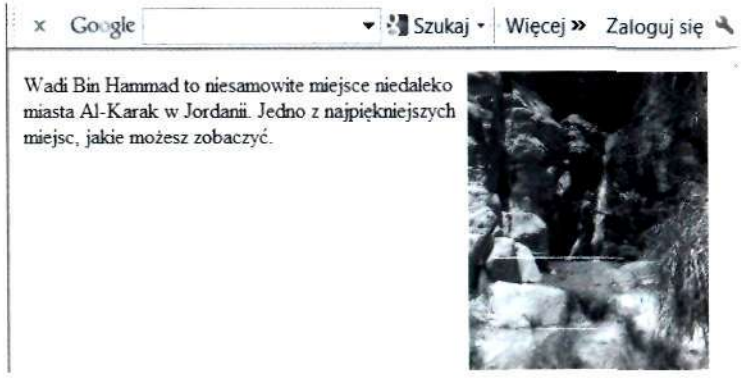
Używając atrybutu `align`, możemy wyświetlić obraz otoczony tekstem.

Przykład 1.49

```
<img src="Wadi Bin Hamman.jpg" align="right" width="180" height="220" >
Wadi Bin Hammad to niesamowite miejsce niedaleko miasta Al-Karak
w Jordanii. Jedno z najpiękniejszych miejsc, jakie możesz zobaczyć.
```

Tekst zostanie wyrównany do prawej krawędzi i otoczy obraz z jego lewej strony.

Wynik interpretacji kodu został pokazany na rysunku 1.22.



Rysunek 1.22. Wstawianie grafiki obok tekstu

Wartość atrybutu `align="left"` spowoduje wyrównanie obrazu do lewej krawędzi, zaś tekst otoczy go z prawej strony.

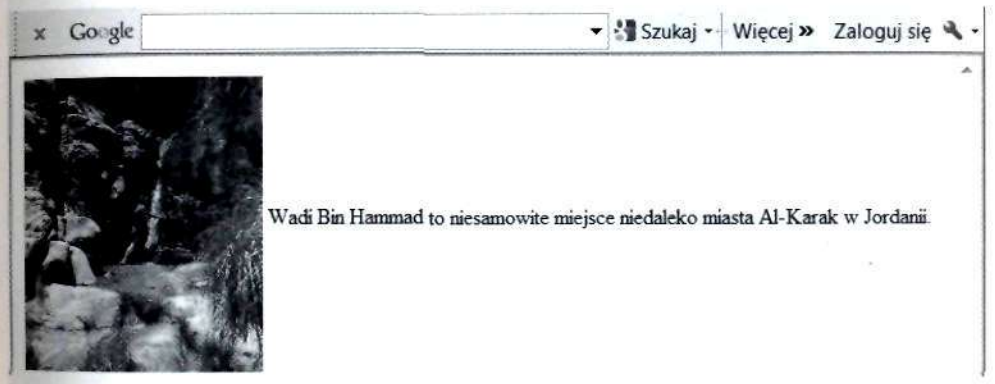
Atrybut `align` może być również wykorzystany do wyrównywania obrazka względem tekstu. Do takiego rodzaju wyrównania używane są następujące wartości:

- `texttop` — obraz zostanie wyrównany do najwyższego tekstu w linii;
- `top` — obraz zostanie wyrównany do najwyższego elementu w linii;
- `middle` — środek obrazu zostanie wyrównany do linii bazowej;
- `bottom` lub `baseline` — dolna krawędź obrazu zostanie wyrównana do linii bazowej.

UWAGA

Obrazu nie można równocześnie wyrównywać względem tekstu i otaczać tekstem.

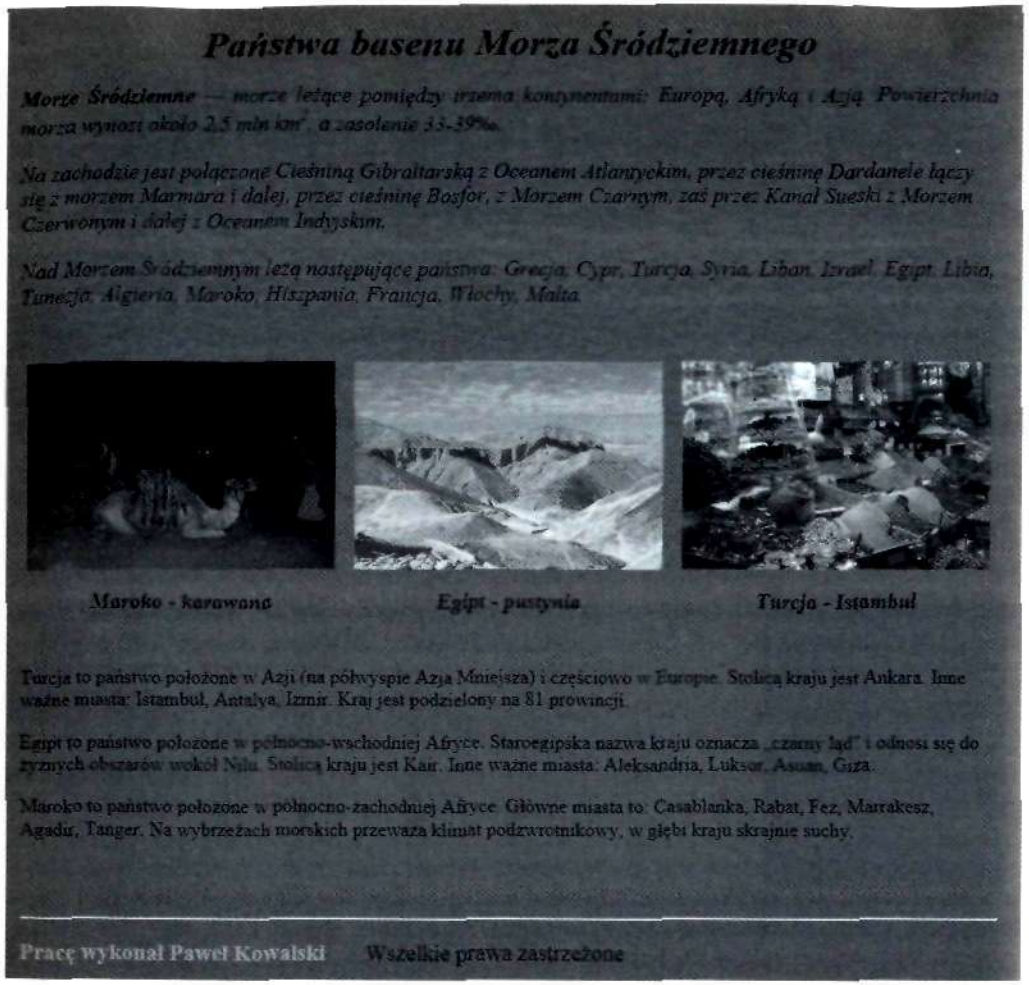
Na rysunku 1.23 pokazano zastosowanie atrybutu `align="middle"`.



Rysunek 1.23. Wyrównanie obrazu względem tekstu

Zadanie 1.9

Przeredaguj stronę internetową utworzoną w zadaniu 1.5. Zaprojektuj układ strony, wykorzystując tabele i narzędzia związane z projektowaniem tabel. Dodaj w odpowiednich miejscach zdjęcia. Strona internetowa powinna wyglądać podobnie jak pokazana na rysunku 1.24.



Rysunek 1.24. Strona internetowa w układzie tabelarycznym

1.6.8. Odsyłacze (hiperłącza)

Częstym elementem stron internetowych są odsyłacze (hiperłącza) prowadzące do innych stron. Mogą one przenosić użytkownika do dowolnej innej strony WWW, ale mogą również prowadzić do elementu na tej samej stronie lub do dowolnego dokumentu bieżącej witryny.

Definiowany odsyłacz ma postać:

```
<a href="adres URL"> Tekst połączenia</a>
```

Atrybut `href="adres URL"` określa, do jakiego dokumentu lub do jakiej strony prowadzi łącze. Tekst odsyłacza będzie widoczny na stronie ze zdefiniowanym połączeniem. Kliknięcie odsyłacza przeniesie użytkownika do wybranej strony lub do dokumentu.

Oto przykłady odsyłaczy:

- do innych dokumentów — `<a href="plik.html">Dokument</a>`;
- do pliku z grafiką — `<a href="grafika.jpg">Rysunek</a>`;
- do innej strony internetowej — `<a href="http://men.gov.pl">MEN<a/>`;
- do zasobu dostępnego przez FTP — `<a href="ftp://ftp.tpnet.pl">TPNET</a>`.

Odsyłacz graficzny

Na stronie internetowej oprócz odsyłaczy tekstowych można umieszczać odsyłacze graficzne. W tym przypadku kliknięcie elementu graficznego przeniesie użytkownika do wybranej strony internetowej.

Przykład 1.50

```
<a href-"adres.htm"><img src="grafika.gif"></a>
```

Odnośniki

Odnośniki pozwalają na przenoszenie się do określonego miejsca w obrębie tej samej strony. Zastosowanie tego mechanizmu ułatwia nawigowanie, zwłaszcza w długim dokumencie niemieszczącym się w całości na ekranie.

Odnośnik jest definiowany z użyciem znacznika `<a>`:

```
<a name="odnośnik">Tekst odnośnika</a>
```

gdzie *odnośnik* określa wybieraną część strony.

Teraz należy zdefiniować odsyłacz do istniejącego odnośnika:

```
<a href="#odnośnik">Tekst odnośnika</a>
```

Mapy odsyłaczy

Dowolna ilustracja umieszczona na stronie internetowej może być odsyłaczem do innej strony WWW. Może też stanowić odsyłacz do wielu różnych stron internetowych. W tym celu ilustrację można podzielić na kilka obszarów o określonym kształcie i do każdego z nich przypisać adres internetowy. Powstanie tzw. mapa odsyłaczy. Kliknięcie takiego obszaru spowoduje przeniesienie do strony, której adres został z nim powiązany.

Definiowanie każdego obszaru osobno jest czasochłonne, ale istnieje możliwość zbudowania mapy odsyłaczy za pomocą znacznika `<map>`. Znacznik `<map>` ma następujące atrybuty:

- `href=" "` — definiuje połączenie;
- `coords=" "` — definiuje współrzędne końcowych punktów aktywnego obszaru;
- `shape=" "` — określa kształt aktywnego obszaru: `circle` — koło, `rect` — kwadrat, `polygon` — wielokąt;
- `titles=" "` — odpowiednik atrybutu `alt` dla wstawianej grafiki (wyświetla tekst zastępczy).

Jeżeli dla wstawianego obrazu będzie deklarowana mapa odsyłaczy, to do znacznika `<img>` należy dodać atrybut `usemap="#nazwamapy"`.

Przykład 1.51

```
<img src="mapa.jpg" width="200" height="100" border="0" align="middle"
alt="mapa odsylaczy" usemap="#nazwa mapy">
<map name="nazwa mapy">
   <area shape="rect" coords="0,0,99,49" href="strona1.html"
alt="strona1" target="_blank">
   <area shape="circle" coords="157,100,40" href="strona2.html"
alt="strona 2" target="_blank">
</map>
```

Użyty w przykładzie atrybut `target="miejsce"` określa, w jakim oknie przeglądarki zostanie otwarta strona. Po wpisaniu wartości `target="_blank"` strona zostanie otwarta w nowym oknie.

1.6.9. Formularze

Dzięki formularzom, które mogą być umieszczane na stronie internetowej, można zbierać informacje od użytkowników witryny. Internauci wypełniają formularz bezpośrednio na stronie — zebrane w ten sposób informacje mogą zostać przesłane np. pocztą elektroniczną (formularz pocztowy) i, po zainstalowaniu specjalnego oprogramowania lub skryptów wykonywanych bezpośrednio w przeglądarce, przetworzone na potrzeby właściciela witryny. Formularze mogą służyć do tworzenia ankiet, książek gości, zamawiania towarów czy przesyłania dodatkowych informacji.

Formularz może zawierać: pola tekstowe, przyciski opcji, pola wyboru, listy rozwijane. W języku HTML istnieje grupa znaczników służących do projektowania formularzy.

Każdy formularz zaczyna się od znacznika otwierającego `<form>` i kończy się znacznikiem zamykającym `</form>`. Pomiędzy tymi znacznikami znajdują się wszystkie inne polecenia dotyczące formularza. Podstawowe elementy formularza to pola tekstowe i listy oraz przycisk powodujący wysłanie danych.

Przykład 1.52

```
<form action="mailto:nauka@gmail.com" method="post">
.........treść formularza
</form>
```

lub

```
<form action="mailto:nauka@gmail.com?subject=temat" method="post">
.........treść formularza
</form>
```

Znacznik `<form>` może mieć następujące atrybuty:

- `action="adres e-mail"`. To najważniejszy atrybut, ponieważ ustala adres poczty elektronicznej, na który będzie wysyłany formularz. Jako wartość tego atrybutu można również podać adres skryptu PHP lub CGI, który będzie wywoływany po zatwierdzeniu formularza. Dane wprowadzone do formularza zostaną przekazane

do wybranego skryptu. Dodanie parametru `?subject=temat` umożliwia podanie tytułu formularza.

- `method="post"`. Określa on sposób (metodę) przekazania informacji do wskazanego skryptu lub pod wskazany adres. Może przyjąć jedną z dwóch wartości: `post` lub `get`. Wartością domyślną jest `get`, ale zaleca się stosowanie metody `post`.
- `enctype=" "`. Ustala on sposób kodowania przesyłanych danych. Wybierany bywa *tylko wtedy, gdy formularz* przekazywany jest metodą `post`. Dla metody `get` zawsze ma ustawioną wartość `"application/x-www-form-urlencoded"`. Domyślnie dla metody `post` też przyjmuje tę wartość. Przy przesyłaniu pliku na serwer należy ustawić dla tego atrybutu wartość `"multipart/form-data"`. Wysyłając zwykły formularz, atrybut ten można więc pominąć, natomiast przy wysyłaniu pliku należy nadać atrybutowi wartość `"multipart/form-data"`.

Pole INPUT

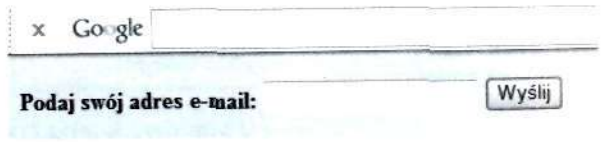
Pole INPUT służy do wprowadzania danych tekstowych do formularza i jest definiowane przy użyciu znacznika `<input>`.

Przykład 1.53

```
<form action="mailto:nauka@gmail.com" method="post">
Podaj swój adres e-mail:
<input name="adres" value="">
<input type="submit" value="Wyślij">
</form>
```

Wynik interpretacji kodu został pokazany na rysunku 1.25.

Rysunek 1.25. Pole INPUT do wprowadzania danych w formularzu



Znacznik `<input>` ma następujące atrybuty:

- `name` — określa nazwę danego pola oraz nazwę zmiennej, do której zostanie podstawiona wprowadzona wartość. Należy zwrócić uwagę, aby wartość tego atrybutu była różna dla wszystkich pól formularza.
- `value` — określa domyślną wartość pola.
- `size` — określa liczbę znaków mieszczących się w widocznej części pola.
- `maxlength` — określa maksymalną liczbę znaków, jaką można wpisać do pola.
- `type` — określa typ pola.

Typy pól

Typ pola może przyjmować następujące wartości:

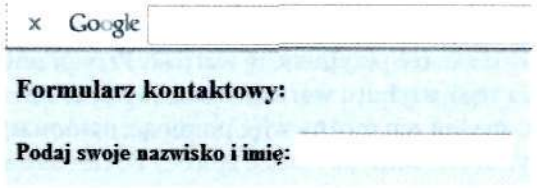
- `type="text"` — pole tekstowe.

Przykład 1.54

```
<form action="mailto:nauka@gmail.com" method="post">
<b>Podaj swoje nazwisko i imię:</b>
<input type="text" name="nazwisko" value="" size="30">
</form>
```

Wynik interpretacji kodu został pokazany na rysunku 1.26.

Rysunek 1.26. Formularz zawierający pole tekstowe



Jeżeli nie zostanie zdefiniowany typ pola, domyślnie będzie ono polem tekstowym. Jeżeli nie podamy wartości atrybutu `size`, to będzie ona wynosiła `20`.

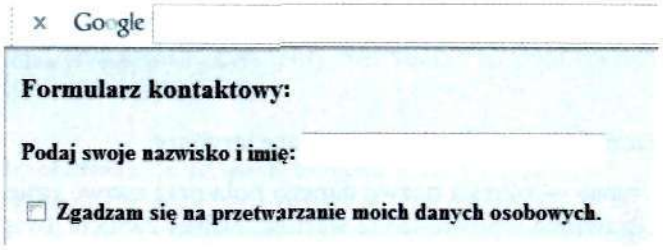
- `type="checkbox"` — pole zaznaczenia opcji (można zaznaczyć opcję lub usunąć zaznaczenie).

Przykład 1.55

```
<form action="mailto:nauka@gmail.com" method="post">
<input type="checkbox" name="opcje" maxlength="1">
Zgadzam się na przetwarzanie moich danych osobowych.
</form>
```

Wynik interpretacji kodu został pokazany na rysunku 1.27.

Rysunek 1.27. Formularz zawierający pole opcji



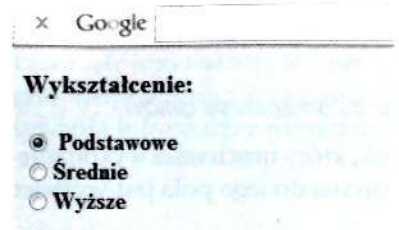
- `type="radio"` — pole wyboru (z grupy opcji można zaznaczyć tylko jedną). Kliknięcie wartości powoduje zaznaczenie danej opcji, ale równocześnie usuwa zaznaczenie innej.

Przykład 1.56

```
<form action="mailto:nauka@gmail.com" method="post">
<b><font size="4">Wykształcenie:</font></b><br><br>
<input type="radio" value="Podstawowe" name="grupa_opcji" checked>
Podstawowe<br>
<input type="radio" value="Średnie" name="grupa_opcji">Średnie<br>
<input type="radio" value="Wyższe" name="grupa_opcji">Wyższe
</form>
```


Wynik interpretacji kodu został pokazany na rysunku 1.28.

Rysunek 1.28. Formularz zawierający grupę opcji



Atrybut `checked` podany przy jednej z wartości spowoduje, że pole to będzie domyślnie wybrane. Dla pól typu `radio` wszystkie utworzone pola muszą mieć taką samą nazwę (atrybut `name=" "`). Atrybut `value=" "` jest w tym przypadku konieczny i musi być inny dla każdego pola.

- `type="password"` — pole wprowadzania hasła (wprowadzane znaki nie są widoczne).

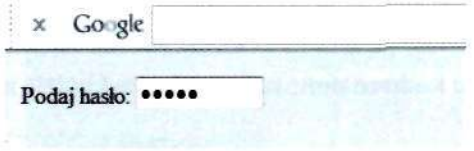
Po wpisaniu hasła pojawią się kropki zamiast wprowadzonej wartości. Należy jednak pamiętać, że mimo to dane wysyłane na serwer nie są szyfrowane.

Przykład 1.57

```
<form action="mailto:nauka@gmail.com" method="post">
Podaj hasło:
<input type="password" name="haslo" size="10" maxlength="14">
</form>
```

Wynik interpretacji kodu został pokazany na rysunku 1.29.

Rysunek 1.29. Formularz zawierający pole wprowadzania hasła

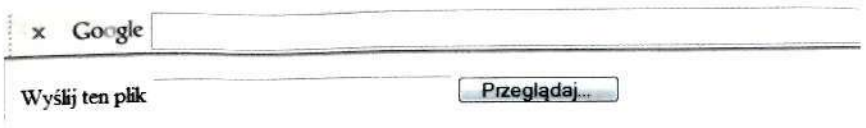


- `type="file"` — pozwala dołączyć do formularza plik, który po zatwierdzeniu formularza zostanie przesłany na serwer. Przy tworzeniu pola typu `file` musi powstać pole, gdzie użytkownik może wpisać ścieżkę dostępu i nazwę pliku, oraz przycisk *Przeglądaj*, za pomocą którego użytkownik może wyszukać plik. Formularz zawierający pole tego typu musi być przekazywany metodą `post` i należy ustawić sposób kodowania danych na `enctype="multipart/form-data"`.

Przykład 1.58

```
<form enctype="multipart/form-data" action="mailto:nauka@gmail.com"
method="post">
Wyślij ten plik
<input type="file" name="plik" size="30">
</form>
```

Wynik interpretacji kodu został pokazany na rysunku 1.30.



Rysunek 1.30. Formularz do dołączania plików

`type="submit"` — przycisk, który uruchamia wykonanie akcji zdefiniowanej w atrybucie `action`. Wartość przypisana do tego pola jest wyświetlana jako napis na przycisku.

Przykład 1.59

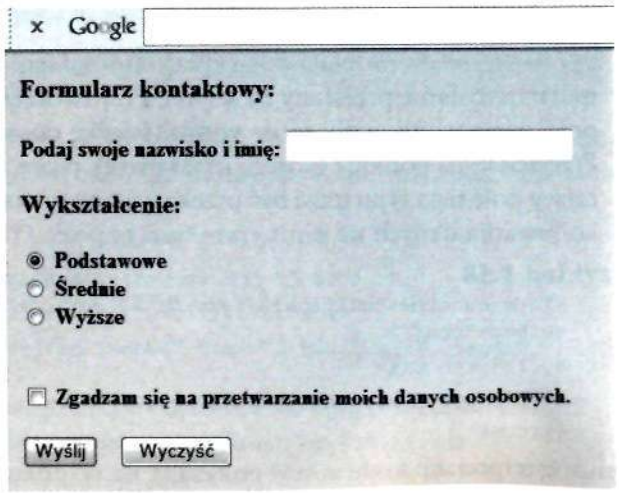
```
<body>
<form action="mailto:nauka@gmail.com" method="post">
<b><font size="4">Formularz kontaktowy:</font></b><br><br>
<b>Podaj swoje nazwisko i imię:</b>
<input name="adres" value=" " size="30"><br><br>

<b><font size="4">Wykształcenie:</font></b><br><br>
<input type="radio" value="Podstawowe" name="grupa_opcji" checked>
Podstawowe<br>
<input type="radio" value="Średnie" name="grupa_opcji"> Średnie<br>
<input type="radio" value="Wyższe" name="grupa_opcji"> Wyższe<br><br><br>

<input type="checkbox" name="opcje"  maxlength="1">
 Zgadzam się na przetwarzanie moich danych osobowych.<br><br>
 <input type="submit" value="Wyślij">   
 <input type="reset" value="Wyczyść">
</form>
</body>
```

Wynik działania kodu zdefiniowanego w przykładzie został pokazany na rysunku 1.31.

Rysunek 1.31. Formularz zawierający przyciski „submit" i „reset"



- `type="reset"` — przycisk, którego kliknięcie usuwa zawartość wszystkich pól formularza.
- `type="hidden"` — pole tekstowe niewidoczne na stronie. Pole jest niewidoczne i niedostępne dla użytkownika, ale jego wartość jest przesyłana wraz z formularzem. Nie ma znaczenia, w którym miejscu formularza zostaną umieszczone pola ukryte. Mogą one służyć do zapisywania informacji z formularzy wcześniej wypełnionych przez użytkownika.

Przykład 1.60

```
<input type="hidden" name="Data urodzenia" value="3 marca 1980">
```

- `type="button"` — przycisk dowolnego przeznaczenia.

```
<input type="button" value="Przycisk">
```

Utworzony w ten sposób przycisk po kliknięciu nie spowoduje żadnego działania. W celu jego wykorzystania należy zdefiniować program realizujący obsługę zdarzeń.

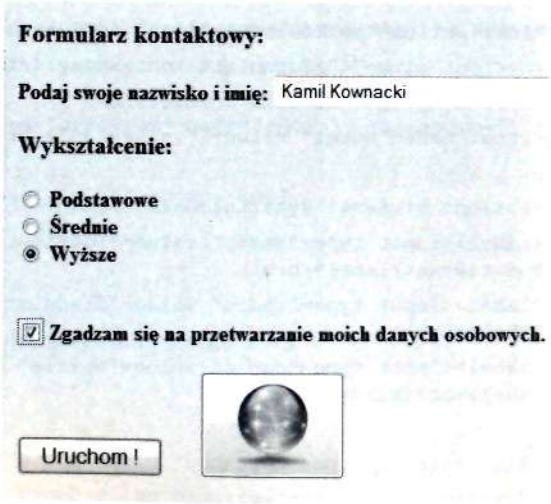
Pole BUTTON

Pole `BUTTON` służy do tworzenia w formularzu przycisków i jest definiowane za pomocą znacznika `<button>`. Ma działanie podobne do znacznika `<input>`, który dla atrybutu `type` przyjmuje jedną z wartości: `button`, `reset` lub `submit`.

```
<button type="submit" name="przycisk1">Wyślij</button>
<button type="reset" name="przycisk2">Usuń</button>
<button type="button" name="przycisk3">Powrót</button>
```

Różnica polega na tym, że przyciski tworzone za pomocą znacznika `<button>` mogą zawierać elementy ustawiające ich wygląd. Znacznik `<button>` może mieć atrybut `style` powodujący zmianę wyglądu tekstu na przycisku (rysunek 1.32).

Rysunek 1.32. Formularz z przyciskami zdefiniowanymi za pomocą znacznika `<button>`



Przykład 1.61

```
<button type="submit" name="przycisk" style="font:12pt arial;
background:red;">Przycisk</button>
```

Aby umieścić na przycisku obraz, należy wykorzystać znacznik <img>.

Przykład 1.62

```
<button type="submit" name="obrazek">
<img src="obraz.png" alt="formularz">
</button>
```

Etykieta

Opis pola, który wyświetla się na formularzu, nie jest związany z tym polem. Używając znacznika <label>, można związać tekst etykiety z polem, którego dotyczy opis.

Definicję etykiety można zapisać następująco:

```
<label> <input type="checkbox" name="nazwa" value="wartość">Etykieta
pola</label>
```

lub

```
<input type="checkbox" name="nazwa" value="wartość" id="pole">
<label for="pole">Etykieta pola</label>
```

Po zdefiniowaniu takiego polecenia wyświetli się pole oraz skojarzona z nim etykieta. Efekt jest taki, że kliknięcie zarówno w obrębie pola, jak i etykiety powoduje zaznaczenie opcji lub usunięcie zaznaczenia. Tekst etykiety może zostać zdefiniowany przed definicją pola INPUT lub po tej definicji. Etykiety można przypisywać do wszystkich typów pól formularza.

Przykład 1.63

```
<body >
<form action="mailto:nauka@gmail.com" method="post">
<b><font size="4">Formularz kontaktowy:</font></b><br><br>
<b>Podaj swoje nazwisko i imię:</b>
<input name="adres" value="" size="30"><br><br>

<b><font size="4">Wykształcenie:</font></b><br><br>
<label><input type="radio" value="Podstawowe" name="grupa_opcji" checked>
Podstawowe</label><br>
<label><input type="radio" value="Średnie" name="grupa_opcji"> Średnie</
label><br>
<label><input type="radio" value="Wyższe" name="grupa_opcji"> Wyższe</
label><br><br><br>

<label><input type="checkbox" name="opcje"  maxlength="1">
 Zgadzam się na przetwarzanie moich danych osobowych.</label><br><br>
 <input type="submit" value="Wyślij">   
 <input type="reset" value="Wyczyść">
</form>
</body>
```


Grupowanie elementów formularza

Jeżeli projektowany formularz zawiera wiele elementów różnego rodzaju, można te elementy pogrupować za pomocą znacznika <fieldset>. Powinien on znaleźć się za znacznikiem <form>, ale przed znacznikami definiującymi poszczególne elementy.

Sposób grupowania elementów:

```
<fieldset>
<legend align="kierunek">
</legend>
Definiowanie elementów grupy
</fieldset>
```

Przykład 1.64

```
<body bgcolor="#dfdfdf">
<form action="mailto:nauka@gmail.com" method="post">

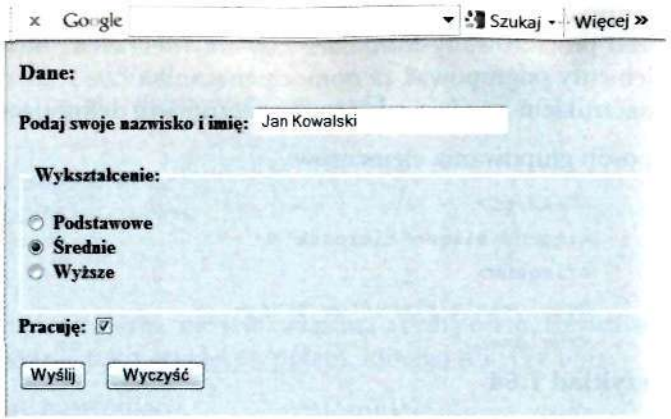
<b><font size="4">Dane:</font></b><br><br>
<b>Podaj swoje nazwisko i imię:</b>
<input name="adres" value="" size="30"><br><br>

<fieldset>
<legend align="left">Wykształcenie:</legend><br>

<label><input type="radio" value="Podstawowe" name="grupa_opcji" checked>
Podstawowe</label><br>
<label><input type="radio" value="Średnie" name="grupa_opcji"> Średnie</
label><br>
<label><input type="radio" value="Wyższe" name="grupa_opcji"> Wyższe</
label><br><br>
</fieldset>
<label> Pracuję: <input type="checkbox" name="opcje"  maxlength="1">
 </label><br><br>
 <input type="submit" value="Wyślij">   
 <input type="reset" value="Wyczyść">
</form>
</body>
```

Wynik interpretacji kodu został pokazany na rysunku 1.33.

Rysunek 1.33.
Grupowanie elementów formularza



Do podpisania grupy elementów został zastosowany znacznik `<legend>`, w którym zdefiniowano nagłówek grupy (*Wykształcenie:*). Atrybut `align="left"` powoduje wyrównanie nagłówka do lewej. Atrybut `align` może przyjmować wartości: `left`, `right`, `top` i `bottom`. Znacznik `<legend>` może zostać pominięty.

Pole SELECT

Pole `SELECT` służy do wyświetlania w formularzu listy wartości i zachęca do wybrania jednej lub kilku z nich. Jest definiowane przy użyciu znacznika `<select>`, wewnątrz którego mogą wystąpić znaczniki `<option>`. Znacznik `<select>` definiuje listę, a poszczególne wartości tej listy są definiowane za pomocą znacznika `<option>`.

Definicja listy jest następująca:

```
<select name="lista">
<option> Pierwszy element listy</option>
<option> Drugi element listy</option>
<option> Trzeci element listy</option>
</select>
```

Dla znacznika `<select>` można używać następujących atrybutów:

- `size="n"` — *n* określa liczbę pozycji widocznych w polu `SELECT`. Jeżeli atrybut nie zostanie zdefiniowany, użytkownik nie będzie mógł wybrać żadnej wartości z listy.
- `multiple` — można na liście zaznaczyć więcej niż jedną pozycję.

Dla znacznika `<option>` można używać atrybutów:

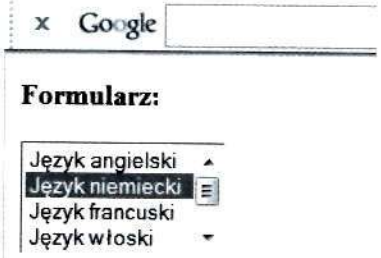
- `selected` — pozycja, dla której został zdefiniowany ten atrybut, będzie automatycznie zaznaczona.
- `value="wartość"` — określa wartość przypisaną pozycji listy.

Przykład 1.65

```
<body >
<form action="mailto:nauka@gmail.com" method="post">
<b><font size="4">Formularz:</font></b><br><br>
<select name="Języki obce" size="3" multiple >
<option> Język angielski</option>
<option selected="selected"> Język niemiecki</option>
<option> Język francuski</option>
<option> Język włoski</option>
<option> Język rosyjski</option>
<option> Język hiszpański</option>
</select>
</form>
</body>
```


Wynik interpretacji kodu powyższej listy został pokazany na rysunku 1.34.

Rysunek 1.34.
Formularz z polem listy wartości



Listy zagnieżdżone

Stosując znaczniki `<select>` i `<option>`, definiujemy listę wartości. Zdefiniowanej liście można nadać strukturę wielopoziomową, tworząc podmenu dla każdej jej wartości. Do tego celu służy znacznik `<optgroup>` z atrybutem `label="tytuł grupy"`, gdzie *tytuł grupy* to nazwa podmenu.

```
<optgroup label="tytuł">
    .........
</optgroup>
```

Dla każdego podmenu należy utworzyć przy użyciu znacznika `<option>` oddzielną listę wartości i przed nim wstawić znacznik `<optgroup>`.

Przykład 1.66

```
<body >
<form action="mailto:nauka@gmail.com" method="post">
<b><font size="4">Formularz:</font></b><br><br>
<select name="Języki obce" size="10" multiple >

<optgroup label="Języki słowiańskie">
<option selected="selected">polski</option>
<option> rosyjski</option>
<option> ukraiński</option>
</optgroup>
```

```
<optgroup label="Języki germańskie">
<option> angielski</option>
<option> niemiecki</option>
</optgroup>

<optgroup label="Języki romańskie">
<option> francuski</option>
<option> włoski</option>
<option> hiszpański</option>
</optgroup>

<optgroup label="Języki bałtyckie">
<option> litewski</option>
<option> łotewski</option>
</optgroup>

</select>
</form>
</body>
```

Wynik interpretacji kodu został pokazany na rysunku 1.35.

Rysunek 1.35.
Formularz z polem listy zagnieżdżonej



Pole TEXTAREA

Do wprowadzenia w formularzu większej ilości tekstu można użyć pola TEXTAREA, które definiujemy za pomocą znacznika <textarea>. Znacznik <textarea> może zawierać następujące atrybuty:

- name="*nazwa*" — nazwa przypisana do pola;
- rows="*n*" — liczba wierszy w polu tekstowym;
- cols="*n*" — liczba kolumn (znaków) w polu tekstowym;

- wrap — w trakcie wprowadzania tekstu linie są dostosowywane do marginesów. Atrybut wrap może przyjmować wartości:
  - off — wyłączenie zawijania tekstu;
  - physical — zawijanie tekstu na stronie i przesyłanie tekstu zawiniętego;
  - virtual — zawijanie tekstu na stronie, ale przesyłanie tekstu bez zawijania.

Na przykład:

```
<textarea name="nazwa" rows="n" cols="m" wrap="wartość"></textarea>
```

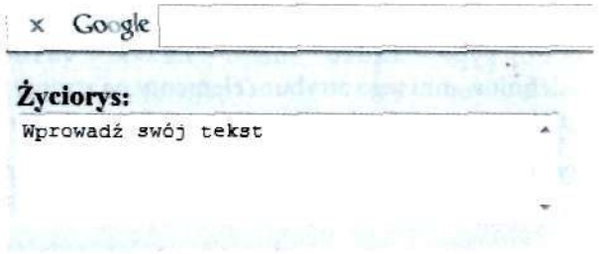
Najczęściej między znacznikami <textarea> i </textarea> nie umieszcza się żadnego tekstu.

Przykład 1.67

```
<body >
<form action="mailto:nauka@gmail.com" method="post">
<b><font size="4">Życiorys:</font></b><br>
<textarea name="blok" rows="4" cols="40" wrap="virtual">Wprowadź swój
tekst</textarea>
</form>
</body>
```

Wynik interpretacji kodu został pokazany na rysunku 1.36.

Rysunek 1.36.
Formularz z polem do wprowadzania bloku tekstu



Nieaktywne pola formularza

Cały formularz można uczynić nieaktywnym poprzez zdefiniowanie atrybutu disabled. Można również dodać atrybut disabled do dowolnego elementu formularza, powodując, że tylko ten element stanie się nieaktywny. Najlepsze efekty daje związanie atrybutu disabled ze znacznikiem <input>. Przykład: <input disabled name="nazwisko" value="Podaj nazwisko" size="25">.

Inną metodą może być uniemożliwienie wprowadzenia w polu zmian przez użytkownika. Do tego celu służy atrybut readonly, którego użycie spowoduje zablokowanie pola. Przykład: <input readonly name="nazwisko" value="Podaj nazwisko" size="25">. Atrybut readonly może być używany dla pól tekstowych, pól haseł, pól wyboru i bloków tekstu.

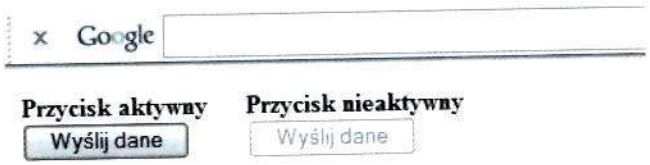
Przykład 1.68

```
<body >
<form action="mailto:nauka@gmail.com" method="post">
```

```
<b>Przycisk aktywny           Przycisk
nieaktywny</b><br>
<input type="submit" value="Wyślij dane" name="przycisk1">    
 
<input type="submit" value="Wyślij dane" name="przycisk2"
disabled="disabled">
</form>
</body>
```

Wynik interpretacji kodu został pokazany na rysunku 1.37.

Rysunek 1.37. Zdefiniowane przyciski: aktywny i nieaktywny



Kolejność pól dla klawisza Tab

Po elementach formularza można poruszać się, korzystając z klawisza *Tab* znajdującego się na klawiaturze. Domyślnie przemieszczanie się między elementami formularza odbywa się zgodnie z kolejnością ich definiowania w kodzie programu. Można zmienić tę kolejność, umieszczając w elementach formularza atrybut `tabindex="n"`, gdzie *n* określa kolejność obsługiwania elementu. Przykład: `<input type="radio" name="nazwa" value="wartość" tabindex="0">`. Po zdefiniowaniu tego atrybutu elementy na stronie będą obsługiwane w kolejności od najniższego numeru określonego w atrybucie `tabindex`.

Przykład 1.69

```
<input name="adres" value="" size="30" tabindex="0"><br><br>

<fieldset>
<legend align="left">Wykształcenie:</legend><br>
<input type="radio" value="Podstawowe" name="grupa_opcji"> Podstawowe<br>
<input type="radio" value="Średnie" name="grupa_opcji"> Średnie<br>
<input type="radio" value="Wyższe" name="grupa_opcji"> Wyższe<br>
</fieldset>
<label> Pracuję: <input type="checkbox" name="opcje" tabindex="1">
 </label><br><br>
```

Klawisze skrótów

Do definiowanych w formularzu elementów można przypisywać klawisze skrótów. Zastosowanie kombinacji klawiszy spowoduje przejście do wybranego elementu. Do definiowania klawiszy skrótów służy atrybut `accesskey` umieszczony w definicji elementu, którego będzie dotyczył.

Przykład 1.70

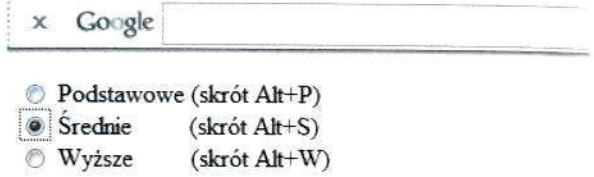
```
input type="radio" value="Podstawowe" name="grupa_opcji" accesskey="P">
Podstawowe skrót Alt+P<br>
```



```
<input type="radio" value="Średnie" name="grupa_opcji" accesskey="S">
Średnie skrót Alt+S <br>
<input type="radio" value="Wyższe" name="grupa_opcji" accesskey="W">
```

Wynik interpretacji kodu został pokazany na rysunku 1.38.

Rysunek 1.38. Formularz z przypisanymi do elementów klawiszami skrótów



UWAGA

W niektórych przeglądarkach klawisze skrótów nie działają prawidłowo.

Usprawnienia w formularzach

Język HTML 5 wprowadza nowe znaczniki i atrybuty usprawniające obsługę formularzy.

Aby automatycznie ustawić kursor w wybranym polu formularza, należy dodać atrybut `autofocus`.

Przykład 1.71

```
<input name="adres" type="text" autofocus>
```

Do wyłączania autouzupełniania w wybranych polach lub w całym formularzu służy atrybut `autocomplete` przyjmujący wartość *on* lub *off*.

Przykład 1.72

```
<input type="text" autocomplete="off">
```

Dla wybranego elementu formularza można zdefiniować listę proponowanych wartości wyświetlanych jako lista rozwijana. Użytkownik może wprowadzić własną wartość spoza listy.

Przykład 1.73

```
<input type="text" name="blok" list="jezyki">
<datalist id="jezyki">
<option> Angielski</option>
<option> Francuski</option>
<option> Włoski</option>
</datalist>
```

W języku HTML 5 znajdują się też elementy walidacji formularzy.

Atrybut `required` służy do sprawdzenia, czy pole zostało wypełnione. Jeżeli takie pole formularza będzie puste, nie zostanie on zatwierdzony.

Przykład 1.74

```
<input type="text" required>
```

Inny element walidacji formularza to sprawdzanie, czy wartość wprowadzona do pola jest zgodna z zadeklarowanym wyrażeniem regularnym.

Przykład 1.75

```
<input type="text" pattern="[0-9]{2}-[0-9]{2}">
```

Oprócz typów pól dostępnych w HTML 4 język HTML 5 udostępnia grupę pól do wprowadzania danych określonego typu. Są to między innymi:

- `<input type="email" >` — do wprowadzania adresu e-mail,
- `<input type="url" >` — do wprowadzania adresu URL,
- `<input type="number" min="1" max="7" step="1" value="2">` — do wprowadzania liczby całkowitej z podanego przedziału,
- `<input type="date" value="2013-06-01">` — do wprowadzania daty,
- `<input type="color" value="#F0F0F0">` — do wprowadzania koloru.

Podsumowanie

Z formularzami na stronach internetowych spotykamy się bardzo często. Przy budowaniu formularzy konieczna jest znajomość tworzenia i działania omówionych elementów. Należy pamiętać, że aby w pełni wykorzystać formularze, nie wystarczy znać tylko język HTML. Niezbędna jest jeszcze znajomość języka przetwarzania po stronie serwera (ang. *server-side*) i języka po stronie klienta (ang. *client-side*). Formularz projektujemy w języku HTML, kod przetwarzający dane, które zostały wprowadzone do formularza, piszemy w języku przetwarzania po stronie serwera (np. PHP, Perl), a obsługę zdarzeń związanych z formularzem tworzymy w języku po stronie klienta (np. JavaScript, VBScript).

Zadanie 1.10

Wykorzystując możliwości definiowania różnych elementów formularza, zaprojektuj ankietę na temat zainteresowań osób odwiedzających Twoją stronę. Ankieta powinna zawierać między innymi podstawowe dane osobowe. Zastanów się, jakie informacje chciałbyś uzyskać od odwiedzających witrynę.

1.6.10. Multimedia

W większości przeglądarek można bez problemu wyświetlać tekst i grafikę zapisaną w formatach GIF, JPG lub PNG, natomiast pliki z innymi rozszerzeniami, np. zawierające dźwięk, animację lub wideo, wymagają dodatkowych narzędzi. Są to programy pomocnicze lub wtyczki (ang. *plug-in*).

Wtyczka to specjalny program, który doinstalowany do przeglądarki zwiększa jej możliwości. Najczęściej spotykamy wtyczki do plików dźwiękowych i do filmów.

Pliki multimedialne mogą mieć różne rozszerzenia. W tabeli 1.3 podano najczęściej spotykane.

Tabela 1.3. Formaty plików multimedialnych

Rozszerzenie	Typ pliku
.wav	format plików dźwiękowych typu *Wave Form Audio Format*
.mid, .midi	format plików dźwiękowych MIDI
.avi	format plików wideo typu AVI
.mp3	format kompresji plików audio MP3
.mpg, .mpeg	format plików wideo typu MPEG
.mov	format związany z technologią QuickTime
.wmv	format plików wideo typu WMV

Znacznik embed

Do umieszczenia na stronie pliku multimedialnego można wykorzystać znacznik `<embed>`. Nie występuje on w specyfikacji języka HTML 4.01, ale jest obsługiwany przez niektóre przeglądarki (Internet Explorer — IE, Firefox, Opera). W przeciwieństwie do innych znaczników atrybuty stosowane w połączeniu ze znacznikiem `<embed>` zależą od typu wtyczki używanej do odtwarzania pliku.

Jedynym wymaganym atrybutem jest `src`. Jako wartość tego atrybutu powinna zostać podana ścieżka dostępu do pliku, który będzie odtwarzany, np. `<embed src="plik">`.

Zwykle podawane są również atrybuty `width` i `height`, określające obszar wyświetlania wtyczki. W przeciwieństwie do obrazów (format GIF, JPG) obiekty multimedialne nie zawsze mają zdefiniowane wymiary. Jeżeli pominie się te atrybuty, przeglądarki w różny sposób zinterpretują ich brak. Czasami pojawiają się komunikaty o błędach, czasami przeglądarka nadaje obrazom preferowane rozmiary, a czasami rozmiary zostają nadane w sposób przypadkowy.

Podczas stosowania różnych wtyczek należy sprawdzić w dokumentacji, jakie rozmiary są dla nich odpowiednie. Inną metodą ustalania obszaru wyświetlania wtyczki jest eksperymentowanie z różnymi rozmiarami dla różnych przeglądarek.

Przykład 1.76

```
<embed src="muzyka.mp3" width="25" height="30">
```

Kolejnym atrybutem jest `align`, który wyrównuje osadzony obiekt w stosunku do otaczającego go tekstu.

Przykład 1.77

```
<embed src="muzyka.mp3" width="25" height="30" align="right">
```

Atrybut `name` definiuje nazwę osadzonego obiektu. Nie jest on konieczny, ale czasami przydaje się przy korzystaniu ze skryptów (np. JavaScript).

Atrybut `autostart="typ"` określa sposób uruchamiania odtwarzania dźwięku lub filmu. Atrybut może przyjmować wartości:

- `true` — plik zostanie automatycznie uruchomiony po wczytaniu strony;
- `false` — plik będzie uruchamiany ręcznie przez panel sterujący.

Przykład 1.78

```
<embed src="muzyka.mp3" width="140" height="50" autostart="false">
```

Atrybut `loop="typ"` określa, ile razy dźwięk lub film zostanie odtworzony, a także czy dźwięk lub film powinien być odtwarzany w pętli, czy nie. Atrybut może przyjmować wartości:

- `true` — odtwarzanie w pętli,
- `false` — brak powtarzania,
- *n* — odtwarzanie *n* razy.

Przykład 1.79

```
<embed src="pik.wav" loop="false" width="140" height="50">
```

lub

```
<embed src="hej.wav" loop="3" width="140" height="50">
```

lub

```
<embed src="sam.avi" loop="true" width="140" height="150">
```

Niektóre przeglądarki nie rozpoznają prawidłowo atrybutu `loop="n"` (np. Internet Explorer interpretuje ten zapis jako odtwarzanie dźwięku w pętli), ale interpretują prawidłowo atrybut `playcount`, który również określa, ile razy dźwięk lub film będzie odtworzony.

Przykład 1.80

```
<embed src="pik.wav" playcount="5" width="140" height="50">
```

Ten zapis zostanie prawidłowo zinterpretowany przez IE.

Można połączyć atrybuty `loop` i `playcount`, aby w różnych przeglądarkach uzyskać ten sam efekt.

Przykład 1.81

```
<embed src="pik.wav" loop="5" playcount="5" width="140" height="50">
```

W IE `playcount` ma pierwszeństwo przed `loop`.

Atrybut `volume="v"` dla plików dźwiękowych określa poziom głośności, przy czym *v* oznacza głośność od `0` (brak dźwięku) do `100`. Domyślną wartością jest `50`. Niezależnie od tych ustawień użytkownik ma możliwość samodzielnej regulacji głośności.

Przykład 1.82

```
<embed src="muz.mid" width="140" height="50" volume="20">
```

Atrybut `controls="rodzaj"` dla plików dźwiękowych określa rodzaj wyświetlanego elementu sterującego. Atrybut `controls` może przyjmować następujące wartości:

- `console` — konsola. Jest to ustawienie domyślne. Przeglądarka powinna wyświetlić pełny zestaw ikon sterujących dźwiękiem (przyciski: *Start*, *Wstrzymaj*, *Stop* oraz suwak głośności).

Przykład 1.83

```
<embed src="muz.mid" width="140" height="60" controls="console">
```

- `smallconsole` — mała konsola. Przeglądarka powinna wyświetlić zminimalizowany zestaw ikon sterujących dźwiękiem (przyciski: *Start*, *Stop* oraz suwak głośności).
- `playbutton` — przycisk *Odtwórz*. Pokazywany jest tylko przycisk odtwarzania.
- `pausebutton` — przycisk *Wstrzymaj*. Pokazywany jest tylko przycisk pauzy.
- `stopbutton` — przycisk *Stop*. Pokazywany jest tylko przycisk zatrzymania.
- `volumebutton` — suwak głośności. Pokazywany jest tylko suwak do regulowania głośności.

Jeżeli zostanie użyty atrybut `controls`, należy koniecznie zdefiniować atrybuty określające szerokość i wysokość elementów sterujących: `width="x"` oraz `height="y"`.

Jeżeli w przeglądarce nie ma oprogramowania do uruchomienia pliku multimedialnego umieszczonego na stronie, atrybut `pluginspage` wskazuje adres, pod którym znajduje się niezbędne oprogramowanie. Dobrym rozwiązaniem jest umieszczenie tego atrybutu, jeżeli używa się mniej popularnych formatów plików.

Przykład 1.84

```
<embed src="muzyka.aiff" width="25" height="30" autostart="false"
pluginspage="http://www.apple.com/quicktime/download" QuickTime>
```

Znacznik `<embed>` może sprawiać kłopoty nawet w przeglądarkach, które obsługują pliki multimedialne. Należy pamiętać o podawaniu rozmiarów umieszczanych obiektów multimedialnych, ponieważ może się okazać, że użytkownik ma problemy z ich obsługą.

Przykład 1.85

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Muzyka</title>
</head>
<body bgcolor="8BBEF2">
<p align="left"><font size="6" face="arial"><b>Dźwięk i film:</b></font></
p><br>
<embed src="muzyka.mp3" width="60" height="60" autostart="false"
pluginspage="http://www.apple.com/quicktime/download">  </embed>
<embed src="rod.wma" controls="playbutton" "volumebutton" "stopbutton"
width="170" height="50" autostart="false" > </embed>
<embed src="film1.avi" width="250" height="250" autostart="false" > </
embed><br>
</body>
</html>
```

Poniżej pokazane zostały rezultaty uruchomienia podanego w przykładzie kodu w różnych przeglądarkach (rysunki 1.39 – 1.43).



Rysunek 1.39. Przeglądarka Internet Explorer 9



Rysunek 1.40. Przeglądarka Internet Explorer 9. Próba zainstalowania brakującej wtyczki

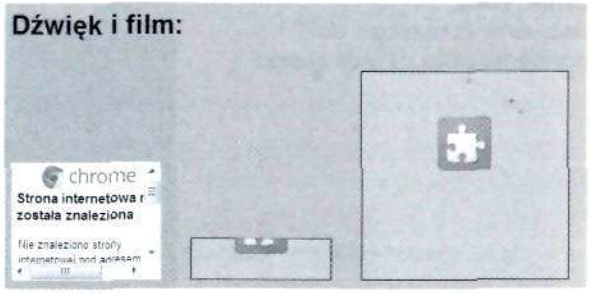


Rysunek 1.41. Przeglądarka Firefox. Brak możliwości uruchomienia plików



Rysunek 1.42. Przeglądarka Firefox. Próba zainstalowania wtyczki

Rysunek 1.43. Przeglądarka Google Chrome. Brak możliwości uruchomienia plików



Odsyłacz do pliku

Inną, dużo prostszą metodą prezentowania plików multimedialnych na stronie jest zdefiniowanie odsyłacza do tych plików. Metoda ta działa w każdej przeglądarce i sprawdza się przy każdym formacie plików.

Przykład 1.86

```
<a href-"muzyka.mp3"> Moja muzyka</a>
```

lub

```
<a href="film.avi"> <img src="obraz" width="60" height="100" border="0"
alt="wyświetl film"></a>
```

Po uruchomieniu polecenia zawierającego odsyłacz do pliku zostanie automatycznie otwarte okno z programem obsługującym ten typ plików i nastąpi odtworzenie materiału. Ten sposób odtwarzania plików nie ma wad związanych z poleceniami multimedialnymi, ale plik nie będzie otwierany na stronie, lecz w osobnej aplikacji, w związku z czym na komputerze użytkownika musi być zainstalowany odpowiedni program, który go odczyta.

Przykład 1.87

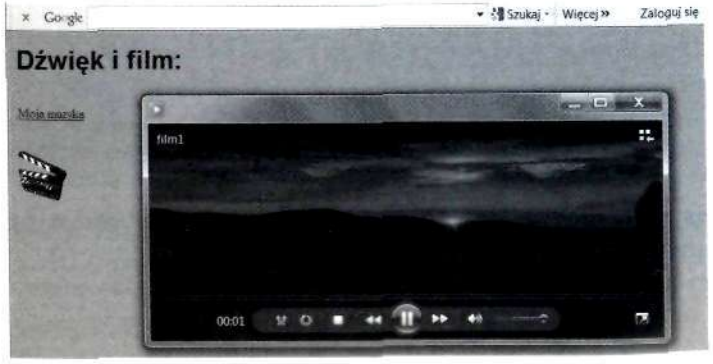
```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>
Muzyka
</title>
</head>
<body bgcolor="8BBEF2">
<p align="left"><font size="6" face="arial"><b>Dźwięk i film:</b></font></p><br>
<a href="rod.wma"> Moja muzyka</a><br><br><br>
<a href="film.avi"><img src="obraz.png" width="60" height="60" border="0" alt="wyświetl film"></a>
</body>
</html>
```

Rysunek 1.44 pokazuje efekt kliknięcia odsyłacza *Moja muzyka*. W nowym oknie został uruchomiony program Windows Media Player, w którym będzie odtworzony plik muzyczny.



Rysunek 1.44. Zastosowanie odsyłacza do uruchomienia pliku dźwiękowego

Podobnie będzie wyglądało odtworzenie pliku filmowego. Gdy klikniemy ikonę powiązaną z odsyłaczem do pliku filmowego, w nowym oknie zostanie odtworzony film (rysunek 1.45).



Rysunek 1.45. Zastosowanie odsyłacza do uruchomienia pliku filmowego

1.6.11. Multimedia w HTML 5

W HTML 5 do osadzania plików multimedialnych na stronie internetowej mogą zostać wykorzystane nowe znaczniki `<audio>` i `<video>`, które równocześnie dostarczają prostych odtwarzaczy dźwięku i wideo. W przypadku takiego odtwarzania plików multimedialnych nie są już potrzebne dodatkowe wtyczki. Bardziej skomplikowane projekty nadal wymagają użycia dodatkowych narzędzi.

Dźwięk i wideo w HTML 5

Aby zamieścić na stronie internetowej film, należy dla znacznika `<video>` w atrybucie `src` podać ścieżkę do pliku z filmem oraz ustalić wysokość i szerokość wyświetlanego odtwarzacza. Za pomocą właściwości `poster` można określić ścieżkę do obrazka, który będzie wyświetlany w trakcie ładowania filmu. Można ustawić również atrybuty:

- `autoplay` — automatyczne odtwarzanie w trakcie ładowania,
- `controls` — wyświetlenie panelu sterowania wideo przez użytkownika,
- `loop` — zapętlanie się odtwarzania wideo,
- `autobuffer` — automatyczne buforowanie wideo,
- `preload` — wczytanie wideo wraz z załadowaniem strony.

Przykład 1.88

```
<!DOCTYPE HTML>
<html>
<head>
<title>Film</title>
</head>
<body>
<video src="film.mp4" height="500" width="600" poster="obraz.png"
autoplay controls loop>
Znacznik video nie jest obsługiwany!
</video>
</body>
</html>
```

Jeżeli przeglądarka obsługuje język HTML 5, to na stronie internetowej zostanie zarezerwowany obszar o podanych wymiarach do wyświetlenia filmu. Wymiary te powinny odpowiadać rozdzielczości filmu, ponieważ zostanie on do nich dopasowany.

Do umieszczania na stronie internetowej elementów audio należy użyć znacznika `<audio>`.

Przykład 1.89

```
<!DOCTYPE HTML>
<html>
<head>
<title>Muzyka</title>
</head>
```

```
<body>
<audio controls>
<source src="muza.ogg" type="audio/ogg" />
<source src="muza.wav" type="audio/wav" />
Znacznik audio nie jest obsługiwany!
</audio>
</body>
</html>
```

Nie każdy format plików audio jest obsługiwany przez wszystkie przeglądarki, dlatego w przykładzie zostały podane dwie wersje pliku muzycznego. Znaczniki `<audio>` i `<video>` pozwalają umieścić na stronie internetowej pliki tak, aby przeglądarka internetowa mogła wybrać format, który obsługuje.

Znacznik Canvas

Znacznik `<canvas>` służy do tworzenia grafiki rastrowej na stronach WWW. Dzięki temu znacznikowi można rysować grafikę w czasie rzeczywistym. Znacznik `<canvas>` definiuje na ekranie prostokątny obszar (płótno), w którym tworzony jest rysunek.

Działanie tego znacznika oparte jest na renderingu trybu natychmiastowego (ang. *immediate mode rendering*). Rendering trybu natychmiastowego daje maksymalną kontrolę nad każdym pikselem obrazu, ale wymaga jawnego pisania kodu JavaScriptu rysującego grafikę.

Dzięki znacznikowi `<canvas>` możliwe są:

- rysowanie prostokątów i ścieżek (linie, łuki, krzywe Béziera lub krzywe drugiego stopnia),
- wypełnianie figur kolorem (z kanałem alfa), wzorem lub gradientem (liniowym lub radialnym),
- rysowanie napisów (tekstu),
- obsługa przezroczystości,
- przekształcenia (przesuwanie, skalowanie, obracanie, przekrzywianie itp.),
- osadzanie obrazów rastrowych (PNG, JPEG, GIF),
- cieniowanie.

Korzystanie ze znacznika `<canvas>` jest dość pracochłonne (np. nie ma możliwości rysowania bardziej skomplikowanych figur geometrycznych, łatwego rysowania wykresów itp.).

Przykład 1.90

```
<canvas id="moja_kanwa" width="300" height="200">
Brak obsługi canvas
</canvas>
<script>
var canvas = document.getElementById("moja_kanwa" );
var context = canvas.getContext('2d');
context.fillStyle = "red";
context.fillRect(50,50,50,50);
context.strokeStyle="black";
context.strokeRect(40,40,70,70);
</script>
```


Jeżeli przeglądarka obsługuje HTML 5, w podanym przykładzie zostanie narysowany prostokąt. Do tego celu zostały użyte funkcje `fillRect` (wypełnienie kolorem) i `strokeRect` (rysowanie krawędzi). Dla obydwu funkcji dwa pierwsze parametry określają współrzędne punktu początkowego, dwa kolejne szerokość i wysokość. Zmienne `fillStyle` i `strokeStyle` definiują kolor pędzla rysującego.

1.6.12. Zdarzenia

Korzystając ze strony internetowej, użytkownik wykonuje różne czynności. Do elementów, które zostały zaprojektowane na stronie internetowej, mogą być przypisane zdarzenia. Jeżeli użytkownik wykona jedną z czynności, która jest zdefiniowana jako zdarzenie, to element strony może zareagować na wykonaną czynność. Aby było to możliwe, należy mieć skrypt, który realizuje określone działania, i należy powiązać ten skrypt ze zdarzeniem, które może wystąpić.

Program to tak zwany skrypt. Możemy napisać go w jednym z języków skryptowych (JavaScript, VBScript) lub skorzystać z gotowych skryptów dostępnych w internecie.

W języku HTML zdefiniowano 18 zdarzeń wewnętrznych, które uruchamiają skrypty. Łączenie skryptów ze zdarzeniami występuje najczęściej w formularzach, przy wprowadzaniu i wysyłaniu danych.

Zdarzenia wewnętrzne to:

- `onclick` — występuje, gdy nastąpiło kliknięcie myszą;
- `ondblclick` — występuje, gdy nastąpiło dwukrotne kliknięcie myszą;
- `onmousedown` — występuje, gdy przycisk myszy został wciśnięty;
- `onmouseup` — występuje, gdy przycisk myszy został zwolniony;
- `onmouseover` — występuje, gdy kursor myszy został umieszczony na elemencie;
- `onmousemove` — występuje, gdy kursor myszy został przesunięty wewnątrz elementu;
- `onmouseout` — występuje, gdy kursor myszy został przesunięty poza element;
- `onkeypress` — występuje, gdy klawisz klawiatury został naciśnięty i zwolniony;
- `onkeydown` — występuje, gdy klawisz klawiatury został naciśnięty;
- `onkeyup` — występuje, gdy klawisz klawiatury został zwolniony;
- `onfocus` — występuje, gdy został wybrany element formularza;
- `onblur` — występuje, gdy kursor opuszcza element formularza;

- `onselect` — występuje, gdy w polu tekstowym został zaznaczony tekst (dotyczy tylko pól tekstowych);
- `onchange` — występuje, gdy element formularza zmienia wartość;
- `onsubmit` — występuje, gdy nastąpiło wysłanie formularza;
- `onreset` — występuje, gdy nastąpiło wyczyszczenie formularza;
- `onload` — występuje, gdy dokument został wczytany;
- `onunload` — występuje podczas opuszczania strony przez użytkownika.

Dla większości zdarzeń brak zdefiniowanej obsługi jest równoznaczny z ustawieniem domyślnym.

1.6.13. Skrypty

Skrypt to niewielki program umieszczony bezpośrednio na stronie internetowej umożliwiający wykonanie działań dynamicznych. Skrypty mogą być automatycznie wykonywane przez przeglądarkę podczas ładowania strony lub uruchamiane, gdy zaistnieje powiązane z nim zdarzenie.

Wywołanie skryptu odbywa się poprzez znacznik `<script>`.

Dla skryptów automatycznie uruchamianych ma on postać:

```
<script language="język" type="text/język">
```

Atrybut `language="język"` określa język, w jakim skrypt został napisany.

Atrybut `type="text/język"` określa używany język skryptowy.

Przykład 1.91 (dla języka JavaScript)

```
<script type="text/javascript">
treść ...
</script>
```

Przykład 1.92 (dla języka VBScript)

```
<script type="text/vbscript">
treść
</script>
```

ZADANIE

Grupa osób biorących udział w projekcie ma za pomocą języka HTML utworzyć kilka stron internetowych, które będą podstronami witryny na temat konkursów przedmiotowych odbywających się na terenie szkoły. Każda z osób biorących udział w projekcie przygotuje jedną stronę. Na stronie znajdzie się informacja o wybranym konkursie (np.: jego termin, tematyka, warunki uczestnictwa w konkursie), zdjęcia, odnośniki do informacji dostępnych w internecie. Na stronie można zastosować różne rodzaje list, cytaty, tabele oraz formularze umożliwiające rejestrację uczestników konkursów.

PYTANIA I POLECENIA KONTROLNE

1. Co należy zrobić, aby w dokumencie HTML były dostępne polskie litery?
2. Co to jest znacznik HTML? Podaj przykłady znaczników, które powinny się znaleźć na stronie.
3. Co to są atrybuty HTML? Podaj przykłady atrybutów.
4. W jaki sposób w dokumencie HTML umieszczany jest komentarz?
5. W jaki sposób w dokumencie HTML można zapisywać wartość atrybutu *color*?
6. Za pomocą jakiego znacznika w dokumencie HTML tworzone są listy punktowane?
7. Gdzie w dokumencie HTML mają zastosowanie listy definicji?
8. Jaki atrybut powinien zostać użyty dla znacznika *<table>*, aby można było otoczyć tabelę tekstem?
9. Które formaty plików graficznych są najczęściej stosowane na stronach internetowych?
10. Do czego służą umieszczane na stronach internetowych formularze?
11. Do czego służy atrybut znacznika *<form>* zapisany w postaci: *method="post"*?
12. Jeżeli w znaczniku *<input>* nie wystąpi atrybut *type*, to jaki typ domyślnie przyjmie pole INPUT?
13. Jaka jest różnica między przyciskiem tworzonym za pomocą znacznika *<button>* a przyciskiem powstałym przy użyciu znacznika *<input>*, który dla atrybutu *type* przyjmie wartość *button*?
14. Do czego służy pole *SELECT* używane w formularzu?
15. Wymień kilka elementów walidacji formularzy, które pojawiły się w języku HTML 5.
16. Jakie znaczniki w HTML 5 są wykorzystywane do osadzania plików multimedialnych na stronie internetowej?
17. Do czego w HTML 5 służy znacznik *<canvas>*? Opisz jego działanie.

2

Kaskadowe arkusze stylów (CSS)

W początkowej fazie rozwoju język HTML był rzeczywiście strukturalnym językiem znaczników. W kolejnych jego wersjach dodano elementy opisujące wygląd strony (np. zmianę koloru czy wielkości tekstu). Z czasem pomieszanie warstwy strukturalnej dokumentu z warstwą prezentacyjną zaczęło stwarzać coraz większe problemy.

Rozwiązanie tych problemów stało się możliwe po oddzieleniu warstwy strukturalnej i treści od warstwy prezentacyjnej. Za warstwę strukturalną, semantyczną i treść wciąż odpowiedzialny jest język HTML, natomiast za warstwę prezentacyjną — język CSS (ang. *Cascading Style Sheets*).

Język CSS określa układ graficzny dokumentu HTML: parametry czcionki, wysokość i szerokość obrazków, ich położenie, rodzaj tła itp. Wszystkie polecenia dotyczące formatowania powinny zostać umieszczone w oddzielnym pliku (arkuszu) i być powiązane z elementami zdefiniowanymi w kodzie HTML.

Cechy języka CSS:

- Oddziela strukturę informacyjną witryny od jej warstwy prezentacyjnej.
- Daje większe możliwości formatowania tekstu.
- Pozwala zapisać wszystkie informacje dotyczące wyglądu dokumentu w oddzielnym pliku tekstowym dołączanym później do dokumentów HTML.
- Umożliwia formatowanie wielu dokumentów przy użyciu jednego arkusza stylów.
- Umożliwia stosowanie różnych układów graficznych w zależności od typu urządzenia, na którym zamierzamy wyświetlać projektowaną stronę.

Formatowanie z wykorzystaniem CSS stało się podstawą tworzenia stron internetowych. Należy pamiętać również o tym, że w języku HTML występują przestarzałe znaczniki i atrybuty formatowania (elementy zdeprecjonowane), które będą stopniowo wycofywane z przeglądarek internetowych i zastępowane nowymi metodami formatowania dostępnymi w języku CSS.

Język CSS oferuje nowe możliwości formatowania, które były niedostępne w HTML, np. dotyczące formatowania tekstu, formatowania tła, definiowania obramowania, dodatkowe właściwości definiowania list, pozycjonowania, zmianę wyglądu odsyłaczy, stosowanie filtrów.

2.1. Wstawianie stylów

Kaskadowe arkusze stylów definiują jedynie sposób formatowania elementów dokumentu HTML, ale ich nie tworzą. Elementy te muszą zostać zdefiniowane za pomocą znaczników w dokumencie HTML.

W podstawowej składni kaskadowych arkuszy stylów jest kilka stałych elementów:

```
selektor {właściwość: wartość;}
```

- *selektor* — określa, do jakich znaczników języka HTML odnosi się dalsza część definicji stylu. Selektorem może być dowolny znacznik.
- *właściwość* — określa, jaki atrybut znacznika będzie ustawiany.
- *wartość* — określa wartość atrybutu.

Przykład 2.1

```
body {background-color: #dfdfdf; }
```

Przykład pokazuje ustawienie koloru tła strony. To samo polecenie w języku HTML miałoby postać:

```
<body bgcolor="#DFDFDF">
```

Dla jednego selektora można zdefiniować kilka atrybutów. W definicji stylu należy je oddzielić średnikami.

W dokumencie HTML istnieje kilka sposobów na dołączanie stylów CSS:

- styl lokalny (w linii),
- wewnętrzny arkusz stylów,
- zewnętrzny arkusz stylów.

Sposób wstawiania stylów zależy od konkretnej sytuacji. Podczas projektowania witryny najczęściej stosuje się różne sposoby osadzania arkuszy stylów.

2.1.1. Styl lokalny (w linii)

Korzystając ze stylu lokalnego, można zdefiniować formatowanie pojedynczego elementu strony. Taki styl jest definiowany w tej samej linii, w której znajduje się element formatowany, dlatego nazywany jest on również stylem w linii (ang. *inline*). Umieszczany jest bezpośrednio w dokumencie HTML.

Przykład 2.2

```
<html>
<head>
<title> Kaskadowe arkusze stylów</title>
```

```
</head>
<body style="background-color: #DFDFDF;">
<p style="color: red">Definiowanie stylów</p>
</body>
</html>
```

Znacznik — rozciąganie stylu

Znacznik służy do grupowania kilku elementów strony (np. słów, obrazków, akapitów) w celu nadania im określonego stylu. Ten znacznik sam w sobie nie ma narzuconych żadnych właściwości, nie wywiera żadnego wpływu na zgrupowane elementy, dlatego elementy te będzie określał wyłącznie styl. Styl dla znacznika ustala się przy użyciu stylu CSS.

```
<span style="właściwość: wartość";> .... </span>
```

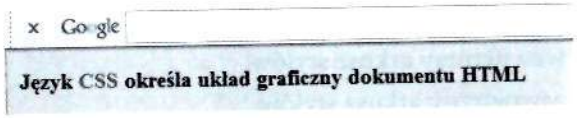
Znacznik zwykle jest wykorzystywany wtedy, gdy trzeba inaczej sformatować kilka znaków w obszarze o określonym sposobie formatowania.

Przykład 2.3

```
<html>
<head>
<title> Kaskadowe arkusze stylów</title>
</head>
<body style="background-color: #DFDFDF;">
<h3>Język <span style="color: red">CSS</span> określa układ graficzny
dokumentu HTML</h3>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.1.

Rysunek 2.1. Zastosowanie znacznika do fragmentu tekstu



Znacznik div — wydzielone bloki

Używając znacznika , elementy strony można grupować w bloki i formatować za pomocą stylów. Zdefiniowane bloki mogą zawierać m.in. akapity, listy oraz inne bloki. Znaczniki umożliwiają wydzielanie na stronie większych logicznych fragmentów w celu nadania im specyficznego formatowania z użyciem stylów.

Metoda definiowania stylu dla bloków jest podobna jak w przypadku znacznika . Ustala się go za pomocą stylu lokalnego.

```
<div style="właściwość: wartość";> .... </div>
```

Przykład 2.4

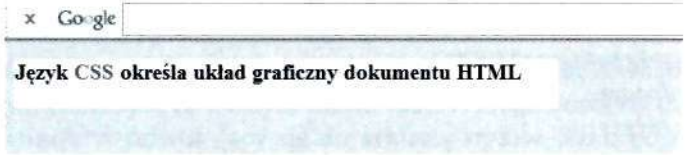
```
<html>
<head>
<title> Bloki div</title>
```



```
</head>
<body style="background-color: #DFDFDF;">
<div style="width: 500px; height: 50px; background-color: yellow;">
<h3>Język <span style="color: red">CSS</span> określa układ graficzny
dokumentu HTML</h3></div>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.2.

Rysunek 2.2. Definiowanie bloku



2.1.2. Wewnętrzny arkusz stylów

Wewnętrzny arkusz stylów jest umieszczany w dokumencie HTML dzięki zastosowaniu znacznika <style>, który zawsze występuje w części nagłówkowej dokumentu (<head>). Tę metodę wstawiania do dokumentu arkusza stylów stosuje się, gdy elementy formatowane pojawiają się na stronie wielokrotnie i wszystkie powinny mieć takie same atrybuty formatowania. Przyjmijmy, że chcemy, żeby wszystkie teksty zdefiniowane jako akapity miały kolor niebieski. Po wpisaniu odpowiedniej deklaracji stylów wszystkie akapity będą napisane czcionką niebieską bez definiowania jej koloru przy każdym elemencie. Zmiana koloru czcionki w deklaracji stylów zmieni kolor czcionki wszystkich akapitów.

Przykład 2.5

```
<html>
<head>
<title> Kaskadowe arkusze stylów</title>
<style type="text/css">
p { color: blue; }
</style>
</head>
<body>
<p>Definiowanie stylów</p>
<p>Wewnętrzny arkusz stylów</p>
</body>
</html>
```

Przykład 2.6

```
<html>
<head>
<title> Wewnętrzny arkusz stylów</title>
```

```
<style type="text/css">
body {background-color: yellow;}
h2 {font-size: 32pt;}
h3 {color: green; font-size: 24pt;}
p {color: blue; text-align: center; }
</style>
</head>
<body>
<h2> Wewnętrzne arkusze stylów</h2>
<h3>Definiowanie stylów</h3>
<p>Arkusz stylów</p>
</body>
</html>
```

Zadanie 2.1
Sprawdź działanie kodu podanego w przykładzie 2.6.

Zadanie 2.2
Używając stylu lokalnego, zdefiniuj rodzaj, kolor i rozmiar czcionki dla tekstu wyświetlanego na stronie. To samo wykonaj, używając wewnętrznego arkusza stylów. Kiedy warto zastosować styl lokalny, a kiedy wewnętrzny arkusz stylów?

2.1.3. Zewnętrzny arkusz stylów

Największą z zalet stosowania CSS jest możliwość wstawiania zewnętrznych arkuszy stylów. Polega ona na utworzeniu pliku tekstowego z rozszerzeniem *.css*, który będzie zawierał definicję wszystkich stylów używanych w projektowanej witrynie. W dokumencie HTML powinien znaleźć się odnośnik do tzw. zewnętrznego arkusza stylów, czyli do pliku z rozszerzeniem *.css*.

Odnośnik umieszczony w dokumencie HTML powinien mieć postać:

```
<link rel="stylesheet" type="text/css" href="arkusz.css">
```

Wartością atrybutu `href` powinna być ścieżka dostępu do pliku zawierającego zdefiniowane style. Aby utrzymać porządek w strukturze plików strony, warto umieszczać pliki stylów w odrębnych folderach, np. *css*.

Odnośnik do arkusza stylów powinien zostać umieszczony w części nagłówkowej dokumentu (`<head>`).

Przykład 2.7

```
<html>
<head>
<title> Kaskadowe arkusze stylów</title>
<link rel="stylesheet" type="text/css" href="arkusz.css">
</head>
```



```
<body>
<p>Definiowanie stylów</p>
<p>Wewnętrzny arkusz stylów</p>
</body>
</html>
```

Zaletą tak zdefiniowanych arkuszy stylów jest to, że do wielu dokumentów HTML można dołączyć ten sam plik arkusza stylów, czyli za pomocą jednego pliku CSS można formatować w taki sam sposób wiele stron WWW (np. kilka stron internetowych tej samej witryny).

Można również w jednym dokumencie HTML umieszczać dowolną liczbę zewnętrznych arkuszy stylów. W dokumencie tym za każdym razem musi wystąpić osobny element `<link rel="stylesheet">` odwołujący się do arkusza stylów. Jeżeli pojawi się konflikt dotyczący sposobu formatowania tego samego elementu w różnych arkuszach stylów, wiążące będą deklaracje z arkusza dołączonego najpóźniej.

Zadanie 2.3
Utwórz plik z rozszerzeniem *.css*. Zdefiniuj w nim styl tekstu wyświetlanego na stronie oraz styl tła strony. Dołącz go do dokumentu HTML opisującego przykładową stronę internetową.

2.1.4. Import arkusza stylów

Do zewnętrznego lub wewnętrznego arkusza stylów można zaimportować zewnętrzny arkusz stylów. Plik z zaimportowanym arkuszem stylów może znajdować się w dowolnym miejscu; należy tylko prawidłowo zdefiniować ścieżkę dostępu do niego. Importowanie arkusza stylów odbywa się podobnie jak dołączanie zewnętrznego arkusza. Priorytet dołączonego arkusza jest taki sam jak priorytet arkusza, do którego został on zaimportowany.

Polecenie importowania arkusza ma postać:

```
<style type="text/css">
/* <! [CDATA[ */
@import url (adres importowanego arkusza stylów.css);
/* ]]> */
</style>
```

Adres arkusza stylów powinien zostać umieszczony w nawiasach okrągłych. Przed poleceniem `import` musi zostać umieszczony znak `@`, natomiast na końcu adresu arkusza stylów średnik.

Oprócz importowania arkusza można między znacznikami `<style>` i `</style>` umieszczać inne deklaracje stylów. Polecenia importowania arkusza muszą występować na początku arkusza stylów, przed właściwymi regułami CSS.

2.1.5. Kaskadowość arkuszy stylów

Zdarza się, że w dokumencie są umieszczone odwołania do kilku zewnętrznych arkuszy stylów lub na stronie używane są zewnętrzny arkusz stylów, deklaracja stylów w nagłówku strony oraz style lokalne. Wtedy może się pojawić konflikt dotyczący sposobu formatowania tego samego elementu w różnych arkuszach stylów. Zawsze pierwszeństwo mają style, które są umieszczone bliżej formatowanego elementu. Kaskadowość arkuszy stylów polega na ścisłym określeniu priorytetów stylów i przestrzeganiu zasad formatowania zgodnie z priorytetami. Ważność stylów została ustalona w następujący sposób:

1. Styl lokalny.
2. Rozciąganie stylu.
3. Wydzielone bloki.
4. Wewnętrzny arkusz stylów.
5. Zewnętrzny arkusz stylów.
6. *Import stylów do zewnętrznego arkusza.*
7. Atrybuty definiowane w dokumencie HTML.

Styl z numerem 1 ma najwyższy priorytet, a styl z numerem 7 — najniższy. Styl z najwyższym priorytetem ma pierwszeństwo w formatowaniu elementów strony. Style o niższym priorytecie formatują element tylko wtedy, gdy style o wyższym priorytecie nie dotyczą tego elementu. Należy zauważyć, że atrybuty definiowane dla elementu w dokumencie HTML mają najniższy priorytet, a więc style CSS definiowane w dowolny sposób zawsze są ważniejsze od tych atrybutów.

Postępując zgodnie z zasadami kaskadowości stylów, w dokumencie HTML polecenie dołączenia zewnętrznego arkusza stylów trzeba umieścić przed definicją wewnętrznego arkusza stylów.

Zasady kaskadowości można zmieniać. Do tego celu służy polecenie `!important` umieszczone w deklaracji stylu po wartości, której dotyczy. Takie umiejscowienie tego polecenia spowoduje, że właściwość zdefiniowana dla elementu będzie miała pierwszeństwo, nawet jeżeli ma niższy priorytet. Polecenie `!important` dotyczy tylko właściwości, dla której zostało użyte; pozostałe właściwości elementu będą obsługiwane zgodnie z ogólnymi zasadami kaskadowości stylów.

Przykład 2.8
Użycie polecenia `!important` w definicji stylu:

```
p {font-size: 20pt !important; color: green;}
```

Fragment kodu:

```
<p style= "font-size: 12pt; color: red"> Kaskadowe arkusze stylów</p>
```

Przykład 2.9

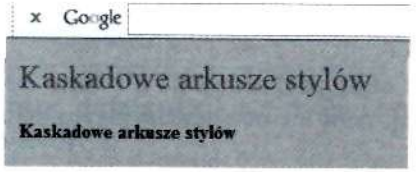
```
<html>
<head>
<title>Arkusz stylów</title>
```



```
<style type="text/css">
body {background-color: #c0c0c0;}
p {font-size: 20pt !important; color: green;}
</style>
</head>
<body>
<p style= "font-size: 12pt; color: red"> Kaskadowe arkusze stylów</p>
<h2 style= "font-size: 12pt;"> Kaskadowe arkusze stylów</h2>
</body>
</html>
```

Na rysunku 2.3 został pokazany efekt działania polecenia `!important`. Dla znacznika `<p>` zastosowano kolor czcionki zdefiniowany za pomocą stylu lokalnego, natomiast rozmiar czcionki pochodzi z wewnętrznego arkusza stylów.

Rysunek 2.3. Efekt działania polecenia !important



Zadanie 2.4
W zewnętrznym arkuszu stylów został zdefiniowany znacznik `h2`:

```
h2 {color: blue; text-align: center; font-size: 12pt;}
```

W wewnętrznym arkuszu stylów również został zdefiniowany znacznik `h2`:

```
h2 {text-align: right; font-size: 24pt;}
```

Jeżeli strona ma powiązanie z zewnętrznym arkuszem stylów, to jakie właściwości będzie miał znacznik `h2`? Sprawdź swoją odpowiedź, pisząc kod przykładowej strony internetowej.

2.1.6. Dziedziczenie

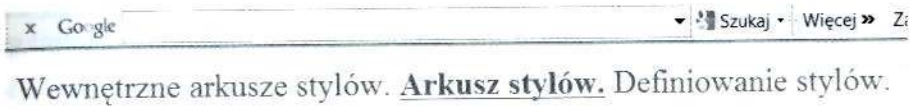
W języku HTML często występuje zagnieżdżanie jednego elementu wewnątrz drugiego. Jeżeli dla elementu nadrzędnego w arkuszach stylów zostały zdefiniowane właściwości, to w większości przypadków elementowi podrzędnemu zostaną przypisane te same właściwości, nawet jeżeli nie zostały dla niego zdefiniowane. Mechanizm ten nazywamy **dziedziczeniem**.

Przykład 2.10

```
<html>
<head>
<title> Wewnętrzny arkusz stylów</title>
<style type="text/css">
body {background-color: yellow;}
p {font-size: 20pt; color: green;}
```

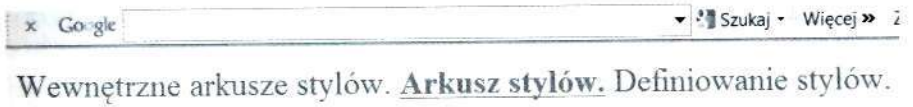
```
</style>
</head>
<body>
<p> Wewnętrzne arkusze stylów. <u><b>Arkusz stylów.</b></u> Definiowanie
stylów. </p>
</body>
</html>
```

W podanym przykładzie dla akapitu <p> został zdefiniowany kolor zielony. Wewnątrz akapitu wystąpiło pogrubienie i podkreślenie, którym nie nadano żadnego stylu, więc dziedziczą go one po akapicie (rysunek 2.4).



Rysunek 2.4. Dziedziczenie stylu dla tekstu „Arkusz stylów."

Na rysunku 2.5 widzimy efekt zdefiniowania atrybutu koloru dla fragmentu tekstu wewnątrz akapitu. Ten fragment nie odziedziczył stylu po akapicie.



Rysunek 2.5. Brak dziedziczenia stylu dla tekstu „Arkusz stylów."

Zadanie 2.5
Kod podany w przykładzie 2.10 zmodyfikuj tak, aby uzyskać wynik pokazany na rysunku 2.5.

2.2. Składnia języka CSS

Arkusz stylów jest zwykłym plikiem tekstowym i można go utworzyć za pomocą dowolnego edytora tekstu. Można również skorzystać ze specjalnego edytora CSS.

Edytory CSS przyspieszają i ułatwiają pracę, ponieważ automatyzują niektóre działania, na przykład można za ich pomocą automatycznie zdefiniować wielkość i kolor czcionki, kolor tła czy kolor odsyłaczy.

Plik będący zewnętrznym arkuszem stylów musi zostać zapisany z rozszerzeniem *.css*.

Jeżeli do arkusza stylów będą importowane inne arkusze stylów, to polecenia importowania `@import` muszą znaleźć się na początku tego arkusza w postaci:

```
@import url (ścieżka/plik.css);
```

W języku CSS parametry formatowania definiuje się za pomocą **reguł stylów**. Każda reguła odnosi się do określonego elementu i składa się z dwóch części: **selektora**

i **deklaracji**. Selektor określa, jakiego elementu ma dotyczyć formatowanie. Deklaracja jest umieszczana w nawiasach klamrowych { . . . } i definiuje formatowanie elementu. Deklaracja musi zawierać przynajmniej jedną właściwość i przypisaną do niej wartość. Poza tym jednym ograniczeniem deklarowanych właściwości może być dowolna liczba. Deklaracja zawsze kończy się średnikiem.

Podstawowa składnia wygląda następująco:

```
selektor {właściwość: wartość;}
```

Powyższa składnia dotyczy stylów wewnętrznych, definiowanych w części nagłówkowej dokumentu HTML, oraz stylów zewnętrznych. Natomiast dla stylów lokalnych składnia ma postać:

```
<znacznik style="właściwość: wartość;">
```

W definicji stylu można grupować selektory i przypisywać im te same wartości:

```
selektor1, selektor2, selektor3 {właściwość: wartość;}
```

Przykład 2.11

```
p, h3, h4 {font-family: calibri; color: blue;}
```

Do dokumentu HTML można dołączyć arkusz stylów zapisany w pliku *arkusz.css* z podaną przykładową zawartością.

Przykład 2.12

```
body {background-color: #DFDFDF; }
p {font-color: blue; font-size: 5}
```

Jeżeli dokument HTML i plik z arkuszem stylów zostaną umieszczone w odpowiednich lokalizacjach i do dokumentu HTML przy użyciu elementu `<link>` zostanie dołączony zewnętrzny plik CSS, to po otwarciu dokumentu zdefiniowane w arkuszu stylów ustawienia dotyczące tła strony oraz wyglądu tekstu będą widoczne na tej stronie.

W pliku CSS można umieszczać komentarze. Komentarze zaczynają się znakami `/*`, a kończą się znakami `*/`.

Przykład 2.13

```
/*To jest komentarz*/
body {background-color: #DFDFDF; }/*tutaj też został dodany komentarz*/
p {color: blue; font-size: 5; font-family: courier new; }
```

2.2.1. Wartości i jednostki

Każda właściwość używana do definiowania selektora zawiera zbiór dopuszczalnych wartości oraz dopuszczalne rodzaje wartości.

Mogą to być:

Liczby — całkowite i rzeczywiste. Liczba całkowita składa się z cyfr od 0 do 9. Liczba rzeczywista może zawierać część całkowitą i część ułamkową. Część ułamkowa zapisana jest po kropce.

Procenty. Wartość procentowa jest zawsze określana względem innej wartości. Może zostać użyta na przykład do określenia wielkości bloków. Podana w procentach wielkość bloku będzie się zmieniała wraz ze zmianą rozmiaru bloku, który go zawiera (rozmiar ten zwykle zależy od rozmiaru okna przeglądarki).

Słowa kluczowe. Nie zawsze wartości wyrażone za pomocą słów kluczowych są poprawnie interpretowane przez przeglądarki.

Jednostki długości. Mogą być zapisywane jako wartości bezwzględne i wartości względne.

Wartości bezwzględne: `in` (cal, 1 in = 2,54 cm), `cm` (centymetr), `mm` (milimetr), `pt` (punkty, 1 cal = 72 punkty), `pc` (pica, 1 pc = 12 pt), `px` (piksele, 1 px = 0,75 pt). Jednostki bezwzględne są rzadko stosowane. Należy ich używać wtedy, gdy chcemy, aby wybrany element miał taki sam rozmiar niezależnie od rozdzielczości monitora.

Wartości względne: `em` (1 em jest wielkością rozmiaru czcionki; wartość 0,1 em jest najmniejszą wartością, którą przeglądarki są w stanie rozpoznać), `ex` (wysokość małej litery *x* w użytej czcionce). Jednostki względne są stosowane, gdy odwołujemy się do rozmiaru innego elementu.

URL — np. `url(zdjecia/obraz.jpg)`. Błędem jest wstawienie spacji przed nawiasem otwierającym.

W języku HTML przyjmuje się, że wartości liczbowe bez podanych jednostek mają jednostkę `px`, natomiast w języku CSS brak jednostki jest traktowany jako błąd. Niektóre przeglądarki ignorują ten błąd i przyjmują jednostkę `px`. Gdy podana wartość jest równa 0, jednostka nie ma znaczenia.

Kolory. Stosuje się je na takich samych zasadach jak w języku HTML. W wersji CSS 2.2 oprócz 16 podstawowych kolorów został zdefiniowany dodatkowy kolor podstawowy `orange` (`#FFA500`) — pomarańczowy. W CSS 3 zostało zdefiniowanych dodatkowo 130 kolorów rozszerzonych.

Przykłady podawania wartości kolorów w CSS 3:

1. *predefiniowane* — podawanie kolorów w postaci ich nazw.
```
p {background-color: orange; }
```
2. *#RRGGBB* — podawanie wartości każdej składowej kolorów podstawowych (`RR` — czerwony, `GG` — zielony, `BB` — niebieski). Wszystkie liczby są dwucyfrowe i zapisane w systemie szesnastkowym, np. `#FF00CC`.
```
p {background-color: #ff0000; }
```
3. *#RGB* — jako składowe kolorów trzeba podawać liczby jednocyfrowe w systemie szesnastkowym, np. `#F0C`.
```
p {background-color: #f00; }
```
4. `rgb (R, G, B)` — można podać oddzielnie każdą składową koloru w systemie dziesiętnym.
```
p {background-color: rgb (255,0,0); }
```


5. `rgb (R%, G%, B%)` — można podać oddzielnie każdą składową koloru w postaci wartości procentowej.
6. *28 kolorów systemowych* — kolory pochodzą z systemu operacyjnego (kolor pulpitu, okien, menu). Gdy użytkownik zmieni kolory w swoim systemie, zmienią się również kolory zdefiniowane tą metodą.
7. `hsl (H, S%, L%)` — odmienny od RGB sposób opisu kolorów. `H` (*Hue*) określa barwę, `S` (*Saturation*) — nasycenie, `L` (*Lightness*) — jasność. Kolor powstaje przez wybranie określonej barwy (`H`), a następnie sterowanie nasyceniem (`S`) i jasnością (`L`) aż do uzyskania pożądanego efektu.
```
p {background-color: hsl (120,65%,75%); }
```
8. `rgba (R, G, B, A)`, `rgba (R%, G%, B%, A%)`, `hsla (H, S%, L%, A)` — uzyskiwanie koloru z przezroczystością. `A` (*alpha*) określa wartość przezroczystości z zakresu od `0` (całkowita przezroczystość) do `1` (brak przezroczystości).
```
p {background-color: rgba (255,0,0,0.5); }
p {background-color: hsla (120,65%,75%,0.3); }
```

2.3. Selektory

Selektorem może być dowolny element języka HTML, dla którego chcemy zdefiniować parametry formatowania. W zależności od tego, w jaki sposób odwołujemy się w definicji reguły do formatowanych elementów, wyróżniamy następujące rodzaje selektorów:

- selektory elementów,
- selektory atrybutów,
- selektory specjalne,
- selektory pseudoklas,
- selektory pseudoelementów.

2.3.1. Selektory elementów

Składnia dla selektorów elementów to podana już podstawowa składnia języka CSS:

```
selektor {właściwość: wartość;}
```

Selektor typu
Jest to podstawowy typ selektora. Służy do definiowania formatowania znaczników występujących na stronie (np.: `p`, `h3`, `table`, `img`).

Przykład 2.14
```
p {color: blue; font-size: 4pt;}
h3 {color: green; font-size: 24pt;}
```

Selektor uniwersalny
Selektor uniwersalny (inaczej: ogólny) to selektor pasujący do wszystkich znaczników. Jest oznaczany gwiazdką `*`.

Zamiast grupować elementy, np.:

```
p, h1, h2, h3, h4, table {font-family: courier new; color: green;}
```

można użyć selektora uniwersalnego:

```
* {font-family: courier new; color: green;}
```

Selektor potomka

Przy użyciu selektora potomka można formatować elementy, które są zawarte wewnątrz innych znaczników, czyli leżą niżej w hierarchii drzewa dokumentu.

Jeżeli w dokumencie HTML wewnątrz znacznika znajduje się inny znacznik, np. wewnątrz znacznika <h2> zostały umieszczone znaczniki <u> oraz <i>:

```
<h2> Wewnętrzne arkusze stylów. <u><i>Definiowanie stylów. </i></u>Arkusz
stylów. </h2>
```

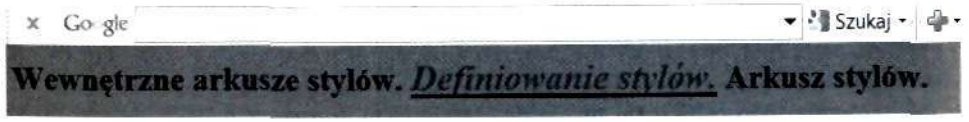
to definicja stylu dla znacznika <i> będzie wyglądać następująco:

```
h2 i { font-size: 20pt; color: blue; }
```

Przykład 2.15

```
<html>
<head>
<title> Selektor potomka</title>
<style type="text/css">
body {background-color: grey;}
h2 i {font-size: 20pt; color: blue;}
</style>
</head>
<body>
<h2> Wewnętrzne arkusze stylów. <u><i> Definiowanie stylów. </i></u>
Arkusz stylów.</h2>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.6.



Rysunek 2.6. Selektor potomka

Jak widać, potomek nie musi być umieszczony bezpośrednio w znaczniku, którego jest potomkiem. W powyższym przykładzie w znaczniku <h2> znajduje się znacznik <u>, a dopiero w nim znacznik <i>, którego dotyczy definiowany styl.

Selektor potomka jest często wykorzystywany do definiowania odsyłaczy występujących wewnątrz bloku.

Selektor dziecka

Selektor dziecka służy do definiowania formatowania elementów, które znajdują się o jeden rząd niżej w hierarchii drzewa dokumentu. Ma on postać:

```
rodzic > dziecko {właściwość: wartość;}
```

gdzie symbol > oznacza bezpośredni związek między elementami.

Znacznik będący dzieckiem musi wystąpić bezpośrednio wewnątrz znacznika nadrzędnego.

Przykład 2.16

```
<html>
<head>
<title> Selektor dziecka</title>
<style type="text/css">
body {background-color: #c0c0c0;}
p>u {font-size: 20pt; color: yellow;}
</style>
</head>
<body>
<p> Wewnętrzne arkusze stylów. <u><i>Definiowanie stylów.</i></u> Arkusz
stylów.</p>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.7.



Rysunek 2.7. Selektor dziecka

Kolor *żółty* i wielkość liter 20pt zostały nadane tylko tekstom otoczonym znacznikiem <u> i umieszczonym bezpośrednio w akapicie (znacznik <p>).

Selektor braci

Dla elementów znajdujących się w tym samym rzędzie hierarchii można zdefiniować selektor braci. Umożliwia on nadanie drugiemu z sąsiadujących elementów zdefiniowanych atrybutów formatowania. Selektor braci przyjmuje postać:

```
brat1 + brat2 {właściwość: wartość;}
```

gdzie symbol + oznacza połączenie elementów.

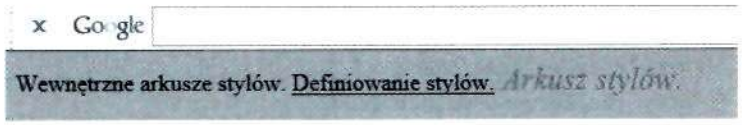
Przykład 2.17

```
<html>
<head>
<title> Selektor braci</title>
<style type="text/css">
```

```
body {background-color: #c0c0c0;}
u + i {font-size: 16pt; color: red;}
</style>
</head>
<body>
<p> Wewnętrzne arkusze stylów. <u>Definiowanie stylów.</u><i> Arkusz
stylów.</i> </p>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.8.

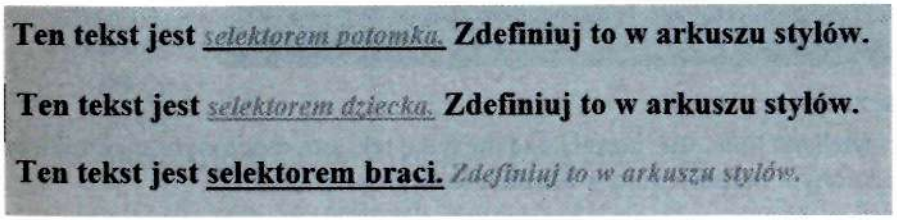
Rysunek 2.8. Selektor braci



W powyższym przykładzie znaczniki `<u>` oraz `<i>` znajdują się w tym samym rzędzie hierarchii, dlatego można dla nich ustawić selektor braci. Znacznik `<i>` występuje bezpośrednio po znaczniku `<u>` i tylko dla tekstu otoczonego tym znacznikiem jest realizowane formatowanie zdefiniowane dla selektora braci.

Zadanie 2.6

Wykorzystując arkusze stylów oraz selektory potomka, dziecka i braci, zdefiniuj style, które pozwolą na wyświetlenie tekstów sformatowanych tak jak na rysunku 2.9. Cały tekst powinien zostać zdefiniowany przy użyciu znacznika `<h2>`, natomiast wyróżnione fragmenty — za pomocą znaczników `<i>` oraz `<u>`.



Rysunek 2.9. Efekt użycia selektorów potomka (pierwszy wiersz), dziecka (drugi wiersz) i braci (trzeci wiersz)

2.3.2. Selektory atrybutów

W języku CSS można formatować znaczniki na podstawie posiadanych przez nie atrybutów. Na przykład jeżeli dla znacznika `<p>` (akapit) zostanie zdefiniowany atrybut `align` (wyrównanie), to inaczej będzie wyświetlany akapit wyrównany do lewej, a inaczej akapit wyśrodkowany.

Postać selektora atrybutów:

```
selektor [atrybut="wartość atrybutu"] {właściwość: wartość;}
```


Przykład 2.18

```
p [align="center"] {font-size: 16pt; color: yellow;}
```

Prosty selektor atrybutu

Prosty selektor atrybutu jest wykorzystywany dla elementów o nadanym określonym atrybucie, którego wartość nie ma znaczenia. Ma on postać:

```
selektor [atrybut] {właściwość: wartość;}
```

lub

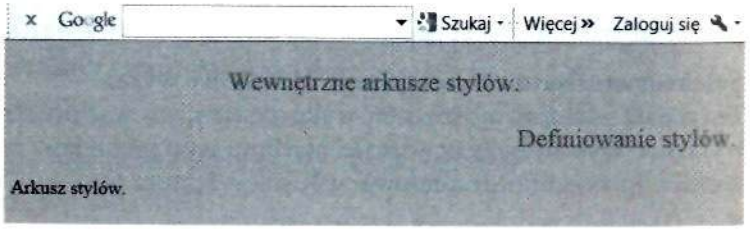
```
[atrybut] {właściwość: wartość;}
```

Przykład 2.19

```
<html>
<head>
<title> Selektor atrybutu</title>
<style type="text/css">
body {background-color: #c0c0c0;}
p[align]{font-size: 16pt; color: red;}
</style>
</head>
<body>
<p align="center">Wewnętrzne arkusze stylów.</p>
<p align="right">Definiowanie stylów.</p>
<p> Arkusz stylów.</p>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.10.

Rysunek 2.10. Prosty selektor atrybutu



W powyższym przykładzie dla akapitu pierwszego i drugiego został zdefiniowany atrybut `align` o różnych wartościach. Niezależnie od wartości tego atrybutu akapity zostały sformatowane zgodnie z definicją stylu dla znacznika `<p>`. Ponieważ trzeci akapit nie ma zdefiniowanego atrybutu `align`, nie jest formatowany.

Selektor atrybutu o określonej wartości

Ten rodzaj selektora określa formatowanie, gdy atrybut ma określoną wartość. Ma on postać:

```
selektor [atrybut="wartość"] {właściwość: wartość;}
```

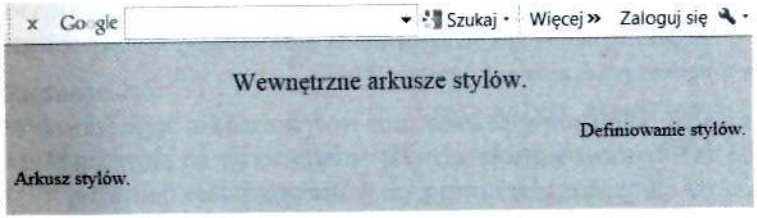
lub

```
[atrybut="wartość"] {właściwość: wartość;}
```

Przykład 2.20

```
<html>
<head>
<title> Selektor atrybutu</title>
<style type="text/css">
body {background-color: #c0c0c0;}
p[align="center"]{font-size: 16pt; color: blue;}
</style>
</head>
<body>
<p align="center" > Wewnętrzne arkusze stylów.</p>
<p align="right">Definiowanie stylów.</p>
<p> Arkusz stylów.</p>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.11.



Rysunek 2.11. Selektor atrybutu o określonej wartości

W tym przykładzie tylko pierwszy akapit został sformatowany zgodnie z definicją stylów, ponieważ tylko tu został zdefiniowany atrybut `align` z wartością `center`.

Selektor atrybutu zawierającego określony wyraz

Ten rodzaj selektora może zostać wykorzystany, gdy wartość atrybutu składa się z kilku wyrazów. Wystarczy, że w wartości atrybutu wystąpi podany wyraz, aby dany element został odpowiednio sformatowany. Postać selektora to:

```
selektor [atrybut~="wyraz"] {właściwość: wartość;}
```

lub

```
[atrybut~="wyraz"] {właściwość: wartość;}
```

Przykład 2.21

```
<html>
<head>
<title> Selektor atrybutu</title>
<style type="text/css">
body {background-color: #c0c0c0;}
p[title~="jest"]{font-size: 16pt; color: green;}
</style>
</head>
<body>
<p title="To jest mój styl" > Wewnętrzne arkusze stylów.</p>
<p title="To jest mój akapit">Definiowanie stylów.</p>
<p title="Uwaga na style"> Arkusz stylów.</p>
</body>
</html>
```


Zadanie 2.7

Wykorzystując arkusze stylów oraz selektory atrybutów, zdefiniuj style, które pozwolą na wyświetlenie tekstów umieszczanych w akapicie w różnych kolorach w zależności od wyrównania tych tekstów na stronie (np. dla tekstów wyrównanych do lewej będzie to kolor niebieski, dla tekstów wyśrodkowanych kolor zielony, a dla tekstów wyrównanych do prawej kolor czerwony). Niech dla tekstów umieszczonych w znaczniku `<h2>` kolory będą różne w zależności od zastosowanej wielkości czcionki (np. czcionka o rozmiarze `16pt` — kolor żółty, czcionka o rozmiarze `20pt` — kolor granatowy).

2.3.3. Selektory specjalne

Selektor klasy

Selektory pozwalają na nadanie określonych atrybutów takim samym elementom występującym na stronie. Czasami jednak chcemy zastosować pewien styl formatowania do jednej grupy elementów, a do drugiej inny. Możemy wtedy dla każdej grupy utworzyć klasę o dowolnej nazwie i zdefiniować styl dla selektora klasy.

Selektor klasy ma postać:

```
selektor.klasa { właściwość: wartość; }
```

gdzie *klasa* to dowolna nazwa.

Nazwa klasy musi spełniać kilka warunków:

- Musi zaczynać się od kropki.
- Mogą być w niej użyte litery, cyfry, znak podkreślenia, łącznik.
- Po kropce musi zostać wpisana litera.
- Rozróżniane są małe i wielkie litery.

Przykład 2.22

```
p.tekst1 {color: red; font-size: 24pt;}
p.tekst2 {color: green; font-size: 20pt;}
p.tekst3 {font-family: courier; color: blue; font-size: 16pt;}
```

Odwołanie do klasy w dokumencie HTML ma postać:

```
<znacznik class="nazwa_klasy"> .... </znacznik>
```

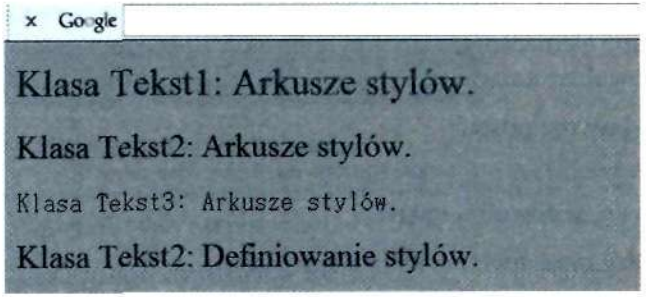
W odwołaniu nazwa klasy jest wartością atrybutu `class`. W nazwie klasy podawanej jako wartość atrybutu nie wstawia się kropki.

Przykład 2.23

```
<html>
<head>
<title> Selektor atrybutu</title>
<style type="text/css">
body {background-color: #c0c0c0;}
p.tekst1 {color: red; font-size: 24pt;}
p.tekst2 {color: green; font-size: 20pt;}
p.tekst3 {font-family: courier; color: blue; font-size: 16pt;}
</style>
</head>
<body>
<p class="tekst1"> Klasa Tekst1: Arkusze stylów.</p>
<p class="tekst2"> Klasa Tekst2: Arkusze stylów.</p>
<p class="tekst3"> Klasa Tekst3: Arkusze stylów.</p>
<p class="tekst2"> Klasa Tekst2: Definiowanie stylów.</p>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.12.

Rysunek 2.12. Selektor klasy



Zdarza się, że definiując klasę, nie określamy, jakich znaczników będzie ona dotyczyła. Tak zdefiniowana klasa może zostać użyta do sformatowania dowolnego elementu bez względu na typ znacznika. Ma ona postać:

```
.klasa { właściwość: wartość; }
```

Każdy element, któremu nadamy klasę o podanej nazwie, zostanie sformatowany zgodnie z definicją stylu dla klasy.

Przykład 2.24

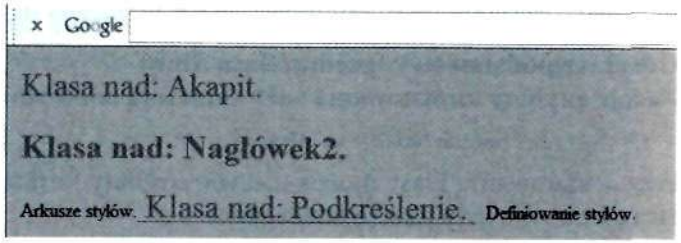
```
<html>
<head>
<title> Selektor atrybutu</title>
<style type="text/css">
body {background-color: #c0c0c0;}
.nad {color: red; font-size: 20pt;}
</style>
```



```
</head>
<body>
<p class="nad"> Klasa nad: Akapit.</p>
<h2 class="nad"> Klasa nad: Nagłówek2.</h2>
Arkusze stylów.<u class="nad"> Klasa nad: Podkreślenie. </u>
Definiowanie stylów.
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.13.

Rysunek 2.13. Efekt zastosowania klasy do formatowania różnych rodzajów znaczników



Deklaracja klasy jest przydatna zwłaszcza wtedy, gdy w różnych miejscach witryny mają występować elementy o takich samych parametrach formatowania, lecz nie możemy (lub nie chcemy) przy ich definiowaniu posługiwać się selektorem typu.

Selektor identyfikatora

Selektor identyfikatora jest stosowany, gdy chcemy nadać określone atrybuty formatowania elementowi, który ma przypisany jednoznaczny identyfikator. Selektor identyfikatora ma postać:

```
selektor#identyfikator {właściwość: wartość;}
```

lub

```
#identyfikator {właściwość: wartość;}
```

Deklaracja identyfikatora musi rozpoczynać się znakiem #.

Przykład 2.25

```
h3#naglo3 {color: yellow;}
```

Odwołanie do identyfikatora w kodzie ma postać:

```
<h3 id="naglo3"> Styl nagłówkowy</h3>
```

2.3.4. Pseudoklasy

W języku CSS style są dodawane do elementów lub grup elementów na podstawie ich nazw, atrybutów lub zawartości.

Jest też inna możliwość dodawania stylu. Element może nabywać styl lub tracić go w związku z działaniem użytkownika lub zmieniać się w zależności od umiejscowienia. Przykładem takiego zachowania są linki (*odsyłacze*), które zmieniają swój wygląd po

najechaniu na nie kursorem myszy, ich kliknięciu lub otwarciu. Do takiej dynamicznej zmiany stylu elementu służą **pseudoklasy**.

Odsyłacze (linki) mają właściwości, które określają działanie użytkownika. Są to:

- `:active` — link odwiedzany, strona jest aktualnie wczytana;
- `:link` — link nieaktywny, nie został przez użytkownika odwiedzony;
- `:visited` — link odwiedzony, strona była otwierana;
- `:hover` — link gotowy do kliknięcia, kursor myszy ustawiony nad linkiem.

Przy użyciu specjalnych selektorów dla odsyłaczy można definiować style, które będą nadawane odsyłaczom w zależności od ich bieżącego stanu.

Odsyłacz podstawowy (pseudoklasa :link)

Nadaje atrybuty formatowania wszystkim jeszcze nieodwiedzonym odnośnikom.

```
a:link {właściwość: wartość;}
```

Przy deklarowaniu klasy można ustawić atrybuty formatowania dla wybranych odsyłaczy, które nie były jeszcze odwiedzane.

```
a.nazwa_klasy:link {właściwość: wartość;}
```

Przykład 2.26

```
a:link {color: green; background: yellow;}
a.nowa:link {color: orange;}
```

Odsyłacz odwiedzony (pseudoklasa :visited)

Nadaje atrybuty formatowania odsyłaczowi, jeśli ten był odwiedzony, a informacja o tym została umieszczona w pamięci przeglądarki.

```
a:visited {właściwość: wartość;}
```

Z deklaracją klasy ustawiane są atrybuty formatowania dla odsyłaczy wybranej klasy, które były otwierane.

```
a.nazwa_klasy:visited {właściwość: wartość;}
```

Przykład 2.27

```
a:visited {color: red;}
a.po_nowa:visited {color: red;}
```

Wskazanie kursorem myszy (pseudoklasa :hover)

Nadaje atrybuty formatowania odsyłaczowi, gdy mysz znajduje się nad nim, ale nie aktywuje go.

```
a:hover {właściwość: wartość;}
```

Z deklaracją klasy ustawiane są atrybuty formatowania dla odsyłaczy wybranej klasy, które są w danej chwili wskazane myszą.

```
a.nazwa_klasy:hover {właściwość: wartość;}
```

Przykład 2.28

```
a:hover {color: yellow;}
a.po_nowa:hover {color: yellow;}
```


UWAGA

Pseudoklasa `:hover` może być używana do definiowania stylów elementów innych niż odsyłacze.

Przykład 2.29

```
<html>
<head>
<title> Pseudoklasa :hover</title>
<style type="text/css">
body {background-color: #c0c0c0;}
.blok {height: 25px; width: 200px; background: aqua; color: black; }
.blok:hover { background: yellow; color: red; }
</style>
</head>
<body>
<div class="blok"> Strony internetowe</div>
</body>
</html>
```

Przykład pokazuje zastosowanie pseudoklasy `:hover` dla znacznika `<div>`.

Odsyłacz aktywny (pseudoklasa :active)

Nadaje atrybuty formatowania odsyłaczowi, który w danej chwili jest aktywny, np. gdy użytkownik naciśnie i przytrzyma przycisk myszy.

```
a:active {właściwość: wartość;}
```

Z deklaracją klasy ustawiane są atrybuty formatowania dla odsyłaczy wybranej klasy, które są w danej chwili aktywne.

```
a.nazwa_klasy:active {właściwość: wartość;}
```

Przykład 2.30

```
a:active {color: blue;}
a.po_nowa:active {color: blue;}
```

Kolejność deklarowania w arkuszu stylów pseudoklas przypisanych do odsyłaczy ma znaczenie dla ich prawidłowego działania.

Oto poprawna kolejność deklarowania pseudoklas:

```
a:link {właściwość: wartość; }
a:visited {właściwość: wartość; }
a:hover {właściwość: wartość; }
a:active {właściwość: wartość; }
```

Przykład 2.31

```
<html>
<head>
<title> Pseudoklasa odnośnik</title>
```

```
<style type="text/css">
body {background-color: #c0c0c0;}
a:link {color: green; }
a:visited {color: red; }
a:hover {color: yellow; }
a:active {color: blue; }
</style>
</head>
<body>
<p> Link do strony wydawnictwa: <a href="http://helion.pl/">HELION</a></p>
</body>
</html>
```

Pseudoklasa :focus
Nadaje atrybuty formatowania odsyłaczowi (wcześniej wybranemu) lub polu formularza, na którym został ustawiony kursor, na przykład gdy odsyłacz został wybrany za pośrednictwem klawisza *Tab* lub w polu formularza znalazł się kursor.

```
selektor :focus {właściwość: wartość; }
```

Zadanie 2.8
Wykorzystując arkusze stylów, zdefiniuj pseudoklasy, które umożliwią dynamiczną zmianę stylu linków znajdujących się na przykładowej stronie internetowej. Link podstawowy powinien mieć kolor czerwony, odwiedzany — kolor zielony, wskazany przez kursor myszy — kolor pomarańczowy, a link aktywny — kolor żółty.

2.3.5. Selektory pseudoelementów

W języku HTML nie ma mechanizmów dostępu do takich elementów strony jak pierwsza litera lub pierwsza linia akapitu bez otaczania ich znacznikami. Do formatowania tych specjalnych elementów można wykorzystać **pseudoelementy** dostępne w języku CSS.

Pierwsza linia (:first-line)
Pseudoelement `:first-line` nadaje określone formatowanie wszystkim pierwszym liniom znacznika, do którego odnosi się selektor.

```
selektor :first-line {właściwość: wartość; }
```

Selektorem może być dowolny znacznik języka HTML.

Przykład 2.32

```
p :first-line { color: red; font-size: 16pt;}
```

Pierwsza litera (:first-letter)
Pseudoelement `:first-letter` nadaje odrębne formatowanie pierwszej literze np. akapitu. Służy głównie do poprawienia wyglądu tekstu poprzez zaprojektowanie ozdobnej pierwszej litery (inicjału).

```
selektor :first-letter {właściwość: wartość; }
```

Przykład 2.33

```
p :first-letter { font-family: arial; font-size: 24pt; color: blue; }
```


2.4. Właściwości elementów

W kaskadowych arkuszach stylów formatowanie elementów strony jest realizowane przez ustawianie właściwości tych elementów.

2.4.1. Czcionki

Czcionki są najistotniejszym elementem strony, który podlega formatowaniu.

Rodzaj czcionki
Ze względu na to, że nie wszystkie czcionki są dostępne na komputerze użytkownika, przy wyborze rodzaju czcionki określa się jej preferowany typ. Można również zdefiniować kilka rodzajów czcionek. Jeżeli na komputerze użytkownika nie ma pierwszego rodzaju, to z listy zostanie wybrany kolejny.

Rodzaj czcionki ustala się za pomocą atrybutu `font-family`.

```
selektor {font-family: rodzaj1, rodzaj2, rodzaj3, ...}
```

Przy wprowadzaniu kilku rodzajów czcionek należy ich nazwy oddzielić przecinkiem i spacją. Jeżeli nazwa czcionki składa się z kilku wyrazów, całą nazwę ujmujemy w apostrofy, np. `'Times New Roman'`.

Zamiast nazwy czcionki można podać nazwę rodziny czcionek.

Nazwy rodzin czcionek to:

- `serif` — czcionki szeryfowe, mające wykończenie liter i cyfr. Do tej grupy należą m.in.: Times New Roman, Georgia, Bodoni.
- `sans-serif` — czcionki bezszeryfowe. Końcówki znaków są proste; np.: Arial, Verdana, Futura.
- `monospace` — czcionki monotypiczne. Znaki mają stałą szerokość; np. Courier, Courier New.
- `cursive` — czcionki mające cechy czcionki pochyłej. Znaki wyglądają jak pisane ręcznie; np. Comic Sans, Florence.
- `fantasy` — czcionki dekoracyjne; np. Impact, OldTown.

Przykład 2.34

```
<p style="font-family: Georgia, Courier, Arial"> Arkusze stylów</p>
<p style="font-family: Georgia, Courier, serif"> Arkusze stylów</p>
```

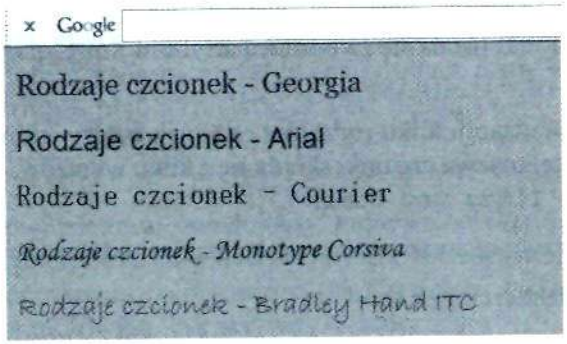
Przykład 2.35

```
<html>
<head>
<title>Rodziny czcionek</title>
<style type="text/css">
body {background-color: #c0c0c0;}
p {font size: 18pt; color: green;}
</style>
```

```
</head>
<body>
<p style="font-family: Georgia;"> Rodzaje czcionek - Georgia</p>
<p style="font-family: Arial;"> Rodzaje czcionek - Arial</p>
<p style="font-family: Courier;"> Rodzaje czcionek - Courier</p>
<p style="font-family: 'Monotype Corsiva';"> Rodzaje czcionek - Monotype
Corsiva</p>
<p style="font-family: 'Bradley Hand ITC';"> Rodzaje czcionek - Bradley
Hand ITC</p>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.14.

Rysunek 2.14. Zastosowanie różnych rodzajów czcionek



Rozmiar czcionki

Wielkość czcionki jest definiowana w atrybucie `font-size`.

```
selektor {font-size: rozmiar;}
```

Można ją ustawiać na cztery sposoby:

1. **Według słów kluczowych**, które pozwalają określić rozmiar czcionki względem rozmiaru podstawowego. W języku CSS występuje siedem słów kluczowych:

- `xx-small` — czcionka najmniejsza,
- `x-small` — czcionka mniejsza,
- `small` — czcionka mała,
- `medium` — czcionka średnia (jest to podstawowy rozmiar czcionki),
- `large` — czcionka duża,
- `x-large` — czcionka większa,
- `xx-large` — czcionka największa.

Przykład 2.36

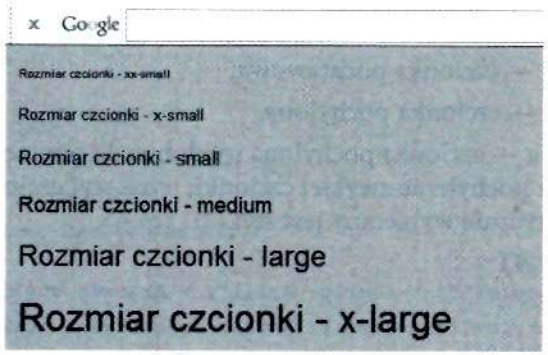
```
<html>
<head>
<title>Rozmiar czcionki</title>
<style type="text/css">
```



```
body {background-color: #c0c0c0;}
p {color: blue; font-family: Arial;}
</style>
</head>
<body>
<p style="font-size: xx-small;"> Rozmiar czcionki - xx-small</p>
<p style="font-size: x-small;"> Rozmiar czcionki - x-small</p>
<p style="font-size: small;"> Rozmiar czcionki - small</p>
<p style="font-size: medium;"> Rozmiar czcionki - medium</p>
<p style="font-size: large;"> Rozmiar czcionki - large</p>
<p style="font-size: x-large;"> Rozmiar czcionki - x-large</p>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.15.

Rysunek 2.15. Zastosowanie różnych rozmiarów czcionki



Przykład 2.37

```
<p style="font-size: x-small"> Arkusze stylów</p>
<p style-"font-size: large"> Arkusze stylów</p>
```

2. **Za pomocą wartości względnych** — wielkość czcionki jest ustalana względem wielkości czcionki elementu nadrzędnego.

- `smaller` — czcionka mniejsza od bieżącej,
- `larger` — czcionka większa od bieżącej.

Przykład 2.38

```
<p style="font-size: smaller"> Arkusze stylów</p>
<p style="font-size: larger"> Arkusze stylów</p>
```

3. **Poprzez wielkość podaną w jednostkach miary.** Jednostki, które mogą zostać użyte do określenia wielkości czcionki, to:

- *piksele* (`px`),
- *punkty* (`pt`),
- *centymetry* (`cm`),
- *cale* (`in`),

- *milimetry* (mm),
- *pica* (pc).

Przykład 2.39

```
<p style="font-size: 12pt"> Arkusze stylów</p>
<p style="font-size: 1cm"> Arkusze stylów</p>
```

4. **Poprzez wielkość podaną w procentach.** Każda przeglądarka ma zdefiniowany podstawowy rozmiar tekstu. Wartości procentowe określają wielkość czcionki w stosunku do rozmiaru podstawowego.

Przykład 2.40

```
<p style="font-size: 120%"> Arkusze stylów</p>
<p style="font-size: 75%"> Arkusze stylów</p>
```

Styl czcionki

Styl czcionki jest definiowany za pomocą atrybutu `font-style`.

```
selektor {font-style: styl }
```

Dostępne są następujące style:

- `normal` — czcionka podstawowa,
- `italic` — czcionka pochylona,
- `oblique` — czcionka pochylona (podobna do poprzedniej; może być wygenerowana przez pochylenie zwykłej czcionki; jeżeli styl czcionki `italic` jest niedostępny, automatycznie wybierany jest styl `oblique`).

Przykład 2.41

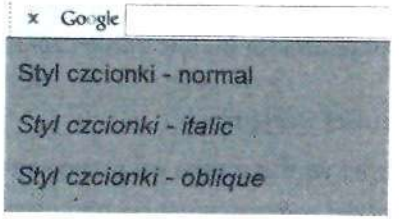
```
<h3 style="font-family: Arial"; > Arkusze stylów. </h3>
<h2 style="font-family: Arial; font-style: italic;" > Arkusze stylów.
Styl Italic. </h2>
```

Przykład 2.42

```
<html>
<head>
<title>Styl czcionki</title>
<style type="text/css">
body {background-color: #c0c0c0;}
p {font-size: 16pt; color: red; font-family: Arial;}
</style>
</head>
<body>
<p  > Styl czcionki - normal</p>
<p  style= "font-style: italic; "> Styl czcionki - italic</p>
<p  style= "font-style: oblique; "> Styl czcionki - oblique</p>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.16.

Rysunek 2.16. Zastosowane różne style czcionki



Wariant czcionki

Atrybut `font-variant` pozwala wybrać między czcionką normalną a kapitalikami (wyświetlanie wielkich liter o rozmiarze czcionki przeznaczonym dla małych liter).

```
selektor {font-variant: wartość; }
```

Atrybut `font-variant` może przyjmować wartości:

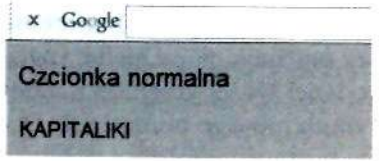
- `normal` — czcionka normalna,
- `small-caps` — kapitaliki.

Przykład 2.43

```
<html>
<head>
<title>Kapitaliki</title>
<style type="text/css">
body {background-color: #c0c0c0;}
p {font-size: 16pt; font-family: Arial;}
</style>
</head>
<body>
<p>Czcionka normalna</p>
<p style= "font-variant: small-caps; "> kapitaliki </p>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.17.

Rysunek 2.17. Czcionka normalna i kapitaliki



Grubość czcionki

Za pomocą atrybutu `font-weight` można określić grubość czcionki.

```
selektor {font-weight: wartość; }
```

Grubość czcionki określana jest następującymi wartościami:

- `normal` — czcionka normalna,
- `bold` — czcionka pogrubiona,

- `100`, `200`, `300`, `400` (równoważne z `normal`), `500`, `600`, `700` (równoważne z `bold`), `800`, `900` — określają grubość czcionki. Nie każda czcionka ma wszystkie dziewięć stopni grubości.

Można również używać wartości względnych:

- `lighter` — czcionka mniej wytłuszczona od przypisanej domyślnie,
- `bolder` — czcionka bardziej wytłuszczona od przypisanej domyślnie.

Odstęp między wierszami

Do określenia wielkości odstępu między wierszami tekstu zapisanego w bloku służy atrybut `line-height`.

```
selektor {line-height: odstep; }
```

Jako wartość atrybutu należy podać konkretną wielkość odstępu. Może to być:

- *liczba* — określa odstęp jako wielokrotność aktualnego rozmiaru czcionki;
- *wysokość* — jest podawana z jednostką miary: `px`, `pt`, `cm` itp., określa stałą wysokość odstępu;
- *wartość procentowa* — określa odstęp jako wartość procentową aktualnego rozmiaru czcionki.

Łączenie właściwości czcionki

Podczas definiowania formatu czcionki można użyć następujących atrybutów:

- `font-style`,
- `font-variant`,
- `font-weight`,
- `font-size`,
- `line-height`,
- `font-family`.

Często czcionce przypisywanych jest wiele atrybutów równocześnie. W takim przypadku można użyć wspólnego polecenia zawierającego wszystkie atrybuty dotyczące czcionek.

```
selektor {font: wartości atrybutów; }
```

Jako wartości atrybutów powinny zostać podane konkretne wartości, które będą definiowane dla czcionek. Należy oddzielić je od siebie spacjami. Wartości atrybutów powinny zostać wpisane w podanej wyżej kolejności. Zmiana kolejności może spowodować złą interpretację atrybutu lub jego pominięcie. Dodatkowo przed wartością atrybutu `line-height` trzeba postawić ukośnik (`/`). Niektóre z wartości atrybutów mogą zostać pominięte, ale wartości dla atrybutów `font-size` i `font-family` są wymagane.

Przykład 2.44

```
<p style="font: italic bold 12pt /20pt Georgia">
```


Zadanie 2.9

Wykorzystując arkusze stylów, zdefiniuj style określające właściwości czcionki (rodzaj, rozmiar, styl) dla znaczników: `<h2>`, `<h3>`, `<p>`.

2.4.2. Tekst

Język CSS pozwala na dowolne formatowanie tekstu poprzez dodanie do niego stylu. Umożliwia to nie tylko szybką zmianę wyglądu tekstu na stronie, ale także swobodne manipulowanie tekstem i umieszczaną w nim grafiką.

Wcięcie tekstu

Przy użyciu atrybutu `text-indent` można definiować wcięcie pierwszego wiersza akapitu.

```
selektor {text-indent: wartość; }
```

Jako wartość należy podać wielkość wcięcia.

Przykład 2.45

```
p { text-indent: 20px; }
```

Wyrównanie tekstu

Wyrównanie tekstu definiujemy za pomocą atrybutu `text-align`. Odpowiada on atrybutowi `align` w języku HTML.

```
selektor {text-align: wartość; }
```

Wyrównanie określa się za pomocą wartości:

- `left` — wyrównanie tekstu do lewego marginesu;
- `right` — wyrównanie tekstu do prawego marginesu;
- `center` — wyśrodkowanie tekstu;
- `justify` — rozłożenie tekstu równomiernie między marginesami.

Przykład 2.46

```
p { text-align: justify; }
h2 { text-align: right; }
h3 { text-align: center; }
```

Ozdabianie tekstu

Do dodawania efektów takich jak: podkreślenie, przekreślenie tekstu, umieszczenie linii nad tekstem, używany jest atrybut `text-decoration`.

```
selektor {text-decoration: wartość; }
```

Rodzaj efektu określa się za pomocą wartości:

- `none` — bez zmian;
- `underline` — tekst podkreślony;
- `overline` — linia umieszczona nad tekstem;
- `line-through` — tekst przekreślony;
- `blink` — migotanie tekstu (nie wszystkie przeglądarki interpretują ten efekt).

Przykład 2.47

```
p { text-decoration: underline; }
h2 { text-decoration: overline; }
h3 { text-decoration: line-through; }
```

Podane wartości można łączyć w jednej deklaracji.

Przykład 2.48

```
p { text-decoration: underline line-through overline; }
```

Zadanie 2.10

Sprawdź, jaki efekt powstanie po zastosowaniu deklaracji podanej w przykładzie.

Odstęp między literami

Do definiowania odstępu między literami tekstu stosuje się atrybut `letter-spacing`.

```
selektor { letter-spacing: wartość; }
```

Przykład 2.49

```
p { letter-spacing: 3px; }
```

Odstęp między wyrazami

Do definiowania odstępu między wszystkimi wyrazami tekstu stosuje się atrybut `word-spacing`.

```
selektor { word-spacing: wartość; }
```

Przykład 2.50

```
p { word-spacing: 12px; }
```

Transformacja tekstu

Do ustawiania wielkości liter tekstu służy atrybut `text-transform`. Atrybut ten kontroluje wielkość liter w tekście i dokonuje ich transformacji.

```
selektor { text-transform: wartość; }
```

Atrybut może przyjmować następujące wartości:

- `capitalize` — zmienia na wielkie litery wszystkie pierwsze litery wyrazów w tekście, np. słowa „jan kowalski" zmieni na „Jan Kowalski".
- `uppercase` — zmienia wszystkie litery tekstu na wielkie, np. słowa „jan kowalski" zmieni na „JAN KOWALSKI".
- `lowercase` — zmienia wszystkie litery tekstu na małe, np. słowa „JAN KOWALSKI" zmieni na „jan kowalski".
- `none` — bez transformacji. Tekst jest wyświetlany na stronie tak, jak został umieszczony w kodzie.

Przykład 2.51

```
h3 { text-transform: capitalize; }
h2 { text-transform: uppercase; }
```

Białe znaki

Przy użyciu atrybutu `white-space` można sterować wyświetlaniem na stronie spacji oraz innych białych znaków. Wstawienie w kodzie kilku spacji obok siebie spowoduje

ich zamianę na pojedynczą spację. Również wstawienie spacji na początku tekstu jest niemożliwe. Atrybut `white-space` pozwoli pozbyć się tych niedogodności.

```
selektor { white-space: wartość; }
```

Atrybut może przyjmować następujące wartości:

- `normal` — zamienia kilka sąsiednich spacji na jedną. Wiersze tekstu są przełamywane automatycznie.
- `pre` — umożliwia wyświetlanie kilku spacji obok siebie. Wiersze tekstu są przełamywane w miejscu wystąpienia znaku nowej linii („enter").
- `nowrap` — zamienia kilka sąsiednich spacji na jedną. Wiersze tekstu są przełamywane w miejscu wystąpienia znacznika `<br>`.
- `pre-wrap` — umożliwia wyświetlanie kilku spacji obok siebie. Wiersze tekstu są przełamywane w miejscu wystąpienia znaku nowej linii („enter") oraz automatycznie na końcu szerokości strony.
- `pre-line` — zamienia kilka sąsiednich spacji na jedną. Wiersze tekstu są przełamywane w miejscu wystąpienia znaku nowej linii („enter") oraz automatycznie na końcu szerokości strony.

Nie wszystkie przeglądarki prawidłowo interpretują ten atrybut.

Przykład 2.52

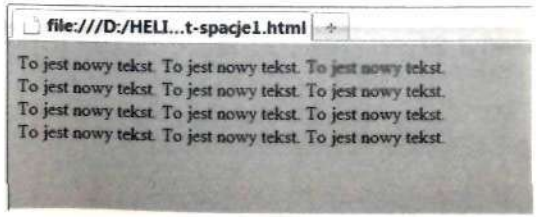
```
p { white-space: nowrap; }
```

Przykład 2.53

```
<html>
<head>
<style type="text/css">
body {background-color: #c0c0c0;}
p {white-space: pre-line; }
</style>
</head>
<body>
<p>To jest nowy tekst.  To jest nowy tekst.  To jest nowy tekst.
To jest nowy tekst.  To jest nowy tekst.  To jest nowy tekst.
To jest nowy tekst.  To jest nowy tekst.  To jest nowy tekst.
To jest nowy tekst.  To jest nowy tekst.  To jest nowy tekst.</p>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.18.

Rysunek 2.18. Tekst został przełamany w miejscu wystąpienia znaku nowej linii („enter"). Nie ma powtarzających się spacji



2.4.3. Tło i kolor

W języku HTML kolor tekstu oraz właściwości tła są definiowane w nagłówku. W języku CSS można definiować te właściwości dla różnych elementów.

Kolor
Korzystając z atrybutu `color`, można opisać pierwszoplanowy kolor wybranego elementu.

```
selektor { color: wartość; }
```

Przykład 2.54

```
h3 { color: #f0f0f0; }
```

Kolor tła
Kolor tła dla elementu jest definiowany za pomocą atrybutu `background-color`.

```
selektor { background-color: wartość; }
```

Przykład 2.55

```
p { background-color: yellow; }
<p style= "background-color: yellow";> Tło koloru żółtego</p>
h2 {color: green; background-color: #c0c0c0; }
```

Grafika jako tło
Do umieszczenia obrazka jako tła służy atrybut `background-image`.

```
selektor { background-image: wartość; }
```

Wartość atrybutu jest nazwą pliku, w którym została zapisana grafika.

Przykład 2.56

```
body { background-color: #f0f0f0; background-image: url (tlo.jpg); }
```

W podanym przykładzie obrazek został zlokalizowany w tym samym folderze co arkusz stylów. Gdy odwołujemy się do obrazka znajdującego się w innej lokalizacji, musimy podać pełną ścieżkę do pliku.

Powtarzanie grafiki w tle
Przy powtarzaniu obrazka w tle można określić sposób jego powtarzania, stosując atrybut `background-repeat`.

```
selektor { background-repeat: powtarzanie; }
```

Jako wartość powtarzania można wybrać:

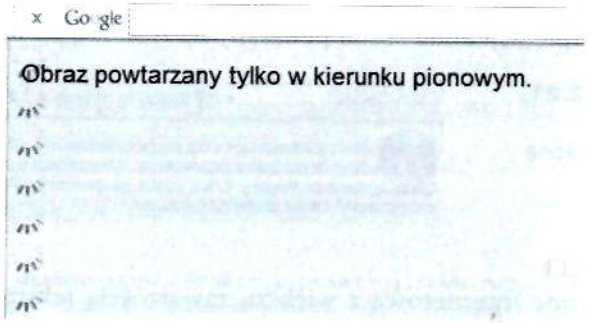
- `repeat` — powtarzanie tła w obu kierunkach;
- `repeat-x` — powtarzanie tła tylko w kierunku poziomym;
- `repeat-y` — powtarzanie tła tylko w kierunku pionowym;
- `no-repeat` — brak powtarzania tła.

Przykład 2.57

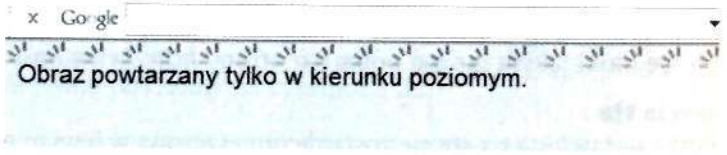
```
<html>
<head>
<style type="text/css">
body {background-color: #ffccff; background-image: url(obraz.gif);
background-repeat: repeat-y; }
p {font-size: 16pt; font-family: Arial;}
</style>
</head>
<body>
<p>Obraz powtarzany tylko w kierunku pionowym.</p>
</body>
</html>
```


Wynik powtarzania tła tylko w kierunku pionowym został pokazany na rysunku 2.19, natomiast wynik powtarzania tła tylko w kierunku poziomym — na rysunku 2.20.

Rysunek 2.19. Wstawiony obraz powtarzany w pionie



Rysunek 2.20. Wstawiony obraz powtarzany w poziomie



Blokada grafiki w tle
Jeżeli treść strony jest przewijana za pomocą suwaka, to wstawiona jako tło strony grafika przesuwa się razem z tekstem. Aby grafika w trakcie takich działań była nieruchoma, należy użyć atrybutu `background-attachment`.

```
selektor { background-attachment: blokada; }
```

Jako wartość blokady można wybrać:

- `scroll` — przewijanie tła;
- `fixed` — tło nieruchome względem okna przeglądarki;
- `local` — tło nieruchome względem elementu, dla którego zostało zdefiniowane.

Przykład 2.58

```
body {background-color: #ffccff; background-image: url(obraz.gif);
background-attachment: fixed; }
```

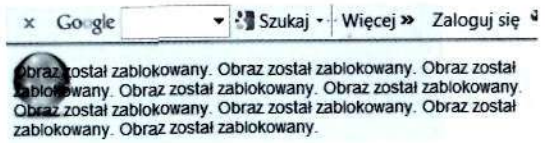
Przykład 2.59

```
<html>
<head>
```

```
<style type="text/css">
body {background-color: #fff000; background-image: url('obraz.png');
background-repeat: no-repeat; background-attachment: fixed;}
p {font-size: 12pt; font-family: Arial; white-space: pre-wrap; }
</style>
</head>
<body>
<p>Obraz został zablokowany. Obraz został zablokowany. Obraz został
zablokowany. Obraz został zablokowany. Obraz został zablokowany. Obraz
został zablokowany. Obraz został zablokowany. Obraz został zablokowany.
Obraz został zablokowany.</p>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.21.

Rysunek 2.21. Grafika tła unieruchomiona



Zadanie 2.11

Utwórz stronę internetową z większą zawartością tekstu. Wstaw grafikę jako tło strony lub jako tło umieszczonego na niej elementu. Sprawdź działanie atrybutu `background-attachment` dla wartości `scroll`, `fixed` i `local`.

Gotowe bloki tekstu można pobrać ze strony *http://www.lipsum.com/*.

Pozycja tła

Grafika wstawiona na stronę zostanie umieszczona w lewym górnym rogu ekranu. Do zmiany standardowych ustawień i pozycjonowania grafiki w dowolnym miejscu na stronie służy atrybut `background-position`.

```
selektor { background-position: pozycja; }
```

Wartość atrybutu może być ustalana w różny sposób. Można określać położenie grafiki na stronie, podając jedną wartość, można to zrobić za pomocą dwóch parametrów, można też zapisać jej położenie jako wartość procentową szerokości ekranu.

Niżej zostały podane wartości parametru `pozycja`.

Jedna wartość:

- `center` — grafika zostanie umieszczona na środku strony;
- `left` — grafika zostanie umieszczona z lewej strony;
- `right` — grafika zostanie umieszczona z prawej strony;
- `top` — grafika zostanie umieszczona na górze strony;
- `bottom` — grafika zostanie umieszczona na dole strony;
- `jednostka długości` — grafika zostanie umieszczona w określonej odległości od lewego marginesu.

Dwie wartości:

- `left top` — grafika zostanie umieszczona w lewym górnym rogu strony;
- `left bottom` — grafika zostanie umieszczona w lewym dolnym rogu strony;
- `right top` — grafika zostanie umieszczona w prawym górnym rogu strony;
- `right bottom` — grafika zostanie umieszczona w prawym dolnym rogu strony.

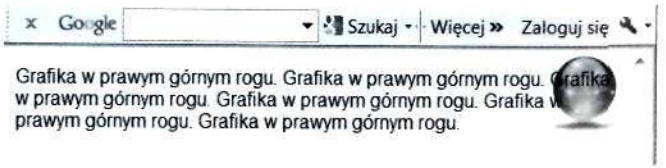
Definiowanie tych parametrów ma sens, jeżeli ograniczymy powtarzanie grafiki na stronie (mogą zostać w tym celu użyte parametry powtarzania: `no-repeat`, `repeat-x`, `repeat-y`).

Przykład 2.60

```
<html>
<head>
<style type="text/css">
body {background-color: #ffff00; background-image: url(obraz.png);
background-repeat: no-repeat; background-position: right top}
p {font-size: 12pt; font-family: Arial; }
</style>
</head>
<body>
<p>Grafika w prawym górnym rogu. Grafika w prawym górnym rogu. Grafika
w prawym górnym rogu. Grafika w prawym górnym rogu. Grafika w prawym górnym
rogu. Grafika w prawym górnym rogu. </p>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.22.

Rysunek 2.22. Grafika została umieszczona w prawym górnym rogu strony



Usunięcie tła

Jeżeli na stronie (lub na kilku stronach) nie chcemy stosować wcześniej zdefiniowanego tła dla stron lub elementów, to najprostszym rozwiązaniem jest niedołączanie do tych stron zewnętrznego arkusza stylów. Takie rozwiązanie nie zawsze jest jednak dobre, ponieważ inne deklaracje znajdujące się w zewnętrznym arkuszu stylów najczęściej są wykorzystywane na stronie. Lepszym rozwiązaniem jest zachowanie odnośnika do zewnętrznego arkusza stylów, a dodatkowo w wewnętrznym arkuszu stylów lub w stylu `inline` usunięcie tła dla danej strony. Do tego celu można użyć polecenia:

```
selektor { background: none; }
```

Polecenie to usuwa wszystkie atrybuty dotyczące tła.

Podobną metodą można usunąć także tylko wybrane atrybuty tła. Należy wtedy w poleceniu użyć nazwy atrybutu, który zostanie usunięty, np. `background-color: none`.

Atrybuty tła

Przy użyciu atrybutu `background` można w dogodny sposób zdefiniować wszystkie właściwości tła.

```
selektor { background: wartości atrybutów; }
```

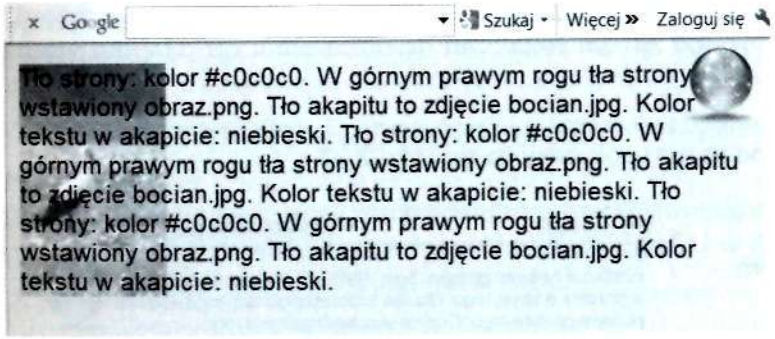
Jako wartości atrybutów trzeba podać konkretne wartości oddzielone spacjami. W ten sposób można zdefiniować następujące atrybuty tła:

- *kolor*,
- *tło obrazkowe*,
- *powtarzanie*,
- *zaczepienie*,
- *pozycja*.

Nie trzeba podawać pełnej listy atrybutów. Opuszczonemu atrybutowi zostanie przypisana wartość domyślna. Nie wszystkie przeglądarki akceptują tę skróconą formę deklaracji atrybutów tła i wtedy trzeba podawać ich pełną deklarację.

Zadanie 2.12

Zaprojektuj stronę podobną do przedstawionej na rysunku 2.23. Do projektowania strony wykorzystaj własną grafikę.

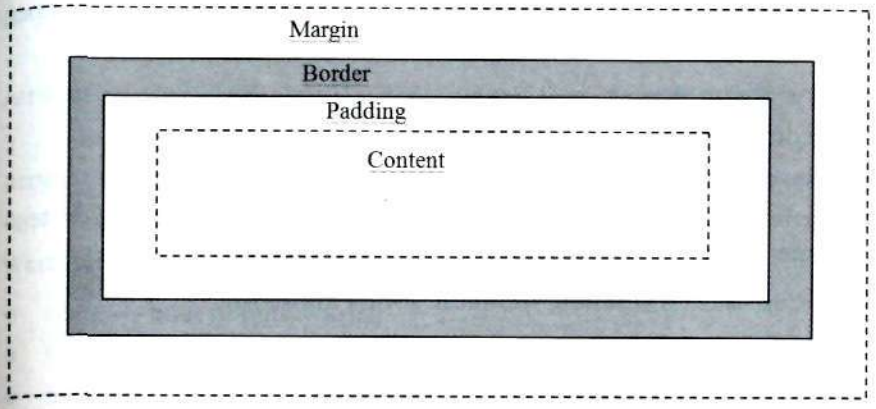


Rysunek 2.23. Przykład strony

2.5. Model blokowy CSS

Wszystkie elementy projektowane w języku HTML można traktować jak prostokątne bloki odpowiednio sformatowane i rozmieszczone na stronie internetowej. Każdy z takich bloków ma marginesy, obramowania, wypełnienie i zawartość.

Model blokowy został pokazany na rysunku 2.24.



Rysunek 2.24. Model blokowy CSS

- `Margin` — margines wokół ramki (margines zewnętrzny). Jest to pusty obszar wokół ramki, który nie ma koloru tła i jest przezroczysty.
- `Border` — obramowanie wokół zawartości elementu; ma styl i kolor.
- `Padding` — odstęp między obramowaniem i zawartością elementu (margines wewnętrzny).
- `Content` — zawartość elementu, np. teksty lub obrazy.

Każdy element w modelu blokowym jest opisywany przez trzy atrybuty: padding, margin i border. Podane atrybuty tworzą układ graficzny strony internetowej.

2.5.1. Obramowania

Obramowanie to ramka (ang. *border*) narysowana wokół elementu. Obramowanie może być wykorzystywane do dekoracji elementu lub do oddzielenia go od innych elementów. Można je definiować dla jednej lub dla wielu krawędzi bloku.

Szerokość obramowania

Do ustawienia szerokości obramowania służy atrybut `border-width`. Przy jego użyciu można definiować obramowanie jednakowe dla wszystkich krawędzi lub inne dla każdej z nich.

```
selektor { border-width: wartości ; }
```

Możliwe ustawienia atrybutu:

- *jedna wartość* — jednakowa szerokość dla wszystkich krawędzi;
- *dwie wartości* — taka sama szerokość dla krawędzi poziomych i taka sama szerokość dla krawędzi pionowych;
- *trzy wartości* — pierwsza wartość określa szerokość krawędzi górnej, druga — jednocześnie szerokość dwóch krawędzi pionowych, trzecia — szerokość krawędzi dolnej;
- *cztery wartości* — każda wartość określa osobno szerokość kolejnych krawędzi (górnej, prawej, dolnej, lewej).

Szerokość krawędzi może być definiowana w dowolnych jednostkach lub za pomocą trzech wartości:

- `thin` — cienkie obramowanie;
- `medium` — średnie obramowanie;
- `thick` — grube obramowanie.

Możliwe jest również definiowanie szerokości pojedynczych krawędzi. Służą do tego atrybuty przypisane do poszczególnych krawędzi.

Atrybut `border-top-width` definiuje szerokość górnej krawędzi.

```
selektor { border-top-width: szerokość ; }
```

Atrybut `border-bottom-width` definiuje szerokość dolnej krawędzi.

```
selektor { border-bottom-width: szerokość ; }
```

Atrybut `border-left-width` definiuje szerokość lewej krawędzi.

```
selektor { border-left-width: szerokość ; }
```

Atrybut `border-right-width` definiuje szerokość prawej krawędzi.

```
selektor { border-right-width: szerokość ; }
```

Przykład 2.61

```
p {border-width: 1px; }
h2 {border-bottom-width: 20px; }
h3 {border-width: thick; }
```

Styl obramowania

Do definiowania stylu obramowania służy atrybut `border-style`. Za jego pomocą można definiować styl obramowania jednakowy dla wszystkich krawędzi lub dla każdej inny.

```
selektor { border-style: wartości; }
```

Podobnie jak w przypadku atrybutu `border-width` możliwe są następujące ustawienia atrybutu:

- *jedna wartość* — jednakowy styl dla wszystkich krawędzi;
- *dwie wartości* — taki sam styl dla krawędzi poziomych i taki sam styl dla krawędzi pionowych;
- *trzy wartości* — *pierwsza wartość określa styl krawędzi górnej, druga* — jednocześnie styl dwóch krawędzi pionowych, trzecia — styl krawędzi dolnej;
- *cztery wartości* — każda wartość określa osobno styl kolejnych krawędzi (górnej, prawej, dolnej, lewej).

Możliwe jest, jak w przypadku atrybutu `border-width`, definiowanie stylu pojedynczych krawędzi. Służą do tego atrybuty przypisane do poszczególnych krawędzi.

Atrybut `border-top-style` definiuje styl górnej krawędzi.

```
selektor { border-top-style: styl; }
```


Atrybut `border-bottom-style` definiuje styl dolnej krawędzi.

```
selektor { border-bottom-style: styl; }
```

Atrybut `border-left-style` definiuje styl lewej krawędzi.

```
selektor { border-left-style: styl; }
```

Atrybut `border-right-style` definiuje styl prawej krawędzi.

```
selektor { border-right-style: styl; }
```

Wartości przypisywane do stylów obramowania to:

- `none` — *brak obramowania*,
- `hidden` — obramowanie ukryte,
- `dotted` — linia kropkowana,
- `dashed` — linia kreskowana,
- `solid` — linia ciągła,
- `double` — linia ciągła podwójna,
- `groove` — „rowek",
- `ridge` — „grzbiet",
- `inset` — „ramka",
- `outset` — „przycisk".



UWAGA

Przy projektowaniu ramek obowiązkowe jest podanie wartości dla stylu i dla szerokości.

Kolor obramowania

Do definiowania koloru obramowania służy atrybut `border-color`. Jako wartości koloru można używać kodów kolorów zapisanych heksadecymalnie lub ich nazw.

```
selektor { border-color: kolor; }
```

Podobnie jak w przypadku innych atrybutów obramowania możliwe jest podanie jednej, dwóch, trzech lub czterech wartości koloru.

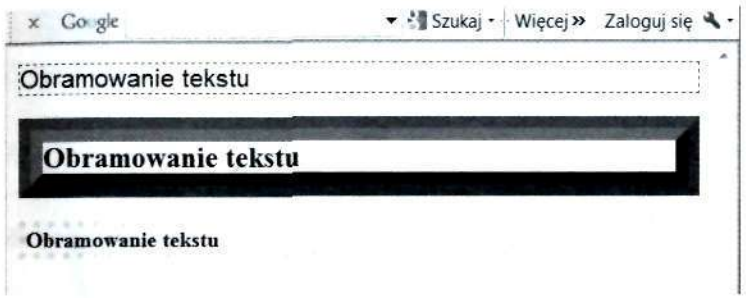
Możliwe jest też definiowanie koloru pojedynczych krawędzi. Służą do tego atrybuty: `border-top-color`, `border-bottom-color`, `border-left-color`, `border-right-color`.

Przykład 2.62

```
<html>
<head>
<style type="text/css">
body {background-color: #ffff00; }
p {font-size: 16pt; font-family: Arial; border-width: 1px; border-style:
dashed; border-color: blue; }
```

```
h2 {border-width: 20px; border-style: outset; border-color: red; }
h3 {border-width: thick; border-style: dotted; border-color: gold; }
</style>
</head>
<body>
<p>Obramowanie tekstu  </p>
<h2>Obramowanie tekstu  </h2>
<h3>Obramowanie tekstu  </h3>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.25.



Rysunek 2.25. Definiowanie koloru obramowania (border)

Atrybuty obramowania

Podobnie jak w przypadku tła, używając atrybutu `border`, można zdefiniować równocześnie kilka właściwości obramowania.

```
selektor { border: wartości atrybutów; }
```

Wartości atrybutów dotyczą szerokości (`width`), stylu (`style`) i koloru (`color`). Mogą być wpisane w dowolnej kolejności i muszą być oddzielone znakiem spacji.

Przykład 2.63

```
p { border: double 1px red; }
h2 {border: 20px outset blue; }
h3 {border: thick dotted gold; }
```

2.5.2. Marginesy zewnętrzne (margin)

Margines zewnętrzny określa przestrzeń wokół definiowanego elementu. Przestrzeń ta oddziela element od innych elementów strony.

Dla każdego elementu można zdefiniować jego odległość od góry, od dołu, od lewej i od prawej strony. Służy do tego atrybut `margin`.

```
selektor { margin-pozycja: rozmiar; }
```


Jako *pozycja* może wystąpić jedno z określeń dotyczących położenia marginesu:

- `top` — margines górny;
- `bottom` — margines dolny;
- `left` — margines lewy;
- `right` — margines prawy.

Jako *rozmiar* podawana jest wartość w jednostkach długości (np. `px`, `cm`). Można też wpisać jako rozmiar `auto`. Spowoduje to nadanie marginesowi przez przeglądarkę wartości automatycznej.

Przykład 2.64

```
p {margin-top: 1cm; }
h2 {margin-bottom: 2cm; }
h3 {margin-left: 20%; }
```

Za pomocą atrybutu `margin` można też ustawiać marginesy dla wszystkich czterech boków elementu.

```
selektor { margin: wartości; }
```

Jako wartości atrybutu można podać:

- *jedną wartość* — wszystkie marginesy jednakowe;
- *dwie wartości* — pierwsza oznacza margines górny i dolny, druga margines lewy i prawy;
- *trzy wartości* — pierwsza oznacza margines górny, druga margines lewy i prawy, trzecia margines dolny;
- *cztery wartości* — każda wartość określa osobno marginesy górny, prawy, dolny, lewy.

Gdy przy takim zapisie chcemy pominąć jeden z marginesów, wystarczy ustawić jego wartość na `0`.

Przykład 2.65

```
p {margin: 1cm; }
h2 {margin: 20px 10px; }
h3 {margin: 2cm 15px 1.5cm 12px; }
```

2.5.3. Marginesy wewnętrzne (padding)

Margines wewnętrzny to przestrzeń między zawartością elementu a obramowaniem.

Podobnie jak dla marginesów zewnętrznych, można zdefiniować tę odległość od góry, od dołu, od lewej i od prawej strony. Do definiowania marginesów wewnętrznych służy atrybut `padding`.

```
selektor { padding-pozycja: rozmiar; }
```

Jako *pozycja* mogą wystąpić określenia dotyczące położenia marginesów:

- `top` — margines górny;
- `bottom` — margines dolny;
- `left` — margines lewy;
- `right` — margines prawy.

Przykład 2.66

```
p {padding-top: 25px; }
h3 {padding-left: 2cm; }
```

Można również ustawiać marginesy wewnętrzne dla wszystkich czterech boków elementu na takich samych zasadach jak dla marginesu zewnętrznego.

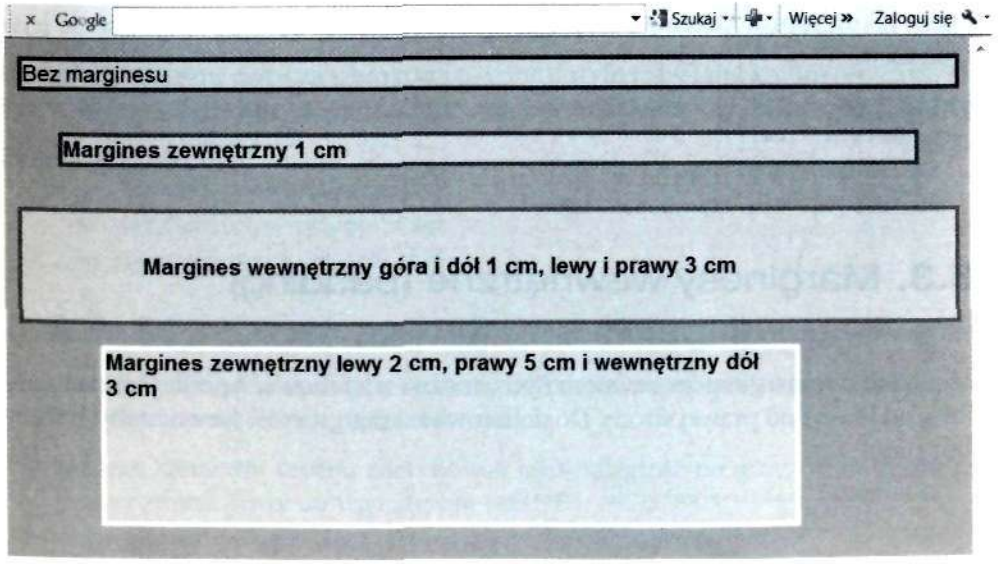
```
selektor { padding: wartości; }
```

Wartości atrybutu:

- *jedna wartość* — wszystkie marginesy jednakowe;
- *dwie wartości* — pierwsza oznacza margines górny i dolny, druga margines lewy i prawy;
- *trzy wartości* — pierwsza oznacza margines górny, druga margines lewy i prawy, trzecia margines dolny;
- *cztery wartości* — każda wartość określa osobno marginesy górny, prawy, dolny, lewy.

Aby pominąć jeden z marginesów wewnętrznych, należy wpisać wartość 0.

Różnice związane z definiowaniem marginesów zewnętrznych i wewnętrznych pokazuje rysunek 2.26.



Rysunek 2.26. Definiowanie marginesów wewnętrznych i zewnętrznych

Zadanie 2.13

Wykorzystując poznane narzędzia, zaprojektuj stronę podobną do tej na rysunku 2.26.

2.5.4. Obrys

Obramowanie wokół elementu może być tworzone również poprzez zdefiniowanie obrysu. Obrys różni się od obramowania tym, że:

- nie zajmuje miejsca, dzięki czemu nie wpływa na rozmiar i położenie elementu;
- jest tworzony na wierzchu elementu;
- jest definiowany dla wszystkich krawędzi równocześnie.

Obrys można definiować przy użyciu atrybutu `outline`.

```
selektor { outline: wartości; }
```

W ten sposób wybiera się kolor (ang. *color*), styl (ang. *style*) i grubość (ang. *width*) obrysu. Obowiązkowym atrybutem jest styl obrysu. Pozostałe dwa atrybuty występują opcjonalnie. Sposób definiowania obrysu jest taki sam jak przy definiowaniu obramowania.

Przykład 2.67

```
p { outline: solid 3px red; }
h2 { outline-style: dashed; }
h3 { outline-width: 10px; outline-color: blue; }
```

2.5.5. Rozmiary

W modelu blokowym można ustalać dokładne rozmiary różnych elementów, np. akapitu, tabeli, obrazka, bloku div.

Szerokość

Do definiowania szerokości elementu służy atrybut `width`.

```
selektor { width: szerokość; }
```

Przykład 2.68

```
h3 {width: 20%; }
```

Szerokość minimalna

Za pomocą atrybutu `min-width` można określić minimalną szerokość elementu. Pozwala to ustawić rozmiar elementu tak, aby nigdy nie był mniejszy od podanej wartości.

```
selektor { min-width: szerokość; }
```

Przykład 2.69

```
img {min-width: 200px; }
```

Szerokość maksymalna

Atrybut `max-width` pozwala określić maksymalną szerokość elementu. Element może być węższy od podanej wartości, ale nie może być od niej szerszy.

```
selektor { max-width: szerokość; }
```

Przykład 2.70

```
img {max-width: 300px; }
```

Wysokość

Atrybut `height` ustala wysokość elementu.

```
selektor { height: wysokość; }
```

Przykład 2.71

```
h3 {height: 3cm; }
```

Wysokość minimalna

Za pomocą atrybutu `min-height` można określić minimalną wysokość elementu. Oznacza to, że wysokość elementu nigdy nie będzie mniejsza od podanej wartości.

```
selektor { min-height: wysokość; }
```

Przykład 2.72

```
img {min-height: 300px; }
```

Wysokość maksymalna

Atrybut `max-height` pozwala określić maksymalną wysokość elementu. Element może być niższy od podanej wartości, ale nie może być od niej wyższy.

```
selektor { max-height: wysokość; }
```

Przykład 2.73

```
img {max-height: 5cm; }
```

Przykład 2.74

```
<html>
<head>
<style type="text/css">
img.rozmiar1 {height:auto; margin-right: 20px;}
img.rozmiar2 {height:120px; margin-right: 20px;}
p.blok {height:100px; width:130px;}
</style>
</head>
<body>
<img class="rozmiar1" src="bejr2.jpg" width="95" height="84" />
<img class="rozmiar2" src="bejr2.jpg" width="95" height="84" />
<img src="obraz.jpg" />
<p class="blok">Akapit o szerokości 130px i wysokości 100px. Akapit
o szerokości 130px i wysokości 100px. </p>
<p>Akapit bez określonej szerokości i wysokości. Akapit bez określonej
szerokości i wysokości.
 Akapit bez określonej szerokości i wysokości. Akapit bez określonej
szerokości i wysokości.
  Akapit bez określonej szerokości i wysokości. Akapit bez określonej
szerokości i wysokości.</p>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.27.



Rysunek 2.27. Określenie rozmiarów elementów strony

Przepełnienie

Jeżeli zawartość elementu nie mieści się w rozmiarach podanych za pomocą atrybutów `width` i `height`, możliwe jest ukrycie niemieszczącej się zawartości, powiększenie rozmiarów elementu i pokazanie całej zawartości lub wyświetlenie suwaków, które pozwolą (poprzez przewijanie) na obejrzenie całej zawartości elementu. Do określania sposobu pokazywania zawartości elementu w takiej sytuacji służy atrybut `overflow`.

```
selektor { overflow: sposób; }
```

Wartościami parametru *sposób* mogą być:

- `visible` — pokazywana jest cała zawartość elementu niezależnie od zdefiniowanego rozmiaru;
- `hidden` — niemieszcząca się zawartość zostaje ukryta;
- `scroll` — wyświetlane są suwaki, które pozwalają na obejrzenie całej zawartości elementu;
- `auto` — jeżeli jest to konieczne, suwaki zostają wyświetlone.

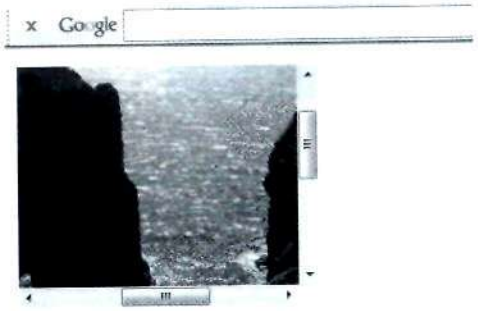
Przykład 2.75

```
<html>
<head>
<style type="text/css">
div {width: 250px; height: 200px; overflow: scroll;  }
</style>
</head>
<body>
<div>
<img src="zdj.jpg" />
</div>
```

```
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.28.

Rysunek 2.28.
Suwaki pozwalają obejrzeć całe zdjęcie



2.6. Inne elementy

2.6.1. Listy

W języku HTML można definiować listy punktowane (`<ul>`) i numerowane (`<ol>`). Jeżeli nie odpowiada nam sposób, w jaki są one wyświetlane przez przeglądarki, w języku CSS możemy zdefiniować inny sposób wyświetlania list.

Możliwe są: zmiana rodzaju punktora, zdefiniowanie własnego punktora oraz określenie położenia punktora i jego odległości od tekstu.

Styl listy
Większość przeglądarek w listach punktowanych wyświetla okrągłe punktory, a w listach numerowanych cyfry.

Za pomocą atrybutu `list-style-type` można zmienić punktor lub cyfrę wykazu na liście.

```
selektor { list-style-type: typ; }
```

Selektorami mogą być znaczniki definiujące listy: `<ul>`, `<ol>` oraz `<li>`.

Typ określa wygląd punktora (*markera*).

Dostępne typy punktorów to:

- `disc`,
- `circle`,
- `square`,
- `decimal`,
- `lower-roman`,

- `upper-roman`,
- `lower-alpha`,
- `upper-alpha`,
- `none`.

Nowsze wersje niektórych przeglądarek obsługują dodatkowe typy punktorów:

- `decimal-leading-zero`,
- `lower-greek`,
- `upper-greek`,
- `lower-latin`,
- `upper-latin`,
- `cjk-ideographic`,
- `hiragana`,
- `hiragana-iroha`,
- `kakakana-iroha`,
- `armenian`,
- `georgian`,
- `hebrew`.

Na rysunku 2.29 zostały pokazane różne typy punktorów.

Rysunek 2.29.
Definiowanie punktorów listy

Listy punktowane:

- Strony internetowe
- Bazy danych
- Aplikacje internetowe

- Strony internetowe
- Bazy danych
- Aplikacje internetowe

Listy numerowane:

I. Strony internetowe
II. Bazy danych
III. Aplikacje internetowe

a. Strony internetowe
b. Bazy danych
c. Aplikacje internetowe

Zadanie 2.14

Zaprojektuj na stronie internetowej listę punktowaną, listę numerowaną oraz listę z podpunktami. Zastosuj różne typy punktorów. Sprawdź ich wygląd. Stronę przetestuj w kilku przeglądarkach internetowych. Która z nich prawidłowo interpretuje wszystkie typy punktorów?

Obraz jako punktor

Punktorem listy może być obrazek zapisany w pliku. Do ustawienia obrazka jako markera służy atrybut `list-style-image`.

```
selektor { list-style-image: url(ścieżka_dostępu); }
```

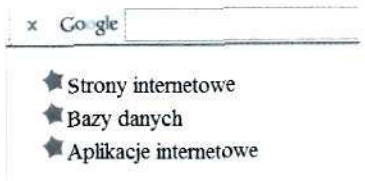
Ścieżka dostępu określa położenie pliku z obrazkiem, który ma być punktorem.

Przykład 2.76

```
<html>
<head>
<style type="text/css">
ul {list-style-image: url(punktor.png);font size: 14pt; }
</style>
</head>
<body>
<ul>
  <li>Strony internetowe</li>
  <li>Bazy danych</li>
  <li>Aplikacje internetowe</li>
</ul>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.30.

Rysunek 2.30. Obraz jako punktor



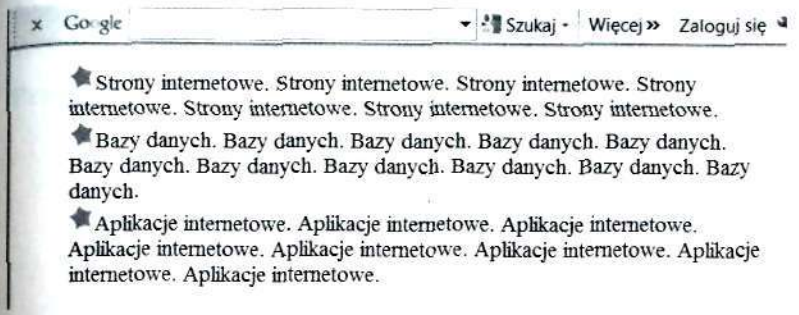
Pozycja punktora

Oprócz typu punktora można ustalić położenie punktora względem tekstu. Służy do tego atrybut `list-style-position`.

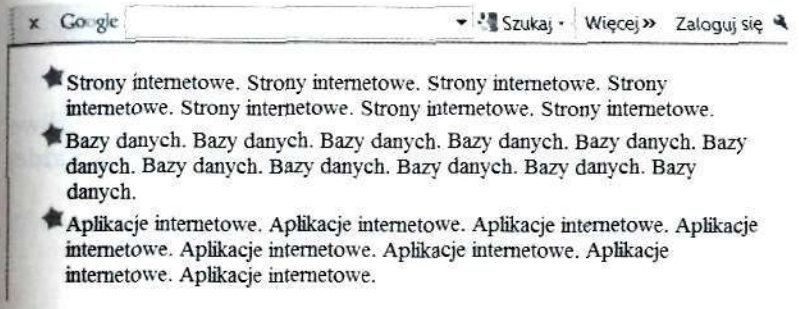
```
selektor { list-style-position: pozycja ;}
```

Pozycja określa zachowanie się tekstu, który nie zmieści się w jednej linii, względem punktora. Możliwe są dwie wersje:

- `inside` — punktor pojawia się wewnątrz zawartości (rysunek 2.31);
- `outside` — punktor pojawia się na zewnątrz zawartości (rysunek 2.32).



Rysunek 2.31. Punktor wewnątrz zawartości



Rysunek 2.32. Punktor na zewnątrz zawartości

Atrybuty listy

Używając atrybutu `list-style`, można równocześnie zdefiniować właściwości listy: `list-style-type`, `list-style-image` oraz `list-style-position`. Wartości atrybutów muszą być oddzielone spacjami.

```
selektor { list-style: wartości atrybutów; }
```

Przykład 2.77

```
<ul style="list-style: square inside;" >
 <li>Typ stylu</li>
 <li>Pozycja punktora</li>
 <li>Obrazek</li>
</ul>
```

2.6.2. Tabele

W języku HTML tabele są stosowane do przedstawiania danych. Czasami stosuje się je także do rozmieszczania elementów na stronie. Jeżeli chcemy za pomocą tabeli zaprezentować dane, to użycie stylów CSS pozwoli znacznie poprawić ich wygląd.

Podpis

W celu określenia położenia podpisu dodanego do tabeli używamy atrybutu `caption-side`.

```
selektor { caption-side: ustawienie; }
```

Parametr `ustawienie` może przyjmować następujące wartości:

- `top` — podpis zostanie wyświetlony nad tabelą;
- `bottom` — podpis zostanie wyświetlony pod tabelą;
- `left` — podpis zostanie wyświetlony z lewej strony tabeli;
- `right` — podpis zostanie wyświetlony z prawej strony tabeli.

Nie wszystkie przeglądarki prawidłowo interpretują ten atrybut.

Obramowanie

Do definiowania krawędzi tabeli używa się atrybutu `border`.

```
selektor { border: wartości; }
```

Dla tego atrybutu można ustawić grubość, rodzaj oraz kolor obramowania.

Przykład 2.78

```
table, tr, td {border: 2px solid red; }
```

Atrybut `border-collapse` określa sposób obramowania komórek tabeli. Możliwe jest obramowanie sąsiednich komórek za pomocą jednej linii lub obramowanie każdej komórki osobno.

```
selektor { border-collaspe: model; }
```

Wartości parametru `model` to:

- `collapse` — pojedyncze obramowanie;
- `separate` — komórki tabeli będą od siebie odseparowane.

Przykład 2.79

```
table {border-collaspe: separate; }
```

Można także określić sposób obramowania dla pustych komórek. Służy do tego atrybut `empty-cells`.

```
selektor { empty-cells: wartość; }
```

Atrybut może przyjmować wartości:

- `show` — obramowanie wokół pustych komórek powinno być wyświetlane;
- `hide` — obramowanie wokół pustych komórek powinno zostać ukryte.

Do tabel oraz do ich komórek można stosować wszystkie atrybuty definiowane dla obramowań w modelu blokowym.

Rozmieszczenie

W celu rozplanowania tabeli na stronie można użyć atrybutu `table-layout`.

```
selektor { table-layout: rozmieszczenie; }
```

Atrybut może przyjmować wartości:

- `auto` — szerokość tabeli jest ustawiana automatycznie i zależy od zawartości komórek;
- `fixed` — szerokość tabeli wynika z zadeklarowanej szerokości poszczególnych komórek oraz szerokości całej tabeli.

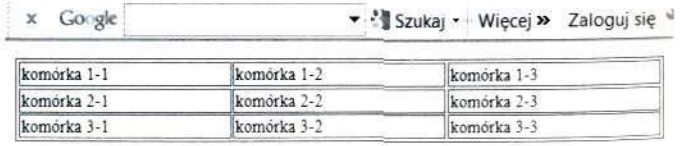
Różne przeglądarki odmiennie interpretują to polecenie, jeśli nie zostanie podana ogólna szerokość tabeli lub jeśli szerokość tabeli jest większa niż suma szerokości jej komórek.

Przykład 2.80

```
<html>
<head>
<style type="text/css">
table, tr, td {border: 1px solid blue; border-collaspe: collaspe; table-
layout: fixed; }
</style>
</head>
<body>
<table>
<tr>
<td>komórka 1-1</td>
<td>komórka 1-2</td>
<td>komórka 1-3</td>
</tr>
<tr>
<td>komórka 2-1</td>
<td>komórka 2-2</td>
<td>komórka 2-3</td>
</tr>
<tr>
<td>komórka 3-1</td>
<td>komórka 3-2</td>
<td>komórka 3-3</td>
</tr>
</table>
</body>
</html>
```

Wynik interpretacji kodu w IE został pokazany na rysunku 2.33.

Rysunek 2.33. Wyświetlenie tabeli w IE



Wynik interpretacji kodu w Google Chrome został pokazany na rysunku 2.34.

Rysunek 2.34. Wyświetlenie tabeli w Google Chrome



Odstępy między komórkami

Do definiowania odstępów między komórkami służy atrybut `border-spacing`.

```
selektor { border-spacing: odstęp; }
```

Odstęp to zapisana w jednostkach długości odległość między komórkami tabeli. Można podać jedną wartość (i wtedy dotyczy ona wszystkich odstępów) lub dwie wartości rozdzielone spacją (i wtedy definiują one poziomy oraz pionowy odstęp między komórkami).

2.6.3. Grupowanie selektorów

Jeżeli w arkuszu stylów zostały zdefiniowane elementy o tym samym stylu, to można te elementy połączyć w grupy i nadać im styl poprzez wspólną deklarację.

Przykład 2.81

```
p {color: green; }
h2 {color: green; }
h3 {color: green; }
```

Ten ciąg deklaracji można zastąpić jedną wspólną definicją stylu.

Przykład 2.82

```
p, h2, h3 {color: green; }
```

Kolejne elementy przypisane do stylu powinny zostać oddzielone przecinkami.

2.6.4. Pozycjonowanie

Pozycjonowanie pozwala zdefiniować położenie elementów na stronie internetowej. Elementy można rozmieszczać nie tylko względem brzegów strony, ale również względem jej poszczególnych elementów. Można również umieszczać elementy tak, że jeden będzie przykrywał inny.

Do pozycjonowania elementów służy atrybut `position`.

```
selektor { position: rodzaj; parametry; }
```

Rodzaj określa sposób pozycjonowania elementu i może przyjmować następujące wartości:

- `static` — przywraca normalne pozycjonowanie elementu. Jest to wartość przydatna wtedy, gdy wcześniej (np. w arkuszach stylów) została podana deklaracja pozycjonowania tego typu elementów. Użycie w definicji stylu `inline` dla wybranego elementu deklaracji `position: static` zniesie tę deklarację dla danego elementu.

- `relative` (inaczej: pozycjonowanie względne) — pozwala przesunąć wybrany element w inne miejsce w stosunku do położenia pierwotnego. Poprzez ustawienie parametrów można określać sposób przesunięcia.
- `absolute` — pozwala przesunąć wybrany element w inne miejsce względem brzegów strony, bloku lub ramki (jeżeli element został umieszczony w bloku lub w ramce). Poprzez ustawienie parametrów można określić sposób przesunięcia. Element zostanie umieszczony „na sztywno" w miejscu określonym przez parametry.
- `fixed` — działa podobnie do pozycjonowania absolutnego. Różnica polega na tym, że pozycjonowanie `fixed` ustala położenie elementu zawsze względem krawędzi okna przeglądarki. Tak pozycjonowany element jest nieruchomy przy przewijaniu strony (jest przez cały czas widoczny w tym samym miejscu).

Jako *parametr* należy podawać:

- `left:` *wartość* — przesunięcie o określoną wartość w stosunku do lewej krawędzi położenia pierwotnego;
- `top:` *wartość* — przesunięcie o określoną wartość w stosunku do górnej krawędzi położenia pierwotnego;
- `right:` *wartość* — przesunięcie o określoną wartość w stosunku do prawej krawędzi położenia pierwotnego;
- `bottom:` *wartość* — przesunięcie o określoną wartość w stosunku do dolnej krawędzi położenia pierwotnego.

Można łączyć ze sobą różne parametry. Trzeba jednak pamiętać o tym, że parametr `left` ma pierwszeństwo przed `right`, a parametr `top` ma pierwszeństwo przed `bottom`.

Wartość parametru *parametr* oznacza określoną odległość od podanej krawędzi; odległość ta powinna być podawana w jednostkach długości. Wpisanie wartości ujemnej spowoduje przesunięcie w stronę przeciwną. Można użyć słowa `auto`, aby ustawić wartość domyślną.

Przykład 2.83

```
h1 { position: absolute; top: 100px; left: 200px; }
img { position: relative; left: 350px; bottom: 150px; }
div {position: fixed; top: 20px; right: 10px; border: dotted 1px; padding:
5px; background-color: yellow; }
```

Przezroczystość

Przy użyciu atrybutu `opacity` można zdefiniować przezroczystość dla elementów wstawianych na stronę.

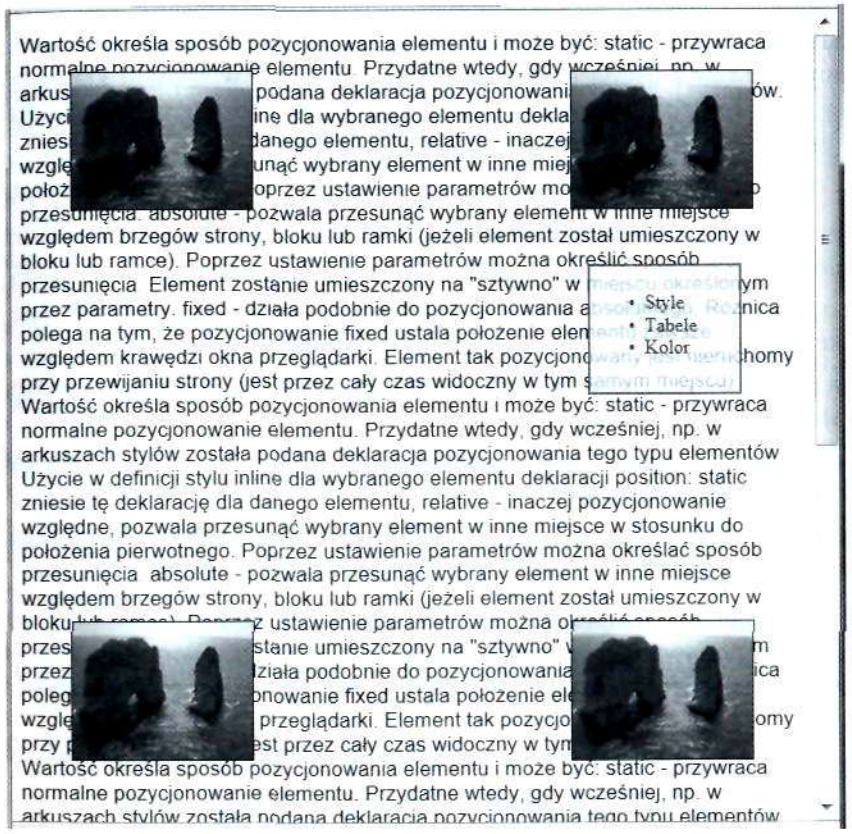
```
selektor { opacity: nieprzezroczystość; }
```

Nieprzezroczystość to liczba z przedziału od `0,0` do `1,0`. `0` oznacza całkowitą przezroczystość, `1` — brak przezroczystości.

Zdefiniowany poziom przezroczystości jest nadawany wszystkim elementom znajdującym się wewnątrz określonego elementu. Przezroczystości elementów wewnętrznych nie można zmniejszyć (np. nie mogą one stać się zupełnie nieprzezroczyste), ale można im nadać większą przezroczystość w stosunku do elementu zewnętrznego.

Zadanie 2.15

Wykorzystując poznane narzędzia do pozycjonowania elementów, zaprojektuj stronę, w której rogach będą rozmieszczone cztery zdjęcia. Lista z prawej strony będzie elementem nieruchomym w czasie przewijania zawartości witryny (rysunek 2.35). Jeżeli potrzebujesz większej ilości przykładowego tekstu do wstawienia, możesz wykorzystać stronę *http://www.lipsum.com/*. Wstawione elementy odpowiednio sformatuj.

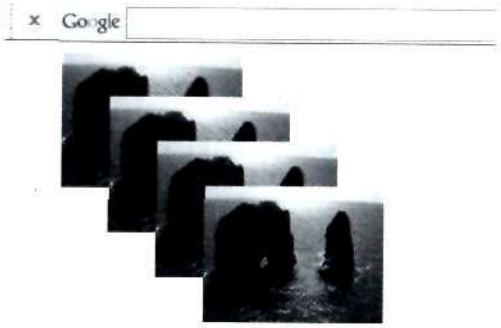


Rysunek 2.35. Strona zawiera pozycjonowanie obrazów

Zadanie 2.16

Wykorzystując poznane narzędzia do pozycjonowania elementów, ułóż zdjęcia na stronie podobnie jak na rysunku 2.36.

Rysunek 2.36. Przykładowe położenie elementów na stronie uzyskane w wyniku pozycjonowania



Nakładanie elementów (warstwy)

Gdy kilka elementów nachodzi na siebie, to można określić, w jaki sposób będą one nakładane. Służy do tego atrybut `position`. Nakładającym się elementom można nadać numer (`z-index`) i zastosować regułę, że elementy z wyższym numerem będą nakładane na te z numerem niższym (rysunek 2.37).

```
selektor { position: rodzaj; parametry; z-index: numer; }
```

Rysunek 2.37. Zmiana kolejności elementów



Wyrażenie `position: rodzaj; parametry;` określa parametry pozycjonowania.

Numer mówi o kolejności nakładania pozycjonowanych elementów. Element, któremu nadamy numer `1`, znajdzie się na samym spodzie i będzie zakryty przez inne elementy.

Przykład 2.84

```
#blok3 {position: relative; left: 120px; top: -160px; z-index: 4; }
```

Poprawnie interpretowane są również wartości ujemne.

Obcinanie elementu

Do wycięcia z elementu jego fragmentu w kształcie prostokąta służy atrybut `position: absolute; clip: rect;` z odpowiednimi parametrami.

```
selektor { position: absolute; clip: rect(góra, prawo, dół, lewo ); }
```

Wartości: *góra*, *prawo*, *dół*, *lewo*, oznaczają współrzędne wyciętego prostokąta. Wartości te są określane względem lewego górnego rogu elementu. Powinny być podawane w jednostkach długości. Można używać określenia `auto`, co oznacza pominięcie cięcia z danej strony.

Przykład 2.85

```
<html>
<head>
<style type="text/css">
#blok {position: absolute; clip: rect(0px, 100px, 110px, 20px); }
</style>
</head>
<body>
<img src="bejr2.jpg" >
<div id="blok"><img src="bejr2.jpg"> </div>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.38.

Rysunek 2.38. Obcinanie elementu



Zadanie 2.17

Wykorzystując style CSS i poznane narzędzia, zaprojektuj stronę internetową informującą o zbliżającym się ważnym wydarzeniu sportowym.

Wyrównanie w pionie

Atrybut `vertical-align` pozwala na zdefiniowanie wyrównania elementu w stosunku do innych elementów strony.

```
selektor { vertical-align: sposób; }
```

Jako wartość parametru *sposób* należy podać:

- `baseline` — wyrównuje linię bazową elementu do linii bazowej elementu nadrzędnego;
- `middle` — ustawia element na środku wysokości elementów sąsiednich;
- `text-top` — wyrównuje górną krawędź elementu do górnej krawędzi tekstu elementu nadrzędnego;
- `text-bottom` — wyrównuje podstawę elementu do podstawy tekstu elementu nadrzędnego;
- `super` — ustawia element jako indeks górny;
- `sub` — ustawia element jako indeks dolny;

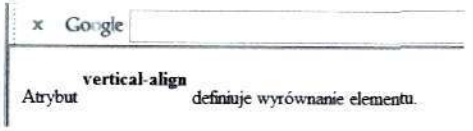
- `top` — wyrównuje górną krawędź elementu do górnej krawędzi najwyższego elementu w linii;
- `bottom` — wyrównuje podstawę elementu do podstawy elementu położonego najniżej w linii.

Przykład 2.86

```
<html>
<head>
<style type="text/css">
span {vertical-align: 4mm; }
</style>
</head>
<body>
<p> Atrybut <span><b>vertical-align</b></span> definiuje wyrównanie
elementu. </p>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.39.

Rysunek 2.39. Wyrównanie elementu w stosunku do innych elementów



Ustawienie w poziomie

Do ustawiania elementu względem elementów, które z nim sąsiadują w poziomie, służy atrybut `float`. Potocznie mówi się, że element „pływa" po lewej lub prawej stronie okna przeglądarki lub bloku.

```
selektor { float: sposób; }
```

Parametr *sposób* może przyjmować następujące wartości:

- `left` — element zostanie ustawiony po lewej stronie sąsiednich elementów;
- `right` — element zostanie ustawiony po prawej stronie sąsiednich elementów;
- `none` — element nie będzie sąsiadował z innymi elementami.

Najczęściej atrybut ten jest wykorzystywany do określenia sposobu oblewania tekstem elementu pływającego (`float`).

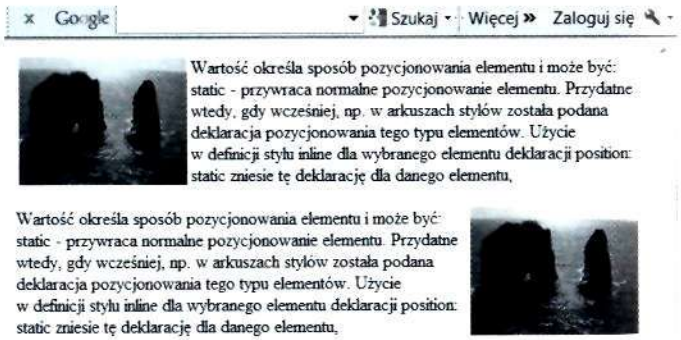
Przykład 2.87

```
<html>
<head>
<style type="text/css">
#blok1 {float:left; }
#blok2 {float:right; }
</style>
</head>
```

```
<body>
<img id="blok1" src="bejr.jpg" >
<p>Wartość określa sposób pozycjonowania elementu i może być:
  static - przywraca normalne pozycjonowanie elementu. Przydatne wtedy,
gdy wcześniej, np. w arkuszach stylów, została podana deklaracja
pozycjonowania tego typu elementów. Użycie w definicji stylu inline
dla wybranego elementu deklaracji position: static zniesie tę deklarację
dla danego elementu
</p>
<img id="blok2" src="bejr.jpg" >
<p>Wartość określa sposób pozycjonowania elementu i może być:
  static - przywraca normalne pozycjonowanie elementu. Przydatne wtedy,
gdy wcześniej, np. w arkuszach stylów, została podana deklaracja
pozycjonowania tego typu elementów. Użycie w definicji stylu inline
dla wybranego elementu deklaracji position: static zniesie tę deklarację
dla danego elementu
  </p>
</body>
</html>
```

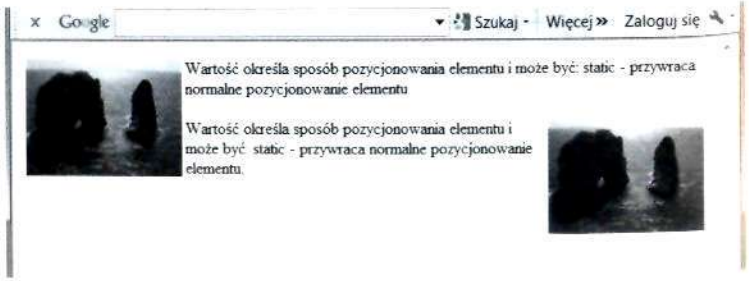
Wynik interpretacji kodu został pokazany na rysunku 2.40.

Rysunek 2.40. Zastosowanie atrybutu float do ustawienia tekstu wokół zdjęcia



Jeżeli tekst umieszczony z prawej lub lewej strony zdjęcia będzie zbyt krótki, to zdjęcia nie zostaną ułożone jedno pod drugim, tak jak na rysunku wyżej, lecz znajdą się obok siebie (rysunek 2.41). Dodatkowo ułożenie zdjęć będzie zależało od rozdzielczości ekranu i rozmiaru okna przeglądarki.

Rysunek 2.41. Błędy przy rozmieszczaniu tekstu wokół zdjęcia



Aby temu zapobiec, należy użyć atrybutu `clear`, który ustala pozycję kolejnych elementów względem elementu pływającego.

```
selektor { clear: sposób; }
```

Wartość atrybutu `clear` określa sposób zachowania się kolejnych elementów względem elementu pływającego:

- `left` — lewy bok następnego elementu nie przylega do elementu pływającego;
- `right` — prawy bok następnego elementu nie przylega do elementu pływającego;
- `both` — żaden bok następnego elementu nie przylega do elementu pływającego;
- `none` — brak ograniczeń.

Zadanie 2.18

Popraw podany w przykładzie kod tak, aby zdjęcie drugie było wyświetlane poniżej zdjęcia pierwszego niezależnie od długości tekstu występującego obok zdjęcia oraz niezależnie od wielkości okna przeglądarki i rozdzielczości ekranu.

Kolumny na stronie

Atrybut `float` może zostać wykorzystany do tworzenia kolumn w dokumencie. W tym celu musimy zawartość każdej kolumny zawrzeć w znaczniku `<div>`. Następnie, np. przy trzech kolumnach, należy określić szerokość każdego znacznika `<div>` na 33%. Po zdefiniowaniu dla każdego znacznika `<div>` atrybutu `float: left` uzyskamy „pływanie" kolejnych kolumn z lewej strony.

Na rysunku 2.42 został podany przykład dokumentu z trzema kolumnami.



Rysunek 2.42. Dokument z trzema kolumnami

Zadanie 2.19

Utwórz na stronie galerię zdjęć na wybrany temat. Rozplanuj położenie obrazków tak, aby obok siebie wyświetlane były po trzy zdjęcia. Układ zdjęć nie powinien zostać zaburzony przy zmianie wielkości okna przeglądarki. Określ minimalny rozmiar każdego zdjęcia.

Zadanie 2.20
Zaprojektuj własną stronę internetową, na której zostanie zastosowany układ kolumnowy. Informacje w zależności od treści rozplanuj w odpowiednich kolumnach.

2.6.5. Wyświetlanie

Wszystkie definiowane elementy są wyświetlane na stronie w sposób domyślny. Jeżeli chcemy zrezygnować z takiego pokazywania elementu, możemy zdefiniować atrybut `display`. Za jego pomocą można określić, jak dany element będzie prezentowany.

```
selektor { display: sposób; }
```

Parametr atrybutu określa *sposób* pokazywania elementu na stronie i może przyjmować następujące wartości:

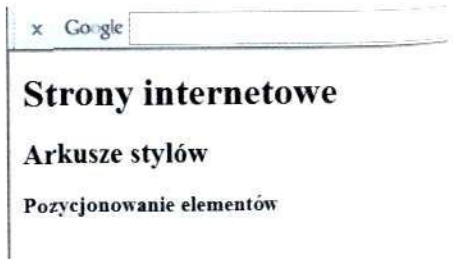
- `block` — element wyświetlany w bloku z odstępami od góry i od dołu;
- `inline` — element wyświetlany w linii z innymi elementami;
- `list-item` — element wyświetlany jako pozycja listy `<li>`;
- `none` — element nie będzie wyświetlany;
- `inline-block` — element wyświetlany podobnie jak element zastępowany; mimo że element ma charakter blokowy (odstęp, obramowanie, np. `div`), jest wyświetlany jako liniowy;
- `run-in` — jeżeli po elemencie występuje element będący blokiem, to ten pierwszy element zostanie wyświetlony w jednej linii z blokiem; w przeciwnym razie element zostanie wyświetlony w bloku.

UWAGA

Element zastępowany to element, dla którego istnieje tylko wymiar wewnętrzny, np.: `img` (zawartość elementu jest zastępowana obrazem określonym przez atrybut `src`), `input`, `textarea`, `select`, `object`.

Atrybut `display` jest bardzo przydatny, gdy chcemy zrezygnować z domyślnego sposobu wyświetlania. Jeżeli na przykład element jest wyświetlany w bloku (czyli dodawane są linie odstępu między sąsiednimi elementami), to po zastosowaniu `display: inline` elementy zostaną ustawione w jednej linii.

Domyślne wyświetlanie elementów zostało pokazane na rysunku 2.43.

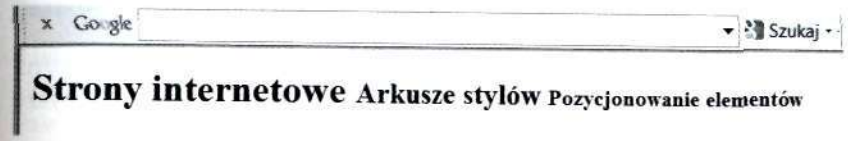


Rysunek 2.43. Domyślne wyświetlanie elementów

Przykład 2.88 (zastosowanie display: inline)

```
<html>
<head>
<style type="text/css">
.blok {display: inline;}
</style>
</head>
<body>
<h1 class="blok"> Strony internetowe </h1>
<h2 class="blok">Arkusze stylów </h2>
<h3 class="blok">Pozycjonowanie elementów </h3>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.44.

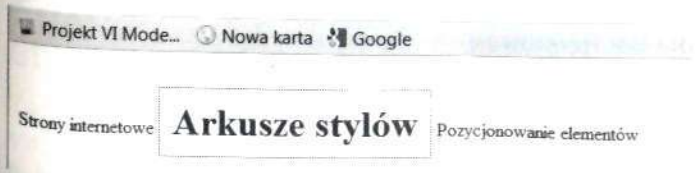


Rysunek 2.44. Zastosowanie atrybutu display: inline

Przykład 2.89 (zastosowanie display: inline-block)

```
<html>
<head>
<style type="text/css">
.blok {display: inline-block; border: dotted 1px; padding: 10px;}
</style>
</head>
<body>
<div> Strony internetowe
<h1 class="blok">Arkusze stylów </h1>
Pozycjonowanie elementów
</div>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.45.



Rysunek 2.45. Zastosowanie atrybutu display: inline-block

Ukrywanie elementów

Do sterowania wyświetlaniem elementów na stronie służy atrybut `visibility`.

```
selektor { visibility: typ; }
```

Jako wartość parametru *typ* należy podać jedną z poniższych wartości.

- `visible` — element będzie widoczny na ekranie.
- `hidden` — element będzie ukryty, niewidoczny na ekranie. W jego miejscu pojawi się pusta przestrzeń.
- `collapse` — zastosowana dla tabeli ukrywa całą zawartość wiersza lub kolumny. Dla innych elementów działa jak `hidden`.

Podobny efekt jak w przypadku `visibility: hidden` można uzyskać, stosując atrybut `display: none`. Różnica polega na tym, że jeśli użyjemy `display: none`, element zostanie całkowicie usunięty ze strony.

2.6.6. Kursory

W zależności od działania, jakie będzie realizowane na stronie internetowej, można zdefiniować odpowiadający temu działaniu kształt kursora myszy. Po ustawieniu myszy na danym elemencie nastąpi zmiana jej kursora.

Do definiowania kształtu kursora myszy służy atrybut `cursor`.

```
selektor { cursor: rodzaj; }
```

Jako wartość parametru *rodzaj* można podać:

- `auto` — automatyczny wybór kursora myszy;
- `crosshair` — krzyżyk;
- `default` — strzałka (kursor domyślny);
- `pointer` — wskaźnik (dłoń z palcem wskazującym);
- `move` — kursor przesunięcia (strzałka czterostronna);
- `e-resize` — strzałka prawo-lewo (pozioma);
- `ne-resize` — strzałka lewo-prawo (ukośna);
- `nw-resize` — strzałka prawo-lewo (ukośna);
- `n-resize` — strzałka góra-dół (pionowa);
- `se-resize` — strzałka prawo-lewo (ukośna);
- `sw-resize` — strzałka lewo-prawo (ukośna);
- `s-resize` — strzałka góra-dół (pionowa);
- `w-resize` — strzałka prawo-lewo (pozioma);
- `text` — kursor tekstowy (pionowa kreska);
- `wait` — kursor „czekaj" (klepsydra);
- `help` — kursor pomocy (znak zapytania).

Kształt kursora może być wczytywany z pliku. Należy wtedy podać ścieżkę dostępu do pliku.

```
selektor { cursor: url(ścieżka dostępu), rodzaj; }
```

Na końcu deklaracji trzeba podać jeden ze standardowych rodzajów kursora. Jeżeli pominiemy ten parametr, a przeglądarka nie może obsłużyć kursora z pliku, żaden kursor nie zostanie wyświetlony.

Pliki z kursorem najczęściej mają rozszerzenie *.cur*, a kursory animowane — *.ani*.

2.7. Menu w języku CSS

Zaprojektowane na stronie internetowej menu umożliwia nawigację w jej obrębie. Paski nawigacyjne buduje się najczęściej na podstawie standardowej listy nieuporządkowanej. Lista jest zbiorem występujących po sobie pozycji; podobnie budowane jest menu. Za pomocą języka CSS można utworzoną listę zamienić w atrakcyjne menu w postaci pasków nawigacyjnych.

Lista menu

Budowanie menu zaczynamy od utworzenia listy w języku HTML. Lista taka powinna zawierać linki do odpowiednich miejsc wywoływanych przez menu.

Przykład 2.90

```
<ul>
 <li><a href="start.html">Start</a></li>
 <li><a href="nowe.html">Aktualności</a></li>
 <li><a href="szkolenia.html">Szkolenia</a></li>
 <li><a href="uslugi.html">Usługi</a></li>
</ul>
```

Po zdefiniowaniu zawartości menu można przystąpić do jego formatowania z użyciem stylów CSS. Dołączenie deklaracji stylów pozwoli na stworzenie elastycznego i funkcjonalnego menu nawigacyjnego. Sam wygląd menu zostanie zmieniony tylko przez dołączenie arkusza stylów z odpowiednimi deklaracjami.

Dobrym rozwiązaniem jest zdefiniowanie dla znacznika `<ul>` identyfikatora lub klasy, aby w wyniku formatowania tworzonego menu nie doprowadzić do przeformatowania zwykłych list, które mogą się znaleźć na stronie. Pozwoli to również na umieszczanie na stronie kilku różnie wyglądających zestawów menu.

2.7.1. Menu pionowe — deklaracje w arkuszu stylów

Zanim przystąpimy do definiowania menu, z przygotowanej listy trzeba usunąć punktory oraz wyzerować jej marginesy. Należy również zadeklarować wyświetlanie listy w bloku oraz określić szerokość tego bloku.

Przykład 2.91

```
ul, li { display: block; list-style: none; margin: 0; padding: 0; }
ul {width: 180px; }
```

Tak zapisana deklaracja w arkuszu stylów jest podstawą do tworzenia pionowych pasków nawigacyjnych.

Następne deklaracje dotyczą znacznika <a> występującego w zdefiniowanej liście oraz znacznika <ul>.

Przykład 2.92

```
ul a:link, ul a:visited { display: block; width: 168px;
  text-decoration: none;
  background-color: #c0c0c0;
  color: black;
  padding: 4px;
  border: 2px outset #c0c0c0; }

ul a:hover { border-style: inset;
  color: blue;
  padding: 6px 3px 3px 6px; }
```

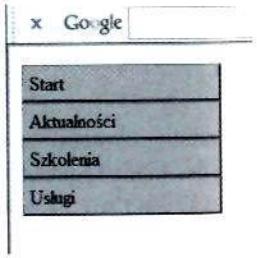
W tej części definicji stylów został zdefiniowany wygląd przycisków menu.

Każdy z elementów listy zostanie wyświetlony w bloku, w związku z czym nie tylko tekst, ale cały blok stanie się odnośnikiem.

Deklaracje `a:link` oraz `a:visited` określają wygląd przycisków przed odwiedzeniem strony lub po jej odwiedzeniu (zniesione podkreślenie, ustalony kolor tła i kolor tekstu, margines wewnętrzny `5px` i ramka `2px` z ustawionym stylem ramki).

Deklaracja `a:hover` określa wygląd przycisku, gdy znajdzie się nad nim kursor myszy (zmiana stylu obramowania oraz przesunięcie marginesu wewnętrznego).

Przykładowe menu pionowe zostało pokazane na rysunku 2.46.



Rysunek 2.46. Menu pionowe

Zadanie 2.21
Zmieniając deklaracje stylów w arkuszu stylów, zaprojektuj na stronie internetowej własne menu pionowe.

2.7.2. Menu poziome — deklaracje w arkuszu stylów

Przy tworzeniu menu poziomego również korzystamy ze zdefiniowanej wcześniej listy nieuporządkowanej.

Domyślnie lista jest tworzona w układzie pionowym. Ułożenie pozycji listy obok siebie można uzyskać na dwa sposoby:

- wyświetlając kolejne elementy listy w linii (atrybut `display: inline`);
- ustawiając pływanie dla każdej pozycji listy (atrybut `float`).

Podobnie jak w przypadku menu pionowego, z przygotowanego menu należy usunąć punktory oraz wyzerować jego marginesy.

Przykład 2.93

```
ul, li { display: block; list-style: none; margin: 0; padding: 0; }
```

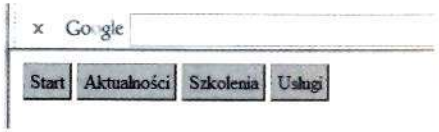
Wyświetlanie kolejnych elementów listy w linii
Przykład 2.94

```
ul li { display: inline;
white-space: nowrap; }

ul a:link, ul a:visited {
  text-decoration: none;
  background-color: #c0c0c0;
  color: black;
  padding: 4px;
  border: 2px outset #c0c0c0; }

ul a:hover { border-style: inset; color: blue;
  padding: 6px 3px 3px 6px; }
```

Dodane polecenie `white-space: nowrap;` blokuje zawijanie tekstu w każdej pozycji menu (rysunek 2.47). Jeśli wszystkie pozycje menu nie zmieszczą się w jednej linii, podział nastąpi między elementami listy, a nie wewnątrz elementu.



Rysunek 2.47. Menu poziome — kolejne elementy listy w linii

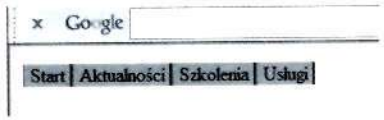
Metoda ta daje poprawne efekty w większości przeglądarek i nie wpływa na elementy znajdujące się poniżej menu. Nie można jednak za jej pomocą zdefiniować jednakowej szerokości pozycji menu.

Float dla każdej pozycji listy
Przykład 2.95

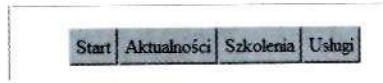
```
ul, li { float: left; }
ul a:link, ul a:visited {
  text-decoration: none;
  background-color: #c0c0c0;
  color: black;
  padding: 4px;
  border: 2px outset #c0c0c0; }
```

```
ul a:hover { border-style: inset; color: blue;
  padding: 6px 3px 3px 6px; }
```

Po zastosowaniu tej metody wynik w przeglądarce Internet Explorer nie jest zadowalający (rysunek 2.48), natomiast inne przeglądarki wyświetlają menu prawidłowo (rysunek 2.49).



Rysunek 2.48. Menu poziome w przeglądarce IE



Rysunek 2.49. Menu poziome w przeglądarce Google Chrome

Wygląd menu w IE można poprawić, umieszczając elementy menu w blokach. Jeżeli elementy zostaną umieszczone w bloku, będzie można zdefiniować dla nich stałą szerokość.

Przykład 2.96

```
ul, ul li {display: block; list-style: none; margin: 0; padding: 0; }
ul li {float: left; margin-right: 10px; }

ul a:link, ul a:visited { display: block;
 position: relative;
  text-decoration: none;
  background-color: #c0c0c0;
  color: black;
  padding: 4px;
  border: 2px outset #c0c0c0; }

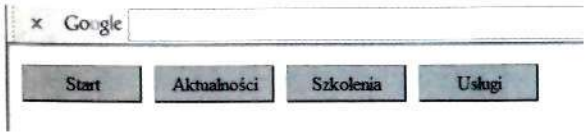
ul a:hover { border-style: inset; color: blue;
  padding: 6px 3px 3px 6px; }
```

Po umieszczeniu elementów menu w blokach wynik w przeglądarce Internet Explorer jest prawidłowy (rysunek 2.50).



Rysunek 2.50. Elementy menu umieszczone w blokach

Dodatkowo można określić stałą szerokość bloków i wyśrodkowanie tekstu menu (rysunek 2.51).



Rysunek 2.51. Elementy menu z określoną stałą szerokością

Przykład 2.97

```
ul a:link, ul a:visited { display: block; width: 100px;
 text-align: center;
  position: relative;
```



```
  text-decoration: none;
  background-color: #c0c0c0;
  color: black;
  padding: 4px;
  border: 2px outset #c0c0c0; }
```

Umieszczenie elementów w blokach daje możliwość zdefiniowania jednakowej szerokości wszystkich pozycji menu oraz ustalenia odległości między pozycjami. Mogą natomiast pojawić się problemy z prawidłowym wyświetlaniem w niektórych przeglądarkach. Trzeba również pamiętać o przypisaniu właściwości `clear` następnemu elementowi strony w celu uniknięcia „przyklejenia" tego elementu do menu.

2.7.3. Menu zagnieżdżone

Podobnie jak w przypadku menu pionowego lub poziomego, budując menu zagnieżdżone, należy zacząć od utworzenia listy. Lista powinna zawierać linki do odpowiednich miejsc wywoływanych przez menu.

Przykład 2.98

```
<ul id="blok">
  <li><a href="aktualnosci.html">Aktualności</a></li>
  <li><a href="szkolenia.html">Szkolenia</a>
    <ul>
      <li><a href="kurs1.html">HTML</a></li>
      <li><a href="kurs2.html">CSS</a></li>
      <li><a href="kurs3.html">JavaScript</a></li>
    </ul>
  </li>
  <li><a href="uslugi.html">Usługi</a></li>
</ul>
```

Po zdefiniowaniu zawartości menu można przystąpić do tworzenia menu zagnieżdżonego. Ponieważ zaproponowana lista ma podlistę tylko na drugiej pozycji, nasze menu będzie się rozwijało tylko na tej właśnie pozycji.

Podobnie jak w przypadku menu pionowego i poziomego, z przygotowanego menu trzeba usunąć punktory oraz wyzerować jego marginesy.

Przykład 2.99

```
ul, ul li {margin: 0; padding: 0;}
#blok li {list-style: none;}
```

W definicji stylów został zdefiniowany identyfikator `#blok`, który w treści dokumentu HTML został przypisany do znacznika listy `<ul>`.

Następnym etapem jest zdefiniowanie wyglądu menu.

Przykład 2.100

```
#blok, #blok ul {width: 140px;}

#blok li {padding-bottom: 1px;  position: relative;}
#blok ul { position: absolute; top: 0; left: 160px; padding-left: 1px;
visibility: hidden; }
#blok li:hover ul {visibility: visible; width: auto; height: auto;}
```

Listom została nadana szerokość `140px`, a elementy listy zostały rozdzielone przerwą (`padding-bottom: 1px`). Zdefiniowano położenie względne dla listy głównej w stosunku do strony internetowej oraz położenie podlisty w stosunku do listy głównej.

Dla wewnętrznej listy (`#blok ul`) zdefiniowano pozycjonowanie absolutne w stosunku do listy głównej oraz jej ukrycie (`visibility: hidden;`).

Gdy ustawimy kursor myszy na elemencie menu, wyświetli się podmenu.

Pozostało jeszcze zdefiniować, jak ma wyglądać element listy, gdy ustawimy nad nim kursor myszy. *Aby element menu głównego był nadal podświetlony, gdy zostanie* wybrany element podmenu, należy odpowiednio wykorzystać zasady dziedziczenia.

Przykład 2.101

```
#blok a, #blok li:hover li a {display: block; font: 12px/30px verdana,
sans-serif; text-decoration: none; padding: 0 10px; width: 140px; color:
black; background-color: #c0c0cf;}
#blok li:hover a, #blok li:hover li:hover a {background-color: #333033;
color: white; }
```

Pierwsza linia definiuje wygląd każdego elementu menu, druga — wygląd elementu wskazanego kursorem myszy. Jeżeli będzie to element podmenu, element z menu głównego też pozostanie podświetlony.

Efekt pracy pokazano w przykładzie poniżej.

Przykład 2.102

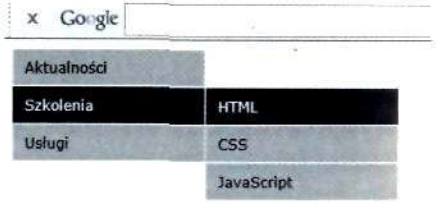
```
<head>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">
<title>Menu zagnieżdżone</title>
<style type="text/css">
ul, ul li {margin: 0; padding: 0;}
#blok li {list-style: none;}
#blok, #blok ul {width: 140px;}

#blok ul { position: absolute; top: 0; left: 160px; padding-left: 1px;
visibility: hidden; }
 #blok li {padding-bottom: 1px; position: relative;}
#blok li:hover ul {visibility: visible; width: auto; height: auto;}
```



```
#blok a, #blok li:hover li a {display: block; font: 12px/30px verdana,
sans-serif; text-decoration: none; padding: 0 10px; width: 140px; color:
black; background-color: #c0c0cf;}
#blok li:hover a, #blok li:hover li:hover a {background-color: #333033;
color: white; }
</style>
</head>
<body>
<ul id="blok">
  <li><a href="aktualnosci.html">Aktualności</a></li>
  <li><a href="szkolenia.html">Szkolenia</a>
    <ul>
      <li><a href="kurs1.html">HTML</a></li>
      <li><a href="kurs2.html">CSS</a></li>
      <li><a href="kurs3.html">JavaScript</a></li>
    </ul>
  </li>
  <li><a href="uslugi.html">Usługi</a></li>
</ul>
</body>
</html>
```

Wynik interpretacji kodu został pokazany na rysunku 2.52.



Rysunek 2.52. Menu zagnieżdżone

Zadanie 2.22

Korzystając z powyższego kodu, zdefiniuj menu zagnieżdżone dla pozostałych dwóch opcji menu głównego. Samodzielnie zaprojektuj wygląd menu z zastosowaniem arkusza stylów i poleceń języka CSS.

2.8. Zasady projektowania szablonu strony internetowej

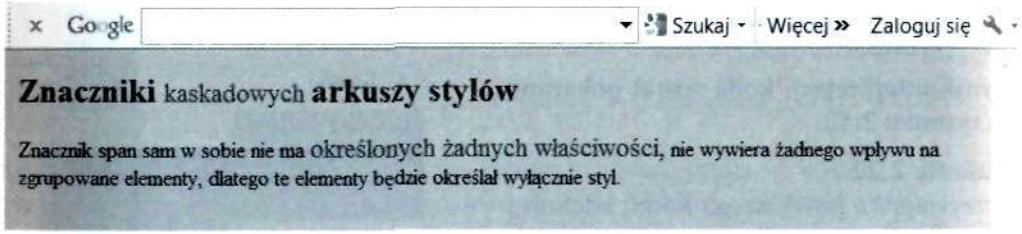
2.8.1. Definiowanie reguł

Reguły definiowane w arkuszu CSS dla znacznika dotyczą wszystkich użytych w dokumencie znaczników tego typu. Oznacza to, że jeżeli zdefiniujemy w arkuszu stylów wygląd znacznika `<span>`, to w każdym miejscu na stronie tekst otoczony znacznikiem `<span>` będzie wyglądał tak samo, niezależnie od tego, czy znajdzie się w nagłówku strony, w treści dokumentu, czy w menu.

Przykład 2.103

```
<html>
<head>
<style type="text/css">
body {background-color: #c0c0c0; }
span {font-size: 14pt; color: blue; font-weight: normal; }
</style>
</head>
<body>
<h2>Znaczniki <span> kaskadowych </span>arkuszy stylów </h2>
<p>Znacznik span sam w sobie nie ma <span> określonych żadnych
właściwości,</span> nie wywiera żadnego wpływu na zgrupowane elementy,
dlatego te elementy będzie określał wyłącznie styl. </p>
</body>
</html>
```

W podanym przykładzie znacznikiem został otoczony fragment tekstu umieszczony w nagłówku <h2> oraz fragment akapitu. Wynik interpretacji kodu widać na rysunku 2.53. Teksty sformatowane z użyciem znacznika wyglądają tak samo.



Rysunek 2.53. Formatowanie tekstów otoczonych znacznikiem

Aby odróżnić przy formatowaniu podobne elementy umieszczane w różnych miejscach dokumentu, możemy zastosować jedną z trzech technik:

- reguły kontekstowe,
- identyfikatory,
- klasy.

Reguły kontekstowe

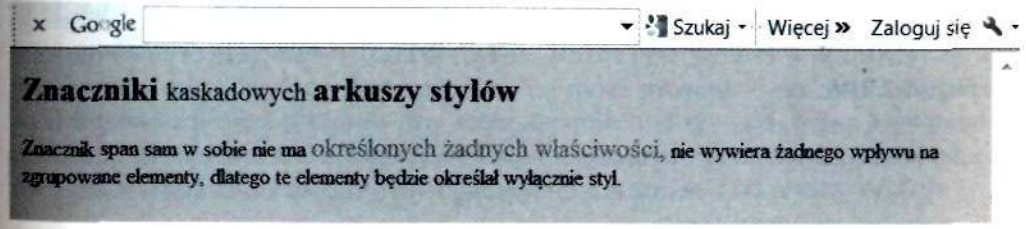
Reguły kontekstowe pozwalają na zdefiniowanie w arkuszu stylów wyglądu każdego elementu w zależności od jego położenia.

W podanym przykładzie teksty sformatowane z użyciem znacznika wyglądają tak samo, mimo że znajdują się w różnych miejscach i byłoby wskazane, aby wyglądały inaczej. Jeżeli w trakcie definiowania reguł określimy kontekst znacznika , będzie możliwe sformatowanie w inny sposób obu fragmentów tekstu.

Przykład 2.104

```
<html>
<head>
<style type="text/css">
body {background-color: #c0c0c0; }
h2 span {font-size: 14pt; color: blue; font-weight: normal; }
p span {font-size: 14pt; color: red; font-weight: normal; }
</style>
</head>
<body>
<h2>Znaczniki <span> kaskadowych </span>arkuszy stylów </h2>
<p>Znacznik span sam w sobie nie ma <span> określonych żadnych
właściwości,</span> nie wywiera żadnego wpływu na zgrupowane elementy,
dlatego te elementy będzie określał wyłącznie styl. </p>
</body>
</html>
```

W podanym przykładzie znacznik został zdefiniowany w kontekście nagłówka <h2> oraz akapitu <p>. Efekt jest taki, że każdy fragment otoczony znacznikiem wygląda inaczej (rysunek 2.54).



Rysunek 2.54. Zastosowanie reguł kontekstowych

Identyfikatory

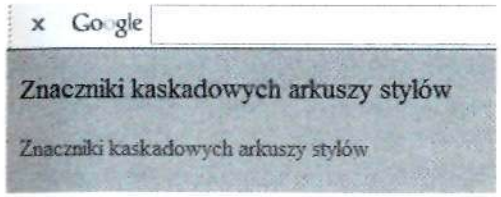
Nie zawsze możliwe jest definiowanie wyglądu elementów w zależności od kontekstu. Na przykład próba zdefiniowania różnego wyglądu dla dwóch akapitów znajdujących się w tym samym obszarze dokumentu przy użyciu reguł kontekstowych nie powiedzie się, ponieważ obydwa akapity występują w tym samym kontekście. W takiej sytuacji można zastosować identyfikatory. Identyfikator jednoznacznie określa wybrany element. Dzięki niemu każdy, nawet najmniejszy element strony może wyglądać inaczej.

Przykład 2.105

```
<html>
<head>
<style type="text/css">
body {background-color: #c0c0c0; }
p#ak1 {font-size: 14pt; color: blue; }
p#ak2 {font-size: 12pt; color: green; }
</style>
```

```
</head>
<body>
<p id="ak1">Znaczniki kaskadowych arkuszy stylów </p>
<p id="ak2">Znaczniki kaskadowych arkuszy stylów </p>
</body>
</html>
```

Każdy akapit będzie wyświetlony w inny sposób, ponieważ zostały im przypisane identyfikatory, a w arkuszu stylów dla każdego identyfikatora zdefiniowano inny sposób wyświetlania (rysunek 2.55).



Rysunek 2.55.
Zastosowanie identyfikatorów

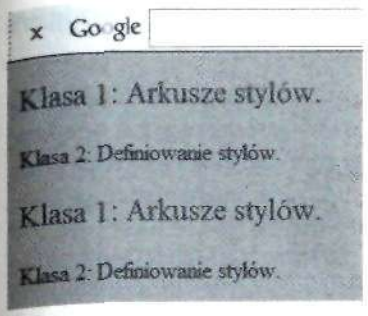
Klasy

Czasami na stronie znajduje się grupa elementów, które powinny wyglądać tak samo. W takiej sytuacji nie trzeba każdemu z nich przypisywać identyfikatora. Można te elementy połączyć w grupy i dla każdej zdefiniować klasę, a następnie, używając pojedynczych reguł w CSS, za jej pomocą określić wygląd całej grupy elementów.

Przykład 2.106

```
<html>
<head>
<title> Klasy</title>
<style type="text/css">
body {background-color: #c0c0c0;}
p.tekst1 {color: red; font-size: 16pt;}
p.tekst2 {color: green; font-size: 12pt;}
</style>
</head>
<body>
<p class="tekst1"> Klasa 1.: Arkusze stylów.</p>
<p class="tekst2"> Klasa 2.: Definiowanie stylów.</p>
<p class="tekst1"> Klasa 1.: Arkusze stylów.</p>
<p class="tekst2"> Klasa 2.: Definiowanie stylów.</p>
</body>
</html>
```

Wynik zdefiniowania dwóch klas dla akapitu został pokazany na rysunku 2.56.



Rysunek 2.56.
Zastosowanie różnych klas do formatowania akapitów

Jeżeli wybrany element jest definiowany za pomocą kilku reguł, to obowiązują następujące priorytety:

1. Identyfikator.
2. Klasa.
3. Znacznik kontekstowy.
4. Znacznik.

Znacznik *

W modelu blokowym CSS wszystkie elementy można traktować jak prostokątne bloki. Dla każdego z tych bloków można definiować marginesy zewnętrzne i wewnętrzne oraz obramowania. Jeżeli dla elementu umieszczanego na stronie te parametry nie zostaną zdefiniowane, to przyjmą wartości domyślne. Różne przeglądarki w odmienny sposób interpretują wartości domyślne elementu, co może prowadzić do nieprawidłowego wyświetlania zawartości strony. Aby temu zapobiec, należy jako jedną z pierwszych deklaracji strony umieścić deklarację zerowania marginesów zewnętrznych i wewnętrznych dla wszystkich elementów występujących na stronie.

Specjalny znacznik, który możemy wykorzystać do tego celu, to <*>. Zdefiniowanie dla niego reguły oznacza, że powstała reguła odnosi się do wszystkich znaczników na stronie.

WSKAZÓWKA

Zdefiniowanie zerowych marginesów i odstępów wewnętrznych powinno być początkiem pracy nad każdą stroną.

Przykład 2.107

```
* { margin: 0px; padding: 0px; }
```

Bloki danych

Do rozmieszczania elementów na stronie stosuje się najczęściej omówiony już element grupujący . Jest to element, który grupuje elementy strony, ale nie w celu ich formatowania, tylko w celu ich odpowiedniego rozmieszczenia. Znaczniki są odpowiednikami ramek, w których zgrupowane elementy można dowolnie przemieszczać na stronie. Znacznikom można przypisywać identyfikatory lub klasy. Poprzez deklarację reguły dla identyfikatorów i klas każdy znacznik może być

formatowany. Wewnątrz znacznika mogą być umieszczane inne elementy, np. obrazy, teksty lub kolejne znaczniki .

Element tworzy blok, w którym są umieszczane inne elementy, może więc się zdarzyć, że element umieszczony wewnątrz znacznika jest szerszy niż wielkość tego znacznika i nie mieści się w nim. W takim przypadku zbyt szerokie elementy mogą zachodzić na sąsiednie bloki, zasłaniając ich zawartość, lub blok może zostać przeniesiony niżej. Takie zachowanie bloków niszczy układ całej strony.

Aby uniknąć tych problemów, należy dla znaczników umieszczanych na stronie zdefiniować atrybut overflow jako hidden. Atrybut overflow definiuje sposób wyświetlania, gdy zawartość nie mieści się w zdefiniowanym rozmiarze bloku; wartość hidden wymusi obcięcie niemieszczącego się fragmentu. Fragment zawartości przestanie być widoczny na stronie, ale zostanie zachowany jej układ.

Przykład 2.108

```
div { overflow: hidden; }
```

WSKAZÓWKA

Zaczynając pracę nad nową stroną, trzeba dla wszystkich znaczników zdefiniować atrybut overflow: hidden.

Kolory

Dobrym zwyczajem jest również zdefiniowanie koloru tła i koloru tekstu. Domyślny kolor tła to biały, a tekstu — czarny. Nie można jednak zakładać, że są to kolory obowiązujące u wszystkich użytkowników naszej strony. Jeżeli użytkownik zmieni kolory domyślne w systemie operacyjnym swojego komputera, to wyświetlając zaprojektowaną przez nas stronę, może uzyskać nieoczekiwane efekty kolorystyczne.

Przykład 2.109

```
body { background-color: #fff;
  color: #000;}
```

Wymiary strony

Przystępując do projektowania szablonu strony o stałej szerokości, należy ustalić tę szerokość. Zwykle przyjmuje się, że strona internetowa jest przygotowana dla ekranu o rozdzielczości 1024px. Oczywiście można wybrać dowolną szerokość, ale trzymając się standardów obowiązujących w internecie, uzyskamy lepsze wyniki wyświetlania strony w dowolnej przeglądarce.

Przyjmując, że na stronie pojawia się suwak do przewijania treści, przy zerowych marginesach strony szerokość szablonu powinna wynosić 1000px. Jeżeli dla projektowanego szablonu strony zostanie zdefiniowany zewnętrzny element , w którym będzie umieszczona cała zawartość strony, i nadamy mu identyfikator ramka, to właśnie ten element powinien mieć szerokość 1000px.

Przykład 2.110

```
#ramka {
  width: 1000px;
  margin-right: auto;
  margin-left: auto;
}
```

Aby strona była zawsze widoczna na środku przeglądarki, niezależnie od rozmiaru jej okna, należy ustawić wartości marginesu prawego i lewego na auto. Oznacza to, że obydwa marginesy będą miały taką samą wartość i że będzie ona automatycznie dostosowywana do szerokości okna przeglądarki.

2.8.2. Szablon strony (div)

Pamiętajmy, że używając znacznika , możemy grupować w bloki elementy strony i formatować je za pomocą stylów. Bloki mogą zawierać akapity, listy, obrazy oraz inne bloki. Umożliwiają one wydzielanie na stronie większych, logicznych fragmentów w celu nadania im specyficznego formatowania za pomocą stylów CSS. Z tych powodów bloki świetnie nadają się do budowy szablonu strony internetowej. Ponieważ tak zbudowana strona może zawierać wiele znaczników , w celu ich odróżnienia warto każdemu z nich przypisać identyfikator (atrybut id). Jeżeli definiujemy znaczniki , które mogą się powtórzyć na stronie, należy przypisać je do klasy (atrybut class).

W praktyce najczęściej mamy do czynienia z szablonami typu:

- nagłówek, jedna kolumna, stopka;
- nagłówek, dwie kolumny, stopka;
- nagłówek, trzy kolumny, stopka.

Jeżeli zajrzymy na najpopularniejsze portale internetowe, najczęściej spotkamy wersję z dwoma lub trzema kolumnami. Witryny tematyczne często mają układ dwukolumnowy: pierwsza kolumna zawiera menu tematyczne, druga — treść. Mogą wystąpić różne modyfikacje tych układów, ale podstawowa struktura w postaci kolumn zostaje zachowana. Można planować układ z większą liczbą kolumn, ale przy wielu kolumnach strona staje się nieczytelna.

Szablon oparty na stałej liczbie kolumn jest najczęściej stosowaną formą układu strony. Jest on niezależny od rozdzielczości ekranu, a elementy rozplanowane na stronie zawsze pozostają w miejscach, w których zostały zaplanowane.

Szablon z trzema kolumnami

Na nasze potrzeby przygotujemy szablon strony z trzema kolumnami.

W pierwszym etapie musimy w języku HTML zdefiniować szablon strony przy użyciu znaczników .

Przykład 2.111

```
<body>
<div id="naglowek"> Tutaj znajdzie się zawartość nagłówka strony</div>
<div id="ramka">
<div class="kolumny" id="kol1"> Tutaj znajdzie się zawartość kolumny1</
div>
<div class="kolumny" id="kol2"> Tutaj znajdzie się zawartość kolumny2</
div>
<div class="kolumny" id="kol3"> Tutaj znajdzie się zawartość kolumny3</
div>
</div>
<div id="stopka"> Tutaj znajdzie się zawartość stopki strony</div>
</div>
</body>
```

Tak przygotowany szablon stanowi strukturę strony, którą za pomocą stylów CSS można w dowolny sposób formatować. Projektując kolejne arkusze CSS dla tego samego kodu HTML, możemy zmieniać wygląd strony, natomiast wypełniając kod odpowiednią treścią, możemy tworzyć kolejne strony oparte na tym samym szablonie.

W przygotowanym szablonie wszystkie bloki zostały ułożone jeden pod drugim. Aby kolumny szablonu ułożyć pionowo obok siebie, należy użyć atrybutu `float`.

Atrybut ten był już wykorzystywany do oblewania ilustracji tekstem. Jeżeli dla kilku sąsiednich elementów nadamy atrybutowi `float` wartość `left`, to elementy te ustawią się obok siebie, pod warunkiem że pozwoli na to szerokość bloku, w którym się znajdują.

W związku z tym wystarczy dla kolumn zdefiniować regułę zawierającą atrybut `float: left`. Musimy pamiętać, aby suma szerokości wszystkich kolumn nie przekroczyła `1000px`.

Przykład 2.112

```
#kol1 {float: left; width: 30%; }
#kol2 {float: left; width: 30%; }
#kol3 {float: left; width: 30%; }
```

lub

Przykład 2.113

```
.kolumny { float: left; width: 30%; }
```

Należy pamiętać również o tym, aby stopce nadać odpowiednie atrybuty uniemożliwiające jej wyświetlenie z lewej strony kolumn. Posłuży do tego atrybut `clear` z wartością `both`.

Przykład 2.114

```
#stopka { clear: both; width: 100%; }
```

Jeśli do projektowania szablonu strony wykorzystamy wcześniej poznane reguły projektowania, to kod szablonu strony może być podobny do podanego w poniższym przykładzie.

Przykład 2.115

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://
www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Szablon strony</title>
<style type="text/css">
* {
  margin: 0px;
  padding: 0px;
}
div {
  overflow: hidden;
}
#ramka {
  width: 1000px;
  margin-right: auto;
  margin-left: auto;
}
#naglowek {
  width: 100%;
  background-color: #FCF;
}
#stopka {
  width: 100%;
  background-color: #93F;
}
.kolumny {
  width: 30%;
  margin-top: 10px;
  margin-right: 2%;
  margin-bottom: 2%;
  margin-left: 10px;
}
#kol1 {
  float: left;
  background-color: #3FF;
}
#kol2 {
  float: left;
  background-color: #CF3;
}
```

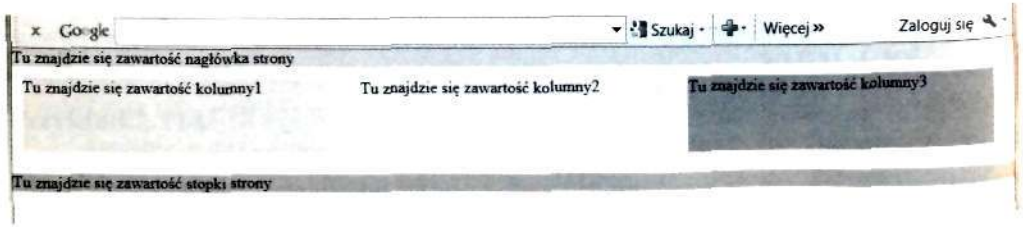
```
#kol3 {
  background-color: #F93;
}
</style>
</head>

<body>
<div id="ramka">
<div id="naglowek">Tu znajdzie się zawartość nagłówka strony</div>

  <div class="kolumny" id="kol1">
    <p>Tu znajdzie się zawartość kolumny1</p>
    <p> </p>
    <p> </p>
    <p> </p>
  </div>
  <div class="kolumny" id="kol2">
    <p>Tu znajdzie się zawartość kolumny2</p>
    <p> </p>
    <p> </p>
    <p> </p>
  </div>
  <div class="kolumny" id="kol3">
    <p>Tu znajdzie się zawartość kolumny3</p>
    <p> </p>
    <p> </p>
    <p> </p>
  </div>

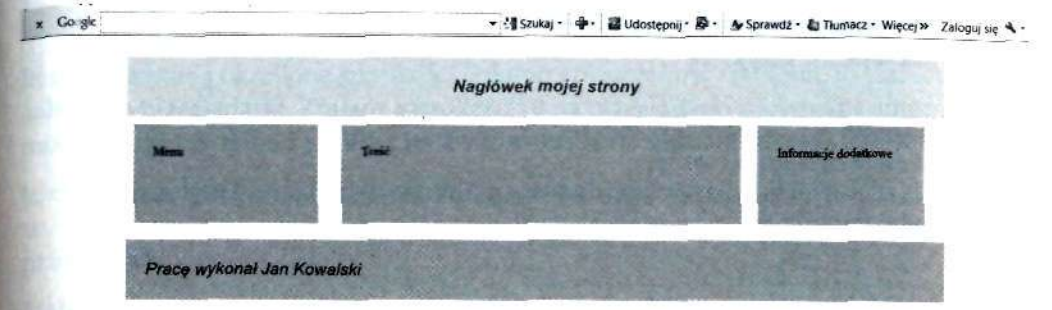
<div id="stopka">Tu znajdzie się zawartość stopki strony</div>
</div>
</body>
</html>
```

Wynik interpretacji kodu pokazuje rysunek 2.57.



Rysunek 2.57. Szablon strony z trzema kolumnami

Po określeniu przeznaczenia strony szablon może zostać zindywidualizowany, co pokazuje rysunek 2.58.



Rysunek 2.58. Zmodyfikowany szablon strony z trzema kolumnami

Zadanie 2.23
Przygotuj własny szablon strony internetowej, który wykorzystasz w dalszej pracy.

2.8.3. Płynny szablon strony

Płynny szablon pozwala, aby przy zmianie szerokości okna przeglądarki zmieniała się szerokość kolumn strony. Najczęściej zmiana szerokości dotyczy tylko wybranych kolumn, np. kolumna z menu pozostaje niezmieniona, natomiast płynnie zmienia się szerokość kolumny z treścią.

Na nasze potrzeby przygotujemy szablon strony z dwoma kolumnami.

Przykład 2.116

```
<body>
<div id="naglowek"> Tutaj znajdzie się zawartość nagłówka strony</div>
<div id="ramka">
<div class="kolumny" id="kol1"> Tutaj znajdzie się zawartość kolumny1</
div>
<div class="kolumny" id="kol2"> Tutaj znajdzie się zawartość kolumny2</
div>
<div id="stopka"> Tutaj znajdzie się zawartość stopki strony</div>
</div>
</body>
```

Dla tak przygotowanego szablonu można przy wykorzystaniu reguł CSS określić sposób wyświetlania kolumn.

Przykład 2.117

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
```

```
* { margin: 0px; padding: 0px;}
div {overflow: hidden; padding: 25px; }
#ramka { margin-right: auto; margin-left: auto;
  min-width: 400px;
  max-width: 900px;
}
#naglowek {
  background-color: #FCF;
  font-size:16pt; color:red; font-family: Arial; font-style: italic;
  text-align: center; font-weight:bold;
}
#stopka {
  background-color: #c0c0c0;
  font-size:16pt; color:blue; font-family: Arial; font-style: italic;
  clear:both;
}
.kolumny {
  margin-top: 10px;
  margin-right: 2%;
  margin-bottom: 2%;
  margin-left: 10px;
  background-color: #c0c0c0;
}
#kol1 {
  float: left;
  width:100px;
}
#kol2 {
}
</style>
</head>

<body>
<div id="ramka">
<div id="naglowek">Nagłówek mojej strony</div>
  <div class="kolumny" id="kol1">
    <p>Menu</p>
    </div>
  <div class="kolumny" id="kol2">
    <p>Treść

</p>
  </div>
```



```
<div id="stopka">Pracę wykonał Jan Kowalski</div>
</div>
</body>
</html>
```

Jeżeli nie zostaną określone szerokości kolumn ani szerokość szablonu, to szablon zajmie całe okno przeglądarki. Zmiana szerokości okna przeglądarki spowoduje automatyczną modyfikację zawartości strony i jej dostosowanie do aktualnych rozmiarów okna.

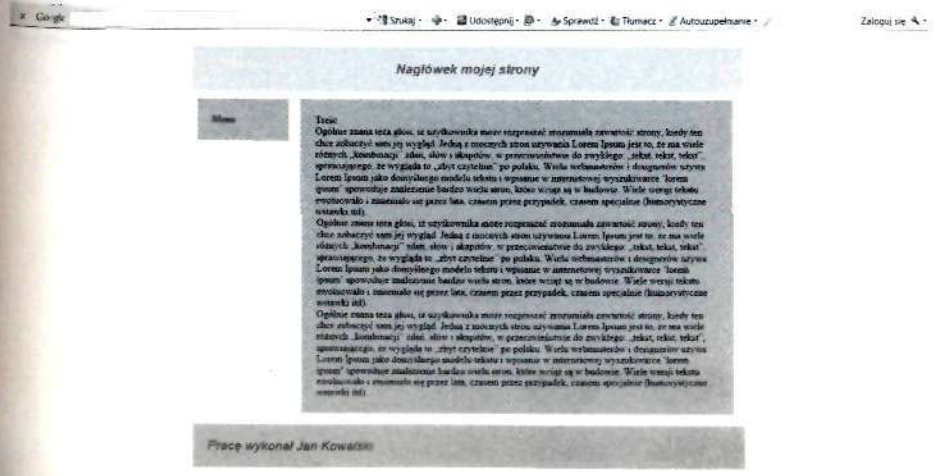
Jeżeli w podanym wyżej kodzie umieścimy w kolumnie drugiej większy fragment tekstu, zaobserwujemy zachowanie się zawartości strony przy zmianie szerokości okna przeglądarki.

Mało praktyczne jest projektowanie strony, która będzie się rozciągała do granic możliwości monitora (szczególnie gdy weźmiemy pod uwagę monitory o rozdzielczości Full HD). Lepiej zaprojektować stronę tak, aby powyżej pewnej szerokości ekranu jej szerokość już się nie zwiększała, a poniżej pewnej szerokości okna przeglądarki — nie zmniejszała. Aby uzyskać ten efekt, należy zdefiniować minimalną i maksymalną szerokość bloku zewnętrznego (w powyższym przykładzie identyfikator `ramka`).

Przykład 2.118

```
#ramka { margin-right: auto;
  margin-left: auto;
  min-width: 400px;
  max-width: 900px; }
```

Po ustawieniu tych parametrów zawartość strony będzie skalowana w zakresie podanych wartości. Dalsze zwiększanie szerokości spowoduje pokazanie tła strony (rysunek 2.59), natomiast zmniejszanie szerokości poniżej zadeklarowanej spowoduje ukrycie części zawartości i wyświetlenie suwaków, za pomocą których będzie można obejrzeć ukryte fragmenty strony (rysunek 2.60).



Rysunek 2.59. Płynny szablon strony. Okno przeglądarki zmaksymalizowane



Rysunek 2.60.
Płynny szablon strony. Okno przeglądarki zminimalizowane

Jeżeli planujemy umieszczenie menu w pierwszej kolumnie, to kolumna ta powinna mieć stałą szerokość, a dynamicznie zmieniać się powinna tylko szerokość drugiej kolumny.

Przykład 2.119

```
#koll { float: left; width: 100px; }
```

Gdy taką deklarację wprowadzimy dla kolumny pierwszej, to dla drugiej kolumny nie będziemy musieli deklarować szerokości, ponieważ automatycznie wypełni ona pozostałą część strony. Należy pamiętać o jeszcze jednej zależności, mianowicie w przypadku układów o płynnej szerokości nie ma potrzeby ustawiania atrybutu `float` dla kolumny położonej z prawej strony. Dlatego z kodu dokumentu trzeba usunąć ten parametr.

Zadanie 2.24

Na podstawie przygotowanego wcześniej szablonu strony zaprojektuj stronę internetową o dowolnej tematyce.

PYTANIA I POLECENIA KONTROLNE

1. Jaką rolę w tworzeniu strony internetowej odgrywają arkusze stylów CSS?
2. W jaki sposób i gdzie definiowany jest styl lokalny?
3. Do czego służy znacznik **?
4. Podaj różnice w stosowaniu zewnętrznego i wewnętrznego arkusza stylów CSS.
5. Podaj ogólną postać selektora klasy.
6. Kiedy stosuje się selektor identyfikatora?
7. Co oznacza zapis: *a:link {color: green; background: yellow;}* występujący w arkuszu stylów?
8. Jaką rolę w modelu blokowym CSS odgrywają *Border* i *Padding*?
9. Na czym polega pozycjonowanie elementów na stronie internetowej?
10. Jaką rolę podczas definiowania stylów odgrywa znacznik *<*>*?

ZADANIE

Dla utworzonej witryny internetowej na temat konkursów przedmiotowych odbywających się w szkole należy zdefiniować styl wyświetlanych na stronach informacji. Uczestnicy projektu ustalą wygląd tytułów stron, tekstów zawierający informacje, różnego rodzaju list, zdefiniują zachowanie się odsyłaczy do innych stron internetowych oraz określą sposób rozmieszczenia zdjęć na stronie. Po tych ustaleniach zostanie utworzony zewnętrzny arkusz stylów zawierający definicję stylów dla wszystkich elementów strony internetowej. Utworzony arkusz stylów zostanie dołączony do dokumentów HTML przygotowanych przez uczestników projektu.

3

Edytory WYSIWYG

Edytory typu **WYSIWYG** (ang. *What You See Is What You Get* — „otrzymujesz to, co widzisz") pozwalają uzyskać dokument tekstowy lub graficzny zbliżony do obrazu tworzonego na ekranie. Użytkownik podczas pracy z dokumentem widzi na ekranie dokładnie to, co później zostanie wydrukowane lub wyświetlone w oknie przeglądarki internetowej. Wśród edytorów typu WYSIWYG wyróżnia się: procesory tekstu, edytory stron WWW, programy CAD, programy DTP.

3.1. Edytory stron WWW

Edytory stron WWW to programy, które służą do tworzenia stron internetowych bez konieczności wpisywania kodu źródłowego. Podstawowym operacjom zostały przypisane ikony. Ikony przypisano również obiektom najczęściej umieszczanym na stronach, takim jak tabele, obrazy, bloki. Wybór ikony powoduje umieszczenie na stronie obiektu i pozwala definiować jego właściwości. W tle automatycznie tworzony jest kod odpowiadający za projektowany w edytorze wygląd strony i za jej funkcjonalność. Edytory stron WWW to bardzo wygodne narzędzia oszczędzające czas i ułatwiające pracę projektantom stron internetowych. Ich zaletą jest również to, że można — po przełączeniu edytora w odpowiedni tryb pracy — obejrzeć kod źródłowy tworzonej strony. Istnieje też możliwość dzielenia ekranu na dwie części: w jednej widoczny jest kod źródłowy, w drugiej — układ projektowanej strony.

3.1.1. Funkcje edytorów WYSIWYG

Najprostsze edytory WYSIWYG umożliwiają zmianę podstawowych parametrów tworzonego dokumentu, takich jak pogrubienie czcionki, jej wielkość i krój.

Większość edytorów pozwala również na wstawianie plików graficznych i linków. Niektóre edytory zawierają narzędzia do tworzenia tabel i ich modyfikowania czy do tworzenia list numerowanych i punktowanych. Najbardziej zaawansowane edytory umożliwiają tworzenie i modyfikowanie informacji nagłówkowych, obsługę arkuszy CSS i skryptów, mapowanie odsyłaczy. Są wyposażone w narzędzia do: tworzenia szablonów i bibliotek, podglądu dokumentu w przeglądarce, kontroli spójności, weryfikacji odsyłaczy i wysyłania strony lub witryny na serwer.

Praca z takimi edytorami nie wymaga od użytkownika znajomości języka HTML, co jest ich główną zaletą. Dzięki nim można szybko i bez konieczności żmudnego pisania kodu tworzyć atrakcyjne wizualnie strony.

Do ich wad zaliczymy rozbudowany system opisów elementów wstawianych na stronę oraz dodawanie różnych niepotrzebnych znaczników. Kod staje się przez to nieczytelny, a zmiany w strukturze dokumentu są bardzo trudne. Powstające pliki mają duże rozmiary. Powoduje to o wiele dłuższe wczytywanie się strony, transfer jest wolniejszy. Tak utworzone strony są trudne do optymalizacji.

3.1.2. Dostępne edytory WYSIWYG

Adobe Dreamweaver
Jednym z edytorów WYSIWYG jest program **Adobe Dreamweaver**. Jest to narzędzie do tworzenia zaawansowanych stron i aplikacji internetowych. Wspiera technologie PHP5, ASP, ASP.NET, JSP i AJAX. W jego skład wchodzą narzędzia do tworzenia stron internetowych, np. narzędzie do kontroli właściwości CSS czy do testowania stron w różnych przeglądarkach.

Areo 3W
Areo 3W to program przeznaczony do tworzenia stron internetowych w trybie WYSIWYG. Zawiera takie funkcje jak: podświetlanie składni języków HTML, XHTML, JavaScript, CSS, PHP, SQL, sprawdzanie i poprawianie kodu źródłowego, automatyczne zamykanie znaczników HTML i PHP. Dodatkowo ma dołączony program do animacji plików GIF, pozwalający tworzyć ruchome obrazki do wykorzystania na stronach internetowych. Publikację gotowych materiałów ułatwia wbudowany klient FTP.

WYSIWYG Web Builder
WYSIWYG Web Builder to bezpłatna wersja programu do tworzenia stron internetowych pracującego w trybie WYSIWYG. Program ten przede wszystkim generuje strony, natomiast jego możliwości edytowania kodu źródłowego HTML są znacznie ograniczone. Samo tworzenie stron jest bardzo proste. Program ma gotowe szablony stron, a także wbudowanego klienta FTP, dzięki czemu utworzoną stronę można łatwo opublikować.

Visual Web Developer 2010 Express Edition
Visual Web Developer 2010 Express Edition to darmowe narzędzie przeznaczone do tworzenia aplikacji internetowych (ASP.NET) na platformie .NET Framework 3.5. Jest to część bezpłatnego środowiska programistycznego Visual Studio 2010 Express. Program ma edytor wizualny, rozbudowany edytor kodu, system inteligentnych podpowiedzi oraz debugger. Oferuje pełną integrację z Microsoft SQL Server 2008.

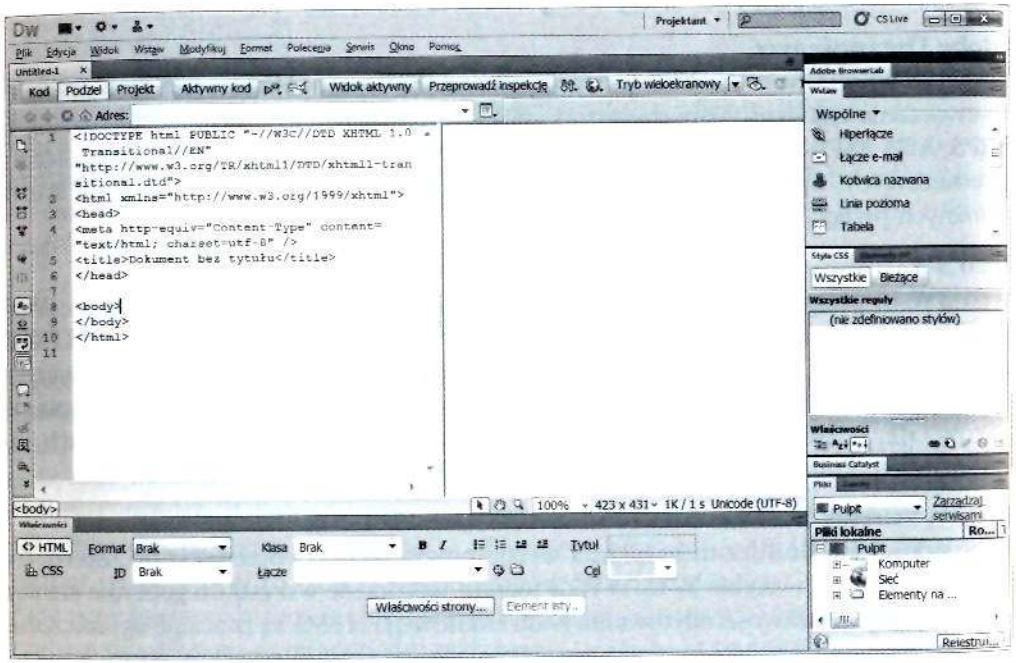
Wersja Express to bezpłatne wydanie komercyjnego pakietu Visual Studio 2010. Charakteryzuje się modułową budową i dostosowaniem dla początkujących programistów przy jednoczesnym zachowaniu wszystkich najważniejszych cech pełnego Visual Studio. Oprogramowanie można wykorzystywać w celach komercyjnych. Wszystkie aplikacje wraz z kodem można też bezproblemowo przenieść do pełnej wersji Visual Studio 2010 i tam je dalej rozwijać.

3.2. Adobe Dreamweaver

Edytorem WYSIWYG, który daje duże możliwości tworzenia stron internetowych, jest program Adobe Dreamweaver.

3.2.1. Tworzenie dokumentów HTML

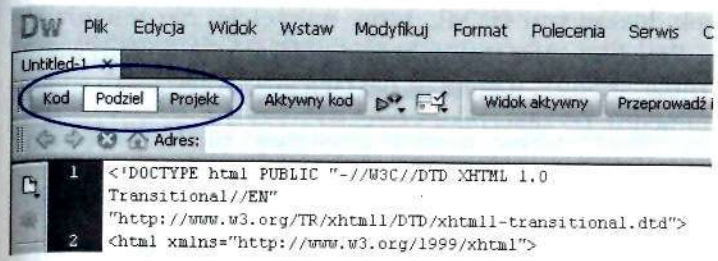
Aby rozpocząć tworzenie dokumentu HTML, po uruchomieniu programu należy w oknie powitalnym wybrać opcję *HTML*. Zostanie otwarte okno robocze pokazane na rysunku 3.1.



Rysunek 3.1. Okno programu Dreamweaver CS5

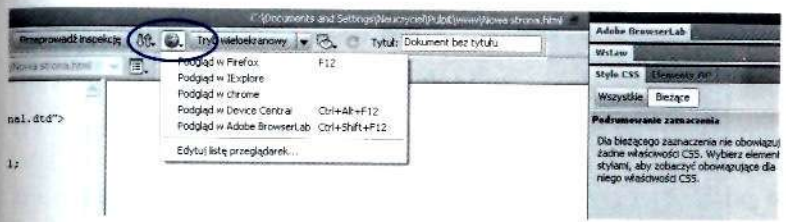
Pracę rozpoczynamy od zapisania dokumentu na dysku, wybierając *Plik/Zapisz jako* i podając nazwę pliku. Dreamweaver sam doda rozszerzenie *html*. Na zakładce z dokumentami pojawi się nazwa zapisanego pliku.

Aplikacja umożliwia pracę w dwóch trybach: w trybie tworzenia lub modyfikowania kodu HTML oraz w trybie projektu (zakładki *Kod* oraz *Projekt* w lewym górnym rogu okna) (rysunek 3.2). Można wybrać zakładkę *Podziel*, w wyniku czego okno programu zostanie podzielone na dwie części. W pierwszej widoczny będzie kod programu, w drugiej — projekt. Jest to bardzo wygodny sposób pracy, ponieważ tworząc stronę, możemy analizować i modyfikować jej kod, a modyfikując kod — oglądać wpływ zmian na wygląd strony bez konieczności uruchamiania przeglądarki.



Rysunek 3.2. Opcje wyboru trybu pracy

Niestety, nie wszystkie efekty definiowania kodu są dobrze interpretowane w widoku *Projekt* i dlatego w celu sprawnego przeglądania stron program został wyposażony w ikonę uruchamiającą wyświetlanie kodu strony w wybranej przez użytkownika przeglądarce internetowej. Ikona ta znajduje się na pasku zadań w górnej części okna (rysunek 3.3).



Rysunek 3.3. Ikona uruchomienia przeglądarki

Widok *Projekt* służy nie tylko do podglądania, jak wygląda projektowana strona. Widok ten jest również edytorem, za pomocą którego można zmieniać jej wygląd, umieszczając na niej różnego rodzaju elementy i definiując ich właściwości bez konieczności modyfikowania kodu źródłowego.

Jeżeli po rozpoczęciu pracy z nowym dokumentem przełączymy się na widok *Kod*, to przekonamy się, że program Dreamweaver wygenerował już pewien kod (rysunek 3.4).

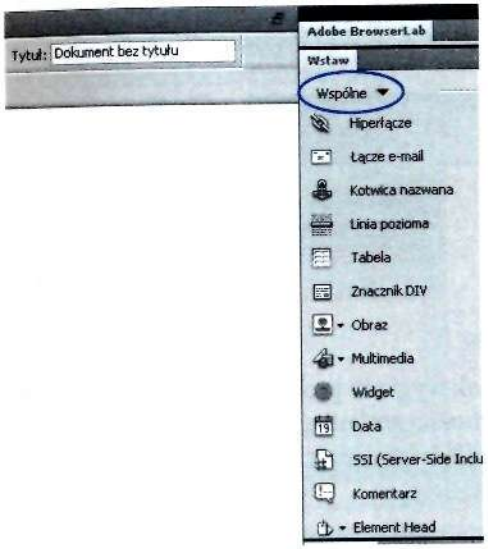


Rysunek 3.4. Nagłówek dokumentu generowany automatycznie

Jest to znany nam nagłówek dokumentu HTML, który najczęściej nie zmienia się w kolejnych dokumentach. Projektowanie strony będzie polegało na dokonywaniu kolejnych wpisów po znaczniku `<body>`, natomiast deklaracje wewnętrznych stylów znajdą się w nagłówku dokumentu, po znaczniku `<title>`.

3.2.2. Formatowanie tekstu

Dodawanie nowych elementów i ich formatowanie odbywa się z wykorzystaniem paneli *Wstaw* i *Style CSS* wyświetlanych z prawej strony ekranu. Jeżeli rozwiniemy panel *Wstaw* i wybierzemy z listy opcję *Wspólne*, to na liście znajdziemy elementy, które mogą zostać umieszczone na projektowanej stronie (rysunek 3.5).



Rysunek 3.5.
Opcja Wspólne w panelu Wstaw



Rysunek 3.6.
Opcja Tekst w panelu Wstaw

Wybór na liście opcji *Tekst* udostępni elementy związane z formatowaniem tekstu (rysunek 3.6).

Przy ich użyciu można w łatwy sposób formatować tekst i dodawać kolejne znaczniki kodu HTML.

Zadanie 3.1
Korzystając z narzędzi do definiowania tekstu i widoku *Projekt*, utwórz stronę taką jak widoczna na rysunku 3.7.

Formatowanie tekstu

Coś się dzieje dzisiaj z moją myszą / pogrubiony

Coś się dzieje dzisiaj z moją myszą / kursywa, rozmiar czcionki większy

Coś się dzieje dzisiaj z moją myszą / podkreślony

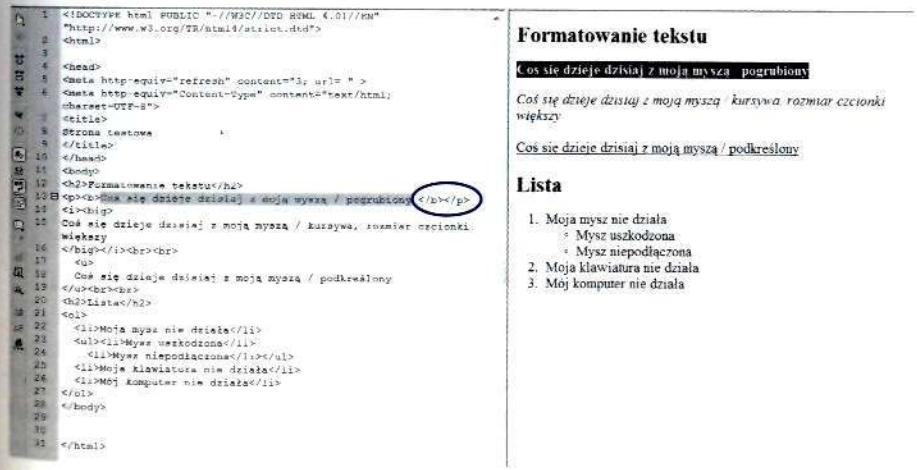
Lista

1. Moja mysz nie działa
   - Mysz uszkodzona
   - Mysz niepodłączona
2. Moja klawiatura nie działa
3. Mój komputer nie działa

Rysunek 3.7. Przykład strony wykorzystującej narzędzia związane z tekstem

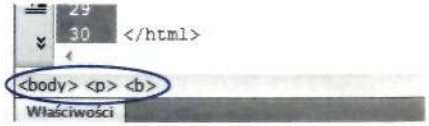
Sprawdź efekt swojej pracy w widoku *Kod*.

Usuwanie zdefiniowanych w widoku *Projekt* funkcji formatowania wymaga przełączenia się do widoku *Kod* i kontrolowania zaznaczonego tekstu. Jeżeli w widoku *Projekt* zostanie zaznaczony tekst, którego formatowanie będzie zmieniane, w widoku *Kod* będzie widać, że został zaznaczony sam tekst, bez otaczających go znaczników (rysunek 3.8).



Rysunek 3.8. Zaznaczony tekst w widokach Kod i Projekt

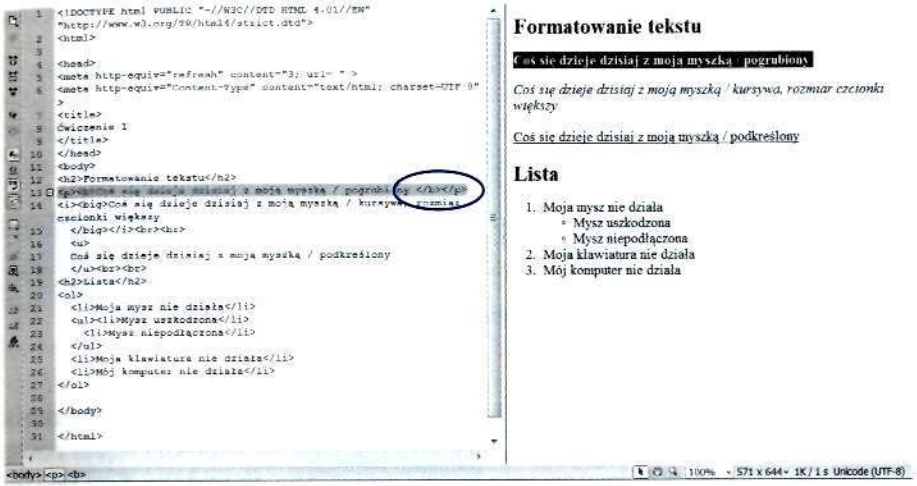
W takim przypadku należy wykorzystać drzewo kontekstowe, którego struktura jest pokazywana w dolnej części okna, poniżej widoku *Kod*. Drzewo to przedstawia listę wszystkich znaczników otaczających wybrany element (rysunek 3.9).



Rysunek 3.9.
Drzewo kontekstowe

Po zaznaczeniu elementu w widoku *Projekt* lub *Kod* widać w strukturze drzewa, jakie znaczniki go otaczają.

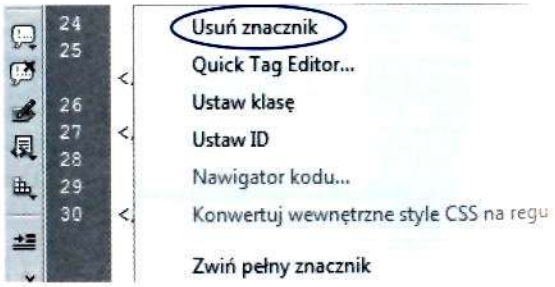
Korzystając z tego drzewa, zaznaczamy znaczniki z zawartością. Jeżeli chcemy zaznaczyć znacznik <p>, to jego wybranie w drzewie kontekstowym zaznaczy nie tylko sam element, ale również otaczające go znaczniki (rysunek 3.10).



Rysunek 3.10. Zaznaczanie tekstu w drzewie kontekstowym

W widoku *Projekt* zaznaczenie nie zmieniło się, natomiast w widoku *Kod* oprócz napisu zostały zaznaczone znaczniki <p> i <b>.

Jeżeli pogrubienie tekstu powinno zostać usunięte, wystarczy w drzewie kontekstowym kliknąć prawym przyciskiem myszy znacznik pogrubienia i wybrać opcję *Usuń znacznik* (rysunek 3.11).

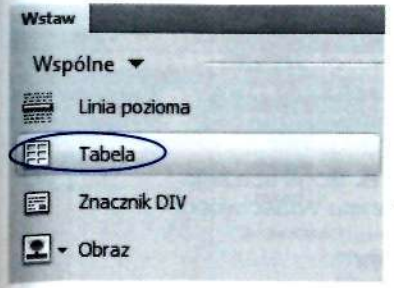


Rysunek 3.11. Usuwanie znacznika

Nie zawsze zaznaczanie elementów lub ustalanie położenia kursora w widoku *Projekt* daje prawidłowe wyniki. Dobrym rozwiązaniem jest sprawdzanie swoich działań w widoku *Kod*.

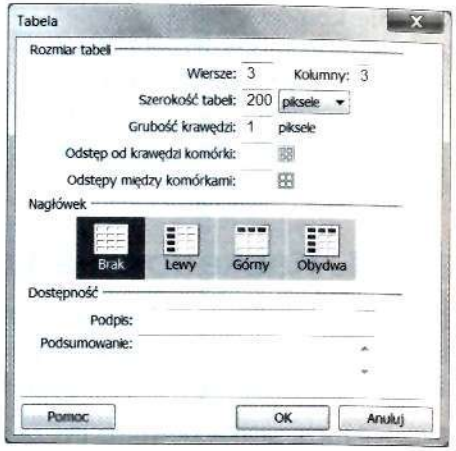
3.2.3. Tabele

Aby w widoku *Projekt* utworzyć tabelę, należy w panelu *Wstaw* wybrać z listy opcję *Wspólne* i na liście zaznaczyć opcję *Tabela* (rysunek 3.12). Zostanie otwarte okno, w którym można zdefiniować parametry tworzonej tabeli (rysunek 3.13).

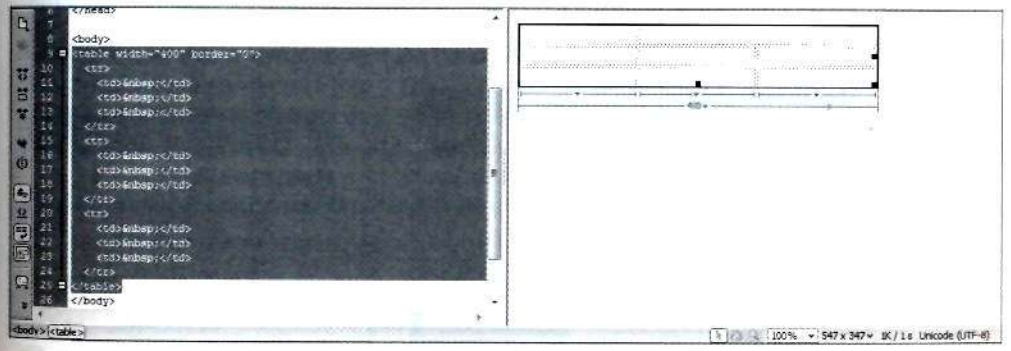


Rysunek 3.12. Projektowanie tabeli

Po zdefiniowaniu ustawień tabeli program automatycznie wygeneruje opisujący ją kod (rysunek 3.14). Jeżeli chcemy zmieniać parametry utworzonej tabeli, możemy to robić zarówno w widoku *Projekt* (przesuwając krawędzie tabeli), jak i w widoku *Kod* (zmieniając kod opisujący tabelę).

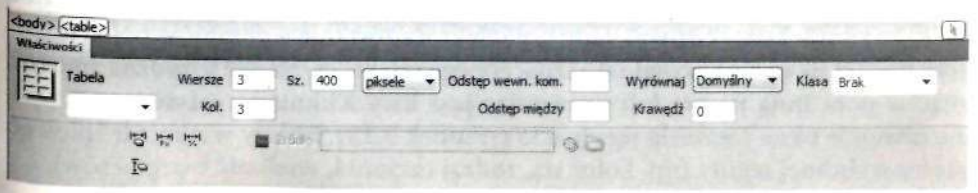


Rysunek 3.13. Ustawianie parametrów tworzonej tabeli



Rysunek 3.14. Tabela w widoku Projekt i wygenerowany kod w widoku Kod

Inną metodą modyfikowania utworzonej tabeli jest zmiana jej parametrów w panelu *Właściwości*. Panel *Właściwości* znajduje się w dolnej części okna, poniżej drzewa kontekstowego (rysunek 3.15); pokazuje właściwości aktualnie wybranego (zaznaczonego) elementu. Zaznaczenie może dotyczyć całej tabeli, wybranego wiersza, wybranej kolumny lub komórki. Zmiany, które zostaną wprowadzone w panelu *Właściwości*, są przenoszone do kodu strony.



Rysunek 3.15. Panel Właściwości

Scalenie komórek tabeli również jest definiowane w panelu *Właściwości*. Po zaznaczeniu scalanych komórek wystarczy w tym panelu kliknąć ikonę *Scalanie* (rysunek 3.16).



Rysunek 3.16. Ikona scalania komórek w panelu Właściwości

3.2.4. Kaskadowe arkusze stylów

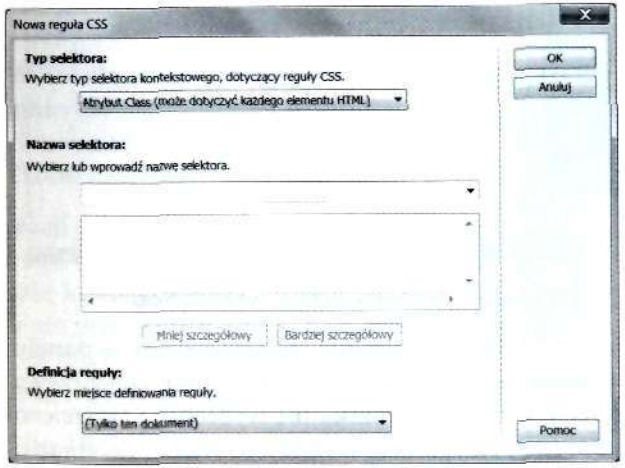
Do tworzenia arkuszy stylów CSS i zarządzania nimi służy panel *Style CSS* znajdujący się z prawej strony okna programu Dreamweaver (rysunek 3.17).

W tym panelu zawsze powinna być zaznaczona opcja *Wszystkie*, aby były dostępne wszystkie reguły zdefiniowane dla strony.

Dodawanie nowych stylów odbywa się za pomocą ikony *Nowa reguła CSS* , znajdującej się w dolnej części panelu. Po jej kliknięciu zostanie otwarte okno definiowania nowej reguły (rysunek 3.18).

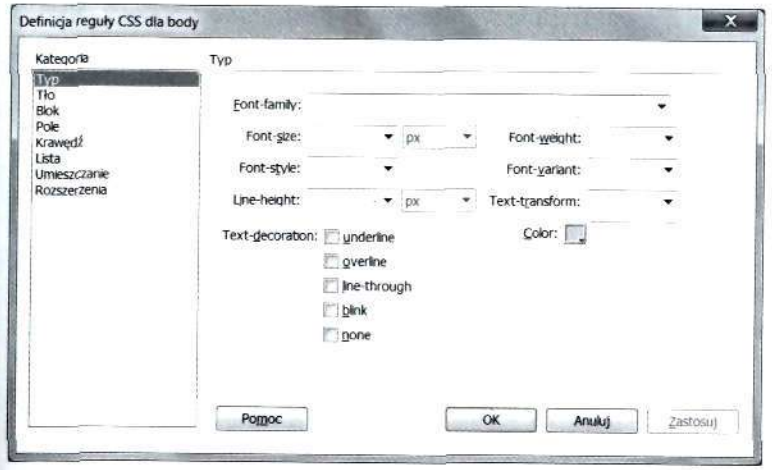


Rysunek 3.17. Panel Style CSS



Rysunek 3.18. Okno definiowania nowej reguły CSS

Pierwszym parametrem, który trzeba ustawić, jest *Typ selektora* (klasa, identyfikator, znacznik, znacznik kontekstowy). Gdy typ selektora zostanie wybrany, w polu *Nazwa selektora* pojawi się nazwa aktualnie zaznaczonego selektora. Nazwę tę można zmienić, wpisując w polu inną nazwę lub wybierając ją z listy. Kliknięcie przycisku *OK* spowoduje otwarcie okna *Definicja reguły CSS* (rysunek 3.19). Należy w nim zdefiniować parametry wybranej reguły (np. kolor tła, rodzaj czcionki, wielkość marginesów).



Rysunek 3.19. Okno definiowania reguły CSS

W oknie tym można ustawić wszystkie poznane wcześniej atrybuty dla wybranego elementu strony. Należy pamiętać, że reguła CSS zdefiniowana dla znacznika dotyczy wszystkich użytych w dokumencie znaczników tego typu.

Efektem zdefiniowania reguły CSS są zmiany w kodzie programu. W nagłówku zostanie dodany znacznik `<style type="text/css">` oraz znacznik, dla którego była definiowana reguła CSS, z listą zdefiniowanych atrybutów. Natomiast w panelu *Style CSS* pojawi się zdefiniowany element.

Dodawanie, usuwanie i edytowanie nowych reguł odbywa się zawsze w panelu *Style CSS* przy użyciu ikon znajdujących się w jego dolnej części. Ikona z ołówkiem służy do modyfikowania istniejącej reguły, natomiast ikona z koszem — do jej usunięcia.

3.2.5. Tworzenie szablonu strony w programie Dreamweaver

O tym, jak proste jest tworzenie strony internetowej z wykorzystaniem narzędzi programu Dreamweaver, przekonamy się na przykładzie budowania szablonu strony z nagłówkiem, stopką i dwoma kolumnami.

Przykład 3.1

Projektowanie każdej strony powinno zaczynać się od wyzerowania wszystkich marginesów i określenia, jak ma się zachować znacznik `<div>`, gdy wstawiony element nie będzie się w nim mieścił:

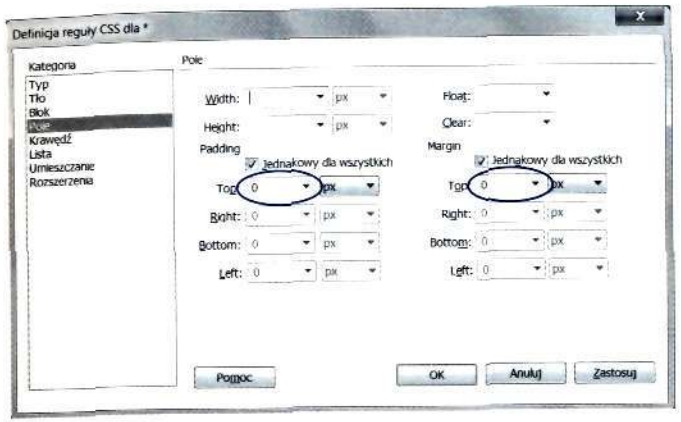
```
* { margin: 0px; padding: 0px; }
```

i

```
div { overflow: hidden; }
```

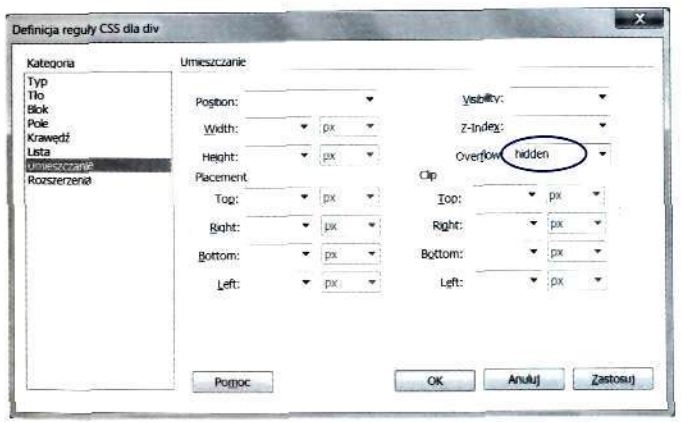
W tym celu w panelu *Style CSS* definiujemy poniższe reguły.

Klikamy ikonę *Nowa reguła CSS*, a następnie wybieramy *Typ selektora/Znacznik*. Nazwa selektora to *. Wszystko zatwierdzamy przyciskiem *OK*. W oknie *Definicja reguły CSS* wybieramy kategorię *Pole* i ustawiamy *Margin* i *Padding* na *0* (rysunek 3.20).



Rysunek 3.20. Zerowanie parametrów margin i padding

Następnie tą samą metodą definiujemy regułę dla znacznika `<div>`. Tym razem w oknie *Definicja reguły CSS* jako nazwę selektora wpisujemy `div` (lub wybieramy *div* z listy). Wybieramy kategorię *Umieszczanie* i ustawiamy parametr *Overflow* na *hidden* (rysunek 3.21).

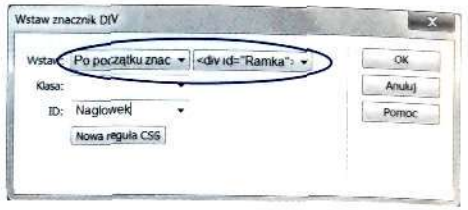


Rysunek 3.21. Definiowanie znacznika `<div>`

Następnym etapem pracy będzie zdefiniowanie ramki zewnętrznej, określającej rozmiar wyświetlanej strony, i umieszczenie w niej nagłówka, dwóch kolumn obok siebie oraz stopki. Operacje te zostaną wykonane przez zdefiniowanie i odpowiednie sformatowanie znaczników `<div>`.

Wybieramy okno *Projekt* i przystępujemy do wstawiania znaczników. Polega to na zaznaczeniu w panelu *Wstaw* opcji *Wspólne* i znalezieniu na liście znacznika *div*. Wyświetli się okno *Wstaw znacznik DIV*, w którym musimy ustawić następujące parametry:

Wstaw/W punkcie wstawiania, *Id/Ramka* (można wprowadzić dowolną nazwę identyfikatora).

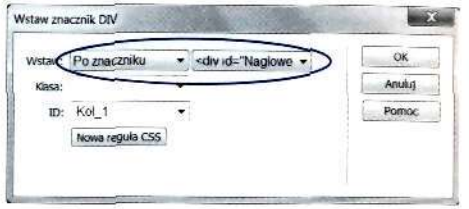


Rysunek 3.22.
Definiowanie znacznika Naglowek

W widoku *Projekt* zostanie umieszczony zdefiniowany znacznik. Należy zwrócić uwagę, że tekst, który jest wyświetlony wewnątrz znacznika, służy tylko do identyfikowania położenia tego znacznika i w dalszej pracy nad stroną trzeba ten tekst usunąć. Do utworzonego znacznika `<div>` wstawimy następne znaczniki `<div>`. Tworząc znacznik, który będzie nagłówkiem strony, definiujemy jego parametry — *Wstaw: Po początku znacznika*; z listy wybieramy *Ramka* oraz *ID: Naglowek* (rysunek 3.22).

Następnie zdefiniujemy dwie kolumny, które będą się znajdowały pod nagłówkiem.

Kolumna pierwsza będzie umieszczona poniżej nagłówka, dlatego definiujemy parametr *Wstaw: Po znaczniku* i z listy wybieramy *Naglowek* oraz *ID: Kol_1* (rysunek 3.23). Dla kolumny drugiej wybieramy *Wstaw: Po znaczniku* i z listy *Kol_1* oraz *ID: Kol_2*. Dla stopki parametry definiujemy podobnie — *Wstaw: Po znaczniku* i z listy wybieramy *Kol_2* oraz *ID: Stopka*.



Rysunek 3.23.
Definiowanie znacznika Kol_1

Po tych działaniach w widoku *Projekt* zobaczymy blok o nazwie *Ramka*, a wewnątrz niego cztery bloki, *Naglowek*, *Kol_1*, *Kol_2*, *Stopka*, leżące jeden pod drugim.

Wygenerowany fragment kodu strony wygląda następująco:

Przykład 3.2

```
<body>
<div id="Ramka">
  <div id="Naglowek">Tu znajdzie się zawartość  id "Naglowek"</div>
  <div id="Kol_1">Tu znajdzie się zawartość  id "Kol_1"</div>
  <div id="Kol_2">Tu znajdzie się zawartość  id "Kol_2"</div>
  <div id="Stopka">Tu znajdzie się zawartość  id "Stopka"</div>
</div>
</body>
```

Następnym krokiem będzie zdefiniowanie reguł CSS dla utworzonych znaczników `<div>`. Początkowo w celu rozróżniania bloków można każdemu z nich nadać inny kolor tła.

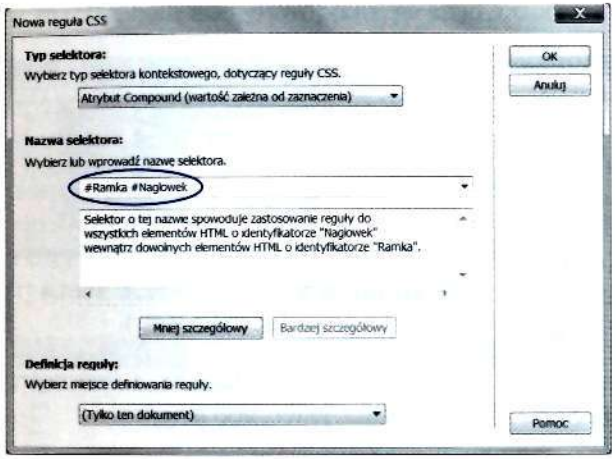
Aby utworzyć regułę dla znacznika *Ramka*, klikamy go w drzewie kontekstowym (znajdującym się powyżej panelu *Właściwości*) (rysunek 3.24), a następnie w panelu *Style CSS* klikamy ikonę tworzenia nowej reguły CSS dla znacznika *Ramka* .



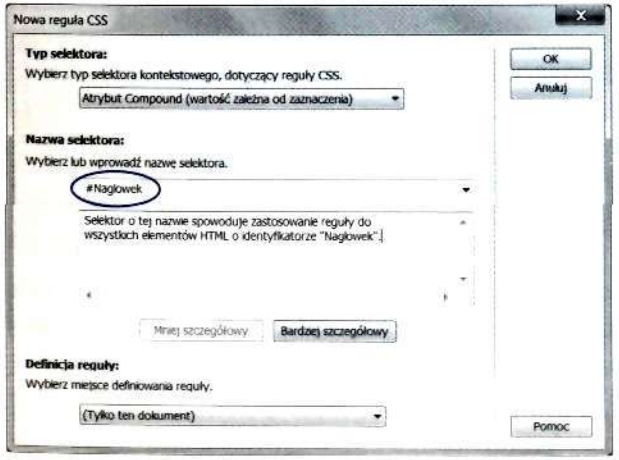
Rysunek 3.24. Drzewo kontekstowe — wybrany znacznik Ramka

W oknie *Definicja reguły CSS* w kategorii *Tło* wybieramy dowolny kolor tła (*background-color*). Podobnie definiujemy reguły CSS dla pozostałych znaczników `<div>`.

Przy definiowaniu reguł CSS dla kolejnych znaczników `<div>`, które są umieszczone wewnątrz bloku *Ramka*, w oknie *Nowa reguła CSS* nazwa selektora pojawi się w kontekście (rysunek 3.25). Należy ten kontekst usunąć, klikając przycisk *Mniej szczegółowy* tyle razy, ile potrzeba, by pozostała tylko nazwa selektora (rysunek 3.26).



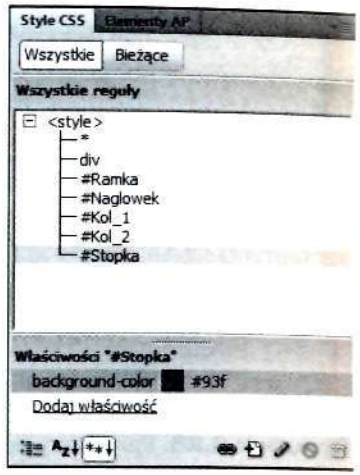
Rysunek 3.25. Nazwa selektora wyświetlona w kontekście



Rysunek 3.26. Nazwa selektora po użyciu przycisku Mniej szczegółowy

Po wykonaniu tych działań panel *Style CSS* zawiera zdefiniowane reguły (rysunek 3.27).

Kod programu został natomiast uzupełniony o definicję reguł w wewnętrznym arkuszu stylów.



Rysunek 3.27. Panel Style CSS po zdefiniowaniu reguł CSS

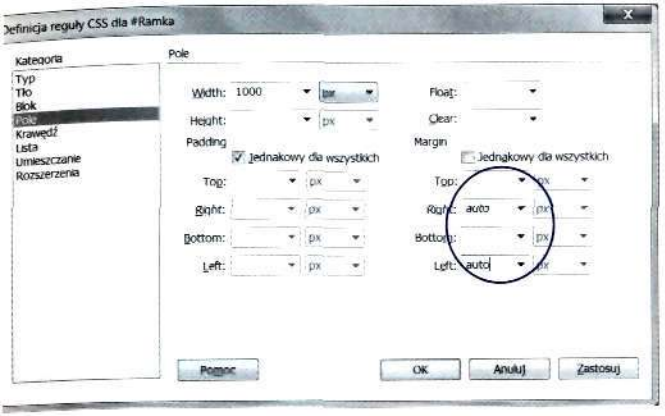
Przykład 3.3

```
<style type="text/css">
* {
  margin: 0px;
  padding: 0px;
}
div {
  overflow: hidden;
}
#Ramka {
  background-color: #CCC;
}
#Naglowek {
  background-color: #fcf;
}
#Kol_1 {
  background-color: #3ff;
}
#Kol_2 {
  background-color: #cf3;
}
#Stopka {
  background-color: #93f;
}
</style>
```

Przygotowywany szablon ma stałą szerokość. Wcześniej wyjaśniliśmy, że standardowo strony są przygotowywane dla ekranów o rozdzielczości 1024 px (szerokość) — dla takich rozdzielczości szerokość strony powinna wynosić 1000 px.

Dla tworzonego szablonu uzyskamy określoną szerokość przez zdefiniowanie szerokości bloku *Ramka*, który ogranicza całą zawartość strony.

W tym celu wchodzimy do panelu edytowania reguły zdefiniowanej dla selektora *Ramka* i w kategorii *Pole* parametrowi *Width* (ang. — „szerokość") przypisujemy wartość *1000 px*. Aby blok *Ramka* zawsze był wyświetlany na środku okna przeglądarki, ustawiamy marginesy lewy i prawy na wartość *auto* (rysunek 3.28). W efekcie niezależnie od rozmiaru okna przeglądarki strona zawsze będzie wyświetlana na jego środku.



Rysunek 3.28. Parametry ustawione dla bloku Ramka

Pozostało ustawienie kolumn jedna obok drugiej. W tym celu należy określić rozmiary kolumn, a następnie, wykorzystując parametr *Float*, ustawić je obok siebie.

Edytujemy regułę zdefiniowaną dla kolumny *Kol_1* i w kategorii *Pole* określamy jej szerokość np. na 300 px. W widoku *Projekt* widać, że kolumna została zwężona. Podobnie postępujemy w przypadku kolumny *Kol_2*, ustawiając szerokość np. na 700 px (suma szerokości kolumn nie może przekroczyć szerokości bloku zewnętrznego *Ramka*).

Następnym krokiem jest zdefiniowanie w kategorii *Pole* dla bloków *Kol_1* i *Kol_2* parametru *Float=left*. Aby blok *Stopka* pozostał pod kolumnami, trzeba ustawić dla niego parametr *Clear=both*.

Tak przygotowany szablon strony można obejrzeć w przeglądarce internetowej, klikając ikonę znajdującą się w górnej części okna programu. Następnie można przystąpić do wypełniania szablonu treścią.

3.2.6. Zewnętrzny arkusz stylów CSS

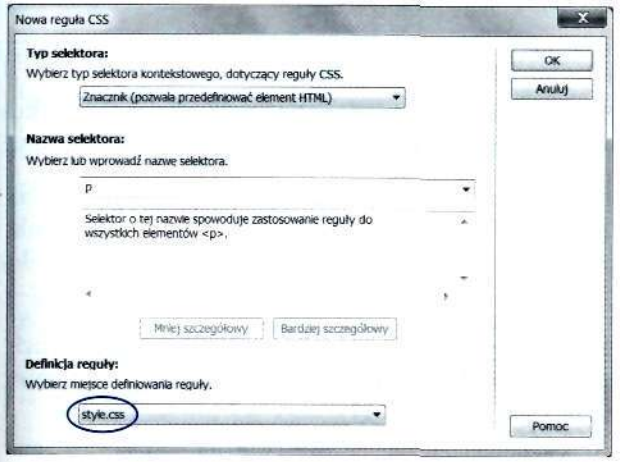
Jeżeli tworzona witryna internetowa składa się z wielu stron, których wygląd będzie taki sam (lub bardzo podobny), to zbędne jest definiowanie stylów CSS w każdym z plików. W programie Dreamweaver istnieje możliwość tworzenia zewnętrznych stylów CSS, zapisywanych w oddzielnych plikach i dołączanych do dokumentów HTML.

Aby utworzyć zewnętrzny arkusz stylów CSS, trzeba po wybraniu z menu opcji *Plik/Nowy* zaznaczyć *Typ strony/CSS*. Powstanie nowy plik, który można zapisać w tym samym folderze, gdzie znajdują się pliki tworzonej witryny, z rozszerzeniem *css* (np. *style.css*). Dobrym zwyczajem jest umieszczanie plików CSS w oddzielnym folderze *style*.

Następnie otwarty plik HTML należy powiązać z plikiem zawierającym zewnętrzny arkusz stylów CSS.

W tym celu musimy w panelu *Style CSS* kliknąć ikonę *Dołącz arkusz stylów*, znajdującą się w panelu obok ikony *Nowa reguła CSS*. W otwartym oknie określamy nazwę

dołączanego pliku. W wyniku tego do pliku HTML zostanie dołączony plik zawierający zewnętrzny arkusz stylów, co zostanie uwidocznione w panelu *Style CSS*. Teraz wszystkie nowe reguły definiowane dla projektowanej strony będą umieszczane w pliku zawierającym zewnętrzny arkusz stylów. Widoczne jest to w oknie tworzenia nowej reguły (opcja *Wybierz miejsce definiowania reguły* — rysunek 3.29) oraz w panelu *Style CSS* po rozwinięciu drzewa kontekstowego (rysunek 3.30).



Rysunek 3.29. Umieszczanie tworzonych reguł w zewnętrznym arkuszu stylów



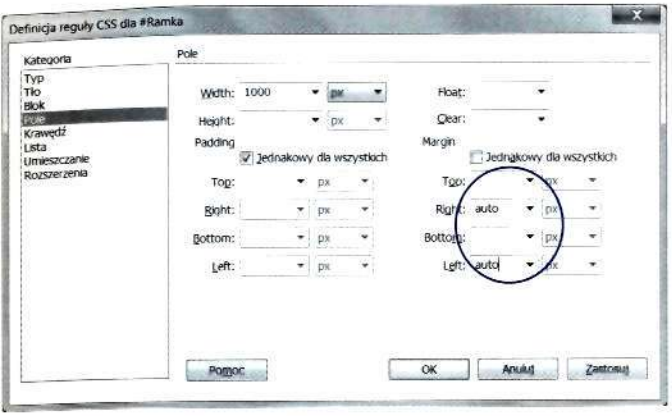
Rysunek 3.30. Reguły widoczne w panelu Style CSS

Po tych działaniach w kodzie programu, w miejscu, gdzie do tej pory były definiowane style, pozostaje tylko jeden wpis:

```
<link href="style.css" rel="stylesheet" type="text/css" />
```

Jest to odnośnik do pliku, w którym zostały zapisane zdefiniowane reguły.

Po utworzeniu kolejnych stron wystarczy dołączyć do nich istniejący zewnętrzny arkusz stylów, aby wszystkie elementy, dla których zostały zdefiniowane reguły CSS, wyglądały tak samo.



Rysunek 3.28. Parametry ustawione dla bloku Ramka

Pozostało ustawienie kolumn jedna obok drugiej. W tym celu należy określić rozmiary kolumn, a następnie, wykorzystując parametr *Float*, ustawić je obok siebie.

Edytujemy regułę zdefiniowaną dla kolumny *Kol_1* i w kategorii *Pole* określamy jej szerokość np. na 300 px. W widoku *Projekt* widać, że kolumna została zwężona. Podobnie postępujemy w przypadku kolumny *Kol_2*, ustawiając szerokość np. na 700 px (suma szerokości kolumn nie może przekroczyć szerokości bloku zewnętrznego *Ramka*).

Następnym krokiem jest zdefiniowanie w kategorii *Pole* dla bloków *Kol_1* i *Kol_2* parametru *Float=left*. Aby blok *Stopka* pozostał pod kolumnami, trzeba ustawić dla niego parametr *Clear=both*.

Tak przygotowany szablon strony można obejrzeć w przeglądarce internetowej, klikając ikonę znajdującą się w górnej części okna programu. Następnie można przystąpić do wypełniania szablonu treścią.

3.2.6. Zewnętrzny arkusz stylów CSS

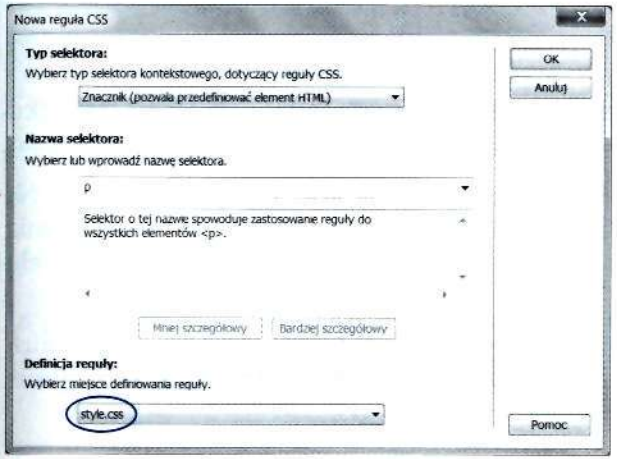
Jeżeli tworzona witryna internetowa składa się z wielu stron, których wygląd będzie taki sam (lub bardzo podobny), to zbędne jest definiowanie stylów CSS w każdym z plików. W programie Dreamweaver istnieje możliwość tworzenia zewnętrznych stylów CSS, zapisywanych w oddzielnych plikach i dołączanych do dokumentów HTML.

Aby utworzyć zewnętrzny arkusz stylów CSS, trzeba po wybraniu z menu opcji *Plik/Nowy* zaznaczyć *Typ strony/CSS*. Powstanie nowy plik, który można zapisać w tym samym folderze, gdzie znajdują się pliki tworzonej witryny, z rozszerzeniem *css* (np. *style.css*). Dobrym zwyczajem jest umieszczanie plików CSS w oddzielnym folderze *style*.

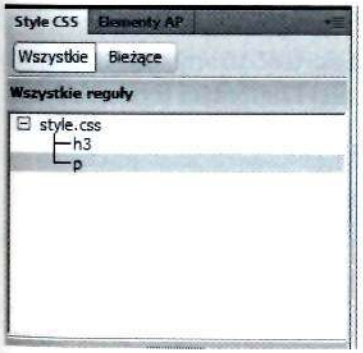
Następnie otwarty plik HTML należy powiązać z plikiem zawierającym zewnętrzny arkusz stylów CSS.

W tym celu musimy w panelu *Style CSS* kliknąć ikonę *Dołącz arkusz stylów*, znajdującą się w panelu obok ikony *Nowa reguła CSS*. W otwartym oknie określamy nazwę

dołączanego pliku. W wyniku tego do pliku HTML zostanie dołączony plik zawierający zewnętrzny arkusz stylów, co zostanie uwidocznione w panelu *Style CSS*. Teraz wszystkie nowe reguły definiowane dla projektowanej strony będą umieszczane w pliku zawierającym zewnętrzny arkusz stylów. Widoczne jest to w oknie tworzenia nowej reguły (opcja *Wybierz miejsce definiowania reguły* — rysunek 3.29) oraz w panelu *Style CSS* po rozwinięciu drzewa kontekstowego (rysunek 3.30).



Rysunek 3.29. Umieszczanie tworzonych reguł w zewnętrznym arkuszu stylów



Rysunek 3.30. Reguły widoczne w panelu Style CSS

Po tych działaniach w kodzie programu, w miejscu, gdzie do tej pory były definiowane style, pozostaje tylko jeden wpis:

```
<link href="style.css" rel="stylesheet" type="text/css" />
```

Jest to odnośnik do pliku, w którym zostały zapisane zdefiniowane reguły.

Po utworzeniu kolejnych stron wystarczy dołączyć do nich istniejący zewnętrzny arkusz stylów, aby wszystkie elementy, dla których zostały zdefiniowane reguły CSS, wyglądały tak samo.

3.2.7. Media w CSS

Wykorzystując możliwości arkuszy stylów, można utworzyć stronę internetową, która będzie wyglądała różnie na różnych typach urządzeń (dotyczy to tylko CSS3).

W tym celu należy utworzyć oddzielne pliki ze stylami CSS dla każdego typu urządzenia, na którym zamierzamy wyświetlać projektowaną stronę. Pliki te będą automatycznie uruchamiane przez przeglądarkę w zależności od urządzenia, na którym w danym momencie będzie otwierana strona.

Przykład 3.4
Aby zaprojektowaną stronę internetową można było wyświetlać w telefonie komórkowym, na tablecie i na komputerze, tworzymy trzy pliki CSS: *mobil.css*, *tablet.css*, *komp.css*.

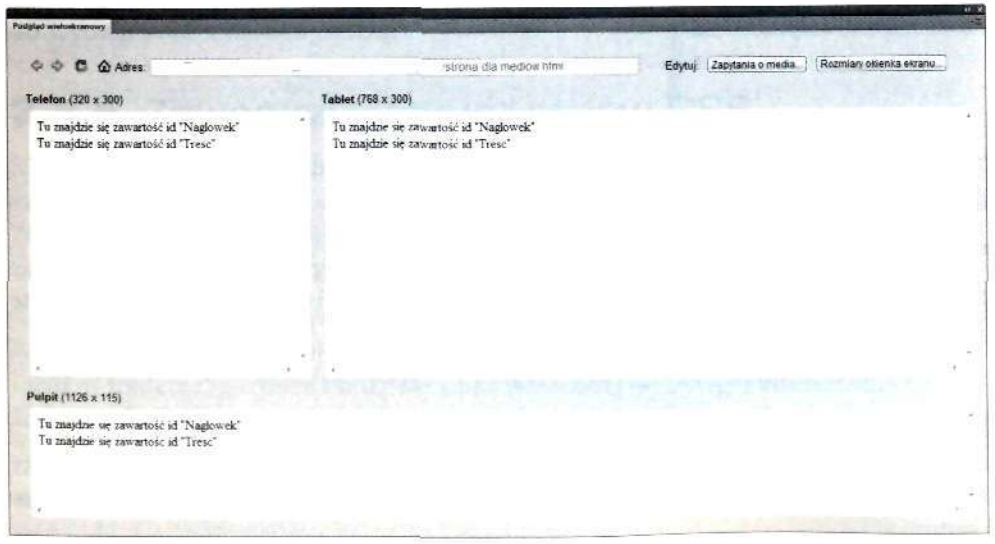
Tworzymy również plik HTML, który będzie zawierał projekt strony internetowej. Na razie nie ma w nim zdefiniowanych żadnych reguł. Aby sprawdzić efekt wyświetlania strony na różnego typu urządzeniach, należy umieścić w tym pliku przykładowe elementy, np. dwa znaczniki `<div>`.

Do pliku HTML dołączymy trzy utworzone pliki CSS — będziemy definiowali w nich kolejne reguły. W tym celu klikamy przycisk *Tryb wieloekranowy* dostępny w górnej części okna programu na pasku *Dokument* (rysunek 3.31).



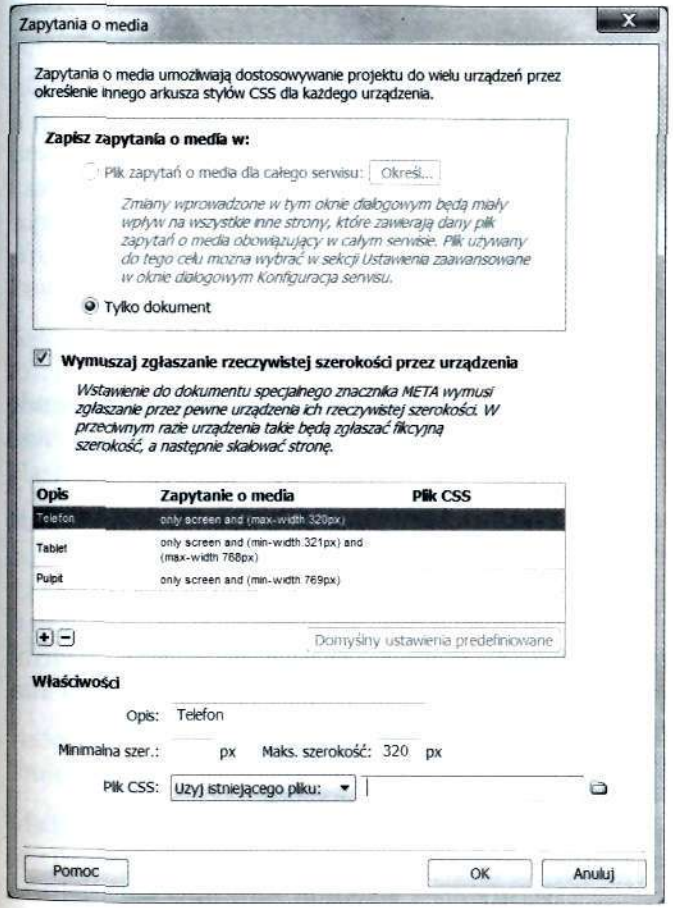
Rysunek 3.31. Wybór opcji Tryb wieloekranowy

Po jego kliknięciu zostanie otwarte okno, które pokaże zawartość strony na trzech podstawowych urządzeniach: telefonie, tablecie, pulpicie (ekranie) komputera (rysunek 3.32).



Rysunek 3.32. Podgląd wieloekranowy

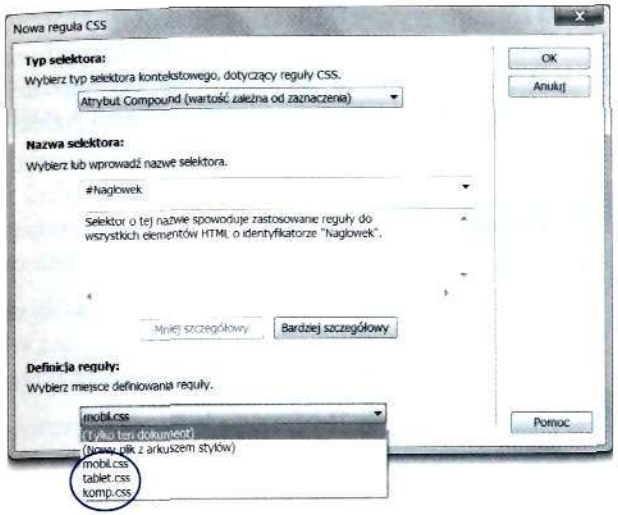
Po kliknięciu przycisku *Zapytania o media* zostanie wyświetlone kolejne okno, w którym można zdefiniować urządzenia przeznaczone do wyświetlania strony. Można również kliknąć przycisk *Domyślne ustawienia predefiniowane*, w wyniku czego otrzymamy listę trzech podstawowych urządzeń zdefiniowanych w programie, do których należy przypisać odpowiednie pliki CSS (rysunek 3.33).



Rysunek 3.33. Okno wyboru mediów

Aby przypisać do telefonu plik *mobil.css*, wybieramy na liście *Telefon*, zmieniamy opcję *Utwórz nowy plik:* na *Użyj istniejącego pliku:* i znajdujemy plik. Podobnie postępujemy w przypadku pozostałych urządzeń. Gdy zatwierdzimy wprowadzone ustawienia, w panelu *Style CSS* zostaną pokazane dołączone do dokumentu trzy pliki z arkuszami stylów.

Przyjmujemy, że pierwszy utworzony na stronie blok powinien wyglądać inaczej na każdym z urządzeń. Dla tego bloku klikamy ikonę *Nowa reguła CSS* i w otwartym oknie z listy rozwijanej *Wybierz miejsce definiowania reguły* wybieramy plik, w którym zostanie zapisana definiowana reguła (rysunek 3.34). Po zatwierdzeniu wyboru określamy parametry definiowanego elementu.

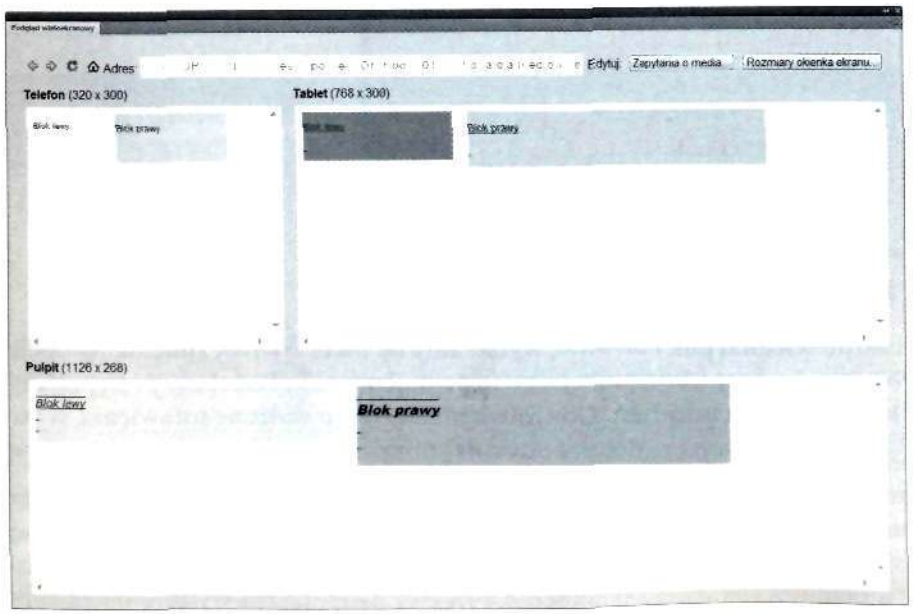


Rysunek 3.34. Wybór miejsca definiowania reguły

Gdy zapiszemy regułę bloku `<div>`, w pliku *mobil.css* ponownie definiujemy regułę dla tego samego bloku, ale z innymi ustawieniami, i zapisujemy ją w pliku *tablet.css*. Tak samo postępujemy z plikiem *komp.css*.

Gdy zdefiniujemy reguły dla bloku pierwszego, możemy przystąpić do ich definiowania dla bloku drugiego.

Przykładowy wynik tych działań został pokazany na rysunku 3.35.



Rysunek 3.35. Sposób wyświetlania strony na różnych urządzeniach

3.2.8. Umieszczanie multimediów na stronie

Aby utworzona strona internetowa była stroną multimedialną, należy umieścić na niej animację, sekwencję wideo lub wykorzystać inne media dostępne w internecie.

Animacje Flash

Pliki otrzymane jako animacje Flash są zapisywane w formacie SWF. Przy użyciu wtyczki Flash Player można je odtwarzać w dowolnej przeglądarce.

Aby wstawić na stronę animację Flash, należy w panelu *Wstaw* wybrać z listy *Wspólne*, a następnie *Multimedia* i z listy wszystkich multimediów wybrać opcję *SWF*. Zostanie otwarte okno katalogów, w którym musimy znaleźć plik zawierający animację Flash i zatwierdzić wybór. Wstawionej animacji nie można obejrzeć w widoku *Projekt*. Należy zapisać plik z projektem strony, akceptując wszystkie ostrzeżenia, a następnie otworzyć stronę w dowolnej przeglądarce internetowej. W panelu *Właściwości* można modyfikować parametry wstawionej animacji, dostosowując ją do potrzeb strony.

Sekwencje wideo

Korzystając z odpowiedniego oprogramowania, sekwencje wideo można wyeksportować do formatu FLV. Jest to format typu Flash Video, który pozwala (podobnie jak animacje Flash) uruchamiać wideo w dowolnej przeglądarce za pomocą wtyczki Adobe Flash Player.

Podczas umieszczania na stronie sekwencji wideo postępujemy podobnie jak przy animacji Flash. W panelu *Wstaw* wybieramy z listy *Wspólne*, a następnie *Multimedia* i z listy wszystkich multimediów wybieramy opcję *FLV*. Zostanie otwarte okno, gdzie należy wskazać plik FLV, który będzie odtwarzany na stronie. Można również ustalić rozmiar okna, w którym będzie pokazywane wideo, oraz jedną z dostępnych w programie Dreamweaver skórek dla odtwarzacza wideo.

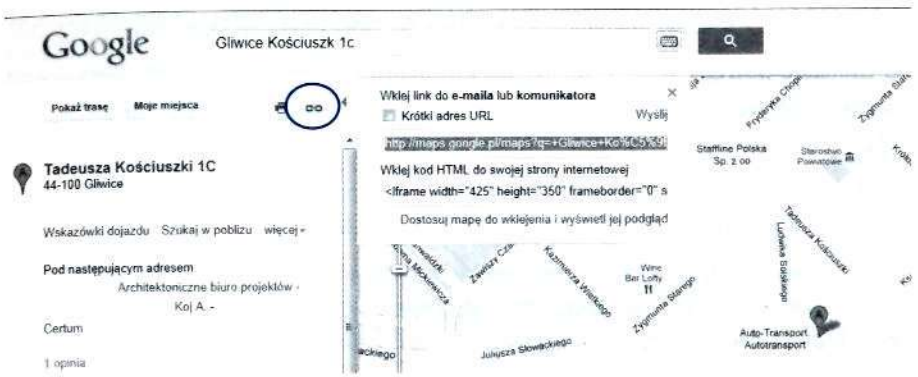
Podobnie jak w przypadku animacji Flash, filmu wideo nie można zobaczyć w widoku *Projekt*.

Multimedia zewnętrzne

Jeżeli pozwalają na to dostawcy, na tworzonej stronie internetowej można umieszczać zewnętrzne multimedia dostępne w internecie.

Przykładem serwisu, z którego zasobów można skorzystać, jest *www.maps.google.pl*.

Po wybraniu lokalizacji, którą chcielibyśmy umieścić na stronie, wystarczy kliknąć przycisk *Link* (rysunek 3.36), a następnie skopiować z pola tekstowego *Wklej kod HTML do swojej strony internetowej* umieszczony tam kod.



Rysunek 3.36. Okno serwisu maps.google.pl

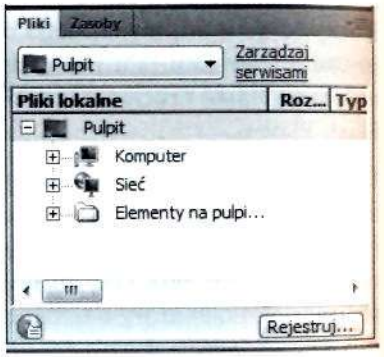
3.2.9. Pliki lokalne

Strona internetowa często zawiera różnego rodzaju zdjęcia, filmy wideo, pliki do pobrania i inne materiały, które są wykorzystywane podczas pracy. Program Dreamweaver ma panel *Pliki*, który służy do zarządzania witryną internetową (rysunek 3.37). Znajduje się on w dolnej części okna programu z prawej strony.

W panelu tym można definiować własne serwisy, na których będą umieszczane elementy strony internetowej, edytować fragmenty strony oraz publikować witrynę internetową na serwerze.

Do zarządzania serwisami służy przycisk *Zarządzaj serwisami*. Po jego kliknięciu możemy utworzyć nowy serwis internetowy, który będzie przypisany do projektu tworzonej witryny internetowej (rysunek 3.38).

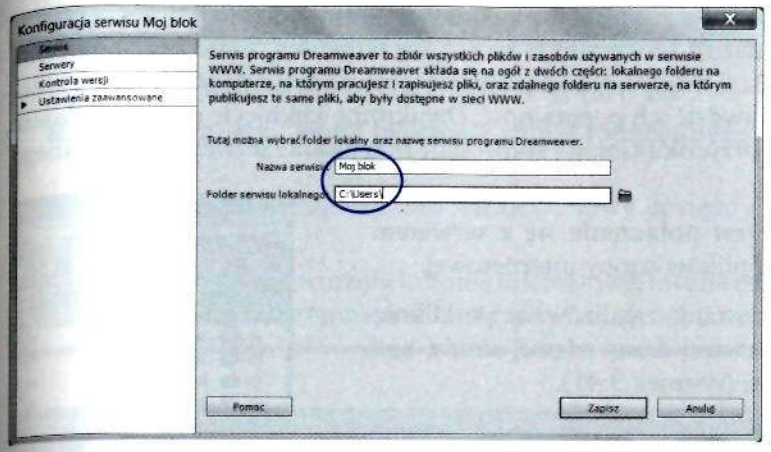
Po kliknięciu przycisku *Nowy* zostanie otwarte kolejne okno, w którym należy podać nazwę tworzonego serwisu oraz folder przechowywania zasobów tworzonej witryny (rysunek 3.39).



Rysunek 3.37. Panel Pliki



Rysunek 3.38. Okno tworzenia nowego serwisu

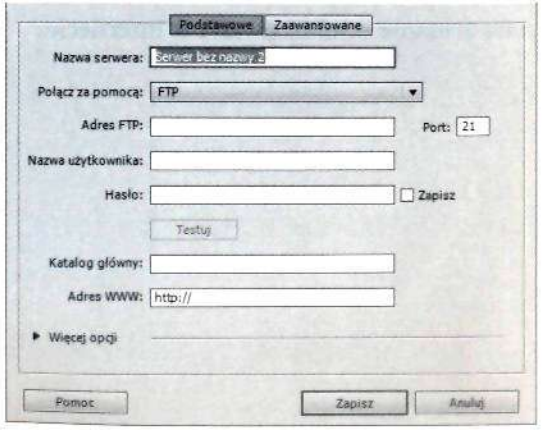


Rysunek 3.39. Okno konfigurowania serwisu strony internetowej

Gdy klikniemy przycisk *Zapisz*, w oknie *Zarządzaj serwisami* pojawi się projektowany serwis. Serwisów może być kilka, jeżeli pracujemy nad tworzeniem kilku witryn internetowych. Gdy w oknie *Zarządzaj serwisami* klikniemy przycisk *Gotowe*, w panelu *Pliki* pojawi się tylko zawartość serwisu, nad którym obecnie pracujemy. Klikając w tym panelu przycisk *Mój serwis*, można ponownie przełączyć się do dowolnego folderu dostępnego na komputerze użytkownika.

3.2.10. Publikowanie witryny internetowej

Korzystając z programu Dreamweaver, utworzoną witrynę można opublikować na dostępnym serwerze stron internetowych. Aby to zrobić, trzeba najpierw określić miejsce, gdzie znajdują się pliki przeznaczone do umieszczenia na serwerze. W naszym przypadku jest to adres utworzonego serwisu internetowego. Należy w panelu *Pliki* kliknąć przycisk znajdujący się nad okienkiem *Pliki lokalne*, a następnie opcję *Zarządzaj serwisami*. Pojawi się znane już okno *Zarządzaj serwisami* z listą zdefiniowanych serwisów. Trzeba w nim wybrać nazwę aktualnego serwisu i kliknąć przycisk *Edycja*. W otwartym oknie *Konfiguracja serwisu*, w panelu z lewej strony, musimy wybrać opcję *Serwery*, a następnie w głównym oknie kliknąć przycisk +. Otworzy się okno konfigurowania ustawień serwera FTP. Można w nim skonfigurować serwer, na którym będzie publikowana utworzona strona internetowa (rysunek 3.40).



Rysunek 3.40. Konfigurowanie serwera FTP

Do odpowiednich pól należy wpisać otrzymane od dostawcy miejsca na serwerze parametry, takie jak: nazwa serwera FTP, nazwa użytkownika, hasło oraz położenie katalogu głównego naszej strony internetowej. Po wprowadzeniu danych można, klikając przycisk *Testuj*, sprawdzić ich poprawność. Dwukrotne kliknięcie przycisku *Zapisz*, a następnie kliknięcie przycisku *Gotowe* spowoduje przypisanie serwera do wybranego serwisu lokalnego.

Następnym krokiem jest połączenie się z serwerem i umieszczenie na nim plików strony internetowej.



Rysunek 3.41.
Łączenie z serwerem

Łączenie z serwerem zostanie zrealizowane po kliknięciu w panelu *Pliki* pierwszej ikony z lewej strony, nad okienkiem *Pliki lokalne* (rysunek 3.41).

Po pomyślnym połączeniu z serwerem ikona łączenia zmieni wygląd. Jeżeli połączenia nie uda się zrealizować, to najprawdopodobniej dane zostaną wprowadzone nieprawidłowo.

W oknie *Pliki lokalne* domyślnie jest wyświetlana zawartość komputera użytkownika. W celu przełączenia się na serwer należy kliknąć przycisk *Widok lokalny* i z wyświetlonej listy wybrać *Serwer zdalny*. W oknie pojawi się zawartość dostępnego katalogu głównego znajdującego się na serwerze. W obrębie miejsca przydzielonego na serwerze można wykonywać standardowe operacje dotyczące plików i folderów.



Rysunek 3.42.
Umieszczanie plików strony na serwerze

Aby umieścić na serwerze pliki utworzonego serwisu, trzeba ponownie wybrać i wyświetlić w oknie *Pliki lokalne* zawartość projektowanej strony. Następnie trzeba zaznaczyć pliki przeznaczone do przeniesienia na serwer i kliknąć ikonę ze strzałką skierowaną do góry (rysunek 3.42), co spowoduje wysłanie strony na serwer zdalny. Strona zostanie opublikowana w internecie.

PYTANIA I POLECENIA KONTROLNE

1. Wymień podstawowe cechy edytorów WYSIWYG.
2. Wymień zalety i wady tworzenia strony internetowej za pomocą edytora WYSIWYG.
3. Czym charakteryzuje się tryb wieloekranowy dostępny w aplikacji Adobe Dreamweaver?
4. W jaki sposób na tworzonej stronie internetowej można umieszczać zewnętrzne multimedia dostępne w internecie?

ZADANIE

Za pomocą programu Dreamweaver należy utworzyć witrynę internetową na temat organizowanych w szkole wycieczek tematycznych. W celu ujednolicenia układu informacji na stronie należy opracować szablon strony. Każdy z uczestników projektu powinien umieścić w przygotowanym szablonie opracowany wcześniej materiał na temat wybranej wycieczki. W programie Dreamweaver należy również utworzyć zewnętrzny arkusz stylów zawierający definicję stylów dla wszystkich elementów strony internetowej i dołączyć go do dokumentów HTML opracowanych przez uczestników projektu.

4

Zasady projektowania strony internetowej

Strony internetowe tworzone za pomocą języka HTML wzbogaconego o kaskadowe arkusze stylów oraz odpowiednią grafikę mogą być ciekawym i przydatnym dla nas źródłem informacji. Podczas ich projektowania trzeba zwrócić uwagę na następujące aspekty:

- **Zawartość** — informacje, które są umieszczane na stronie, powinny być rzetelne, odnosić się do tematu strony i być skierowane do odpowiedniego odbiorcy.
- **Użyteczność** — informacja umieszczona na stronie ma być zrozumiała, a nawigacja prosta i niezawodna. Użytkownik nie powinien mieć problemów z odnalezieniem informacji.
- **Wygląd** — tekst i elementy graficzne muszą tworzyć spójną całość. Kolejne strony witryny internetowej powinny zachowywać ten sam styl.
- **Widoczność** — strona powinna być łatwa do znalezienia przy użyciu wyszukiwarek i katalogów stron internetowych.

4.1. Projektowanie stron

Przed rozpoczęciem tworzenia witryny internetowej należy przygotować jej projekt.

Proces projektowania strony internetowej jest jednym z ważniejszych stadiów jej tworzenia. Składa się z trzech etapów:

- poznawania,
- poszukiwania,
- implementacji.

Poznawanie
Na tym etapie powinniśmy porozmawiać z klientem zlecającym utworzenie strony. Warto dowiedzieć się, kim on jest, jakiego rodzaju działalność prowadzi, czym zajmuje się jego firma. Jeszcze wcześniej powinniśmy zapoznać się z dziedziną gospodarki, z którą związany jest klient. W trakcie spotkania należy ustalić, jaki cel klient chce osiągnąć dzięki witrynie, jaki typ informacji powinien znaleźć się na stronach, do kogo będzie skierowana witryna. Jeżeli uda się nam dobrze zrozumieć potrzeby klienta, łatwiej będzie nam właściwie zaprojektować witrynę.

Poszukiwanie
Na tym etapie powinniśmy zebrać wszystkie informacje uzyskane od klienta, przeanalizować je i zastanowić się, w jaki sposób je wykorzystać. Należy postawić się na miejscu użytkownika witryny i odpowiedzieć na pytania:

- Dlaczego jestem na tej stronie?
- Jeżeli interesuje mnie jakiś przedmiot, usługa lub działalność, to co chciałbym na ich temat wiedzieć?
- Gdzie mógłbym znaleźć wyczerpującą informację na ten temat?

Po zebraniu informacji i sporządzeniu listy wszystkich elementów, które powinny się znaleźć w projektowanej witrynie, dobrze jest podzielić te elementy na grupy i podgrupy. W procesie tworzenia projektu zawartość grup i podgrup będzie z pewnością wielokrotnie zmieniana. Jeżeli uda się je ułożyć w odpowiedniej kolejności, będzie to kolejność obowiązująca w projektowanym panelu nawigacji. Opcji wyboru nie może być za dużo, jednak ukrycie informacji zbyt głęboko może spowodować, że użytkownik do niej nie dotrze. Optymalnego rozwiązania nie ma. Należy starać się, żeby internauta miał dostęp do informacji w sposób jak najbardziej intuicyjny.

Implementacja
Ten etap pracy to tworzenie układu witryny, na razie na kartce papieru. Pozwoli to na uniezależnienie się od możliwości technicznych przeglądarek internetowych, języka HTML czy arkuszy CSS i skupienie się na tym, jak ma wyglądać ostateczny efekt pracy, czyli witryna internetowa.

Można wykonać kilka szkiców układów strony, wyznaczyć obszary odpowiadające jej elementom, porozmieszczać komponenty, zaplanować kolory pierwszego planu oraz tła, jak również zaprojektować przyciski menu i suwaki.

Dobra strona
Są dwa wyznaczniki tego, czy strona internetowa została zaprojektowana właściwie.

Pierwszy to użyteczność strony — to, czy informacja jest prezentowana w sposób efektywny i czy witryna jest funkcjonalna.

Drugi to estetyka strony — ciekawy efekt wizualny, wysoka wartość artystyczna witryny.

Powinniśmy starać się osiągnąć jak najlepsze efekty w obu tych dziedzinach.

4.1.1. Architektura strony internetowej

Każda strona internetowa powinna zawierać kilka podstawowych bloków.

Blok z zawartością
Każda strona internetowa ma blok zawartości. Najczęściej składa się on ze znaczników `<div>`. Taki podział umożliwia rozplanowanie na stronie różnych treści.

Szerokość tego bloku może być stała lub płynna. Szerokość stała gwarantuje, że blok zawsze będzie taki sam, niezależnie od wielkości okna przeglądarki. Szerokość płynna oznacza, że blok może się rozszerzać, wypełniając całą szerokość okna przeglądarki.

Nagłówek
W nagłówku powinna znaleźć się treść będąca wizytówką klienta lub firmy, np. logo lub nazwa firmy. Zadaniem tego bloku jest zwiększenie rozpoznawalności witryny i przypomnienie użytkownikom, do jakiej witryny należy oglądana strona.

Nawigacja
Kluczowym elementem witryny jest blok nawigacji. Powinien on być łatwy do odszukania i prosty w użyciu. Najczęściej element ten jest umieszczany u góry strony i tam użytkownicy spodziewają się go zobaczyć. Jeżeli blok nawigacji ciągnie się w dół wzdłuż zawartości, to powinien znajdować się jak najbliżej górnej krawędzi witryny, a wszystkie główne elementy menu powinny być widoczne zaraz po otwarciu strony.

Treść
Treść powinna być najważniejszym elementem strony. Przeciętny internauta po wejściu na stronę zwykle opuszcza ją po kilku sekundach, treść powinna więc być interesująca i zatrzymać go na dłużej. Położenie elementu z treścią należy zaplanować tak, aby od razu przyciągał uwagę użytkownika.

Stopka
Stopka znajduje się u dołu strony i zawiera np. dane kontaktowe, linki do niektórych sekcji witryny, informacje o prawach autorskich.

Biała przestrzeń
Pojęcie „biała przestrzeń" odnosi się do obszaru strony, na którym nie ma ani tekstu, ani ilustracji. Wolna przestrzeń na stronie jest tak samo ważna jak treść. Dobrze rozplanowana pozwoli stronie „oddychać". Jej brak będzie sprawiał wrażenie stłoczenia elementów.

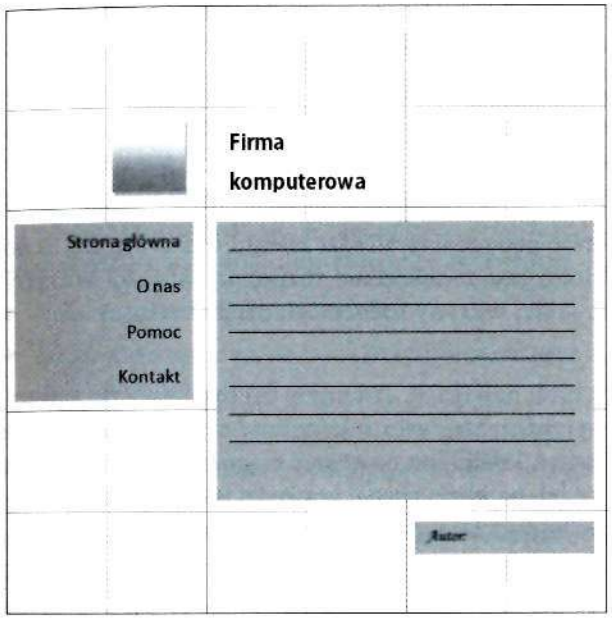
4.1.2. Zasady projektowania strony internetowej

Układ strony
Przystępując do projektowania struktury strony, należy starać się, żeby charakteryzowała się ona przejrzystą formą. Jedną z metod zachowania przejrzystości strony jest zaprojektowanie jej układu z zachowaniem podziału na dwie lub trzy kolumny.

Metodą określenia położenia elementów strony jest zastosowanie podziału bloku z zawartością na kilka równych części w poziomie i pionie. Korzystając z gotowych

szablonów, można utworzyć siatkę linii pomocniczych. Mając przygotowaną siatkę, możemy przystąpić do układania elementów strony.

Przykład układu strony opartego na siatce został pokazany na rysunku 4.1.



Rysunek 4.1. Układ strony oparty na siatce

4.1.3. Specyfikacja strony

Efektem wykonanej pracy powinna być specyfikacja strony. Dokument, który powstanie, powinien zawierać: rozplanowany wygląd strony, sposób działania witryny oraz jej przeznaczenie.

Przykład specyfikacji strony

Cel powstania strony
Opracowywana witryna ma charakter komercyjny. Zawiera informacje na temat działalności firmy oraz na temat oferowanych usług.

.....

Strona będzie zawierała podstawowe informacje o firmie.

.....

Język
Językiem projektowania strony będzie HTML 5.

Wygląd strony zostanie oparty na stylach CSS zgodnie ze specyfikacją (*http://www.w3.org/Style/CSS/*).

Przeglądarki internetowe i systemy
Projektowana witryna powinna być poprawnie wyświetlana we wszystkich przeglądarkach. Ponieważ w praktyce niektóre przeglądarki mogą wyświetlać strony w inny sposób niż standardowo założony, zostało określone oprogramowanie, które zapewnia prawidłowe funkcjonowanie witryny.

Przeglądarki:

- Internet Explorer,
- Google Chrome,
- Firefox.

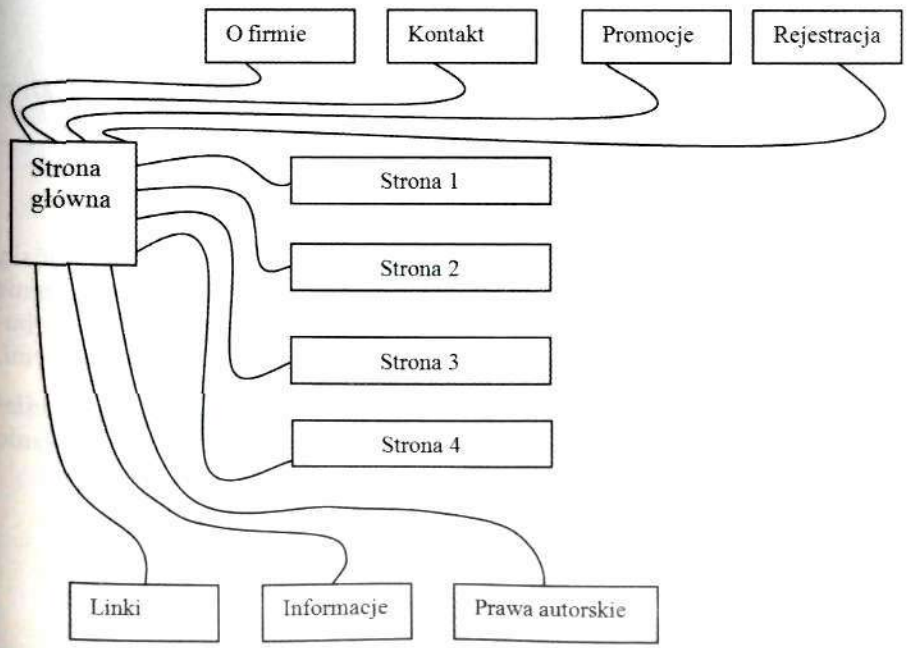
Systemy operacyjne:

- Windows,
- systemy z rodziny Linux.

Wymagania sprzętowe
Strona będzie wyświetlana zgodnie z założeniami, jeśli zostaną spełnione minimalne wymagania sprzętowe.

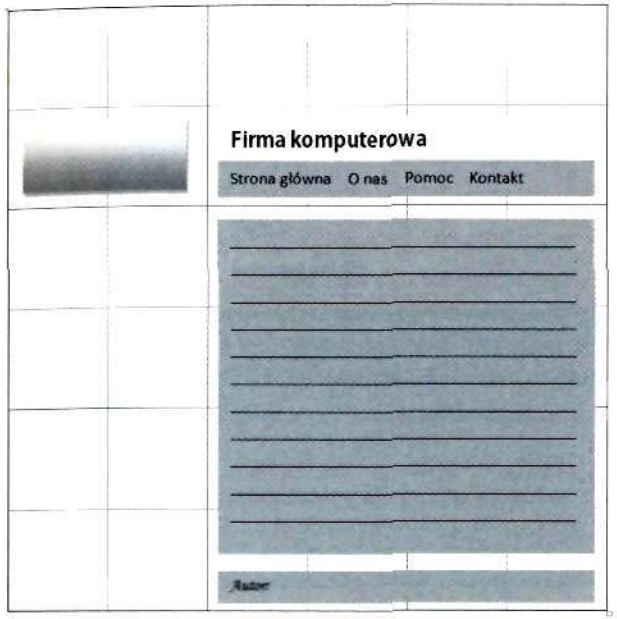
1. .....
2. .....
3. .....

Konstrukcja witryny



Rysunek 4.2. Przykładowa konstrukcja witryny

Układ strony



Rysunek 4.3. Przykładowy układ strony

Przygotowany układ strony pozwoli na określenie (przynajmniej orientacyjnie), jakie elementy powinny się na niej znaleźć i w których jej miejscach.

Wygląd ogólny
Strona ma być czytelna. Kolory powinny być jasne, nieintensywne, a teksty wyraźne, łatwe do przeczytania.

4.1.4. Projektowanie układu strony internetowej

Po zebraniu informacji na temat tworzonej witryny internetowej należy zastanowić się, jaki układ powinien zostać na niej zastosowany. Dobrym rozwiązaniem jest przejrzenie różnych stron i szablonów dostępnych w internecie. Mogą być one inspiracją przy budowaniu własnej strony. Umożliwiają również zapoznanie się z trendami graficznymi.

Kolejnymi etapami powinny być: sporządzenie na kilku kartkach szkicu witryny, określenie układu kolejnych stron, zaprojektowanie kilku wersji tej samej strony, zaplanowanie rozmieszczenia elementów strony.

PYTANIA I POLECENIA KONTROLNE

1. Z jakich etapów powinien składać się proces projektowania strony internetowej?
2. Wymień wyznaczniki określające właściwie zaprojektowaną stronę.
3. Z jakich bloków powinna się składać każda strona internetowa?
4. Jakie informacje powinna zawierać specyfikacja strony internetowej?

ZADANIE

Uczestnicy projektu wspólnie opracują szkic witryny internetowej, która będzie zawierała informacje na temat szkoły. Szkic witryny powinien zawierać następujące elementy:

- Określenie rodzaju informacji pojawiającej się na stronie.
- Projekt konstrukcji witryny i połączenia między stronami.
- Projekt układu strony.
- Ogólny wygląd strony.
- Rodzaj elementów graficznych, które będą wykorzystane na stronie.

5

Grafika na stronie internetowej

5.1. Rodzaje grafiki

Obrazy dostępne na komputerze mogą być tworzone przy użyciu jednej z dwóch podstawowych technik. W zależności od zastosowanej techniki wyróżniamy grafikę rastrową oraz grafikę wektorową. Grafiki te różnią się sposobem tworzenia i zapisu obrazów oraz sposobem ich edytowania.

5.1.1. Grafika wektorowa

Grafika wektorowa (inaczej: obiektowa) to grafika, w której obraz jest tworzony z obiektów podstawowych (prymitywnych), najczęściej figur geometrycznych (rysunek 5.1). Obrazy tej grafiki są całkowicie skalowalne, tzn. mogą być dowolnie powiększane lub zmniejszane bez straty jakości (rysunek 5.2). Wadą jest to, że grafikę wektorową trzeba tworzyć od podstaw, składając ją z wielu obiektów podstawowych. Formaty plików to np. SVG i CDR. Popularne programy do tworzenia grafiki wektorowej to CorelDRAW, Adobe Illustrator, AutoCAD.



Rysunek 5.1. Grafika wektorowa

W grafice wektorowej każdy z obiektów jest opisywany za pomocą parametrów, np. dla odcinka podawane są współrzędne jego końców, a w przypadku okręgu podaje się współrzędne środka i długość promienia. Tworzone obiekty mają atrybuty decydujące o ich wyglądzie, np. o grubości i kolorze linii, kolorze wypełnienia, stopniu przezroczystości. Atrybuty zależą głównie od stosowanego standardu opisu grafiki wektorowej.

Rysunek 5.2. Powiększanie obrazu nie zmienia jego jakości

Obrazy grafiki wektorowej mogą być łatwo konwertowane na ich odpowiedniki bitmapowe, należy jedynie podać rozdzielczość tworzonego obrazu wynikowego.

Konwersja obrazu grafiki bitmapowej na grafikę wektorową jest trudna i często nie daje oczekiwanych rezultatów.

5.1.2. Grafika rastrowa

Grafika rastrowa (inaczej: bitmapowa) to grafika, w której obraz jest tworzony z leżących blisko siebie punktów (*pikseli*). To, jak gęsto na danej powierzchni są ułożone piksele, stanowi o rozdzielczości obrazu. Ten rodzaj grafiki najczęściej jest stosowany do cyfrowego zapisu zdjęć lub obrazów ze skanera, gdyż wiernie oddaje wszystkie szczegóły (rysunek 5.3). Wadą grafiki rastrowej jest to, że przy dużym powiększeniu obrazu punkty (*piksele*), z których się on składa, zaczynają być widoczne (rysunek 5.4). Najczęściej spotykane formaty plików rastrowych to: BMP, JPG, PNG, GIF, TIFF. Popularne programy do tworzenia grafiki rastrowej to Paint, GIMP, Photoshop.



Rysunek 5.3. Grafika rastrowa



Rysunek 5.4. Piksele widoczne przy dużym powiększeniu

Piksel jest najmniejszym elementem obrazu wyświetlanego na ekranie monitora; to także najmniejszy element obrazu uzyskanego z aparatu cyfrowego lub skanera.

5.2. Sposoby zapisu obrazu cyfrowego

Obraz zapisany w postaci cyfrowej składa się z leżących blisko siebie punktów (pikseli). Piksel jest najmniejszym elementem zapisanego obrazu. Ma postać kwadratu lub prostokąta i jest wypełniony jednym kolorem. Kolor jest definiowany dla każdego piksela osobno. W zależności od użytej palety barw punkt może przyjmować nawet do 65 tysięcy kolorów. Podczas zapisywania obrazu zapamiętywane jest położenie punktu i jego kolor. Piksele tworzące obraz są ułożone w kolumnach i wierszach, tworząc siatkę zwaną mapą bitową. Ze względu na to, że obraz składa się z oddzielnych pikseli, przy jego powiększaniu uzyskujemy efekt schodkowania.

Przy zapisie obrazu cyfrowego są stosowane różne metody jego kompresji. Kompresja może być bezstratna (zmniejszenie objętości pliku nie powoduje pogorszenia jakości obrazu) lub stratna (po skompresowaniu następuje pogorszenie jakości obrazu).

5.2.1. Jakość obrazu

Na jakość obrazu wpływają takie parametry jak: rozdzielczość, rozmiar obrazu i głębia koloru.

Rozdzielczość
Rozdzielczość obrazu można określać na dwa sposoby.

Rozdzielczość geometryczna określa liczbę pikseli przypadających na jednostkę długości obrazu. *Miarą rozdzielczości jest liczba pikseli przypadająca na cal — ppi (ang. pixels per inch)*. Można stosować inną miarę rozdzielczości, np. piksel na milimetr. Wybór rozdzielczości powinien zależeć od przeznaczenia obrazu. Im większa rozdzielczość, tym lepsza jest jego jakość, ale również większy rozmiar (większa objętość) pliku. Zbyt niska rozdzielczość to pogorszenie jakości obrazu.

Rozdzielczość radiometryczna określa jasność piksela. Do zapisu jasności piksela najczęściej wykorzystuje się 8 bitów i wtedy można uzyskać 256 stopni jasności. Czasami stosuje się zapis 4-bitowy, co daje 16 stopni jasności. W profesjonalnych zastosowaniach do zapisu jasności wykorzystuje się 10, 12 i więcej bitów.

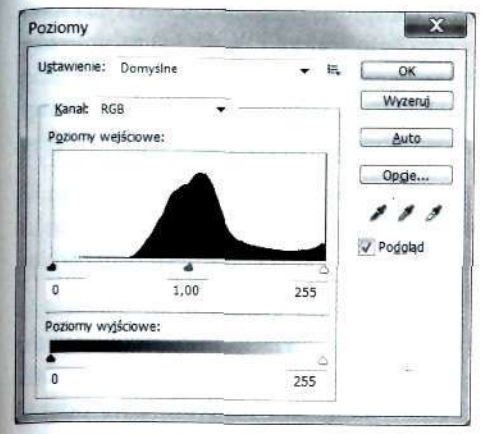
Rozmiar obrazu
Rozmiar obrazu określa fizyczną wielkość obrazu zdefiniowaną za pomocą dwóch parametrów: szerokości (ang. *width*) oraz wysokości (ang. *height*). Można go podawać w pikselach, calach lub milimetrach.

Głębia koloru
Głębia koloru określa liczbę barw możliwych do uzyskania w obrazie. Skoro kolor każdego piksela jest zapisywany przy użyciu określonej liczby bitów, głębia koloru jest informacją o tym, ile bitów potrzeba, aby opisać kolor każdego piksela. Im większa liczba bitów przeznaczonych na zapisanie piksela, tym większa paleta kolorów i lepsza jakość obrazu. Aby można było odwzorować kolory naturalne, liczba bitów opisujących kolor nie powinna być mniejsza niż 24. Miarą głębi koloru jest bpp (ang. *bits per pixels*).

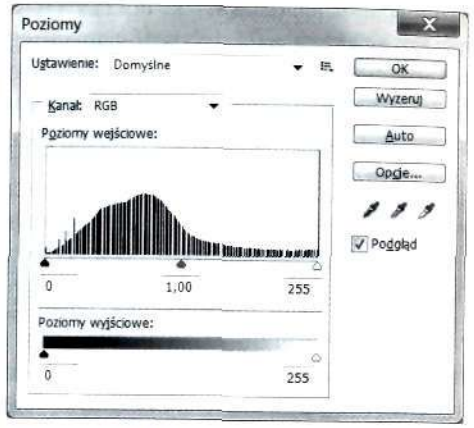
Histogram obrazu cyfrowego
Histogram obrazu to wykres przedstawiający liczbę pikseli o określonej jasności (intensywności) występujących w obrazie. Na podstawie histogramu można określić cechy obrazu. Obraz o dobrej jakości ma histogram w miarę wyrównany, a jasności pikseli przyjmują wartości od zera do wartości maksymalnej.

W celu poprawienia jakości obrazu można „rozciągnąć" histogram (np. z rysunków 5.5 i 5.7). Dzięki temu poprawiona zostanie rozróżnialność elementów obrazu. Niestety, nie doprowadzi to do pojawienia się w obrazie informacji, których tam nie było (*rysunki 5.6 i 5.8*).

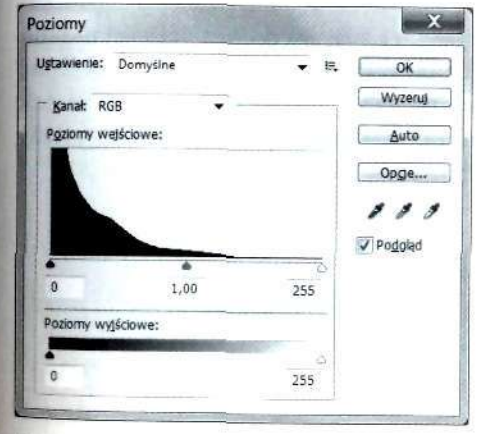
W celu uwypuklenia szczegółów obrazu, które są mało widoczne ze względu na niewielki kontrast, można zastosować wyrównanie histogramu.



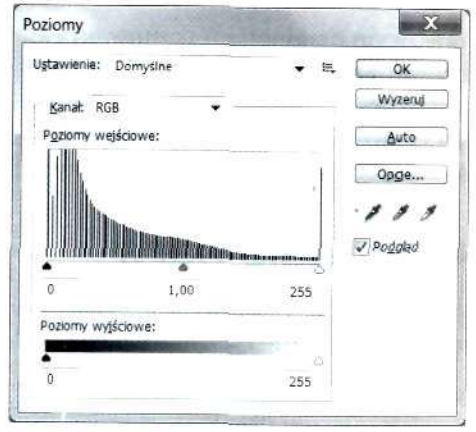
Rysunek 5.5.
Histogram obrazu jasnego



Rysunek 5.6.
Histogram rozciągnięty



Rysunek 5.7.
Histogram obrazu ciemnego



Rysunek 5.8.
Histogram rozciągnięty

5.3. Metody pozyskiwania obrazów

Zdjęcia fotograficzne
Zdjęcia wykonujemy najczęściej, korzystając z cyfrowych aparatów fotograficznych wyposażonych w karty pamięci (flash). Dodatkowo aparaty cyfrowe często mają wyświetlacz LCD, który udostępnia podgląd obrazu w czasie robienia zdjęcia. Aparat cyfrowy wykorzystuje zestaw soczewek (układ optyczny), które rzutują obraz na przetwornik optoelektroniczny (matryca CMOS — ang. *Complementary Metal Oxide Semiconductor*). Za pomocą przetwornika analogowo-cyfrowego obraz jest zamieniany z postaci analogowej na binarną. Postać binarna obrazu zapisywana jest na karcie pamięci w jednym z formatów przeznaczonych dla grafiki rastrowej. Format zapisu często można wybrać w oprogramowaniu aparatu. Najpopularniejszym formatem używanym do zapisu fotografii cyfrowej jest JPG.

Rozdzielczość zdjęć tworzonych przy użyciu aparatu fotograficznego wynosi najczęściej 72 ppi. Im większa rozdzielczość zapisanego obrazu, tym lepsza jakość i tym większy format zdjęcia w pikselach. W specyfikacji aparatu fotograficznego powinna być podana maksymalna rozdzielczość matrycy (określająca maksymalną rozdzielczość zdjęć w megapikselach — Mpx) oraz maksymalna rozdzielczość obrazu (określająca maksymalny rozmiar obrazu w pikselach).

Skaner

Skaner to urządzenie służące do odczytywania obrazu, np. z kartki papieru, i przetwarzania go na postać binarną w celu dalszej obróbki za pomocą komputera.

Podczas skanowania obraz jest oświetlany, a odbite od niego światło jest kierowane do czujników, które przetwarzają natężenie światła każdej z barw na wartości elektryczne. W przetworniku analogowo-cyfrowym wartości elektryczne są zamieniane na postać cyfrową. Postać binarna obrazu jest zapisywana na komputerze w jednym z formatów dostępnych dla grafiki rastrowej. Ustawienie formatu zapisu obrazu odbywa się w programie obsługującym skaner.

Podstawowym parametrem skanowania jest rozdzielczość. Jej wielkość zależy od układu optycznego zastosowanego w skanerze. Skanując zdjęcia i rysunki, najlepiej wybrać rozdzielczość 300 dpi i 24-bitową paletę kolorów. Osiąga się wtedy optymalny stosunek jakości obrazu do objętości pliku. Aby uzyskać obraz bardzo dobrej jakości, należy zwiększyć rozdzielczość, co jednak powoduje wzrost wielkości pliku wynikowego.

5.4. Modele barw

Do opisu kolorów używanych w grafice komputerowej stosuje się modele barw. Każdy model barw ma ściśle określoną paletę kolorów.

Istnieją następujące modele barw:

- RGB,
- CMYK,
- HSV (HSB).

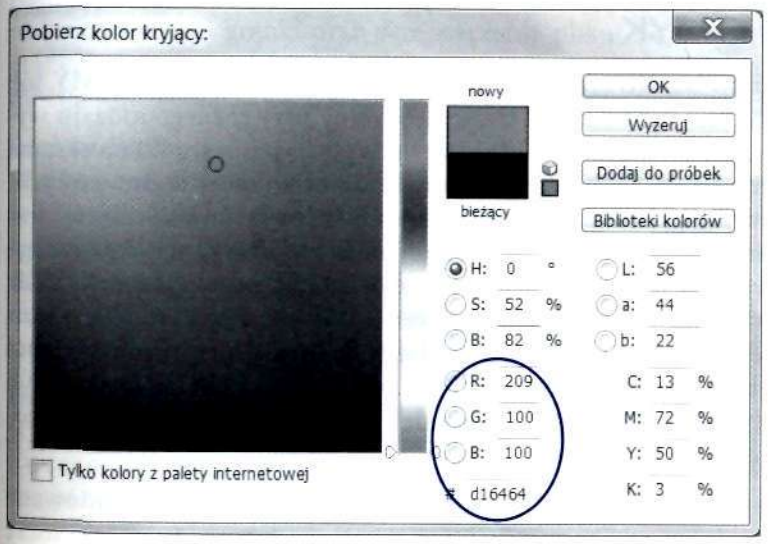
5.4.1. Model RGB

Podstawowym modelem barw stosowanym w grafice komputerowej jest model *RGB*. W tym modelu barw dowolny kolor można uzyskać, składając go z trzech podstawowych kolorów: czerwonego (ang. *Red*), zielonego (ang. *Green*) i niebieskiego (ang. *Blue*). Taki dobór barw wynika z właściwości oka ludzkiego, które reaguje na te trzy kolory. Te, mieszając się, dodają swoje wartości, co w zależności od natężenia każdej z barw tworzy w wyniku określony kolor. Jest to tzw. model addytywny. Ten model barw najczęściej jest stosowany w urządzeniach wyświetlających obraz, np.: monitorach komputerów, telewizorach, aparatach fotograficznych i skanerach.

Każda z barw składowych koloru jest zapisywana za pomocą 8 bitów (1 bajt) i przyjmuje wartości z zakresu *0 – 255*. Jeżeli dla wszystkich składowych barw zostanie ustawiona wartość *0*, to otrzymamy kolor czarny — natomiast gdy dla wszystkich składowych barw zostanie ustawiona wartość *255*, to otrzymamy kolor biały. Pozostałe kolory wynikają z odpowiednich ustawień wartości kolorów składowych.

Skoro wszystkich kombinacji wartości składowych jest ponad 16 mln, to tyle kolorów można uzyskać, mieszając poszczególne barwy.

W modelu RGB kolor każdego piksela jest opisywany za pomocą wartości kolorów *składowych*. *W grafice komputerowej czy przy tworzeniu stron internetowych* często definiuje się kolory wprost, podając ich wartości w kodzie heksadecymalnym (np. `background-color: #D16464` — rysunek 5.9).



Rysunek 5.9. Próbnik kolorów. Możliwość wybrania składowych koloru poprzez podanie wartości R, G, B lub kodu heksadecymalnego

WSKAZÓWKA

Jeżeli definiowany kolor będzie wielokrotnie wykorzystywany w projekcie, dobrze jest zapamiętać numer tego koloru. Pozwoli to na łatwe odtworzenie tej barwy bez konieczności żmudnego dopasowywania na zasadzie porównywania kolorów.

Należy zwrócić uwagę na to, że kolor uzyskiwany na różnych urządzeniach wyświetlających może znacznie się różnić. Dzieje się tak dlatego, że każde z urządzeń wyświetlających może mieć własny zakres barw możliwych do uzyskania.

Kanał alfa
Kanał alfa definiuje przezroczyste obszary grafiki. Jest on zapisywany wewnątrz obrazu razem z trzema wartościami modelu barw RGB. Wartość zerowa tego parametru oznacza pełną przezroczystość, wartość maksymalna — pełne zabarwienie.

Przy zapisie 32-bitowym kanał alfa jest zapisany na 8 bitach, natomiast trzy pozostałe kanały 8-bitowe przechowują informacje o poziomie nasycenia kolorów RGB. Powstały w ten sposób format RGBA (RGB + Alpha) pozwala określić stopień przezroczystości oraz wartość poszczególnych kolorów modelu RGB. Nie wszystkie programy do obróbki grafiki obsługują kanał alfa. Te, które go obsługują, definiują go w różny sposób. Również nie wszystkie formaty plików graficznych po zapisaniu przechowują informacje o kanale alfa.

5.4.2. Model CMYK

Kolejnym modelem barw bardzo często stosowanym w grafice jest model **CMY** lub rozszerzony model **CMYK**. Jest to model barw stosowany przy różnego rodzaju wydrukach. Model barw RGB znakomicie nadaje się do wyświetlania grafiki na ekranie — określone kolory uzyskujemy wówczas poprzez składanie kolorów podstawowych. Ta metoda nie sprawdza się natomiast przy mieszaniu kolorów farb w trakcie drukowania grafiki. Podstawowymi kolorami nadającymi się do uzyskania określonego koloru w procesie wydruku są kolory będące dopełnieniami barw w modelu RGB. Dopełnieniem koloru **R** (*Red*) jest kolor **G+B** (*Green* + *Blue*). Jest to kolor *Cyan* (turkusowy). Dopełnieniem koloru **G** (*Green*) jest kolor **R+B** (*Red* + *Blue*), czyli *Magenta* (purpurowy), a dopełnieniem koloru **B** (*Blue*) jest kolor **R+G** (*Red* + *Green*), czyli *Yellow* (żółty). Są to barwy podstawowe dla modelu **CMY** i na ich podstawie tworzone są pozostałe kolory.

Modele RGB i CMY są modelami dopełniającymi się. Oznacza to, że barwy definiowane w modelu CMY są dopełnieniami barw z RGB i w związku z tym możemy przeliczać wartości koloru z jednego modelu na drugi. Większość programów do tworzenia grafiki robi to automatycznie, ale proces ten można przeprowadzić samodzielnie. Odejmując od liczby *255* wartość barwy np. w modelu RGB, otrzymamy wartość jej dopełnienia w modelu CMY.

W praktyce częściej od modelu CMY jest spotykany model CMYK. Oprócz barw modelu CMY zawiera on barwę *Black* (czarny). W tym modelu jest ona oznaczona literą K w celu odróżnienia jej od barwy niebieskiej (*Blue*) używanej w modelu RGB. Stosowanie modelu CMYK wynika z tego, że mimo iż kolor czarny można otrzymać z kolorów CMY, w praktyce jest to trudne (najczęściej uzyskuje się kolor ciemnobrunatny). Dodanie koloru czarnego poprawia jakość drukowanego obrazu. Innym powodem są względy finansowe — do drukowania czarnym atramentem nie trzeba zużywać kolorowych atramentów modelu CMY.

5.4.3. Model HSV

Model barw **HSV** (ang. *Hue Saturation Value*) powstał na podstawie sposobu, w jaki barwy są postrzegane przez człowieka. Człowiek najczęściej korzysta ze standardowej palety kolorów. Dobierając barwę, wybiera z palety odpowiedni kolor farby, a dodając kolor biały lub czarny, przyciemnia lub rozmywa kolor podstawowy.

W modelu tym parametr **H** określa kolor (np. czerwony, zielony), parametr **S** — nasycenie koloru (np. czy kolor jest rozmyty, tzn. czy zawiera dużo szarości), a parametr **V** — jasność koloru. Czasami w celu określenia jasności koloru używa się liter **B**, **L** lub **I**.

5.5. Formaty plików graficznych

Format pliku określa sposób jego zapisu. W grafice komputerowej o formacie decyduje rodzaj tworzonej grafiki oraz przeznaczenie pliku. Najważniejszymi cechami pliku graficznego są jego rozmiar oraz jakość zapisanego obrazu. Znając przeznaczenie pliku, można jego format dobrać tak, aby pogodzić te dwie wykluczające się cechy.

Format BMP
BMP to podstawowy format grafiki rastrowej. Każdy piksel opisany jest przez kolor w formacie RGB. W przypadku obrazów o dużej rozdzielczości (np. zdjęć z aparatu fotograficznego) rozmiary plików mogą dochodzić do kilkudziesięciu MB. Format BMP umożliwia kompresję danych, ale i tak nie zmienia to znacznie rozmiarów plików. Duże rozmiary plików są podstawową wadą tego formatu. Zaletą natomiast jest bezstratny zapis oraz możliwość otwarcia pliku w większości programów.

Format JPEG
JPEG to jeden z najpopularniejszych formatów plików graficznych. Został opracowany na potrzeby internetu. Charakteryzuje się bardzo dużą kompresją. Ze względu na to, że jest oparty na kompresji stratnej, wielokrotny zapis między tym formatem a innymi prowadzi do zniekształceń ostatecznego obrazu. Widoczne jest to szczególnie w grafikach zawierających wyraźne linie i krawędzie, dlatego JPEG jest niechętnie stosowany przez grafików do zapisu efektów ich pracy. Natomiast w fotografiach, gdzie nie ma ostrych krawędzi i zmian barwy, błędy kompresji są widoczne dopiero przy wysokim jej współczynniku. Z tego powodu format JPEG jest często stosowany do zapisu zdjęć w aparatach cyfrowych. Dzięki niewielkim rozmiarom plików jest też najczęściej używanym formatem zdjęć na stronach internetowych.

Format GIF
GIF to format plików graficznych opracowany w celu umożliwienia zapisu prostych obrazów o niewielkiej liczbie kolorów. Jest wykorzystywany przy projektowaniu nagłówków, przycisków, ikon i prostej grafiki. Pozwala na zapisywanie i odtwarzanie prostych animacji. Umożliwia tworzenie grafik transparentnych (przezroczystych). Jedną z zalet tego formatu jest niewielki rozmiar plików. Z tych powodów GIF jest często stosowany przy projektowaniu stron internetowych.

Format PNG
Format PNG służy do zapisu prostych plików graficznych. Został opracowany z myślą o projektowaniu stron internetowych i w związku z wprowadzeniem opłat za wykorzystywanie w programach graficznych formatu GIF. Nie obsługuje animacji. Zaletami tego formatu są bezstratna kompresja i tworzenie plików o niewielkich rozmiarach oraz obsługiwanie kanału alfa.

Format PSD
PSD jest formatem programu Adobe Photoshop. Zapisuje wszystkie elementy formatowania tego programu (warstwy, maski, efekty, przezroczystość). Pliki zapisane w tym formacie mogą być odczytywane przez inne programy graficzne, m.in. Paint Shop Pro czy GIMP, ale przed taką operacją dokument trzeba spłaszczyć (jeżeli zawiera kilka warstw, to muszą one zostać połączone w jedną). Jest to format służący do edytowania wielu warstw, do ich tworzenia i modyfikowania. Nie powoduje utraty jakości obrazu, ale zazwyczaj tworzy pliki o dużych rozmiarach. Po opracowaniu gotowego obrazu najlepiej ostateczny efekt zapisać w innym formacie.

Format SVG
SVG (ang. *Scalable Vector Graphics*) jest uniwersalnym formatem dwuwymiarowej grafiki wektorowej. Powstał z myślą o zastosowaniu na stronach WWW. W formacie tym można zapisywać standardowe obiekty graficzne, np. prostokąty, elipsy, krzywe, jak również efekty specjalne, np. filtry, maski przezroczystości, wypełnienia gradientowe. SVG bywa też stosowany do zapisywania i odtwarzania prostych animacji.

Format TIFF
TIFF (ang. *Tagged Image File Format*) jest formatem plików graficznych, który służy do zapisywania grafiki rastrowej. Format ten pozwala na zapisywanie obrazów stworzonych w skali szarości oraz w wielu trybach koloru. Przechowuje ścieżki i kanały alfa, profile koloru i komentarze tekstowe. TIFF umożliwia stosowanie kompresji bezstratnej. Pliki zapisane w tym formacie mają rozszerzenie *.tif* lub *.tiff*.

5.6. Ogólne zasady tworzenia grafiki na potrzeby strony internetowej

Jednym z istotnych elementów projektowanej strony internetowej jest grafika. Jeżeli strona ma być lekka, użyteczna, dobrze zoptymalizowana i prosta w obsłudze, to grafika, którą będziemy tworzyli, musi współgrać z innymi elementami witryny. Narzędzi graficznych można używać do tworzenia przycisków nawigacyjnych, ikon, tła strony, logo, tapety oraz do modyfikowania fotografii.

5.6.1. Zasady tworzenia grafiki

- Grafikę tworzymy w skali 1 : 1. Staramy się unikać skalowania obrazów, może to bowiem spowodować pogorszenie ich jakości.
- Obrazy tworzymy w tzw. rozdzielczości ekranowej. Wynosi ona 72 ppi. Taka rozdzielczość pozwoli nam dopasować grafikę do tego, co będzie widoczne na monitorze.

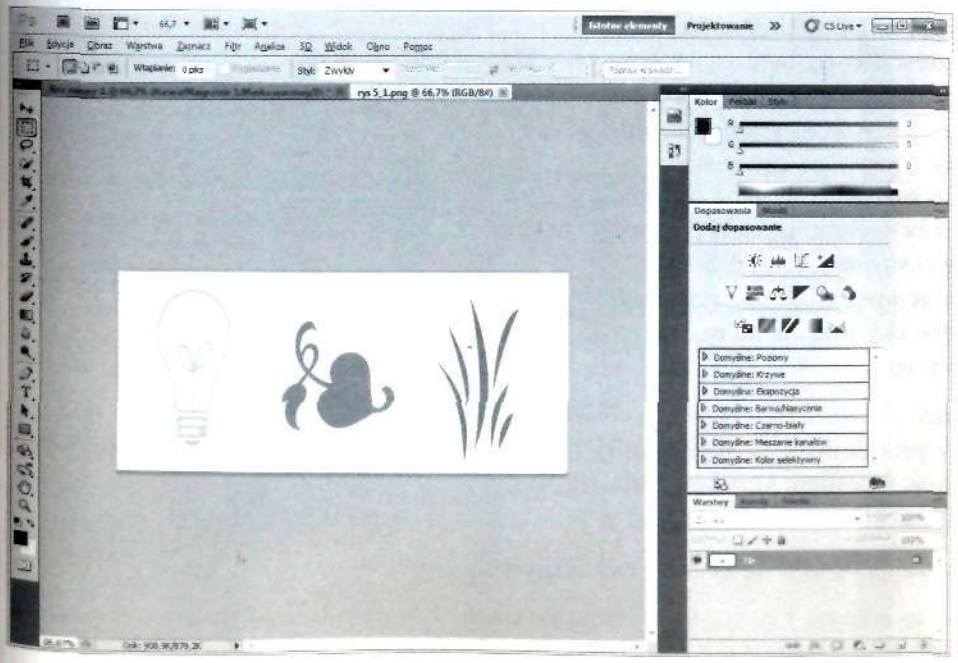
- Wybierając paletę kolorów, unikamy stosowania modelu CMYK. Powinien on być wykorzystywany tylko do druku. W internecie korzystamy z modelu RGB.
- Stosujemy formaty plików zgodnie z zasadą, że zdjęcia zapisujemy w formacie JPG, a grafiki z małą liczbą kolorów — w formacie GIF. Jeżeli zależy nam na jakości lub chcemy zastosować przezroczystość, zapisujemy grafikę w formacie PNG.

5.7. Modyfikowanie grafiki na potrzeby strony internetowej

Tworząc grafikę na potrzeby strony internetowej, można korzystać z różnych technik. Najpopularniejsza i powszechnie stosowana w internecie jest grafika bitmapowa. Nadaje się ona do przedstawiania i modyfikowania fotografii oraz do tworzenia skomplikowanych obrazów.

Jednym z narzędzi, które spełniają wymogi projektowania zaawansowanej grafiki bitmapowej, jest program Photoshop firmy Adobe.

Podstawowe okno programu Adobe Photoshop zawiera obszar roboczy, w którym dokonujemy obróbki obrazu. U góry znajduje się pasek *Menu*, a poniżej menu został umieszczony pasek *Właściwości*, którego zawartość zmienia się w zależności od narzędzia wybranego w pasku narzędzi. Pasek *Narzędzia* standardowo znajduje się z lewej strony głównego okna. Z prawej strony zaś widnieje zestaw paneli służących do edytowania grafiki (rysunek 5.10).



Rysunek 5.10. Okno programu Photoshop

5.7.1. Nawigacja w obszarze roboczym

Aby usprawnić pracę z aplikacją, można wykorzystać narzędzia przeznaczone do nawigowania w obszarze roboczym. Są one dostępne na pasku *Narzędzia*. Dodatkowo do większości z nich zostały zdefiniowane skróty klawiszowe.

Lupa

Lupa to narzędzie służące do powiększania obrazu, wybierane po kliknięciu ikony . Za pomocą tego narzędzia można dowolnie zmniejszać lub powiększać obszar roboczy projektu. Do tego samego celu można wykorzystać poniższe skróty klawiszowe:

- *Ctrl+-* pomniejsza obraz.
- *Ctrl++* powiększa obraz.
- *Ctrl+0* dostosowuje wielkość obrazu do ekranu.

Rączka

Narzędzie *Rączka* służy do przesuwania obrazu po ekranie. Dostępne jest na pasku narzędziowym po kliknięciu ikony . Można je również wywoływać poprzez naciśnięcie klawisza spacji na klawiaturze.

Cofanie

Użycie skrótu *Ctrl+Z* cofa nasze działania o jeden krok, a jego ponowne użycie przywraca cofniętą operację. Do cofania o kilka kroków służy skrót *Ctrl+Alt+Z*.

Restart widoku

Jeżeli podczas edytowania zmienimy układ okien w programie, możemy zrestartować widok. W prawym górnym rogu okna programu znajduje się przycisk *Istotne elementy*. Kliknięcie tego przycisku spowoduje rozwinięcie listy, na której należy wybrać opcję *Przywróć istotne elementy* (rysunek 5.11). Po tej operacji wszystkie standardowo dostępne okna powinny wrócić na swoje miejsca.



Rysunek 5.11. Restart widoku programu

Możemy również zresetować wszystkie ustawienia, które zdefiniowaliśmy podczas pracy z programem. W tym celu trzeba zamknąć program i uruchomić go ponownie, trzymając wciśnięte klawisze *Ctrl+Alt+Shift*.

5.7.2. Modyfikowanie zdjęć

Zdjęcia są jednym z istotnych elementów strony internetowej. Można je wykonać samodzielnie lub skorzystać z gotowych obrazków. Gotowe fotografie można pozyskiwać ze stron internetowych oferujących nieodpłatnie duże ilości zdjęć. Często są to zdjęcia

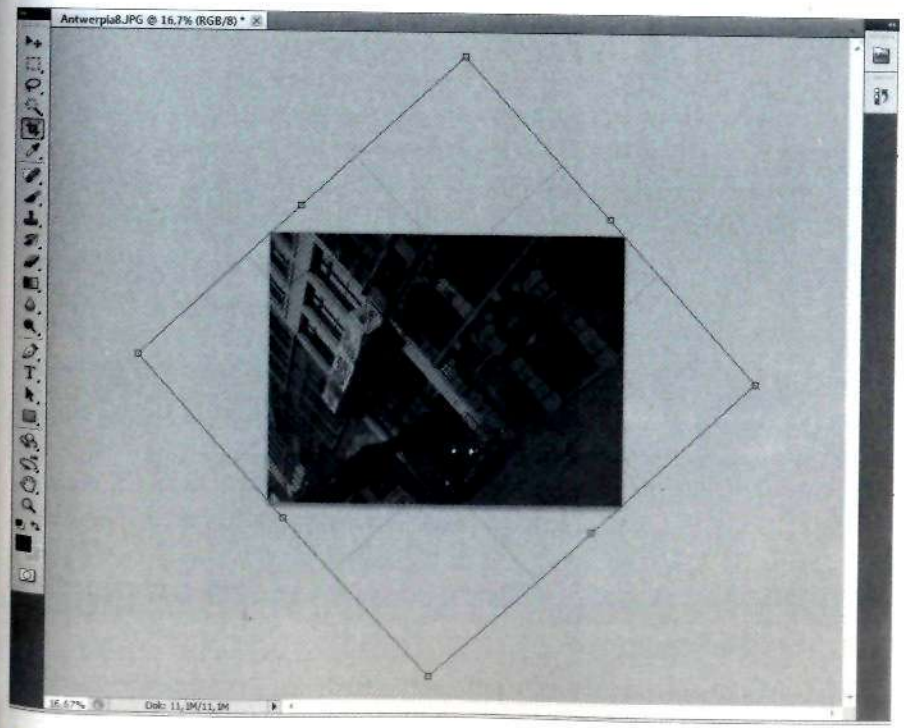
bardzo wysokiej jakości, a jedynym warunkiem ich wykorzystania jest podanie pod zdjęciem nazwiska autora. Jedną z największych i najlepszych stron z darmowymi zdjęciami jest *http://sxc.hu*. Inne interesujące strony to: *http://www.flickr.com/creativecommons/*, *http://www.stockvault.net/* oraz wyszukiwarka zdjęć *http://www.everystockphoto.com/*.

W programie Photoshop znajduje się wiele narzędzi, które mogą zostać wykorzystane do poprawienia wyglądu zdjęcia zapisanego w pliku. Modyfikowanie wyglądu zdjęcia może polegać na ustawieniu odpowiedniej głębi kolorów, wykadrowaniu go lub skorygowaniu jego zawartości.

Kadrowanie i korekcja zdjęć

Kadrowanie jest jedną z technik najczęściej stosowanych w celu poprawienia jakości zdjęć. Pozwala na usunięcie ze zdjęcia niepotrzebnych fragmentów oraz na jego „wyprostowanie".

Po wybraniu narzędzia do kadrowania (ikona , na pasku narzędziowym) należy zaznaczyć obszar zdjęcia, którego będzie dotyczyło kadrowanie. Pojawi się siatka ułatwiająca kadrowanie (rysunek 5.12). Przy jej użyciu można dowolnie przycinać obraz, ale można również, chwytając za punkt zaczepienia znajdujący się powyżej każdego rogu obrazu, obrócić go, tak aby linie poziome były idealnie poziome, a linie pionowe były równo ustawione w pionie. Dotyczy to szczególnie obrazów, w których te linie są wyraźnie widoczne (np. schodów, kolumn — rysunek 5.13).

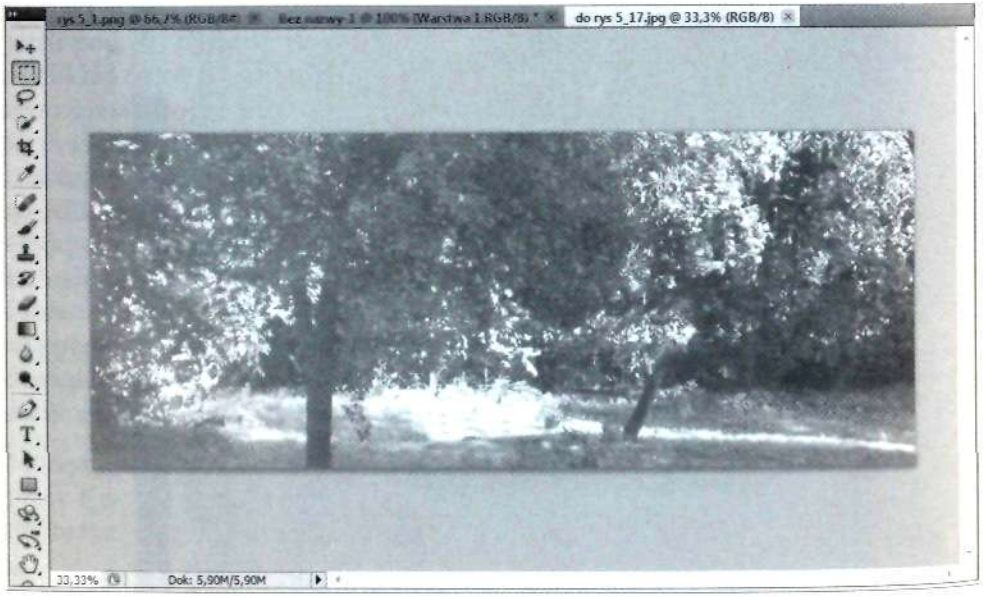


Rysunek 5.12. Kadrowanie zdjęcia



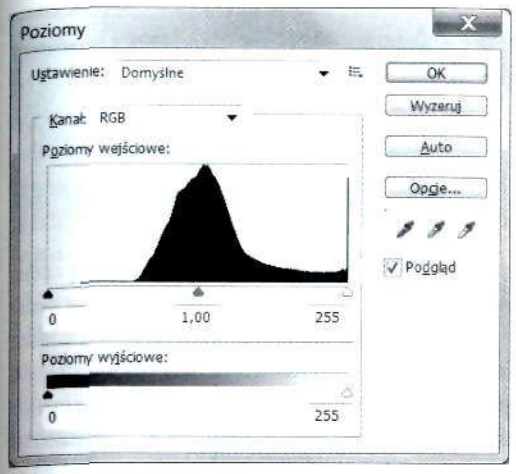
Rysunek 5.13. Zdjęcie wykadrowane

Jeżeli zdjęcie jest niewyraźne (zamglone) (rysunek 5.14), można poprawić głębię kolorów, analizując histogram obrazu. Dzięki temu narzędziu można zmienić głębię kolorów poprzez zwiększanie i zmniejszanie dopasowania jasnych i ciemnych barw.



Rysunek 5.14. Modyfikowane zdjęcie

W tym celu trzeba wybrać z menu *Obraz/Dopasowania/Poziomy*. Zostanie wyświetlone okno z wykresem, na którym widać, że zdjęcie jest niedoświetlone (lewa część wykresu jest pusta — rysunek 5.15).



Rysunek 5.15. Poziomy (histogram kolorów)

Aby to naprawić, należy przesunąć punkt kontrolny poziomu czerni (czarny trójkąt z lewej strony) oraz punkt kontrolny poziomu bieli (biały trójkąt z prawej strony). Można jeszcze punktem szarości (szary trójkąt na środku) określić równowagę między jasnymi i ciemnymi fragmentami obrazu (rysunek 5.16).



Rysunek 5.16. Zdjęcie po korekcie

Jeżeli na obrazie dominuje jakiś kolor, można tę wadę skorygować, wybierając opcję *Autokolor* (w menu *Obraz/Autokolor*). Nie zawsze automatyczne dopasowanie barwy da na obrazie prawidłowe kolory. Jeśli tak się nie stanie, należy ręcznie je dopasować, wybierając *Obraz/Dopasowania/Balans kolorów*.

UWAGA

Jeżeli zdjęcie jest mocno wyblakłe i wykres koloru widoczny na histogramie jest wąski, to po rozciągnięciu histogramu na zdjęciu będą widoczne gradienty (brak pewnych kolorów).

Usuwanie zbędnych elementów zdjęcia

Innym sposobem modyfikowania zdjęcia jest usunięcie zbędnych obiektów, postaci lub napisów. Mechanizm ich usuwania polega na zastąpieniu niepożądanego elementu pikselami zdjęcia pobranymi z jego okolic. Można do tego celu wykorzystać narzędzie *Stempel*, które służy do klonowania wybranego obszaru obrazu w inne miejsce. Na pasku narzędziowym jest ono oznaczone ikoną . Narzędzie to wykorzystuje końcówkę pędzla, co oznacza, że przed jego użyciem musimy zdefiniować grubość oraz kształt pędzla.

Aby użyć tego narzędzia, należy zdefiniować obszar do klonowania. Trzeba w tym celu, przytrzymując wciśnięty klawisz *Alt*, w odpowiednim miejscu kliknąć lewym przyciskiem myszy. Następuje wówczas określenie klonowanego obszaru. Gdy przeniesiemy kursor w miejsce jego wstawienia i klikniemy lewym przyciskiem myszy, zostanie zdefiniowana odległość między miejscem pobrania i miejscem wklejenia. Następnie malując po wybranym obszarze, spowodujemy sklonowanie wybranego fragmentu obrazu. Należy pamiętać o tym, by zwalniać co pewien czas przycisk myszy, aby zaktualizować odległość między miejscem pobrania i miejscem wklejenia. Efekt użycia narzędzia *Stempel* został pokazany na rysunkach 5.17 i 5.18.

Przykład 5.1

W pokazanym przykładzie (rysunek 5.17) po zastosowaniu narzędzia *Stempel* została usunięta postać (rysunek 5.18).

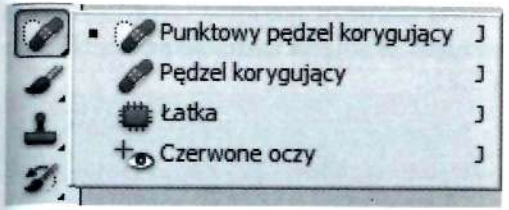


Rysunek 5.17. Zdjęcie przed zastosowaniem narzędzia Stempel



Rysunek 5.18. Efekt zastosowania narzędzia Stempel

Podobny efekt usunięcia zbędnych elementów ze zdjęcia uzyskamy, stosując narzędzie *Pędzel korygujący* znajdujące się pod ikoną związaną z narzędziem *Punktowy pędzel korygujący* (rysunek 5.19). Również to narzędzie przenosi zapamiętany fragment w inne miejsce. Różnica polega na tym, że wklejane piksele są dopasowywane pod względem jasności i koloru do otoczenia, w którym się znajdą. Wadą tego narzędzia jest to, że nie dopasowuje ono prawidłowo wyglądu pikseli na krawędziach mocno skontrastowanych.



Rysunek 5.19. Narzędzia korygujące

Czerwone oczy

Jedną z częściej wykonywanych czynności podczas korygowania błędów na zdjęciu jest usuwanie efektu czerwonych oczu. Aby wyeliminować tę wadę, należy wybrać narzędzie *Czerwone oczy*. Wskaźnik myszy przybierze kształt oka. Wystarczy kliknąć tym wskaźnikiem w miejscu wystąpienia efektu czerwonych oczu, aby go usunąć.

Zadanie 5.1

Przygotuj zdjęcia, które zamierzasz wykorzystać na stronie internetowej o szkole. Popraw ich wygląd, wykadruj wybrane fragmenty oraz usuń zbędne elementy.

Zmiana tła zdjęcia

Aby utworzyć baner reklamowy, ikonę lub tło, czasami wykorzystujemy tylko niektóre obiekty znajdujące się na zdjęciu. Jeżeli tło zdjęcia jest jednolite, można je usunąć i zastąpić innym, stosując narzędzie *Różdżka*. Służy ono do zaznaczania obszarów o podobnym kolorze. Po wybraniu ikony , klikając w jednolitym kolorystycznie obszarze, spowodujemy jego zaznaczenie. Dodanie kolejnych obszarów o zbliżonej barwie uzyskamy, zaznaczając te obszary przy wciśniętym klawiszu *Shift* (rysunek 5.20).



Rysunek 5.20. Zaznaczenie obszaru w kolorze nieba

W tym przypadku w wyniku użycia narzędzia *Różdżka* został zaznaczony obszar nieba. Jeżeli do dalszej obróbki zdjęcia potrzebny jest obszar odwrotny do zaznaczonego, można wybrać z menu *Zaznacz/Odwrotność*.

Przykład 5.2
W pokazanym przykładzie (rysunek 5.21) usunięto ze zdjęcia zaznaczone tło. Na jego miejsce zostało wstawione inne tło (rysunek 5.22).



Rysunek 5.21. Pozostał tylko zaznaczony fragment zdjęcia



Rysunek 5.22. Obraz zmodyfikowany

5.8. Tworzenie grafiki na potrzeby strony internetowej

Podczas tworzenia zaawansowanej grafiki na potrzeby strony internetowej niezbędne są narzędzia, które pozwolą na niezależne edytowanie różnych jej elementów. Takim narzędziem dostępnym w zaawansowanych programach graficznych są warstwy.

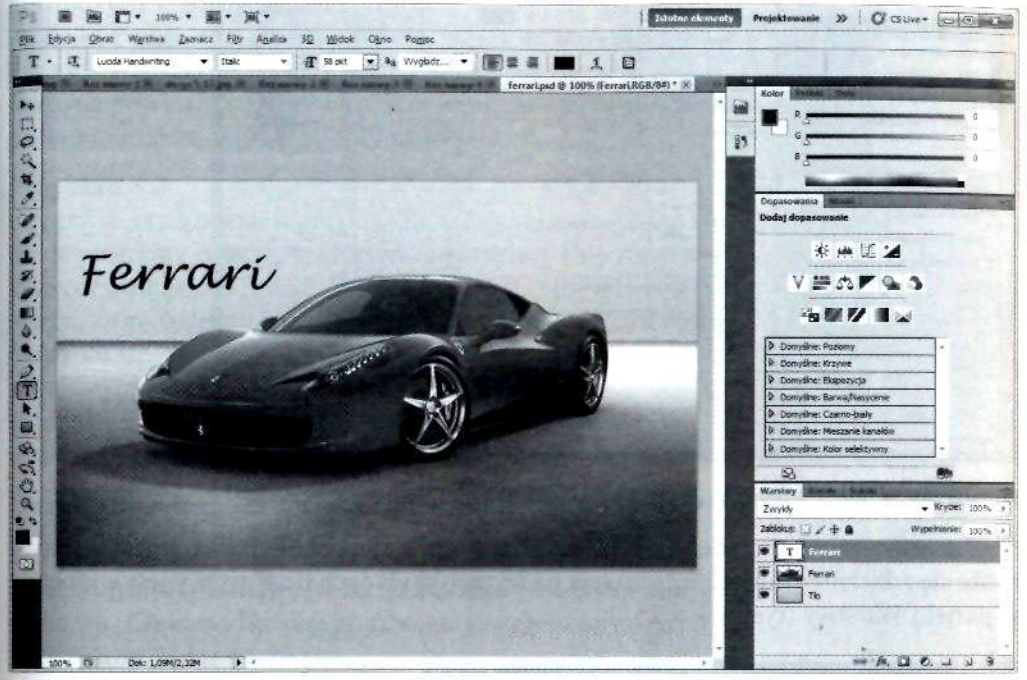
5.8.1. Warstwy

Przyjmuje się, że warstwy to odpowiedniki przezroczystych folii nakładanych na siebie. Na każdej z nich został umieszczony fragment obrazu. Po złożeniu utworzą one całość. Podobnie jest w programie Photoshop — każdy fragment obrazu powinien być tworzony na osobnej warstwie. Pozwala to na edytowanie wybranego elementu obrazu, podczas gdy pozostałe elementy pozostają niezmienione. Praca na warstwach umożliwia pełną kontrolę nad tworzonym obrazem. Wybraną warstwę można ukryć, aby nie przesłaniała innych warstw. Można też zmieniać kolejność wyświetlania warstw lub usuwać niepotrzebne warstwy.

Gdy rozpoczynamy pracę nad obrazem, ma on jedną warstwę — o nazwie *Tło*. Powinna ona znajdować się na spodzie stosu warstw. Na tę warstwę nakładane są kolejne.

W programie Photoshop do zarządzania warstwami służy panel *Warstwy*. Znajduje się on w prawej dolnej części okna programu (rysunek 5.23). Warstwy w panelu są wyświetlone w postaci stosu. Warstwa, która znajduje się na górze, przykrywa warstwy leżące niżej. Na samym spodzie znajduje się warstwa *Tło*. Klikając warstwę, wybieramy ją do edytowania. Warstwy do edytowania można wybierać również bezpośrednio na

obrazie, klikając wybrany fragment obrazu z wciśniętym klawiszem *Ctrl*. Warstwy na stosie można przemieszczać, zmieniając kolejność ich ułożenia.

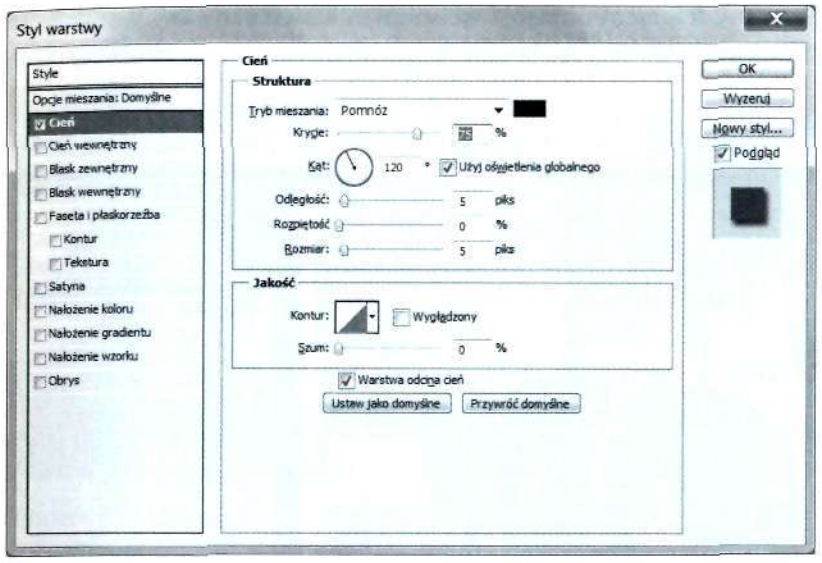


Rysunek 5.23. Panel z warstwami

Na rysunku 5.23 widać trzy warstwy. Warstwa *Tło* zawiera tło grafiki (kolor bordowy), warstwa *Ferrari* to samochód na jasnym podeście, trzecia warstwa zawiera napis „Ferrari".

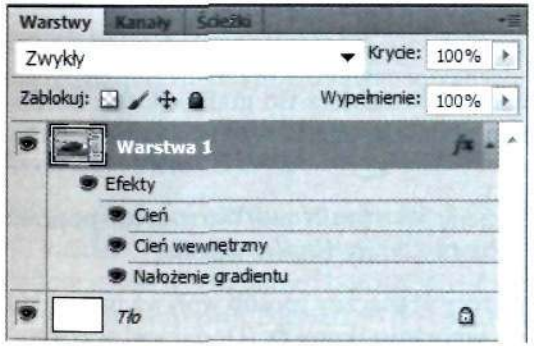
Kolejne warstwy są tworzone po wybraniu z menu *Warstwa/Nowa/Warstwa* lub poprzez wybranie w dolnym prawym rogu panelu *Warstwy* ikony *Utwórz warstwę*.

Dla każdej warstwy można zdefiniować styl, który w znaczny sposób wpływa na wygląd obrazu. W dolnej części panelu *Warstwy* po wybraniu ikony *fx* (Dodaj styl warstwy) pojawi się lista dostępnych opcji, a po wybraniu jednej z nich zostanie otwarte okienko, w którym można definiować styl warstwy (rysunek 5.24).



Rysunek 5.24. Okno Styl warstwy

Nawet prosty obraz po zdefiniowaniu dostępnych w tym oknie parametrów warstwy może się znacznie zmienić. Informacja o efektach dodanych do warstwy jest widoczna w panelu *Warstwy* (rysunek 5.25).



Rysunek 5.25. Efekty ustawione dla warstwy

Przy pracy na wielu warstwach z wykorzystaniem różnych narzędzi przydatne są następujące skróty klawiszowe:

- *Ctrl*+kliknięcie w obszarze obiektu (po wybraniu narzędzia *Przesunięcie*) — wybranie warstwy, na której znajduje się obiekt.
- *Ctrl*+*T* — zaznaczenie elementu warstwy w celu jego przekształcenia.

5.8.2. Tworzenie grafiki z wykorzystaniem warstw

Jednym ze sposobów projektowania grafiki jest tworzenie kompozycji z gotowych elementów graficznych. Efektem może być graficzny element strony, baner reklamowy lub logotyp. Połączenie i dobór zdjęć i innych elementów graficznych powinno polegać na takim ich dopasowaniu, aby powstała grafika stanowiła spójną całość.



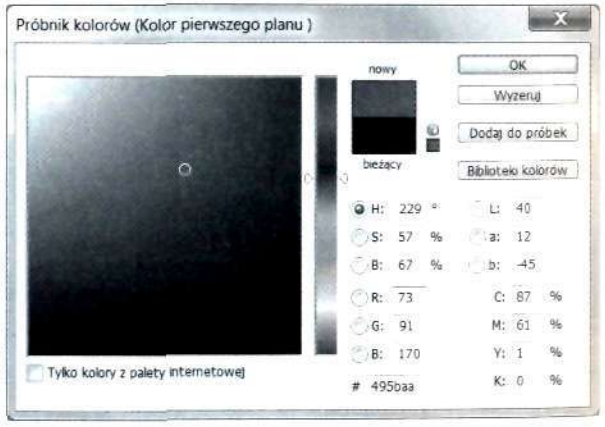
Rysunek 5.26. Ustawianie koloru pierwszoplanowego i koloru tła

Przykład 5.3

Dla grafiki otrzymanej w przykładzie 5.2 utworzymy własne tło. Wybieramy zdjęcie, w którym zamierzamy zmienić tło. Wykorzystując narzędzie *Różdżka*, zaznaczamy elementy tła, które należy usunąć. Wybieramy z menu *Zaznacz/Odwrotność*, aby zaznaczone zostały elementy potrzebne w dalszej obróbce grafiki. Następnie wybieramy z menu *Warstwa/Nowa/Warstwa przez kopiowanie*. Zostanie utworzona warstwa z zaznaczonych elementów. Nazwijmy ją *Grafika*.

Teraz do otrzymanego obrazka dodamy własne tło, np. błękitne niebo z chmurami.

Tworzymy nową warstwę (w menu *Warstwa/Nowa/Warstwa*) o nazwie *Chmury*. Na niej spróbujemy wygenerować tło. Ustawiamy kolor pierwszoplanowy na niebieski i kolor tła również niebieski, ale dużo jaśniejszy. Ustawienie tych parametrów jest realizowane na pasku *Narzędzia* (rysunek 5.26). Po kliknięciu ikony z górnym kwadratem zostanie otwarte okno z paletą kolorów (rysunek 5.27), w której trzeba wybrać odpowiedni kolor. Następnie należy kliknąć dolny kwadrat i ponownie w otwartym oknie wybrać kolor tła.



Rysunek 5.27. Okno palety kolorów

Z menu wybieramy *Filtr/Renderowanie/Chmury*. Utworzona warstwa zostanie wypełniona „sztucznymi" chmurami.

Kolejnym działaniem będzie uporządkowanie utworzonych warstw. Warstwę o nazwie *Tło* po odblokowaniu można usunąć (chwytamy za kłódkę i przeciągamy ją do kosza — ikona w prawym dolnym rogu okna *Warstwy*; podobnie postępujemy z warstwą). Pozostają dwie utworzone przez nas warstwy. Warstwę *Chmury* przesuwamy na dół,

pod warstwę *Grafika*. Można jeszcze poprawić wygląd nieba. W tym celu wybieramy warstwę *Chmury* i zaznaczamy ją, stosując kombinację klawiszy *Ctrl+T*. Trzymając na klawiaturze wciśnięty klawisz *Ctrl*, łapiemy za górne rogi i rozciągamy niebo, aby powstał trapez (rysunek 5.28). Z perspektywy obserwatora chmury powinny łączyć się z horyzontem. Można zmieniać kształt trapezu, aby uzyskać jak najlepszy rezultat. Ustawienia zatwierdzamy, naciskając *Enter*.



Rysunek 5.28. Manipulowanie wyglądem tła

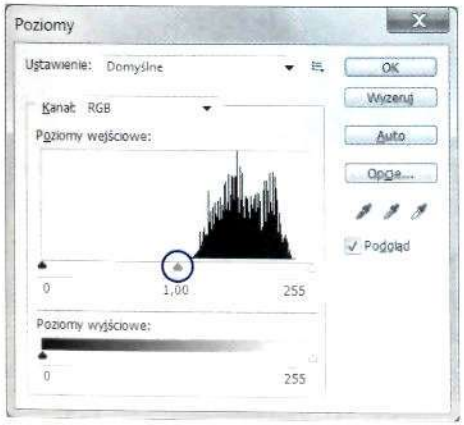
Można jeszcze bardziej poprawić wygląd chmur przez zmianę nasycenia i jasności.

W menu wybieramy *Obraz/Dopasowania/Poziomy* lub naciskamy na klawiaturze *Ctrl+L*.

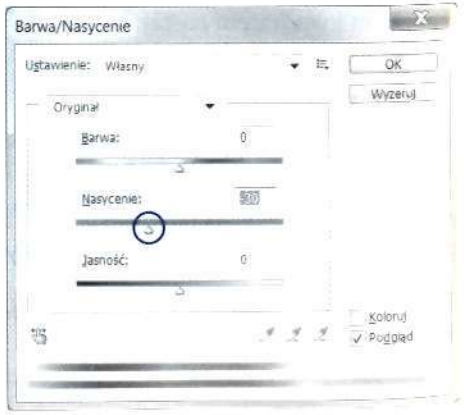
W otwartym oknie w celu rozjaśnienia obrazu przesuwamy punkt szarości (szary trójkąt na środku wykresu) w lewo (rysunek 5.29).

Można jeszcze zmniejszyć nasycenie barwy, wybierając z menu *Obraz/Dopasowania/Barwa/Nasycenie* (rysunek 5.30).

Stosując podobne metody, można tworzyć tło dla dowolnej kompozycji, niezależnie od tego, czy będzie to tło dla zdjęcia (jak w podanym przykładzie), baner reklamowy, tło tworzonej przez nas grafiki statycznej, czy też tło strony internetowej.



Rysunek 5.29.
Poziom dopasowania jasnych i ciemnych barw



Rysunek 5.30.
Zmiana nasycenia barwy

Przykład 5.4

Zaprojektujemy na stronę internetową grafikę zachęcającą do podróżowania po świecie. Przygotowane zostały zdjęcia z ciekawym krajobrazem, z interesującą postacią, z elementem graficznym symbolizującym kulę ziemską (rysunki 5.31 – 5.33).



Rysunek 5.31. Krajobraz



Rysunek 5.32.
Postać



Rysunek 5.33.
Grafika

Efektem pracy będzie grafika pokazana na rysunku 5.36.



Rysunek 5.34. Efekt pracy z grafiką

Tworzymy nowy dokument o rozmiarze 1920×980 px. Wybieramy narzędzie *Gradient* i w palecie kolorów dobieramy odpowiednie kolory. Po ich zatwierdzeniu tworzymy kolor tła, przeciągając myszą po warstwie tła z góry do dołu. Można dowolnie eksperymentować w celu uzyskania odpowiedniego efektu.

Z folderu ze zdjęciami przeciągamy po kolei na tło tworzonej grafiki zdjęcia z krajobrazem i z przygotowanym elementem graficznym. Wybór każdego zdjęcia należy zatwierdzić, naciskając *Enter*. Zostaną utworzone kolejne warstwy odpowiadające dodawanym zdjęciom.

Wybieramy warstwę z krajobrazem (w panelu *Warstwy*) i zaznaczamy zdjęcie (*Ctrl+T*). Zmieniamy jego rozmiar, dostosowując go do wielkości obrazu i pozostawiając z prawej

strony widoczny fragment tła. Przy zmianie wielkości zdjęcia zachowamy proporcje, trzymając wciśnięty klawisz *Shift*. Zatwierdzamy zmiany.

Wybieramy warstwę z elementem graficznym. Tak jak poprzednio, ustalamy rozmiar i położenie zdjęcia w obszarze widocznego tła. Jeżeli zdjęcie ma własne tło (jak na rysunku 5.33), trzeba wybrać narzędzie *Różdżka* i zaznaczyć tło zdjęcia. Następnie należy z menu wybrać *Zaznacz/Odwrotność*, aby odwrócić zaznaczenie. Ponownie z menu wybieramy *Warstwa/Nowa/Warstwa przez kopiowanie*. Pozostawiamy nowo utworzoną warstwę, natomiast warstwę, z której był kopiowany element graficzny, usuwamy. Zdjęcie z postacią umieszczamy w nowym dokumencie w celu usunięcia tła towarzyszącego postaci. Używając narzędzia *Różdżka*, podobnie jak poprzednio, zaznaczamy postać, a następnie, korzystając z opcji *Kopiuj/Wklej*, przenosimy ją do dokumentu z tworzoną grafiką i umieszczamy w odpowiednim miejscu.

Pozostało wstawienie tekstu. Wybieramy narzędzie *Tekst*. Na pasku właściwości w górnej części okna programu ustawiamy parametry związane z tekstem (styl, wielkość, kolor), klikamy w miejscu, gdzie zamierzamy umieścić tekst, i wpisujemy go.

Grafika została utworzona i można ją zapisać w wybranym przez nas formacie. Domyślnym formatem programu Photoshop jest PSD, ale na potrzeby internetu powinniśmy zapisać grafikę w formacie JPG lub PNG.

5.9. Tło strony internetowej

Tło strony internetowej jest jednym z ważniejszych jej elementów. Tłem może być obrazek, tekstura, wybrany kolor. W tle strony internetowej może pojawić się logo lub element reklamowy. W internecie jest wiele stron, które oferują zbiory bezpłatnych tekstur dla tła strony WWW. Jako tło może też zostać wykorzystane gotowe zdjęcie odpowiednio przetworzone na potrzeby strony. Można również zaprojektować własne tło.

Najczęściej projektowanym tłem strony jest jednolity kolor, czasami gradient lub grafika. Kolor może zostać określony w kodzie strony przez użycie atrybutu `bgcolor`, obecnie jednak stosuje się definiowanie kolorów w arkuszu stylów przy użyciu atrybutu `background-color`. Można również użyć pliku graficznego, który zostanie umieszczony na stronie jako tło.

Efektownie wyglądają tła zaprojektowane z użyciem gradientu. Przygotowując gradient, uwzględniamy kontekst. Gradient staramy się uzyskać, posługując się jednym kolorem i zmieniając jego jasność oraz nasycenie. Czasami, aby uzyskać określony efekt, dodajemy kolejne kolory.

5.9.1. Tworzenie własnego tła strony internetowej

Możemy spróbować utworzyć własną grafikę, która stanie się tłem strony.

Przykład 5.5

W programie Photoshop tworzymy nowy dokument. W otwartym oknie *Nowy* definiujemy szerokość 960 i wysokość 720 px. W polu *Zawartość tła* wybieramy opcję *Kolor tła*.

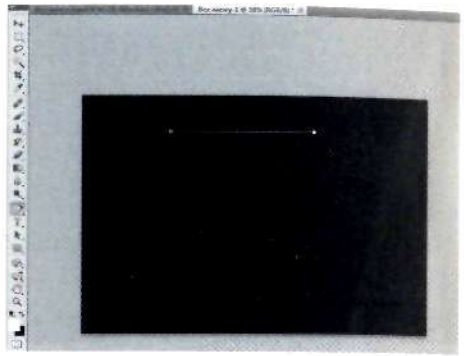
Klikamy przycisk *OK*. Powinien powstać obraz z czarnym kolorem tła. Jeżeli kolor tła jest inny, należy na pasku *Narzędzia* (rysunek 5.26) zmienić kolor na czarny. Tworzymy nową warstwę i nazywamy ją *Linia*. Wybieramy narzędzie *Pióro* i w górnej części zaznaczamy początek linii poziomej oraz (z wciśniętym klawiszem *Shift*) jej koniec (rysunek 5.35).

Wybieramy *Pędzel*, ustawiamy jego rozmiar na 8 px i twardość na 0%. Wybieramy *Zaznaczanie ścieżek* i klikając warstwę *Linia* prawym przyciskiem myszy, w menu wybieramy opcję *Obrysuj ścieżkę*. W otwartym oknie wybieramy *Narzędzie: Pędzel* i zaznaczamy opcję *Symulacja nacisku*. Zatwierdzamy ustawienia przyciskiem *OK* (rysunek 5.36).

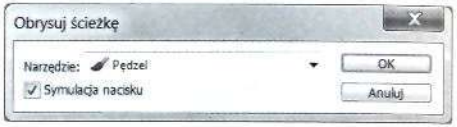
Naciskamy klawisz *Delete*, usuwając linię, a pozostawiając jej obrys (rysunek 5.37).

W dolnej części okna *Warstwy* (prawy dolny róg okna programu) klikamy ikonę *Utwórz nową warstwę wypełnienia lub korelacyjną* i z menu wybieramy *Barwa/Nasycenie*. W otwartym oknie *Dopasowania* zaznaczamy opcję *Koloruj*, na pasku *Barwa* wybieramy kolor błękitny i zmniejszamy nasycenie. Zaznaczamy warstwę *Linia* i kopiujemy ją 8 razy, ustawiając linie jedna pod drugą (użycie klawiszy: *Ctrl+J*, *Ctrl+T*, przesunięcie w dół, *Enter* — 8 razy). Efekt powinien być taki jak na rysunku 5.38.

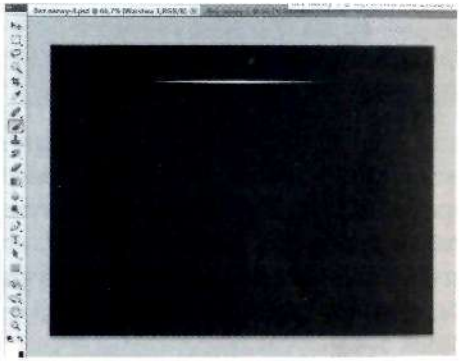
Powstały rysunek musimy scalić. W tym celu zaznaczamy wszystkie warstwy z liniami, trzymając wciśnięty klawisz *Shift*. Klikamy prawym przyciskiem myszy i wybieramy opcję *Scal warstwy*. Zaznaczamy powstałą warstwę, naciskamy klawisze *Ctrl+T*, klikamy prawym przyciskiem myszy i wybieramy opcję



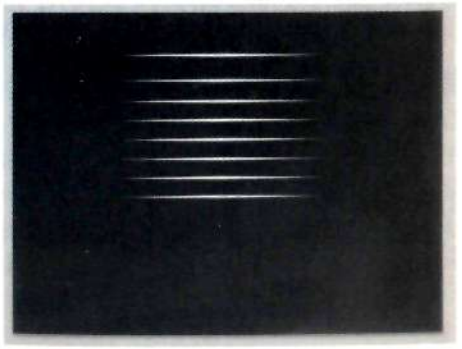
Rysunek 5.35. Początkowy etap tworzenia grafiki



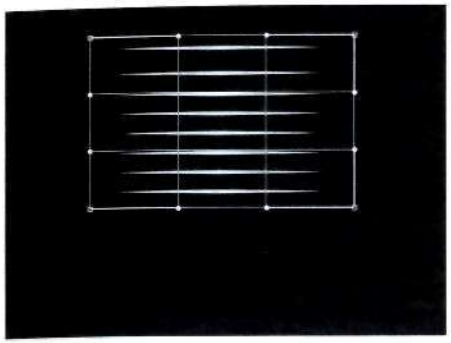
Rysunek 5.36. Tworzenie obrysu ścieżki



Rysunek 5.37. Obrys linii



Rysunek 5.38. Tworzenie linii



Rysunek 5.39.
Zaznaczenie scalonej warstwy

Wypaczenie. Efekt wykonanych działań przedstawia rysunek 5.39.

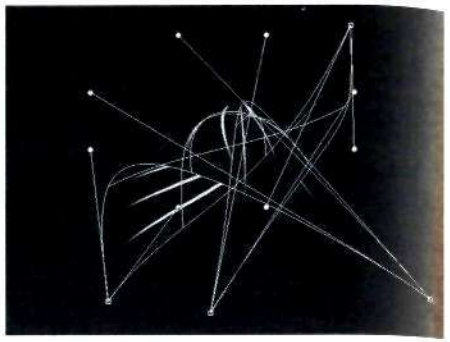
Zawijamy rogi zaznaczonego obszaru. Sposób zawijania zależy tylko od naszej wyobraźni (rysunek 5.40).

Po ustaleniu określonego kształtu naciskamy klawisz *Enter* i zatwierdzamy wprowadzone zmiany (rysunek 5.41).

Kopiujemy powstałą warstwę (*Ctrl+J*) i zaznaczamy ją (*Ctrl+T*). Klikamy prawym przyciskiem myszy i wybieramy z menu np. *Odbij w poziomie*. Powstałe warstwy można względem siebie przesuwać, aby uzyskać różne kształty. Ostateczny wygląd tapety zależy tylko od naszej inwencji (rysunek 5.42).

Stosując różne narzędzia programu, można uzyskać dowolne efekty i tworzyć dowolne tapety do wykorzystania w projektowaniu witryn internetowych.

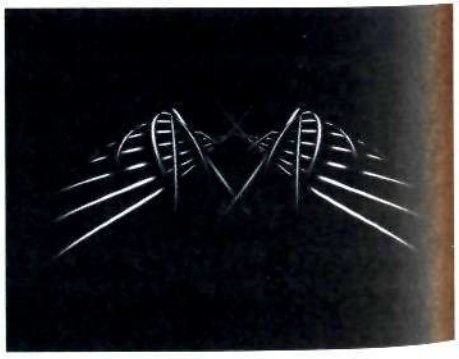
Zadanie 5.2
Wykorzystując poznane narzędzia, utwórz tło strony głównej witryny internetowej Twojej szkoły oraz tła dla kolejnych podstron.



Rysunek 5.40.
Zawijanie zaznaczonego obszaru



Rysunek 5.41.
Utworzona grafika

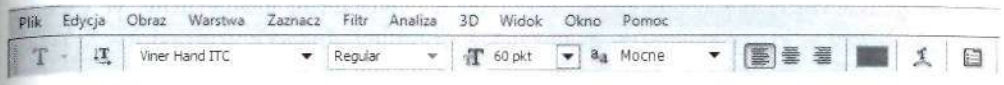


Rysunek 5.42. Efekt tworzenia tapety

5.9.2. Umieszczanie tekstu na tapecie

Do tworzonej grafiki można na każdym etapie pracy dodać dowolny tekst.

Przykład 5.6
Aby do zaprojektowanej tapety dodać napis, należy wybrać narzędzie *Tekst*. W pasku *Właściwości* w górnej części okna programu zostaną wyświetlone właściwości tworzonego napisu (rysunek 5.43).



Rysunek 5.43. Właściwości tekstu

Przed utworzeniem napisu można ustalić: rodzinę czcionek, styl czcionki, rozmiar, metodę wygładzania, wyrównanie oraz kolor tekstu. Po kliknięciu obszaru tworzenia grafiki automatycznie powstanie nowa warstwa, na której będzie umieszczony wprowadzany tekst. Wpisany tekst należy zatwierdzić, klikając na pasku *Właściwości* ikonę ✓. Tekst zapisany w oddzielnej warstwie może być modyfikowany tak jak każdy element graficzny umieszczony na warstwie (rysunek 5.44).



Rysunek 5.44.
Umieszczanie tekstu jako elementu grafiki

Powstałą grafikę zapisujemy w formacie JPG — wybieramy *Plik/Zapisz jako* i format JPG z listy w otwartym oknie w polu *Format*.

Zadanie 5.3
Utwórz elementy graficzne, które wykorzystasz w witrynie internetowej.

5.9.3. Cięcie i wstawianie tła na stronę internetową

Jednym ze sposobów wstawiania tła na stronę internetową jest określenie, z jakiego pliku będzie ono pobierane:

```
body {background-image: url('tło.jpg');}
```

Ze względu na czas ładowania strony projektanci stron internetowych, wykorzystując pliki graficzne, zwracają uwagę na to, aby były one jak najmniejsze. Jednym ze stosowanych podejść jest wybieranie plików małych, specjalnie utworzonych na potrzeby tła. Pliki takie zawierają grafikę o niewielkich rozmiarach, najczęściej o szerokości 1 px i wysokości zdefiniowanej według potrzeb, np. 600 px. Jeżeli taki plik zostanie powtórzony, utworzy ładnie wyglądającą grafikę tła.

Do powtarzania stosowane są formuły `background-repeat: repeat-x` albo `background-repeat: repeat-y`. Wtedy plik wypełniający tło będzie powtarzał się w pionie albo w poziomie (zależnie od wybranego rozwiązania):

```
body {
background-image: url('tlo.jpg');
background-repeat: repeat-x;
}
```

W takich plikach grafika najczęściej opiera się na tworzeniu gradientów. W efekcie powstają obrazy, w których zmiany następują w obrębie jednego koloru i dotyczą tylko jego odcieni; czasami zmiany dotyczą przejść między kolorami.

Gdy projektujemy tło oparte na gradiencie, należy utworzyć grafikę o odpowiedniej długości (zależnej od wysokości strony), a następnie przyciąć ją do szerokości 1 px, aby tło ładowało się jak najkrócej. Kolejnym krokiem jest umieszczenie tła na stronie z użyciem polecenia `repeat`. Jeżeli ponadto określimy kolor strony (`background-color`), podana barwa wypełni resztę strony pozostając poza obszarem grafiki, tworząc przejście między nią a zaprojektowanym tłem. W takim przypadku często jako `background-color` wybiera się odcień gradientu, który będzie sąsiadował z resztą strony.

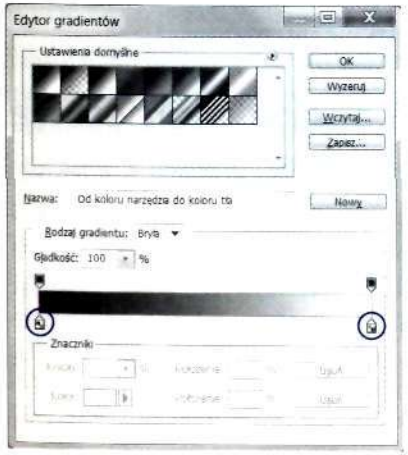
Przykład 5.7

W programie Photoshop tworzymy nowy dokument o rozmiarach 960×620 px. Wybieramy narzędzie *Gradient* i w pasku właściwości klikamy opcję *Edytor gradientów* (rysunek 5.45).



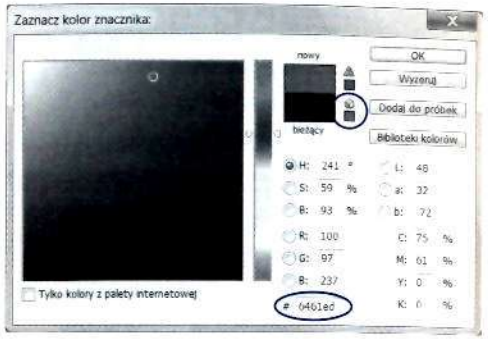
Rysunek 5.45. Właściwości gradientu

Zostanie otwarte okno edytowania gradientu, w którym można ustawić opcje koloru początkowego i koloru końcowego (rysunek 5.46).



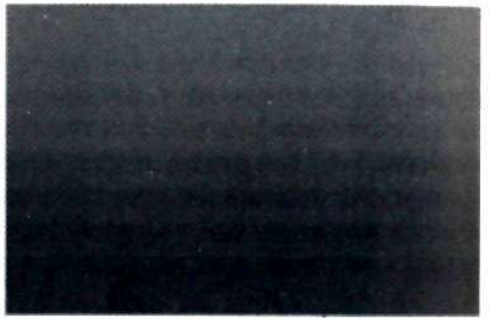
Rysunek 5.46. Edytowanie gradientu

W otwartym oknie klikamy dwukrotnie lewy znacznik koloru. W kolejnym oknie wybieramy odpowiedni kolor z palety (rysunek 5.47). Możemy również wybrać kolor, podając jego wartość *hex* (np. #6461ed) lub wartość *RGB*. Zatwierdzamy wybrany kolor i klikamy prawy znacznik koloru. Ustawiamy drugi kolor gradientu (np. #253040). Zapamiętujemy ustawione kolory.



Rysunek 5.47. Wybór koloru

Może okazać się, że wybrany kolor nie jest kolorem palety internetowej (w oknie koloru znacznika pojawi się ostrzeżenie). Skorygujemy to, klikając kwadrat koloru przy przycisku *Dodaj do próbek*. Zostanie automatycznie wybrany bezpieczny kolor palety internetowej. Po zatwierdzeniu kolorów przeciągamy myszą od górnej do dolnej krawędzi tworzonego obrazu. Powstanie obraz oparty na gradiencie (rysunek 5.48).



Rysunek 5.48.
Obraz oparty na gradiencie

Wybieramy narzędzie *Zaznaczanie pojedynczych kolumn*, po czym zaznaczamy kolumnę i wycinamy ją, wybierając z menu *Obraz/Kadruj*. Zapisujemy plik w formacie PNG.

Tak utworzony element graficzny może zostać wykorzystany do tworzenia tła strony internetowej.

Przykład 5.8

```
<html>
<head>
<style type="text/css">
 body {
 background-image: url('tło.png');
 background-repeat: repeat-x;
 }
</style>
</head>
<body>

</body>
</html>
```

Aby poza obszarem grafiki barwa jej dolnej części wypełniła resztę strony, należy w poleceniu `background-color` zdefiniować ten odcień gradientu. W tworzonej grafice był to kolor #253040. Uzupełniony skrypt wygląda teraz tak:

```
<html>
<head>
<style type="text/css">
 body {
 background-image: url('tło1.png');
 background-repeat: repeat-x;
 background-color: #253040}
</style>
</head>
<body>

</body>
</html>
```

Zadanie 5.4
Zaprojektuj grafikę, która będzie tłem strony internetowej. Wytnij z niej pasek o niewielkiej szerokości i dowolnej wysokości. Napisz skrypt wykorzystujący powstałą grafikę jako tło strony. Sprawdź jego działanie.

Zadanie 5.5
Zaprojektuj kolejną grafikę, która będzie tłem strony internetowej. Wytnij z niej pasek o niewielkiej wysokości i dowolnej szerokości. Napisz skrypt wykorzystujący powstałą grafikę jako tło strony internetowej. Sprawdź jego działanie.

5.10. Ikona i logo jako elementy strony internetowej

5.10.1. Tworzenie ikon

Kolejnym elementem graficznym, który może pojawić się na stronie internetowej, jest ikona. W internecie jest wiele stron, które oferują za darmo lub za niewielką opłatą zbiory ikon przeznaczonych do różnych celów. Jeżeli planujemy sami wykonać tego typu grafikę, możemy wykorzystać narzędzia do rysowania oferowane przez program Photoshop.

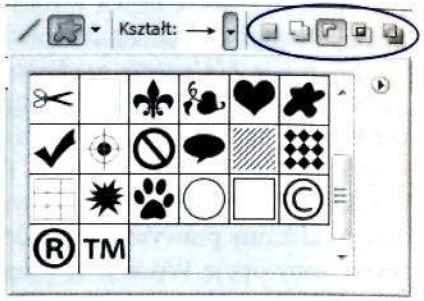
Przykład 5.9
Utworzymy ikony pokazane na rysunku 5.49.



Rysunek 5.49. Ikony na stronę internetową

Otwieramy nowy dokument. Wybieramy narzędzie *Elipsa* i trzymając wciśnięty klawisz *Shift*, rysujemy koło. Ustawiamy kolor koła, dwukrotnie klikając powstałą warstwę. W otwartym oknie wybieramy opcję *Nałożenie koloru* i ustalamy kolor, np. szary.

Aby wyciąć w kole strzałkę, wykorzystamy narzędzie *Kształt własny*. Po wybraniu tego narzędzia na pasku *Właściwości* zostanie wyświetlona opcja *Kształt* z listą gotowych kształtów (rysunek 5.50). Jeżeli na liście nie ma szukanego kształtu, należy kliknąć ikonę strzałki ▸ w górnym prawym rogu okna i z listy dołączyć kolejne kształty.



Rysunek 5.50. Gotowe kształty dostępne na pasku Właściwości i opcje łączenia kształtów

Można tworzyć własny obraz przez dodawanie lub odejmowanie kolejnych kształtów. Na pasku *Właściwości* dostępna jest opcja realizująca takie operacje (rysunek 5.50).

Kontynuując pracę, wybieramy narzędzie *Kształt własny*, rozwijamy listę kształtów i klikając ikonę strzałki, dodajemy grupę *Strzałki*. W pasku *Właściwości* wybieramy opcję *Odejmij od obszaru kształtu* (rysunek 5.50) i rysujemy odpowiedni kształt.

Podczas rysowania strzałki w razie potrzeby można przytrzymać klawisz spacji, aby przesunąć rysowany kształt w odpowiednie miejsce. Różne programy graficzne (np. GIMP) zawierają narzędzia do łatwego tworzenia ikon. W internecie można znaleźć proste programy, których jedynym przeznaczeniem jest ich tworzenie, np. iICO. Można go pobrać ze strony *http://www.mintrasystems.com/products/mintra/product.php?p=iico*. Jest to program bardzo prosty w obsłudze, który umożliwia tworzenie własnych ikon oraz pobieranie ikon z istniejących plików graficznych.

Zadanie 5.6
Narysuj samodzielnie drugą ikonę, stosując poznane metody pracy.

Zadanie 5.7
Wykonaj ikony, które wykorzystasz na stronie internetowej do wypunktowania kierunków kształcenia. Zastanów się, jakie ikony mogłyby jeszcze pojawić się na stronie, i utwórz je.

5.10.2. Tworzenie logo

Używając podobnych narzędzi, można zaprojektować logo firmy (logotyp). Takie logo może zostać wykorzystane na drukach firmowych, na wizytówkach lub może być — jako symbol identyfikujący firmę — umieszczone na stronie internetowej.

Przykład 5.10
Zaprojektujemy logo podobne do pokazanego na rysunku 5.51.



Rysunek 5.51. Logo firmy

Tworzymy nowy dokument. Wybieramy narzędzie *Elipsa* i trzymając wciśnięty klawisz *Shift*, rysujemy koło.

Zostanie utworzona nowa warstwa. Ustawiamy kolor koła, klikając dwukrotnie powstałą warstwę. W otwartym oknie wybieramy opcję *Nałożenie gradientu* i klikając opcję *Gradient*, ustalamy kolory gradientu. Kolejny element grafiki to okrąg. Wybieramy narzędzie *Kształt własny* i na pasku *Właściwości* w opcji *Kształt* szukamy okręgu. Rysujemy okrąg taki jak na rysunku 5.52. Powstała kolejna warstwa. Jeżeli okrąg nie jest biały, to klikamy prawym przyciskiem myszy powstałą warstwę i wybieramy opcję *Wyczyść styl warstwy*. Następnym elementem będzie biała kropka umieszczona w okręgu. Wybieramy narzędzie **Elipsa** i rysujemy kółko. Powoduje to powstanie kolejnej warstwy.



Rysunek 5.52. Grafika do logo firmy

Wszystkie elementy grafiki powinny zostać wyśrodkowane. Zaznaczamy warstwy z wciśniętym klawiszem *Shift*. Naciskamy klawisz *V*, aby wybrać narzędzie *Przesunięcie*, i na pasku *Właściwości* klikamy opcję *Wyrównaj środki w pionie* oraz *Wyrównaj środki w poziomie*.

Kolejnym krokiem jest modyfikacja kształtu zewnętrznego koła. Zaznaczamy warstwę z tym kołem (pierwsza warstwa) i wybieramy narzędzie *Konwertowanie punktów*. Po kliknięciu krawędzi koła pojawią się punkty. Klikamy dolny punkt, który zostanie usunięty, a następnie, trzymając wciśnięty klawisz *Shift*, przesuwamy pozostałe punkty tak, aby uzyskać oczekiwany kształt (rysunek 5.53).



Rysunek 5.53. Modyfikowanie grafiki

Zaznaczamy wszystkie warstwy z grafiką i łączymy je w jedną grupę, naciskając na klawiaturze *Ctrl+G*. Naciskamy *Ctrl+T*, aby zaznaczyć grafikę, i z przytrzymanym klawiszem *Shift* zmniejszamy odpowiednio jej rozmiar.

Pozostało jeszcze dodanie napisu. Wybieramy narzędzie *Tekst*, dobieramy czcionkę oraz jej kolor i wpisujemy tekst. Ustawiamy jego położenie względem grafiki. Dodajemy drugi tekst, odpowiednio go formatując. Grupujemy wszystkie warstwy. Stosując narzędzie *Kadrowanie*, przycinamy grafikę do odpowiednich rozmiarów i zapisujemy ją w pliku.

Zadanie 5.8
Zaprojektuj i utwórz logo swojej szkoły. Pamiętaj, że powinno być ono proste i związane z działalnością placówki.

5.11. Nawigacja

Nawigacja między stronami witryny internetowej jest jedną z podstawowych funkcjonalności, które powinny znaleźć się na stronie.

5.11.1. Tworzenie przycisków nawigacyjnych

Do utworzenia prostokątnego przycisku wykorzystamy narzędzie *Zaokrąglony prostokąt*.

Przykład 5.11
Tworzymy nowy dokument. Wybieramy narzędzie *Zaokrąglony prostokąt* i na pasku *Właściwości* ustawiamy zaokrąglenie rogów (*Promień*) oraz kolor wypełnienia (rysunek 5.54). Po narysowaniu prostokąta powstała nowa warstwa. Można do niej dodać zestaw efektów (ikona *fx*), które zmienią kształt przycisku.



Rysunek 5.54. Właściwości narzędzia Zaokrąglony prostokąt

Inną metodą określenia wyglądu przycisku jest zastosowanie gotowego stylu.

Przykład 5.12
Gdy narysujemy przycisk przy użyciu narzędzia *Zaokrąglony prostokąt*, na pasku *Właściwości* wybieramy opcję *Styl warstwy*. W otwartym oknie klikamy ikonę strzałki położoną w jego górnym prawym rogu. Dołączamy do listy kolejne kształty przycisków i wybieramy jeden z nich (rysunek 5.55). Dodajemy napis, wybierając narzędzie *Tekst* i ustawiając dla niego na pasku *Właściwości* rodzaj czcionki, jej wielkość oraz kolor.



Rysunek 5.55. Projektowanie przycisku

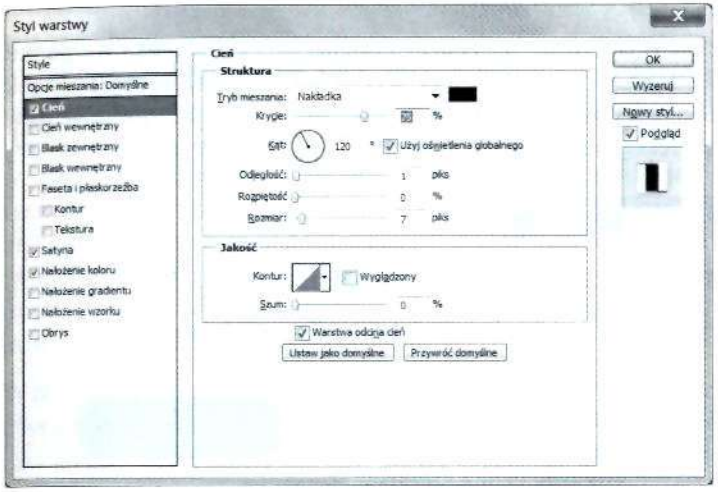
Samodzielne tworzenie przycisków menu może dać lepsze efekty. Należy tylko poeksperymentować ze stylami warstwy.

Przykład 5.13
Utworzymy przycisk widoczny na rysunku 5.56. Do tego celu wykorzystamy narzędzie *Zaokrąglony prostokąt*. Wygląd przycisku poprawimy, ustawiając odpowiednio parametry w opcji *Styl warstwy*.



Rysunek 5.56. Style warstwy wpływają na wygląd przycisku

Tworzymy nowy dokument o wymiarach 200×120 px. Wypełniamy obszar tła kolorem. W tym celu z warstwy tła usuwamy kłódkę. Wybieramy *Styl warstwy* (ikona *fx* w panelu *Warstwy*) i na liście opcję *Nałożenie koloru*. Klikamy ikonę z kolorem i ustawiamy odpowiedni kolor (np. #252a30). Wybieramy narzędzie *Zaokrąglony prostokąt*, w pasku *Właściwości* ustawiamy *Promień zaokrąglenia* (np. 20 px) i rysujemy prostokąt o odpowiednich rozmiarach. Teraz musimy nadać narysowanej figurze styl. Klikamy dwukrotnie warstwę z prostokątem i w otwartym oknie *Styl warstwy* ustawiamy wybrane parametry (rysunek 5.57).



Rysunek 5.57. Definiowanie stylu przycisku

Cień nada przyciskowi głębię — ustawiamy jego parametry np. tak jak na rysunku 5.57.

Satyna podkreśli kształty, np.: *Tryb mieszania Zwykły*, *Krycie* 100%, *Kąt* 9°, *Odległość* 170 px, *Rozmiar* 60 px.

W przypadku opcji *Nałożenie koloru* można przetestować różne kolory. Wystarczy dodać napis i zapisać do pliku utworzoną grafikę.



Rysunek 5.58. Menu

Zadanie 5.9
Zaprojektuj i utwórz menu podobne do pokazanego na rysunku 5.58.

Zadanie 5.10
Zaprojektuj i utwórz przyciski nawigacyjne ułatwiające poruszanie się po stronie. Zwróć uwagę, aby współgrały one z logo szkoły oraz z utworzonym wcześniej tłem strony. Umieść na przyciskach odpowiednie napisy.

5.12. Mapowanie obrazu

Obraz przygotowany na potrzeby strony internetowej może stać się źródłem odsyłaczy do innych stron internetowych. W tym celu grafikę należy podzielić na kilka obszarów o określonym kształcie i do każdego z nich przypisać adres internetowy. Powstanie tzw. mapa odsyłaczy. Kliknięcie takiego obszaru spowoduje przeniesienie do strony, której adres został powiązany z obszarem.

Ten mechanizm kodu HTML może zostać wykorzystany w menu nawigacyjnym. Gdy zaprojektowane w programie graficznym przyciski umieścimy w witrynie, można je będzie powiązać za pomocą odsyłaczy z innymi jej stronami.

Fragment kodu, który definiuje mapę odsyłaczy, znajduje się w przykładzie 5.14, a efekt jego wykonania został pokazany na rysunku 5.59.



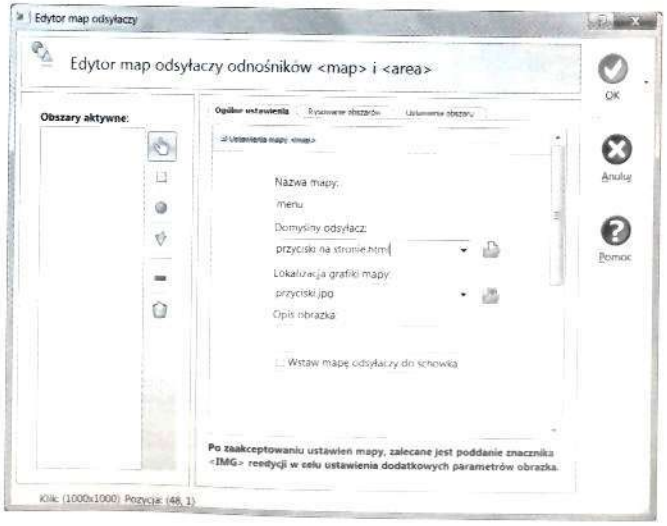
Rysunek 5.59. Fragment strony ze zdefiniowanymi przyciskami

Przykład 5.14

```
<div class="kolumny" id="kol1">
 <p> <img border=0 src="przyciski.jpg" usemap="#menu"></p>
 <map name= "menu">
 <area shape="rect" coords="0,0,99,49" href="www.wp.pl" alt="strona 1"
target="_blank">
 <area shape="rect" coords="157,100,40" href="www.onet.pl" alt="strona 2"
target="_blank">
</map>
</div>
```

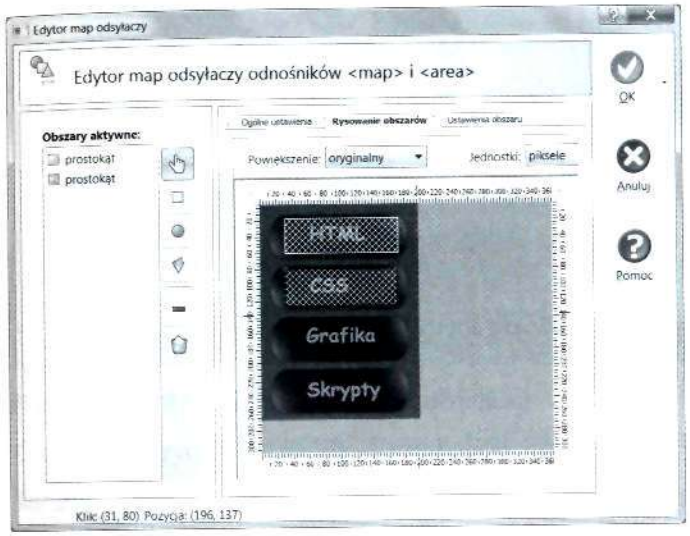
Aby usprawnić definiowanie obszarów, których kliknięcie przeniesie użytkownika do strony zdefiniowanej w kodzie HTML, można użyć edytorów map odnośników. W takie edytory zostały wyposażone niektóre programy graficzne. Jednym z nich jest program Pajączek.

Po uruchomieniu programu należy z menu wybrać *Narzędzia/Grafika i multimedia/Edytor map odsyłaczy*. W oknie, które zostanie otwarte, na karcie *Ogólne ustawienia* trzeba wpisać nazwę mapy, domyślny odsyłacz (adres pliku HTML), lokalizację mapowanej grafiki (adres pliku z grafiką, np. JPG) oraz opis obrazka (rysunek 5.60).



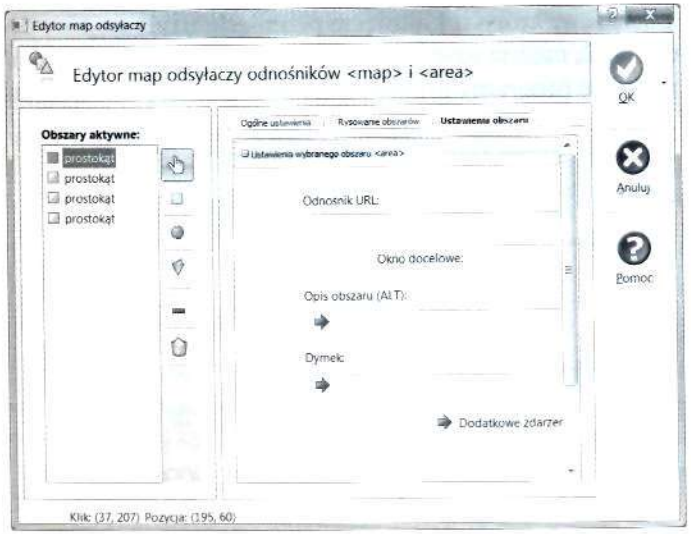
Rysunek 5.60. Edytor map odsyłaczy

Po przełączeniu się na kartę *Rysowanie obszarów* można definiować obszary odnośników (rysunek 5.61). W panelu z lewej strony wybieramy kształt aktywnego obszaru. Może to być prostokąt, koło lub kształt nieregularny.



Rysunek 5.61. Zaznaczanie obszarów aktywnych

Na karcie *Ustawienia obszaru* należy dla każdego zdefiniowanego obszaru określić adres otwieranej strony (rysunek 5.62).



Rysunek 5.62. Definiowanie adresu URL dla zaznaczonego obszaru

Po zdefiniowaniu wszystkich obszarów i wymaganych dla nich parametrów klikamy przycisk *OK*. Program Pajączek wygeneruje kod odpowiadający mapowanej grafice.

Przykład 5.15

```
<img src="przyciski.jpg" usemap="#moje_menu" border="0" width="100"
height="135" alt="">
<map name="moje_menu" id="moje_menu">
  <area shape="default" href="przyciski na stronie.html">
  <area shape="rect" coords="25,7,80,29" href="http://www.wp.pl"
target="blank">
  <area shape="rect" coords="24,40,77,63" href="http://www.helion.pl"
target="blank">
  <area shape="rect" coords="21,74,82,94" href="http://www.onet.pl"
target="blank">
  <area shape="rect" coords="21,107,82,128" href="http://www.interia.pl"
target="blank">
</map>
```

Powyższy listing zawiera fragment kodu HTML, który został automatycznie wygenerowany po zdefiniowaniu obszarów aktywnych.

5.13. Zapisywanie obrazów na potrzeby witryny internetowej

5.13.1. Zasady zapisywania grafiki

Najczęściej stosowane formaty plików graficznych wykorzystywane na stronach internetowych to: JPG, GIF oraz PNG.

Wybór formatu pliku

Obrazy proste, które mają niewielką liczbę kolorów oraz nie zawierają cieniowania i przejść tonalnych, zapisujemy w formacie GIF. Przy zapisie do tego formatu wybieramy jak najmniejszą liczbę kolorów, aby nie pogorszyć jakości obrazu. W tym formacie może być zapisane np. logo, które z reguły zawiera dwa, trzy kolory. Aby obraz nie stracił na jakości, przy zapisie wybieramy 8 – 16 kolorów.

Do zapisu zdjęć, grafik zawierających przejścia tonalne itp. należy używać formatu JPG z odpowiednim procentem (stopniem) kompresji. Optymalizacja w granicach 60 – 70% zwykle jest poziomem kompresji pozwalającym zachować dobrą jakość obrazu.

W formacie PNG zapisujemy grafiki zawierające efekt przezroczystości lub grafiki o wysokiej jakości. Mimo że format GIF również obsługuje funkcję przezroczystości, w formacie PNG obraz wygląda lepiej, a jego krawędzie są gładsze.

Optymalizacja plików graficznych

Optymalizacja plików graficznych to działania prowadzące do otrzymania pliku o możliwie małym rozmiarze przy zachowaniu możliwie najlepszej jakości (najlepszego wyglądu). Pozytywnym efektem optymalizacji jest oszczędność miejsca na serwerze oraz szybsze ładowanie się stron.

Pliki zapisywane w formacie GIF optymalizujemy przez minimalizowanie liczby kolorów, natomiast pliki JPG — przez procentowe określenie poziomu kompresji.

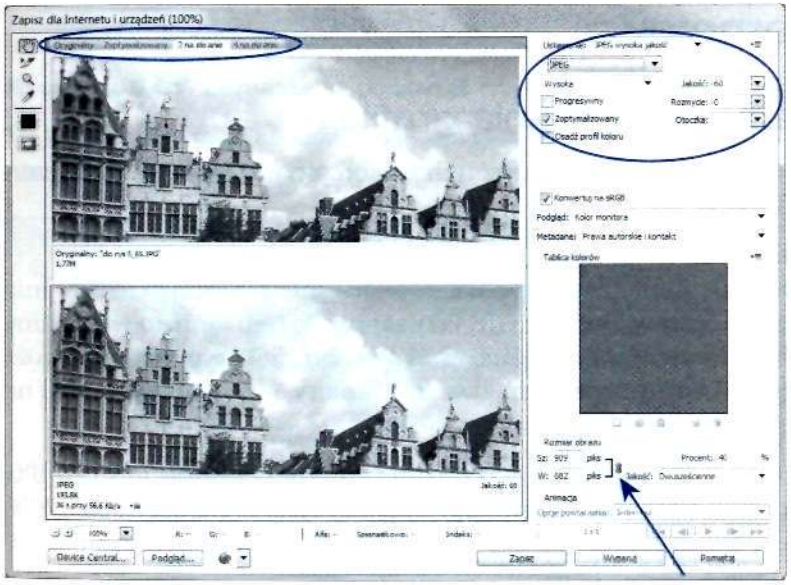
Większość programów graficznych przeznaczonych do tworzenia grafiki na potrzeby stron internetowych ma funkcje zapisu grafiki z możliwością optymalizacji pliku.

Istnieją również wyspecjalizowane programy do optymalizowania plików graficznych, np. Radical Image Optimization Tool.

5.13.2. Zapisywanie obrazów w programie Photoshop

Aby zapisać obraz utworzony na potrzeby internetu w Photoshopie, należy wybrać *Plik/Zapisz dla Internetu i urządzeń*. Pozwoli to na kontrolowanie rozmiaru pliku i jakości obrazu.

Zostanie otwarte okno, w którym można ustawić opcje optymalizacji zapisywania pliku oraz opcje przeglądania zoptymalizowanych obrazów (rysunek 5.63).



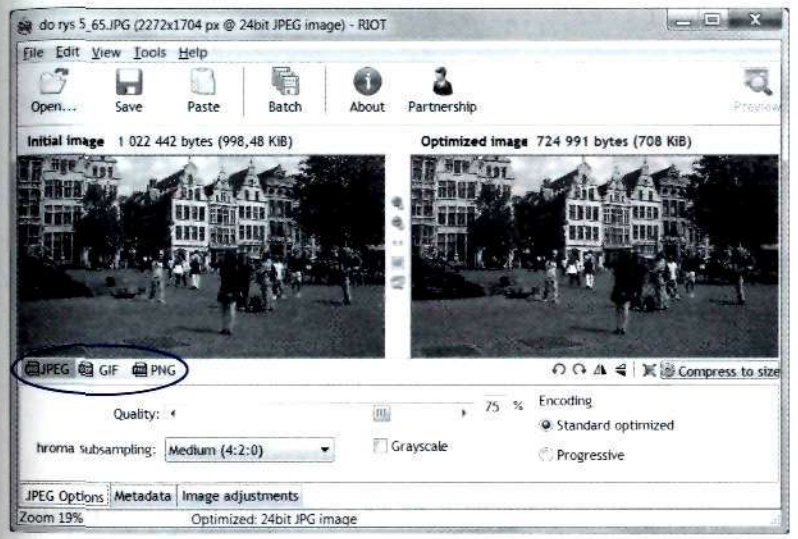
Rysunek 5.63. Okno Zapisz dla Internetu i urządzeń

W górnej części okna na zakładkach ustalamy sposób wyświetlania obrazu. Na przykład wybranie zakładki *2 na ekranie* spowoduje wyświetlenie oryginalnej pracy oraz obrazu optymalizowanego. Z lewej strony u góry został umieszczony przybornik. Po prawej stronie znajduje się zestaw narzędzi, za pomocą których określamy liczbę kolorów (format GIF) lub stopień kompresji (format JPG) oraz format zapisywanego pliku.

W dolnej części okna można określić wielkość grafiki, która zostanie wykorzystana na stronie. Grafika powinna być dopasowana do układu strony, dlatego w opcji *Rozmiar obrazu* można podać jej wymagane wymiary. Rozdzielczość wyświetlania zostanie automatycznie ustawiona na 72 ppi. Klikając przycisk *Zapisz*, zapisujemy zoptymalizowany plik w wybranym miejscu na dysku.

5.13.3. Program Radical Image Optimization Tool

Program ten umożliwia zapisanie zoptymalizowanej wersji pliku graficznego w jednym z trzech formatów: JPG, GIF lub PNG. Po uruchomieniu programu wybieramy plik graficzny i otwieramy go (rysunek 5.64).



Rysunek 5.64. Optymalizacja pliku

Wybieramy format pliku wynikowego, klikając odpowiednią ikonę pod zdjęciem. W dolnej części okna ustawiamy wybrane opcje optymalizacji.

Format JPG

Opcje optymalizacji pliku dla formatu JPG zostały pokazane na rysunku 5.65.



Rysunek 5.65. Opcje optymalizacji pliku JPG

Quality — określa stopień kompresji obrazu dla formatu JPG.

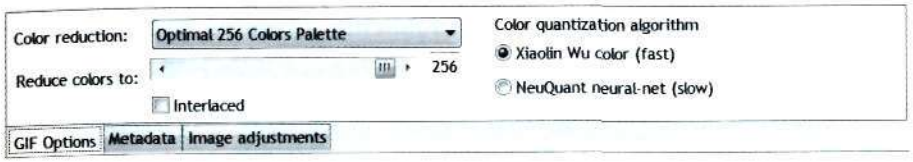
Chroma subsampling — technika kodowania obrazów JPG. Prowadzi do zmniejszenia rozmiaru pliku kosztem jego jakości.

Grayscale — zapis obrazu w skali szarości.

Progressive — lepsza technika kompresji. Pozwala zmniejszyć rozmiar pliku przy zachowaniu jego wysokiej jakości.

Format GIF

Opcje optymalizacji pliku dla formatu GIF zostały pokazane na rysunku 5.66.



Rysunek 5.66. Opcje optymalizacji pliku GIF

Color reduction — pozwala zredukować liczbę kolorów. Możliwość wyboru jednego z wbudowanych ustawień.

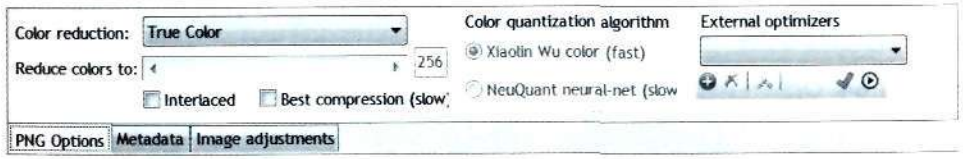
Reduce colors to — pozwala zredukować liczbę kolorów. Liczba kolorów wybierana przez użytkownika.

Interlaced — zapis z przeplotem.

Color quantization algorithm — algorytm redukujący liczbę kolorów w obrazie.

Format PNG

Opcje optymalizacji pliku dla formatu PNG zostały pokazane na rysunku 5.67.



Rysunek 5.67. Opcje optymalizacji pliku PNG

Color reduction — zmniejszanie liczby kolorów. Możliwość wyboru jednego z wbudowanych ustawień.

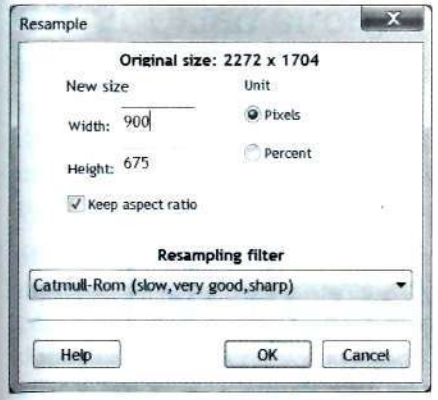
Reduce colors to — zmniejszanie liczby kolorów. Liczba kolorów wybierana przez użytkownika.

Interlaced — zapis z przeplotem.

Best compression (slow) — maksymalny stopień kompresji.

Color quantization algorithm — algorytm redukujący liczbę kolorów w obrazie.

Po wybraniu formatu pliku wynikowego i ustaleniu opcji optymalizacji można określić rozmiar grafiki, wybierając z menu *Edit/Resample* (rysunek 5.68).



Rysunek 5.68. Opcje rozmiaru grafiki

Zoptymalizowany plik zapisujemy w określonej lokalizacji, stosując skrót klawiszowy *Ctrl+S*.

PYTANIA I POLECENIA KONTROLNE

1. Podaj charakterystyczne cechy grafiki wektorowej.
2. Podaj charakterystyczne cechy grafiki rastrowej.
3. Jakie parametry wpływają na jakość obrazu w grafice rastrowej?
4. Czym charakteryzuje się model barw RGB?
5. Gdzie najczęściej jest stosowany model barw CMYK?
6. Wymień zalety wykorzystywania do zapisu obrazu formatu PNG.
7. Jakich narzędzi dostępnych w programie graficznym Adobe Photoshop można użyć w celu usunięcia ze zdjęcia zbędnych elementów?
8. W jakim celu w programach graficznych są stosowane warstwy?
9. Wymień kilka metod tworzenia tła strony internetowej.
10. Na czym polega optymalizacja plików graficznych przy ich zapisywaniu na potrzeby strony internetowej?

ZADANIE

Utworzone w rozdziale 5 pliki zawierające grafikę zostaną wykorzystane do przygotowania witryny internetowej zawierającej informacje na temat szkoły. W tym celu należy je zoptymalizować. Uczestnicy projektu przeanalizują i wybiorą te grafiki, które mogą zostać użyte podczas tworzenia witryny internetowej.

6

Graficzny projekt strony internetowej

6.1. Opracowanie projektu

Przystępując do tworzenia projektu graficznego strony internetowej, powinniśmy dążyć do uzyskania przejrzystej i łatwej w obsłudze struktury. Użytkownik, który znajdzie się na stronie, nie powinien mieć problemu z określeniem, czego dotyczy witryna, jaka jest jej nazwa, jakie kategorie podstron są dostępne, jakie są opcje wyboru i w jaki sposób można czegoś poszukać.

Dobrym rozwiązaniem jest zastosowanie się do ogólnie przyjętych zasad. Ułatwi to użytkownikom poruszanie się po witrynie i szukanie odpowiednich informacji.

Standardami są:

- umiejscowienie logo w lewym górnym rogu;
- panel nawigacji łatwy do znalezienia, najlepiej u góry strony;
- wyróżnienie elementu, który jest linkiem;
- zamieszczanie zdjęć bardzo dobrej jakości;
- umieszczanie w stopce danych kontaktowych i informacji o prawach autorskich.

6.1.1. Użyteczność strony internetowej

Użyteczność strony sprowadza się do:

- intuicyjnej nawigacji,
- łatwości wyszukiwania informacji,
- komunikacji zrozumiałej dla użytkownika.

6.1.2. Układ strony

Przy projektowaniu stron internetowych przyjmuje się, że standardowa rozdzielczość ekranu komputera to 1280×800 lub 1024×768 px. Stosownie do tych zasad standardowa szerokość projektowanej strony powinna wynosić 960 – 999 px, zaś wysokość efektywna około 600 px. Zwykle wysokość strony się zmienia, ale okno, otwarte jako pierwsze, powinno zawierać najistotniejsze informacje.

Standard stosowany na stronach internetowych to logo jako główny element strony umieszczony w górnym lewym rogu. Menu jest umieszczane z prawej strony logo lub pod nim. Kolejny element spotykany na większości stron to tzw. slider. Jest to kilka dobrej jakości i dobrze wyeksponowanych zdjęć, które zmieniają się automatycznie. Tych kilka zdjęć ma ogromny wpływ na pozytywny odbiór strony. Slider to ważne narzędzie, które umożliwia komunikację z użytkownikiem. Tu umieszczamy tekst i zdjęcia, które są najważniejsze w odbiorze witryny. Jeżeli slider został dobrze zaprojektowany, to bez konieczności zaglądania na kolejne podstrony użytkownik dowie się, czego dotyczy witryna. Poniżej tego elementu powinna znaleźć się treść strony, najczęściej w postaci kolumn z tekstem. Układ kolumn przyczynia się do tego, że informacje są uporządkowane i łatwiej przyswajane przez odbiorcę.

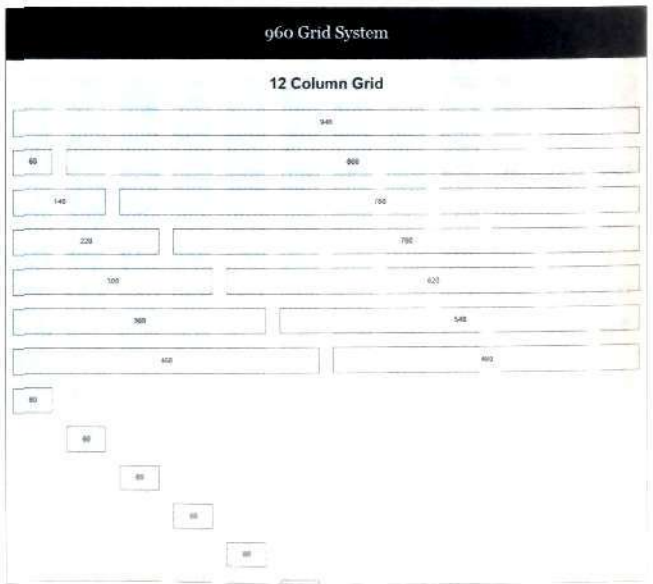
Ostatnim elementem strony jest stopka. Czasami jest rozbudowana i zawiera wiele różnych informacji, np.: menu (aby nie trzeba było wracać na górę strony), ikony (np. z łączami do akcji społecznościowych), opcje logowania, czyli wszystko to, co jest potrzebne na stronie, ale nie jest tak istotne, żeby zamieszczać to w głównej jej części.

Projektowanie wyglądu strony ułatwia tzw. *960 GRID SYSTEM* (dostępny na stronie *http://960.gs*). Zawiera on szablony pomagające w projektowaniu strony o szerokości 960 px (rysunek 6.1).

Szablony dają możliwość projektowania stron z podziałem na wiele kolumn. Standardowo GRID proponuje podział na 12 lub 16 kolumn.

Opierając się na takim szablonie i stosując się do zaproponowanych podziałów, w prosty sposób utworzymy uporządkowaną stronę.

Z podanej strony można pobrać zestaw szablonów, klikając przycisk *Big ol' DOWNLOAD button* — w folderze *templates*



Rysunek 6.1. Szablon 960 GRID SYSTEM

znajdzie się zestaw szablonów przygotowanych dla różnych aplikacji. W celu wykorzystania szablonu w programie Photoshop wybieramy folder *photoshop*, a w nim szablon dla 16 kolumn. Zostanie wówczas otwarty szablon pomocny w tworzeniu graficznego projektu strony internetowej. Zawiera on linie pomocnicze, za pomocą których łatwo można budować wybrane segmenty strony.

6.2. Projektowanie strony

Mając przygotowany na kartce szkic strony, przechodzimy do jej projektowania w programie graficznym.

Przykład 6.1

Nasz projekt graficzny zostanie utworzony w programie Photoshop, w którym został już otwarty szablon GRID. Abyśmy mogli wygodniej pracować, powiększamy obszar roboczy, wybierając z menu *Obraz/styl* i w otwartym oknie ustawiamy szerokość na 1280 px, a wysokość np. na 2000 px, tak aby nie przejmować się długością projektowanej strony.

Przyciemnione kolumny widoczne w szablonie są niepotrzebne, więc można je usunąć po odblokowaniu warstwy *16col Grid*. Następnym krokiem powinno być utworzenie nowej warstwy i usunięcie warstwy *Background*. Tak przygotowany szablon jest gotowy do dalszej obróbki.

Rozpoczynamy tworzenie projektu graficznego.

Wybieramy narzędzie *Prostokąt* i rysujemy prostokąt w górnej części szablonu. Gdy zaznaczymy narysowany obiekt i naciśniemy klawisze *Ctrl+T*, w pasku *Właściwości* będziemy mogli dokładnie określić w pikselach wielkość prostokąta (np. *640,00 piks*; rysunek 6.2). Po ustawieniu odpowiednich wielkości zatwierdzamy je klawiszem *Enter*.

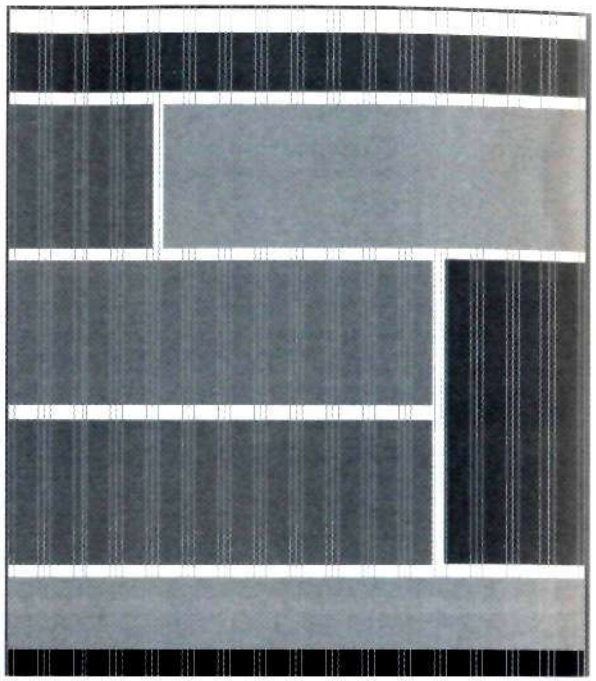


Rysunek 6.2. Ustawienie rozmiaru elementu

Następnie można powielić utworzony prostokąt i dostosować odpowiednio jego wysokość. Zaznaczamy prostokąt i, trzymając wciśnięty klawisz *Alt*, przeciągamy figurę w dół. Po skopiowaniu prostokąta zostanie utworzona kolejna warstwa. Na pasku *Właściwości* ustalamy wysokość nowego elementu (np. 100 px). Zatwierdzamy ustawienia. Nowy element przenosimy w górę, dosuwając go do pierwszej figury. Zmieniamy kolorystykę, aby nowy element odróżniał się od poprzedniego. To będzie obszar logo.

Podobnie tworzymy kolejny element, np. o wysokości 240 px. Będzie to slider. Zmieniamy kolor prostokąta i odsuwamy go od poprzedniego. W podobny sposób budujemy następne bloki projektu.

Tworzymy jeszcze warstwę tła, przesuwamy ją na dół i wypełniamy kolorem białym, wykorzystując kombinację klawiszy *Shift+F5* i w otwartym oknie wybierając *Kolor pierwszego planu*. Przycinamy wielkość projektu do planowanej wielkości strony — w tym celu wybieramy narzędzie *Kadrowanie* i zaznaczamy obszar strony. Zatwierdzamy go klawiszem *Enter* (rysunek 6.3).

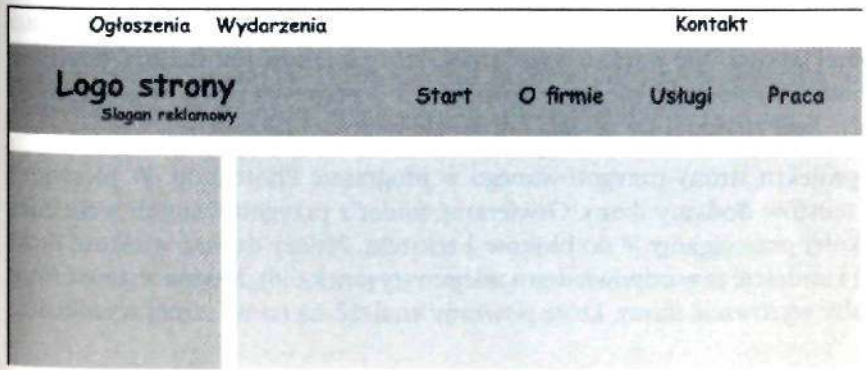


Rysunek 6.3.
Projekt strony w Photoshopie

Jest to pierwszy etap pracy nad projektem graficznym strony. Przenieśliśmy układ zaprojektowany na kartce papieru do programu graficznego.

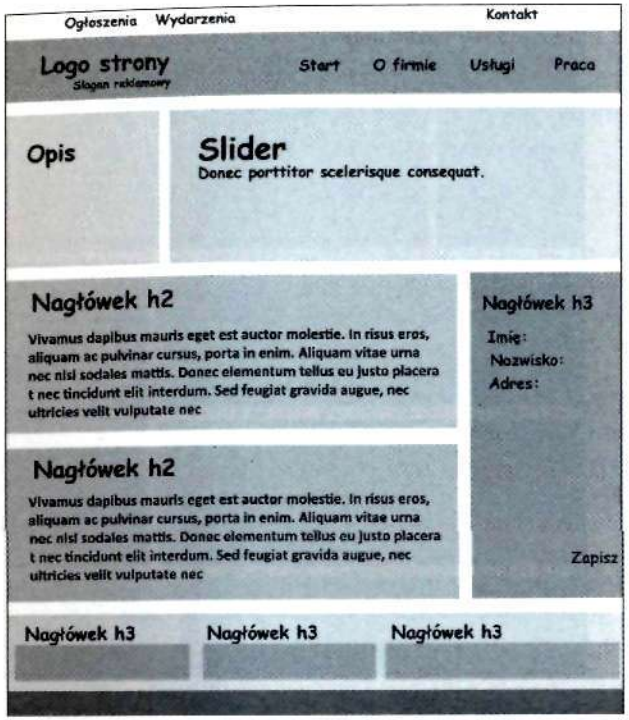
Teraz należy wypełnić projekt treścią. Zaczniemy od wpisania tekstów.

Wpisujemy teksty na pierwszej warstwie (rysunek 6.4). Do wpisywania większych fragmentów można wykorzystać generator tekstu znajdujący się na stronie *www.lipsum.com*, który tworzy przykładowe fragmenty. Jeżeli linie pomocnicze przeszkadzają nam w pracy, można je wyłączyć, wybierając w menu *Widok/Usuń linie pomocnicze*.



Rysunek 6.4. Górny fragment projektowanej strony

Efekt tworzenia typografii strony może wyglądać tak jak na rysunku 6.5. Warstwy zawierające podobne elementy powinny zostać pogrupowane.



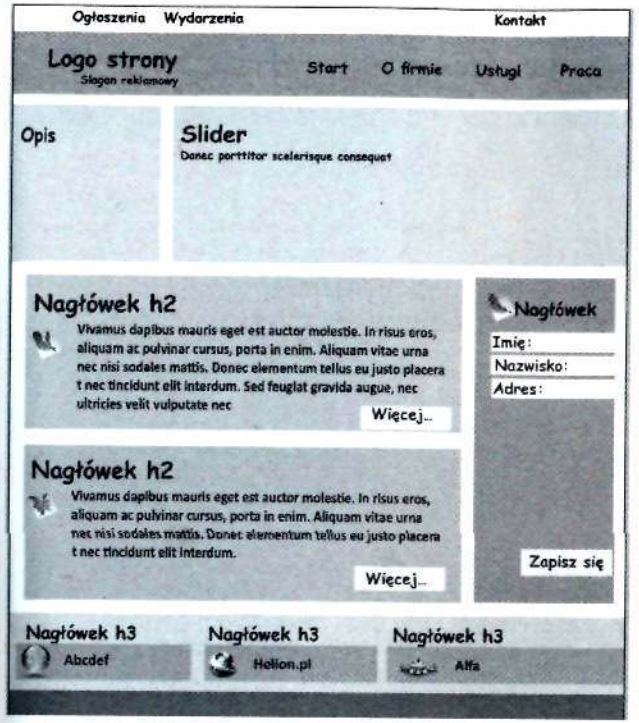
Rysunek 6.5. Przykład typografii strony

Tak przygotowany projekt strony należy uzupełnić elementami graficznymi. Na stronę najczęściej wstawia się ikony, zdjęcia i przyciski.

Ikony wzbogacają stronę wizualnie i ułatwiają nawigację. Ikona powinna być streszczeniem każdej sekcji, powinna wyrażać to, co chcemy przekazać tekstem.

Zdjęcia są bardzo ważnym elementem w sliderze. Zdjęcie użyte w tym bloku powinno być bardzo dobrej jakości. Nie warto używać zdjęć, których jakość jest średnia. Jeżeli nie mamy odpowiednich fotografii, możemy skorzystać z dostępnych stocków graficznych (banków zdjęć). Tam znajdują się zdjęcia naprawdę wysokiej jakości.

Wracamy do projektu strony przygotowanego w programie Photoshop. W pierwszej kolejności do tekstów dodamy ikony. Otwieramy folder z przygotowanymi wcześniej ikonami i po kolei przeciągamy je do bloków z tekstem. Należy dobrać wielkość ikon (np. 48×48 px) i umieścić je w odpowiednim miejscu (rysunek 6.6). Można wstawić linię pomocniczą, aby wyrównać ikony, które powinny znaleźć się na tej samej wysokości.



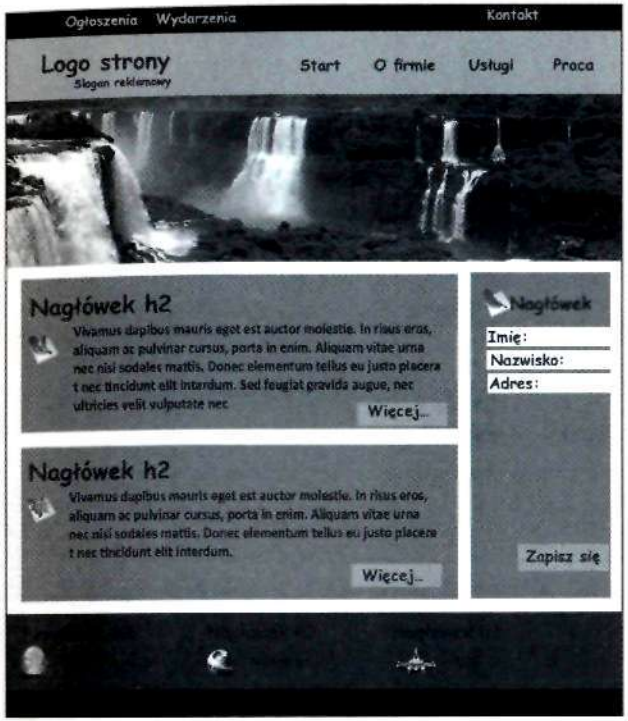
Rysunek 6.6. Dodanie ikon i przycisków do projektu strony

Kolejny rodzaj elementów to przyciski. Tworzymy je, wykorzystując narzędzie *Prostokąt*. Trzeba pamiętać o umieszczaniu takich elementów pod warstwą tekstu.

Ostatni etap pracy to dodanie zdjęć, pokolorowanie bloków i dodanie efektów.

Wybieramy wcześniej przygotowane zdjęcie i przeciągamy je nad warstwę slidera. Dopasowujemy rozmiar zdjęcia do warstwy.

Zmieniamy kolorystykę. Dopasowanie kolorystyki strony wymaga przetestowania. Przykładowa strona może wyglądać tak jak na rysunku 6.7.



Rysunek 6.7. Projekt przykładowej strony

6.3. Dobór palety barw

Istotnym aspektem projektowania witryny internetowej jest optymalny dobór barw.

Zasadą stosowaną przy projektowaniu stron WWW jest wybieranie jak najmniejszej liczby kolorów. Powinny to być trzy lub cztery kolory, czasami pięć. Zwiększenie liczby barw, szczególnie gdy są źle dobrane, powoduje wrażenie niespójności.

Istnieje wiele narzędzi dostępnych w internecie, które pomagają w wyborze właściwej palety barw.

6.3.1. Kuler

Kuler to opracowany przez firmę Adobe program, który można wykorzystać do szybkiego tworzenia palet kolorystycznych używanych na stronach internetowych. Ale przede wszystkim jest to baza wielu tysięcy zestawów kolorów. Można też zaprojektować własne palety kolorystyczne.

Z programu można korzystać bezpośrednio na stronie internetowej. Można też pobrać go na komputer i używać jako samodzielnej aplikacji. Dostępny jest w witrynie *https://kuler.adobe.com/*, gdzie mamy do dyspozycji zestawy kolorów widoczne w menu z lewej

strony. Są to zestawy: najnowsze, najpopularniejsze, najwyżej oceniane i losowo wyświetlane (rysunek 6.8).



Rysunek 6.8. Okno programu Kuler

Aby stworzyć własną paletę kolorów, należy wybrać opcję *Create*. Paletę można wygenerować na podstawie kolorów (*From a Color*) lub z obrazu (*From an Image*). Po kliknięciu opcji *From an Image* można do dalszej pracy wybrać własne zdjęcie lub fotografię z serwisu Flickr.

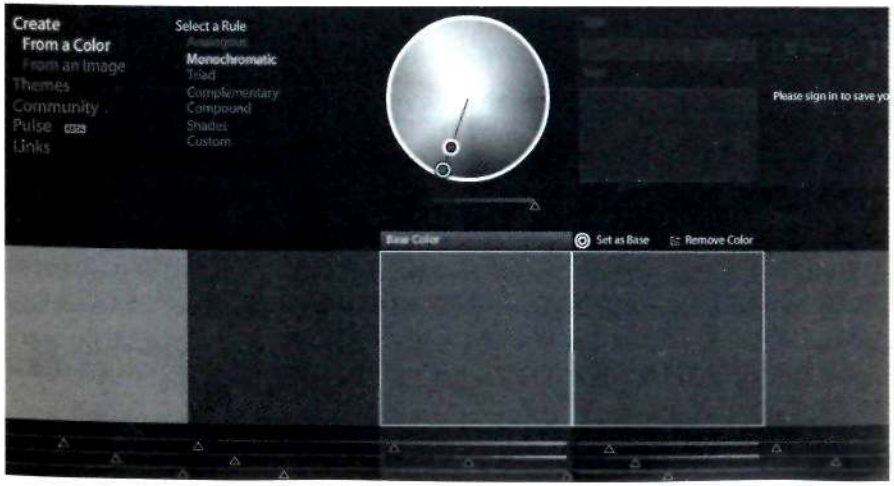
Przykład 6.2
Uruchom narzędzie Kuler. Utwórz paletę z własnego zdjęcia — kliknij przycisk *Upload*, aby wczytać obraz. Na zdjęciu zostaną zaznaczone miejsca próbkowania (rysunek 6.9).



Rysunek 6.9. Kuler — miejsca próbkowania zdjęcia

Miejsca próbkowania można dowolnie przemieszczać na zdjęciu, wskazując kolory, które powinny znaleźć się w palecie. W dolnej części okna wyświetli się paleta wybranych barw.

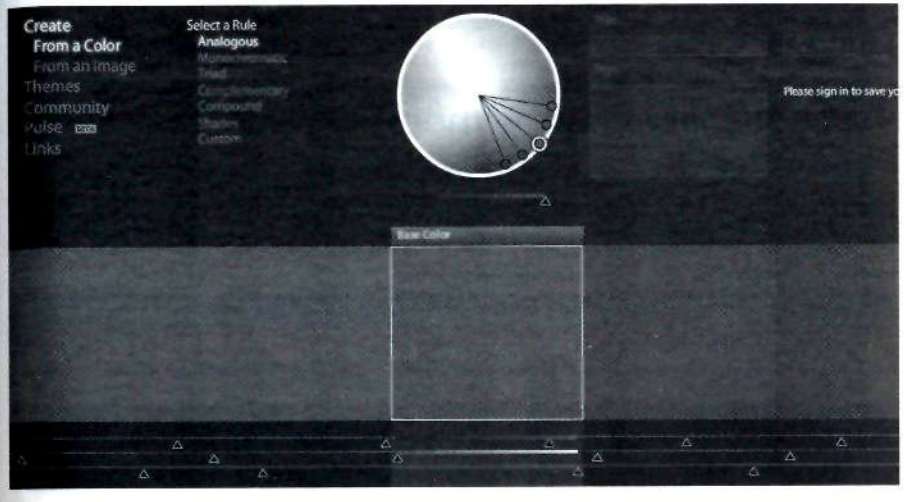
Gdy klikniemy opcję *From a Color*, w oknie pojawi się koło podstawowych kolorów; dostępna będzie opcja *Select a Rule* z rodzajami palet oferowanymi przez program: *Analogous*, *Monochromatic*, *Triad*, *Complementary*, *Compound*, *Shades* i *Custom*. Decydujemy się np. na paletę monochromatyczną. Przesuwając wskaźnikiem po kole, możemy wybrać interesującą nas paletę kolorów (rysunek 6.10).



Rysunek 6.10. Paleta Monochromatic

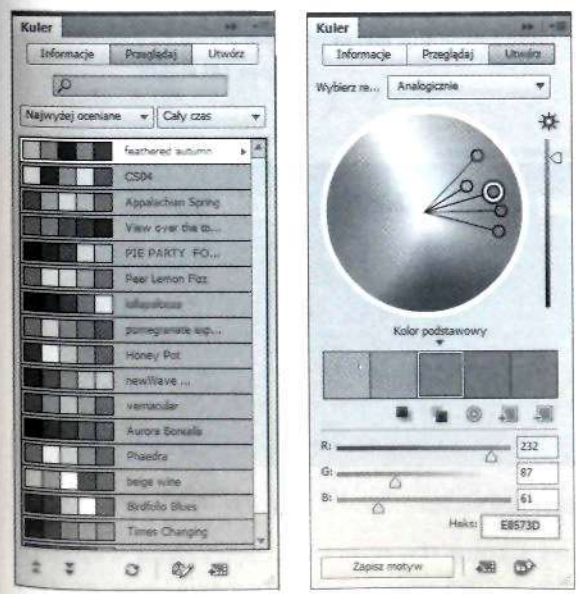
Przesuwając wskaźnik w stronę środka, uzyskujemy łagodniejsze kolory; bliżej obwodu kolory są bardziej wyraziste.

Wybranie palety analogowej spowoduje pojawienie się na kole większej liczby wskaźników. Manipulując nimi, możemy ustalić własną paletę kolorów (rysunek 6.11).



Rysunek 6.11. Paleta Analogous

Kuler jest wbudowany w program Photoshop od wersji CS4. Aby uruchomić to narzędzie w Photoshopie, należy z menu wybrać *Okno/Rozszerzenia/Kuler* (rysunek 6.12).



Rysunek 6.12. Kuler w programie Photoshop

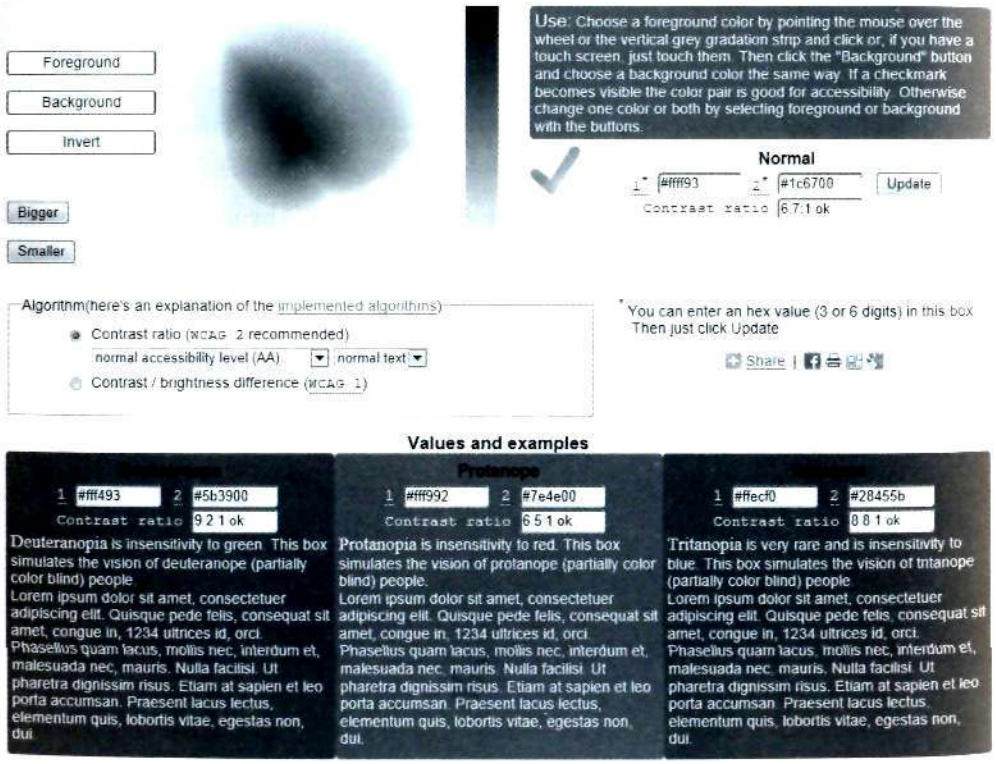
Zakładka *Utwórz* pozwala na tworzenie własnej palety kolorów.

W polu *Wybierz* można określić rodzaj palet oferowanych przez program.

Dobór palety barw zaczynamy od ustalenia jej podstawowego koloru. Zwykle jest to kolor ze zdjęcia będącego dominującym elementem strony, np. tego, które wybraliśmy do warstwy slidera. Kolor można określić, podając jego wartość liczbową lub klikając pipetą w danym miejscu zdjęcia.

6.3.2. Accessibility Color Wheel

Przydatne w doborze kolorów tła oraz tekstu może być narzędzie Accessibility Color Wheel dostępne na stronie *http://gmazzocato.altervista.org/colorwheel/wheel.php*. Po otwarciu strony można określić kolor tła i kolor tekstu, aby przekonać się, czy tekst będzie dobrze widoczny na danym tle. Dodatkowo program poinformuje nas odpowiednim komunikatem, czy prawidłowo dobraliśmy barwy (rysunek 6.13).



Rysunek 6.13. Testowanie koloru tekstu i koloru tła

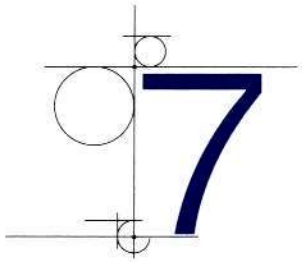
Gdy wybierzemy kolor pierwszoplanowy i kolor tła, w prawej części okna wyświetli się podgląd zaproponowanej kolorystyki, a niżej pojawi się informacja — w postaci heksadecymalnej — o wybranych kolorach oraz o ich kontraście.

PYTANIA KONTROLNE

1. Jakich zasad należy przestrzegać podczas tworzenia projektu graficznego strony internetowej?
2. Dlaczego standardowa szerokość projektowanej strony internetowej powinna wynosić 960 – 999 px?
3. Jakiego rodzaju elementy graficzne powinny się znaleźć na projektowanej stronie?
4. Do czego podczas tworzenia projektu graficznego strony internetowej można wykorzystać program Kuler?

ZADANIE

Uczestnicy projektu opracują projekt graficzny strony głównej dla witryny zawierającej informacje na temat szkoły oraz projekt graficzny, który będzie obowiązywał na pozostałych stronach witryny. Zostanie zaprojektowana typografia strony. Będzie ona uzupełniona elementami graficznymi typu: ikony, zdjęcia i przyciski. Uczestnicy projektu, korzystając z poznanych narzędzi, dopasują kolorystykę strony do umieszczonych na niej elementów graficznych.



Animacja na stronie internetowej

Animacja komputerowa wiąże się z tworzeniem ruchomych obrazów przy wykorzystaniu komputera. Jest to połączenie grafiki i jej zaprogramowanego ruchu.

Przy użyciu narzędzi, które są dostępne na komputerze, tworzone są obiekty nieznacznie różniące się od siebie. Jeżeli zostaną one wyświetlone jeden po drugim dostatecznie szybko, to będziemy mieć wrażenie, że obraz jest ruchomy. Aby animacja była płynna, obrazy powinny zmieniać się z odpowiednią częstotliwością liczoną w klatkach na sekundę — **fps** (ang. *frames per second*). Liczba klatek dająca wrażenie płynnego ruchu to 18 – 24 na sekundę. W filmach standardem jest częstotliwość 24 fps. W animacjach komputerowych przyjmuje się 25 – 35 klatek na sekundę.

7.1. Metody tworzenia animacji

Animacja obiektu może dotyczyć jego położenia, kształtu i rozmiaru oraz barwy i tekstury. W tworzeniu animacji komputerowych stosowane są różne metody. Są to:

- **Animacja poklatkowa** — kolejne klatki animacji tworzone są po kolei. Każda klatka zawiera pełny obraz. Jest to metoda czasochłonna i mało efektywna.
- **Animacja z zastosowaniem klatek kluczowych** — tworzone są wybrane klatki; pozostałe są interpolowane przez komputer na podstawie ramek kluczowych.
- **Skrypty** — opisują dynamiczną zmianę właściwości obiektów, np. położenia czy wyglądu, w wyniku zaistniałego zdarzenia.

7.1.1. Flash

Flash to technologia tworzenia zaawansowanych animacji komputerowych z wykorzystaniem klatek kluczowych. Pozwala na tworzenie stron internetowych, których kod źródłowy jest niedostępny dla użytkowników przeglądarek internetowych. Często jednak prostsza i szybsza w użyciu od technologii Flash okazuje się technologia HTML.

Powstałe pliki, zwane *plikami Flash*, są zapisywane w formacie SWF. Popularne programy do tworzenia animacji to: Adobe Flash, Alligator Flash Designer, SWiSH Max4, Hippo Animator.

7.1.2. ActionScript

ActionScript to obiektowy język programowania wykorzystywany do sterowania animacjami Flash. Kod jest tworzony w trakcie projektowania animacji i zapisywany w pliku razem z animacją. Zwykle język ten wykorzystywany jest przy zdarzeniach związanych z myszą (kliknięcie, najechanie kursorem, usunięcie kursora). ActionScript nadaje animacjom interaktywność.

7.1.3. Animowany GIF

Animowany GIF to seria połączonych obrazów zapisana w pliku GIF. Po uruchomieniu takiego pliku obrazy są wyświetlane kolejno jeden po drugim, tworząc animację. Do tworzenia animowanych GIF-ów wykorzystywana jest metoda animacji poklatkowej.

7.2. Tworzenie animacji na potrzeby strony internetowej

Strona internetowa powinna przyciągać uwagę użytkownika, ale nadmierna liczba efektów specjalnych czy animacji często przeszkadza w szukaniu konkretnych informacji. Animacje „ożywiają" strony internetowe, ale należy je tworzyć tak, aby były integralną częścią witryny.

7.2.1. Podstawowe formy animacji

Baner reklamowy
Animowany baner reklamowy jest bardzo dobrą formą reklamy w internecie. Zaprojektowanie przyciągającego uwagę banera nie jest łatwe, ponieważ taki element musi spełniać wiele wymagań. Powinien być niewielki, przyciągać wzrok i wywoływać zaciekawienie. Metod na wzbudzenie zainteresowania jest wiele — np. dobranie odpowiednich kolorów i ciekawych tekstów, efektowna animowana grafika. Ważną cechą takiej animacji jest jej dynamika.

Top na stronę
Animowany top tworzony na stronę internetową spełnia głównie funkcje reprezentacyjną i ozdabiającą. Tego typu animacja powinna być dostosowana do strony i tworzyć z nią graficzną całość.

Animowane przyciski i menu
Animowane przyciski to jedna z form tworzenia przycisków związanych z interakcją. W momencie, gdy umieścimy kursor myszy na takim przycisku, zmienia on swój wygląd,

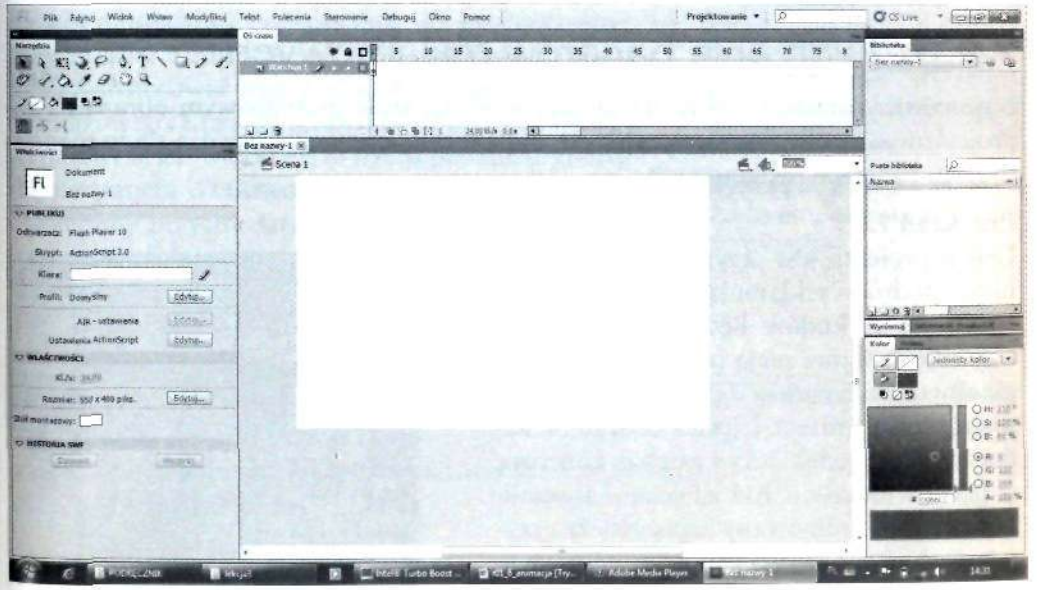
a jego kliknięcie wywołuje jakieś działanie. Na stronie mogą pojawić się pojedyncze przyciski lub przyciski będące elementami menu nawigacyjnego.

Animowane mapy
Przykładem takiej mapy jest animowana mapka Polski. Najechanie kursorem na obszar województwa powoduje zmianę koloru, a kliknięcie — powiększenie województwa. Dla każdego województwa można zdefiniować otwieranie wybranej strony. To animacja prosta graficznie i często używana na różnych stronach.

7.2.2. Tworzenie animacji

Do tworzenia animacji komputerowych można wykorzystywać różne programy. Narzędziem, które najlepiej spełnia wymogi projektowania animacji flashowej, jest program Adobe Flash. Animacje są w nim tworzone z wykorzystaniem grafiki wektorowej. Powstające pliki są zapisywane z rozszerzeniem *.swf*. Można je umieszczać na stronie internetowej i odtwarzać po zainstalowaniu wtyczki Adobe Flash Player. Oprócz prostych animacji w programie można tworzyć animacje interaktywne. Przy użyciu programu są tworzone np. dostępne w internecie gry Flash, reklamy oraz prezentacje internetowe. Za pomocą programu Adobe Flash można też tworzyć strony internetowe.

Ma on rozbudowany interfejs użytkownika. Po uruchomieniu programu zostanie otwarte okno robocze pokazane na rysunku 7.1.



Rysunek 7.1. Okno programu Flash

Okno główne programu składa się z kilku paneli. Z lewej strony, u góry, umieszczony został panel *Narzędzia*, który zawiera podstawowe narzędzia do tworzenia, edytowania i animowania obrazów. Niżej znajdujący się panel *Właściwości* zawiera właściwości

tworzonego lub edytowanego obiektu. W górnej części okna znajduje się bardzo ważny dla animacji panel *Oś czasu*. Pokazuje on warstwy, z których składa się animacja, oraz czas trwania animacji.

Standardowa częstotliwość odtwarzania klatek dla obszaru Europy wynosi **24 klatki na sekundę**. Ta właśnie wielkość jest wyświetlana w dolnej części panelu *Oś czasu* (Częstość klatek). Można ją zmieniać, klikając w obszarze okienka z wartością i wpisując inną prędkość. Obok jest wyświetlana informacja o czasie trwania całej animacji (Upłynęło czasu).

W środkowej części okna programu znajduje się **stół montażowy**. Jest to biały prostokąt wyznaczający obszar widoczny dla odbiorców tworzonej animacji. Właściwości stołu montażowego można zobaczyć w panelu *Właściwości*. Do zmiany jego wielkości służy przycisk *Edytuj*, po kliknięciu którego można wpisać nowe wymiary. Wielkość wyświetlanego stołu montażowego (widok) można dostosować do własnych potrzeb. W jego prawym górnym rogu znajduje się lista przybliżeń. Wybierając opcję *Dopasuj*, powiększymy stół montażowy do rozmiarów, które mieszczą się w oknie. Dla bardzo dużych przybliżeń można wybrać powiększenie nawet 800%.

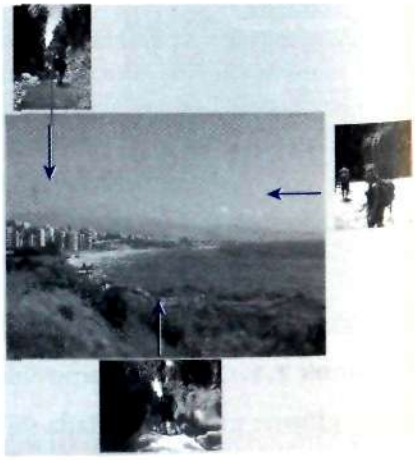
Kolejnym ważnym panelem, znajdującym się z prawej strony, u góry okna, jest *Biblioteka*. Znajdują się w nim wszystkie pliki zaimportowane do programu, które zostaną wykorzystane do tworzenia animacji. Poniżej panelu *Biblioteka* jest panel *Wyrównaj*, który zawiera opcje związane z ustawianiem położenia obiektów względem siebie.

7.2.3. Tworzenie banerów reklamowych

Najprostsza animacja polega na umieszczeniu na stole montażowym obrazów i zaprojektowaniu ich przemieszczania się. Taka forma animacji sprawdza się w różnego rodzaju banerach reklamowych.

Przykład 7.1

Celem projektu jest przygotowanie banera biura podróży reklamującego wyjazdy do egzotycznych krajów. Będziemy animować trzy zdjęcia, które mają pojawić się na oryginalnym tle. Rysunek 7.2 pokazuje pozycję początkową animacji (poza obszarem widocznym), rysunek 7.3 — pozycję końcową (zdjęcia widoczne). Pod zdjęciami zostanie umieszczony animowany napis. Należy przygotować cztery zdjęcia zapisane w formacie JPG: trzy zostaną wykorzystane do animacji, czwarte będzie jej tłem.



Rysunek 7.2. Stan początkowy

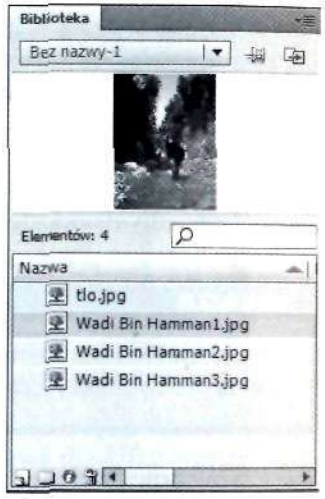
Tworzenie animacji rozpoczynamy od zaimportowania potrzebnych plików. W tym celu wybieramy z menu *Plik/Importuj*. Zostanie

wyświetlona lista czterech różnych sposobów importowania. Najczęściej korzystamy z opcji *Importuj na stół montażowy* lub *Importuj do biblioteki*. Różnica polega na tym, że na stół montażowy importujemy jeden plik lub plik programu Photoshop, który jest przystosowany do pracy z Flashem i zostanie umieszczony dokładnie w wybranej warstwie, natomiast w przypadku importowania do biblioteki pliki zostaną umieszczone w bibliotece i będzie można z nich korzystać podczas tworzenia różnych efektów animacji.



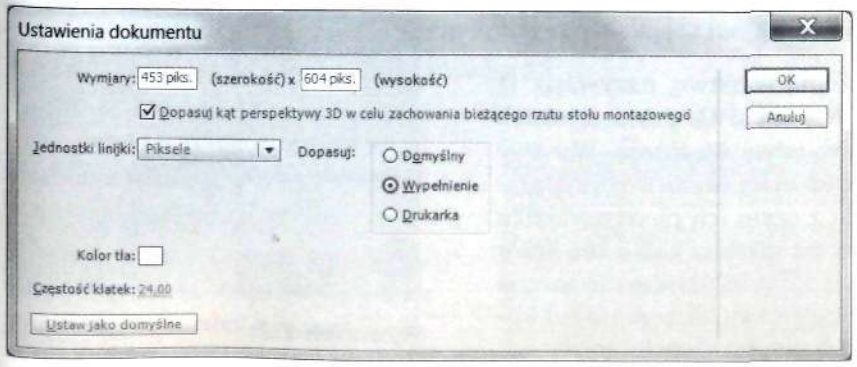
Rysunek 7.3. Stan końcowy

Wybieramy opcję *Importuj do biblioteki*, następnie znajdujemy folder i określamy, które pliki powinny zostać zaimportowane. Zaimportowane pliki zostaną umieszczone w panelu *Biblioteka* (rysunek 7.4). Wybierając z listy nazwę pliku, można w podglądzie sprawdzić jego zawartość.



Rysunek 7.4. Pliki zaimportowane do biblioteki

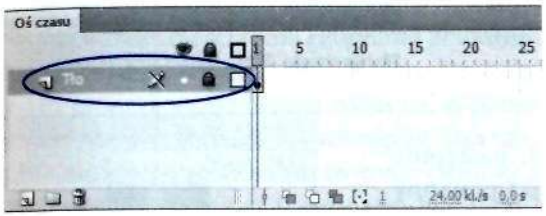
Przystępujemy do przygotowania tła animacji. W tym celu w panelu *Biblioteka* wybieramy plik ze zdjęciem, które będzie stanowiło tło, i przenosimy to zdjęcie na stół montażowy. Przeniesiony plik ma najczęściej inny rozmiar niż ustawienia stołu, w związku z czym należy dostosować jego wielkość do wielkości stołu montażowego. Klikamy w obszarze stołu montażowego (jeżeli nie jest widoczny, to trzeba przesunąć grafikę tła) i w panelu *Właściwości* przy opisie jego rozmiaru wybieramy przycisk *Edytuj*. Zostanie otwarte okno *Ustawienia dokumentu*, w którym zaznaczamy opcję *Wypełnienie* (rysunek 7.5). Adobe Flash sam dopasuje rozmiar grafiki tła do rozmiaru stołu montażowego.



Rysunek 7.5. Dopasowywanie rozmiaru grafiki do rozmiaru stołu montażowego

Gdy rozmiary grafiki tła są zdefiniowane, można ją wyrównać względem stołu montażowego. Do tego celu zostanie wykorzystany panel *Wyrównaj*. Zaznaczamy w nim opcję *Wyrównaj do stołu montażowego* i wybieramy *Wyrównaj górne krawędzie* oraz *Rozmieść lewe krawędzie*.

Następnie w panelu *Oś czasu* zmieniamy nazwę utworzonej warstwy na *Tło* (rysunek 7.6) i blokujemy ją, klikając ikonę z kłódką , aby nie dopuścić do przypadkowych zmian tła.



Rysunek 7.6. Zdefiniowanie warstwy tła

Skoro do animacji zostaną dodane trzy zdjęcia i każde z nich będzie animowane w inny sposób, musimy dla każdego zdjęcia utworzyć oddzielną warstwę.

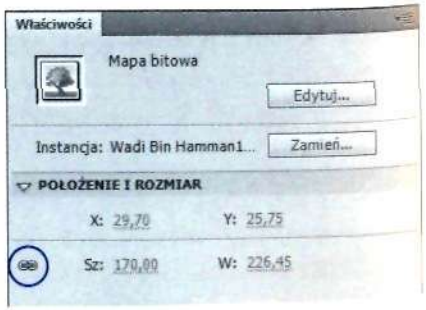
UWAGA

Umieszczenie kilku elementów na jednej warstwie spowoduje, że będą one poruszały się tak, jakby zostały sklejone.

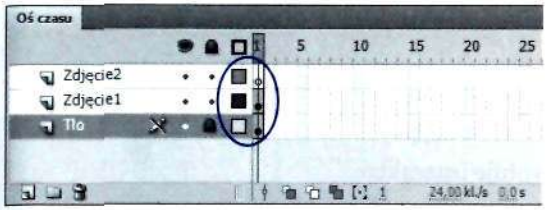
Tworzymy nową warstwę, klikając ikonę *Nowa warstwa* w panelu *Oś czasu*, i zmieniamy jej nazwę, np. na *Zdjęcie1*. Następnie wybieramy w panelu *Biblioteka* odpowiednie zdjęcie i przenosimy je na stół montażowy.

Zmiany położenia grafiki można dokonać za pomocą narzędzia *Zaznaczenie* znajdującego się w panelu *Narzędzia* (ikona), natomiast rozmiary zmieniamy w panelu *Właściwości* (rysunek 7.7). Aby zachować proporcje rozmiaru zmienianego obrazu, należy zaznaczyć ikonę *Powiąż wartości szerokości i wysokości* .

Tworzymy kolejną warstwę, nazywając ją np. *Zdjęcie2*. W panelu *Oś czasu* (rysunek 7.8) widać, że warstwy się różnią. Warstwy *Tło* oraz *Zdjęcie1* mają już utworzoną grafikę, w związku z czym ich pierwsza klatka jest zaznaczona na szaro, a kółko ma kolor czarny. Warstwie *Zdjęcie2* jeszcze nie została przypisana żadna grafika, dlatego tło ma kolor biały, a kółko nie jest wypełnione.



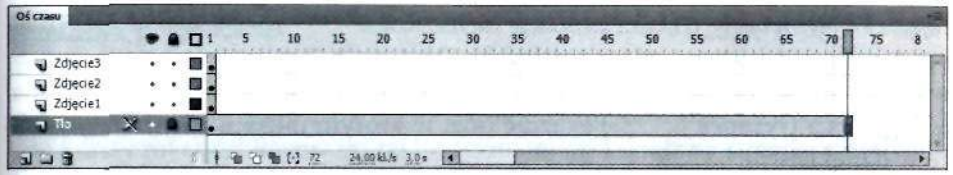
Rysunek 7.7. Zmiana rozmiaru obrazu



Rysunek 7.8. Oznaczenie pierwszej klatki dla różnych warstw

Dodajemy zdjęcie do warstwy *Zdjęcie2* oraz tworzymy kolejną warstwę i umieszczamy na niej ostatnie zdjęcie.

Następnym krokiem będzie dodanie animacji do zdjęć. Zakładamy, że animacja będzie trwała około 3 s, czyli zajmie 72 klatki. Klikając 72. klatkę, dla każdej warstwy możemy dodać zakres czasu. Zaznaczamy w warstwie *Tło* klatkę 72. i wybieramy z menu *Wstaw/Oś czasu/Klatka*. Na osi czasu został zaznaczony czas zarezerwowany na animację (rysunek 7.9).



Rysunek 7.9. Zakres animacji zaznaczony na osi czasu

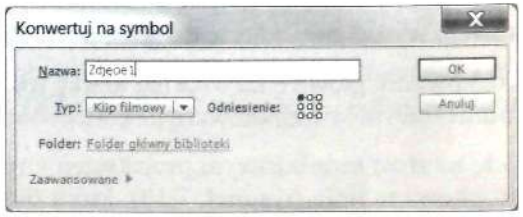
Podobnie postępujemy z pozostałymi warstwami. Klatkę końcową można zaznaczyć klawiszem *F5*. Można równocześnie zaznaczyć klatkę końcową dla kilku warstw i naciskając *F5*, spowodować, że dla wszystkich warstw zostanie zaznaczony czas trwania animacji.

W programie Flash mogą być animowane tylko obiekty przekształcone na symbol. Przekształcenia tego można dokonać następująco:

- chwytając grafikę umieszczoną na stole montażowym i przeciągając ją do panelu *Biblioteka*;
- klikając grafikę prawym przyciskiem myszy i wybierając z menu *Konwertuj na symbol*;
- zaznaczając grafikę i naciskając klawisz *F8*.

Symbol to obiekt graficzny, który jest traktowany jako całość. Może być animowany i przekształcany. Ten sam symbol można wielokrotnie wykorzystać w animacji.

Po wybraniu jednego ze sposobów konwertowania pojawi się okno *Konwertuj na symbol* (rysunek 7.10). Wpisujemy dowolną nazwę symbolu i z listy *Typ* wybieramy *Klip filmowy*.



Rysunek 7.10. Okno Konwertuj na symbol

Tworząc w programie Adobe Flash symbol, mamy do wyboru trzy typy:

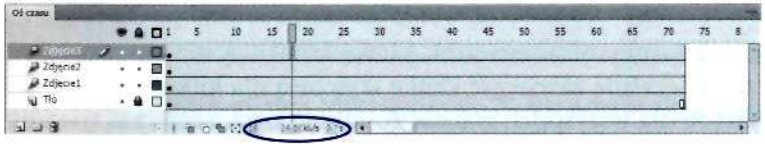
- *Klip filmowy* — przeznaczony jest dla obiektów animowanych. Obiekty te mogą zawierać animację.
- *Przycisk* — wybieramy, gdy obiekt wywołuje interakcję z użytkownikiem.
- *Grafika* — wybieramy, gdy obiekt jest używany wielokrotnie, ale nie będzie animowany.



Rysunek 7.11. Zawartość panelu Biblioteka

Tworzymy symbole dla trzech zdjęć z panelu *Biblioteka*, za każdym razem wybierając typ *Klip filmowy*. Gdy wykonamy te działania, w panelu znajdą się wszystkie utworzone obiekty (rysunek 7.11). Pliki przekształcone na symbol mają ikonę .

Teraz można przystąpić do tworzenia animacji. Klikamy prawym przyciskiem myszy *Zdjęcie1* lub oś czasu warstwy *Zdjęcie1* i wybieramy *Utwórz animację ruchu*. Kolor warstwy zmienił się z szarego na jasnoniebieski. Oznacza to, że warstwa jest animowana. W górnej części osi czasu znajduje się zaznaczona na czerwono głowica (rysunek 7.12). Pokazuje ona, w którym miejscu animacji znajdujemy się w danej chwili.



Rysunek 7.12. Warstwy na osi czasu — głowica

Na dole osi czasu została wyświetlona informacja, że głowica znajduje się w 18. klatce i jest to 0,7 sekundy odtwarzania animacji. W czasie tworzenia animacji to właśnie głowica wyznacza miejsce, w którym projektujemy położenie elementu.

Tworzymy animację również dla warstw *Zdjęcie2* i *Zdjęcie3*.

Jak pamiętamy, celem projektu jest animacja zdjęć z pozycji początkowej (poza obszarem widocznym) do pozycji końcowej (zdjęcia widoczne).

Zaznaczamy warstwę *Zdjęcie1*, ustawiamy głowicę (panel *Oś czasu*) na klatce początkowej i wysuwamy zdjęcie do góry, poza obszar stołu montażowego.

Przesuwamy głowicę na ostatnią klatkę (panel *Oś czasu*) i ustawiamy zdjęcie na stole montażowym w zaplanowanym końcowym położeniu.

Gdy na stole montażowym projektujemy ruch dla fotografii *Zdjęcie1*, pojawia się wykropkowana linia (rysunek 7.13), która określa ścieżkę ruchu zdjęcia. Kropki na linii oznaczają kolejne klatki tworzonej animacji. Ścieżkę ruchu dla wybranego elementu



Rysunek 7.13. Ścieżka ruchu przed edytowaniem



Rysunek 7.14. Ścieżka ruchu po modyfikacji

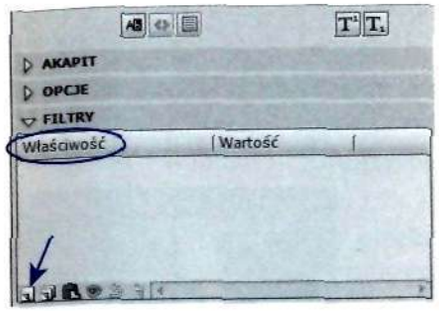
można zmieniać. Jeżeli w panelu *Narzędzia* jest wybrane narzędzie *Zaznaczenie* , to gdy kursor myszy zbliżymy do tej linii, zmieni on kształt na strzałkę z małym łukiem. Można wtedy chwycić ścieżkę ruchu i dowolnie ją wyginać (rysunek 7.14).

Powtarzamy tworzenie animacji dla dwóch pozostałych warstw, zwracając uwagę na położenie głowicy. Po zaprojektowaniu animacji można uruchomić przebieg głowicy, naciskając klawisz *Enter*.

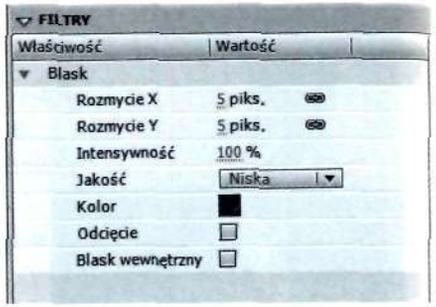
W celu przetestowania i sprawdzenia, jak animacja będzie wyglądała w ostatecznej postaci, należy wybrać z menu *Sterowanie/Testuj film/Test* lub uruchomić animację skrótem klawiszowym *Ctrl+Enter*. Zostanie utworzony plik z rozszerzeniem *swf*. Animacja będzie odtwarzana w pętli.

Do utworzonej animacji dodamy animację tekstu. Tworzymy nową warstwę i nazywamy ją *Tekst*. W panelu *Narzędzia* wybieramy ikonę *Tekst* T . W panelu *Właściwości* wybieramy *Tekst klasyczny*, a następnie *Tekst statyczny* i ustalamy nazwę czcionki. Możemy również wybrać rozmiar oraz kolor czcionki. Klikamy w obszarze stołu montażowego i wpisujemy własny tekst. Wpisany tekst jest aktywny i w panelu *Właściwości* można zmieniać jego parametry. Jeżeli nie jest aktywny, wystarczy wybrać narzędzie *Zaznaczenie* i kliknąć tekst.

Dla tekstu zastosujemy filtr z efektem *Blask*. W dolnej części panelu *Właściwości* dla zaznaczonego tekstu znajduje się opcja *FILTRY* (rysunek 7.15), a w jej dolnym lewym rogu ikona *Dodaj filtr* .



Rysunek 7.15.
Filtry w panelu Właściwości



Rysunek 7.16.
Parametry efektu Blask

Klikając tę ikonę, możemy z listy wybrać efekt *Blask*. Wybrany efekt zostanie automatycznie dodany do tekstu. Zmieniając w panelu *Właściwości* (rysunek 7.16) parametry efektu *Blask*, można dostosować go do własnych potrzeb.

Po przetestowaniu animacji widać, że tekst, który został umieszczony w warstwie znajdującej się na samej górze, przysłania animowane zdjęcie. Nie wygląda to dobrze. W związku z tym w panelu *Oś czasu* zmieniamy kolejność warstw, umieszczając warstwę *Tekst* tuż za warstwą *Tło*. Wykonujemy to, chwytając nazwę warstwy *Tekst* i przesuwając tę warstwę w dół na odpowiednią pozycję.

Ostatnim krokiem będzie animowanie tekstu, tak aby tekst na początku animacji był ukryty, a w trakcie jej trwania stawał się coraz bardziej widoczny. Przekształcamy tekst na symbol. Po tej operacji w panelu *Właściwości* pojawiły się nowe możliwości dla utworzonego symbolu. Dodajemy do tekstu animację (klikamy prawym przyciskiem myszy warstwę *Tekst* i wybieramy *Utwórz animację ruchu*). Ustawiamy głowicę na pierwszej klatce (warstwa *Tekst*), klikamy tekst na stole montażowym i w panelu *Właściwości* rozwijamy panel *Efekty koloru*. Z listy rozwijanej *Styl* wybieramy *Alfa* (przezroczystość) i przesuwamy suwak *Alfa* na 0 (brak widoczności tekstu). Przesuwamy głowicę na ostatnią klatkę i ustawiamy *Alfa* na 100% (pełna widoczność).

UWAGA

Kanał alfa (ang. *alpha channel*) w grafice i animacji komputerowej definiuje przezroczyste obszary grafiki. Jest on zapisywany wewnątrz grafiki jako dodatkowa informacja razem z trzema wartościami barw składowych RGB. Wartość zerowa oznacza całkowitą przezroczystość, natomiast wartość maksymalna — pełne zabarwienie.

Efekt działania zaprojektowanej animacji został pokazany na rysunkach 7.17 – 7.19.



Rysunek 7.17.
Efekt animacji — początek



Rysunek 7.18.
Efekt animacji — w trakcie



Rysunek 7.19. Efekt animacji — koniec

Zadanie 7.1
W zaprojektowanej animacji dodaj efekt powiększania tekstu w trakcie jej trwania.

Zadanie 7.2
Zaprojektuj animowaną reklamę swojej szkoły. Umieść w niej zdjęcia szkoły oraz animowane napisy.

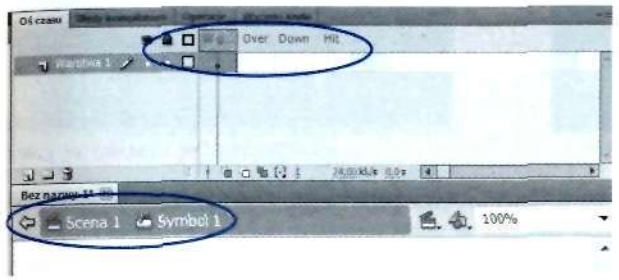
7.2.4. Animowany przycisk

Tworzenie animowanego przycisku to zdefiniowanie reakcji przycisku na zdarzenia pochodzące od myszy. Najczęściej projektujemy zmianę wyglądu przycisku po najechaniu na niego kursorem myszy oraz po odsunięciu kursora.

Przycisk, który będzie reagował na zdarzenia pochodzące od myszy, można utworzyć w prosty sposób. Wykorzystujemy przy tym podstawowe narzędzia technologii Flash.

Przykład 7.2
Utworzymy przycisk na podstawie zwykłego tekstu. W tym celu otwieramy nowy dokument w programie Flash. Wybieramy narzędzie *Tekst*. W panelu *Właściwości* wybieramy

rodzaj, wielkość oraz kolor czcionki i wpisujemy tekst, np. *Kontakt*. Z utworzonego tekstu zrobimy przycisk. Zaznaczamy tekst, klikamy prawym przyciskiem myszy i wybieramy opcję *Konwertuj na symbol*. W otwartym oknie wybieramy typ symbolu *Przycisk* i zatwierdzamy wybór. Utworzony symbol klikamy dwukrotnie, przechodząc z okna *Scena 1* do okna *Symbol 1*. Widoczna u góry oś czasu się zmieniła (rysunek 7.20). Jest to związane z działaniem symbolu typu *Przycisk*.



Rysunek 7.20. Panel Oś czasu dla symbolu Przycisk

Na osi czasu mamy cztery klatki:

- *W górę* — odpowiadającą za wygląd przycisku, gdy nie podejmujemy żadnego działania;
- *Over* — odpowiadającą za wygląd przycisku, gdy najedziemy na niego kursorem myszy;
- *Down* — odpowiadającą za wygląd przycisku, gdy klikamy przycisk;
- *Hit* — określającą obszar, w którym przycisk będzie aktywny.

Zaznaczamy klatkę drugą (*Over*) i naciskamy klawisz *F6* (wstawienie klatki kluczowej). Rysujemy wokół przycisku ciemną ramkę (rysunek 7.21), wybierając *Narzędzie Prostokąt*, a następnie w panelu *Właściwości* kolor i grubość konturu oraz zaokrąglenie rogów.



Rysunek 7.21. Ustawienia dla przycisku w klatce Over

Zaznaczamy klatkę *Down*, naciskamy klawisz *F6*, po czym zmieniamy tekst i kolor czcionki (rysunek 7.22).



Rysunek 7.22. Ustawienia dla przycisku w klatce Down

Po zastosowaniu kombinacji klawiszy *Ctrl+Enter* możemy obejrzeć efekt zaprogramowanego działania. Najechanie kursorem myszy na napis powoduje pojawienie się ramki, a kliknięcie przycisku — zmianę tekstu i jego koloru. Niestety, trzeba najechać dokładnie na napis, żeby uzyskać oczekiwany efekt. Programując klatkę *Hit*, możemy rozszerzyć obszar aktywny, w którym działanie myszą będzie wywoływało interakcję.



Rysunek 7.23. Ustawienia dla przycisku w klatce Hit

Zaznaczamy klatkę *Hit*, naciskamy klawisz *F6* i wokół napisu rysujemy prostokąt w innym kolorze, zaznaczając cały przycisk z niewielkim marginesem (rysunek 7.23).

Wracamy do okna *Scena 1*, klikając zakładkę *Scena 1* (rysunek 7.20), i sprawdzamy działanie przycisku. Przed zapisaniem można jeszcze wybrać w menu *Plik/Podgląd publikowania/HTML* i obejrzeć uzyskany efekt w przeglądarce internetowej.

7.2.5. Powiązanie przycisku z hiperłączem

Po zaprojektowaniu przycisku można powiązać jego działanie z wykonaniem akcji na stronie, np. po jego kliknięciu może się otworzyć inna strona internetowa. Aby było to możliwe, należy w języku ActionScript przygotować skrypt opisujący efekt działania po zaistnieniu zdarzenia kliknięcia przycisku myszą.

Odwoływanie się do obiektów w języku ActionScript jest możliwe tylko przez nazwy. Dlatego wcześniej trzeba nadać nazwy tym obiektom.

Przykład 7.3
Otwieramy dokument z zaprojektowanym przyciskiem i zaznaczamy go. Przechodzimy do panelu *Właściwości* i w opcji *Nazwa instancji* wpisujemy nazwę obiektu, np. *guzik*. W panelu *Oś czasu* zaznaczamy pierwszą klatkę i naciskamy klawisz *F9* (lub wybieramy w menu *Okno/Operacje*). Zostanie otwarte okno języka ActionScript. Możemy wpisać skrypt, który zostanie uruchomiony po kliknięciu przycisku myszą.

Oto skrypt w języku ActionScript 3.0:

```
guzik.addEventListener(MouseEvent.MOUSE_DOWN, mouseDownHandler);
function mouseDownHandler(event:MouseEvent):void
{
    navigateToURL(new URLRequest("http://www.helion.pl/"));
}
```

Gdy uruchomimy animację i klikniemy przycisk *Kontakt*, w nowym oknie zostanie uruchomiona strona *helion.pl*. Jeżeli przycisk ma służyć do otwierania podstron projektowanej witryny, w poleceniu `navigateToURL(new URLRequest("http://www.helion.pl/"));` należy wpisać adres wywoływanej podstrony.

Znacznik Object lub Embed
Pliki Flash (SWF) można umieszczać na stronie za pomocą znaczników `<object>` lub `<embed>` oraz korzystając z języka JavaScript.

Znaczniki `<embed>` i `<object>` stosujemy, gdy w kodzie nie używamy języka JavaScript. Znacznik `<embed>` jest rozpoznawany przez przeglądarkę Internet Explorer, natomiast inne przeglądarki rozpoznają znacznik `<object>`.

Dla znacznika `<object>` polecenie dodania na stronę pliku Flash ma postać:

```
<object width="640" height="480">
<param name="movie" value="menu.swf">
</object>
```

Dla znacznika `<embed>` dodanie na stronę pliku Flash ma postać:

```
<embed src="/menu.swf" width="640" height="480">
</embed>
```

Przykład 7.4

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>
Strona testowa
</title>
</head>
<body bgcolor="c0c0c0">
<object data="Przycisk1.swf" type="application/x-shockwave-flash"
width="200" height="100">
    <param name="movie" value="Przycisk1.swf"></param>
</object>
</body>
</html>
```

7.2.6. Animowane menu

Aby utworzyć bardziej zaawansowane animowane menu, można również skorzystać ze skryptów języka ActionScript.

Przykład 7.5

Animowane menu będzie zawierało zaprojektowany przez nas przycisk, który zareaguje na najechanie na niego kursorem myszy. Przycisk menu będzie wyświetlany na czerwono. Gdy najedziemy kursorem myszy na przycisk, jego kolor zmieni się; odsunięcie kursora spowoduje powrót do wcześniejszego koloru.



Rysunek 7.24. Tekst menu

Otwieramy nowy plik w programie Adobe Flash. Ustawiamy czarny kolor tła. Wybieramy narzędzie *Tekst*. W panelu *Właściwości* ustawiamy rodzaj i wielkość czcionki. Wybieramy dla tekstu kolor czerwony i wpisujemy tekst menu, np. *Animacja* (rysunek 7.24).

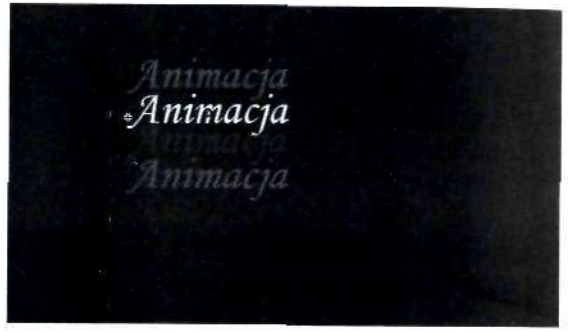
Zaznaczamy wpisany tekst. Aby zamienić tekst na symbol, przeciągamy go do panelu *Biblioteka* lub naciskamy klawisz *F8* (konwersja na symbol). W otwartym oknie wybieramy typ symbolu *Klip filmowy*. Klikamy dwukrotnie tekst na stole montażowym, przechodząc z okna *Scena 1* do okna *Symbol 1*. Zmieniamy nazwę warstwy w panelu *Oś czasu* (np. na *Animacja*). Tworzymy trzy kolejne warstwy. Wracamy na warstwę *Animacja*, klikamy prawym przyciskiem myszy pierwszą klatkę i z listy wybieramy *Kopiuj klatki*. Następnie w warstwie wyżej klikamy prawym przyciskiem myszy pierwszą klatkę i z listy wybieramy *Wklej klatki*. Zmieniamy nazwę tej warstwy na *Animacja1*. Zaznaczamy tekst na warstwie *Animacja1* i przesuwamy go w górę, jak na rysunku 7.25, a następnie zmieniamy jego kolor na biały.

Obydwa teksty przekształcamy na symbole typu *Klip filmowy*. Dla warstw *Animacja* i *Animacja1* tworzymy animację ruchu (klikamy prawym przyciskiem myszy pierwszą klatkę warstwy i wybieramy opcję *Utwórz animację ruchu*). Animację tworzymy dla 20 klatek (klawisz *F5*).



Rysunek 7.25.
Przesunięcie tekstu na kolejnej warstwie

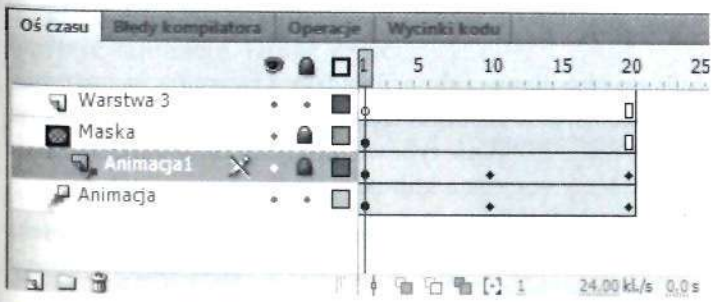
Wybieramy 10. klatkę w obydwu warstwach i naciskamy klawisz *F6* (klatka kluczowa). To samo powtarzamy w odniesieniu do klatki 20. Po zaznaczeniu klatek kluczowych tworzymy animację. Ustawiamy głowicę na klatce 10. i zaznaczamy tę klatkę dla obu warstw. Przesuwamy napisy w dół, tak aby napis biały znalazł się w miejscu napisu czerwonego (rysunek 7.26).



Rysunek 7.26. Przesunięcie tekstów w dół

Przesuwając głowicę po linii czasu, widzimy, że teksty przesuwają się w dół i wracają do góry.

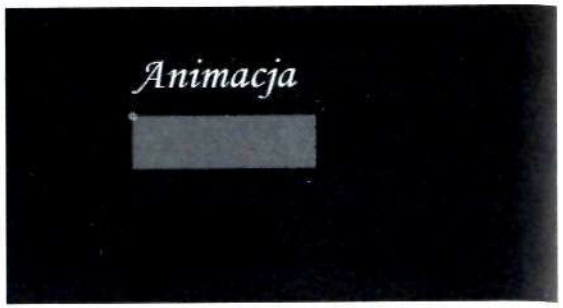
Na kolejnej warstwie utworzymy maskę dla napisów. Zmieniamy nazwę warstwy na *Maska* (rysunek 7.27).



Rysunek 7.27. Warstwy tworzonej animacji

Wybieramy narzędzie *Prostokąt*, zmieniamy jego kolor (np. na niebieski) i na warstwie *Maska* rysujemy prostokąt, który zasłoni dolny napis „Animacja" (rysunek 7.28). Aby prostokąt był dłużej widoczny, na warstwie *Maska* zaznaczamy klatkę 20. i na klawiaturze naciskamy klawisz *F5*.

Klikamy warstwę prawym przyciskiem myszy i wybieramy opcję *Maska*. Przesuwając głowicę po osi czasu, widzimy, że biały napis wyłania się z czarnego tła (rysunek 7.29).



Rysunek 7.28. Zakrycie dolnego tekstu



Rysunek 7.29.
Wysuwanie się napisu z czarnego tła

W celu animacji napisu czerwonego wybieramy warstwę *Animacja*, ustawiamy głowicę na klatce 10., zaznaczamy czerwony napis i w panelu *Właściwości* rozwijamy opcję *FILTRY*. W dolnej części okna klikamy ikonę *Dodaj filtr* i z listy wybieramy *Rozmycie*. Wartość rozmycia należy ustawić indywidualnie w zależności od wybranej czcionki oraz efektu, jaki chcemy uzyskać. W panelu *Właściwości* wybieramy opcję *Efekt koloru*, następnie *Styl* i z listy wybieramy *Alfa* (efekt przezroczystości). Ustawiamy efekt *Alfa* na 0%. Przechodzimy do klatki 20. i efekt *Alfa* ustawiamy na 100%. Teraz biały napis wyłania się z czarnego tła, a czerwony zanika.

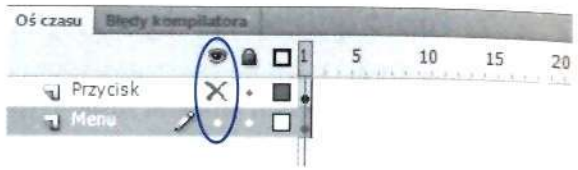
Trzeba jeszcze zdefiniować zatrzymanie animacji. Górną warstwę nazywamy *Akcja*. Wybieramy pierwszą klatkę tej warstwy i otwieramy panel *Operacje* (menu *Okno/Operacje* lub klawisz *F9*). Wpisujemy polecenie `stop();` i przechodzimy do klatki 10., po czym na klawiaturze naciskamy klawisz *F7* (wstawienie pustej klatki) i ponownie w panelu *Operacje* wpisujemy polecenie `stop();`.

Powracamy do okna *Scena 1*. Po uruchomieniu animacji widzimy, że animacja zatrzymała się, ponieważ w pierwszej klatce została zapisana akcja `stop()`. Po przetestowaniu działania przycisku możemy zauważyć, że brakuje typowej ikony rączki, która informuje *na stronie internetowej, że dany element można kliknąć.*

Rączka zostanie utworzona za pomocą niewidzialnego przycisku. W panelu *Biblioteka* w dole okna wybieramy ikonę *Nowy symbol* i w otwartym oknie określamy typ symbolu jako *Przycisk*.

Automatycznie zostanie otwarte okno do edytowania symbolu. W panelu *Oś czasu* zaznaczamy ostatnią klatkę *Hit* i naciskamy klawisz *F6* (klatka kluczowa). Na stole montażowym rysujemy prostokąt (dowolnego koloru, ponieważ i tak nie będzie widoczny) i wracamy do okna *Scena 1*. Zmieniamy nazwę istniejącej warstwy na *Menu* i tworzymy nową warstwę o nazwie *Przycisk*. Na tę warstwę przeciągamy z *Biblioteki*

utworzony przycisk. Po wybraniu narzędzia *Przekształcanie swobodne* można ten przycisk modyfikować, dobierając jego wielkość i położenie do tworzonej animacji. Żeby móc odwoływać się do obiektów w języku ActionScript, należy tym obiektom nadać nazwy. Przechodzimy do panelu *Właściwości* utworzonego przycisku (warstwa *Przycisk*) i w opcji *Nazwa instancji* wpisujemy nazwę *Ukryty*. W panelu *Oś czasu* wyłączamy pokazywanie warstwy *Przycisk* (rysunek 7.30) i dla zaznaczonej animacji w warstwie *Menu* wpisujemy nazwę instancji `Anim`.



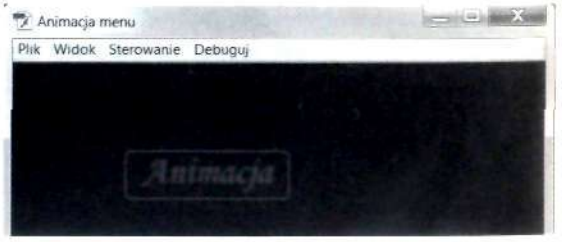
Rysunek 7.30. Ukryta warstwa Przycisk

Odkrywamy warstwę *Przycisk* i powyżej tworzymy nową warstwę. Nazywamy ją *Akcja*. Będzie ona zawierała akcję, która uruchomi naszą animację. Wybieramy pierwszą klatkę tej warstwy i otwieramy panel *Operacje* (*F9*). Wpisujemy polecenia:

```
Ukryty.addEventListener(MouseEvent.MOUSE_OVER, wybierz);
function wybierz (myszka:MouseEvent) : void
  {
   Anim.gotoAndPlay(2);
  }
Ukryty.addEventListener(MouseEvent.MOUSE_OUT, schowaj);
function schowaj (myszka:MouseEvent) : void
  {
   Anim.gotoAndPlay(11);
  }
```

Skrypt w pierwszej linii opisuje zdarzenie, które nastąpi po najechaniu kursorem myszy na niewidzialny przycisk (`MouseEvent.MOUSE_OVER`), i nazwę funkcji, która zostanie wywołana (*wybierz*). Dalej została zdefiniowana funkcja *wybierz*. Odwołuje się ona do klipu *Anim*, w którym należy przejść do klatki 2. (`Anim.gotoAndPlay(2)`) i odtworzyć animację. Druga część skryptu opisuje zdarzenie, które nastąpi po usunięciu kursora z niewidzialnego przycisku.

Aby animacja lepiej się prezentowała, można dodać kolejną warstwę i narysować wokół niej prostokąt z zaokrąglonymi rogami i brakiem wypełnienia (rysunek 7.31).

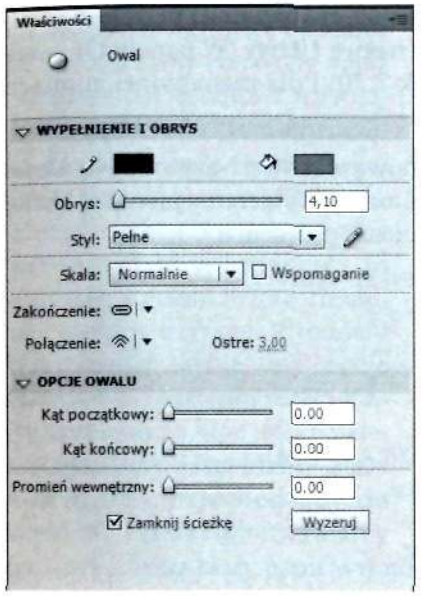


Rysunek 7.31. Animowane menu

7.2.7. Narzędzia do rysowania

W programie Adobe Flash można definiować animacje dla narysowanych przez siebie obiektów. W związku z tym program został wyposażony w narzędzia do rysowania. Są to *Prostokąt* i *Owal* oraz *Linia* i *Ołówek*.

Po wybraniu jednego z tych narzędzi w panelu *Właściwości* można określić parametry rysowanej figury (rysunek 7.32).



Rysunek 7.32. Właściwości rysowanej figury

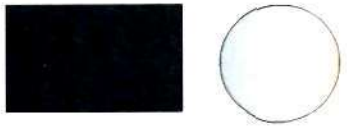
Mamy do dyspozycji trzy tryby rysowania:

- *tryb rysowania ze złączeniem*,
- *tryb rysowania obiektów*,
- *tryb rysowania obiektów prymitywnych*.

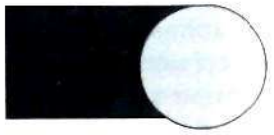
Tryb rysowania ze złączeniem

Tryb rysowania ze złączeniem charakteryzuje się tym, że rysowane figury geometryczne wchodzą ze sobą w interakcje. Polega to na tym, że po narysowaniu dwóch figur, nałożeniu pierwszej na drugą i ponownym ich rozsunięciu z figury drugiej zostanie wycięty wspólny fragment obu figur (rysunki 7.33 – 7.35).

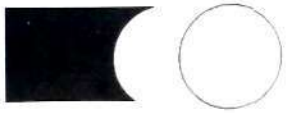
Przystępując do rysowania ze złączeniem, po wybraniu jednego z narzędzi do rysowania figur geometrycznych (*Prostokąt*, *Linia*, *Ołówek*) należy



Rysunek 7.33.
Przed złączeniem

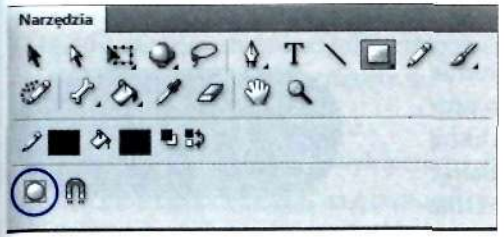


Rysunek 7.34.
Złączenie



Rysunek 7.35.
Rozłączenie

pamiętać o tym, aby w dolnej części panelu *Narzędzia* była wyłączona ikona *Rysowanie obiektów* (rysunek 7.36).



Rysunek 7.36.
Panel Narzędzia.
Wyłączona ikona
Rysowanie obiektów

Przykład 7.6

Otwórz nowy dokument. Wybierz narzędzie rysowania prostokąta. Sprawdź, czy ikona *Rysowanie obiektów* jest wyłączona (jeżeli nie, to wyłącz ją). Narysuj prostokąt i obok niego koło.

Wybierz narzędzie *Zaznaczenie*. Kliknij dwukrotnie koło. Przesuń je tak, aby częściowo zakryło prostokąt. W celu usunięcia zaznaczenia koła kliknij obszar obok figur. Ponownie zaznacz koło, klikając je dwukrotnie. Odsuń koło od prostokąta. Z prostokąta zostanie wycięty owal, który był wspólny dla obu figur po ich nałożeniu na siebie.

Jeżeli tą samą metodą zostaną połączone dwie figury, które nie mają obrysu i są tego samego koloru, to figury te zostaną ze sobą sklejone.

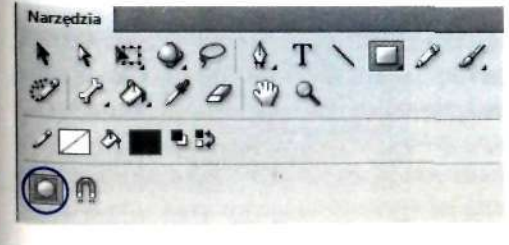
Przykład 7.7

Narysuj prostokąt i obok niego koło w tym samym kolorze. Usuń z figur obrysy (kliknij dwukrotnie każdy obrys i usuń go). Przesuń koło na prostokąt. W celu usunięcia zaznaczenia kliknij pole obok figur. Ponownie zaznacz koło i spróbuj odsunąć je od prostokąta. Powstała jedna figura, której nie można rozdzielić.

Figury, które mają taki sam kolor wypełnienia, będą się ze sobą łączyły, natomiast z figur o różnych kolorach wypełnienia zostaną wycięte wspólne części.

Tryb rysowania obiektów

Przystępując do rysowania obiektów, po wybraniu narzędzia do rysowania figur geometrycznych (*Prostokąt*, *Linia*, *Ołówek*) trzeba pamiętać o tym, aby w dolnej części panelu *Narzędzia* była włączona ikona *Rysowanie obiektów* (rysunek 7.37).



Rysunek 7.37.
Panel Narzędzia.
Włączona ikona
Rysowanie obiektów

Przy rysowaniu obiektów nie zachodzi interakcja między nakładającymi się figurami.

Przykład 7.8
Otwórz nowy dokument. Wybierz narzędzie do rysowania prostokąta. Sprawdź, czy ikona *Rysowanie obiektów* jest włączona (jeżeli nie, to włącz ją). Narysuj prostokąt i obok niego koło.

Wybierz narzędzie *Zaznaczenie*. Kliknij dwukrotnie koło. Przesuń koło tak, aby częściowo zakryło prostokąt. W celu usunięcia zaznaczenia koła kliknij obszar obok figur. Ponownie zaznacz koło, klikając je dwukrotnie. Odsuń koło od prostokąta. Obie figury zachowały swój pierwotny kształt.



Rysunek 7.38.
Prymitywny owal

Tryb rysowania obiektów prymitywnych
Aby narysować obiekty prymitywne, należy w panelu *Narzędzia* rozwinąć listę obiektów znajdujących się przy narzędziu *Prostokąt* i wybrać *Narzędzie Prostokąt* (trzecia pozycja) lub *Narzędzie Owal* (czwarta pozycja).



Rysunek 7.39.
Prymitywny owal
po zmianie właściwości

Prymitywny owal ma zaokrąglone narożniki (rysunek 7.38). W panelu *Właściwości* można zmieniać stopień ich zaokrąglenia.

Zmieniając właściwości prymitywnego owalu, można też uzyskać np. niepełny owal (rysunek 7.39).

Do określenia wyglądu prymitywnego owalu w panelu *Właściwości* służą parametry w sekcji *Opcje owalu* (*Kąt początkowy*, *Kąt końcowy*, *Promień wewnętrzny* — rysunek 7.40). Manipulując tymi parametrami, można uzyskać oryginalne figury.



Rysunek 7.40.
Właściwości prymitywnego owalu

Znając narzędzia do rysowania oraz ich właściwości, można tworzyć dowolne elementy i animować je w dalszych etapach pracy.

Zadanie 7.3
Narysuj prymitywny prostokąt i prymitywny owal — przyciski, które wykorzystasz na stronie internetowej. Manipulując parametrami tych obiektów dostępnymi w panelu *Właściwości*, utwórz atrakcyjne wizualnie elementy. Korzystając z technologii Flash i języka Action Script, dodaj efekty animowania przycisków.

7.2.8. Rysowanie

Przy użyciu dostępnych w programie Adobe Flash narzędzi do rysowania można tworzyć dowolne obiekty. W kolejnych etapach pracy mogą one być animowane. Rysowane obiekty można edytować lub przekształcać. Można z nich tworzyć banery reklamowe lub elementy strony internetowej.

Przekształcanie swobodne
Gdy w panelu *Narzędzia* wybierzemy ikonę *Przekształcanie swobodne*, zaznaczony obiekt zostanie otoczony obwiednią z punktami, za pomocą których można ten obiekt obracać lub dowolnie zmieniać jego kształt (rysunek 7.41).



Rysunek 7.41.
Przekształcanie swobodne

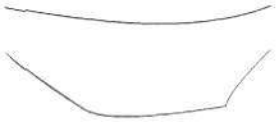
Zaznaczenie
Gdy wybierzemy ikonę *Zaznaczenie* i ustawimy kursor myszy w pobliżu krawędzi narysowanego obiektu, kursor zmieni kształt na . Jeżeli kursor ma obok strzałki kształt wygiętego łuku, to można chwycić krawędź narysowanego obiektu i przesuwać ją oraz dowolnie wyginać. Wykonując tę samą czynność z wciśniętym klawiszem *Alt*, tworzymy punkty złamania krawędzi.

Zaznaczenie obiektu przez dwukrotne kliknięcie i przesunięcie go z wciśniętym klawiszem *Alt* spowoduje skopiowanie obiektu.

Przykład 7.9
Zaprojektujemy rysunek, który w kolejnym etapie pracy zostanie wykorzystany do animacji. Obiektem animowanym będzie łódka płynąca po wodzie.

Zaczniemy od narysowania łódki.

Za pomocą narzędzia *Linia* rysujemy dwie równoległe linie. Jedna będzie dnem łódki, druga pokładem. Wybieramy narzędzie *Zaznaczenie* i przy wykorzystaniu wyżej omówionych metod (kursor myszy, klawisz *Alt*) wyginamy dolną linię, aby przyjęła kształt dna łódki, i górną, aby uzyskała kształt pokładu (rysunek 7.42).



Rysunek 7.42.
Rysowanie kształtu łódki

Po dwukrotnym kliknięciu linii pokładu przesuwamy ją tak, aby została połączona z linią dna łódki. Aby sprawdzić połączenie linii, powiększamy widok stołu montażowego do 400% (prawy górny róg). Rysujemy pionową linię, która będzie masztem łódki. W panelu *Właściwości* dobieramy jej grubość i kolor (rysunek 7.43).



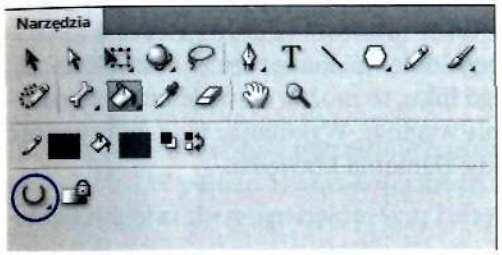
Rysunek 7.43.
Łódka z masztem

Do pokolorowania łódki użyjemy narzędzia *Wiadro z farbą* z panelu *Narzędzia*. Zaznaczamy ikonę i w panelu *Właściwości* określamy kolor łódki, a następnie klikamy w obszarze jej kształtu umieszczonego na stole montażowym (rysunek 7.44).



Rysunek 7.44.
Pokolorowana łódka

Jeżeli kolor się nie pojawi, to znaczy, że linie dna łódki i pokładu nie zostały dokładnie połączone. Możemy spróbować naprawić tę niedoskonałość, klikając w panelu *Narzędzia* ikonę *Rozmiar odstępu* (przy wybranym narzędziu *Wiadro z farbą*), a następnie wybierając z listy opcję *Zamknij duże odstępy* (rysunek 7.45).



Rysunek 7.45.
Panel Narzędzia, ikona Rozmiar odstępu

Żagiel narysujemy, korzystając z narzędzia *Gwiazda–Wielokąt*. Wybieramy to narzędzie i w panelu *Właściwości* klikamy przycisk *Opcje*, aby określić liczbę boków wielokąta. W polu *Liczba boków* wpisujemy 3. Określamy czarną cienką krawędź rysowania i białe wypełnienie. Rysujemy trójkąt i przy użyciu narzędzi *Zaznaczenie* oraz *Przekształcanie swobodne* modelujemy wygląd żagla.



Rysunek 7.46.
Łódka z żaglami

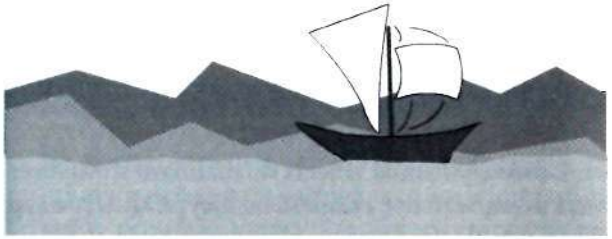
Stosując tę metodę, można dodać drugi żagiel (rysunek 7.46).

Pozostało zdefiniowanie animacji. Aby łódkę można było animować, należy przekształcić ją na symbol. Wybieramy narzędzie *Zaznaczenie* i rysując wokół łódki prostokąt, zaznaczamy wszystkie jej elementy. Następnie przeciągamy łódkę do panelu *Biblioteka*. W otwartym oknie podajemy nazwę symbolu i wybieramy typ *Klip filmowy*. W panelu *Oś czasu* dla warstwy *Łódka* zaznaczamy klatkę 90. (czas trwania animacji) i wstawiamy klatkę końcową, naciskając klawisz *F5*. Następnie klikamy zaznaczony przedział czasu prawym przyciskiem myszy i wybieramy opcję *Utwórz animację ruchu*. Ustawiamy głowicę na osi czasu na klatce pierwszej, a na stole montażowym — łódkę w miejscu początkowym ruchu. Na osi czasu przesuwamy głowicę na ostatnią klatkę

czasu trwania animacji (klatka *90.*) i na stole montażowym przesuwamy łódkę do miejsca zakończenia jej ruchu.

Sprawdzamy działanie animacji, stosując kombinację klawiszy *Ctrl+Enter*.

Zadanie 7.4
Do utworzonej w przykładzie 7.9 animacji dodaj warstwę tła i utwórz, korzystając z narzędzi programu Adobe Flash, morskie fale, po których popłynie łódka (np. jak na rysunku 7.47). Zdefiniuj odpowiedni czas wyświetlania tła animacji.

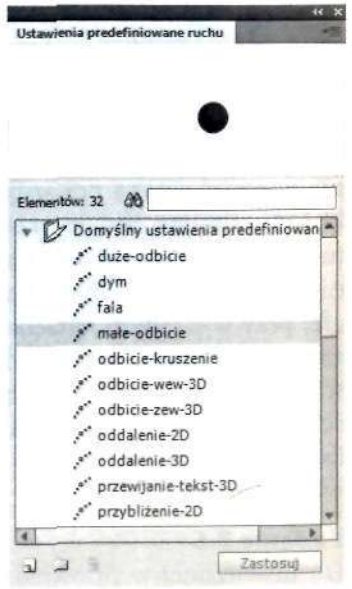


Rysunek 7.47. Przykładowe tło dla łódki

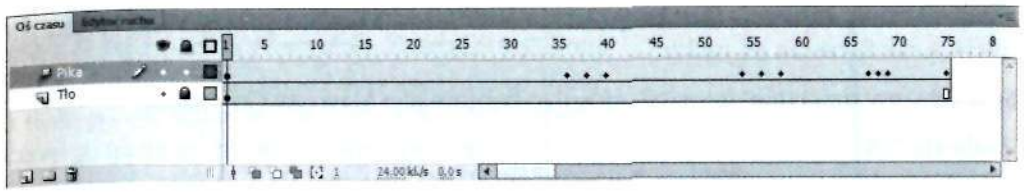
Uzupełnij animację atrakcyjnym tekstem animowanym reklamującym podróże żaglówką po morzach i oceanach.

7.2.9. Ustawienia predefiniowane ruchu

Ustawienia predefiniowane ruchu to ustawienia ruchu wstępnie zdefiniowane w programie Adobe Flash. Można je stosować dla obiektów umieszczonych na stole montażowym. Należy zaznaczyć obiekt i w panelu *Ustawienia predefiniowane ruchu* wybrać odpowiedni ruch (rysunek 7.48). Większość dostępnych ruchów to ustawienia spotykane na stronach internetowych, w reklamach lub banerach informacyjnych. Po kliknięciu przycisku *Zastosuj* efekt zostanie przypisany do obiektu. W panelu *Oś czasu* pojawi się zdefiniowana animacja ruchu (rysunek 7.49). Wszystkie właściwości ruchu zostaną ustawione przez program.



Rysunek 7.48.
Panel Ustawienia predefiniowane ruchu



Rysunek 7.49. Panel Oś czasu z ustawioną automatycznie animacją

Przykład 7.10

Wykorzystując panel *Ustawienia predefiniowane ruchu*, przygotujemy animację zawierającą ruch piłki odbijającej się na boisku (trawie).

Przygotuj w programie graficznym obraz piłki. Zapisz go w formacie GIF. Przygotuj zdjęcie boiska i zapisz je w formacie JPG.

Zaimportuj przygotowane obrazy do *Biblioteki* nowego dokumentu programu Adobe Flash.

Na warstwie tła umieść zdjęcie boiska. Dostosuj rozmiar zdjęcia do rozmiaru stołu montażowego i zablokuj tę warstwę. Utwórz nową warstwę i umieść na niej piłkę. Używając narzędzia *Przekształcanie swobodne* i klawisza *Shift*, dostosuj rozmiar piłki do rozmiaru tła. Aby animować ruch piłki na boisku, przekształć obiekt na symbol. Umieść piłkę w miejscu początkowym jej ruchu. W celu wyświetlenia panelu *Ustawienia predefiniowane ruchu* w menu wybierz *Okno/Ustawienia predefiniowane ruchu*. W otwartym oknie wybierz ruch, który najbardziej przypomina lot piłki (np. *małe-odbicie*). Zastosuj wybrany ruch. Na stole montażowym pojawiła się ścieżka ruchu piłki. Linia ciągła określa kierunek, w którym piłka będzie się poruszała, natomiast linia wykropkowana pokazuje klatki i określa, jak będzie przebiegał ruch piłki (rysunek 7.50). Ustawienie kursora myszy obok linii ciągłej przy wybranym narzędziu *Zaznaczenie* pozwoli zmienić kształt ścieżki ruchu. Aby przesunąć całą ścieżkę ruchu, wystarczy kliknąć linię ciągłą i przesunąć ją w odpowiednie miejsce.



Rysunek 7.50.
Ruch predefiniowany dla piłki

Zadanie 7.5

Do utworzonej w przykładzie 7.10 animacji dodaj animowany napis, np. „Zbuduj własny stadion".

PYTANIA I POLECENIA KONTROLNE

1. Wymień metody stosowane do tworzenia animacji komputerowych.
2. Jakiego rodzaju animacje mogą się znaleźć na stronie internetowej?
3. Jakie zastosowanie ma kanał alfa w grafice komputerowej i animacji?
4. W jakich sytuacjach podczas tworzenia animacji komputerowej wykorzystywany jest język ActionScript?
5. W jakim celu w programie Adobe Flash należy przekształcić obiekt na symbol?
6. Jakie typy symboli są dostępne w programie Adobe Flash?
7. Wymień sposoby umieszczania plików Flash (*.swf*) na stronie internetowej.

ZADANIE

Na potrzeby tworzonej witryny internetowej zawierającej informacje na temat szkoły zostaną przygotowane animacje typu: baner reklamowy szkoły, animowane przyciski menu. Z animacji opracowanych przez uczestników projektu należy wybrać te, które powinny się znaleźć na stronach witryny internetowej.

8

Dźwięk i wideo

8.1. Dźwięk

8.1.1. Formaty plików dźwiękowych

Aby można było odtworzyć dźwięk, musi on być zapisany w pliku we właściwym formacie.

Do podstawowych formatów audio należą:

- plik audio **AIFF** (rozszerzenie *.aiff*),
- plik audio **AU** (rozszerzenie *.au*),
- plik audio **MP3** (rozszerzenie *.mp3*),
- plik **Wave Form Audio Format** (rozszerzenie *.wav*),
- plik **Windows Media Audio** (rozszerzenie *.wma*)
- plik w formacie **Vorbis** (rozszerzenie *.ogg*).

Format WAVE

WAVE jest najprostszym formatem plików dźwiękowych. W formacie tym dźwięk nie jest poddawany żadnej kompresji. Jego wadą jest duży rozmiar plików, a zaletą — brak utraty jakości dźwięku. Dodatkowym walorem jest łatwość edytowania plików tego typu. Jest to format obsługiwany przez większość programów do edytowania dźwięków, w związku z czym często bywa wykorzystywany jako standard przenoszenia danych między aplikacjami.

Format MP3

Format **MP3** (przez fachowców określany jako MPEG-1 Audio Layer 3) jest standardem kodowania audio MPEG. Jest to format kompresji stratnej wykorzystujący niedoskonałości ludzkiego ucha. Z pliku dźwiękowego zostają usunięte częstotliwości, których człowiek nie jest w stanie usłyszeć. Kompresja pozwala na znaczne zmniejszenie rozmiaru pliku w zależności od tego, jaki rodzaj dźwięku jest kompresowany. Na przykład kompresja tekstu mówionego zapisanego na potrzeby internetu może być większa niż kompresja muzyki, przy której standardów jakości nie powinniśmy obniżać. Główną

zaletą formatu MP3 jest niewielki rozmiar pliku przy dobrej jakości dźwięku. Jego wada to utrata pewnych częstotliwości z oryginalnego materiału, a co za tym idzie — uniemożliwienie odbiorcy usłyszenia niektórych dźwięków.

Format WMA
Jest to format firmy Microsoft wprowadzony na rynek jako konkurencja dla formatu MP3. Zasada kompresji jest podobna w obu formatach. Również jakość dźwięku jest porównywalna.

Format MIDI
W formacie tym nie jest zapisywany sam dźwięk, ale informacje o tym, jaki dźwięk powinien zostać odtworzony, jaka ma być jego częstotliwość i jaki ma być instrument odtwarzania.

Format AIFF
Jest to format plików dla komputerów Macintosh, ale może być również wykorzystywany na innych komputerach. Pliki nie są kompresowane. Zaletą tego formatu jest dobra jakość dźwięku oraz to, że jest obsługiwany przez wiele przeglądarek bez konieczności instalowania wtyczek. Wadą są duże rozmiary plików, co ogranicza możliwość ich wykorzystania.

Format RealMedia
Format ten został opracowany z myślą o zastosowaniach w internecie. Jest wykorzystywany do strumieniowego przesyłania danych takich jak muzyka czy filmy wideo. Zaletą tego formatu jest duży współczynnik kompresji przy dobrej jakości. Do odtwarzania plików zapisanych w tym formacie służy odtwarzacz Real Player.

Format ogg
Ogg Vorbis to format kompresji plików audio. Umożliwia on zachowanie wysokiej jakości dźwięku przy niewielkiej objętości. Jest alternatywą dla popularnego formatu MP3. Skompresowane pliki mają mniejszy rozmiar niż pliki MP3 przy podobnej jakości ich odtwarzania. Jest bezpłatny, otwarty (kod udostępniany na zasadzie licencji GNU) i nieopatentowany.

8.1.2. Kompresja dźwięku

Pliki zawierające muzykę mają duże rozmiary. Do zapisania jednej minuty muzyki potrzeba około 10 MB, a utwory muzyczne trwają zwykle dłużej. Przesyłanie tak obszernych plików przez internet jest problemem. Kompresja, czyli pakowanie danych dźwiękowych, pozwala na zmniejszenie ich rozmiaru. Kompresję dźwięku dzielimy na stratną i bezstratną.

Kompresja stratna
Kompresja stratna to usunięcie z zapisu dźwięków niesłyszalnych lub słabo słyszalnych w celu zmniejszenia rozmiaru pliku. Od stopnia kompresji zależy nie tylko rozmiar pliku, ale również jakość dźwięku. Im większy stopień kompresji, tym gorsza jakość. Kompresja stratna jest procesem nieodwracalnym.

Kompresja bezstratna
Kompresja bezstratna to metoda zmniejszenia objętości pliku z możliwością odtworzenia dźwięku w wersji identycznej z pierwotną. Najczęściej polega na zmianie sposobu zapisu danych, dzięki czemu plik zajmuje mniej miejsca.

W standardowych formatach zapisywania dźwięku (np. w plikach o rozszerzeniu *.wav*) stosuje się ten sam sposób zapisu niezależnie od tego, czy dźwięk ma skomplikowany przebieg, czy panuje cisza. Zmieniając sposób zapisu dźwięku (np. przez specjalne oznaczenie ciszy), można zaoszczędzić sporo miejsca.

8.1.3. Odtwarzanie audio

Każdy system operacyjny jest wyposażony w odtwarzacz plików audio, który obsługuje najpopularniejsze formaty plików. W systemie Windows jest to Windows Media Player. W internecie można znaleźć wiele odtwarzaczy komercyjnych i bezpłatnych, np.: Winamp, QuickTime, iTunes, Real Player. Również programy do nagrywania zawierają odtwarzacze multimedialne.

Windows Media Player
Windows Media Player to program, który oprócz odtwarzania ułatwia wyszukiwanie plików multimedialnych na komputerze i w sieci. Umożliwia on odtwarzanie strumieni multimediów z internetu, zgrywanie muzyki z płyt CD-Audio oraz jej nagrywanie na płytach CD.

QuickTime
QuickTime jest popularnym odtwarzaczem multimedialnym opracowanym przez firmę Apple dla systemów Windows i Mac OS. Pozwala na odtwarzanie plików audio i wideo udostępnianych strumieniowo za pośrednictwem internetu. Obsługuje format H.264 i różne standardy dźwięku przestrzennego, umożliwiając odtwarzanie multimediów w bardzo dobrej jakości. QuickTime automatycznie analizuje przepustowość łącza, co pozwala na oglądanie materiałów w najlepszej jakości.

iTunes
iTunes to aplikacja opracowana przez firmę Apple jako odtwarzacz multimedialny. Umożliwia odtwarzanie muzyki, a także słuchanie radia internetowego, nagrywanie płyt CD, katalogowanie nagrań i konwertowanie plików audio.

Real Player
Real Player firmy Real Networks to aplikacja do odbioru transmisji audio i wideo przez internet w czasie rzeczywistym. Stosowane przez program formaty RealAudio i RealVideo są bardzo popularnymi standardami transmitowania audycji radiowych i telewizyjnych w internecie. Program umożliwia konwersję plików audio i wideo na wiele formatów, odtwarzanie płyt DVD oraz nagrywanie płyt CD-Audio. Ma wbudowaną przeglądarkę stron WWW.

8.2. Edytowanie i obróbka dźwięku

Do prostego edytowania dźwięku może zostać wykorzystany rejestrator dźwięku obecny w systemie operacyjnym. Przy jego użyciu użytkownik może nagrać np. swój głos. W internecie dostępne są programy służące do nagrywania, edytowania, miksowania i zapisywania plików dźwiękowych w wielu formatach.

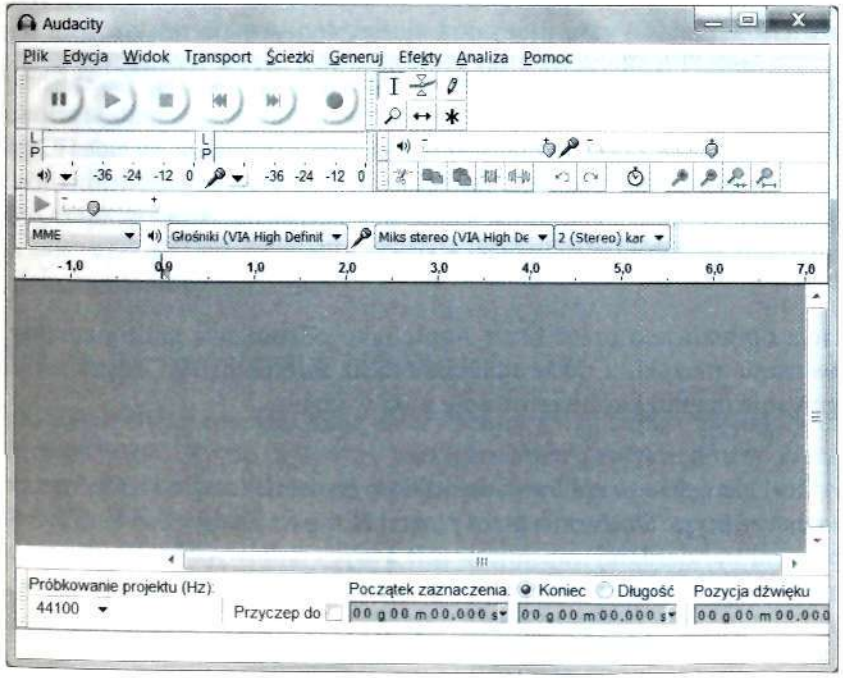
8.2.1. Przetwarzanie dźwięku — program Audacity

Prostym edytorem plików dźwiękowych jest program Audacity, który umożliwia nagrywanie i edytowanie dźwięku w formatach WAV, AIFF, MP3 oraz AU, a także ma wbudowane narzędzia do konwersji formatów, edytor obwiedni i narzędzia do nakładania dodatkowych efektów, np. dynamiki, korekcji, modulacji. Może służyć do usuwania usterek w gotowych plikach muzycznych. Współpracuje z kodekiem LAME, który umożliwia tworzenie plików MP3.

Instalowanie Audacity

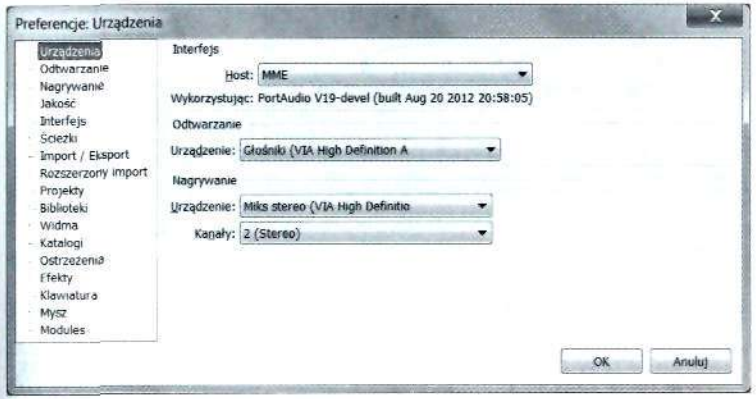
Program Audacity jest bezpłatny i można go pobrać ze strony *http://audacity.sourceforge.net/*. Znajduje się na niej najnowsza wersja Audacity oraz współpracujący z programem kodek plików MP3 LAME. Należy pobrać obydwa pliki i zainstalować programy.

Po zainstalowaniu program Audacity zostanie uruchomiony automatycznie (rysunek 8.1).



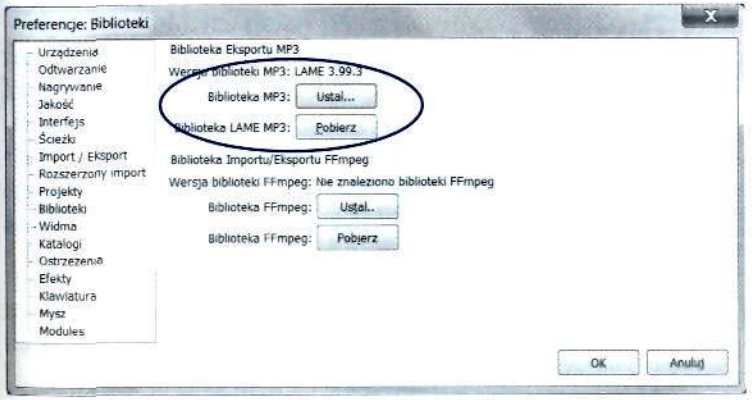
Rysunek 8.1. Okno programu Audacity

Pierwszym krokiem powinno być sprawdzenie i w razie potrzeby zmodyfikowanie ustawień dotyczących urządzeń do odtwarzania i nagrywania dźwięku. Wybieramy w menu *Edycja/Preferencje...*, w otwartym oknie zaznaczamy opcję *Urządzenia* (rysunek 8.2) i wskazujemy urządzenie do odtwarzania dźwięku (pole *Odtwarzanie*) oraz urządzenie do nagrywania (pole *Nagrywanie*). W opcji *Kanały* ustawiamy dwa kanały stereo.



Rysunek 8.2. Ustawienia urządzeń odtwarzania i nagrywania

Aby dodać kodek LAME, w tym samym oknie *Preferencje* wybieramy opcję *Biblioteki* (rysunek 8.3).

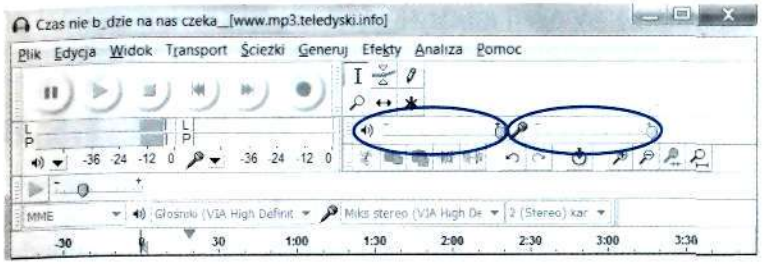


Rysunek 8.3. Ustawienia panelu Biblioteki

Biblioteki LAME zostaną dołączone automatycznie. W otwartym oknie pojawi się opis dołączonej biblioteki lub komunikat, że biblioteka eksportu MP3 nie została znaleziona. W takim przypadku należy kliknąć przycisk *Ustal...* i znaleźć kodek LAME zainstalowany na komputerze. Jeżeli kodek nie jest zainstalowany, trzeba kliknąć przycisk *Pobierz* i zainstalować wymagane oprogramowanie. Zatwierdzamy wprowadzone ustawienia, klikając przycisk *OK*.

Jeżeli ustawienia dla bibliotek zostały poprawnie zdefiniowane, program Audacity może odczytywać i zapisywać pliki MP3.

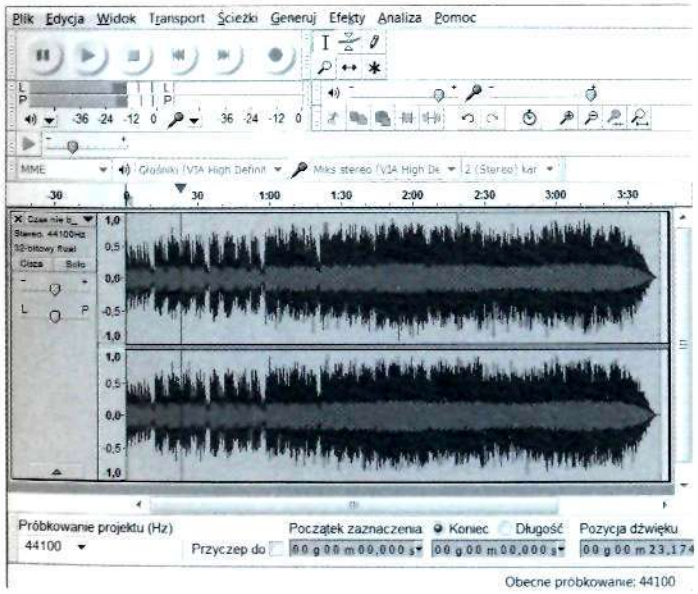
W oknie programu można zmienić ustawienia systemowe wzmocnienia głośników lub słuchawek (*Głośność wyjściowa*) oraz ustawienia wzmocnienia mikrofonu (*Głośność wejściowa*), przesuwając odpowiednio suwaki tych opcji (rysunek 8.4).



Rysunek 8.4. Ustawienie wzmocnienia głośników (słuchawek) i mikrofonu

8.2.2. Modyfikowanie dźwięku

Modyfikowanie plików zapisanych w formacie MP3 jest jedną z najczęściej wykonywanych czynności związanych z obróbką dźwięku. Zawartość pliku wczytanego przy użyciu opcji menu *Plik/Otwórz* zostanie wyświetlona w programie w postaci wykresu spektrograficznego (rysunek 8.5).

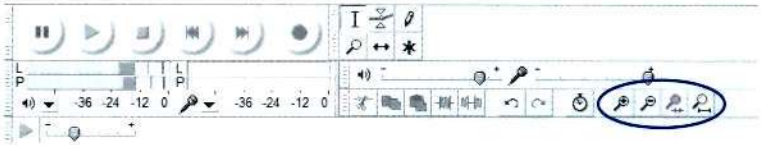


Rysunek 8.5. Plik MP3 odtwarzany w programie Audacity

Ze względu na to, że utwór został nagrany w formacie stereo, wyświetlane są dwie ścieżki reprezentujące lewy i prawy kanał. Jeśli klikniemy w którymś miejscu w obrębie wykresu dźwięku, zostanie tam wstawiona pionowa linia. Od tego miejsca po kliknięciu przycisku *Odtwórz* rozpocznie się odtwarzanie muzyki.

Lupa

Zawartość pliku dźwiękowego jest wyświetlana w programie w postaci wykresu. W celu łatwiejszego operowania na wykresie można użyć narzędzia *Lupa*. Pasek edytowania zawiera zestaw lup, które pozwalają manipulować widocznym w oknie fragmentem utworu. Lupa *Dopasuj projekt* pozwoli na dostosowanie wykresu utworu do wielkości okna. *Dopasuj zaznaczenie* dopasuje zaznaczony fragment do wielkości okna (rysunek 8.6).



Rysunek 8.6. Lupy na pasku edytowania

Pracę z dźwiękiem zawsze wykonujemy na zaznaczonym fragmencie. Zaznaczenie nastąpi po kliknięciu w dowolnym miejscu utworu i przeciągnięciu myszą po wykresie w prawo lub w lewo.

Wyciszenie

Jedną z metod modyfikowania utworów muzycznych jest wyciszenie początkowego lub końcowego fragmentu utworu.

Przykład 8.1

Wczytujemy plik muzyczny przeznaczony do modyfikacji. Chcemy wyciszyć początkowy fragment utworu. Odsłuchujemy go i określamy, jaki fragment zostanie wyciszony. Klikamy na wykresie w miejscu końca wyciszenia i przeciągamy myszą do samego początku utworu. Wybieramy ikonę lupy ze strzałkami i powiększamy zaznaczony fragment. Odtwarzamy go. Przed wykonaniem jakiejkolwiek operacji powinniśmy go odsłuchać, aby upewnić się, że zaznaczenie zostało wykonane prawidłowo.

Aby wyciszyć zaznaczony fragment, wybieramy ikonę *Cisza* na pasku *Edycja* (rysunek 8.7).



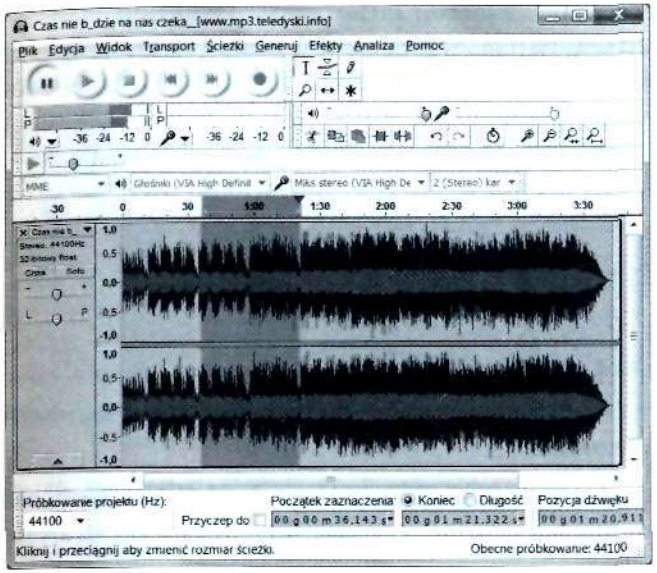
Rysunek 8.7. Ikona wyciszenia

Wycinanie fragmentu

Często wycinamy część utworu w celu dalszego wykorzystania tylko tego fragmentu.

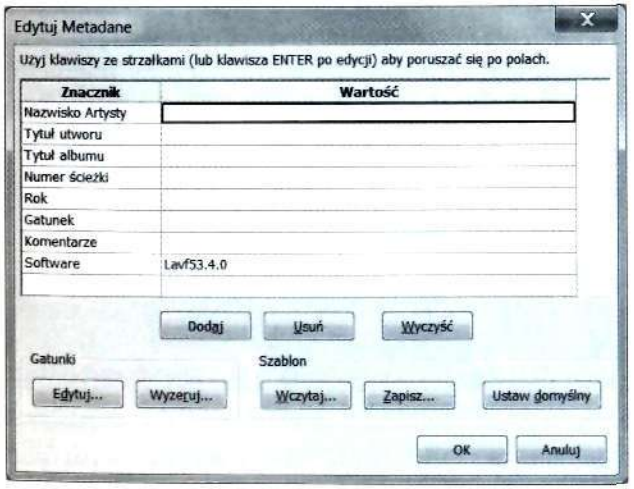
Przykład 8.2

Otwieramy plik dźwiękowy i uruchamiamy go. Odsłuchując utwór, na skali czasu (nad wykresem) ustawiamy czas odpowiadający fragmentowi wycinanemu z całości. Następnie klikamy w obszarze wykresu, aby ustawić znacznik początku fragmentu wycięcia, i przeciągając myszą, zaznaczamy koniec fragmentu wycięcia (rysunek 8.8).



Rysunek 8.8. Wybrany fragment zaznaczony na wykresie

Zapisujemy zaznaczoną część utworu i poddajemy dalszej obróbce tylko ten fragment. Wybieramy *Plik/Eksportuj zaznaczenie* i w otwartym oknie wpisujemy nazwę pliku oraz wskazujemy format MP3. Pojawi się okno do wprowadzenia dodatkowych informacji o zapisywanym fragmencie (rysunek 8.9). Możemy wypełnić je dowolną treścią.



Rysunek 8.9. Dodatkowe informacje o utworze

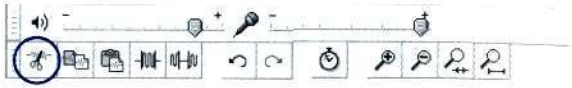
Zapisany fragment ponownie otwieramy. Odsłuchujemy początkową i końcową część. Jeżeli utwór kończy się zbyt gwałtownie, można skorygować jego fragment końcowy. Zaznaczamy ostatnie 3 sekundy utworu (należy próbować z różnymi wartościami, aż uzyskamy oczekiwany efekt), wybieramy z menu opcję *Efekty* i z listy efekt *Wyciszanie*.

Przy odtwarzaniu końcowego fragmentu głośność będzie stopniowo malała. Zapisujemy utwór, wybierając z menu *Plik/Eksportuj...*.

Inaczej postępujemy, jeśli zaznaczony fragment utworu chcemy po prostu usunąć.

Przykład 8.3

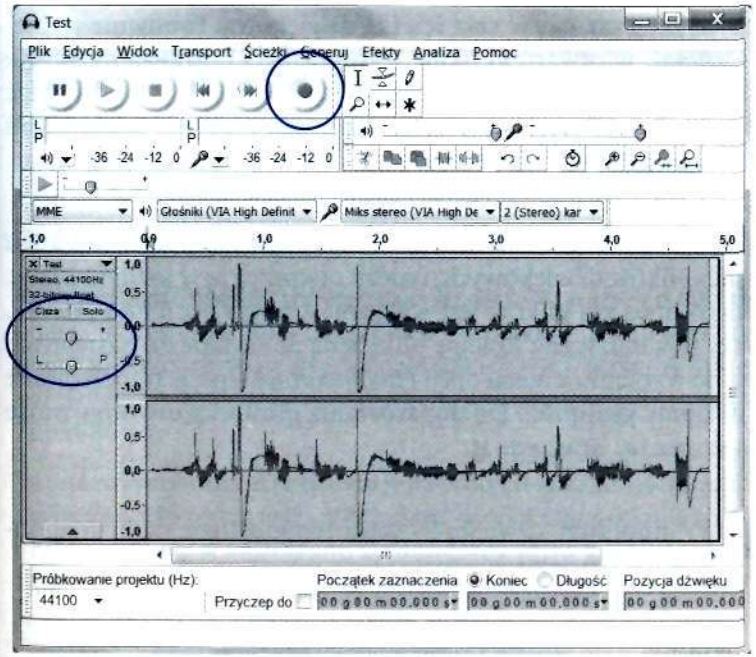
Planujemy usunąć z wybranego utworu 30-sekundowy fragment początkowy. Otwieramy plik muzyczny, zaznaczamy fragment przeznaczony do wycięcia i klikamy na pasku *Edycja* ikonę *Wytnij* (rysunek 8.10). Zaznaczony fragment zostanie usunięty. Trzeba jeszcze odsłuchać utwór i — jeżeli to konieczne — skorygować wycięcie, a następnie zapisać plik.



Rysunek 8.10. Ikona Wytnij

Nagrywanie dźwięku

Korzystając z programu Audacity, użytkownik może nagrać np. swój głos i odpowiednio go zmodyfikować. Do tego celu służy różowy przycisk `Nagraj` na pasku *Transport* (rysunek 8.11).



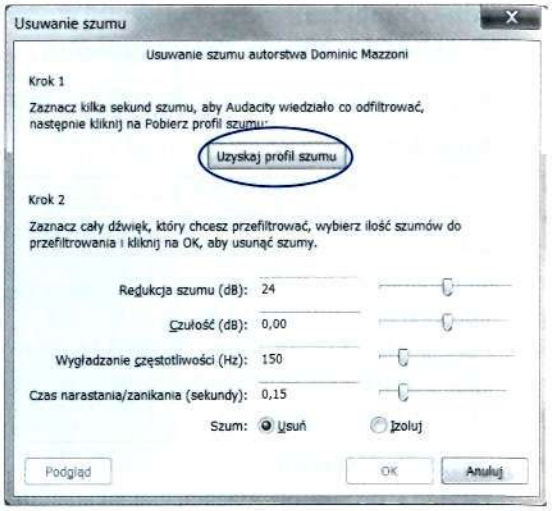
Rysunek 8.11. Narzędzia nagrywania głosu

Nagrany głos można podnieść lub obniżyć za pomocą suwaków z lewej strony wykresu. Można też zapisać go do pliku MP3.

Usuwanie szumów

Przykład 8.4

Otwieramy plik dźwiękowy zawierający szumy, które chcemy usunąć. Zaznaczamy fragment ścieżki dźwiękowej, najlepiej żeby był to kilkusekundowy fragment, na którym zarejestrowana jest wyłącznie cisza (czyli sam szum mikrofonu), następnie z menu wybieramy *Efekty* i *Usuwanie szumu*. W otwartym oknie klikamy przycisk *Uzyskaj profil szumu* (rysunek 8.12) i zatwierdzamy operację przyciskiem *OK*. Uzyskana przez program Audacity próbka szumu zostanie wykorzystana przy jego usuwaniu.



Rysunek 8.12. Definiowanie profilu szumu

Naciskamy klawisze *Ctrl+A*, żeby zaznaczyć całą ścieżkę dźwiękową. Ponownie wybieramy *Efekty* i *Usuwanie szumu*, w otwartym oknie (rysunek 8.12) ustawiamy suwak *Redukcja szumu (dB)* na *0*. Przed zapisaniem klikamy przycisk *Podgląd*, aby odsłuchać ścieżkę po zastosowaniu ustawień. W razie potrzeby możemy manipulować pozostałymi suwakami aż do uzyskania pożądanych rezultatów.

Łączenie plików muzycznych

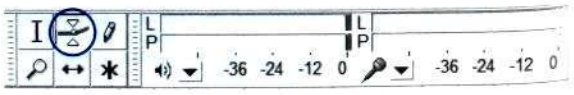
Domyślnie kolejne pliki audio są otwierane w nowych oknach programu Audacity. Jeżeli zamierzamy połączyć kilka plików dźwiękowych, należy otworzyć je w jednym oknie. Najprostszą metodą takiego otwarcia plików jest wybranie z menu opcji *Plik/Otwórz*. Zostanie otwarty pierwszy plik muzyczny. Kolejne pliki będą dodawane do następnych ścieżek tego samego okna po wybraniu z menu opcji *Plik/Importuj/Dźwięk*. Importowane pliki mogą mieć różne poziomy głośności. Do dopasowania głośności utworów może zostać wykorzystane `Narzędzie obwiedni`.

Obwiednia

Obwiednia to zapis cyfrowy amplitudy dźwięku — zapis ten możliwie najpełniej odzwierciedla *oryginalne brzmienie dźwięku*.

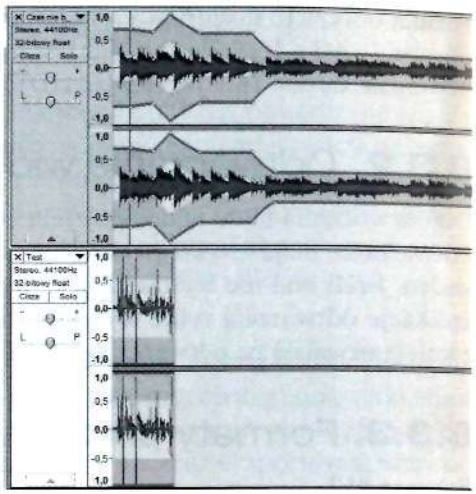
Przykład 8.5

Importujemy do programu Audacity dwa pliki dźwiękowe. Jeden zawiera muzykę, drugi — nagrany wcześniej głos. Jeżeli muzyka jest za głośna w stosunku do nagranego głosu, to w celu jej przyciszenia można wykorzystać `Narzędzie obwiedni`. Jego ikona znajduje się na pasku *Narzędzia* (rysunek 8.13).



Rysunek 8.13. Ikona narzędzia obwiedni

Po wybraniu tej ikony ustawiamy kursor myszy nad wykresem ścieżki z muzyką, chwytamy górną krawędź obwiedni wykresu i przeciągamy myszą w dół. Powoduje to ściszenie utworu. Klikając w kolejnym miejscu obwiedni, tworzymy na niej następny punkt i podnosimy myszą obwiednię do góry, powodując zgłośnienie muzyki. W ten sposób można tworzyć kolejne punkty na obwiedni i odpowiednio modelować głośność utworu (rysunek 8.14).



Rysunek 8.14. Definiowanie obwiedni

Zadanie 8.1

Przygotuj trzy pliki dźwiękowe. Umieść je na osobnych ścieżkach. Stosując poznane metody pracy z dźwiękiem, zmodyfikuj kolejne ścieżki (przytnij je oraz wycisz) i utwórz miks muzyczny.

8.3. Pliki wideo

Pliki wideo służą do zapisywania filmów. Do pliku wideo mogą być dołączone napisy — np. informacje o filmie lub jego autorze — komentarze lub paski informacyjne.

8.3.1. Standardy

Rozdzielczość obrazu mówi o jego jakości. Im wyższa rozdzielczość, tym wyraźniejszy obraz. W systemie **PAL** (ang. *Phase Alternating Line*), wykorzystywanym w Europie, obraz jest wyświetlany w rozdzielczości 704×576 px. W systemie **NTSC** (ang. *National Television System Committee*), stosowanym w USA, rozdzielczość obrazu wynosi 720×480 px.

W wykorzystywanym w telewizji wysokiej rozdzielczości standardzie **HDTV** (ang. *High Definition TV*) obraz jest wyświetlany w większej rozdzielczości niż w standardach **PAL** i **NTSC**. Obecnie stosowane są standardy **HDTV 720p** (rozdzielczość obrazu wynosi 1280×720 px) oraz **1080i/1080p** (rozdzielczość obrazu wynosi 1920×1080 px). Zaletami standardu **HDTV** są lepsza jakość obrazu na dużych ekranach i pełne wykorzystanie możliwości ekranów panoramicznych.

Liczba klatek na sekundę określa liczbę klatek filmu wyświetlanych w czasie jednej sekundy. Przyjmuje się, że aby obraz zmieniał się płynnie, ten parametr musi wynosić 24 – 30 klatek na sekundę. W systemie PAL wynosi on 25 klatek na sekundę, w systemie NTSC — 30 klatek na sekundę.

Format obrazu to stosunek szerokości obrazu do jego wysokości. Stosuje się dwa formaty obrazu — 4:3 i 16:9. Format **4:3** jest używany w telewizji w systemach PAL i NTSC, natomiast format **16:9** w telewizji HDTV (wysokiej rozdzielczości) i telewizji cyfrowej.

8.3.2. Odtwarzanie wideo

Do odtworzenia filmu wideo potrzebne jest odpowiednie oprogramowanie. Dostępne odtwarzacze mają zwykle pakiet kodeków do odtwarzania różnych formatów plików wideo. Jeżeli one nie wystarczają, można doinstalować dodatkowe kodeki. Niektóre aplikacje odtwarzają tylko wybrany format plików (Real Player, QuickTime), inne z kolei pozwalają na odtwarzanie wielu typowych formatów (Windows Media Player).

8.3.3. Formaty plików wideo

Format AVI
Format *AVI* został opracowany przez firmę Microsoft. Jest stosowany do zapisu dźwięku i sekwencji wideo. Dane można zapisywać w postaci skompresowanej lub nieskompresowanej. Aby można było odtwarzać pliki zapisane w tym formacie, należy zainstalować właściwy kodek, np. AVI ReComp 1.5.5. Pliki mają rozszerzenie *.avi*.

Format MPEG
Jest to format stratnej kompresji plików audio i wideo. Zaletą tego formatu jest duża kompresja bez straty jakości obrazu. Kolejne wersje tego standardu to **MPEG-1**, **MPEG-2** i **MPEG-4**. Format MPEG-4 został przystosowany do kompresji danych strumieniowych. Format **MPEG-7** to standard opisu danych multimedialnych, natomiast format **MPEG-21** jest standardem, który ma być wprowadzony w przyszłości; przyczyni się on do dalszej standaryzacji treści multimedialnych. Pliki mają rozszerzenie *.mpg* lub *.mpeg*.

Format Windows Media Video
Windows Media Video to format opracowany przez firmę Microsoft dla plików audio i wideo. Charakteryzuje się dużym stopniem kompresji. Pliki wideo mają rozszerzenie *.wmv*.

Format DivX
Jest to format stratnej kompresji plików audio i wideo. Obecne wersje zgodne z MPEG-4 pozwalają zapisać na płycie kompaktowej filmy o długości do 90 min i jakości niewiele ustępującej DVD-Video. Format jest wykorzystywany do przesyłania filmów przez internet.

Format QuickTime
Format **QuickTime** został opracowany przez firmę Apple dla plików multimedialnych. Pliki w tym formacie mogą zawierać nagrania audio, wideo, animacje, obrazy, obiekty 3D. Jest to format często stosowany do publikowania filmów w internecie. Pliki mają rozszerzenie *.qt* lub *.mov*.

8.3.4. Programy do edytowania wideo

AVID Media Composer
AVID Media Composer to jedna z najpopularniejszych aplikacji przeznaczonych do montażu i edytowania wideo w czasie rzeczywistym. Program oferuje: zestaw narzędzi do profesjonalnej korekcji barw, narzędzie do usuwania greenscreena (SpectraMatte), natywne wsparcie dla formatów P2, XDCAM, RED i QuickTime. Ponadto edytor umożliwia wysyłanie plików wideo do popularnych serwisów społecznościowych, ma funkcję stabilizowania obrazu oraz pozwala na obsługę wielu pluginów i filtrów.

Avid Studio
Avid Studio to aplikacja przeznaczona do nieliniowego edytowania wideo. Dostępna jest w polskiej wersji językowej. Główne jej zalety to prostota obsługi oraz setki wbudowanych efektów i przejść. Program umożliwia pracę z nieograniczoną liczbą ścieżek audio i wideo, ma wbudowany stabilizator obrazu oraz pozwala eksportować filmy do najpopularniejszych formatów, np. wykorzystywanych w serwisie YouTube, czy do aplikacji QuickTime HD. Oferuje również profesjonalne narzędzia do edytowania kolorów.

Adobe Premiere
Adobe Premiere to bardzo ceniona aplikacja do nieliniowego edytowania wideo. Jest wyposażona w zaawansowane narzędzia i funkcje do łączenia obrazu i dźwięku w jedną całość. Praca odbywa się w czasie rzeczywistym. Narzędzia do edytowania plików audio i wideo umożliwiają precyzyjną kontrolę nad każdym etapem tworzenia projektu. Z łatwością można manipulować obrazem, dźwiękiem, animacjami i grafiką, aby tworzyć filmy wysokiej jakości.

8.4. Planowanie filmu

Przystępując do tworzenia filmu, należy starannie zaplanować etapy pracy.

Pierwszy etap — określenie, co powinien zawierać film.

Drugi etap — opracowanie scenariusza filmu (opis scen i wykorzystywanych materiałów).

Trzeci etap — gromadzenie materiałów (przygotowywanie plików filmowych, plików muzycznych, zdjęć, tekstów).

Czwarty etap — montaż filmu.

Montaż filmu to nie tylko połączenie zebranych materiałów w całość, ale również wycinanie niepotrzebnych scen z plików wideo, dodawanie efektów, dodawanie napisów i synchronizacja ścieżki dźwiękowej.

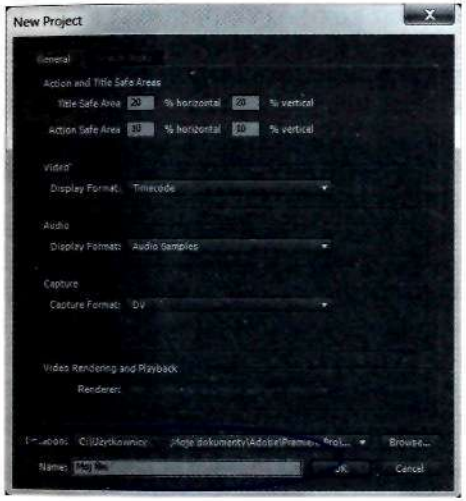
Zadanie 8.2
Przygotuj materiały do filmu pod tytułem *Spacer po moim mieście*. Wykonaj zdjęcia miejsc, które chciałbyś pokazać w filmie. Nakręć klipy filmowe. Do ich tworzenia możesz wykorzystać kamerę, aparat cyfrowy lub telefon komórkowy z funkcją rejestracji filmów. Przygotuj pliki muzyczne. Opracuj scenariusz.

8.5. Tworzenie filmu w programie Adobe Premiere

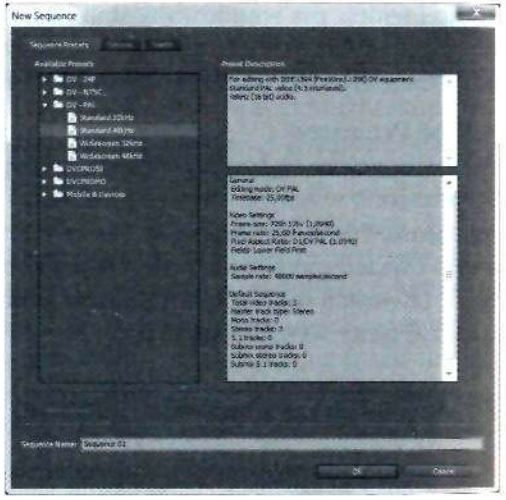
Adobe Premiere to narzędzie dla profesjonalistów tworzących filmy wysokiej jakości, ale również mniej doświadczony użytkownik tego programu doceni jego możliwości i prostotę obsługi.

8.5.1. Środowisko pracy

Po uruchomieniu programu i wybraniu opcji *New Project* zostanie otwarte okno (rysunek 8.15); należy w nim podać nazwę tworzonego projektu i wskazać folder, w którym będzie zapisany.



Rysunek 8.15. Okno New Project

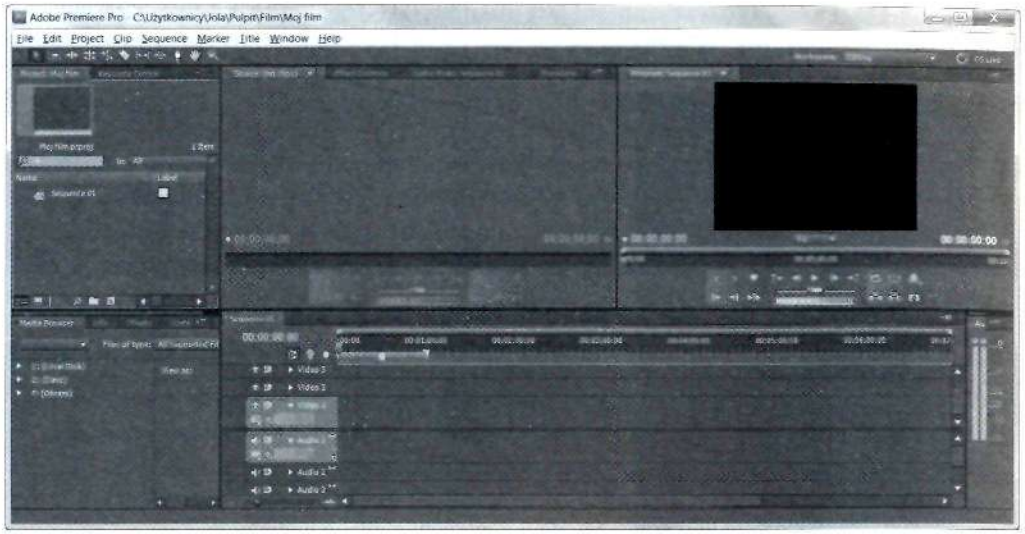


Rysunek 8.16. Definiowanie sekwencji

Gdy klikniemy przycisk *OK*, pojawi się kolejne okno, w którym wyświetlą się profile dostępne w programie (rysunek 8.16).

Wybieramy profil *DV-PAL*, *Standard 48kHz* i podajemy dowolną nazwę sekwencji. Zostanie otwarte okno pracy w programie Adobe Premiere (rysunek 8.17).

Okno pracy składa się z kilku paneli. Najważniejszy jest panel *Timeline* (linia czasu), znajdujący się z prawej strony w dolnej części okna. W nim będą umieszczone odpowiednio zmodyfikowane sekwencje wideo i audio. Tu będziemy przycinać materiał, nakładać efekty i rozmieszczać materiał na ścieżkach wideo i audio. Powyżej tego panelu z prawej strony znajduje się panel *Program* (okno programu), gdzie będzie można oglądać rezultat wykonanego montażu oraz na bieżąco śledzić efekty wprowadzanych modyfikacji. W górnym lewym rogu widoczny jest panel *Project* (okno projektu), w którym zostaną umieszczone materiały potrzebne do tworzenia projektu. Kolejny panel to *Media Browser*, znajdujący się poniżej panelu *Project*. W tym panelu wyświetlana jest



Rysunek 8.17. Okno pracy w programie Adobe Premiere

struktura katalogów, gdzie możemy wyszukiwać potrzebne do pracy pliki. Pod zakładką *Effects* znajduje się panel z listą efektów podzielonych zgodnie z ich przeznaczeniem; wykorzystuje się je do modyfikowania audio i wideo lub do definiowania przejść. Powyżej panelu *Project* widnieje panel *Tools*, zawierający narzędzia do przycinania, modyfikowania i powiększania sekwencji wideo lub audio oraz do wykonywania innych działań. W górnej części, pomiędzy panelami *Project* i *Program*, znajduje się panel zawierający m.in. okno *Effect Controls*. W panelu tym będą wyświetlane parametry efektów dodawanych do sekwencji wideo lub audio.

W programie Adobe Premiere aktywny panel jest zaznaczany żółtym obramowaniem. Aby wyświetlić jego zawartość bez zmiany układu pozostałych paneli, należy po jego zaznaczeniu nacisnąć na klawiaturze klawisz ze znakiem ~ (bez klawisza *Shift*) — wybrany panel zostanie powiększony do rozmiarów okna programu Adobe Premiere. Ponowne naciśnięcie klawisza ze znakiem ~ przywraca poprzedni rozmiar panelu.

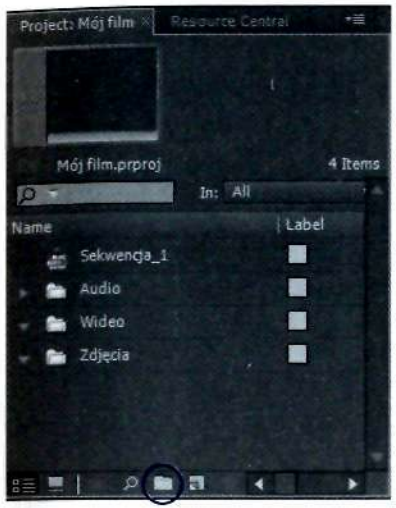
8.5.2. Montaż filmu

Tworzenie filmu rozpoczynamy od przygotowania materiałów. Powinniśmy dysponować własnym materiałem filmowym, zdjęciami i plikami z muzyką.

Import plików

Pierwszym etapem pracy jest zaimportowanie przygotowanego materiału do okna *Project*. W tym celu wybieramy w menu *File/Import* (skrót *Ctrl+I*) i w otwartym oknie znajdujemy potrzebne pliki. Zaznaczamy je i importujemy do panelu *Project*. Pliki można wybierać również w oknie *Media Browser*. Zaimportowane elementy trzeba uporządkować. W panelu *Project* można tworzyć foldery za pomocą ikony *New Bin*

(rysunek 8.18). Dobrym rozwiązaniem jest utworzenie oddzielnych folderów dla plików wideo, plików audio oraz dla zdjęć.



Rysunek 8.18. Ikona New Bin w oknie Project

Zadanie 8.3

Zaimportuj do projektu w programie Adobe Premiere wykonane przez siebie zdjęcia i klipy filmowe oraz pliki audio przygotowane do filmu *Spacer po moim mieście*.

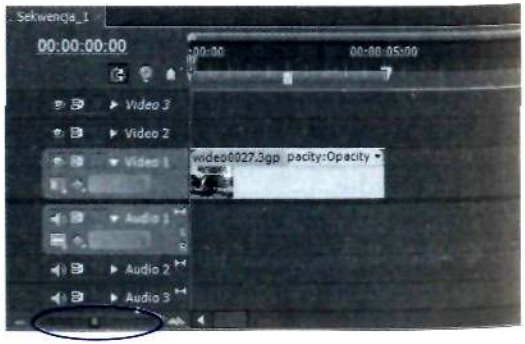
Przycinanie

Podstawowe narzędzia do montażu to `Selection Tool` i `Razor Tool`, dostępne w panelu *Narzędzia*.

`Selection Tool` jest wywoływane klawiszem *V* i służy do wyboru elementu, natomiast narzędzie `Razor Tool` jest wywoływane klawiszem *C* i służy do przycinania fragmentów sekwencji wideo.

Po zaimportowaniu do programu materiałów można rozpocząć pracę nad nimi.

Przenosimy pierwszą sekwencję wideo do panelu linii czasu. W panelu tym domyślnie widoczne są trzy ścieżki wideo i trzy ścieżki audio. Zaznaczamy wybrany element w panelu *Project* i przeciągamy go do panelu *Timeline*, na ścieżkę *Video 1*. Jeżeli widoczna w panelu *Timeline* sekwencja wideo ma niewielkie rozmiary, można powiększyć jej widok, wybierając na klawiaturze klawisz + (plus) bez naciskania klawisza *Shift*.



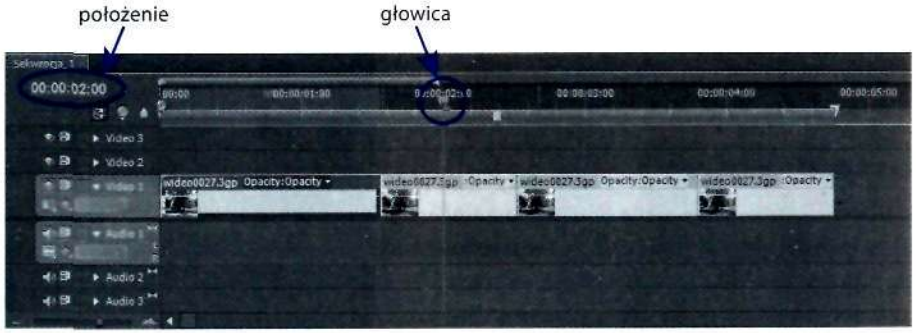
Rysunek 8.19. Okno Timeline z suwakiem do powiększania sekwencji wideo

Aby zmniejszyć widok, należy wybrać klawisz - (minus), również bez naciskania klawisza *Shift*. Ten sam rezultat można osiągnąć, zmieniając położenie suwaka znajdującego się w dolnej części panelu *Timeline* (rysunek 8.19).

Przycinanie sekwencji wideo z przodu lub z tyłu można zrealizować z wykorzystaniem wskaźnika myszy. Jeżeli ustawimy kursor myszy na początku bądź na końcu sekwencji wideo, to zmieni on swój wygląd na czerwony prostokąt. Przy przesuwaniu wskaźnika w prawo lub w lewo pojawi się informacja, o ile klatek element zostanie przycięty. Przycięty materiał można odzyskać, przesuwając wskaźnik myszy w przeciwną stronę.

Drugim narzędziem do przycinania jest `Razor Tool`. Po jego wybraniu kursor myszy zmieni się w symbol żyletki — można wtedy w dowolny sposób przyciąć materiał filmowy.

Montaż materiału nie polega na przypadkowym jego przycinaniu, ale na przycinaniu go w zaplanowanych miejscach. Do określenia miejsca przycięcia służy głowica znajdująca się nad materiałem filmowym. Gdy głowica jest przesuwana, w lewym górnym rogu panelu *Timeline* jest wyświetlane jej położenie (rysunek 8.20).



Rysunek 8.20. Przycinany materiał filmowy oraz położenie głowicy

Po ustaleniu położenia głowicy wystarczy kliknąć narzędziem `Razor Tool` w wyznaczonym miejscu (bez specjalnego dopasowania), aby materiał został w nim przycięty. W celu usunięcia niepotrzebnego fragmentu trzeba przejść w tryb wskazywania, zaznaczyć wybrany fragment filmu i usunąć go (np. klawiszem *Delete*). Usunięty materiał zawsze można odzyskać. Po usunięciu niepotrzebnych fragmentów powstaną w klipie puste miejsca. Można je usunąć, klikając je i naciskając klawisz *Delete*.

Scenopis

Jeżeli tworzony film będzie składał się z wielu elementów (kilku sekwencji filmowych, kilku zdjęć), należy je uporządkować i określić na linii czasu, w jakiej kolejności będą odtwarzane. W tym celu możemy wszystkie materiały umieścić w jednym folderze. Następnie wykorzystujemy narzędzia programu Adobe Premiere do przygotowania scenopisu.

W panelu *Project* wybieramy folder ze zgromadzonymi zdjęciami i klipami filmowymi. Klikamy ten folder dwukrotnie, aby go otworzyć. W dolnej części otwartego panelu znajduje się ikona `Icon View`. Po jej kliknięciu zmieni się sposób wyświetlania zawartości folderu (rysunek 8.21). W tym widoku zawartość folderu może być

modyfikowana — można zmieniać kolejność elementów lub powielać je, aż powstanie gotowy scenopis.



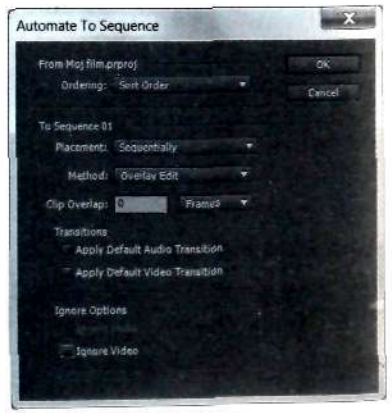
Rysunek 8.21. Tworzenie scenopisu. Ikona Icon View

Zdarza się, że różne fragmenty tego samego klipu filmowego są wykorzystywane w kilku miejscach filmu. W panelu scenopisu można skopiować klip i umieścić jego kopie w wybranych miejscach. To, który fragment klipu będzie odtworzony, ustala się w panelu podglądu (górna część otwartego okna), zatrzymując przeglądanie sekwencji wideo w odpowiednim miejscu. Po kliknięciu ikony *Poster Frame* miejsce to zostanie zapamiętane w scenopisie (ikona obok panelu podglądu — rysunek 8.22).



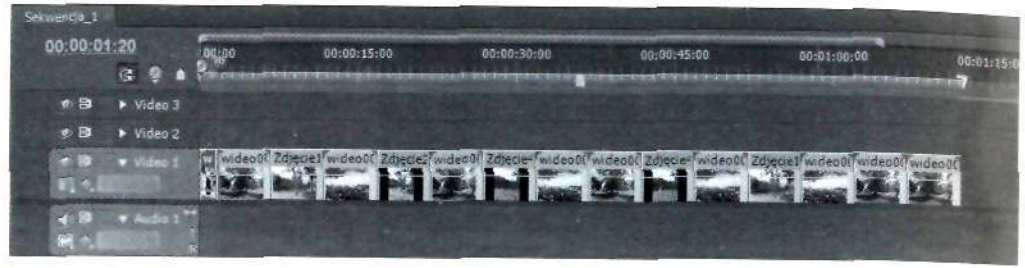
Rysunek 8.22. Fragment okna scenopisu

Po ustaleniu scenariusza filmu można automatycznie cały materiał przenieść do panelu *Timeline*. Aby to zrobić, po zaznaczeniu wszystkich elementów do przeniesienia wybieramy w dole panelu scenopisu ikonę *Automate to Sequence*, co spowoduje otwarcie panelu do modyfikacji ustawień przenoszonego materiału (rysunek 8.23). Po ustawieniu odpowiednich parametrów klikamy *OK* i zamykamy panel scenopisu. Nasz materiał znalazł się w panelu *Timeline* (rysunek 8.24).



Rysunek 8.23. Ustawienia parametrów scenopisu

Kolejny etap pracy to przycinanie i usuwanie niepotrzebnych fragmentów z przygotowanego scenopisu (narzędzia `Selection Tool` i `Razor Tool`).



Rysunek 8.24. Elementy scenopisu w oknie

Pracę może usprawnić narzędzie `Ripple Edit Tool` (z panelu narzędzi), które nie tylko przycina film, ale również usuwa niepotrzebne fragmenty klipu.

W celu zastosowania tego narzędzia należy ustawić przy użyciu głowicy miejsce cięcia, a następnie chwycić myszą za krawędź początkową lub końcową modyfikowanego fragmentu i przeciągnąć do miejsca ustawienia głowicy. W wyniku takiego działania materiał zostanie przycięty, a niepotrzebny fragment filmu usunięty.

Zadanie 8.4

Utwórz scenopis filmu *Spacer po moim mieście*. Wykonaj montaż filmu.

Efekty przejść wideo

W celu uatrakcyjnienia filmu zastosujemy efekty przejść. Pozwolą one na łagodne rozpoczęcie lub zakończenie filmu oraz na uzyskanie efektownych przejść między kadrami.

Efekty przejść mieszczą się z lewej strony panelu *Timeline*, w panelu *Effects*. Wybierając na liście efektów opcję *Video Transitions*, znajdziemy pogrupowane efekty dotyczące sekwencji wideo (rysunek 8.25).

Dodawanie efektu polega na znalezieniu go w panelu *Effects* i przeciągnięciu do panelu *Timeline*, w odpowiednie miejsce filmu.

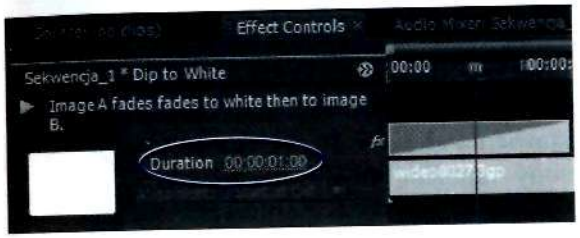


Rysunek 8.25. Okno efektów wideo

Efekty mogą być umieszczane albo na lewej, albo na prawej krawędzi klipu lub na połączeniu między klipami (rysunek 8.26). Nie można ich umieszczać na klipie. Oprócz położenia efektu możemy ustalać czas jego trwania. Najprostszą metodą jest zaznaczenie efektu przejścia w panelu *Timeline* i odpowiednie jego wydłużenie. Czas trwania efektu można również ustawić w panelu *Effect Controls* (powyżej panelu *Timeline*), zmieniając opcję *Duration* (rysunek 8.27).



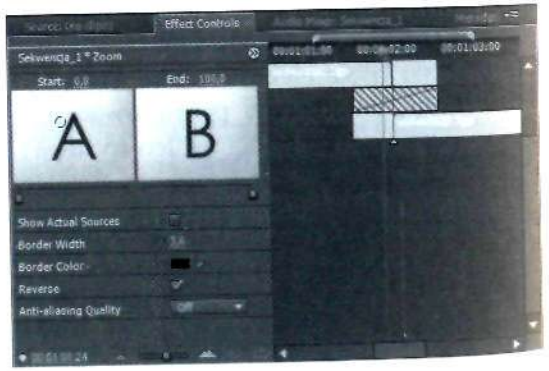
Rysunek 8.26. Nałożenie efektu Dip to Black na końcu filmu



Rysunek 8.27. Czas trwania efektu w panelu Effect Controls

Najczęściej stosowane efekty

- Efekt *Dip to Black* — przejście z obrazu do koloru czarnego. Efekt ten jest zwykle umieszczany na końcu filmu, ponieważ powoduje łagodne wygaszenie obrazu. Może być również umieszczany na początku filmu lub jako przejście między klipami.
- Efekt *Cross Dissolve* — łagodne przejście z jednego kadru w drugi. Efekt jest nakładany na przejście między klipami.
- Efekt *Dip to White* — przejście z obrazu do koloru białego. Jeżeli znacznie skrócimy czas trwania efektu, to uzyskamy efekt błysku między dwoma kadrami. Efekt jest nakładany na przejście między klipami.



Rysunek 8.28. Okno Effect Controls

Efektów stosowanych do wideo jest bardzo dużo. W trakcie pracy nad filmem zwykle przegląda się i testuje te, które mogłyby być użyte. Dostępne są też efekty z dodatkowymi parametrami, które można modyfikować w panelu *Effect Controls* (rysunek 8.28).

Zadanie 8.5

Wybierz efekty przejść i dodaj je do tworzonego filmu *Spacer po moim mieście*. Przetestuj ich działanie.

PYTANIA I POLECENIA KONTROLNE

1. Wymień klika podstawowych formatów plików audio.
2. Na czym polega kompresja stratna plików audio?
3. Na czym polega kompresja bezstratna plików audio?
4. Wymień standardowe rozdzielczości obrazu w plikach wideo.
5. Wymień podstawowe formaty plików wideo.
6. Na czym polega montaż materiału wideo?
7. Wymień kilka efektów dostępnych w aplikacji Adobe Premiere, które można stosować podczas montażu materiału wideo.

ZADANIE

Na potrzeby tworzonej witryny internetowej zawierającej informacje na temat szkoły uczestnicy projektu opracują pliki dźwiękowe, które mogą zostać umieszczone na stronie (np.: krótkie rozmowy, informacje lub pliki muzyczne). Zostaną również przygotowane materiały filmowe pokazujące ciekawe miejsca w szkole. Mogą to być zdjęcia lub klipy filmowe miejsc, które należałoby pokazać na stronie internetowej. Pracując zespołowo, uczestnicy projektu mają odpowiednio zmodyfikować pliki dźwiękowe oraz zmontować film. Do filmu zostaną dodane efekty przejść oraz zostanie przetestowane ich działanie.

9

Testowanie i publikowanie strony

Mimo że strony internetowe wyglądają dobrze i działają prawidłowo, czasami zawierają różnego rodzaju błędy, długo się ładują lub nie są wykrywane przez wyszukiwarki internetowe. Aby ustrzec się przed takimi zdarzeniami, należy projektowaną stronę internetową wielokrotnie i w różny sposób sprawdzić i przetestować. Testowanie i sprawdzanie powinno odbywać się w następujących etapach:

- sprawdzanie poprawności strony,
- walidacja strony,
- testowanie strony w przeglądarkach,
- publikowanie strony,
- testowanie szybkości ładowania strony,
- testowanie funkcjonalności strony,
- optymalizacja strony,
- pozycjonowanie strony.

9.1. Sprawdzanie poprawności strony

Przed umieszczeniem strony w internecie trzeba sprawdzić jej poprawność. Nie chodzi tylko o to, czy strona prawidłowo otwiera się w kilku najpopularniejszych przeglądarkach. Należy sprawdzić, czy na stronie nie ma błędów ortograficznych, czy działają wszystkie odnośniki, czy grafika ładuje się szybko oraz czy strona nie zawiera innych błędów.

Ortografia i interpunkcja

W pierwszej kolejności, korzystając z dostępnych programów do edytowania tekstu, należy sprawdzić, czy tekst, który umieściliśmy na stronie, nie zawiera błędów ortograficznych i stylistycznych. Niektóre programy do tworzenia stron internetowych mają wbudowane słowniki ortograficzne, które już w trakcie tworzenia strony podkreślają błędnie wprowadzany tekst.

Nic bardziej nie zniechęca do odwiedzania strony niż źle sformatowany i zapisany z błędami tekst, niepoprawnie wstawione znaki interpunkcyjne i chaotycznie porozmieszczane spacje.

Nagłówek strony
Należy zadbać, aby nagłówek strony zawierał: tytuł strony, słowa kluczowe, opis strony, a także definicję strony kodowej oraz wersji językowej.

Poprawność hiperłączy
Należy sprawdzić poprawność wszystkich odnośników działających w obrębie strony oraz odnośników do innych stron. Poprawność odnośników trzeba sprawdzić również po umieszczeniu strony na serwerze.

Grafika
Należy pamiętać o tym, aby wszystkie ilustracje umieszczane na stronie były opatrzone zastępczym tekstem (ułatwi to między innymi nawigację osobom niepełnosprawnym, np. niewidomym korzystającym z odpowiednich czytników). W przypadku gdy grafika z jakiegoś powodu nie zostanie wyświetlona na stronie, użytkownik będzie mógł zorientować się, co ona przedstawia.

Ilustracje umieszczane na stronie powinny być zapisane w plikach o odpowiednich formatach. Proste obrazy należy zapisać w formacie GIF. Do zapisu zdjęć i grafik zawierających przejścia tonalne trzeba używać formatu JPG, wybierając odpowiedni procent kompresji. W formacie PNG należy zapisywać grafiki zawierające efekt przezroczystości.

Pliki graficzne powinny zostać zoptymalizowane, tak aby były możliwie małe przy zachowaniu możliwie najlepszej jakości (najlepszego wyglądu). Dzięki temu można zaoszczędzić miejsce na serwerze oraz przyspieszyć wczytywanie stron.

Po umieszczeniu strony na serwerze należy jeszcze raz sprawdzić, czy nie ma problemu z ładowaniem grafiki.

Przeglądarka internetowa
Nie wszyscy użytkownicy internetu korzystają z tej samej przeglądarki. Należy nie tylko sprawdzić, jak strona będzie wyświetlana w różnych przeglądarkach, ale przetestować ją na różnym sprzęcie i w różnych systemach operacyjnych.

Również typ i rozdzielczość monitora wpływają na wygląd strony. Trzeba sprawdzić, jak strona wygląda w zależności od zastosowanej rozdzielczości ekranu oraz typu monitora.

9.2. Walidacja strony

Przeglądarki internetowe są wyposażone w narzędzia do naprawy błędów w kodzie źródłowym. Niestety, nie ma gwarancji, że poprawnie zinterpretują każdy kod. Jeżeli kod zostanie zinterpretowany nieprawidłowo, strona może zostać błędnie wyświetlona lub w ogóle się nie wyświetli. Dlatego powinno się sprawdzać poprawność składni każdego dokumentu HTML i CSS.

W celu sprawdzenia, czy strona jest wolna od błędów, można użyć odpowiedniego programu lub usługi dostępnej w internecie. Walidatory sprawdzają zgodność kodu ze standardami wyznaczonymi przez organizację W3C.

UWAGA

Organizacja W3C (ang. World Wide Web Consortium) opracowuje i zatwierdza standardy dotyczące pisania i przesyłania stron WWW. Zrzesza ona ponad 400 organizacji, firm, agencji rządowych i uczelni z całego świata. Należy pamiętać, że rekomendacje publikowane przez tę organizację nie mają mocy prawnej nakazującej ich użycie.

Walidatory HTML
Działanie walidatora HTML polega na sprawdzeniu kodu HTML strony oraz jego zgodności ze standardami wyznaczonymi przez organizację W3C.

Są różne rodzaje walidatorów. Jedne szukają tylko błędów, inne proponują sposoby ich naprawy, jeszcze inne objaśniają, które instrukcje mogą prowadzić do nieoczekiwanych błędów. Organizacja W3C opracowała własny bezpłatny walidator HTML, który udostępnia na stronie *http://validator.w3.org/*.

Walidatory CSS
Podobnie do walidatora HTML działa walidator kodu CSS. Sprawdza on poprawność kaskadowych arkuszy stylów pod kątem zgodności ze standardami CSS wyznaczonymi przez organizację W3C. Darmowy walidator udostępniany przez organizację W3C znajduje się na stronie *http://jigsaw.w3.org/css-validator/*.

Walidacja strony gwarantuje tylko sprawdzenie kodu pod kątem poprawności składni, natomiast nie gwarantuje poprawnego wyświetlania witryny w różnych przeglądarkach. Wynika to z faktu, że niektóre nowe funkcjonalności proponowane przez twórców przeglądarek nie są częścią standardu W3C. W takim wypadku walidator może ostrzegać o błędach, pomimo że większość popularnych przeglądarek poprawnie wyświetli stronę.

9.2.1. Walidatory

Walidatory skanują dokument w poszukiwaniu niezgodności ze specyfikacją języka oraz błędów w kodzie. Wyszukują brakujące lub nieprawidłowo użyte znaczniki lub błędnie zastosowane style.

Jak już wiemy, plik HTML powinien zaczynać się od deklaracji typu dokumentu (DTD). Specyfikuje on standard języka używanego na stronie, na przykład:

```
<!DOCTYPE html>
```

Poza deklaracją typu dokumentu, która określa, jaka wersja HTML jest używana na stronie, należy ustalić przestrzeń nazw XML:

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="pl">
```

Atrybut `lang="pl"` określa język dokumentu.

Walidator będzie sprawdzał zgodność kodu z zadeklarowanym standardem języka HTML używanego na stronie.

9.2.2. Programy do walidacji stron

Istnieją programy do walidacji stron, z których można korzystać na komputerze użytkownika, pracując w trybie *offline*. Wtedy należy podać nazwę i lokalizację pliku, który ma zostać sprawdzony. Istnieją również programy pracujące w trybie *online*, które sprawdzają strony umieszczone na serwerze i dostępne w internecie. W takim przypadku wystarczy podać adres URL strony oraz opcje jej sprawdzania.

Większość walidatorów, które testują stronę dostępną w internecie, oprócz sprawdzania kodu ma także możliwość sprawdzania czasu ładowania strony, poprawności odnośników czy wyglądu strony w różnych przeglądarkach.

Wave Accessibility Tool
Narzędzie to sprawdza kod strony pod kątem jego dostępności zgodnie z różnymi standardami. Testowaną stronę wyświetla z ikonami sygnalizującymi ewentualne problemy z dostępnością. Strona do testowania może zostać wczytana z dysku twardego lub może być udostępniona *online*. Można również w wybranej przeglądarce zainstalować pasek narzędziowy lub zakładkę, za pomocą której aktualna strona zostanie wysłana do przetestowania.

W3C HTML Validation Service
Jest to darmowy walidator HTML udostępniany przez organizację W3C. Można wysłać bezpośrednio do walidatora dokument zawierający stronę, która ma zostać przetestowana, lub podać adres URL strony. Jest to najbardziej wiarygodne narzędzie do sprawdzania stron pod kątem zgodności ze standardami.

HTML Tidy (HTML Validator)
Ten walidator nie tylko sprawdza kod HTML pod kątem jego zgodności ze standardami, ale również umożliwia naprawę błędnego kodu. Wyświetla też wskazówki dotyczące miejsc wymagających zmiany.

W3C CSS Validation Service
Jest to darmowy walidator CSS udostępniany przez organizację W3C, przeznaczony do walidacji kaskadowych arkuszy stylów. Można go pobrać i używać na własnym komputerze. Można również wysłać do walidatora własne arkusze stylów lub podać adres URL testowanej strony.

W3C Link Checker
Jest to darmowy walidator opracowany przez organizację W3C do testowania odnośników. Gdy podamy adres URL strony, W3C Link Checker wyszuka na niej niedziałające odnośniki.

Xenu Link Sleuth — Find Broken Links
Ten walidator wyszukuje na stronie niedziałające lub uszkodzone odnośniki. Testowaniu może zostać poddana strona, której adres URL zostanie wysłany do walidatora, jak również strona znajdująca się na dysku twardym użytkownika.

Zadanie 9.1
Dokonaj walidacji utworzonej przez siebie strony internetowej. Przeanalizuj otrzymane wyniki. Popraw kod strony.

Podsumowanie
Walidacja strony nie gwarantuje jej poprawnego wyświetlania. Dlatego wśród twórców stron internetowych istnieje duża grupa przeciwników walidacji. Argumentują oni, że nawet po walidacji kodu należy stronę testować w różnych przeglądarkach. Mimo że kod nie ma błędów w składni, nie zawsze działa zgodnie z naszymi oczekiwaniami. Jeżeli natomiast strona jest poprawnie wyświetlana, to nie ma znaczenia, czy została zapisana zgodnie ze standardami. Jej użytkownik nie będzie szukał w kodzie źródłowym błędów języka HTML czy CSS. Prawdziwym testem jest to, jak strona będzie wyświetlana w przeglądarce.

Zwolennicy walidacji twierdzą, że jeżeli strona nie zawiera błędów, to istnieje większe prawdopodobieństwo, że będzie działać dobrze niezależnie od przeglądarki, systemu operacyjnego czy sprzętu, którym dysponuje użytkownik.

Jeżeli strona zawiera błędy, to wyszukiwarki w trakcie wyszukiwania słów kluczowych w różny sposób wykonują działania związane z tymi błędami. Może to spowodować niezaindeksowanie części lub nawet całej strony.

Ostatni i najważniejszy argument to profesjonalizm strony. Błędy na stronie świadczą o małych umiejętnościach jej twórcy.

9.3. Testowanie strony w przeglądarkach

Zanim witryna zostanie opublikowana, należy sprawdzić, jak będzie wyświetlana w różnych przeglądarkach. Nie zawsze istnieje możliwość zainstalowania równocześnie wielu przeglądarek na komputerze. Problemem może być również zainstalowanie wielu wersji tej samej przeglądarki. Wtedy można wykorzystać emulatory przeglądarek internetowych. Jednym z nich jest **Browser Sandbox** dostępny w internecie pod adresem *http://spoon.net/browsers/*. Na liście tego narzędzia znajdują się różne wersje najpopularniejszych przeglądarek (rysunek 9.1).

Gdy klikniemy ikonę wybranej przeglądarki, uruchomi się ona w środowisku wirtualnym. **Spoon.net** obsługuje wtyczki przeglądarek takie jak Firebug i Internet Explorer Developer Toolbar. Ponadto Spoon.net obsługuje konsole debugowania CSS i JavaScript.

Inny emulator przeglądarek internetowych znajdziemy na stronie *http://browsershots.org/*. Przy użyciu tego narzędzia można sprawdzić wygląd strony internetowej dla różnych systemów operacyjnych i w różnych przeglądarkach. Program wykonuje zrzuty ekranu z wybranych wersji przeglądarek.



Rysunek 9.1. Emulatory przeglądarek

Zadanie 9.2
Przetestuj działanie utworzonej strony internetowej w różnych przeglądarkach. Do testowania wykorzystaj dostępne w internecie emulatory przeglądarek. Porównaj efekty wyświetlania strony w tych przeglądarkach.

9.4. Publikowanie strony internetowej

Publikowanie strony internetowej może odbywać się na kilka sposobów:

- umieszczanie strony na płatnym serwerze;
- umieszczanie strony na darmowym serwerze;
- utworzenie własnego serwera (np. XAMPP).

Umieszczanie strony na płatnym lub darmowym serwerze
Usługodawców oferujących miejsce na serwerze jest w internecie wielu. Można wybrać jedną z następujących usług:

- hosting — wynajęcie od dostawcy internetowego miejsca na serwerze do przechowywania plików;
- serwer dedykowany — udostępnienie oddzielnego komputera jako serwera wraz z systemem operacyjnym i wsparciem serwisowym dla jednego usługobiorcy;
- kolokacja — wynajęcie od dostawcy internetowego miejsca na serwer będący własnością usługobiorcy.

Oferty dostępne w internecie różnią się parametrami oraz ceną. Można znaleźć również darmowe usługi hostingu. Najczęściej wiążą się one z dużymi ograniczeniami takimi jak:

- niewystarczająca przestrzeń dyskowa,
- brak dostępu do baz danych,
- brak obsługi języków skryptowych,
- reklamy umieszczane przez usługodawcę.

Po wybraniu jednej z ofert dotyczących miejsca na serwerze należy umieścić stronę lub witrynę internetową w wyznaczonym obszarze. Do tego celu służy kolejna usługa internetowa — protokół FTP.

9.4.1. Protokoły internetowe

Do publikowania stron w internecie niezbędne jest miejsce na serwerze oraz oprogramowanie obsługujące jeden z protokołów sieciowych służących do przesyłania plików. Najpopularniejszymi protokołami sieciowymi są FTP oraz SCP.

Protokół FTP
FTP (ang. *File Transfer Protocol*) jest podstawowym protokołem przeznaczonym do przesyłania plików między komputerami. Wykorzystuje on architekturę klient-serwer, tzn. jeden komputer pełni funkcję serwera, a drugi ma zainstalowane odpowiednie oprogramowanie klienta. Oprogramowanie klienta umożliwia wysyłanie plików na serwer i odbieranie ich z niego.

Oprogramowanie zainstalowane na komputerze klienta często jest nazywane **klientem FTP**. Najprostsze programy FTP są dostępne w systemie operacyjnym; nowsze wersje przeglądarek internetowych bywają wyposażone w funkcje klienta FTP. Do popularniejszych programów obsługujących protokół FTP należą Total Commander i FileZilla. Główną wadą protokołu FTP jest brak szyfrowania danych.

Protokół SSH
SSH (ang. *Secure Shell*) to protokół komunikacyjny służący do zdalnej pracy na komputerze poprzez sieć. Może zostać wykorzystany również do przesyłania plików. Od protokołu FTP różni się tym, że transfer danych jest szyfrowany. Nazwy **SSH** używa się również dla całej rodziny protokołów, które oferują podobne usługi. Przesyłanie plików odbywa się w tym przypadku za pomocą opartego na SSH protokołu SCP (ang. *Secure CoPy* — bezpieczne kopiowanie plików). Inne usługi to zdalna kontrola i tunelowanie (tworzenie dodatkowego zaszyfrowanego kanału).

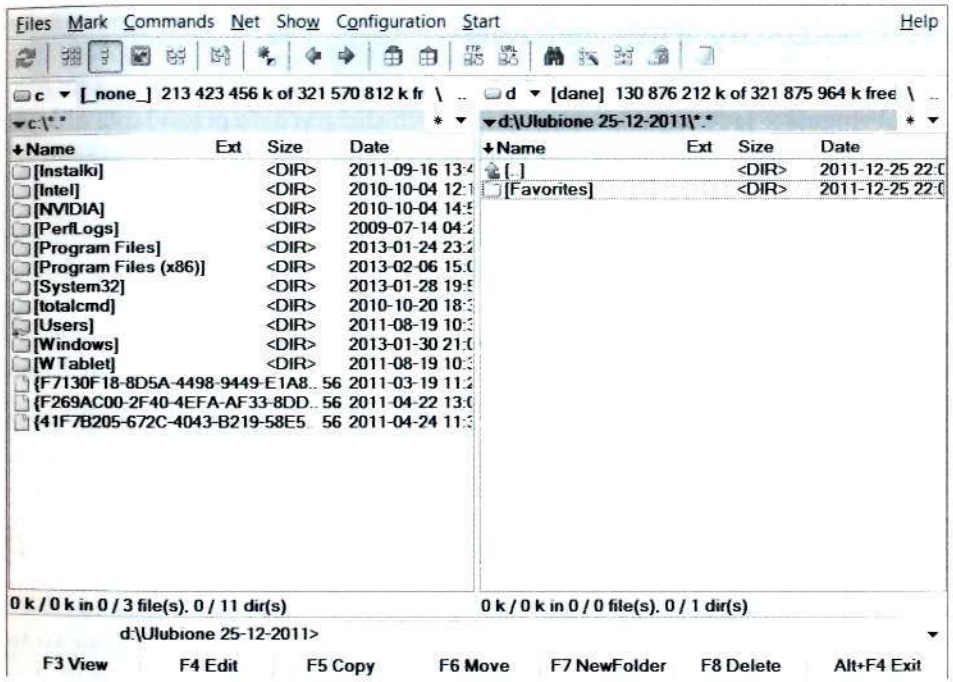
9.4.2. Umieszczanie strony na serwerze

Korzystanie z klienta FTP jest bardzo proste. Po uruchomieniu programu należy zalogować się na konto, a następnie przeciągnąć pliki lub katalogi z zawartością strony z okna klienta do okna serwera.

Total Commander

Total Commander jest popularnym menedżerem plików. Ma również wbudowanego klienta FTP, w związku z czym nadaje się do przesyłania plików zaprojektowanej witryny internetowej na serwer. Aby korzystać z tego narzędzia, trzeba mieć na serwerze konto oraz znać do niego login i hasło.

Po uruchomieniu programu Total Commander (rysunek 9.2) należy wybrać z menu *Net/FTP Connect* (*Sieć/FTP Połączenie*) lub kliknąć ikonę . Zostanie otwarte okno z listą dostępnych połączeń (rysunek 9.3). Po kliknięciu przycisku *New connection* (*Nowe połączenie*) zostanie wyświetlone kolejne okno (rysunek 9.4), w którym należy skonfigurować połączenie z serwerem.



Rysunek 9.2. Okno programu Total Commander



Rysunek 9.3. Okno klienta FTP



Rysunek 9.4. Nowe połączenie

Wpisane dane dotyczące serwera trzeba zatwierdzić.

Teraz na liście połączeń powinno znaleźć się zdefiniowane połączenie. Aby połączyć się z serwerem, wystarczy wybrać to połączenie z listy i kliknąć przycisk *Connect* (*Połącz*).

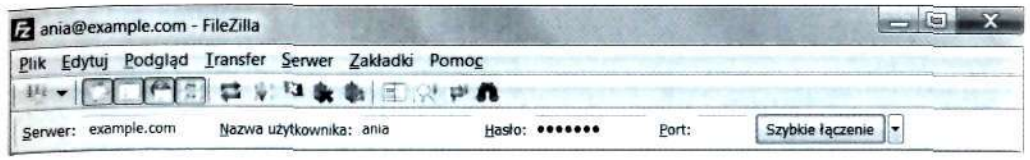
Na serwer należy skopiować całą strukturę witryny, zachowując układ katalogów i plików. Główny plik zawierający stronę startową utworzonej witryny powinien mieć nazwę *index.html*, *index.htm* lub *index.php* (dla serwerów obsługujących język skryptowy PHP).

Struktura katalogów na serwerze może być różna. Zależy to od systemu operacyjnego serwera, oprogramowania obsługującego serwer oraz od administratora. Najczęściej główny katalog serwisu internetowego, w którym należy umieścić witrynę, ma nazwę *public_html*.

FileZilla

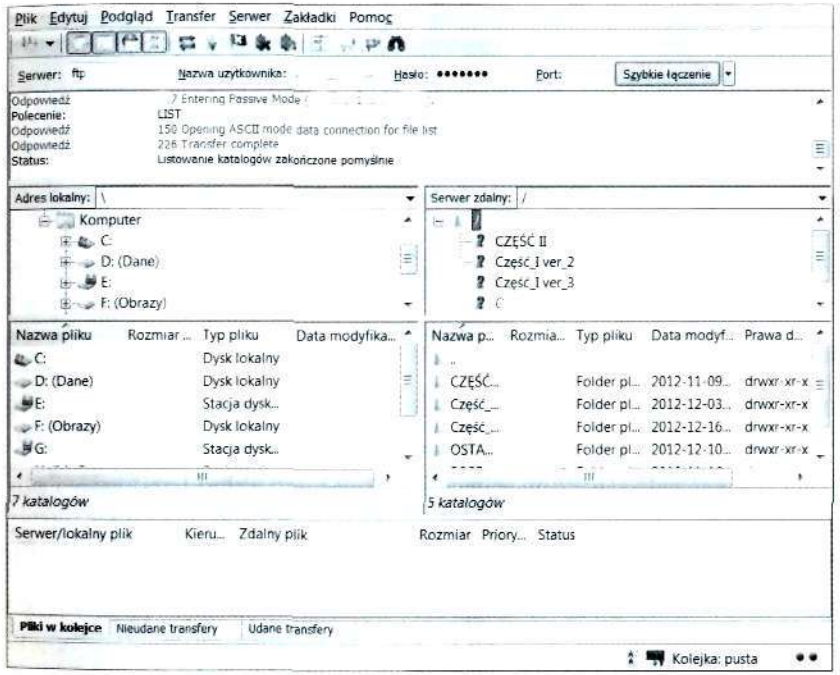
FileZilla jest darmowym klientem FTP dostępnym na zasadach open source. Obsługuje protokoły FTP, FTPS i SFTP oraz współpracuje z protokołem IPv6 i serwerami proxy. Program jest łatwy w obsłudze, a także dostępny w wielu wersjach językowych.

Aby połączyć się z serwerem FTP, po uruchomieniu programu należy w polu *Serwer:* wpisać adres serwera, podać nazwę użytkownika i hasło (jeżeli jest wymagane), podać numer portu (jeśli stosowany port nie jest portem domyślnym) i kliknąć przycisk *Szybkie łączenie* (rysunek 9.5).



Rysunek 9.5. Łączenie się z serwerem FTP

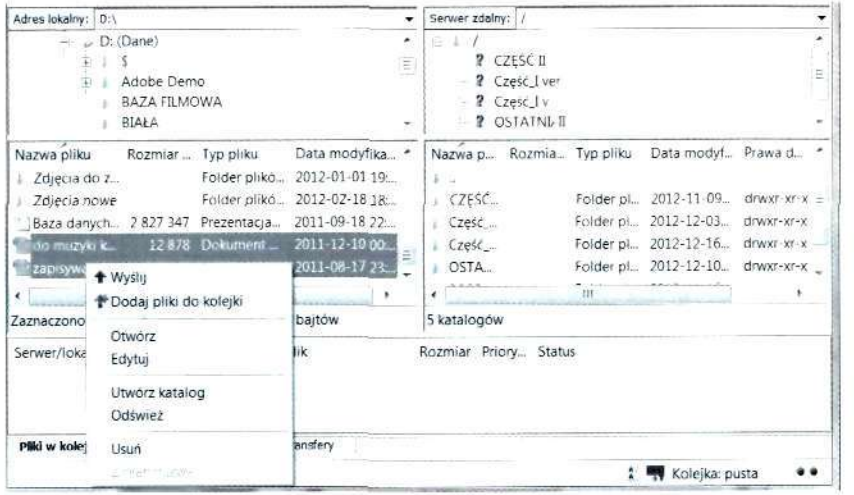
Po połączeniu z serwerem w panelu *Serwer zdalny:* zostanie wyświetlona nazwa dostępnego na serwerze folderu (wraz z drzewem podfolderów). W panelu poniżej wyświetlana jest jego zawartość (rysunek 9.6).



Rysunek 9.6. Połączenie z serwerem

Jednym ze sposobów pobierania plików z serwera lub przesyłania ich na serwer jest ich zaznaczenie i dwukrotne kliknięcie. Zostaną one dodane do kolejki transferu (dolny panel okna). Transfer rozpocznie się automatycznie.

Innym sposobem jest kliknięcie wybranych plików prawym przyciskiem myszy i wybranie z menu kontekstowego opcji *Wyślij* (jeśli pliki mają zostać wysłane na serwer) lub *Pobierz* (jeśli pliki mają zostać z serwera pobrane — rysunek 9.7). Można również przeciągnąć pliki z jednego panelu do drugiego.



Rysunek 9.7. Transfer plików

Jeżeli pliki mają zostać przekazane w późniejszym czasie, można po ich zaznaczeniu kliknąć prawym przyciskiem myszy i z menu kontekstowego wybrać opcję *Dodaj pliki do kolejki*. Po wybraniu z menu *Transfer/Rozpocznij* (*Ctrl+P*) rozpocznie się transfer plików

Zadanie 9.3
Wybierz serwer, na którym zostanie opublikowana przygotowana przez Ciebie strona internetowa. Korzystając z protokołu FTP oraz poznanych narzędzi, umieść swoją witrynę internetową na serwerze.

9.5. Testy szybkości wczytywania strony

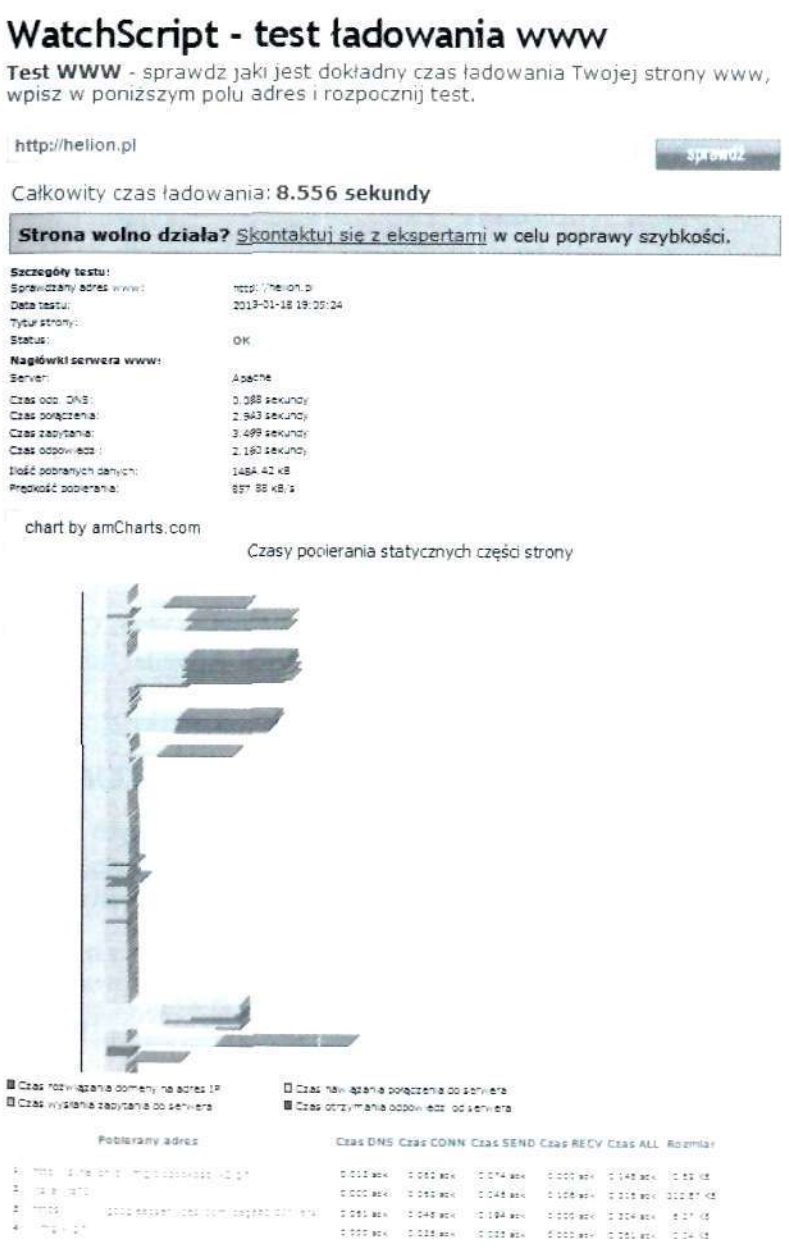
To, jak szybko witryna jest wczytywana, stanowi jeden z najważniejszych elementów jakie wpływają na jej atrakcyjność. Szybkość wczytywania zależy od wielu czynników m.in. od szybkości serwera, na którym została umieszczona strona, elementów multimedialnych znajdujących się na stronie czy błędów w kodzie strony.

W internecie istnieje wiele narzędzi pomagających ocenić prędkość ładowania witryny internetowej oraz narzędzi znajdujących elementy witryny, które powodują największe opóźnienia w jej ładowaniu.

WatchScript
WatchScript to system monitoringu dostępności i zachowania serwisów WWW, serwerów oraz aplikacji internetowych. Narzędzie dostępne jest *online* pod adresem *http://watchscript.pl/*. Wśród darmowych narzędzi WatchScript dostępny jest test ładowania strony internetowej (*http://test.watchscript.pl/*).

Jest to narzędzie do wizualizacji wczytywania poszczególnych elementów strony. Podawane są dokładne dane dotyczące czasu ładowania poszczególnych elementów witryny z podziałem na: czas powiązania domeny z adresem IP, czas nawiązania połączeni

z serwerem, czas wysłania zapytania do serwera, czas otrzymania odpowiedzi od serwera, czas pobierania statycznych części strony (rysunek 9.8).

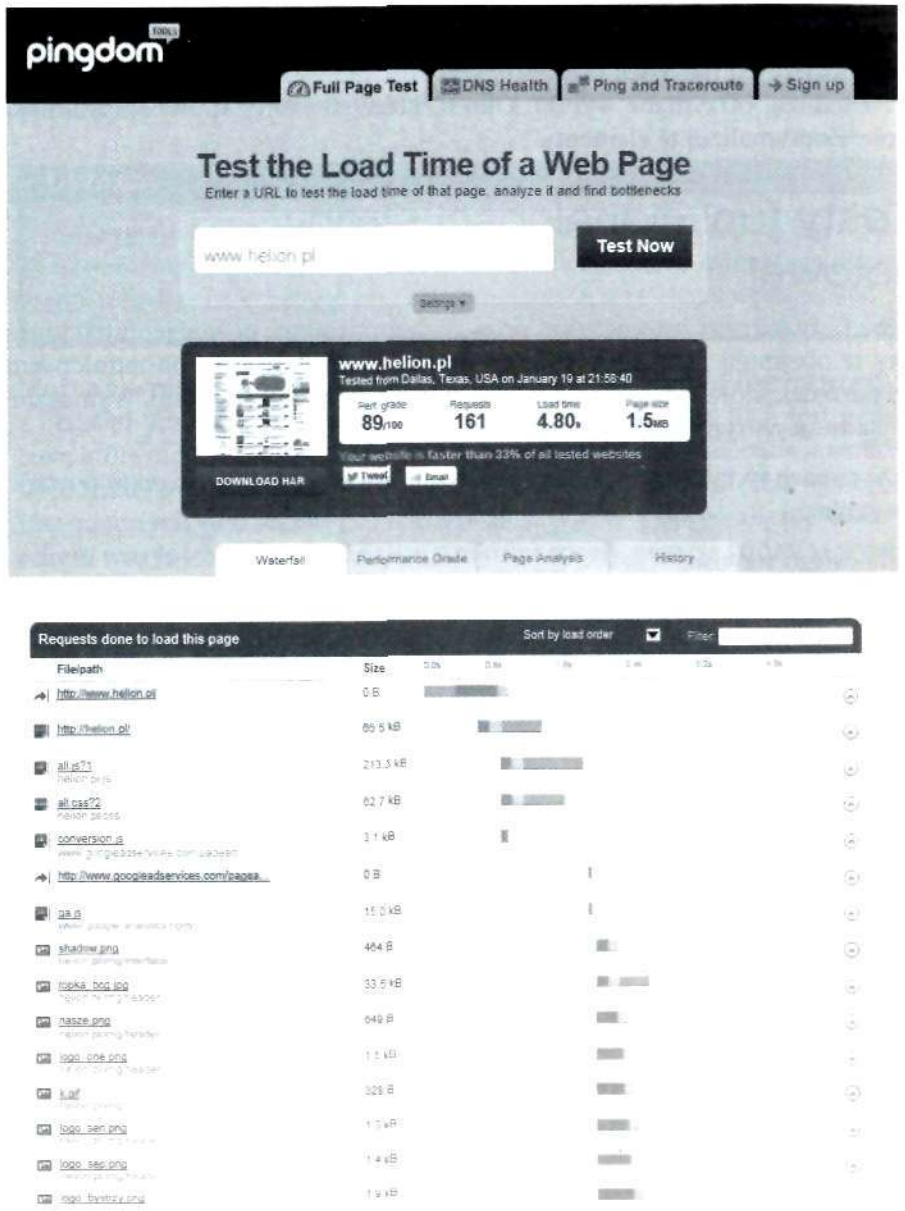


Rysunek 9.8. Test szybkości ładowania przykładowej strony

Po uruchomieniu narzędzia wystarczy podać adres testowanej strony internetowej, a wyniki testu pojawią się automatycznie.

pingdom tools

Innym narzędziem do testowania szybkości ładowania strony jest pingdom tools. Narzędzie jest dostępne pod adresem *http://tools.pingdom.com/fpt/*. Prezentuje parametry poszczególnych elementów witryny wczytywanych w czasie ładowania. Pokazuje wielkość ładowanych elementów witryny, ich umiejscowienie oraz wynikowy czas ładowania całego serwisu (rysunek 9.9).



Rysunek 9.9. Test szybkości ładowania przykładowej strony

Web Page Analyzer

Pod adresem *www.websiteoptimization.com* znajdziemy wiele narzędzi do optymalizowania strony internetowej. Między innymi udostępniane są usługi pozycjonowania stron internetowych, optymalizowania witryny, a także darmowy **Web Page Analyzer** służący do testowania szybkości ładowania strony. Po jej przetestowaniu narzędzie to generuje raport na temat jej poszczególnych elementów. Podaje czas ładowania bloków witryny, jak również czas ładowania witryny przy różnych szybkościach łącza internetowego.

Zadanie 9.4

Korzystając z poznanych narzędzi, przetestuj szybkość ładowania Twojej strony internetowej. Przeanalizuj otrzymane wyniki. Określ, które elementy strony spowalniają jej ładowanie. Zoptymalizuj te elementy.

9.6. Testy funkcjonalności strony internetowej

Profesjonalne testy witryny internetowej pozwalają gruntownie ocenić jej funkcjonowanie. Testy te obejmują badanie wybranej witryny internetowej za pośrednictwem specjalnych narzędzi, które indeksują stronę główną wraz z jej podstronami. Wymagane jest podanie adresu witryny oraz parametrów testu.

Najczęściej w ramach tych testów oceniane są wszystkie elementy serwisu pogrupowane w kilku kategoriach.

Kategorie dla całego serwisu:

- wartość (liczbowa) strony według wskaźnika Google PageRank;
- liczba stron zindeksowanych w wyszukiwarkach Google i Yahoo!;
- liczba linków prowadzących do serwisu poprzez Google i Yahoo!;
- liczba osiągalnych podstron na podstawie zindeksowanej struktury serwisu.

Kategorie dla struktury serwisu:

- liczba linków prowadzących do poszczególnych podstron;
- ślepe odnośniki — błędne linki przekierowujące do nieistniejących stron;
- połączenia RSS i HTTPS.

Kategorie dla podstron:

- wielkość pliku HTML;
- czas ładowania strony;
- walidacja HTML;
- ocena elementów graficznych na stronie pod względem obecności atrybutów `alt` i `title`;
- ocena linków pod względem obecności atrybutu `title`;
- stosunek treści strony do kodu.

Kategorie dla SEO:

- analiza metatagów;
- analiza optymalizacji strony pod kątem kryteriów `title`, `description` i `keywords`;
- ranking słów najczęściej występujących na stronie, w linkach oraz treściach atrybutu `alt` i znacznika `title`;
- analiza pod kątem unikalności treści.

Po przeprowadzonych badaniach zostaje automatycznie wygenerowany raport. Raport najczęściej zawiera oceny testowanych elementów strony oraz ich porównanie ze średnimi wynikami uzyskiwanymi w testach.

UWAGA

SEO (ang. *Search Engine Optimization*) to optymalizacja witryny dla wyszukiwarek internetowych, zwana także pozycjonowaniem. Polega na działaniach prowadzących do uzyskania przez witrynę jak najwyższej pozycji w wynikach wyszukiwania.

Można również przeprowadzać testy funkcjonalności witryny internetowej, korzystając z bezpłatnych narzędzi dostępnych w internecie. Jednym z nich jest *webstatsdomain.com*, które można znaleźć pod adresem *http://www.webstatsdomain.com/domains/www.websitediagnose.com/*. Umożliwia ono przeprowadzenie automatycznego testu stron internetowych pod kątem pozycjonowania WWW i optymalizacji SEO. Po podaniu adresu witryny otrzymamy analizę struktury strony, pozycję strony w wyszukiwarkach i analizę serwisu według rankingu Alexa.com (rysunek 9.10).



Rysunek 9.10. Wynik testowania strony przez webstatsdomain.com

Zadanie 9.5
Korzystając z poznanych narzędzi, przetestuj swoją stronę internetową pod względem funkcjonalności. Przeanalizuj otrzymane wyniki. Określ, które elementy strony powinny zostać poprawione.

9.7. Optymalizacja strony internetowej

Optymalizacja strony internetowej to wszystkie działania dotyczące jej budowy, treści i struktury prowadzące do tego, by znalazła się ona na wysokiej pozycji w wynikach wyszukiwarek internetowych.

Optymalizując treści wchodzące w skład strony, należy zwrócić uwagę na:

- umieszczenie odpowiednich treści w znaczniku `<title>`;
- odpowiednie umieszczenie słów kluczowych w tekstach;
- ustawienie hierarchii nagłówków;
- odpowiednie zaprojektowanie menu;
- odpowiednie zaprojektowanie elementów tworzących linki wewnętrzne witryny;
- ustawienie tekstu zastępczego dla elementów graficznych.

Optymalizując strukturę strony internetowej, należy zwrócić uwagę na:

- stosowanie się na stronie do standardów W3C;
- oddzielenie warstwy logicznej struktury strony od warstwy jej prezentacji (stosowanie zewnętrznych arkuszy CSS);
- zmodyfikowanie elementów strony w celu skrócenia czasu jej ładowania;
- skrócenie długich linków.

Zadanie 9.6
Sprawdź, czy wszystkie elementy utworzonej strony internetowej spełniają wymagania związane z jej optymalizacją.

9.8. Pozycjonowanie strony

Pozycjonowanie strony to działania, których celem jest umieszczenie strony internetowej jak najwyżej w wynikach wyszukiwania. Istnieje wiele technik pozycjonowania strony. Dzielimy je na dwie grupy:

- pozycjonowanie naturalne,
- techniki manipulacyjne (spamerskie).

Pozycjonowanie naturalne polega na podnoszeniu atrakcyjności strony poprzez zamieszczanie ciekawych treści, rozbudowę zawartości strony i dostosowywanie się do zasad działania wyszukiwarek. Ważne jest dobieranie odpowiednich słów i fraz kluczowych, istotna jest popularność strony.

Techniki manipulacyjne polegają na próbie obejścia mechanizmów wyszukiwania w wyszukiwarkach internetowych. Manipulacja może polegać m.in. na zbudowaniu innej zawartości strony dla użytkowników i innej dla automatów wyszukujących. Takie działania mogą doprowadzić do poważnych konsekwencji dla właściciela strony, włącznie z jej usunięciem z indeksu wyszukiwarki.

9.8.1. Działanie automatów wyszukiwarek

Aby strona internetowa była dostępna dla wyszukiwarki internetowej, musi znaleźć się w jej indeksie. Każda wyszukiwarka jest wyposażona w specjalny program zwany **robotem** (inne określenia to **bot, pająk**), którego zadaniem jest ciągłe przeszukiwanie internetu. Robot odczytuje zawartość strony oraz odnajduje na niej odnośniki do innych stron i odczytuje zawartość stron, do których te odnośniki prowadzą.

Zebrane w ten sposób informacje są indeksowane, katalogowane i umieszczane w bazach danych wyszukiwarki. W bazach tych umieszczane są również kopie stron, co pozwala na wyświetlanie strony z pamięci podręcznej wyszukiwarki. Są wyszukiwarki, które indeksują zawartość całej strony internetowej, oraz takie, które indeksują tylko stronę główną. Roboty w regularnych odstępach czasowych, ustalonych przez algorytm wyszukiwarki, odczytują wszystkie strony, aby zaktualizować zawartość indeksów. Strony, których zawartość jest często zmieniana, są aktualizowane częściej. Na przykład strony serwisów informacyjnych mogą być odwiedzane przez roboty wyszukiwarek nawet co kilka lub kilkanaście minut, natomiast serwisy, których treść rzadko się zmienia, są odwiedzane co kilka dni. Wejście robota na stronę jest określane terminem *crawl*.

Informacje, które po wejściu na stronę zebrał robot, są poddawane analizie według specjalnych algorytmów, a wyniki analiz są zwracane w wyszukiwarkach internetowych. Wyliczenia dotyczące ważności strony są dokonywane na podstawie liczby odnośników prowadzących do danej strony. Analizowana jest również zależność między wyrazami znajdującymi się na stronie głównej i na podstronach. Dla oceny strony ważne są również: częstotliwość jej aktualizacji, wiek witryny oraz jej charakter.

Głównymi czynnikami, które wpływają na pozycję strony, są:

- liczba i jakość linków prowadzących do strony,
- słowa kluczowe w linkach prowadzących do strony,
- budowa kodu strony,
- treść strony,
- częstość odwiedzania strony.

Wyszukiwarki stosują do wyszukiwania stron różne algorytmy. Sposób działania robotów każdej z wyszukiwarek jest inny. Z tych powodów wyniki wyszukiwania dla tych samych kryteriów w zależności od wybranej wyszukiwarki mogą się znacznie różnić.

Należy również zdawać sobie sprawę z tego, że wiele stron, które ładnie wyglądają w oknie przeglądarki, jest zupełnie nieczytelnych dla robota wyszukiwarki. Roboty wyszukiwarki „nie widzą" tekstów zawartych w elementach graficznych, nie rozpoznają

menu graficznego. Niektóre z nich traktują Flash jako grafikę, w związku z czym nie potrafią odczytać tekstu z takich stron.

Zadanie 9.7
Sprawdź, czy Twoja strona internetowa jest znajdowana przez różne wyszukiwarki internetowe. Zastanów się, co zrobić, aby podnieść pozycję strony w różnych wyszukiwarkach.

9.8.2. Dodawanie strony do indeksu wyszukiwarki

Jedną z metod dodawania strony do indeksu wyszukiwarki jest wypełnienie specjalnego formularza na stronie wybranej przeglądarki (np. dla przeglądarki Google jest to strona *www.google.pl/addurl*). Nie jest to najlepszy sposób rejestracji strony, ponieważ strona może zostać dodana do bazy dopiero po paru tygodniach od daty rejestracji.

Lepszą metodą jest umieszczenie łącza do nowej strony na innej, działającej już stronie lub umieszczenie jej w katalogu stron, tak aby robot wyszukiwarki na nią trafił. Najlepiej, aby łącze do nowej strony znalazło się na stronie, która jest popularna i często odwiedzana.

Ponieważ roboty niezbyt dobrze radzą sobie z linkami w skryptach JavaScript i ze stronami dynamicznymi, korzystnie jest zamieszczać linki w postaci znacznika `<a href ...>`.

Niektóre wyszukiwarki mają narzędzia do sprawdzania, czy strona jest aktualnie indeksowana przez wyszukiwarkę. Można też w polu zapytań wyszukiwarki umieścić polecenie `site: adres_strony`.

Ponieważ o pozycji strony w wyszukiwarce decyduje głównie liczba zewnętrznych linków, ważne jest, aby takich linków było jak najwięcej. Najprostszą metodą zwiększania liczby linków jest umieszczenie strony w katalogu. W internecie istnieje wiele różnego rodzaju katalogów stron internetowych. W większości z nich umieszczenie wpisu jest bezpłatne.

Zadanie 9.8
Dodaj swoją stronę internetową do wyszukiwarki Google. Aby to zrobić, wypełnij specjalny formularz dostępny na tej stronie.

9.8.3. Mapy strony

Istotnym elementem witryny jest umieszczona na niej mapa strony (ang. *Site Map*). Mapa strony to specjalny dokument XML zawierający strukturę utworzonej witryny internetowej. Zawiera ona w hierarchicznej formie listę podstron znajdujących się w witrynie.

Mapy strony pozwalają poinformować użytkownika o stronach, które są trudne do odnalezienia dla robotów wyszukujących. Są szczególnie przydatne, gdy witryna:

- jest nowa,
- zawiera treści generowane dynamicznie,

- zawiera grafikę,
- zawiera podstrony, które nie są dobrze połączone linkami.

Mapę strony można utworzyć samodzielnie, korzystając z prostego edytora tekstu.

Przykład 9.1

```
<?xml version="1.0" encoding="UTF-8" ?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">

  <url>
    <loc>http://www.moje_podroze.pl/index.html</loc>
    <changefreq>monthly</changefreq>
    <priority>0.7</priority>
  </url>

  <url>
    <loc>http://www.moje_podroze.pl/morze.html</loc>
    <changefreq>weekly</changefreq>
    <priority>0.6</priority>
  </url>

  <url>
    <loc>http://www.moje_podroze.pl/gory.html</loc>
    <changefreq>daily</changefreq>
    <priority>0.5</priority>
  </url>

</urlset>
```

Znacznik `<loc>` określa plik, który ma zostać uwzględniony w mapie strony.

Znacznik `<priority>` określa priorytet danego adresu w stosunku do innych adresów witryny. Priorytetem jest wartość od `0.0` do `1.0`. Najwyższy priorytet powinna mieć strona główna serwisu, a wartość domyślna to `0.5`.

Znacznik `<changefreq>` określa częstotliwość zmian na stronie i może przyjmować wartości:

- `always` — zawsze (przy każdym otwarciu strony),
- `hourly` — raz na godzinę,
- `daily` — raz dziennie,
- `weekly` — co tydzień,
- `monthly` — raz w miesiącu,
- `yearly` — raz do roku,
- `never` — nigdy.

Podana częstotliwość jest tylko wskazówką dla robota. Nie można wymusić na nim określonej częstotliwości indeksowania strony.

W internecie istnieją gotowe narzędzia do tworzenia mapy strony (np. *http://www.xml-sitemaps.com/*). Również w serwisie Google można znaleźć kreator generowania mapy strony, który przeprowadzi użytkownika przez proces jej tworzenia. Efektem działania takich narzędzi jest powstanie pliku *sitemap.xml*, który zawiera mapę strony i powinien zostać umieszczony w katalogu głównym tworzonej witryny.

Zadanie 9.9
Korzystając z narzędzi dostępnych w internecie, utwórz mapę swojej witryny i dołącz ją do projektowanej strony internetowej.

PYTANIA I POLECENIA KONTROLNE

1. Na czym polega sprawdzenie poprawności strony internetowej?
2. Do czego służą walidatory strony internetowej?
3. Jakie narzędzia można stosować do walidacji strony internetowej?
4. Dlaczego utworzoną stronę internetową należy testować w różnych przeglądarkach?
5. Jakie znasz metody publikowania strony internetowej?
6. Od czego zależy szybkość wczytywania strony internetowej?
7. Na czym polegają testy funkcjonalności strony internetowej?
8. Na czym polega optymalizacja strony internetowej?
9. Podaj zasady działania automatów wyszukiwarek internetowych.
10. Na czym polega pozycjonowanie strony internetowej?
11. W jakim celu tworzy się mapy strony internetowej?

ZADANIE

Na podstawie projektu graficznego otrzymanego po zrealizowaniu zadania podsumowującego w rozdziale 6 oraz materiałów zgromadzonych po wykonaniu pozostałych zadań podsumowujących uczestnicy projektu utworzą witrynę internetową zawierającą informacje na temat szkoły. Podczas tworzenia witryny zostaną wykorzystane poznane wcześniej narzędzia (np.: język HTML, arkusze stylów CSS, edytor WYSIWYG).

Po utworzeniu witryny zostanie sprawdzona jej poprawność i witryna zostanie umieszczona na wybranym serwerze. Zostanie przeprowadzona walidacja witryny i działanie strony będzie przetestowane w różnych przeglądarkach. Uczestnicy projektu przetestują szybkość ładowania witryny i zoptymalizują jej działanie.

Skorowidz

960 GRID SYSTEM, 233

A

Accessibility Color Wheel, 242
ActionScript, 15, 246, 258
Adobe Dreamweaver, 161, 162, 166, 168, 169, 180
Adobe Flash, 246, 247, 262
  rysowanie, 262, 265
  ustawienia predefiniowane ruchu, 267
Adobe Photoshop, *Patrz:* Photoshop
Adobe Premiere, 283, 284
adres URL, 45, 46, 257
AJAX, 161
akapit, 31, 78
  rozmiary, 119
Alligator Flash Designer, 246
animacja, 64, 179, 199, 200, 245
  Flash, 179, 245, 246, 247
  interaktywność, 246
  mapy, 247
  poklatkowa, 245, 246
  skryptowa, 245, 258, 261
  ustawienia predefiniowane ruchu, 267
  z zastosowaniem klatek kluczowych, 245
aparat fotograficzny, 195
Areo 3W, 161
arkusz stylów kaskadowy, *Patrz:* CSS
ASP, 161
ASP.NET, 161
atrybut, 15, 17, 18, 82
action, 50
align, 35, 40, 46, 65, 91
alt, 46, 49
autobuffer, 71
autocomplete, 63
autofocus, 63
autoplay, 71
autostart, 66
background-attachment, 109
background-color, 108, 111
background-image, 108
background-position, 110
background-repeat, 108, 218
bgcolor, 26, 35, 214
border, 45, 126
border-bottom-color, 115
border-bottom-width, 114
border-collapse, 126
border-color, 115
border-left-color, 115
border-left-width, 114
border-right-width, 114
border-spacing, 128
border-style, 114
border-top-color, 115
border-top-width, 114
border-width, 113
caption-side, 125
cellspacing, 41
class, 151
clear, 40, 135
color, 108
cols, 60
content, 25
controls, 66, 71
coords, 49
cursor, 138
disabled, 61
display, 136, 141
empty-cells, 126
enctype, 51
float, 132, 141
font-family, 99
font-size, 100, 104
font-style, 102, 104
font-variant, 103, 104
font-weight, 103, 104
height, 45, 65, 120
href, 48, 49, 80
hspace, 35, 41, 54
http-equiv, 25
label, 59, 60
language, 74
letter-spacing, 106
line-height, 104, 104
list-style, 125
list-style-image, 124, 125
list-style-position, 124, 125
list-style-type, 122, 125
logiczny, 19
loop, 66, 71
margin, 116
max-height, 120
maxlength, 51
max-width, 119
method, 51
min-height, 120
min-width, 119
multiple, 58
name, 25, 51, 60, 65
opacity, 129
order-right-color, 115
outline, 119
overflow, 121, 150
padding, 117
parametr, 15
playcount, 66

atrybut
pluginspage, 67
position, 128, 131
preload, 71
readonly, 61
required, 63
rows, 60
rowspan, 43
selected, 58
shape, 49
size, 51, 58
src, 45, 65, 71
start, 35
tabindex, 62
table-layout, 126
target, 49
text-align, 105
text-decoration, 105
text-indent, 105
text-transform, 106
titles, 49
type, 35, 36, 51, 53, 54, 55, 64, 74
usemap, 49
valign, 35
value, 35, 51, 58
vertical-align, 132
visibility, 138
volume, 66
vspace, 35, 41, 54
white-space, 107, 141
width, 45, 65, 119
word-spacing, 106
wrap, 61
zdeprecjonowany, 18, 76
Audacity, 274
AVID Media Composer, 283
Avid Studio, 283

B

baner reklamowy, 246, 248
barwa, *Patrz:* kolor
Béziera krzywa, 72
biała przestrzeń, 187
blok, 44, 78, 119, 149, 187
danych, 149
krawędź, 113
rozmiary, 119
szerokość, 139, 150
border, *Patrz:* obramowanie
bpp, 194
brat, 89

C

Cascading Style Sheets, *Patrz:* CSS
cieniowanie, 72
client-side, *Patrz:* język przetwarzania po stronie klienta
CMYK, 196, 198, 201
Complementary Metal Oxide Semiconductor, *Patrz:* matryca CMOS
CSS, 13, 18, 32, 34, 76, 161, 168
edytor, 84
import, 81
kaskadowość, 82
media, 176
menu, *Patrz:* menu
model blokowy, 112
pseudoelement, *Patrz:* pseudoelement
reguła, 145
kontekstowa, 146
składnia, 84
styl lokalny, 77, 82
styl w linii, 77
walidator, 295, 296
wewnętrzny, 77, 79, 82
zewnętrzny, 77, 80, 82, 174
CSS3, 176
czas rzeczywisty, 72
czcionka, 99, 100, 101, 102, 103, 104, 105
bezszeryfowa, 99
cursive, 99
dekoracyjna, 99
fantasy, 99
grubość, 103
jednostki, 101
monospace, 99
monotypiczna, 99
o stałej szerokości, 99
pochyła, 99
rozmiar, 100
sans-serif, 99
serif, 99
styl, 102
szeryfowa, 99
wariant, 103

D

deklaracja, 85
Document Type Definition, *Patrz:* DTD
dokument, 74
HTML, 13, 14, 15, 22, 23, 77, 162
ciało, 26
formatowanie, 27
tworzenie, 27
strona, *Patrz:* strona dokumentu
struktura, 16
DTD, 16, 17
Frameset, 17
wersja
przejściowa, *Patrz:* wersja Transitional
ramkowa, *Patrz:* wersja Frameset
Strict, 17
ścisła, *Patrz:* wersja Strict
Transitional, 17, 18
działanie dynamiczne, 74
dziecko, 89
dziedziczenie, 83
dźwięk, 19, 64, 71, 271
edytowanie, 274, 276
obwiednia, 280
szum, *Patrz:* szum

E

edytor
CSS, 84
graficzny, 14
map odnośników, 225
tekstu, 13, 14, 21
Notepad++, *Patrz:* Notepad++
Pico, *Patrz:* Pico
Text, *Patrz:* Text
wizualny, *Patrz:* edytory graficzny
WYSIWYG, 160
Adobe Dreamweaver, *Patrz:* Adobe Dreamweaver
Areo 3W, *Patrz:* Areo 3W
Visual Web Developer 2010 Express Edition, *Patrz:* Visual Web Developer 2010 Express Edition
WYSIWYG Web Builder, *Patrz:* WYSIWYG Web Builder
efekt
czerwonych oczu, 207
schodkowania, 193
element
graficzny, *Patrz:* grafika
nadrzędny, 83, 88, 89
nakładanie, 131
obcinanie, 131
podrzędny, 83, 88, 89
pusty, 19
ukrywanie, 138
wyrównanie
w pionie, 132
w poziomie, 132
wyświetlanie, 136, 138
zastępowany, 136
etykieta, 56
Extensible HyperText Markup Language, *Patrz:* XHTML
Extensible Markup Language, *Patrz:* XML

F

FileZilla, 301
Flash Player, 179
format
aiff, 271, 272
AIFF, 271
AU, 271
audio, 271, 272
AVI, 65, 282
BMP, 193, 199
cdr, 192
css, 80, 84
DivX, 282
flv, 179
GIF, *Patrz:* GIF
H.264, 273
JPEG, *Patrz:* JPEG
mid, 65
MIDI, 65, 272
mov, 65
MP3, 65, 271
MPEG, 65, 282
MPEG-1 Audio Layer 3, 271
mpg, 65, 282
ogg, 271, 272
pliku graficznego, 199
PNG, *Patrz:* PNG
PSD, *Patrz:* PSD
QuickTime, 282
RealMedia, 272
SVG, *Patrz:* SVG
SWF, 246, 257
TIFF, *Patrz:* TIFF
Vorbis, 271, 272
wav, 65, 271
Windows Media Video, 282
WMA, 271, 272
wmv, 65, 282
formularz, 50, 51, 62, 63, 73, 74
grupowanie elementów, 57
lista rozwijana, 50
pole
nieaktywne, 61
tekstowe, 50, 52
wyboru, 50
przycisk opcji, 50, 52
FTP, 299
funkcja
fillRect, 73
strokeRect, 73

G

GIF, 44, 64, 72, 161, 193, 199, 200, 201, 227, 228, 229, 230
animowany, 246
głębia koloru, 194
Google PageRank, 306
GoogleMaps, 179
gradient, 72, 205, 214, 218
grafika, 44, 45, 160, 180
animowana, 245, 246
bitmapowa, *Patrz:* grafika rastrowa
druk, 198
na potrzeby strony internetowej, 200, 208, 211, 217, 220, 225, 227, 228, 232
obiektowa, *Patrz:* grafika wektorowa
rastrowa, 72, 193, 199
rozmiary, 119
wektorowa, 192, 247

H

hiperłącze, *Patrz:* odsyłacz
Hippo Animator, 246
histogram, 194, 204, 205
HSL, 87
HSV, 196, 199
HTML, 13, 76, 161, 245
walidator, 295, 296
HTML Tidy, 296
HTML 4, 17, 18, 32
HTML 5, 15, 19, 28, 63, 71
HyperText Markup Language, *Patrz:* HTML

I

identyfikator, 139, 147
ikona, 220, 236
immediate mode rendering, *Patrz:* rendering trybu natychmiastowego
indeks
  dolny, 29
  górny, 29
inline, *Patrz:* CSS styl w linii
iTunes, 273

J

JavaScript, 15, 64, 72, 73, 161, 257
język
  ActionScript, *Patrz:* ActionScript
  CSS, *Patrz:* CSS
  HTML, *Patrz:* HTML
  JavaScript, *Patrz:* JavaScript
  JScript, *Patrz:* Jscript
  Perl, *Patrz:* Perl
  PHP, *Patrz:* PHP
  przetwarzania
    po stronie
      klienta, 64
      serwera, 64
  skryptowy, 15, 73, 74, 246
  VBScript, *Patrz:* VBScript
  XHTML, *Patrz:* XHTML
  XML, *Patrz:* XML
  znaczników, 76
JPEG, 44, 45, 64, 72, 193, 195, 199, 201, 227, 228, 229
JScript, 15
JSP, 161

K

kadrowanie, 203
kanał alfa, 72, 198, 200, 254
karta pamięci, 195
kaskadowość, *Patrz:* CSS kaskadowość
klasa, 93, 95, 139, 148
klatka kluczowa, 245
klawiatura, 73
klawisze skrótów, 62
klient FTP, 161, 298, 299
kodowanie znaków, 18
  polskich, 20, 22
kolor, 33, 72, 86, 87, 108, 150
  głębia, *Patrz:* głębia koloru
  model barw, 196
  paleta, 34, 193, 238
    24-bitowa, 45, 197
    48-bitowa, 45
    redukcja, 45
kolumna, 135
komentarz, 30, 85
kompresja, 45, 199
  bezstratna, 193, 200
  dźwięku, 272
    bezstratna, 272, 273
    stratna, 271, 272
  stratna, 193, 199
krzywa
  Béziera, 72
  drugiego stopnia, 72
Kuler, 238
kursor, 73

L

linia pozioma, 29
link, *Patrz:* odsyłacz
lista, 35, 78, 143
  atrybuty, 125
  definicji, 35, 37
  nieuporządkowana, 139
  numerowana, 35, 122
  punktowana, 35, 36, 122
  rozwijana, 63
  zagnieżdżona, 35, 37, 59
logo, 220, 221, 232, 233

M

mapa
  animowana, 247
  bitowa, 193
mapowanie obrazu, 224
margines, 116, 139
  wewnętrzny, 117, 149
  zewnętrzny, 118, 149
marker, *Patrz:* punktor
maska, 200
matryca CMOS, 195
menu, 139, 224, 232, 233
  animowane, 246, 258
  pionowe, 139
  poziome, 140
  zagnieżdżone, 143
metajęzyk, 16
model
  CMYK, *Patrz:* CMYK
  HSB, *Patrz:* HSV
  HSV, *Patrz:* HSV
  RGB, *Patrz:* RGB
multimedia zewnętrzne, 179
mysz, 73, 138, 246, 255
  kursor, 73, 96
  przycisk, 73, 97, 258

N

nagłówek, 23, 24, 32, 79, 80, 187, 199, 294
nieprzezroczystość, 129, *Patrz też:* przezroczystość
Notepad++, 21

O

obramowanie, 45, 77, 113, 114, 119, 149
  atrybuty, 116
  kolor, 115
  styl, 114
  szerokość, 113
obrys, 119
odnośnik, 49, 80, 294
odstęp między wierszami, 104
odsyłacz, 48, 69, 95, 232, 257
  aktywny, 97
  graficzny, 49
  mapa, 49, 224
  odwiedzony, 96
  podstawowy, 96

P

padding, *Patrz:* margines wewnętrzny
Pajączek, 225
pasek nawigacyjny, 139
Perl, 64
Photoshop, 193, 200, 201, 202, 203, 206, 207, 208, 217, 218, 220, 228, 241
PHP, 64, 161
PHP5, 161
pica, 102
Pico, 21
piksel, 101, 117, 193
  jasność, 194
pingdom tools, 305
plik
  AIFF, 271, 272
  AU, 271
  CSS, 80, 84
  index.htm, 23
  index.html, 23
  konfiguracyjny, 16
  lokalny, 180
  MP3, 271
  multimedialny, 64, 71
  Wave Form Audio Format, 271
  wideo, *Patrz:* wideo
  Windows Media Audio, 271
plug-in, *Patrz:* wtyczka
PNG, 44, 45, 64, 72, 193, 200, 201, 227, 229, 230
podpis, 125
pole
  BUTTON, 55
  INPUT, 51, 56
  SELECT, 58
  tekstowe, 73
  TEXTAREA, 60
polecenie, 15, 23, 24, 27
  !important, 82
  @import, 81, 84
portal
  internetowy, 14
  społecznościowy, 233
  wertykalny, *Patrz:* wortal internetowy
potomek, 88
pozycjonowanie, 128
protokół
  FTP, 299
  sieciowy, 16
  SSH, 299
  wywoływania zdalnego dostępu do obiektów, 16
przeglądarka internetowa, 13, 18, 65, 74, 257, 294, 297
  Internet Explorer, 142, 257
  obsługa błędów, 19
przepełnienie, 121
przeplot, 45
przesyłanie strumieniowe, 272
przetwornik analogowo-cyfrowy, 195
przetwornik optoelektroniczny, 195
przezroczystość, 45, 72, 129, 198, 199, 227, 254
przycisk, 199, 222, 237
  animowany, 246, 252, 255
  powiązanie z hiperłączem, 257
PSD, 200
pseudoelement, 98
  :first-letter, 98
  :first-line, 98
  pierwsza linia, 98
  pierwsza litera, 98
pseudoklasa, 95, 96
  :active, 96
  :focus, 98
  :hover, 96, 97
  :link, 96
  :visited, 96
punktor, 101, 122, 139
  obrazek, 124
  położenie, 124

Q

QuickTime, 273

R

Radical Image Optimization Tool, 228, 229
radio internetowe, 273
ramka, 150, *Patrz też:* obramowanie
Real Player, 273
reguła stylu, 84
rendering trybu natychmiastowego, 72
RGB, 34, 86, 199, 200
RGBA, 87, 198
rozdzielczość, 194, 196
  ekranowa, 200
  geometryczna, 194
  radiometryczna, 194

S

Scalable Vector Graphics, *Patrz:* SVG
selektor, 77, 84, 85, 87
  atrybutu, 90
    o określonej wartości, 91
    prosty, 91
    zawierającego określony wyraz, 92
  brata, 89
  dziecka, 89
  grupowanie, 128
  identyfikatora, 95
  klasy, 93
  potomka, 88
  pseudoelementu, 98
  typu, 87, 95
  uniwersalny, 87

server-side, *Patrz:* język przetwarzania po stronie serwera
serwer, 14, 298, 299
  DNS, 14
  FTP, 14
  HTTP, 14
  internetowy, 14
Simple Object Access Protocol, *Patrz:* SOAP
Site Map, *Patrz:* strona internetowa mapa
skaner, 196
skrypt, 73, 74
slider, 233, 236
SOAP, 16
SQL, 161
SSH, 299
standard
  ASCII, 20
  ISO 8859, 20
  UCS, 20
  Unicode, 20
  UTF, 20
  UTF-8, 20, 23
  Windows-1250, 20
stopka, 187, 232, 233
strona
  dokumentu, 31
  internetowa, 14, 234
    animacja, *Patrz:* animacja
    architektura, 187, 190
    dynamiczna, 14
    implementacja, 186
    indeksowanie, 24, 306
    mapa, 310
    nagłówek, *Patrz:* nagłówek
    nawigacja, 222
    optymalizacja, 308
    opuszczanie, 74
    pozycjonowanie, 308, 309
    projektowanie, 234
    publikowanie, 181, 293, 298, 299
    specyfikacja, 188
    statyczna, 14
    szablon, 145, 150, 151, 155, 169, 232, 233
    test funkcjonalności, 306
    test szybkości wczytywania, 303
    testowanie, 293, 297
    układ, 187, 232, 233
    użyteczność, 186, 232
    walidacja, 294, 296
    zasady projektowania, 185, 187, 190
  o stałej szerokości, 150
  o zmiennej szerokości, 155
  startowa, 23
suwak, 121
SVG, 192, 200
SWiSH Max4, 246
system
  HDTV, 281
  NTSC, 281
  PAL, 281
szum, 280

T

tabela, 39, 125, 160, 166
  komórka, 39
    łączenie, 42
    odstęp, 41, 128
    szerokość, 39
    wyrównanie w pionie, 43
    wyrównanie w poziomie, 43
  krawędź, 126, *Patrz też:* obramowanie
  odstęp, 41
  otoczona tekstem, 40
  podpis, *Patrz:* podpis
  rozmiary, 119
  szerokość, 39
  wiersz, 39
    zagnieżdżona, 43
tablet, 176
tag, *Patrz:* znacznik
Tagged Image File Format, *Patrz:* TIFF
tapeta, 216, 217
tekst
  animowany, 254
  efekty, 105, 254
  formatowanie, 77, 99, 100, 101, 102, 103, 104, 105, 106, 164
  wcięcie pierwszego wiersza, 105
  zastępczy, 46, 49
telefon komórkowy, 176
Text, 21
TIFF, 193, 200
tło, 23, 26, 27, 44, 45, 76, 77, 108, 110, 112, 150, 207, 208, 211
  formatowanie, 77
  strony internetowej, 214, 217
top na stronę internetową, 246
Total Commander, 300

V

VBScript, 64, 73
vertical portal, *Patrz:* wortal internetowy
Visual Studio 2010 Express, 161
Visual Web Developer 2010 Express Edition, 161

W

W3C CSS Validation Service, 296
W3C HTML Validation Service, 296
W3C Link Checker, 296
walidator, 295
warstwa, 200, 208, 250
  prezentacyjna, 76
  semantyczna, 76
  strukturalna, 76
  styl, 210
wartość, 77
WatchScript, 303
Wave Accessibility Tool, 296
Web Page Analyzer, 306
wideo, 19, 64, 71, 179, 180, 271, 273, 284, 285
Winamp, 273
Windows Media Player, 273
witryna internetowa, 14
właściwość, 77, 85, 99
  wartość, 85, 86
wortal internetowy, 14
wtyczka, 64, 71, 179
wykres, 72
WYSIWYG Web Builder, 161
wyszukiwarka, 24, 309
  indeks, 309, 310
wyświetlacz LCD, 195

X

Xenu Link Sleuth, 297
XHTML, 14, 15, 17, 161
XHTML 1.0, 18
XHTML 1.1, 17, 18
XML, 15, 16
  składnia, 17

Z

zdarzenie, 73
  onblur, 73
  onchange, 73
  onclick, 73
  ondblclick, 73
  onfocus, 73
  onkeydown, 73
  onkeypress, 73
  onkeyup, 73
  onload, 73
  onmousemove, 73
  onmouseout, 73
  onmouseover, 73
  onmouseup, 73
  onreset, 73
  onselect, 73
  onsubmit, 73
  onunload, 73
  wewnętrzne, 73
zdjęcie
  kadrowanie, *Patrz:* kadrowanie
  klonowanie, 206
  korekcja, 202, 203, 204, 206, 207
  zmiana tła, 207
zmienna
  fillStyle, 73
  strokeStyle, 73
znacznik, 13, 15, 16, 23, 34
  <*>, 149
  <a>, 46, 48, 49, 140
  <audio>, 71
  <b>, 15
  <basefont>, 28, 34
  <big>, 28, 34
  <body>, 28

, 15, 32, 40
  <button>, 55
  <canvas>, 72
  , 32, 35
  <changefreq>, 311
  <dd>, 37
  , 44, 78, 135, 149, 150, 151, 187
  <dl>, 37
  <dt>, 37
  <embed>, 65, 67, 257
  <fieldset>, 57
  <font>, 27, 28, 35
  <form>, 50, 57
  <head>, 23, 24, 79, 80
  <hr>, 15, 29
  <html>, 23
  <i>, 28, 88
  <iframe>, 17
  <img>, 45, 49
  <input>, 51, 55, 61, 64
  <label>, 56
  , 15, 35, 36, 122
  <link>, 85
  <loc>, 311
  <map>, 49
  <meta>, 23, 24, 25
  <object>, 257
  , 35, 122
  <optgroup>, 59
  <option>, 58, 59
  <p>, 31, 91
  <priority>, 311
  <script>, 74
  <select>, 58, 59
  <small>, 28
  <spacer>, 32
  , 78, 145, 146
  <strike>, 29, 35
  <style>, 79, 81
  <sub>, 29
  <table>, 40, 41
  <textarea>, 60
  <td>, 39, 40, 42, 43
  <title>, 23, 24
  <tr>, 39, 42
  <u>, 29, 35, 88
  , 36, 122, 139, 140
  <video>, 71
  atrybut, *Patrz:* atrybut
znak
  --, 19
  &, 19, 30
   , 30
  |, 19
  |, 19
  €, 30
  <, 19, 30
  >, 19, 30
  §, 30
  ©, 30
  ®, 30
  kodowanie, *Patrz:* kodowanie znaków
  specjalny, 30
  ™, 30



Kwalifikacja E.14
Tworzenie stron internetowych

Część 1

Podręcznik do nauki zawodu **technik informatyk**

Technik informatyk nie jest zwykłym użytkownikiem komputerów. Jeśli uczeń wybiera szkołę o takim profilu, z czasem staje się prawdziwym komputerowym ekspertem.

Kwalifikacja E.14 skupia się na tworzeniu aplikacji internetowych oraz pracach związanych z bazami danych. Opracowany z myślą o przyszłych technikach informatyki podręcznik *Tworzenie stron internetowych* stanowi pierwszą z grupy książek przygotowanych pod kątem tego materiału. Podręcznik ten jest zgodny z najnowszą podstawą programową. To kompletne kompendium, zawierające teorię popartą licznymi przykładami, lecz prowokujące także do samodzielnej pracy i zdobywania praktycznych umiejętności tworzenia dobrych stron WWW czy ulepszania ich szaty graficznej. Dzięki zdobytej wiedzy każdy z uczniów zaprojektuje własną witrynę internetową, pozna zasady języka HTML oraz wykorzysta kaskadowe arkusze stylów. Dodatkowe informacje dotyczące architektury stron internetowych czy wykorzystania elementów graficznych i animacji przyczynią się do tego, że szkolne projekty osiągną profesjonalny poziom. Strony będą nie tylko estetyczne, lecz także przyjazne dla użytkownika, czytelne i zgodne z regułami walidacji.

Technik informatyk to doskonały, charakteryzujący się wysoką jakością, kompletny zestaw edukacyjny, przygotowany przez dysponującego ogromnym doświadczeniem lidera na rynku książek informatycznych – wydawnictwo Helion.

W skład zestawu *Technik informatyk* wchodzą także:

Kwalifikacja E.12. Montaż i eksploatacja komputerów osobistych oraz urządzeń peryferyjnych. Podręcznik do nauki zawodu technik informatyk

Kwalifikacja E.13. Projektowanie lokalnych sieci komputerowych i administrowanie sieciami. Podręcznik do nauki zawodu technik informatyk

Kwalifikacja E.14. Część 2. Tworzenie baz danych i administrowanie bazami. Podręcznik do nauki zawodu technik informatyk

Kwalifikacja E.14. Część 3. Tworzenie aplikacji internetowych. Podręcznik do nauki zawodu technik informatyk

Podręczniki oraz inne pomoce naukowe należące do tej serii zostały opracowane z myślą o wykształceniu kompetentnych techników, którzy bez trudu poradzą sobie z wyzwaniami w świecie współczesnej informatyki. Według nowych przepisów, aby otrzymać dyplom w zawodzie technik informatyk, należy zdać szereg egzaminów potwierdzających kolejne kwalifikacje w zawodzie. Książka ta powstała po to, by ułatwić to zadanie zarówno uczącym się, jak i pedagogom. Wiedza w niej zawarta pomoże zdać egzamin i zyskać wiedzę praktyczną, przydatną w przyszłej pracy.

helion.pl
księgarnia internetowa
Nr katalogowy: 8759
Księgarnia internetowa: http://helion.pl
Zamówienia telefoniczne: 0 801 339900
0 601 339900

Helion
Sprawdź najnowsze promocje:
http://helion.pl/promocje
Książki najchętniej czytane:
http://helion.pl/bestsellery
Zamów informacje o nowościach:
http://helion.pl/nowosci

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
http://helion.pl

Informatyka w najlepszym wydaniu

sięgnij po WIĘCEJ
KOD KORZYŚCI

ISBN 978-83-246-5102-3
9 788324 651023