helion.pl
KSIĘGARNIA INTERNETOWA · WYDAWNICTWO

IDŹ DO
PRZYKŁADOWY ROZDZIAŁ
SPIS TREŚCI

KATALOG KSIĄŻEK
KATALOG ONLINE
ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK
DODAJ DO KOSZYKA

CENNIK I INFORMACJE
ZAMÓW INFORMACJE O NOWOŚCIACH
ZAMÓW CENNIK

CZYTELNIA
FRAGMENTY KSIĄŻEK ONLINE

HTML, XHTML i CSS. Nowoczesne tworzenie stron WWW



Autor: David Schultz, Craig Cook
Tłumaczenie: Rafał Jońca
ISBN: 978-83-246-1498-1
Tytuł oryginału: Beginning HTML with CSS and XHTML: Modern Guide and Reference
Format: 158x235, stron: 384

Kompleksowe omówienie podstawowego budulca witryn WWW, czyli języka HTML

- Jak zbudowany jest dokument HTML?
- W jaki sposób umieszczać na stronach WWW elementy graficzne?
- Jak stosować kaskadowe arkusze stylów?

W ostatnich latach zasady tworzenia witryn WWW zmieniły się znacznie. Dynamiczny rozwój technologii i języków programowania oraz wzrost wydajności serwerów sprawiły, że strony internetowe stały się teraz naprawdę funkcjonalne i atrakcyjne. Języki, takie jak PHP i ASP, oraz technologia AJAX umożliwiły budowanie dynamicznych witryn WWW pobierających dane z baz i działających jak typowe aplikacje biurowe. Pomimo tak ogromnej rewolucji, u podstaw wszystkich serwisów WWW nadal leży język HTML. Jego znajomość jest niezbędna do zbudowania każdej witryny WWW – od najprostszej do najbardziej złożonej. Dopiero po opanowaniu języka HTML i ściśle związanej z nim technologii CSS można wykonać następny krok.

W książce „HTML, XHTML i CSS. Nowoczesne tworzenie stron WWW" znajdziesz omówienie wszystkich elementów najnowszej specyfikacji języka HTML. Czytając ją, poznasz strukturę dokumentu HTML, znaczniki oraz ich atrybuty. Dowiesz się, jak korzystać z kaskadowych arkuszy stylów, tworzyć formularze i umieszczać w dokumentach elementy graficzne, a także multimedialne. Przeczytasz także o języku JavaScript oraz prześledzisz rzeczywisty projekt witryny internetowej.

- Historia języka HTML
- Idea rozdzielenia treści od prezentacji
- Struktura dokumentów HTML i XHTML
- Konstrukcja stylów CSS
- Definiowanie metadanych
- Znaczniki zawartości dokumentu
- Obrazy i formularze na stronach WWW
- Tabele
- Formatowanie elementów HTML za pomocą arkuszy stylów
- Podstawy JavaScript

Zrób pierwszy krok na drodze do projektowania atrakcyjnych stron WWW – poznaj język HTML

Wydawnictwo Helion
ul. Kościuszki 1c
44-100 Gliwice
tel. 032 230 98 63
e-mail: helion@helion.pl

Spis treści

Podziękowania ... 9
O autorach ... 11
Słowo wstępne ... 13
Wprowadzenie ... 15

Rozdział 1. Zaczynamy ... 19
Wprowadzenie do internetu i witryn WWW ... 19
Czym jest HTML? ... 20
Ewolucja języka HTML ... 21
Jeden język, wiele wersji ... 21
Jedna wersja, trzy odmiany ... 22
Walidacja dokumentów ... 23
Oddzielanie zawartości od prezentacji ... 24
Praca z XHTML i CSS ... 25
Edytor tekstu ... 25
Przeglądarki internetowe ... 26
Umieszczanie witryny na serwerze ... 26
Wprowadzenie do adresów URL ... 27
Składniki adresu URL ... 27
Adresy pełne i relatywne ... 29
Podsumowanie ... 30

Rozdział 2. Podstawy dokumentów HTML i arkuszy stylów ... 31
Elementy składowe języka HTML — znaczniki i atrybuty ... 31
Elementy blokowe i wewnętrzne ... 33
Zagnieżdżanie elementów ... 34
Znaki niedrukowane ... 34
Atrybuty standardowe ... 35
Komentarze ... 37
Dokument XHTML ... 37
Typ dokumentu ... 38
Element html ... 40
I pozostała część… ... 41
Drzewo dokumentu ... 41
Podstawy arkuszy stylów ... 42
Anatomia reguły CSS ... 42
Selektory CSS ... 43
Szczegółowość i kaskadowość arkuszy stylów ... 47
Dołączanie arkuszy stylów do dokumentów ... 49

Kolejność stosowania kaskady ... 51
Dyrektywa !important ... 52
Formatowanie kodu CSS ... 53
Komentarze CSS ... 54
Podsumowanie ... 55

Rozdział 3. Metadane dokumentu ... 57
Element head ... 57
<head> ... 57
Elementy wspierające ... 58
<base> ... 59
<link> ... 60
<meta> ... 61
<script> ... 62
<style> ... 63
<title> ... 64
Podsumowanie ... 66

Rozdział 4. Dodawanie zawartości ... 67
Zawartość i struktura ... 67
Element body ... 68
<body> ... 68
Logiczne porcje ... 70
<p> ... 70
Nagłówki: h1, h2, h3, h4, h5 i h6 ... 72
<blockquote> ... 73
<address> ... 75
<pre> ... 76
Listy ... 77
 ... 77
 ... 79
 ... 80
Lista definicji ... 82
<dl> ... 82
<dt> ... 83
<dd> ... 83
Elementy dotyczące fraz ... 85
 ... 85
 ... 86
<cite> ... 87
<q> ... 88
<dfn> ... 89
Skróty: <abbr> i <acronym> ... 90
Modyfikacje dokumentu: <ins> i <del> ... 91
<bdo> ... 92
Elementy związane z programowaniem: <code>, <kbd>, <samp> i <var> ... 93

 ... 95
<hr> ... 96
Elementy do wielu zastosowań ... 97
 ... 98
 ... 99
Osadzanie zewnętrznej treści ... 100
<object> ... 100
<param> ... 101

Elementy prezentacyjne ... 103
<i> i <b> ... 103
<big> i <small> ... 103
<tt> ... 103
<sup> i <sub> ... 103
Znaki specjalne ... 104
Nadawanie zawartości stylu za pomocą CSS ... 106
Deklaracja podstawowych stylów czcionek ... 106
Style list ... 111
Podsumowanie ... 113

Rozdział 5. Obrazy ... 115
Jak działają cyfrowe obrazy? ... 116
Formaty obrazów przyjazne stronom WWW ... 117
Umieszczanie obrazów na stronach WWW ... 120
<img> ... 120
Obrazy w kontekście innych elementów ... 127
Otaczanie obrazu tekstem ... 128
Obrazy tła ... 130
Pozycjonowanie obrazu tła ... 133
Podsumowanie ... 135

Rozdział 6. Łącza i odnośniki ... 137
Znacznik łącza ... 137
<a> ... 137
Użycie elementu a ... 139
Łącza do innych dokumentów ... 139
Łącza do dokumentów w formacie innym niż XHTML ... 141
Łącza z adresami e-mail ... 143
Obraz jako łącze ... 144
Wykorzystanie identyfikatorów ... 145
Użycie CSS dla elementu łącza ... 146
Mapy obrazów ... 149
<map> ... 150
<area> ... 151
Podsumowanie ... 154

Rozdział 7. Tabele ... 155
Podstawy tworzenia tabel ... 155
<table> ... 156
<tr> ... 157
<td> ... 159
Przykład zastosowania tabeli ... 160
<caption> ... 162
<th> ... 164
Zaawansowane wykorzystanie tabel ... 166
<tbody> ... 168
<thead> ... 169
<tfoot> ... 170
<colgroup> ... 173
<col> ... 174

Wykorzystanie CSS do nadawania stylu tabelom ... 176
Dodawanie obramowania ... 176
Wyrównywanie tekstu w tabeli ... 178
Dodawanie odstępów do komórek ... 179
Dodawanie tła do tabeli ... 180
Podsumowanie ... 182

Rozdział 8. Formularze ... 183
Jak działają formularze? ... 183
Elementy składowe formularza ... 184
<form> ... 185
<input> ... 187
<button> ... 196
<select> ... 198
<option> ... 200
<optgroup> ... 201
<textarea> ... 203
Nadawanie formularzom struktury ... 205
<fieldset> ... 205
<legend> ... 207
<label> ... 209
Nadawanie formularzom stylów za pomocą CSS ... 211
Usuwanie obramowania z zestawu pól ... 213
Wyrównywanie etykiet ... 214
Zmiana czcionki kontrolek formularza ... 215
Podsumowanie ... 216

Rozdział 9. CSS, czyli arkusze stylów ... 219
Wykorzystywanie zewnętrznych arkuszy stylów ... 219
Jednostki miary ... 221
Układ graficzny ... 222
Pojemniki ... 222
Rozmiar i rozmieszczenie pojemnika ... 225
Ustalanie położenia pojemnika ... 227
Tło ... 235
Nadawanie stylu tekstowi ... 236
Rodzaje mediów ... 238
Zgodność ... 239
Podsumowanie ... 240

Rozdział 10. Podstawy wykonywania skryptów po stronie klienta ... 241
Czym są skrypty? ... 241
Umieszczanie kodu JavaScript ... 242
Język JavaScript ... 243
Reguły składni języka ... 244
Operatory i wyrażenia ... 246
Instrukcje ... 248
Pętle ... 254
Funkcje ... 258
Tablice ... 259
Tematy zaawansowane ... 259
Obsługa zdarzeń ... 260
DOM ... 262
Walidacja formularzy ... 263
Podsumowanie ... 265

Rozdział 11. Tworzymy witrynę od podstaw 267
Wprowadzenie do analizy przypadku — Spaghetti & Cruft 268
Proces projektowania 268
Krok 1. — zdefiniuj cele 269
Krok 2. — wymyślanie architektury 269
Krok 3. — kreowanie szablonu 270
Krok 4. — wykonanie układu graficznego strony 271
Krok 5. — łączenie wszystkich części 273
Krok 6. — testowanie 273
Tworzenie witryny Spaghetti & Cruft 274
Tworzenie dokumentu 274
Kod nagłówka strony 275
Wykonanie głównego fragmentu strony 276
Tworzenie nawigacji 277
Tworzenie sloganu i stopki 278
Kompletny szablon 278
Arkusz stylów dla Spaghetti & Cruft 280
Nadanie stylu ciału strony 281
Styl nagłówka 287
Układ strony 292
Nadawanie stylów nawigacji 294
Styl stopki 296
Tworzymy rzeczywiste strony 297
Strona z menu 297
Strona recenzji 301
Strona kontaktowa 303
Podsumowanie 307

Dodatek A Lista znaczników języka XHTML 1.0 Strict 309

Dodatek B Nazwy kolorów i ich wartości 345

Dodatek C Znaki specjalne 359

Dodatek D Poziom obsługi CSS w różnych przeglądarkach 363

Skorowidz 371

Rozdział 5.
Obrazy

Cały rozdział 4. został poświęcony dodawaniu treści tekstowych do stron WWW. Teraz przejdźmy do bardziej **multimedialnych** aspektów internetu, czyli obrazków. Obrazy stanowią nieodłączną część wielu witryn internetowych. Czynią je łatwiejszymi do zapamiętania i stymulują wizualnie. Elementy graficzne projektu strony zawierają najczęściej logo witryny i inne motywy pozwalające odróżnić daną witrynę od setek innych.

Obrazy pełnią nie tylko funkcję dekoracyjną, ale również komunikacyjną — są zawartością, która nierzadko lepiej niż słowa oddaje sens wypowiedzi. Zdjęcia, ilustracje, logo, ikony, mapy, wykresy i grafy — to sposoby dzielenia się pomysłami, które trudno byłoby przekazać w tekście. Z drugiej strony warto pamiętać, że nie każdy, kto odwiedzi witrynę, będzie w stanie zobaczyć obrazy. Na barkach autora witryny spoczywa odpowiedzialność za zapewnienie alternatywnych metod dostępu do zawartości, np. tekstu opisującego zawartość lub znaczenie obrazka.

Obrazy, które stanowią część zawartości dokumentu XHTML, można określać za pomocą elementu `img`. Zostaną one wyświetlone przez przeglądarkę tuż obok tekstu. Do osadzania obrazów można również użyć elementu `object`, opisywanego w rozdziale 4. Niestety obsługa tego sposobu wstawiania obrazów nie jest jeszcze powszechna, więc warto pozostać przy sprawdzonym elemencie `img`. Pamiętaj o tym, by w treści dokumentu umieszczać jedynie obrazy informacyjne — obrazy dekoracyjne określaj w arkuszu CSS dołączonym do strony, by odseparować zawartość od aspektów prezentacyjnych.

W tym rozdziale zajmiemy się sposobami umieszczania obrazów na stronach WWW. Najpierw opowiemy o podstawach działania i obsługi cyfrowych obrazów, wyjaśnimy działanie elementu wewnętrznego `img`, zapewniającego osadzanie obrazów w zawartości, a na końcu przejdziemy do opisów wykorzystania arkuszy CSS do wczytywania obrazów poprawiających wygląd strony. Obrazy tła definiowane w CSS poprawiają wygląd strony, ale nie wpływają na jego strukturę semantyczną.

Jak działają cyfrowe obrazy?

Wszystko, co znajduje się w komputerze, przechowywane jest w postaci cyfrowej — obrazy nie są tu wyjątkiem. Oznacza to, że obraz to tak naprawdę seria zer i jedynek umieszczona w pliku. Komputer czyta tablicę cyfr (każda cyfra to **bit**) i przekształca ją na sygnał przesyłany do drugiego urządzenia, które skonwertuje go w postać małych kropek kolorowego światła widzianego przez nasze oko — czerwonych, niebieskich, szarych itd. Plik zawiera dodatkowo instrukcje, w jaki sposób te kropki światła (nazywane **pikselami**) powinny zostać ułożone w mozaikę, z której powstanie wynikowy obraz. Poszczególne piksele tworzące obraz łatwo zauważyć, jeśli dokładnie przyjrzeć się ekranowi komputera lub telewizora. Ekstremalne powiększenie prezentuje rysunek 5.1.



Rysunek 5.1. *Powiększenie cyfrowego obrazu uwidacznia piksele, z których się składa*

Ponieważ obrazy składają się z map bitów, nazywamy je często **bitmapami** lub **obrazami rastrowymi**. To właśnie obrazy bitmapowe stanowią podstawę wyświetlania obrazów na stronach WWW. Zapamiętanie koloru i położenia każdego piksela zabiera dużo miejsca, szczególnie jeśli jeden obraz składa się z setek tysięcy pikseli w milionach możliwych kolorów (24-bitowy kolor oznacza ponad 16 milionów różnych kolorów).

Grafika wektorowa

Poza bitmapami istnieje jeszcze jeden sposób przechowywania cyfrowych obrazów — jako zbiór instrukcji matematycznych, które komputer powinien wykonać, aby narysować odpowiednie kształty na ekranie lub papierze. Tego rodzaju obrazy nazywa się **wektorowymi**. Mogą one być wyświetlane w dowolnej rozdzielczości bez pogorszenia jakości i modyfikacji wyglądu. Niestety, obsługa grafiki wektorowej wymaga specjalistycznego oprogramowania, które nie jest dołączane do większości przeglądarek internetowych, więc niemal wszystkie obrazy używane w internecie są obrazami rastrowymi.

Jakiś czas temu pojawiła się inicjatywa wprowadzenia grafiki wektorowej w formacie dostosowanym do użycia na stronach WWW. Język **SVG** (skrót od ang. *Scalable Vector Graphics*) wzorowany na XML nie jest jak dotąd powszechnie obsługiwany przez przeglądarki, więc jego zastosowanie jest ograniczone. Więcej informacji na temat obsługi tego formatu znajduje się na witrynie *http://www.w3.org/Graphics/SVG/*.

Obrazy używane w internecie są najczęściej **skompresowane** w celu zmniejszenia ich rozmiaru, by pobieranie strony WWW mogło się sprawnie odbyć nawet na powolnych połączeniach internetowych. Redukując liczbę kolorów lub liczbę pikseli, można łatwo zmniejszyć ogólny rozmiar obrazu. Jeśli kiedykolwiek pobierałeś ogromny obraz z internetu, doskonale zdajesz sobie sprawę, że bywa to denerwujące. Głównym celem kompresji jest maksymalne zmniejszenie rozmiaru pliku przy jednoczesnym zachowaniu możliwie dobrej jakości.

Formaty obrazów przyjazne stronom WWW

Obrazy kierowane do wykorzystania na stronach WWW można kompresować w trzech formatach: JPEG, GIF i PNG. Każdy z formatów stosuje inny mechanizm kompresji, więc ma różne zalety i wady. Większość przeglądarek graficznych ma wbudowane oprogramowanie poprawnie interpretujące i renderujące tego rodzaju obrazy. Przeglądarki mogą nie obsługiwać innych formatów, więc warto trzymać się wspomnianych trzech. Prawie każdy program do edycji grafiki rastrowej umożliwia zapis obrazu we wszystkich wymienionych formatach.

JPEG

JPEG to skrót od *Joint Photographic Experts Group* — organizacji, która wymyśliła wspomniany format. Algorytm kompresujący redukuje rozmiar wynikowego pliku przez próbkowanie średnich wartości koloru pikseli i redukcję nadmiarowych pikseli. Po dekompresji obrazu usunięte piksele zostają odtworzone na podstawie zapamiętanych próbek.

Ponieważ format ten powoduje utratę części informacji, nazywamy go formatem **stratnym**. Obraz JPEG po dekompresji nigdy nie będzie tej samej jakości, co oryginał. JPEG obsługuje różne poziomy **strat jakości** — im niższy poziom, tym mniej pikseli pozostaje, co zmniejsza rozmiar pliku, ale i pogarsza jakość. Wysoce skompresowane obrazy JPEG wydają się rozmyte i zawierają tak zwane **artefakty**, spowodowane algorytmem kompresji. Rysunek 5.2 przedstawia trzy wersje zdjęcia Jolene po zapisaniu z różnymi poziomami kompresji (obraz został powiększony dwukrotnie, by poprawić widoczność artefaktów). Większa kompresja daje mniejszy rozmiar pliku, ale gorszą jakość.

Co więcej, każda edycja i zapis obrazu JPEG oznacza ponowną kompresję czegoś, co zostało już wcześniej skompresowane. Każda kolejna kompresja coraz bardziej pogarsza jakość obrazu, podobnie jak wielokrotne kserowanie kolejnych kopii. Zawsze warto pozostawiać sobie oryginalną wersję zdjęcia i kompresować ją do formatu stratnego tylko wtedy, gdy zachodzi potrzeba jego aktualizacji na witrynie.

Format JPEG zaoszczędza miejsca przez redukcję pikseli, ale również zapamiętuje wiele informacji o kolorze w stosunkowo niewielkim pliku, co czyni go idealnym formatem dla fotografii i innych obrazów zawierających dużo kolorów i elementów, które gładko zmieniają odcienie. Pliki JPEG stosują rozszerzenie *.jpg* lub *.jpeg*. Krótsza wersja stała się najbardziej popularna, bo niektóre starsze systemy operacyjne nie obsługiwały rozszerzeń dłuższych niż trzyznakowe.



Niska kompresja (56 kB) *Średnia kompresja (12 kB)* *Wysoka kompresja (5 kB)*

Rysunek 5.2. *Ten sam obraz JPEG z różnymi poziomami kompresji. Wersja po prawej stronie ma najmniejszy rozmiar pliku, ale najgorszą jakość*

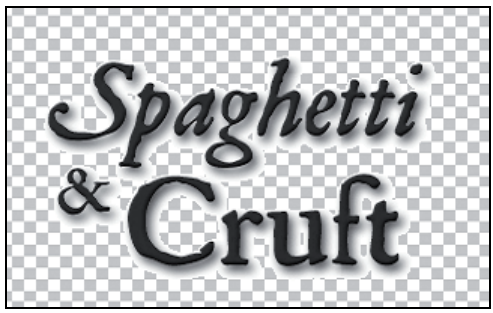
GIF

GIF to skrót od *Graphic Interchange Format*. W tym systemie kompresja odbywa się przez zmniejszenie liczby kolorów używanych w obrazie, przy zachowaniu informacji o każdym pikselu. Ponieważ format GIF jest **bezstratny**, jest doskonały do pracy nad logo, ikonami lub innymi grafikami, które mają ostre krawędzie. Obraz GIF może zawierać maksymalnie 256 kolorów, a gdy jest ich mniej, jeszcze bardziej poprawia to stopień kompresji. Grafy, mapy, grafika z dużą liczbą linii i inne obrazy z dużymi powierzchniami w jednolitym kolorze (lub tylko o kilku barwach) są przykładami obrazów doskonale nadających się do skompresowania algorytmem GIF.

Obrazy GIF mogą zawierać obszary, które są przezroczyste, dzięki czemu możliwe jest wyświetlenie tego, co znajduje się za nimi. Każdy z pikseli albo jest w pełni przezroczysty, albo całkowicie nieprzezroczysty, więc pojawi się mocno zarysowana granica na styku dwóch obszarów. Wiele programów do edycji grafiki umożliwia wybranie **koloru matującego** dla przezroczystych obrazów GIF, co minimalizuje efekt ostrości krawędzi. Rysunek 5.3 przedstawia obraz GIF na tle szachownicy. Zauważ białe obramowanie (kolor matujący), które będzie doskonale współgrało z docelowym, białym kolorem tła witryny.

Kolejną ciekawą funkcją obrazów GIF jest obsługa krótkich animacji. Obraz może zawierać ciąg klatek, wyświetlanych w określonej sekwencji, co umożliwia tworzenie ciekawych, a czasem również bardzo denerwujących, efektów. Oczywiście każda klatka zwiększa rozmiar wynikowego pliku GIF.

Pliki GIF stosują rozszerzenie *.gif*.

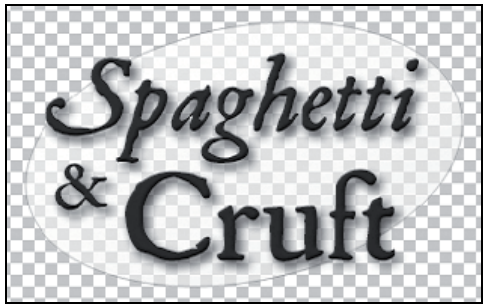


Rysunek 5.3. *Przezroczysty obraz GIF z białym kolorem matującym. Tło w postaci szachownicy przedstawiono jedynie w celach demonstracyjnych*

PNG

PNG (skrót od ang. *Portable Network Graphic*) to format wymyślony jako bezpłatny następca opatentowanego formatu GIF, który dodatkowo rozwija oryginał w kilku płaszczyznach. Podobnie jak GIF, PNG może zawierać maksymalnie 256 kolorów (tak zwany **kolor 8-bitowy**, ponieważ 256 różnych wartości można zmieścić w 8 bitach) i obsługuje przezroczystość. Dzięki innemu algorytmowi kompresji, w niektórych sytuacjach 8-bitowy PNG bywa mniejszy niż jego odpowiednik w formacie GIF.

Co ważne, PNG występuje również w wariancie **24-bitowym**, który obsługuje miliony kolorów. Jest w tym względzie podobny do formatu JPEG, ale charakteryzuje się większymi rozmiarami plików, bo nie stosuje stratnego algorytmu kompresji. Największą zaletą 24-bitowych obrazów PNG jest obsługa **kanału alfa** (kanału przezroczystości), który stanowi jakby osobną warstwę, określającą poziomy przezroczystości poszczególnych pikseli. Podczas gdy przezroczystość w obrazach GIF lub 8-bitowych PNG jest dwustanowa, piksele w 24-bitowym PNG mogą być przezroczyste tylko częściowo, co oznacza zmieszanie koloru obrazu i koloru tła w proporcjach zdefiniowanych przez warstwę alfa. Działanie kanału przezroczystości przedstawia rysunek 5.4. Szachownicę widać przez część logo z elipsą, co zapewnia obrazowi dobre dopasowanie do dowolnego tła.



Rysunek 5.4. *24-bitowy PNG z kanałem przezroczystości. Szachownicę widać przez częściowo przezroczystą elipsę*

Niestety, wiele starszych przeglądarek, włączając w to przeglądarki Internet Explorer przed wersją 7, nie obsługuje kanału przezroczystości w 24-bitowych obrazach PNG. Dopóki starsze przeglądarki nadal są popularne, należy bardzo uważać ze stosowaniem obrazów PNG stosujących kanał przezroczystości.

Bity i bajty

Wszystkie dane w świecie komputerów składają się z zer i jedynek, czyli cyfr, stąd nazwa **technologia cyfrowa**. Zera i jedynki reprezentują dwa stany przełącznika — 1 oznacza włączony, a 0 wyłączony. To baza kodu binarnego, stanowiącego podstawę działania wszystkich komputerów. Pojedyncza cyfra nosi nazwę **bit**, a sekwencja 8 bitów to **bajt**. Przy dużych zbiorach bajtów stosujemy wielokrotności wartości 1024; 1024 bajty to jeden **kilobajt**, 1024 kilobajty to jeden **megabajt**, a 1024 megabajty to jeden **gigabajt**. W ten sposób mierzymy ilość cyfrowych danych.

Wartość koloru każdego piksela obrazu bitmapowego to ciąg zer i jedynek. Im więcej zer i jedynek przypada na jeden piksel, tym więcej różnych kolorów można opisać. Najprostszy obraz może stosować tylko pojedyncze zero i jedynkę do opisu kolorów — oznacza to obraz monochromatyczny, zawierający tylko czerń i biel. Ponieważ każdy bit to dwie wartości, liczba dostępnych kolorów jest zawsze wartością wynikającą z podniesienia liczby 2 do potęgi równej liczbie bitów. Dla dwóch bitów na kolor mamy dostęp do czterech kolorów (2^2) z powodu czterech permutacji (00, 01, 10, 11). Dla czterech bitów kolorów może być 16 (2^4). Dla 8 bitów — 256 (2^8).

Obrazy GIF stosują maksymalnie 8 bitów na piksel, więc mogą stosować najwyżej 256 różnych kolorów. JPEG używa 24 bitów na piksel, czyli obsługuje 16 777 216 różnych kolorów, co w wielu przypadkach stanowi granicę ludzkich możliwości rozróżniania kolorów. Format PNG obsługuje wersję 8-bitową oraz 24-bitową.

W 8-bitowych obrazach GIF i PNG tylko jedna wartość zostaje wskazana jako wartość przezroczystości, więc dany piksel jest widoczny lub niewidoczny. W 24-bitowym formacie PNG każdemu pikselowi przypisano osobne 8 bitów określających jego przezroczystość, więc łącznie mamy dostęp do 256 poziomów przezroczystości: od 0 oznaczającego pełną przezroczystość do 255 oznaczającego pełne krycie.

Umieszczanie obrazów na stronach WWW

Zawartość tekstowa stanowi część dokumentu XHTML. Otaczają ją znaczniki, informujące o znaczeniu poszczególnych słów lub zdań. Obrazy są zewnętrznymi plikami i nie stanowią części dokumentu. Odwołania do obrazów w dokumencie XHTML oznacza się za pomocą elementu `img` (lub `object`, choć nie jest on tak dobrze obsługiwany). Renderowanie strony zawierającej obrazy jest dwuetapowe: najpierw przeglądarka pobiera dokument XHTML, a dopiero później wszystkie powiązane z nim obrazy. Po uzyskaniu obrazka z serwera WWW przeglądarka wyświetla go na ekranie.

Element `img` należy traktować jako element zastępowany, ponieważ on sam nigdy nie jest wyświetlany. Co więcej, element ten jest **elementem pustym**, więc należy go zamykać za pomocą znaków `/>`. Element wymaga wskazania źródła obrazu do wyświetlenia przy użyciu atrybutu `src`. Podany adres URL może być względny lub pełny.

Element `img` wymaga również wskazania atrybutu `alt`, zawierającego alternatywną treść, czyli tekst opisujący zawartość elementu. Alternatywny tekst pojawi się na stronie, jeśli obrazek nie jest dostępny lub przeglądarka nie jest w stanie wyświetlać obrazów. Co ważne, tekst ten pomaga zrozumieć treść strony osobom niedowidzącym. Listing 5.1 przedstawia element `img` jedynie z atrybutami `src` i `alt`, czyli całkowitym minimum pozwalającym uznać go za poprawny.

Listing 5.1. *Najprostsza postać elementu img*

```
<img src="/images/pizza.jpg" alt="Pizza z sosem i oliwkami" />
```

Atrybuty wymagane

- `src` — określa adres URL pliku z obrazem.
- `alt` — określa alternatywny tekst wyświetlany zamiast obrazu.

Atrybuty opcjonalne

- `width` — szerokość obrazu w pikselach.
- `height` — wysokość obrazu w pikselach.
- `ismap` — informuje, że obraz jest powiązany z mapą po stronie serwera.
- `usemap` — identyfikuje mapę po stronie użytkownika.
- `longdesc` — określa adres URL zawierający rozbudowany opis obrazu.

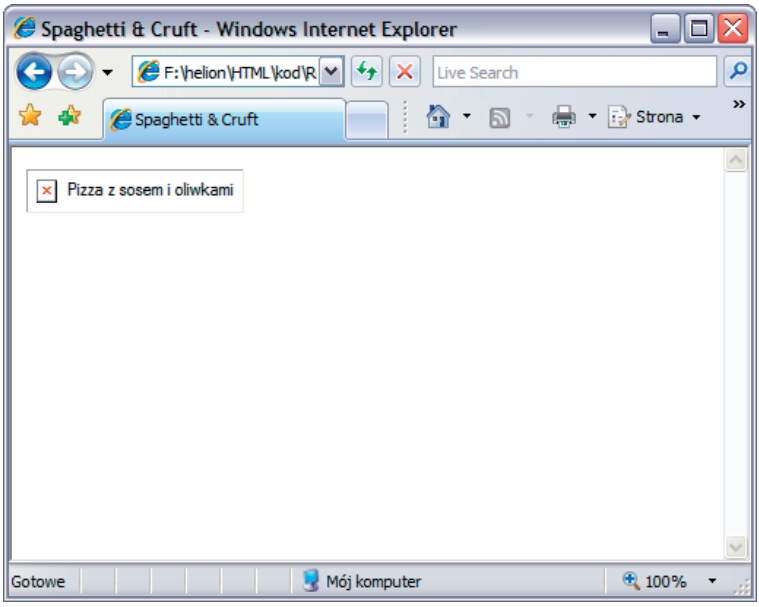
Atrybuty standardowe

- `class`
- `dir`
- `id`
- `lang`
- `style`
- `title`
- `xml:lang`

Atrybut alt

Wszystkie wystąpienia elementu `img` muszą zawierać atrybut `alt`, zapewniający alternatywną treść, jeśli z jakichś powodów obrazek nie jest widoczny, np. użytkownik niedowidzi, używane oprogramowanie nie potrafi wyświetlać obrazów lub obraz nie jest dostępny w podanej lokalizacji. Użycie tekstu opisującego zawartość obrazka pozwala przekazać choć część informacji, jeśli obrazek nie jest dostępny. Atrybut `alt` może zawierać do 1024 znaków (włączając w to spacje), ale warto być możliwie precyzyjnym. Jeśli obraz jest bardzo złożony i wymaga dłuższych wyjaśnień, można skorzystać z atrybutu `longdesc`.

Atrybut `alt` powinien stanowić zastępnik obrazu, gdy ten nie jest dostępny. Jeśli obraz przedstawia zdjęcie kota śpiącego w koszyku, prawidłowy atrybut `alt` powinien zawierać tekst „mój kot w koszyku" lub „mój szary kot śpi w plecionym koszyku" lub nawet „mój szary, paskowany kot śpi w kuchni, w plecionym koszyku". Opis pozwala użytkownikowi wyobrazić sobie zawartość zdjęcia. Rysunek 5.5 pokazuje, w jaki sposób przeglądarka Internet Explorer reaguje, jeśli obraz nie znajduje się na serwerze. Zamiast obrazu pojawia się treść `alt`, która informuje o jego zawartości.

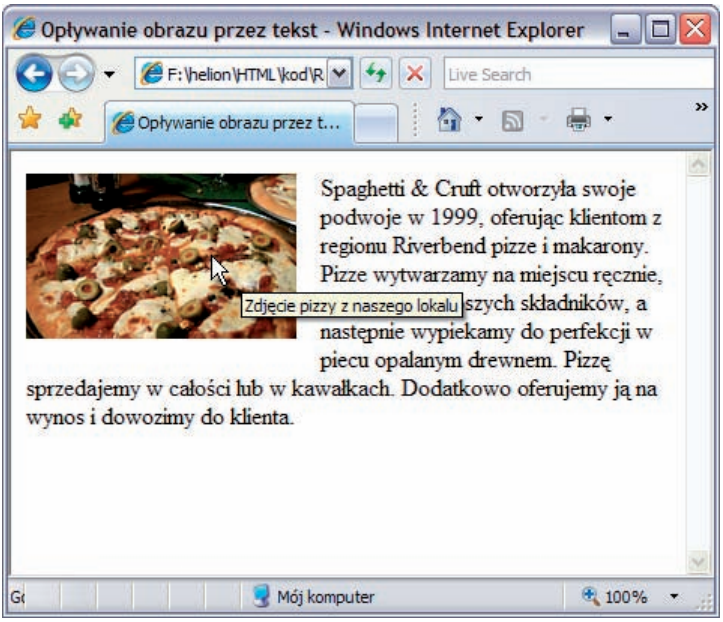


Rysunek 5.5. *Przeglądarka wyświetli alternatywną zawartość, jeśli obrazu nie uda się załadować*

Atrybut `alt` powinien być sensownym zastąpieniem obrazu, więc unikaj niewiele mówiących tekstów w stylu „firmowe logo". W atrybucie zawrzyj nazwę firmy, a nie informację o jakimś logo, użyj więc tekstu w rodzaju „Moja Firma, logo". Jeśli obrazek przedstawia tekst, w atrybucie `alt` powinna znaleźć się treść tego tekstu.

Atrybut `alt` nie powinien skupiać się na tym, czy brakujący obraz zawiera logo, zdjęcie, ilustrację, portret, wykres itp. Użycie tekstu „zdjęcie mojego kota" informuje co prawda, że na zdjęciu znajduje się kot, ale nie mówi nic o jego sytuacji i lokalizacji zdjęcia, co może być bardzo istotne. Staraj się przekazać temat przewodni zdjęcia, a nie tylko jego zawartość. Nie stosuj nazwy pliku z obrazem jako tekstu alternatywnego, bo *mojkot.jpg* niewiele mówi czytelnikowi.

Internet Explorer dla systemu Windows, najpopularniejsza na świecie przeglądarka internetowa, z niewyjaśnionych przyczyn wyświetla zawartość atrybutu `alt` jako **podpowiedź**, czyli dodatkowy tekst, pojawiający się po umieszczeniu kursora myszy nad obrazem (rysunek 5.6).



Rysunek 5.6. *Internet Explorer dla systemu Windows niepoprawnie wyświetla wartość atrybutu alt jako podpowiedź*

Z tego powodu wielu projektantów stron WWW przez lata błędnie użytkowało atrybut `alt`, żeby wyświetlać dodatkowe informacje o obrazie jako podpowiedź: „moje ulubione zdjęcie" lub „mój kot o imieniu Kiciuś". Teksty te nie opisują obrazu ani nie zastępują go, więc nie są odpowiednimi wartościami dla `alt`. Ponieważ tylko Internet Explorer dla Windows stosuje wyświetlanie zawartości `alt` jako podpowiedzi, użytkownicy stosujący inne przeglądarki nie zobaczą tego komunikatu.

Atrybut `title`, jeśli zostanie użyty, będzie wyświetlany jako podpowiedź w większości przeglądarek graficznych. Atrybut ten jest odpowiedni do umieszczania opisu, stanowiącego uzupełnienie treści obrazu. Jeśli zarówno `alt`, jak i `title` są dostępne dla obrazu (listing 5.2), przeglądarka Internet Explorer dla Windows zastosuje tekst zawarty w `title`.

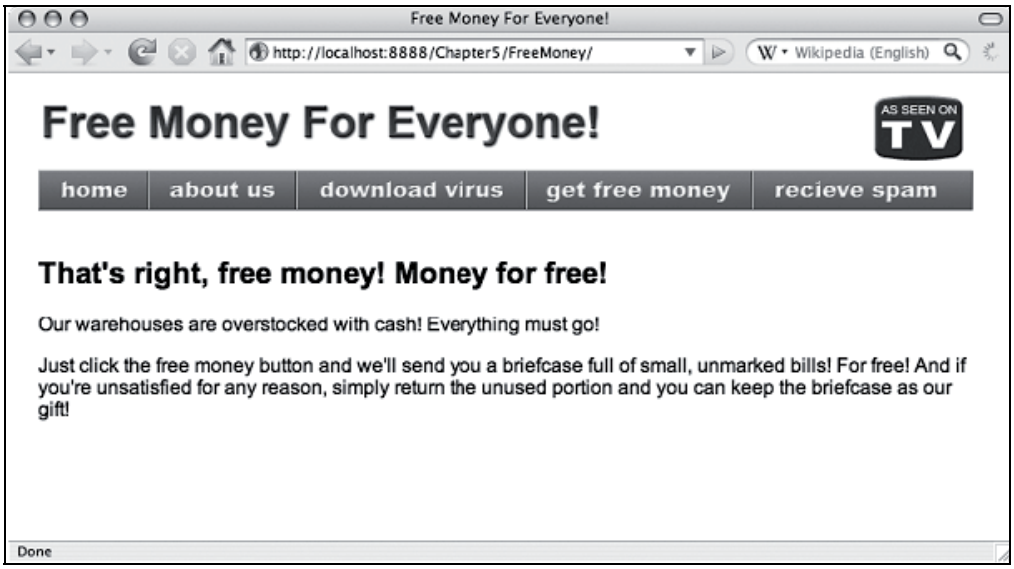
Listing 5.2. *Element img z opisowym atrybutem alt oraz i ogólnym atrybutem title*

```
<img src="/images/pizza.jpg" title="Nasza sławna Pizza Neapoli" alt="Cała pizza
➥z zielonymi oliwkami i roztopionym serem mozzarella" />
```

Co gorsza, niektórzy projektanci całkowicie unikają atrybutu `alt`, by tylko uniknąć wyświetlania podpowiedzi w przeglądarce Internet Explorer. Element `img` bez atrybutu `alt` nie tylko nie jest poprawny z punktu widzenia XHTML, ale również utrudnia życie osobom niedowidzącym. System czytający strony na głos lub przeglądarka tekstowa po prostu poinformują o istnieniu obrazu bez żadnych dodatkowych danych lub przeczytają zawartość atrybutu `src`. Pominięcie `alt` czyni z istotnego obrazu bezwartościowy znacznik.

Informacyjny tekst `alt` jest szczególnie ważny, jeśli obraz stanowi część łącza lub jest przyciskiem powodującym na przykład wysłanie formularza. Tego rodzaju obrazy pełnią określoną funkcję, nie są tylko treścią informacyjną. Jeśli obraz stanowi część łącza,

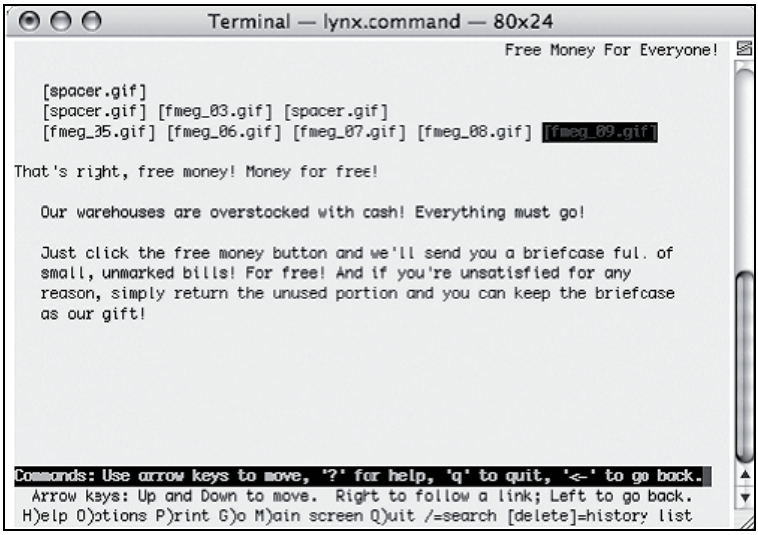
zastosowanie w nim poprawnego atrybutu `alt` jest niezwykle istotne. Rysunek 5.7 przedstawia stronę WWW z nawigacją stosującą obrazki. Odwiedzający z dobrym wzrokiem bez problemu skorzysta ze strony, nawet jeśli nie zawiera ona atrybutów `alt`.



Rysunek 5.7. *Witryna stosuje obrazki jako elementy wskazujące łącza do innych stron. Rysunek przedstawia wygląd strony w przeglądarce Firefox*

Rysunek 5.8 przedstawia ten sam tekst wyświetlony w przeglądarce Lynx — przeglądarce tekstowej, która wyświetla nazwę pliku obrazu, jeśli nie wskazano właściwego opisu w atrybucie `alt`. Bez atrybutów `alt` nawigacja w tej przeglądarce jest praktycznie bezużyteczna.

Rysunek 5.8. *Przeglądarka Lynx pokazuje obrazy bez atrybutów alt, stosując nazwy plików. Uniemożliwia to nawigację w witrynie, bo wymusza na użytkowniku zgadywanie, gdzie mogą prowadzić łącza*



Obrazy, które pełnią rolę dekoracyjną, czyli nie stanowią części rzeczywistej treści, nadal wymagają atrybutu `alt`, ale w takiej sytuacji należy użyć pustej wartości `alt` (`alt=""`), by wskazać, że są one nieistotne. Jako dodatkowy bonus Internet Explorer nie wyświetli podpowiedzi, gdy znajdzie pusty atrybut `alt`.



> Wiele osób niepoprawnie mówi o **znaczniku** `alt`, gdy tak naprawdę ma na myśli **atrybut** `alt`. To bardzo mylące, ponieważ `alt` nie jest i nigdy nie był znacznikiem.

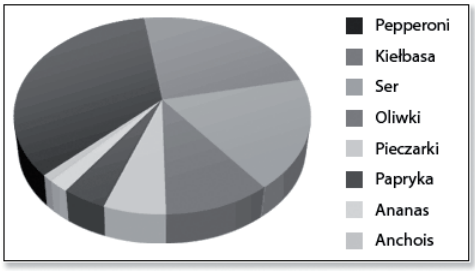
Atrybut longdesc

Opcjonalny atrybut `longdesc` zawiera adres URL wskazujący inną stronę WWW z pełnym wyjaśnieniem znaczenia obrazu, jeśli mógłby on nie zmieścić się w atrybucie `alt`. Jest to szczególnie przydatne dla grafów i wykresów, które mają złożoną reprezentację, ale można je również zaprezentować w postaci tabeli. Listing 5.3 przedstawia element `img` z atrybutem `longdesc`. Obraz przedstawia wykres kołowy, który stosunkowo trudno opisać w atrybucie `alt`.

Listing 5.3. *Element img z atrybutem longdesc*

```
<img src="/images/piechart.gif" alt="Wykres kołowy przedstawiający względną
➥ popularność różnych rodzajów dodatków do pizzy" longdesc="/dodatki.html" />
```

Rysunek 5.9 przedstawia wykres kołowy oraz te same informacje przedstawione w postaci tabeli. Dane tabelaryczne są adresowane do osób, które z różnych powodów nie są w stanie obejrzeć i zinterpretować wykresu. Więcej informacji na temat tabel pojawi się w rozdziale 7.



Różne dodatki do pizzy i odsetek klientów którzy zaznaczyli je jako ulubione

Dodatki	Procent
Pepperoni	35%
Kiełbasa	24%
Ser	18%
Oliwki	10%
Pieczarki	6%
Papryka	4%
Ananas	2%
Anchois	1%

Rysunek 5.9. *Dane w postaci wykresu i ich tabelaryczny odpowiednik*

Atrybuty width i height

Obraz zostanie wyświetlony w przeglądarce zgodnie ze swoim naturalnym rozmiarem. Ponieważ jednak przeglądarka pobiera obrazy **po** wczytaniu całego kodu HTML, nie zna rzeczywistych rozmiarów obrazów, dopóki ich nie pobierze. Po pobraniu każdego obrazu i poznaniu jego rozmiarów zmieni układ tekstu, co może być denerwujące,

jeśli użytkownik rozpoczął już czytanie tekstu. Stosując atrybuty `width` i `height` w elemencie `img`, informujemy przeglądarkę, ile miejsca powinna zarezerwować na obraz i że ma renderować tekst tak, jakby obraz został już wczytany.

Jeśli wartości atrybutów `width` i `height` nie są takie same jak naturalne wymiary obrazu, przeglądarka przeskaluje obraz, by dostosować go do atrybutów. Jeśli tylko to możliwe, unikaj skalowania obrazów w ten sposób. Skalowanie za pomocą przeglądarki najczęściej nie zapewnia najlepszej jakości. W przypadku powiększania pojawią się duże piksele. Zmniejszanie obrazu w ten sposób nie ma dużych efektów ubocznych w kwestii jakości, ale wymaga pobrania obrazu o pełnym rozmiarze, co również nie jest dobre. Najlepiej jeśli atrybuty pasują do naturalnych wymiarów obrazu. Skalowanie lepiej przeprowadzać w wyspecjalizowanym programie graficznym.


Uwaga

Do określania wymiarów obrazu, można również użyć właściwości `width` i `height` z CSS. Jeśli element ma określone atrybuty i dodatkowo nadany styl z wymiarami, styl CSS nadpisze wartości określone przez atrybuty.

Atrybuty usemap i ismap

Mapa obrazkowa to obraz, w którym niektóre obszary zostały oznaczone jako łącza. W ten sposób łączami są tylko fragmenty obrazu, a nie jego całość. Atrybut `usemap` określa element map, który ma być użyty do wyświetlenia mapy obrazkowej działającej **po stronie klienta**. Atrybut `ismap` wskazuje użycie mapy działającej **po stronie serwera** (tego rozwiązania należy unikać z powodu braku przydatności dla osób niedowidzących). Więcej informacji na temat działania map obrazkowych znajduje się w rozdziale 6., opisującym hiperłącza.

Wycofane atrybuty prezentacyjne

Starsze wersje HTML definiowały kilka atrybutów prezentacyjnych dla elementu `img`, ale zostały one wycofane na rzecz CSS. Żaden z tych elementów nie jest poprawny w XHTML, ale wymieniamy je, by pokazać, jakie są ich odpowiedniki w CSS:

- `align` — określa ułożenie obrazu względem otaczającego go tekstu. Dostępnymi wartościami są: `left`, `right`, `top`, `middle` oraz `bottom`. Wyrównywanie do lewej lub do prawej łatwo uzyskać za pomocą właściwości `float`. Trzy pozostałe wyrównania są dostępne za pomocą właściwości `vertical-align`.
- `border` — określa szerokość obramowania, w którym znajdzie się obraz, jeśli będzie stanowił część hiperłącza. Obecnie ten sam efekt udostępnia właściwość `border-width` z CSS.
- `hspace` — określa poziomy odstęp po lewej i po prawej stronie obrazu. Został zastąpiony przez właściwości `margin-left` i `margin-right`.
- `vspace` — określa pionowy odstęp na górze i na dole obrazu. Został zastąpiony przez właściwości `margin-top` i `margin-bottom`.


Uwaga

Niektóre starsze dokumenty zawierają dodatkowo atrybut `lowsrc`. Określa on mniejszą wersję obrazka, która była pobierana przed rozpoczęciem pobierania właściwej wersji. Jest to jednak bardzo przestarzały atrybut, stosowany jedynie przez starsze wersje przeglądarki Netscape Navigator. Obecnie nie należy go stosować.

Obrazy w kontekście innych elementów

Element `img` jest elementem wewnętrznym, więc może znajdować się w tym samym wierszu co tekst. W takiej sytuacji będzie spoczywał na dolnej krawędzi podstawy tekstu, nazywanej linią bazową. Listing 5.4 przedstawia element `img` umieszczony w akapicie z tekstem (elemencie `p`).

Listing 5.4. *Element img wraz z tekstem w jednym akapicie*

```
<p><img src="images/pizza.jpg" width="180" height="110" alt="Zdjęcie pizzy
➥z naszego lokalu" />
Spaghetti &#38; Cruft otworzyła swoje podwoje w 1999, oferując klientom z regionu
Riverbend pizze i makarony. Pizze wytwarzamy na miejscu ręcznie, używając najlepszych
składników, a następnie wypiekamy do perfekcji w piecu opalanym drewnem. Pizzę
sprzedajemy w całości lub w kawałkach. Dodatkowo oferujemy ją na wynos i dowozimy do
klienta.</p>
```

Rysunek 5.10 przedstawia wygląd akapitu w przeglądarce graficznej. Zauważ, że spód obrazu znajduje się w tym samym miejscu, co linia bazowa tekstu.


Spaghetti & Cruft otworzyła swoje podwoje w 1999, oferując klientom z regionu Riverbend pizze i makarony. Pizze wytwarzamy na miejscu ręcznie, używając najlepszych składników, a następnie wypiekamy do perfekcji w piecu opalanym drewnem. Pizzę sprzedajemy w całości lub w kawałkach. Dodatkowo oferujemy ją na wynos i dowozimy do klienta.

Rysunek 5.10. *Zauważ, że obraz został wyświetlony w jednej linii z tekstem (zdjęcie: Jeremy Keith)*

Jeśli obraz, tekst lub oba te elementy naraz znajdują się w elementach blokowych (`div`, `p` itp.), górę bierze standardowe zachowanie bloków, więc obraz i tekst pojawią się w osobnych wierszach. Listing 5.5 przedstawia tę samą zawartość, ale tym razem element `img` znalazł się poza akapitem we własnym elemencie `div`.

Listing 5.5. *Element img w elemencie blokowym div oraz tekst umieszczony w osobnym akapicie*

```
<div><img src="/images/pizza.jpg" width="180" height="110" alt=" Zdjęcie pizzy
➥z naszego lokalu" /></div>

<p>Pizze wytwarzamy na miejscu ręcznie, używając najlepszych
składników, a następnie wypiekamy do perfekcji w piecu opalanym drewnem.</p>
```

Rysunek 5.11 przedstawia wynik uzyskany w przeglądarce graficznej. Obraz (a dokładniej element `div`) i tekst znajdują się w osobnych wierszach.



Pizze wytwarzamy na miejscu ręcznie, używając najlepszych składników, a następnie wypiekamy do perfekcji w piecu opalanym drewnem.

Rysunek 5.11. *Elementy p i div są elementami blokowymi, więc pojawiają się we własnych wierszach*

Wybierając sposób umieszczenia obrazu (wewnątrz lub poza otaczającym go tekstem), pomyśl o jego znaczeniu. Czy stanowi on część tej samej myśli, która pojawia się w tekście? Jeśli tak, umieść go w akapicie. Jeśli obraz ukazuje sytuację luźno związaną z tekstem, warto umieścić go osobno.


Ostrzeżenie

Pamiętaj o tym, że element `body` może zawierać wyłącznie elementy blokowe jako swoje dzieci. Element `img` jest elementem wewnętrznym, więc nie można umieścić go bezpośrednio w elemencie `body`. Aby uzyskać poprawny kod XHTML, musisz umieścić go w elemencie blokowym.

Gdy dwa obrazy znajdą się obok siebie, zachowują się tak samo jak inne elementy wewnętrzne — znajdą się obok siebie w jednej linii i na tym samym poziomie. Obrazy, podobnie jak słowa, automatycznie przechodzą do następnego wiersza, jeśli się nie mieszczą.

Otaczanie obrazu tekstem

W niezliczonej liczbie czasopism, książek i gazet tekst otacza obraz umieszczony w kolumnie, w taki sposób jak strumień omijający przeszkodę. We wcześniejszych wersjach HTML należało zastosować atrybut `align` (wycofany już z użycia). Obecnie do uzyskania tego samego efektu służy właściwość `float` z CSS.

Właściwość przyjmuje trzy wartości: `left`, `right` lub `none`. Gdy element staje się elementem opływanym, przesuwa się możliwie daleko w jedną ze stron (prawą lub lewą) aż do napotkania granicy bloku lub innego opływanego elementu. Cały tekst lub inne elementy, które pojawią się po nim, będą go opływały. Domyślna wartość `none` służy do wycofywania właściwości `float` nałożonej na element przez inną regułę.

Listing 5.6 przedstawia kod HTML zawierający obraz i blok tekstu (oba elementy znajdują się w jednym akapicie). Element `img` wykorzystuje atrybut `class`, by ułatwić dostęp do niego z poziomu CSS.

Listing 5.6. *Obraz w jednym akapicie z tekstem*

```
<p><img src="images/pizza.jpg" width="180" height="110" class="figure" alt="Zdjęcie
↪pizzy z naszego lokalu" />
Spaghetti &#38; Cruft otworzyła swoje podwoje w 1999, oferując klientom z regionu
Riverbend pizze i makarony. Pizze wytwarzamy na miejscu ręcznie, używając najlepszych
składników, a następnie wypiekamy do perfekcji w piecu opalanym drewnem. Pizzę
sprzedajemy w całości lub w kawałkach. Dodatkowo oferujemy ją na wynos i dowozimy do
klienta.</p>
```

Obraz należy do klasy `figure`. Listing 5.7 przedstawia dla tej klasy regułę CSS, która wskazuje, że obraz powinien przesunąć się w lewo i być opływany z prawej strony.

Listing 5.7. *Reguła CSS dla klasy figure*

```
.figure {
  float: left;
}
```

Rysunek 5.12 przedstawia efekt połączonych obu fragmentów kodu. Obraz przesunął się na lewą stronę akapitu, a tekst go opływa.

Rysunek 5.12. *Obrazek przesunął się w lewo, więc tekst może go opływać*



Obraz znajduje się w tej samej linijce, co tekst, ale został dosunięty do lewej strony i dowiązany do góry wiersza, w którym się pojawił. Ponieważ ma większy rozmiar niż otaczający go tekst, przechodzi poniżej linii bazowej. W jednym z wcześniejszych przykładów obraz zwiększał wysokość wiersza tekstu i czynił go trudniejszym do odczytania. Aby odsunąć nieco tekst od obrazu, warto zastosować margines, rozszerzając wcześniejszą regułę CSS (listing 5.8).

Listing 5.8. *Dodanie marginesów do reguły klasy figure*

```
.figure {
  float: left;
  margin-right: 1em;
  margin-bottom: .5em;
}
```

W tym konkretnym przypadku chcemy jedynie ustawić prawy i dolny margines, ponieważ górna i lewa strona nie kolidują z żadnymi innymi elementami. Pozostawienie tych dwóch krawędzi z domyślnym marginesem powoduje dosunięcie obrazu do niewidocznej krawędzi akapitu. Zauważ, że po wprowadzeniu zmian (rysunek 5.13) obraz ma więcej miejsca. Tekst nadal go opływa, ale z uwzględnieniem marginesu.



Rysunek 5.13. *Zastosowanie marginesów dla obrazu lepiej oddziela go od tekstu*

Obrazy tła

Stosując właściwość `background-image` z CSS, bardzo łatwo dodać dekoracyjny obraz na stronie, przy jednoczesnym unikaniu zaśmiecania go elementami prezentacyjnymi. Tylko obrazy, które stanowią treść strony, powinny znaleźć się w kodzie XHTML. Niemalże dowolnemu elementowi CSS można nadać obraz tła. Tło będzie zajmowało dokładnie taką samą przestrzeń jak element. Obraz tła domyślnie ułoży się w **kafelki** w obu kierunkach, zaczynając od lewego górnego narożnika. Efekt nazywamy kafelkowaniem, bo przypomina kafelki ułożone w kuchni lub łazience.

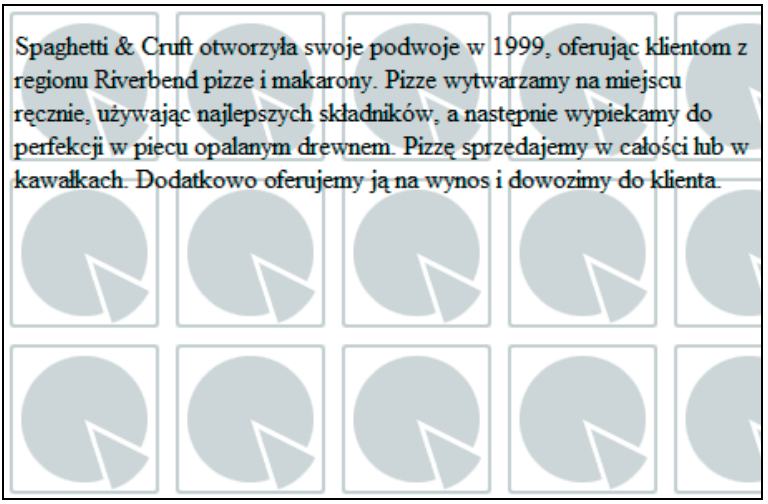
Listing 5.9 przedstawia regułę CSS, która spowoduje dodanie obrazu tła dla elementu `body`. Obraz określono za pomocą adresu URL, stosując słowo kluczowe `url` oraz nawiasy okrągłe.

Listing 5.9. *Obraz tła zastosowany dla elementu body*

```
body {
  background-image: url(/images/background.gif);
}
```

Obraz układa się w kafelki, by wypełnić całe okno przeglądarki po zrenderowaniu dokumentu (rysunek 5.14).

Domyślny tryb kafelkowania łatwo zmienić, stosując właściwość `background-repeat`, określając, że powtarzanie powinno następować tylko w pionie, tylko w poziomie lub wcale. Listing 5.10 rozszerza wcześniejszą regułę, deklarując, że obraz tła powinien powtarzać się tylko w osi X.

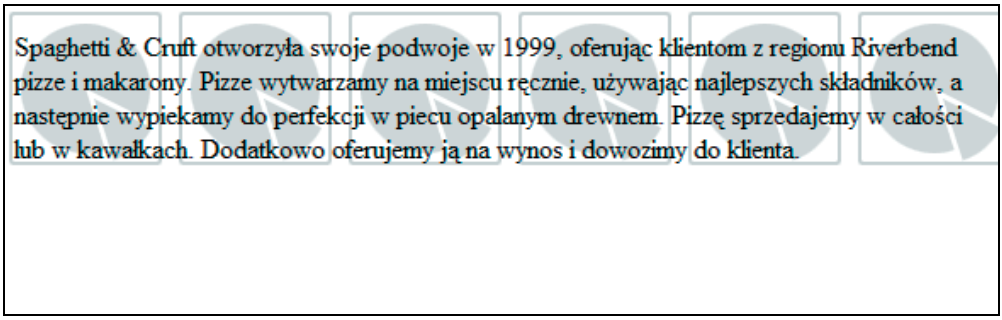


Rysunek 5.14. *Obrazek tła powtarza się w obu kierunkach tyle razy, ile to konieczne*

Listing 5.10. *Dodanie deklaracji background-repeat*

```
body {
  background-image: url(/images/background.gif);
  background-repeat: repeat-x;
}
```

Zauważ, że na rysunku 5.15 tło powtarza się tylko w poziomie.



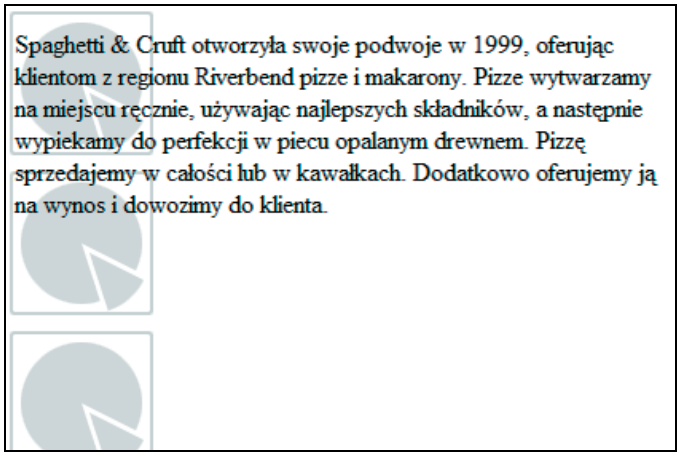
Rysunek 5.15. *Tło powtarzane w poziomie, ale nie w pionie*

Użycie wartości `repeat-y` spowoduje powtarzanie tła tylko w pionie. Listing 5.11 przedstawia zmodyfikowany kod CSS.

Listing 5.11. *Wartość repeat-y spowoduje powtarzanie tła w pionie*

```
body {
  background-image: url(/images/background.gif);
  background-repeat: repeat-y;
}
```

Rysunek 5.16 przedstawia efekt wykonania modyfikacji. Tło powtarza się tylko w osi Y.



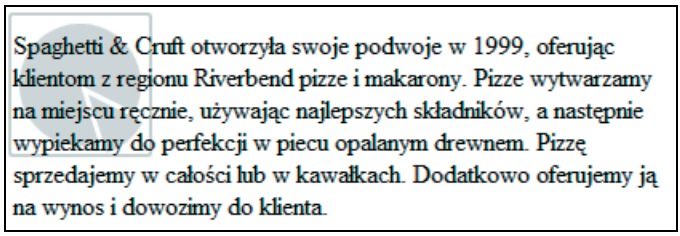
Rysunek 5.16. *Teraz tło powtarza się w pionie, a nie w poziomie*

Domyślną wartością właściwości `background-repeat` jest `repeat`. Można to wykorzystać do nadpisania wartości zmienionej przez inną regułę. Aby wyłączyć powtarzanie tła, wystarczy użyć wartości `no-repeat`, co przedstawia listing 5.12.

Listing 5.12. *Wartość no-repeat zapobiega powtarzaniu tła*

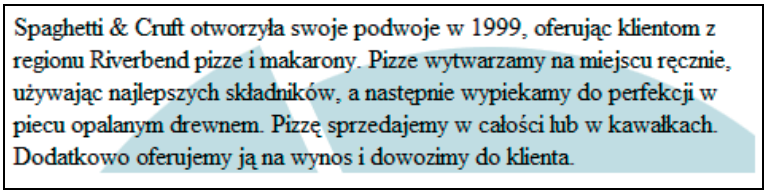
```
body {
  background-image: url(/images/background.gif);
  background-repeat: no-repeat;
}
```

Rysunek 5.17 przedstawia efekt działania wartości `no-repeat`. Obraz tła pojawił się tylko raz.



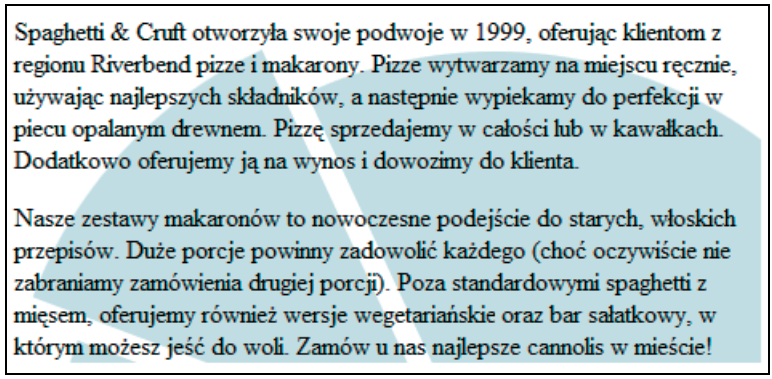
Rysunek 5.17. *Obraz tła pojawia się tylko raz (nie jest powtarzany)*

Jeśli obraz tła jest znacznie większy od właściwej treści, to określa ona obszar, do którego zostanie przycięte tło. Na rysunku 5.18 tło jest znacznie większe niż element, do którego zostało przypisane (element `div`), więc jego znaczna część pozostaje niewidoczna.



Rysunek 5.18. *Jedynie część obrazu tła jest widoczna, ponieważ obraz okazał się znacznie większy niż element, dla którego go zastosowano*

Jeśli element stanie się większy — przez dodanie nowej zawartości, zwiększenie rozmiaru tekstu lub wskazanie jawnie jego rozmiaru za pomocą CSS — więcej obrazu tła pojawi się na ekranie komputera (rysunek 5.19).



Rysunek 5.19. *Dodanie następnego akapitu zwiększa obszar obejmowany przez tło, więc widać więcej obrazu tła*

Tekst może być trudny do odczytania, jeśli obraz tła jest bardzo kolorowy i nie zapewnia odpowiedniego kontrastu z tekstem. Bądź ostrożny w doborze obrazów tła, by nie pogarszały one czytelności tekstu.

Nie zapominaj również o podaniu jednokolorowego tła (właściwość `background-color`), które zostanie użyte, jeśli obraz nie będzie dostępny. Niech zwykłe tło również zapewnia odpowiedni kontrast względem tekstu. Większość przeglądarek domyślnie stosuje czarny tekst i białe tło, ale jeśli chce się zastosować jasne tło, warto dodatkowo ustawić ciemne tło. Obraz tła zawsze przykryje zwykłe jednokolorowe tło, ale jednokolorowe tło pozostanie, gdy obrazu nie uda się wczytać.

Pozycjonowanie obrazu tła

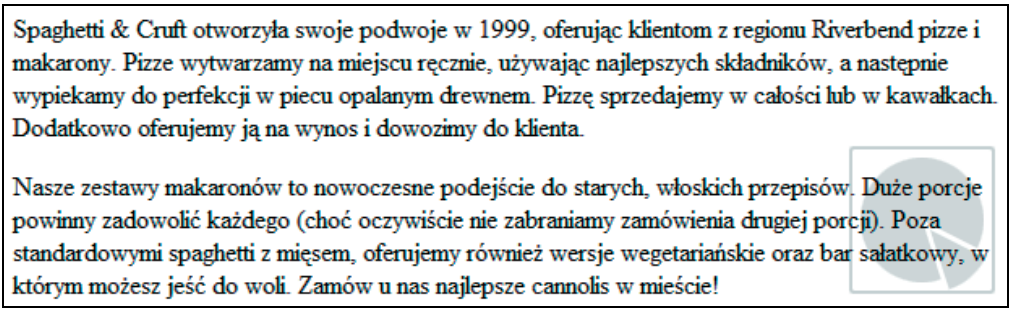
Domyślnie przeglądarka dosuwa obraz tła do lewego górnego narożnika elementu, do którego został przypisany. Jest to również miejsce rozpoczynania kafelkowania. Właściwość `background-position` steruje położeniem początku obrazu tła.

Właściwość przyjmuje dwie wartości. Pierwsza określa położenie w poziomie, a druga w pionie. Położenie w pionie zawsze występuje jako drugie. Gdybyśmy podali tylko jedną wartość, zostanie ona zaklasyfikowana jako położenie w poziomie. Listing 5.13 przedstawia kod CSS, powodujący umieszczenie obrazu tła w prawym dolnym narożniku elementu `div`.

Listing 5.13. *Dodanie deklaracji background-position*

```
div {
  background-image: url(/images/background.gif);
  background-repeat: no-repeat;
  background-position: right bottom;
}
```

Rysunek 5.20 przedstawia wynik — obraz tła znajduje się teraz w prawym dolnym narożniku tekstu. Ponieważ tło jest przypisane do elementu `div`, a element ten domyślnie automatycznie dostosowuje się do długości tekstu, obraz tła nie znajduje się w prawym dolnym narożniku okna, ale w prawym dolnym narożniku elementu.



Rysunek 5.20. *Obraz znajduje się teraz w prawym dolnym narożniku, ale nadal się nie powtarza*

Wartość właściwości `background-position` można określać na kilka sposobów: za pomocą słów kluczowych, długości lub procentów. Możliwymi słowami kluczowymi dla położenia w poziomie są `left`, `center` i `right`, a dla położenia w pionie `top`, `center` i `bottom`. Słowo `center` jest dostępne dla obu kierunków: dla poziomu oznacza połowę szerokości, a dla pionu połowę wysokości.

Długość we właściwości położenia tła to dowolna wartość liczbowa z jednostką, czyli pikselami, milimetrami lub em. Dla długości 0 nie potrzeba żadnej jednostki, ponieważ `0px` to dokładnie to samo co `0em` lub `0mm`. Listing 5.14 przedstawia dwie długości zastosowane do pozycjonowania tła: 40 pikseli dla położenia w poziomie i 3 em dla położenia w pionie.

Listing 5.14. *Zastosowanie długości w background-position*

```
body {
  background-image: url(/images/background.gif);
  background-repeat: no-repeat;
  background-position: 40px 3em;
}
```

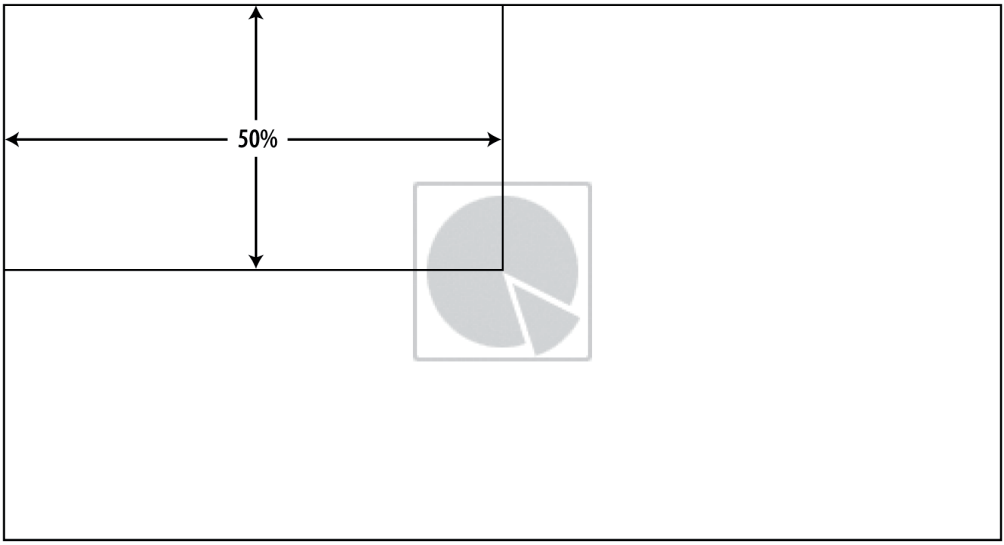
Rysunek 5.21 przedstawia zrenderowany wynik z obrazem tła umieszczonym 40 pikseli od górnego narożnika elementu i 3 em od jego lewej krawędzi, czyli zgodnie z danymi wskazanymi w CSS.

Po podaniu położenia w procentach przeglądarka w trakcie renderingu uwzględnia zarówno rozmiar elementu, jak i rozmiar obrazu tła. Obraz umieszczany 75% od lewej strony elementu zostanie umieszczony w $^3/_4$ szerokości elementu względem $^3/_4$ rozmiaru obrazka. Użycie wartości 50% spowoduje użycie jako punktu referencyjnego środka obrazu i dodatkowo ustawienie go na środku elementu, co ilustruje rysunek 5.22.

Spaghetti & Cruft otworzyła swoje podwoje w 1999, oferując klientom z regionu Riverbend pizze i makarony. Pizze wytwarzamy na miejscu ręcznie, używając najlepszych składników, a następnie wypiekamy do perfekcji w piecu opalanym drewnem. Pizzę sprzedajemy w całości lub w kawałkach. Dodatkowo oferujemy ją na wynos i dowozimy do klienta.

Nasze zestawy makaronów to nowoczesne podejście do starych, włoskich przepisów. Duże porcje powinny zadowolić każdego (choć oczywiście nie zabraniamy zamówienia drugiej porcji). Poza standardowymi spaghetti z mięsem, oferujemy również wersje wegetariańskie oraz bar sałatkowy, w którym możesz jeść do woli. Zamów u nas najlepsze cannolis w mieście!

Rysunek 5.21. *Obraz tła umieszczony dokładnie tam, gdzie wskazuje CSS*

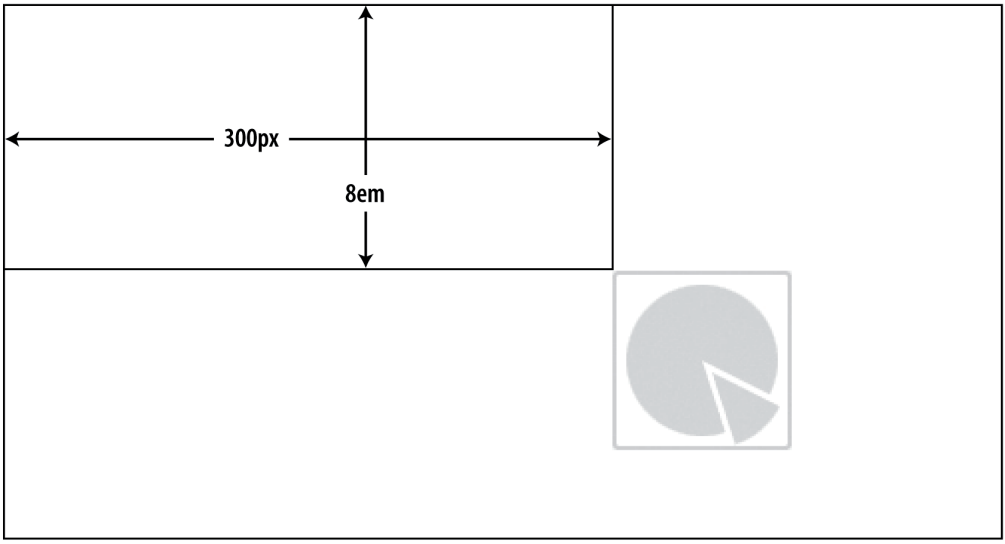


Rysunek 5.22. *Obraz tła umieszczony 50% od góry i od lewej strony spowoduje umieszczenie obrazu tła dokładnie na środku elementu*

Opisany system mierzenia nie jest stosowany dla standardowych jednostek. One zawsze powodują mierzenie od lewego górnego narożnika elementu i lewego górnego narożnika obrazu, co ilustruje rysunek 5.23.

Podsumowanie

Obrazy to bardzo ważny aspekt każdej witryny internetowej, bo to one czynią ją unikatową i rozpoznawalną. Co więcej, pewne zagadnienia znacznie łatwiej wyrazić grafiką niż słowami. Obrazy niosące treść związaną z witryną należy umieszczać za pomocą elementu `img` i dodatkowo zawsze stosować tekst alternatywny dla osób niedowidzących lub systemów wyszukiwawczych. Obrazy związane z prezentacją nie powinny znaleźć się w treści dokumentu, ale zostać określone za pomocą CSS i właściwości



Rysunek 5.23. *Zastosowanie długości powoduje użycie dokładnie tych danych, jakie zostały wskazane w CSS. Obliczenia dotyczą lewych górnych narożników elementu i obrazu*

`background-image`. CSS umożliwia dokładne pozycjonowanie obrazu tła oraz wpływanie na obrazy umieszczone w tekście, tak aby lepiej integrowały się z zawartością dokumentu.

Trzema popularnymi formatami kompresji obrazów używanymi na stronach WWW są: GIF, JPEG i PNG. Wykorzystuj obrazy JPEG dla zdjęć, obrazy GIF i 8-bitowe PNG dla tekstu i ilustracji oraz 24-bitowe PNG dla kanału przezroczystości.

Zapewne bardzo często będziesz korzystał z elementów i właściwości przedstawionych w tym rozdziale, aby wzbogacić szatę graficzną własnej witryny. Strony WWW nie byłyby stronami WWW, gdyby nie jeden element — hiperłącza. To właśnie nimi zajmiemy się w następnym rozdziale. To dzięki nim internet „żyje własnym życiem”.