

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

XHTML i CSS. Dostępne witryny internetowe

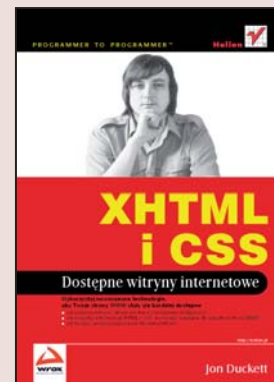
Autor: Jon Duckett

Tłumaczenie: Radosław Meryk

ISBN: 978-83-246-1293-2

Tytuł oryginału: [Accessible XHTML](#)
and [CSS Web Sites Problem Design Solution](#)

Format: 168 x 237, stron: 488



Wykorzystaj nowoczesne technologie, aby Twoje strony WWW stały się bardziej dostępne

- Jak przekonwertować witrynę zgodnie z wymaganiami dostępności?
- Jak korzystać z technologii XHTML i CSS, aby tworzyć przyjazne dla wszystkich strony WWW?
- Jak tworzyć serwisy przeznaczone dla wielu platform?

Nowoczesne technologie zapewniania dostępności koncentrują się na umożliwieniu korzystania z internetu wszystkim ludziom - w jak największym zakresie. Szczególnie istotne jest przystosowanie witryn do potrzeb osób niepełnosprawnych. Tworzenie takich stron ma istotne znaczenie nie tylko ze względów społecznych, ale także ekonomicznych i prawnych (Amerykanie i Brytyjczycy wprowadzili ustawę antydyskryminacyjną, która nakłada na przedsiębiorstwa obowiązek posiadania witryn internetowych dostępnych dla wszystkich). Niektórzy twórcy stron WWW obawiają się, że ograniczy to ich kreatywność. Jednak współczesne narzędzia sprawiają, że aplikacje nie tylko nie tracą na atrakcyjności, ale jeszcze powiększa się grupa potencjalnych odbiorców, co może przekładać się na wzrost zysków firmy.

Książka, którą trzymasz w rękach, prezentuje różne zastosowania nowych technologii na konkretnym przykładzie konwersji witryny napisanej w języku HTML na nowoczesną stronę w XHTML, wykorzystującą kaskadowe arkusze stylów o zwiększonej dostępności. Utworzone w ten sposób strony nie dyskryminują niepełnosprawnych użytkowników, mających trudności z czytaniem małych czcionek lub posługiwaniem się tradycyjnymi urządzeniami sterującymi. Podręcznik ten jest przeznaczony dla wszystkich, którzy korzystali wcześniej z HTML, a teraz chcą zaktualizować swoją wiedzę i umiejętności. Dzięki tej książce naprawdę szybko nauczysz się pisania kodu w XHTML czy korzystania z CSS przy zarządzaniu układem strony, a także poznasz reguły i wytyczne dotyczące dostępności oraz technologie ją wspomagające.

- Transformacja kodu HTML na XHTML
- Narzędzia XML i możliwości ich używania
- Zastosowanie CSS do definiowania stylu dokumentów
- Rodzaje selektorów
- Czcionki i właściwości tekstowe
- Pseudoelementy tekstowe
- Wymiary, kolor i tło
- Walidacja arkusza CSS
- Wykorzystanie CSS do zarządzania układem dokumentu
- Reguły i wytyczne dotyczące dostępności

Aby Twoje strony były atrakcyjne i dostępne dla wszystkich!



Spis treści

O autorze	7
Przedmowa	9
Wprowadzenie	11
Rozdział 1. Przedstawiamy witrynę	17
Problemy z językiem HTML	18
Projekt	19
Terminologia	20
Wprowadzenie do języka HTML	21
Problemy związane z przeglądarkami	25
Lekcje ze środowiska telefonów komórkowych	26
Dostępność	27
Oddzielenie stylu od treści	29
„X” w nazwie XHTML	30
Historia technologii CSS	31
Korzystanie z arkuszy stylów	32
Wprowadzenie do przykładowej witryny	34
Strona główna	37
Strony z listami produktów	44
Strony ze szczegółowymi danymi o produktach	47
Struktura witryny	51
Aktualizacja witryny	52
Podsumowanie	54
Rozdział 2. Transformacja kodu HTML na XHTML	55
Problem	56
Projekt	56
Strict XHTML	58
Transitional XHTML	58
Frameset XHTML	59
Przykładowy dokument XHTML	59
Pisanie dokumentów zgodnych ze składnią XML	61

Dlaczego XML?	71
Różnice pomiędzy wersjami Transitional i Strict języka XHTML 1.0	74
Obsługa XHTML w narzędziach do tworzenia stron WWW	81
Walidacja	83
Rozwiązanie	87
Strona główna	87
Strony z listą produktów	98
Strony ze szczegółowymi danymi o produktach	100
Walidacja	104
Podsumowanie	105
Rozdział 3. Zastosowanie CSS do definiowania stylu dokumentów	107
Problem	108
Projekt	108
Jak działa CSS?	108
Więcej informacji na temat dodawania reguł CSS	115
Selektory	120
Czcionki	124
Właściwości tekstowe	132
Pseudoelementy tekstowe	138
Model ramek	139
Elementy <div> i 	142
Właściwości ramek	144
Wymiary	150
Kolor i tło	154
Rozwiązanie	162
Strona główna	163
Strony z listami produktów	173
Strony ze szczegółowymi danymi produktów	175
Podsumowanie	178
Rozdział 4. Zaawansowane zastosowania języka CSS	179
Problem	180
Projekt	180
Pseudoklasy i pseudoelementy	180
Listy	195
Właściwości tabel	198
Widoczność elementów	202
Wiele arkuszy stylów i reguła @import	204
Priorytety reguł	206
Walidacja arkuszy CSS	207
Rozwiązanie	208
Poprawa wyglądu łączy	208
Obramowania tabel na stronach ze szczegółowymi danymi produktów	210
Podsumowanie	213
Rozdział 5. Wykorzystanie CSS do zarządzania układem dokumentu	215
Problem	216
Projekt	216
Schematy określania pozycji w CSS	216
Warstwy nakładające się na siebie	232

Popularne układy	235
Możliwości wyboru układu	251
Rozwiązanie	253
Podsumowanie	262
Rozdział 6. Dostępność	263
Problem	264
Projekt	265
Technologie wspomagające	266
Dodatkowe korzyści ze stosowania się do reguł dostępności	267
Wytyczne dotyczące dostępności	268
Tworzenie dostępnych stron WWW	270
Rozwiązanie	302
Zdefiniowanie odpowiedników tekstowych dla elementów nietekstowych	302
Sprawdzenie dynamicznej zawartości oraz kolorów i kontrastu	303
Dodanie łączów w celu pominięcia powtarzających się łączów nawigacyjnych	303
Podsumowanie	306
Rozdział 7. Tworzenie dostępnych tabel i formularzy oraz testowanie witryny	307
Problem	308
Projekt	309
Tabele	309
Formularze	322
Zagadnienia dostępności witryn dla osób niesłyszących	336
Dostępne ramki	337
Ostatnia deska ratunku — sam tekst	340
Sprawdzanie zgodności witryny z wymogami dostępności	341
Rozwiązanie	344
Strony ze szczegółowymi danymi o produktach	344
Testowanie witryny za pomocą programu Bobby	346
Testowanie za pomocą LIFT	347
Podsumowanie	348
Rozdział 8. Spojrzenie w przyszłość	349
Problem	350
Projekt	350
Sytuacja bieżąca	350
Nowe urządzenia	351
Lekcje ze środowiska urządzeń mobilnych	352
Moduły abstrakcyjne języka XHTML	353
XHTML 1.1	355
XHTML Basic	361
Dokumenty hybrydowe	363
Język XHTML na różnych urządzeniach	368
Rozwiązanie	377
Podsumowanie	381
Dodatek A Końcowy kod przykładu	383
Strona główna	384
Strony z listą produktów	386
Strony ze szczegółowymi danymi o produktach	389
Arkusz stylów CSS	391

Dodatek B Elementy XHTML	397
Dodatek C Właściwości CSS	435
Właściwości dotyczące czcionek	435
Właściwości tekstowe	439
Właściwości dotyczące koloru i tła	442
Właściwości dotyczące obramowania	444
Wymiary	446
Właściwości marginesów	449
Właściwości wypełnienia	449
Właściwości list	450
Właściwości opisujące pozycje	451
Właściwości obrysu	456
Właściwości tabel	456
Właściwości dodatkowe	458
Właściwości międzynarodowe	460
Jednostki długości	461
Dodatek D Sekwencje sterujące	463
Wbudowane sekwencje sterujące języka XML	464
Encje znakowe XHTML	464
Dodatek E Typy MIME	467
Tekst	468
Grafika	468
Dokumenty wieloczęściowe	468
Dźwięk	468
Wideo	469
Aplikacje	469
Skorowidz	471

1

Przedstawiamy witrynę

W ciągu stosunkowo krótkiego czasu internet rozwinął się w niezwykłym tempie. Kiedy zacząłem uczyć się języka HTML, nie wyobrażałem sobie nawet, że dziś tak wiele osób będzie korzystało z internetu. Siedząc przy moim komputerze desktop PC, nie wyobrażałem sobie również tego, że obecnie będę musiał pisać strony, do których można uzyskać dostęp za pomocą tak wielu różnych urządzeń — na przykład telefonów komórkowych bądź przystawek telewizyjnych typu STB. Te nowe urządzenia dość znacznie się różnią od komputerów desktop — mają ekrany innych rozmiarów, a także inną moc obliczeniową i dostępną ilość pamięci, a poza tym pozwalają użytkownikom na dostęp do informacji na wiele różnych sposobów. Biorąc pod uwagę skalę rozwoju internetu oraz sposób, w jaki się zmienił, nie ma się co dziwić, że ci z nas, którzy zajmują się tworzeniem witryn WWW, muszą zadbać o zaktualizowanie swoich umiejętności oraz że narzędzia wykorzystywane do tej pracy również wymagają modernizacji.

W tym rozdziale opisano, dlaczego warto nauczyć się pisania stron za pomocą XHTML i CSS oraz dlaczego tak bardzo ważne jest zadbanie o to, aby strony były dostępne. W tym rozdziale zaprezentujemy również przykładową witrynę, z którą będziemy pracować na kartach tej książki.

W tym rozdziale:

- przeanalizujemy niektóre problemy, jakie sprawia używanie tradycyjnego języka HTML;
- dowiemy się, dlaczego istotne znaczenie ma oddzielenie stylu od treści;
- zapoznamy się z przykładową witryną Fajna Promocja; jest to witryna, którą będziemy aktualizowali na kartach tej książki;
- dowiemy się, jakie są cele modyfikacji projektu witryny;
- dowiemy się czegoś więcej na temat korzyści, jakie można uzyskać, tworząc dostępne witryny z wykorzystaniem XHTML i CSS.

Po przeczytaniu niniejszego rozdziału Czytelnicy będą rozumieli powody, dla których należy zaktualizować swoje umiejętności i zmodyfikować witryny, oraz będą wiedzieli, jakie działania będą wykonywane z przykładową witryną zamieszczoną w tej książce.

Problemy z językiem HTML

Równoległe ze zmianami, jakie zaszły w internecie, zmieniał się również język HTML. Organizacja W3C (World Wide Web Consortium — zajmująca się ustalaniem wielu standardów internetowych) opublikowała kilka wersji języka HTML. Pierwsza pozwalała na korzystanie z zestawu znaczników do opisywania struktury dokumentu, ale nie oferowała zbyt wielu mechanizmów zarządzania ich wyglądem. Wersja HTML 4 jest o wiele bardziej złożona od swojej pierwotnej wersji. Własności dodawane w ciągu wielu lat do języka HTML dostarczyły projektantom stron internetowych wielu mechanizmów zarządzania wyglądem stron. Na przykład, autorzy stron WWW uzyskali mechanizm zarządzania dokładną szerokością tabel (co do piksela), rozmiarem i wielkością czcionek, kolorami, grafiką używaną w tle itp.

Dodanie nowych reguł pozwalających projektantom na zarządzanie wyglądem stron oznaczało możliwość tworzenia atrakcyjnych stron, które doskonale działały na komputerach biurkowych o przeciętnej mocy obliczeniowej. Pomimo tego, projektanci stron w dalszym ciągu napotykali trudności:

- Producenci różnych przeglądarek opracowują różne sposoby wykonywania tych samych zadań, co oznacza, że witryny nie wyglądają ani nie działają tak samo we wszystkich przeglądarkach.
- Różni użytkownicy stosują ekrany różnych rozmiarów oraz różne rozdzielczości, zatem strony nie wyglądają tak samo na wszystkich komputerach.
- Powstały nowe urządzenia pozwalające na dostęp do internetu. Nie były to już wyłącznie biurowe komputery PC, ale także przystawki telewizyjne typu STB, telefony komórkowe itp., a każde nowe urządzenie charakteryzowało się nowym zestawem funkcji. Na przykład strona zawierająca tabelę o szerokości 700 pikseli nie zmieści się na ekranie telefonu o rozmiarze 128 pikseli.
- Jeśli programista (lub jego szef) zechciał zmienić jakieś elementy projektu strony — na przykład kolory lub używane czcionki, musiał zmieniać wszystkie strony, ponieważ informacje te były zapisane w kodzie HTML każdej ze stron.
- W miarę jak projektanci zaczęli używać coraz więcej kodu HTML do zarządzania układem stron, kod źródłowy stron stawał się coraz obszerniejszy, a jego pisanie stawało się coraz bardziej skomplikowane. To z kolei stwarzało więcej okazji do popełniania błędów podczas tworzenia oraz edycji witryn.
- Użytkownicy z wadami wzroku mogą mieć trudności z przeczytaniem niektórych fragmentów tekstu w witrynach, a osoby z poważnym osłabieniem widzenia w ogóle nie są w stanie poruszać się po niektórych witrynach.

W rezultacie niektóre firmy zaczęły tworzyć po kilka wersji swoich witryn WWW — opracowywano różne wersje dla różnych przeglądarek i urządzeń, a nawet tworzone wersje wyłącznie tekstowe mające na celu spełnienie wymagań osób niepełnosprawnych. Oczywiście oznaczało to mnóstwo dodatkowej pracy, a czasami prowadziło do sytuacji, w której pewne wersje witryn nie były tak aktualne, jak pozostałe. Oczywiście takie rozwiązanie jest dalekie od idealnego.

Podsumowując, posługiwanie się tradycyjnym językiem HTML stwarzało następujące problemy:

- W miarę rozwoju języka HTML wprowadzano do niego mnóstwo reguł definiowania stylu pozwalających autorom stron WWW na zarządzanie wyglądem swoich stron. Jednak te usprawnienia w języku oznaczały, że reguły rządzące wyglądem stron mieszało z zasadniczą treścią dokumentów i wiele z nich działało w zamierzony sposób wyłącznie w biurkowych komputerach PC.
- Ponieważ internet jest tak rozbudowanym medium, powinien być dostępny dla maksymalnie szerokiego grona użytkowników. Dotyczy to również osób niepełnosprawnych mających trudności z czytaniem, słyszeniem lub posługujących się myślą nie tak sprawnie, jak inni.

Jak można zauważyć, zmiany, jakie trzeba wprowadzić w sposobie tworzenia stron, odzwierciedlają jedynie wzrost popularności internetu oraz zmiany w sposobach posługiwania się nim. W tym rozdziale dowiemy się, w jaki sposób i dlaczego zmieniły się języki tworzenia stron WWW.

Projekt

Po opublikowaniu HTML 4.0, organizacja W3C (odpowiedzialna za opracowywanie specyfikacji języka HTML) zdecydowała, że nadszedł czas na wielką przemianę w sposobie pisania witryn internetowych. Problemy opisane w poprzednim podrozdziale dawały wskazówkę organizacji W3C, że zmiany powinny odzwierciedlać następujący prosty fakt:

Różni użytkownicy wymagają różnego sposobu prezentacji tych samych informacji.

Oto trzy najczęstsze powody, dla których trzeba prezentować informacje w witrynach WWW na różne sposoby:

- Różne technologie używane do przeglądania stron internetowych mają różne możliwości i nie wszystkie są w stanie wyświetlić ten sam projekt tak samo dla wszystkich użytkowników.
- Informacje dostępne online często są przydatne w różnych formatach, na przykład w postaci wydrukowanej lub w formie slajdów prezentowanych na konferencji.
- Niektóre osoby mają trudności z dostępem do informacji w sposób, który dla innych jest łatwy. Na przykład osoby z wadami wzroku mogą mieć trudności z czytaniem tekstu na ekranie komputera.

Ten ostatni punkt jest szczególnie ważny, bowiem internet w znaczący sposób zmienia codzienne życie wielu osób — zastosowania internetu obejmują szeroki zakres operacji (począwszy od możliwości robienia zakupów, korzystania z usług bankowych i płacenia rachunków online, a skończywszy na próbach elektronicznego głosowania) — w związku z tym nie można wyłączyć z grona jego użytkowników osób niepełnosprawnych.

Internet może mieć olbrzymi wpływ na życie osób niepełnosprawnych, zwłaszcza tych, dla których opuszczanie domu jest trudniejsze niż dla osób sprawnych. W związku z tym jest bardzo ważne, aby z osiągnąć postępu technicznego mogli korzystać wszyscy.

Organizacja W3C zaproponowała następujące rozwiązanie tych problemów: usunięcie z zestawu znaczników wszystkich elementów, przez lata dodawanych do HTML, które służyły do **prezentacji** danych, a zatem takich jak `<font>` i `<center>` oraz atrybuty `bgcolor` i `color`. Pozostałe znaczniki HTML opisują strukturę i semantykę dokumentów, na przykład tytuł dokumentu, początek i koniec akapitu, teksty reprezentujące nagłówki itp. Nie oznacza to, że projektanci stron WWW stracili kontrolę nad wyglądem dokumentów. Zaproponowano im używanie **arkuszy stylów** — osobnych dokumentów określających sposób wyświetlania stron. W rzeczywistości arkusze stylów mają kilka innych zalet. Zostaną one opisane w dalszej części tego rozdziału.

W rezultacie oddzielenia stylu od treści organizacja W3C opracowała dwa standardy: **XHTML** i **CSS**. Producenci przeglądarek powinni zadbać o ich obsługę, natomiast autorzy stron WWW powinni się ich nauczyć. Nie należy się jednak tym zrażać. W rzeczywistości oba języki są bardzo podobne do języka HTML, który już znamy. Dzięki temu proces uczenia się wymienionych technologii jest dość szybki i prosty.

Przed przystąpieniem do analizowania tego, w jaki sposób zmieniły się języki tworzenia stron internetowych, w następnym punkcie objaśnimy kilka kluczowych terminów. Dzięki temu będę miał pewność, że Czytelnicy rozumieją je w taki sam sposób, jak ja. Następnie opowiem trochę o tym, jak to się stało i dlaczego dziś stajemy przed koniecznością nauki nowych języków oraz wyjaśnię różnice pomiędzy prezentacyjnym a strukturalnym zestawem znaczników.

Terminologia

Z pewnością wiele spośród terminów opisanych poniżej będzie brzmiało znajomo dla większości Czytelników. Trzeba jednak pamiętać, że czasami różne osoby opisują tę samą rzecz, używając nieco innych pojęć. Szybki przegląd niezbędnych definicji da nam pewność, że Czytelnicy przed przystąpieniem do aktualizowania swoich umiejętności będą właściwie rozumieli pojęcia zamieszczone w poniższym tekście.

Znacznik to zbiór znaków ujęty w nawiasy trójkątne. Na przykład poniżej zamieszczono popularny **znacznik otwierający**:

```
<td>
```

Towarzyszy mu znacznik **zamykający**. Wszystko, co znajduje się pomiędzy tymi znacznikami, występuje jako **dane tabeli** (ang. *table data* — stąd litery td):

```
</td>
```

Element odnosi się zarówno do znacznika otwierającego, jak i zamykającego, plus wszystkiego, co znajduje się pomiędzy nimi. Na przykład poniżej pokazano element komórki tabeli:

```
<td>234.5</td>
```

Treść występującą pomiędzy znacznikami otwierającym i zamykającym określa się terminem **zawartość elementu**.

Elementy mogą mieć **atrybuty**, które dodatkowo je opisują. Atrybuty zawsze umieszcza się wewnątrz znacznika otwierającego element. Oto przykład użycia atrybutu `align`:

```
<td align="right">
```

Powyższy atrybut wskazuje na to, że zawartość komórki tabeli powinna być wyrównana do prawej.

Dla niektórych elementów nie używa się znaczników zamykających. Na przykład element `<img>` może mieć kilka atrybutów, ale nie ma żadnej zawartości — nie ma niczego pomiędzy znacznikiem otwierającym a zamykającym:

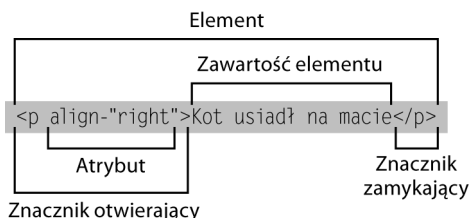
```

```

Elementy bez zawartości określa się jako tzw. **puste elementy**. Jak się przekonamy w rozdziale 2., puste elementy w XHTML zapisuje się inaczej, zatem trzeba się nauczyć pisanie znaczników `<img>`, `<br>` i `<hr>` w nowy sposób.

Przedstawione pojęcia pokazano na rysunku 1.1.

Rysunek 1.1



Wprowadzenie do języka HTML

Usunięcie z języka HTML wszystkich reguł dotyczących stylu można uznać raczej za krok wstecz niż za ekscytującą nową cechę języka. Aby zrozumieć, dlaczego język HTML ewoluował w taki sposób — tzn. dlaczego wprowadzono w nim zestaw znaczników do zarządzania stylami, takich jak elementy `<font>` i atrybut `bgcolor`, skoro miały one być usunięte — należy przyjrzeć się, jak doszło do tego, że HTML rozwinął się w język, który dziś znamy i używamy. Jak to zwykle bywa, odpowiedź okazała się oczywista poniewczasie — cóż, problemy nie zawsze dadzą się przewidzieć.

HTML opracowano jako język znaczników w celu opisania **struktury** i **semantyki** dokumentu. Elementy i atrybuty HTML miały wskazywać takie fragmenty, jak tytuł dokumentu, nagłówki, akapity, dane należące do akapitu itp. W najwcześniejszej postaci internet miał służyć do przesyłania dokumentów naukowych, tak by społeczność miała szybki i łatwy dostęp do opublikowanej pracy. Oto fragment dokumentu, który może być przykładem informacji wygenerowanych dla społeczności naukowej (ten kod jest zapisany w pliku `r01_list01.html` dostępnym do pobrania):

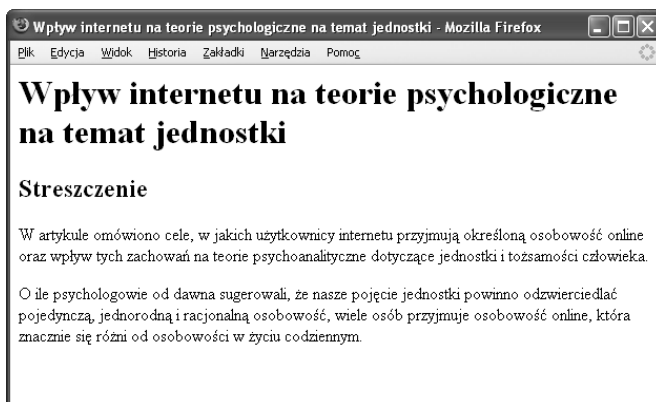
```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/1999/
↳REC-html401-19991224/strict.dtd">
<html>
  <head>
    <title>Wpływ internetu na teorie psychologiczne na temat jednostki</title>
  </head>
  <body>
    <h1>Wpływ internetu na teorie psychologiczne na temat jednostki</h1>
    <h2>Streszczenie</h2>
    <p>W artykule omówiono cele, w jakich użytkownicy internetu przyjmują określoną
↳osobowość online oraz wpływ tych zachowań na teorie psychoanalityczne
↳dotyczące jednostki i tożsamości człowieka.</p>
    <p>O ile psychologowie od dawna sugerowali, że nasze pojęcie jednostki
↳powinno odzwierciedlać pojedynczą, jednorodną i racjonalną osobowość, wiele
↳osób przyjmuje osobowość online, która znacznie się różni od osobowości
↳w życiu codziennym.</p>
    ...
  </body>
</html>

```

Chociaż tego typu dokument był bardzo praktyczny i umożliwiał pracownikom naukowym znacznie łatwiejsze współdzielenie informacji w porównaniu z drukowanymi magazynami, prezentacja tych dokumentów była dość monotonna. Można to zauważyć na rysunku 1.2, gdzie pokazano, jak wyglądałby dokument z listingu w przeglądarce.

Rysunek 1.2



Bardzo szybko wiele osób zdało sobie sprawę z olbrzymiego potencjału internetu. Użytkownicy ci zaczęli tworzyć strony WWW służące różnym celom — od osobistych stron do rozbudowanych serwisów firmowych mających na celu reklamę produktów i usług firm. Nie minęło zbyt wiele czasu, zanim autorzy stron WWW zdali sobie sprawę z tego, że dobrze by było zarządzać wyglądem tworzonych przez nich witryn. Oczekiwali tego samego poziomu kontroli nad wyglądem stron, jaki mają twórcy magazynów drukowanych nad swoimi dokumentami. Na przykład, projektanci stron WWW chcieli mieć możliwość zmiany czcionek i kolorów używanych w dokumentach oraz określania miejsc, w których ma się pojawić tekst na stronie. W rezultacie producenci najpopularniejszych przeglądarek zaczęli dodawać nowe elementy i atrybuty zarządzające wyglądem stron WWW. Zarówno firma Netscape, jak i Microsoft usilnie próbowały zdobyć jak największy udział w rynku przeglądarek dzięki zapewnieniu projektantom WWW większej kontroli nad stronami wyświetlanymi w przeglądarkach. Wiele takich elementów wprowadziła również organizacja W3C w kolejnych wersjach języka HTML.

Ten nowy zestaw znaczników jest znany jako **prezentacyjny** lub inaczej **stylistyczny**, ponieważ ma wpływ na wygląd strony (nie opisuje struktury i semantyki dokumentu w sposób zgodny z pierwotnym przeznaczeniem języka HTML). Najważniejszymi przykładami znaczników nowego typu są elementy `<center>` i `<font>` oraz atrybuty `bgcolor` i `color`.

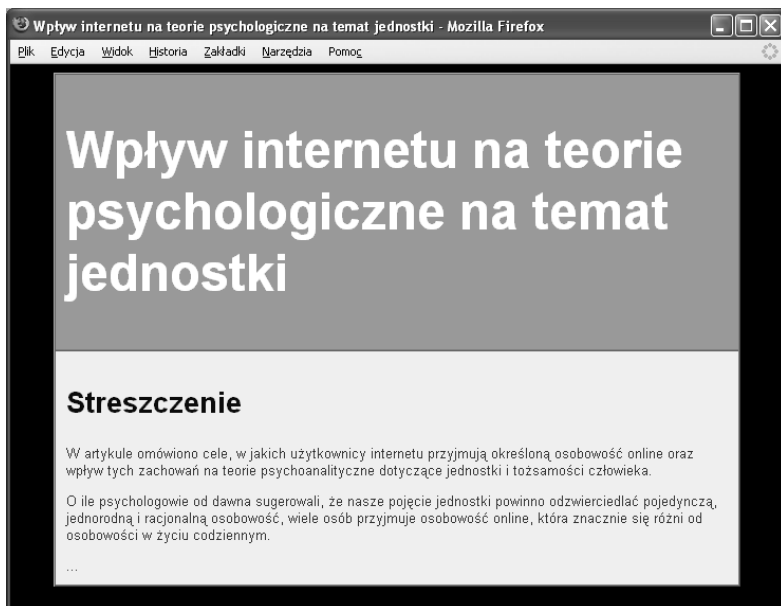
Projektanci stron szybko opanowali sztuczki umożliwiające im zarządzanie układem stron (ustawianiem pozycji tekstu na ekranie) w sposób podobny do tego, jaki wykorzystywali specjaliści od druku. Dwie najczęściej stosowane techniki rozmieszczania tekstu na stronie to użycie tabel w roli siatek oraz przezroczystych, jednopikselowych plików GIF (sztuczka znana powszechnie jako jednopikselowy GIF lub przezroczysty GIF).

Na poniższym listingu pokazano stronę z pliku `r01_list01.html` z dodanymi znacznikami zarządzającymi jej prezentacją. Nowy dokument — dostępny w pliku `r01_list02.html` — z powodu znaczników stylu dodanych do strony jest znacznie obszerniejszy od poprzedniego:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
↳ "http://www.w3.org/TR/1999/REC-html401-19991224/loose.dtd">
<html>
  <head>
    <title>Wpływ internetu na teorie psychologiczne na temat jednostki</title>
  </head>
  <body bgcolor="#000">
    <center>
      <table width="650" border="1" cellpadding="10" cellspacing="0">
        <tr>
          <td bgcolor="#999">
            <h1>
              <font face="Arial, Helvetica, sans-serif" size="5" color="#fff">
                Wpływ internetu na teorie psychologiczne na temat jednostki
              </font>
            </h1>
          </td>
        </tr>
        <tr>
          <td bgcolor="#efefef">
            <h2>
              <font face="Arial, Helvetica, sans-serif" size="4" color="#006">
                Streszczenie
              </font>
            </h2>
            <font face="Arial, Helvetica, sans-serif" size="2" color="#333">
              W artykule omówiono cele, w jakich użytkownicy internetu przyjmują
              ↳ określoną osobowość online oraz wpływ tych zachowań na teorie
              ↳ psychoanalityczne dotyczące jednostki i tożsamości człowieka.<br>
              ↳ 0 ile psychologowie od dawna sugerowali, że nasze pojęcie jednostki
              ↳ powinno odzwierciedlać pojedynczą, jednorodną i racjonalną osobowość,
              ↳ wiele osób przyjmuje osobowość online, która znacznie się różni od
              ↳ osobowości w życiu codziennym.
              ...
            </font>
          </td>
        </tr>
      </table>
    </center>
  </body>
</html>
```

Pomimo że układ dokumentu w dalszym ciągu jest stosunkowo prosty, rozmiar strony znacznie się zwiększył. Liczba znaków (bez spacji) niemal podwoiła się. Plik *r01_list01.html* miał zaledwie 529 znaków, podczas gdy dokument *r01_list02.html* zawiera aż 928 znaków. Ta dodatkowa złożoność zwiększa możliwość popełnienia błędu, zarówno podczas tworzenia strony, jak i podczas wprowadzania modyfikacji. Dokument *r01_list02.html* w postaci, w jakiej wyświetla się w przeglądarce, pokazano na rysunku 1.3.

Rysunek 1.3



Kiedy reguły zarządzające prezentacją strony WWW są pomieszczone z właściwą zawartością strony, zmiana dowolnego aspektu prezentacji witryny (na przykład kroju czcionki nagłówków lub koloru tła strony) oznacza konieczność zmiany *każdej* strony w witrynie. Kiedy weźmie się pod uwagę fakt, że wiele stron jest zaśmieconych znacznikami `<font>` oraz mnóstwem atrybutów prezentacyjnych, znalezienie znacznika, który chcemy zmienić, staje się bardzo trudne. Jak łatwo zauważyć, wprowadzenie definicji stylów do zestawu znaczników HTML stworzyło więcej okazji do popełnienia błędów.

Równocześnie z pisanem nowych wersji przeglądarek zdolnych do obsługi nowego zestawu znaczników, ich producenci starali się ułatwić pisanie stron ich autorom poprzez wprowadzenie możliwości zdefiniowania kodu, który wyświetlałby strony nawet wtedy, kiedy kod HTML zawiera błędy. Oznaczało to, że wielu autorów dokumentów HTML uczyło się złych nawyków. Nawet niektóre wiodące narzędzia do tworzenia stron generowały niezgrabny kod. Co więcej, ponieważ producenci przeglądarek tworzyli je w taki sposób, by mogły generować strony nawet wtedy, gdy dokumenty zawierały błędy, musiały się zwiększyć wymagania w zakresie ilości miejsca na dysku oraz pamięci potrzebnej do działania przeglądarek.

Żadne z wprowadzonych usprawnień nie było z gruntu złe. Gdyby nie można było tworzyć atrakcyjnych stron WWW oraz gdyby tworzenie stron WWW nie było tak łatwe, jak jest dzisiaj, to internet prawdopodobnie nie stałby się tak popularny, jak w tej chwili. Jednak jak już się przekonaliśmy, z powodu wprowadzonych nowych znaczników oraz sposobów kontroli układu dokumentów stosowanych przez projektantów, strony WWW stawały się coraz bardziej obszerne.

Zestaw znaczników nie służył już tylko do opisanego struktury strony oraz wskazywania elementów dokumentu, ale także zawierał mnóstwo znaczników, których wyłącznym celem było zarządzanie prezentacją strony dla urządzenia jednego typu: biurkowego komputera PC.

Problemy związane z przeglądarkami

Problemy związane z pisaniem stron zawierających reguły zarządzające ich wyglądem zaostbrał fakt, iż sposób, w jaki użytkownicy przeglądali strony, zmieniał się równie szybko, jak rosła liczba użytkowników internetu. Jak już wspominałem, postęp w technologiach wiązał się z następującymi faktami:

- zwiększały się możliwości procesorów i kart graficznych;
- stały się dostępne ekrany o wyższej rozdzielczości;
- spadała cena większych ekranów, a wraz ze spadkiem cen, stawały się one coraz powszechniejsze;
- pojawiły się urządzenia inne niż komputery desktop PC zdolne do korzystania z internetu, takie jak telefony komórkowe i przystawki STB; obecnie produkuje się nawet lodówki z wbudowanymi przeglądarkami WWW.

Każdy z tych aspektów stwarzał nowe wyzwania:

- W miarę poprawy jakości procesorów i kart graficznych, coraz więcej osób zachęcano do używania ekranów o wyższej rozdzielczości.
- Ze względu na powszechniejsze wykorzystanie większych ekranów, więcej osób zaczęło używać wyższych rozdzielczości.
- Witryny zaprojektowane dla rozdzielczości 640×480 w rozdzielczości 1024×768 wydawały się mniejsze, dlatego czytanie ich w wyższych rozdzielczościach stawało się trudniejsze.
- Nowe urządzenia zwykle miały mniejsze ekrany, niższe rozdzielczości, mniejszą moc obliczeniową oraz mniej pamięci do dyspozycji.

Najwięcej wyzwań stwarzały nowe typy urządzeń używanych do dostępu do internetu, takie jak telefony komórkowe, komputery PDA (na przykład Blackberry i Palm Pilot) oraz przystawki STB, ponieważ ich ekrany znacznie się różniły od ekranów komputerów PC typu desktop (nie chodziło tylko o nieco większy rozmiar lub trochę wyższą rozdzielczość). Z całą pewnością dla autorów stron WWW nie byłaby dobra konieczność pisania stron WWW od początku dla każdego urządzenia, które może uzyskać dostęp do internetu.

Problemy opisane powyżej — podkreślę to raz jeszcze — są związane ze sposobem, w jaki zaprojektowano język HTML. Gdyby można było oddzielić treść od reguł zarządzających sposobem wyświetlania stron na komputerach desktop PC, można by zaprezentować tę samą treść dla kolejnego urządzenia poprzez stworzenie tylko jednego nowego zbioru reguł prezentacyjnych wykorzystywanych przez każdą stronę serwisu WWW (w odróżnieniu od konieczności modyfikowania każdej strony i dostosowywania jej do stylu właściwego dla nowego urządzenia).

Oczywiście stworzenie od podstaw nowej wersji witryny dla każdego typu urządzenia, które może uzyskać dostęp do internetu, nie jest rozwiązaniem praktycznym. Jak się przekonamy, można się wiele nauczyć, przyglądając się sposobom, w jakie pierwsze urządzenia mobilne korzystały z sieci. Można śmiało stwierdzić, że sytuacja opisana powyżej spowodowała zintensyfikowanie wysiłków organizacji W3C w kierunku opracowania nowej wersji języka XHTML.

Lekcje ze środowiska telefonów komórkowych

Kiedy producenci telefonów komórkowych postanowili stworzyć telefony pozwalające na korzystanie z internetu, wiedzieli, że strony HTML dostępne w internecie nie są odpowiednie dla tych urządzeń. Nie tylko strony te były zaprojektowane dla ekranów znacznie większych i bardziej skomplikowanych, ale również telefony dysponowały znacznie mniejszą ilością pamięci i miały zbyt słabą moc obliczeniową, by można było używać ich do przeglądania stron.

Kluczowym elementem była pamięć, ponieważ jak wspomniałem wcześniej, przeglądarki zaprojektowane do pracy na komputerach biurkowych miały obszerny kod pozwalający na wyświetlanie stron nawet wtedy, gdy były z błędami. Jeśli weźmie się pod uwagę, że wymagania systemowe przeglądarki Netscape 7.2 to 26 MB miejsca na twardym dysku oraz 64 MB pamięci RAM, nietrudno dojść do wniosku, że to za dużo jak na przeciętny telefon komórkowy.

Ponieważ telefony komórkowe korzystają z internetu w tak bardzo różny sposób w porównaniu z komputerami PC, ich producenci opracowali własny język zaprojektowany specjalnie w celu udostępniania stron internetowych dla telefonów komórkowych. W rezultacie powstało kilka konkurujących ze sobą języków mających na celu udostępnianie treści z internetu telefonom komórkowym — na przykład język WML (*Wireless Markup Language*) należący do grupy specyfikacji WAP (*Wireless Application Protocol*), Compact HTML czy też HDML (*Handheld Device Markup Language*). Jeśli jednak przyjrzymy się bliżej tym wszystkim językom, przekonamy się, że mają one pewną wspólną cechę — wszystkie działają tak, jakby stanowiły podzbiór języka HTML, i pozwalają użytkownikom na wykonywanie następujących operacji:

- tworzenie tytułów, nagłówków i tekstu;
- definiowanie łączy pomiędzy stronami;
- osadzenie prostych obrazów;
- tworzenie prostych tabel;
- pobieranie danych od użytkowników za pomocą prostych formularzy.

W językach tych brakowało bardziej zaawansowanych własności przeglądarek WWW, na przykład wyświetlania złożonego układu stron z wykorzystaniem tabel oraz pobierania informacji z wykorzystaniem złożonych elementów sterujących formularzy. Nie można było również wyświetlać animacji Flash oraz uruchamiać skomplikowanych skryptów.

Języki te wymagały również od autorów ścisłego przestrzegania reguł składni — przeglądarki telefonów komórkowych nie radziły sobie z nieścisłościami w kodowaniu, jak główne przeglądarki w komputerach desktop.

Dzięki utworzeniu własnych, mniej rozbudowanych języków, producenci telefonów komórkowych mogli stworzyć mniej skomplikowane przeglądarki wymagające mniej pamięci i mocy obliczeniowej, i dzięki temu zdolne do działania w telefonach. Ponieważ można zauważyć, że wszystkie te działania można porównać do okrywania koła na nowo, podczas gdy zamiast tego można było skorzystać z okrojonego zbioru elementów HTML i atrybutów. Tak się jednak nie stało.

W kilku momentach historii internetu można było dojść do przekonania, że każde nowe urządzenie korzystające z internetu będzie wymagało opracowania własnego języka odzwierciedlającego możliwości tego urządzenia. Oznaczałoby to konieczność tworzenia kilku różnych wersji każdej strony — osobnej dla każdego urządzenia. Czytelników, którzy kiedykolwiek zmagali się z próbami stworzenia witryny działającej jednocześnie w przeglądarce Internet Explorer i Netscape, taka perspektywa z pewnością przyprawia o dreszcze. Co więcej, jeśli każde nowe urządzenie wymagało własnego języka, zaakceptowanie ich było trudniejsze, ponieważ sukces określonego urządzenia wymagał specjalnego opracowania dla niego treści dostępnych w internecie.

Jak się przekonamy w ostatnim rozdziale tej książki, rozwiązanie organizacji W3C umożliwiło wszystkim urządzeniom, zarówno dostępnym na rynku obecnie, jak i w przyszłości, na używanie języków bazujących na XHTML. Zanim jednak Czytelnicy będą mogli nauczyć się pracy z wieloma urządzeniami, muszą przejść długą drogę.

W praktyce, w przypadku tworzenia witryn przeznaczonych dla wielu platform zazwyczaj strony są generowane dynamicznie na podstawie zawartości bazy danych (nie są zaś tworzone statycznie dla każdego urządzenia z osobna). Z tego względu, jak przekonamy się w tej książce, ważne jest zapewnienie, by w bazie danych znalazł się kod XHTML, który można później przekształcić na potrzeby różnych urządzeń.

Dostępność

Wielu projektantów stron internetowych przyjęło zwyczaj projektowania witryn internetowych z dokładnością co do piksela. Do projektowania układu stron wykorzystywali tabele o szerokości mierzonej w pikselach, używali czcionek o stałej szerokości i często korzystali z grafiki w celu opracowywania atrakcyjnego układu stron. Takie podejście ułatwiało zapewnienie takiego samego wyglądu strony na różnych komputerach (lub przynajmniej w różnych przeglądarkach pracujących w komputerach biurkowych). Stwarzało to jednak wiele problemów dla osób z wadami wzroku lub mających trudności w poruszaniu myszą.

Wielu użytkowników komputerów mających problemy ze wzrokiem używa do czytania zawartości ekranu urządzeń znanych jako **czytniki ekranu**. Urządzenia te obsługują skomplikowane zestawy skrótów klawiaturowych umożliwiającym użytkownikom posługiwanie się oprogramowaniem bez konieczności widzenia zawartości ekranu. Czytniki ekranu to tylko jeden z wielu typów urządzeń dostępnych dla użytkowników nie w pełni sprawnych, a w wielu krajach obowiązują obecnie przepisy prawne nakładające obowiązek przystosowania wszystkich stron internetowych dla osób niepełnosprawnych.

Oto niektóre problemy, jakie napotykają użytkownicy z wadami wzroku, odwiedzając serwisy WWW:

- Do projektowania układu stron używa się tabel. Często się zdarza, że czytelniki ekranu przetwarzają tabele w taki sposób, iż zrozumienie treści stron staje się trudne — na przykład poszczególne fragmenty strony są czytane w złej kolejności.
- Z powodu zastosowania złej kombinacji kolorów czytanie tekstu staje się trudne.
- Tekst wyświetla się w czcionkach o stałym rozmiarze, co powoduje brak możliwości zwiększenia rozmiaru czcionki w przypadku, gdy zachodzi taka potrzeba.

Problemy te — podkreślę to ponownie — można by rozwiązać, gdyby reguły opisujące sposób prezentacji strony zostały oddzielone od właściwej zawartości dokumentu (oraz gdyby przestrzegano wskazówek dotyczących dostępności). Na przykład użytkownicy posługujący się czytnikami ekranu mogliby usunąć wizualne formatowanie (ponieważ i tak nie widzą strony) i skoncentrować się na treści strony.

W przypadku dostępności, na sposób projektowania i tworzenia stron internetowych wpływa szereg innych problemów, które wykraczają poza problem oddzielenia stylu od zawartości. Dlatego właśnie napisałem dwa rozdziały poświęcone tym problemom, które zamieściłem za rozdziałami poświęconymi XHTML i CSS. Oto kilka problemów, przed jakimi stają użytkownicy, a które będą opisane w tych rozdziałach:

- Podczas obsługi formularzy nie zawsze jest jasne, jakie informacje należy wprowadzić w elemencie sterującym (kontrolce) formularza, na przykład polu tekstowym, albo jaki przełącznik należy zaznaczyć. Z tego względu ważne jest, aby wszystkim elementom sterującym formularzy towarzyszyły etykiety.
- Jeśli strona zawiera duży nagłówek (często z elementami nawigacyjnymi), użytkownik posługujący się czytnikiem ekranu będzie musiał wysłuchać wszystkich tych informacji przed właściwą zawartością strony. Może to być bardzo żmudne w przypadku, gdy informacje te powtarzają się na każdej stronie. Byłoby lepiej, gdyby takie informacje można było pominąć.
- Jeśli na stronie są wykorzystywane obrazy do reprezentowania tekstu, który jest niezbędny do zrozumienia treści witryny, i jeśli dla tych obrazów nie używa się atrybutu `alt` (w celu dostarczenia tekstowej alternatywy osobom niemogącym odczytać treści obrazu), użytkownicy posługujący się czytnikami ekranu nie będą w stanie dowiedzieć się, jaką treść reprezentował obraz.

Nauka tworzenia witryn z wykorzystaniem XHTML i CSS to pierwszy krok na drodze do tworzenia dostępnych witryn WWW, zatem trzeba nauczyć się tych podstaw przed przystąpieniem do poznawania innych problemów projektowych, zwłaszcza dotyczących zagadnień dostępności (opisano je w rozdziałach 6. i 7.).

Jeśli programista tworzy dostępne witryny WWW, uzyskuje przy okazji także inne korzyści (oprócz spełniania wymogów prawnych dotyczących dostępności oraz pomagania osobom niepełnosprawnym):

- Zwiększenie liczby potencjalnych odwiedzających witrynę (a w związku z tym potencjalne zwiększenie przychodów), ze względu na udostępnienie witryny osobom, dla których nie była ona dostępna wcześniej.

- Udostępnienie treści strony osobom korzystającym z niej w różnych sytuacjach. Na przykład, jeśli do strony można uzyskać dostęp za pomocą czytnika ekranu, istnieje również prawdopodobieństwo, że będzie wykorzystana na rozwijającym się rynku przeglądarek głosowych, których można używać w różnych sytuacjach, na przykład podczas prowadzenia samochodu, uprawiania joggingu oraz wtedy, gdy nie można używać rąk do poruszania myszą.
- Tworzenie kodu, który obok komputerów PC będzie dostępny dla wielu innych urządzeń.

Oddzielenie stylu od treści

U źródła większości problemów, które przedstawiono do tej pory, leżał fakt, iż większość tradycyjnych stron HTML zawierała reguły zarządzające sposobem prezentowania informacji. Istnieje zatem proste rozwiązanie. Przekonać autorów, aby zaprzestali mieszania znaczników prezentacyjnych (inaczej mówiąc, stylistycznych) z tymi, które opisują strukturę dokumentu.

Idea **oddzielenia stylu od zawartości** oznaczała powrót do korzeni języka HTML, kiedy zestaw znaczników opisywał jedynie strukturę i semantykę dokumentu, a nie sposób jego prezentacji. Początkowo może się wydawać, że takie proste dokumenty HTML stanowią krok w tył w kierunku dni, kiedy internet był szarym, ponurym miejscem. W rzeczywistości jednak dokumenty tworzone w ten sposób mogą być równie atrakcyjne wizualnie.

Zamiast umieszczania reguł rządzących wyglądem dokumentu w tym samym pliku, który zawiera właściwą zawartość, reguły te umieszcza się w oddzielnym dokumencie nazywanym **arkuszem stylów**. Na przykład można zapisać w arkuszu stylów regułę, która informuje, że wszystkie nagłówki 1. poziomu powinny być pisane ciemnoniebieską czcionką o rozmiarze 5 punktów i kroju Arial.

Takie podejście ma wiele innych zalet. Na przykład, z jednego arkusza stylów może korzystać kilka dokumentów. W związku z tym można utworzyć jeden zbiór reguł używany do opisu stylu wszystkich stron w witrynie. Można również zastosować odwrotne podejście — powiązać każdy dokument z innym arkuszem stylów, tak by ta sama treść wyświetlała się w różny sposób.

Dla Czytelników, którzy w dalszym ciągu nie są przekonani do oddzielenia reguł stylu od zawartości, poniżej zestawilem listę najważniejszych zalet takiego podejścia do projektowania witryn:

- Możliwość uwolnienia treści. Można ją prezentować na różne sposoby dla różnych użytkowników.
- Prostsze dokumenty źródłowe, łatwiejsze do pisania.
- Jeśli dokumenty źródłowe są prostsze, istnieje mniejsze prawdopodobieństwo popełnienia błędów podczas edycji dokumentów.
- Reguły stylu można utworzyć raz, a następnie wykorzystywać dla każdej strony w witrynie. Nie trzeba ich powtarzać na każdej stronie.
- Utrzymanie stylu witryny jest łatwiejsze nie tylko dlatego, że arkusz stylu spełnia rolę szablonu dla wszystkich stron, ale także dlatego, że umożliwia on zmianę czcionki lub koloru dla całej witryny w wyniku zmodyfikowania zaledwie jednego arkusza stylów.

- Aby udostępnić treść strony dla różnych urzędów, wystarczy zapisać nowy arkusz stylów dla każdego nowego urządzenia, nie trzeba pisać od podstaw całej witryny dla każdego urządzenia.
- Po tym, jak przeglądarka automatycznie pobierze arkusz stylów przy pierwszym wejściu na stronę korzystającą z arkusza, kolejne strony wykorzystujące ten sam arkusz stylów ładują się szybciej, ponieważ przeglądarka zapisuje kopię arkusza stylów. Poza tym wszystkie strony mają mniejsze rozmiary, ponieważ nie zawierają reguł stylu.

Jak można zauważyć, oprócz rozwiązania problemów wymienionych wcześniej, oddzielenie stylu od zawartości ma również wiele innych zalet.

„X” w nazwie XHTML

W tym samym czasie, kiedy organizacja W3C zdecydowała o usunięciu znaczników zarządzających stylem z języka HTML, zdecydowano również pójść o krok dalej i zastąpić HTML językiem, o którym z pewnością słyszała większość Czytelników: XML. XML to język używany do pisania języków znaczników (dlatego czasami określa się go jako **metajęzyk znaczników**). Kiedy organizacja W3C wprowadzała te zmiany, język XML zyskiwał szeroką akceptację we wszystkich dziedzinach programowania. Z języka tego można korzystać na dowolnej platformie, ponieważ (podobnie jak w przypadku języka HTML) do przechowywania danych wykorzystuje on zwykły tekst. XML to jedna z najszerzej stosowanych technologii w ostatniej dekadzie.

Przekształcenie HTML w XML miało przygotować język na następną dekadę i na dalsze lata. Z tego względu, w celu zaakcentowania zmian, zamiast opublikować język HTML 5.0, organizacja W3C postanowiła wyróżnić nową rodzinę dokumentów, nazywając język XHTML (podobnie jak firma Microsoft opublikowała Windows XP zamiast Windowsa 2003, a firma Macromedia opublikowała pakiet Dreamweaver MX zamiast Dreamweaver 6).

W języku XML napisano kilka języków, o których z pewnością słyszeli niektórzy Czytelnicy. Należą do nich języki SVG (Scalable Vector Graphics), MathML (język zaprojektowany do tworzenia równań matematycznych), XSLT (Extensible Stylesheet Language Transformations) oraz XML Schemas. W języku XML napisano również setki specjalistycznych języków znaczników.

Utworzenie odmiany języka HTML zgodnej z regułami języka XML ma wiele zalet. Więcej informacji na temat tych zalet zaprezentujemy w rozdziałach 2. i 8. W skrócie można jednak wymienić następujące:

- W języku XML, jak się przekonamy w rozdziale 2., obowiązują ostrzejsze reguły składni w porównaniu z HTML. To z kolei ma kilka innych zalet:
 - Producenci przeglądarek mogą pisać mniej złożone przeglądarki zdolne do obsługi stron XHTML. Takie przeglądarki idealnie nadają się do zastosowania w małych urządzeniach o mniejszej ilości pamięci w porównaniu z komputerami desktop.
 - Z danymi zapisanymi na stronie XHTML można przeprowadzać złożone operacje, przetwarzać je i przekształcać. Oznacza to, że dane nie są, tak jak we wcześniejszych wersjach HTML, wykorzystywane wyłącznie do wizualnej prezentacji.

- Dla języka XML napisano wiele narzędzi i języków. Wszystkie one są dziś dostępne dla dokumentów XHTML. Dotyczy to również takich narzędzi jak XSLT oraz interfejsów SAX (*Simple API for XML*).

W październiku 1999 roku organizacja W3C opublikowała XHTML 1.0 — była to nowa wersja języka HTML zapisana zgodnie z zasadami składni języka XML. Jak dowiemy się w rozdziale 2., istnieją trzy wersje języka XHTML 1.0:

- **Strict XHTML 1.0** — z której dodatkowo usunięto wszystkie stare znaczniki dotyczące stylu.
- **Transitional XHTML 1.0** — która pozwala autorom stron WWW na używanie znaczników opisu stylów z wersji HTML 4.1 z jednoczesnym zaadoptowaniem składni XML. Wersję tę opracowano przede wszystkim do obsługi starszych przeglądarek znanych jako **przeglądarki klasyczne** (ang. *legacy browsers*).
- **Frameset XHTML 1.0** — używana do tworzenia dokumentów z ramkami.

Nie należy się martwić, jeśli brzmi to trochę skomplikowanie. Wszystko wyjaśni się w rozdziale 2. Wersja Strict XHTML 1.0 to po prostu podzbiór wersji Transitional XHTML 1.0, natomiast Frameset XHTML 1.0 zawiera jedynie kilka znaczników do obsługi ramek. Co najważniejsze, każdy z tych elementów i atrybutów powinien być znany Czytelnikom znającym HTML — w końcu XHTML to nic innego, jak najnowsze wcielenie języka HTML.

Historia technologii CSS

Ponieważ w dokumentach napisanych w wersji Strict XHTML 1.0 nie było znaczników opisujących prezentację, by dokumenty te stały się atrakcyjne, trzeba było powiązać je z arkuszem stylów zarządzającym sposobem prezentacji dokumentów.

Organizacja W3C na długo przed opublikowaniem XHTML utworzyła język arkuszy stylów, który idealnie nadawał się do zastosowania na stronach WWW. Wielu autorów stron HTML korzystało z **kaskadowych arkuszy stylów** (*Cascading Style Sheets* — CSS) do zarządzania podstawowymi aspektami stylu dokumentów, takimi jak czcionki lub kolory tła.

Specyfikację CSS1 opublikowano w grudniu 1996 roku, natomiast CSS2, będącą rozszerzeniem CSS1 — w maju 1998 roku. Ponieważ projektanci WWW już wcześniej używali CSS do zarządzania wyglądem stron WWW oraz z powodu zalet tej technologii, był to oczywisty wybór dla języka XHTML.

Jednym z elementów, które sprawiają, że technologia CSS jest stosunkowo łatwa do przyswojenia, jest fakt, że nazwy tzw. **właściwości** zarządzających sposobem wyświetlania elementów są bardzo podobne do nazw atrybutów używanych w HTML. CSS jest jednak również bardzo rozbudowanym językiem, który pozwala na znacznie więcej niż to, co można osiągnąć za pomocą zestawu znaczników opisujących styl języka HTML. Język CSS oferuje, między innymi, następujące własności:

- bardzo szczegółową kontrolę nad sposobem prezentacji strony;
- kontrolę nad układem dokumentu bez używania tabel (więcej informacji na ten temat można znaleźć w rozdziale 5.);
- właściwości pozwalające na prezentację dokumentów strona po stronie (większość osób używa terminu strona do wydruku);
- właściwości dźwiękowej wersji dokumentów — mogą je wykorzystać użytkownicy z wadami wzroku lub osoby będące w ruchu, które nie mogą patrzeć na ekran.

Sposoby tworzenia atrakcyjnych układów stron z wykorzystaniem CSS pokazano w rozdziałach 3., 4. i 5.

Korzystanie z arkuszy stylów

W celu rozwiązania problemów związanych z przeglądaniem witryny internetowej w różnych przeglądarkach oraz w różnych urządzeniach, a także dlatego, by witryny stały się bardziej dostępne, najlepszym rozwiązaniem jest oddzielenie znaczników definiujących strukturę dokumentu od reguł opisujących sposób jego wyświetlania.

Z oddzielenia projektu od zawartości płyną następujące wnioski:

- Trzeba zrewidować swoje umiejętności pisania dokumentów HTML od podstaw i zapomnieć o znacznikach do definiowania stylów dodanych w późniejszych wersjach HTML. Główną treść strony można teraz napisać w wersji Strict języka XHTML, zawierającej jedynie znaczniki służące do opisywania struktury i semantyki dokumentu.
- Reguły rządzące sposobem wyświetlania strony należy zapisać w oddzielnym dokumencie, do którego odwołuje się strona XHTML. Jest to arkusz stylów napisany w języku CSS.

Bez bardziej szczegółowego omówienia wykorzystywanych języków trudno dokładnie przeanalizować przykłady. Pomimo tego w poniższym przykładzie zaprezentowano wersję dokumentu, którą po raz pierwszy spotkaliśmy w pliku *r01_list01.html*, zapisaną w XHTML. Dalej zamieszczono arkusz stylów CSS zarządzający prezentacją strony.

Jak można zauważyć w tym przykładzie, z dokumentu usunięto nawet tabelę (używaną do definiowania układu strony) i zastąpiono ją elementami `<div>` pozwalającymi na grupowanie elementów, do których można zastosować style (*r01_list03.html*):

```
<?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">
  <head>
    <title>Wpływ internetu na teorie psychologiczne na temat jednostki</title>
    <link rel="stylesheet" type="text/css" href="r01_list03.css" />
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2" />
  </head>
  <body>
```

```

<div class="strona">
  <div class="naglowek">
    <h1>Wpływ internetu na teorie psychologiczne na temat jednostki</h1>
  </div>
  <div class="tresc">
    <h2>Streszczenie</h2>
    <p>W artykule omówiono cele, w jakich użytkownicy internetu przyjmują
    ↳ określoną osobowość online oraz wpływ tych zachowań na teorie
    ↳ psychoanalityczne dotyczące jednostki i tożsamości człowieka.</p>
    <p>0 ile psychologowie od dawna sugerowali, że nasze pojęcie jednostki
    ↳ powinno odzwierciedlać pojedynczą, jednorodną i racjonalną osobowość,
    ↳ wiele osób przyjmuje osobowość online, która znacznie się różni
    ↳ od osobowości w życiu codziennym.</p>
    ...
  </div>
</div>
</body>
</html>

```

Poniżej zamieszczono arkusz stylów towarzyszący temu dokumentowi (*r01_list03.css*). Nie należy się przejmować, jeśli ten dokument na pierwszy rzut oka wygląda na skomplikowany. Kiedy w rozdziale 3. zaczniemy omawianie struktur języka, przekonamy się, że wiele konstrukcji języka CSS odpowiada własnościom, których nauczyliśmy się przy okazji poznawania atrybutów HTML:

```

/* Arkusz CSS dla dokumentu r01_list03.html */

body {
  background-color:#000;
  font-family: arial, verdana, sans-serif;}

div.strona {
  width:650px;
  border-style:solid; border-width:1px; border-color:#666;}
div.naglowek {background-color:#999; padding:10px;}
div.tresc {background-color:#efefef; padding:10px;}

h1 {
  font-size:22pt;
  color:#006;}
h2 {
  font-size:18pt;
  color:#006;}
p {
  font-size:14pt;
  color:#000;}

```

Choć pisanie arkusza stylów podobnego do tego, który pokazano powyżej, tylko dla jednego dokumentu, może się wydawać kłopotliwe, w przypadku witryn złożonych z wielu stron o podobnym stylu taka technika może przynieść znaczne oszczędności czasu. W takim przypadku nie trzeba bowiem dodawać reguł prezentacji do każdego dokumentu, który z tych reguł korzysta.

Liczne zalety oddzielenia stylu od zawartości można zaobserwować dokładniej w dalszej części tej książki. W przypadku pojawienia się w przyszłości nowych urządzeń, trzeba jedynie rozszerzyć rozwiązanie — nie trzeba tworzyć nowego języka dla każdego nowego urządzenia.

Co więcej, w przypadku tworzenia stron WWW zgodnie ze standardami języków XHTML i CSS, do ich przeglądania można użyć starszych przeglądarek, napisanych jeszcze przed pojawieniem się tych standardów.

Wprowadzenie do przykładowej witryny

Podczas lektury niniejszej książki Czytelnicy będą mieli do czynienia z jedną przykładową witryną. Witryna należy do fikcyjnej firmy sprzedającej towary „promocyjne” lub reklamowe, takie jak długopisy, torby, notatniki oraz zabawki antystresowe z nazwami firmy lub nadrukowanym jej logo. W następnym podrozdziale zaprezentowano tę witrynę, włącznie z opisem jej organizacji i kodem. W kolejnych rozdziałach będziemy zajmowali się aktualizowaniem tej witryny.

Fikcyjna firma, dla której będziemy tworzyli serwis internetowy, nazywa się Fajna Promocja. Główną część tego serwisu stanowią katalog produktów i cennik. Firma nie przyjmuje zamówień online, ponieważ przed przyjęciem zamówienia musi otrzymać logo klienta (obok tekstu, który ma się pojawić na towarze). To często wymaga pracy z projektantami grafiki, których zadaniem jest dostarczenie efektów swojej pracy we właściwym formacie (poza tym dzięki temu dział handlowy ma bezpośredni kontakt z klientem). Po otrzymaniu projektu grafiki, klient musi zatwierdzić projekt towaru przed przystąpieniem do produkcji.

Pomimo tego, że witrynę napisano w HTML, pokazany przykład jest przydatny także dla osób posługujących się językami programowania działającymi po stronie serwera, takimi jak ASP/JSP/PHP. Wszystkie te technologie przesyłają bowiem kod HTML do przeglądarki, zatem przykład ten uczy, w jaki sposób pisać kod, który ma być przesłany do klienta.

Dla celów niniejszej książki wyobraźmy sobie, że witrynę Fajna Promocja napisano pod koniec lat dziewięćdziesiątych. Pamiętajmy, że tworzymy witrynę, jaką chcielibyśmy spotkać dziś, a witryna w postaci wyjściowej z całą pewnością wykorzystuje techniki powszechnie używane w tamtym czasie. Przy okazji pracy nad stroną, warto również poświęcić trochę czasu na dopracowanie stylu.

Aby przetestować stronę, należy pobrać jej kompletny kod (zarówno wyjściowej, jak i końcowej wersji serwisu) z serwera FTP wydawnictwa Helion. W tym samym miejscu jest również dostępny kod pozostałych przykładów użytych w tej książce. Można też pobrać kompletny kod zarówno wyjściowej, jak i końcowej wersji serwisu (a także kod pozostałych przykładów użytych w tej książce) z serwera FTP wydawnictwa Helion

Stronę główną serwisu Fajna Promocja pokazano na rysunku 1.4.

Na rysunku 1.4 można zobaczyć, że strona główna zawiera nagłówek oraz kilka elementów nawigacyjnych umieszczonych poniżej logo. Pozostała część strony jest podzielona na trzy kolumny. Taki układ strony jest bardzo często spotykany.

Rysunek 1.4



Jak wspomniałem wcześniej, głównym celem witryny jest zaprezentowanie katalogu produktów i cennika. Poszczególne elementy katalogu znajdują się w lewej kolumnie — taki sposób poruszania się po katalogu można określić jako **nawigacja według kategorii**. Każde łącze z kolumny po lewej stronie prowadzi do strony zawierającej listę produktów. Są na niej informacje sumaryczne dla wszystkich towarów w wybranej kategorii. Przykład strony z listą produktów pokazano na rysunku 1.5. Można na nim zobaczyć listę produktów w kategorii *Torby*.

Strona ta ma identyczny trójkolumnowy układ, jak strona główna. Strony z listą towarów dla poszczególnych kategorii są również identyczne pod każdym względem, z wyjątkiem produktów, które zawierają.

Kiedy użytkownik kliknie obraz lub tytuł wybranej pozycji, przechodzi do indywidualnej strony produktu. Przykład jednej ze stron produktu pokazano na rysunku 1.6. Jak można zauważyć na podstawie tego zrzutu, strony z produktami mają układ dwukolumnowy. Usunięcie trzeciej kolumny daje więcej miejsca do wyświetlenia szczegółowych informacji o produkcie.

U góry każdej strony oraz u dołu głównej kolumny można zauważyć dodatkowe łącza nawigacyjne do stron zawierających takie dane, jak sposób kontaktowania się z firmą, składania zamówień, informacje o firmie itp. Wszystkie strony mają taką samą strukturę, jak strona główna — zastosowano dla nich układ trójkolumnowy. Jedynym wyjątkiem jest strona z produktami, które najlepiej się sprzedają, dla której wykorzystano tę samą strukturę, jaką zastosowano dla stron z listami produktów.

Rysunek 1.5



Rysunek 1.6



Podsumowując, serwis zawiera trzy główne typy stron:

- stronę główną;
- strony z listami produktów;
- strony ze szczegółowymi danymi produktów.

Każda strona serwisu ma jeden spośród wymienionych trzech układów, które opisano bardziej szczegółowo w poniższych punktach. Czytelnicy nie muszą dokładnie studiować poszczególnych wierszy kodu. Warto jednak przyjrzeć się ogólnej budowie witryny, ponieważ będziemy do niej często wracać podczas lektury dalszej części książki. W tym rozdziale zamieszczono również kod wyjściowej wersji witryny, tak by można się było do niego odnieść podczas wprowadzania kolejnych aktualizacji.

Strona główna

Podobnie jak w przypadku każdego serwisu, strona główna jest pierwszą, jaką widzi odwiedzający w momencie wejścia na stronę. Jej zadaniem jest poinformowanie użytkowników o tym, co można znaleźć w witrynie. Jest to również baza, od której użytkownicy mogą rozpocząć poruszanie się po serwisie.

Na rysunku 1.7 pokazano linie nakreślone na zrzucie ekranu strony głównej, które pokazują lokalizację tabel zarządzających układem strony.

Rysunek 1.7



Cała strona jest zapisana wewnątrz tabeli zawierającej jeden wiersz. Tabela ustala szerokość strony, a jej obramowanie tworzy ramkę wokół strony. Wewnątrz tej tabeli są trzy inne tabele.

Na górze można zobaczyć pierwszą tabelę zawierającą nagłówek z logo firmy i celem serwisu. Poniżej znajduje się druga tabela zawierająca łącza do indywidualnych stron z innymi informacjami, na przykład sposobami zamawiania towarów, danymi kontaktowymi oraz towarami, które sprzedają się najlepiej. Trzecia tabela zawiera główną treść strony. W tej trzeciej tabeli są trzy kolumny:

- kolumna 1. zawiera elementy nawigacyjne poszczególnych sekcji (oraz grafikę stempla gwarancji dostawy);
- kolumna 2. zawiera główną treść strony — informacje, po które użytkownik wszedł na stronę;
- kolumna 3. zawiera dodatkowe informacje, na przykład sposób składania zamówień.

Poniżej zamieszczono kod strony głównej (*index.html*), który rozpoczyna się tak, jak można się spodziewać:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
⤵ "http://www.w3.org/TR/1999/REC-html401-19991224/loose.dtd">
<html>
<head>
  <title>Fajna Promocja - gadzety reklamowe dla firm</title>
```

Poniższy skrypt został wygenerowany przez program Macromedia Dreamweaver po dodaniu obrazu zmieniającego się po wskazaniu go myszą (ang. *rollover image*).

```
<script language="JavaScript" type="text/JavaScript">
<!--
function MM_swapImgRestore() { //v3.0
  var i,x,a=document.MM_sr; for(i=0;a&&i<a.length&&(x=a[i])&&x.oSrc;i++) x.src=x.oSrc;
}

function MM_preloadImages() { //v3.0
  var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();
    var i,j=d.MM_p.length,a=MM_preloadImages.arguments; for(i=0; i<a.length; i++)
      if (a[i].indexOf("#")!=0){ d.MM_p[j]=new Image; d.MM_p[j++].src=a[i];}}
}

function MM_findObj(n, d) { //v4.01
  var p,i,x;  if(!d) d=document; if((p=n.indexOf("?"))>0&&parent.frames.length) {
    d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
  if(!(x=d[n])&&d.all) x=d.all[n];
  for (i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
  for(i=0;!x&&d.layers&&i<d.layers.length;i++) x=MM_findObj(n,d.layers[i].document);
  if(!x && d.getElementById) x=d.getElementById(n); return x;
}

function MM_swapImage() { //v3.0
  var i,j=0,x,a=MM_swapImage.arguments; document.MM_sr=new Array;
  for(i=0;i<(a.length-2);i+=3)
```

```

    if ((x=MM_findObj(a[i]))!=null){document.MM_sr[j++]=x; if(!x.oSrc)
    ↪ x.oSrc=x.src; x.src=a[i+2];}
  }
  //-->
</script>
</head>

```

Wewnątrz elementu <body> jest kilka atrybutów definiujących styl, takich jak bgcolor i link. Atrybut onLoad został wygenerowany przez program Dreamweaver (podobnie jak skrypt) i służy do ładowania obrazów *rollover*:

```

<body bgcolor="#fff" alink="#00c" vlink="#036" link="#06c"
onLoad="MM_preloadImages('images/interface/nav_order_on.gif',
↪ 'images/interface/nav_bestSellers_on.gif','images/interface/nav_custom.gif',
↪ 'images/interface/nav_contactUs_on.gif','images/interface/nav_pens_on.gif',
↪ 'images/interface/nav_pads_on.gif','images/interface/nav_bags.gif',
↪ 'images/interface/nav_mugs_on.gif','images/interface/nav_stress_on.gif')">

```

Dalsza część kodu jest odpowiedzialna za wyświetlanie głównej zawartości strony. Umieszczono ją wewnątrz elementu <center> po to, by znalazła się pośrodku strony. Cała strona mieści się wewnątrz jednego elementu <table> o szerokości 800 pikseli. Dzięki temu szerokość strony jest stała. Strona jest otoczona ramką o szerokości jednego piksela. Główna tabela zawiera zaledwie jedną komórkę, w której znajduje się pozostała część strony.

Wewnątrz pojedynczej komórki zawierającej stronę są trzy inne tabele: jedna zawierająca nagłówek strony, druga z elementami nawigacyjnymi i trzecia zawierająca główną treść strony. Najpierw jest kod tabeli zawierającej nagłówek strony:

```

<center>
<table border="1" cellpadding="0" cellspacing="0" width="800"><tr><td>
<!-- nagłówek -->
<table border="0" cellpadding="5" cellspacing="0" width="800">
  <tr>
    <td bgcolor="#006"><font face="Arial, Helvetica, sans-serif" size="1"
    ↪ color="#fff"><b>&nbsp;Gadżety reklamowe, długopisy, notatniki, torby, kubki,
    ↪ zabawki, prezenty reklamowe i wiele innych...</b></font></td>
  </tr>
  <tr>
    <td></td>
  </tr>
</table>
</td>
</tr>
</table>

```

W drugiej tabeli są elementy nawigacji. Jak można zauważyć, w tabeli z elementami nawigacji jest kod do utworzenia obrazów rollover dla każdej pozycji w górnym pasku nawigacji:

```

<!-- nawigacja -->
<table border="0" cellpadding="0" cellspacing="1" width="800" bgcolor="#69f">
  <tr>
    <td width="150" bgcolor="#006"></td>
    <td bgcolor="#006"></td>
    <td bgcolor="#006"><a href="order.html" onMouseOut="MM_swapImgRestore()"
    ↪ onMouseOver="MM_swapImage('JakZamawiac','images/interface/
    ↪ nav_order_on.gif',1)"></a></td>
  </tr>
</table>

```

```
  |
```

W trzeciej i zarazem ostatniej tabeli znajduje się główna treść strony. Tabela zawiera jeden wiersz podzielony na trzy komórki — każdej kolumnie strony odpowiada pojedyncza komórka.

W pierwszej kolumnie znajdują się menu poruszania się po produktach zawierające obrazki rollover, podobne do tych znajdujących się w górnym pasku menu. Aby poprawić czytelność kodu, dodałem odstęp pomiędzy poszczególnymi elementami. Jednak w kodzie dostępnym do pobrania nie ma spacji pomiędzy łączami nawigacyjnymi a obrazami wypełniaczami (ang. *spacer images*) (w innym przypadku w przeglądarce Internet Explorer byłyby widoczne przerwy).

```

<!-- strona główna -->
<table border="0" cellpadding="0" cellspacing="0" width="800">
|  |  |
| --- | --- |
| | |

```

```


<a href="mugs.html" onMouseOut="MM_swapImgRestore()"
  onMouseOver="MM_swapImage('Kubki','',
    'images/interface/nav_mugs.gif',1)">
  
</a>

<a href="products/keyrings/index.html" onMouseOut="MM_swapImgRestore()"
  onMouseOver="MM_swapImage('Breloczki','',
    'images/interface/nav_keyrings_on.gif',1)">
  
</a>

<a href="products/stressbusters/index.html"
  onMouseOut="MM_swapImgRestore()"
  onMouseOver="MM_swapImage('ZabawkiAntystresowe','',
    'images/interface/nav_stress_on.gif',1)">
  
</a>

<center>
  
</center>
</td>

```

Druga kolumna zawiera główny komunikat strony. Pierwszym elementem, który można zobaczyć w tej kolumnie, jest wprowadzający akapit informujący użytkowników o tym, czym zajmuje się firma. Akapit ten jest zapisany w osobnej tabeli, tak aby tekst wyświetlał się w oddzielnej ramce:

```

<td width="470" bgcolor="#fff" valign="top">
  <center>
    
    <table border="1" cellpadding="10" cellspacing="0"
      bgcolor="#efefef" width="400">
      <tr>
        <td>
          <font face="Arial, Helvetica, sans-serif" size="2" color="#006">
            Witamy w serwisie firmy Fajna Promocja specjalizującej się
            ↳w produkcji reklamowych i promocyjnych. Nasz obszerny zbiór toreb
            ↳promocyjnych, kubków, notatników i długopisów ułatwia klientom nazwy
            ↳Państwa firmy.<br><br>
            Wszystkie nasze gadzety promocyjne można spersonalizować poprzez
            ↳umieszczenie logonazwy i adresu internetowego. Firma Fajna Promocja
            ↳może nawetpomóc w zaprojektowaniu gadżetów promocyjnych.</font>
          </td>
        </tr>
      </table>
    </center>
  <br>

```

Poniżej wstępnych akapitów na stronie głównej znajduje się prosta tabela prezentująca kilka typowych towarów. Zwróćmy uwagę na wcięcie napisu „Typowe produkty”, które uzyskano za pomocą przezroczystego pliku GIF o szerokości jednego piksela. Obrazki przedstawiające typowe towary zawierają odnośniki do stron prezentujących szczegółowe informacje dotyczące pozycji z katalogu.

```


<center>
  <table border="1" cellpadding="5" cellspacing="0"
    ↪ bgcolor="#d6d6d6" width="400">
    <tr>
      <td width="200" bgcolor="#fff" align="center">
        <a href="products/mugs/mug2.html">
          
        </a>
      </td>
      <td width="200" bgcolor="#fff" align="center">
        <a href="products/pens/pen2.html">
          
        </a>
      </td>
    </tr>
    <tr>
      <td width="200" align="center">
        <font face="Arial, Helvetica, sans-serif" size="2" color="69f">
          <b>Kubki już od 3 PLN</b>
        </font>
      </td>
      <td width="200" align="center">
        <font face="Arial, Helvetica, sans-serif" size="2" color="69f">
          <b>Długopisy od 1 PLN</b>
        </font>
      </td>
    </tr>
    <tr>
      <td width="200" bgcolor="#fff" align="center">
        <a href="products/bags/bag4.html">
          
        </a>
      </td>
      <td width="200" bgcolor="#fff" align="center">
        <a href="products/pads/pad3.html">
          
        </a>
      </td>
    </tr>
    <tr>
      <td width="200" align="center">
        <font face="Arial, Helvetica, sans-serif" size="2" color="69f">
          <b>Plecaki od 9,99 PLN</b>
        </font>
      </td>
      <td width="200" align="center">
        <font face="Arial, Helvetica, sans-serif" size="2" color="69f">
          <b>Kołonotatniki A4 od 3,50 PLN</b>
        </font>
      </td>
    </tr>
  </table>
</center>
```

```

        </td>
      </tr>
    </table>

```

U dołu środkowej kolumny znajdują się łącza stopki:

```

<font face="Arial, Helvetica, sans-serif" size="1" color="#00c">
  <br><br>
  <a href="about.html">O nas</a> |
  <a href="contact.html">Kontakt</a> |
  <a href="terms.html">Warunki korzystania z usług</a> |
  &copy; 2004 Fajna Promocja
<br><br>
</font>
</center>
</td>

```

Trzecia i zarazem ostatnia kolumna zawiera informacje dotyczące składania zamówień, które można znaleźć w prawej części strony głównej. W tym fragmencie strony skorzystano z tabel zagnieżdżonych do utworzenia ramek wokół tekstu:

```

<td width="180" bgcolor="#efefef" valign="top">
  
  <center>
     <br>
    
    <table border="1" cellpadding="0" cellspacing="0"
      width="155">
      <tr>
        <td>
          
        </td>
      </tr>
      <tr>
        <td bgcolor="#fff">
          <table border="0" cellpadding="5" cellspacing="0">
            <tr>
              <td>
                <font face="Arial, sans-serif" size="2"
                  color="#904c2d">
                  1) Wybierz towar z naszego obszernego katalogu.<br><br>
                  2) Wyślij swoje logo lub tekst, który ma się znaleźć na
                     ↳gadżetach.<br><br>
                  3) Po opracowaniu projektu prześlemy Ci zdjęcie gadżetu.
                     ↳<br><br>
                  4) Potwierdź zamówienie, a dostarczymy towar w ciągu
                     ↳21 dni.<br>
                </font>
              </td>
            </tr>
          </table>
        </td>
      </tr>
    </table>
  </center>
  <br><br>
</td>

```

```

        </td>
      </tr>
    </table>
  </td> </tr> </table>
</center>
</body>
</html>

```

Jak można zauważyć podczas przeglądania witryny, strony dostępne za pośrednictwem górnych i dolnych elementów nawigacyjnych mają bardzo podobną strukturę — główna treść znajduje się w centralnej części, natomiast dalej układ jest identyczny i składa się z trzech kolumn. Wyjątkiem jest strona „hitów sprzedaży”, której struktura jest analogiczna do stron z listą produktów opisanych w dalszej części tego rozdziału.

Strony z listami produktów

Każda z sześciu kategorii towarów ma własny folder w strukturze katalogów witryny. W każdym z tych folderów znajduje się plik *index.html* zawierający listę wszystkich produktów z danej kategorii, stąd nazwa „strona z listą produktów”. Strona ma układ trójkolumnowy, podobnie jak strona główna. W rzeczywistości jedyną realną różnicą pomiędzy tą stroną a stroną główną jest środkowa kolumna tabeli tworząca zasadniczą część strony.

Przyjrzyjmy się zawartości strony umieszczonej wewnątrz znaczników `<body>`. Ponieważ logo i nagłówki strony, a także górne menu nawigacyjne oraz łącza nawigacyjne z lewej strony są niemal identyczne jak te na stronie głównej, nie będę ich szczegółowo omawiał. Są tylko dwie zasadnicze różnice:

- Polecenie menu wskazujące na to, że użytkownik wybrał sekcję Torby, jest wyróżnione.
- Wszystkie adresy URL do obrazów oraz do innych stron poprzedza przedrostek `.././`, co wskazuje na użycie względnych adresów URL powyżej nadrzędnego folderu tego pliku.

```

<center>
<table border="1" cellpadding="0" cellspacing="0" width="800">
  <tr><td>

```

```

    <table border="0" cellpadding="5" cellspacing="0" width="800">

```

```

    <!-- LOGO I NAGŁÓWEK STRONY -->

```

```

  </table>

```

```

  <table border="0" cellpadding="0" cellspacing="1" width="800" bgcolor="#69f">

```

```

  <!-- GÓRNE MENU NAWIGACYJNE -->

```

```

  </table>

```

```

  <table border="0" cellpadding="0" cellspacing="0" width="800">

```

```

    <tr>

```

```

      <td width="150" bgcolor="#d9ecff" valign="top">

```

```

    <!-- MENU NAWIGACYJNE Z LEWEJ STRONY-->

```

```

      </td>

```

Rzeczywiste zmiany dotyczą środkowej kolumny w trzeciej tabeli (zawierającej główną treść strony). Po pierwsze, znajduje się w niej kilka przezroczystych jednopikselowych plików GIF, które pozycjonują nagłówek w pożądanym miejscu. Nagłówek wskazuje kategorię, jaką wybrał użytkownik — w tym przypadku Torby.

```
<td width="470" bgcolor="#fff" valign="top">
  <center>
    
  </center><br>
  
  <font face="Arial, Helvetica, sans-serif" size="5" color="#006">
    <b>Torby</b>
  </font><br>
```

Główną częścią strony z listą produktów jest tabela, która wyświetla różne produkty dostępne w wybranej kategorii. Każdy produkt zajmuje jeden wiersz. Zdjęcie produktu jest w komórce po lewej stronie, natomiast jego opis po prawej. Zarówno zdjęcie, jak i tytuł są odnośnikami do strony zawierającej szczegółowe informacje o produkcie:

```
<center>
  <table border="1" cellpadding="5" cellspacing="0"
    ↪ bgcolor="#d6d6d6" width="400">
    <tr>
      <td width="200" bgcolor="#fff" align="center">
        <a href="bag1.html">
          
        </a>
      </td>
      <td width="200" bgcolor="#fff" valign="top">
        <font face="Arial, Helvetica, sans-serif" size="2" color="#666">
          <b><a href="bag1.html">Torby konferencyjne typ #1</a></b><br>
            Najlepiej sprzedające się, niedrogie torby z dopinamy
            ↪askiem i wewnętrznymi kieszeniami.<br>
          <b>Już od 7,50 PLN </b> przy minimalnej ilości 150 sztuk.
        </font>
      </td>
    </tr>
    <tr>
      <td width="200" bgcolor="#fff" align="center">
        <a href="bag2.html">
          
        </a>
      </td>
      <td width="200" bgcolor="#fff" valign="top">
        <font face="Arial, Helvetica, sans-serif" size="2" color="#666">
          <b><a href="bag2.html">Torby konferencyjne typ #2</a></b><br>
            Wzmocniona konstrukcja poprawiająca trwałość i kieszeń z przodu
            zapinana na zamek błyskawiczny.<br>
          <b>Już od 8,50 PLN </b> przy minimalnej ilości 50 sztuk
        </font>
      </td>
    </tr>
    <tr>
      <td width="200" bgcolor="#fff" align="center">
        <a href="bag3.html">
          
```

```

        </a>
    </td>
    <td width="200" bgcolor="#fff" valign="top">
        <font face="Arial, Helvetica, sans-serif" size="2" color="#666">
            <b><a href="bag3.html">Torby konferencyjne typ #3</a></b><br>
            Wysokiej jakości torby konferencyjne z poliestru
            ↳ z wieloma za doskonałą cenę.<br>
            <b>Już od 9,99 PLN </b> przy minimalnej ilości 100 sztuk
        </font>
    </td>
</tr>
<tr>
    <td width="200" bgcolor="#fff" align="center">
        <a href="bag4.html">
            
        </a>
    </td>
    <td width="200" bgcolor="#fff" valign="top">
        <font face="Arial, Helvetica, sans-serif" size="2" color="#666">
            <b><a href="bag4.html">Aktówki biznesowe</a></b><br>
            Z polyesteru 600D. Dostępne w kolorach granatowym lub niebieskim.
            Idealne dla menedżerów w podróży służbowej<br>
            <b>Już od 8,99 PLN </b> przy minimalnej ilości 100 sztuk
        </font>
    </td>
</tr>
</table>

```

Tak, jak w przypadku większości stron, w stopce tej kolumny jest kilka łączy:

```

<font face="Arial, Helvetica, sans-serif" size="1" color="#00c">
<!-- ŁĄCZA STOPKI STRONY -->
</font>
</center>
</td>

```

Pozostała część strony, włącznie z kolumną po prawej stronie, ma dokładnie taką samą zawartość, jak strona główna. Z tego względu nie będę jej powtarzał w tym miejscu.

```

    <td width="180" bgcolor="#efefef" valign="top">
    <!-- TUTAJ POWINIEN ZNALEŹĆ SIĘ KOD PRAWEJ KOLUMNY -->
    </td>
</tr>
</table>

</td> </tr> </table>
</center>

```

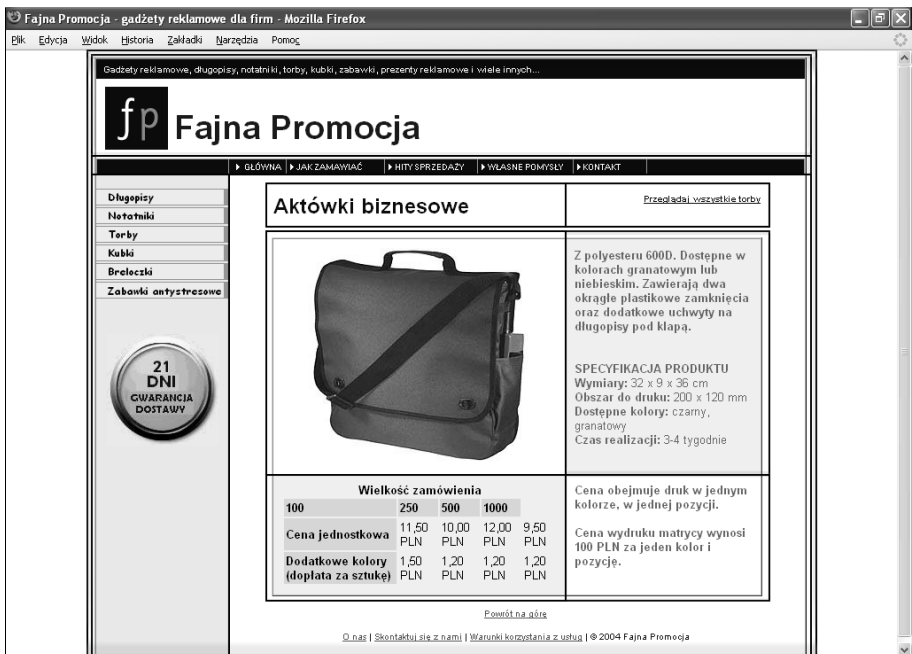
Jak pamiętamy, produkty podzielono na sześć kategorii — długopisy, torby, kubki, notatniki, breloczki i zabawki antystresowe. Każda kategoria ma własny folder w strukturze katalogów. W folderze każdej kategorii jest plik *index.html*, zawierający listę produktów dla wybranej kategorii.

Strony ze szczegółowymi danymi o produktach

Kiedy użytkownik kliknie dowolną pozycję na liście produktów (lub zdjęcie typowego produktu na stronie głównej), wyświetli się indywidualna strona produktu zawierająca szczegółowe informacje na jego temat, włącznie z cenami.

Ponieważ na stronach tego typu trzeba zmieścić więcej informacji niż na innych, główna część strony zawiera tylko dwie kolumny. Na rysunku 1.8 pokazano stronę szczegółowych informacji o aktówkach biznesowych z nakreślonymi liniami pokazującymi komórki tabeli użytej do zarządzania układem strony.

Rysunek 1.8



Tak jak w przypadku stron omawianych wcześniej, cała strona jest zapisana wewnątrz tabeli wyznaczającej szerokość strony. Ramka wokół zewnętrznej krawędzi tabeli wyznacza obramowanie strony. Wewnątrz tej tabeli są trzy inne tabele. Pierwsza zawiera tytuł strony, druga łączy nawigacyjne, natomiast trzecia — główną treść strony. Trzecia tabela zawiera tylko dwie komórki: jedną dla lewostronnych łączy nawigacyjnych, drugą dla szczegółowych informacji o produkcie.

Przyjrzyjmy się teraz kodowi strony (*bag4.html*). Początek jest taki sam jak strony głównej i stron z listą produktów (poza tym, że adresy URL rozpoczynają się od przedrostka *.././*), zatem omawianie rozpoczniemy od głównej treści strony. Podobnie jak w przypadku pozostałych stron, cała zawartość strony jest zapisana w tabeli składającej się z pojedynczej komórki. Wewnątrz tej komórki są tabele dla tytułu strony i łączy nawigacyjnych.

```
<center>
```

```
<table border="1" cellpadding="0" cellspacing="0" width="800"><tr><td>
```

```
<table border="0" cellpadding="5" cellspacing="0" width="800">
  <!-- TYTUŁ STRONY -->
</table>

<table border="0" cellpadding="0" cellspacing="1" width="800" bgcolor="#69F">
  <!-- ŁĄCZA NAWIGACYJNE -->
</table>
```

Dalej jest trzecia tabela. Jej zadaniem jest utworzenie kolumn w głównej części strony. Podobnie jak w przypadku pierwszej kolumny (zapisanej w pierwszej komórce tabeli), są w niej łącza nawigacyjne — nie ma potrzeby, aby omawiać je jeszcze raz.

Druga kolumna jest o wiele szersza — ma 650 pikseli i zawiera znacznie więcej miejsca pozwalającego na wyświetlanie szczegółowych informacji na temat wybranego produktu:

```
<!-- strona główna -->
<table border="0" cellpadding="0" cellspacing="0" width="800">
  <tr>
    <td width="150" bgcolor="#d9ecff" valign="top">
      <!-- LEWOSTRONNE ŁĄCZA NAWIGACYJNE -->
    </td>
    <td width="650" bgcolor="#fff" valign="top">
```

Dalsza część to kod komórki reprezentującej główną część strony z informacjami o produkcie. W tej komórce są dwie tabele — pierwsza zawiera tytuł produktu, natomiast druga — łącza do poszczególnych typów toreb.

```
<br>
<center>
<table border="0" cellpadding="0" cellspacing="0" width="550">
  <tr>
    <td>
      <font face="Arial, Helvetica, sans-serif" size="5" color="#006">
        <b>Aktówki biznesowe</b>
      </font>
    </td>
    <td align="right" valign="top">
      <a href="/products/bags/index.html">
        <font face="Arial, Helvetica, sans-serif" size="1" color="#006">
          <b>Przeglądaj wszystkie torby</b>
        </font>
      </a>
    </td>
  </tr>
</table><br>
```

Choć na pierwszy rzut oka wydaje się, że tak nie jest, druga tabela zawiera zaledwie jedną komórkę. Wynika to z faktu, iż jest to tabela-opakowanie, która wewnątrz zawiera inną tabelę. Ta jednokomórkowa tabela służy do utworzenia ramki wokół zapisanych w niej informacji i składa się z dwóch wierszy i dwóch kolumn:

- Wiersz 1., kolumna 1. zawiera zdjęcie torby.
- Wiersz 1., kolumna 2. zawiera opis torby.
- Wiersz 2., kolumna 1. zawiera ceny.
- Wiersz 2., kolumna 2. zawiera informacje dotyczące wydruku.

Rozpocznijmy od pierwszego wiersza zawierającego zdjęcie i opis torby. Jego zawartość jest dość oczywista:

```
<table border="1" cellpadding="0"
      cellspacing="0" width="550"><tr><td>

      <table border="0" cellpadding="10" cellspacing="0" bgcolor="#fff"
        width="550">
        <tr>
          <td width="350" bgcolor="#fff" align="center" valign="top">
            
          </td>
          <td width="200" bgcolor="#efefef" valign="top">
            <font face="Arial, Helvetica, sans-serif" size="2" color="#666">
              <b>Wyprodukowane z wysokiej jakości polyesteru 600D. Dostępne
                ↳ w kolorach granatowym lub niebieskim.
                Zawierają dwa okrągłe plastikowe zamknięcia oraz dodatkowe
                ↳ uchwyty na długopisy
                pod klapą.</b>
              <br><br><br>
              <b>SPECYFIKACJA PRODUKTU</b><br>
              <b>Wymiary:</b> 32 x 9 x 36 cm<br>
              <b>Obszar do druku:</b> 200 x 120 mm<br>
              <b>Dostępne kolory:</b> czarny, granatowy<br>
              <b>Czas realizacji:</b> 3-4 tygodnie <br><br>
            </font>
          </td>
        </tr>
      </td>
    </tr>
```

Kod drugiego wiersza zamieszczono poniżej. W pierwszej kolumnie drugiego wiersza znajdują się zagnieżdżone tabele zarządzające cenami produktu w zależności od wielkości zamówienia. W drugiej kolumnie wiersza są informacje dotyczące wydruku.

```
<tr>
  <td bgcolor="#efefef" align="center" valign="top">
    <table border="0" cellpadding="2" cellspacing="2">
      <tr>
        <td bgcolor="#d6d6d6">
          <font face="Arial, sans-serif" size="2"><b>Wielkość
            ↳ zamówienia</b></font>
        </td>
        <td bgcolor="#d6d6d6">
          <font face="Arial, sans-serif" size="2"><b>100</b></font>
        </td>
        <td bgcolor="#d6d6d6">
          <font face="Arial, sans-serif" size="2"><b>250</b></font>
        </td>
        <td bgcolor="#d6d6d6">
          <font face="Arial, sans-serif" size="2"><b>500</b></font>
        </td>
        <td bgcolor="#d6d6d6">
          <font face="Arial, sans-serif" size="2"><b>1000</b></font>
        </td>
      </tr>
    </table>
  </td>
</tr>
```

```

        <td bgcolor="#d6d6d6">
            <font face="Arial, sans-serif" size="2"><b>
                ↳Cena jednostkowa</b></font>
            </td>
            <td><font face="Arial, sans-serif" size="2">11,50 PLN</font></td>
            <td><font face="Arial, sans-serif" size="2">10,00 PLN</font></td>
            <td><font face="Arial, sans-serif" size="2">9,50 PLN</font></td>
            <td>
                <font face="Arial, Helvetica, sans-serif" size="2">9,00 PLN</font>
            </td>
        </tr>
        <tr>
            <td bgcolor="#d6d6d6">
                <font face="Arial, sans-serif" size="2">
                    <b>Dodatkowe kolory (dopłata za sztukę)</b>
                </font>
            </td>
            <td><font face="Arial, sans-serif" size="2">1,50 PLN</font></td>
            <td><font face="Arial, sans-serif" size="2">1,20 PLN</font></td>
            <td><font face="Arial, sans-serif" size="2">1,20 PLN</font></td>
            <td><font face="Arial, sans-serif" size="2">1,20 PLN</font></td>
        </tr>
    </table>
</td>
<td width="200" bgcolor="#fff" valign="top">
    <font face="Arial, Helvetica, sans-serif" size="2" color="#666">
        <b>Cena obejmuje druk w jednym kolorze, w jednej pozycji.<br><br>
        Cena wydruku matrycy wynosi 100 PLN za jeden kolor i pozycję.</b>
    </font>
</td>
</tr>
</table>
</td></tr></table>

```

Jak pamiętamy, ta strona zawiera tylko dwie kolumny, zatem dół strony kończy się na końcu komórki. Dodatkowo komórka zawiera łącza wyświetlające się w stopce strony.

```

        <font face="Arial, Helvetica, sans-serif" size="1" color="#00c">
            <!-- ŁĄCZA STOPKI STRONY -->
        </font>
    </center>
</td>
</tr>
</table>
</td>
</tr>
</table>

</td></tr></table>
</center>

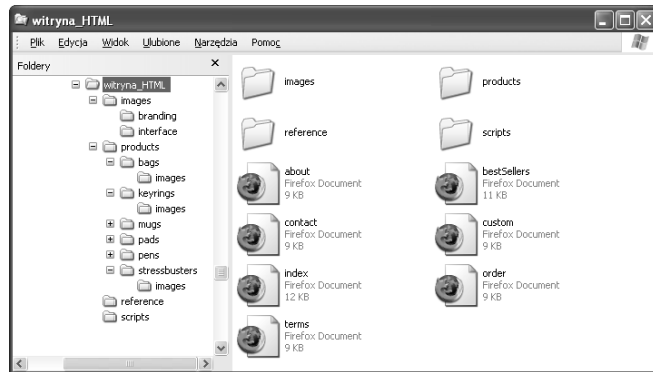
```

Do tego momentu omówiliśmy strukturę wszystkich stron w witrynie. Strony różnią się tylko treścią. Warto jednak przyjrzeć się strukturze katalogów witryny.

Struktura witryny

Po przeanalizowaniu sposobu, w jaki zostały napisane poszczególne strony witryny Fajna Promocja, warto przyjrzeć się, jak wygląda struktura katalogów. Strukturę tę pokazano na rysunku 1.9.

Rysunek 1.9



Strona główna witryny jest zapisana w pliku *index.html* i umieszczona w katalogu głównym. Są w nim również foldery *images*, *products* oraz następujące strony:

Nazwa pliku	Przeznaczenie
<i>aboutUs.html</i>	Zawiera informacje o firmie.
<i>bestSellers.html</i>	Szczegółowe informacje na temat produktów, które sprzedają się najlepiej.
<i>contact.html</i>	Informacje umożliwiające skontaktowanie się z firmą.
<i>custom.html</i>	Sposób składania zamówień na towary niestandardowe, których nie ma w katalogu.
<i>order.html</i>	Sposób składania zamówień na towary z katalogu.
<i>terms.html</i>	Warunki korzystania z serwisu.

Wewnątrz folderu *images* są dwa dodatkowe foldery. Pierwszy z nich nosi nazwę *branding* i zawiera obrazy dotyczące marki, na przykład logo. Drugi ma nazwę *interface* i zawiera obrazy służące do tworzenia interfejsu użytkownika, na przykład przyciski nawigacyjne.

Wewnątrz folderu *products* jest sześć podfolderów — po jednym dla każdej kategorii produktów znajdujących się w ofercie firmy Fajna Promocja: długopisy, torby, kubki, breloczki, notatniki i zabawki antystresowe. Każdy z folderów ma następującą zawartość:

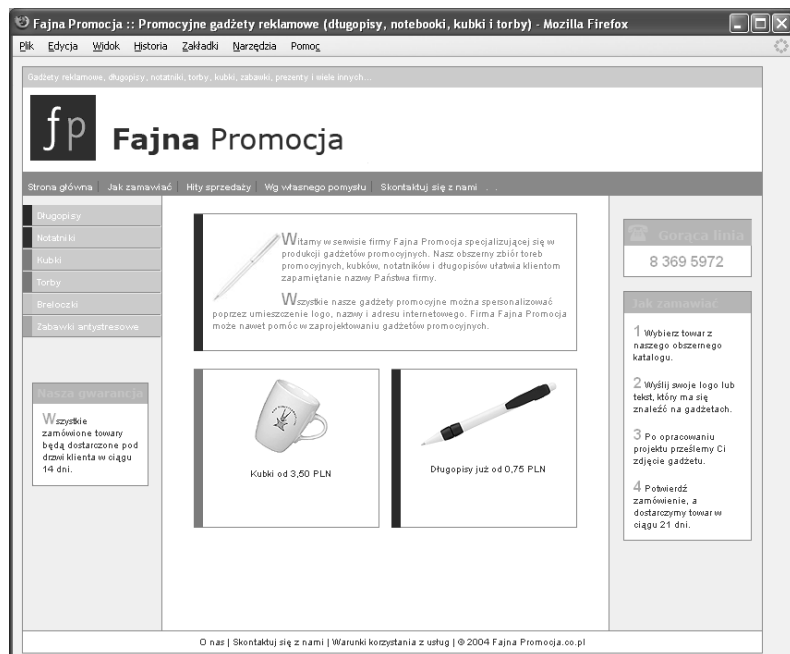
- stronę *index.html* zawierającą przegląd wszystkich produktów w określonej kategorii;
- folder *images* zawierający zdjęcia produktów z danej kategorii (dla każdej pozycji jest dostępna miniaturka i zdjęcie w pełnych rozmiarach);
- strony dla każdego produktu zawierające więcej szczegółów na temat produktu i jego cen.

Przed kontynuowaniem lektury dalszych rozdziałów warto wypróbować witrynę w pierwotnej postaci. W tym celu można ją pobrać spod adresu *ftp://ftp.helion.pl/przyklady/xcsdwi.zip* razem z pozostałymi kodami źródłowymi wykorzystywanymi w tej książce. Po pobraniu plików, można przetestować je na własnym komputerze. Dokładne przetestowanie witryny przyda się w dalszej części książki, gdzie dokładniej przeanalizowano kolejne strony serwisu.

Aktualizacja witryny

W dalszych rozdziałach tej książki, w miarę omawiania kolejnych tematów, będziemy wprowadzali zmiany w pierwotnej witrynie. Na końcu książki stworzymy nową witrynę zgodną z regułami języka XHTML, w której stylem i układem strony zarządza CSS oraz która spełnia wskazówki dotyczące dostępności, zarówno te, które są zalecane przez organizację W3C, jak i te, które wymieniono w Sekcji 508. Projekt witryny również będzie bardziej nowoczesny. Na rysunku 1.10 pokazano zrzut ekranu strony w docelowej postaci — uzyskamy ją po wprowadzeniu wszystkich zmian.

Rysunek 1.10



Aktualizacja witryny w niniejszej książce będzie przebiegała według następującego schematu:

1. Przy okazji omawiania własności języka XHTML (oraz różnic pomiędzy HTML i XHTML) będziemy modyfikowali kolejne struktury HTML, z którymi właśnie się zapoznaliśmy, i zmieniali je zgodnie z zasadami języka XHTML. Na końcu rozdziału 2. cały kod serwisu będzie zapisany zgodnie ze składnią XHTML, a znaczniki służące do prezentacji zostaną z niego usunięte. W rezultacie strona będzie wyglądała znacznie mniej estetycznie. Nie należy się jednak tym przejmować — stanie się na powrót bardziej atrakcyjna w kolejnych rozdziałach, kiedy zastosujemy CSS. Strony XHTML są również łatwiejsze w utrzymaniu, ponieważ nie zaśmiecają ich znaczniki prezentacyjne.

Kiedy nauczymy się pisać strony XHTML, poradzimy sobie również z napisaniem treści prezentowanej na urządzeniach wykorzystujących przeglądarki XHTML, które niekoniecznie obsługują wszystkie własności języka HTML. Co więcej, ponieważ język XHTML napisano w XML, dla wszystkich tworzonych stron można używać dostępnych narzędzi XML, takich jak XSLT, DOM oraz procesory SAX.

- 2.** W rozdziałach 3. i 4. pokażemy, w jaki sposób można wykorzystać język CSS do zarządzania prezentacją witryny, a na koniec rozdziału 4. strona będzie wyglądać podobnie do jej pierwotnej postaci z tą różnicą, że do zarządzania wyglądem będzie wykorzystywała CSS zamiast znaczników prezentacyjnych. W ten sposób uzyskamy separację stylu od treści, co spowoduje, że aktualizacja i pielęgnacja stylu witryny stanie się znacznie łatwiejsza. Po wprowadzeniu tej modyfikacji zmiana koloru lub czcionki w całym serwisie będzie wymagała wprowadzenia zmiany zaledwie w jednym arkuszu stylów. Ponieważ reguły prezentacji są zapisane w jednym arkuszu stylów i nie są powtarzane na każdej stronie (oraz ponieważ po pobraniu strony przeglądarka korzysta z lokalnej kopii arkusza stylów), rozmiar plików serwisu po modyfikacji jest mniejszy, zatem strona będzie ładowała się szybciej. Wykorzystanie arkuszy stylów do zarządzania prezentacją oznacza również możliwość dołączania różnych arkuszy stylów do tego samego dokumentu HTML w celu zaprezentowania go w innym celu lub na innym urządzeniu.
- 3.** W rozdziale 5. opisano sposób zarządzania układem stron z wykorzystaniem CSS. Zamiast wykorzystywania tabel do tworzenia nagłówka i łączy nawigacyjnych u góry strony oraz korzystania z komórek do tworzenia kolumn w głównej treści strony, te części witryny zostaną umieszczone wewnątrz elementów `<div>`, a ich pozycja zostanie określona za pomocą CSS. Wykorzystanie CSS do zarządzania układem strony oznacza, że tabele są wykorzystywane tylko do prezentowania danych tabelarycznych, czyli do tego, do czego były pierwotnie przeznaczone. Poza tym, układ całej witryny można zmodyfikować poprzez zmodyfikowanie zaledwie jednego arkusza stylów. Dzięki temu strony XHTML powinny stać się prostsze, a cała witryna bardziej dostępna.
- 4.** Zagadnieniami dostępności szerzej zajmiemy się w rozdziałach 6. i 7. Nauczymy się z nich technik, które zapewniają spełnienie przez witrynę zaleceń dostępności organizacji W3C oraz zaleceń opisanych w Sekcji 508. Po utworzeniu wersji witryny w XHTML i zastosowaniu arkusza CSS do zarządzania prezentacją i układem strony, bardzo zbliżymy się do celu, jakim jest utworzenie dostępnej witryny. Trzeba jednak pamiętać o wielu dodatkowych problemach, które zostaną opisane w tych rozdziałach. Przeanalizujemy w nich różne części witryny oraz podamy wskazówki dotyczące każdej z nich. Na przykład powiemy, w jaki sposób tworzy się tekstową alternatywę dla elementów nietekstowych (np. grafiki, klipów wideo i plików audio). Dzięki temu użytkownicy mający problemy ze wzrokiem mogą korzystać z formularzy i tabel. Opiszemy również niektóre narzędzia, które można wykorzystać po to, by zapewnić spełnienie przez witrynę wymagań dostępności.
- 5.** Na koniec, w rozdziale 8., pokażemy planowane kierunki rozwoju języka XHTML. Zastanowimy się, w jaki sposób język będzie ewoluował w kierunku obsługi nowych urządzeń, które można wykorzystać do uzyskania dostępu do internetu. W tym rozdziale przekonamy się, że przejście na język XHTML tworzy solidne podstawy do opracowywania stron dla wielu typów nowych urządzeń. Pokażemy nawet przykład tego, jak może wyglądać witryna przeznaczona do wyświetlania na telefonie komórkowym.

Kod ostatecznej wersji witryny jest dostępny do pobrania razem z pozostałym kodem źródłowym przykładów pokazanych w tej książce. Witrynę w oryginale można również zobaczyć w internecie, pod adresem <http://www.FirstPromotions.co.uk/>.

Podsumowanie

W niniejszym rozdziale dowiedzieliśmy się, jak zmienił się internet z biegiem lat, oraz zrozumieliśmy, że autorzy stron WWW muszą zaktualizować swoje umiejętności tak, by uwzględnić te zmiany. Dowiedzieliśmy się, że w miarę postępu techniki poprawiły się rozdzielczości monitorów używanych do dostępu do internetu, a jednocześnie zmieniły się ich rozmiary. Jednocześnie zwiększył się asortyment urządzeń zdolnych do przeglądania stron internetowych. Oznacza to, że obecnie nie wystarczy już utworzyć jednego projektu strony WWW i oczekiwać, że będzie ona działała na wszystkich urządzeniach, które mają możliwość dostępu do internetu.

Chociaż w kolejnych wersjach HTML wprowadzano różnorodne elementy i atrybuty pomagające autorom stron WWW w zarządzaniu wyglądem stron, obecnie należy zaprzestać używania znaczników prezentacyjnych w tych samych dokumentach, które zawierają zasadniczą treść witryny. Należy oddzielić znaczniki opisujące strukturę dokumentu od reguł stylu opisujących sposób, w jaki strona powinna być wyświetlana. Reguły prezentacji należy umieścić w osobnym arkuszu stylów napisanym w języku znanym jako CSS (ang. *Cascading Style Sheets* — kaskadowy arkusz stylów).

Język HTML zastąpiono więc nowym językiem, z którego usunięto znaczniki opisujące styl. Język ten w pewnym stopniu różni się od HTML również składnią, ponieważ następcę języka HTML napisano w języku XML. Z tego powodu nazwa języka zmieniła się na XHTML. Zapisanie języka HTML w XML pozwala zapewnić, że najczęściej używany obecnie w internecie język nie straci na popularności w kolejnych latach, nawet jeśli pojawią się nowe urządzenia o całkowicie nowych możliwościach. Zagadnienie to bardziej szczegółowo opisano w rozdziale 8.

Na koniec pokazaliśmy, że spełnienie wymagań prawnych zapewnienia dostępności stron WWW dla osób niepełnosprawnych wymaga wprowadzenia wielu zmian do sposobu projektowania stron WWW. Nauczenie się pisania witryn WWW w XHTML i CSS to pierwszy krok na drodze do budowania dostępnych witryn WWW, jednak tworzenie dostępnych witryn WWW w pełnym tego słowa znaczeniu wymaga zastosowania się do wielu wskazówek i skorzystania z wielu technik. Zostaną one opisane w rozdziałach 6. i 7.