

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

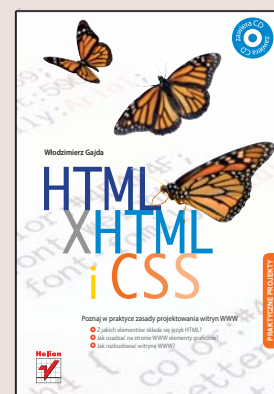
ZAMÓW CENNIK

CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

HTML, XHTML i CSS. Praktyczne projekty

Autor: Włodzimierz Gajda
ISBN: 83-246-0885-0
Format: B5, stron: około 500
Zawiera CD-ROM



Poznaj w praktyce zasady projektowania witryn WWW

- Z jakich elementów składa się język HTML?
- Jak osadzać na stronie WWW elementy graficzne?
- Jak rozbudować witrynę WWW?

Internet rozwija się w niesamowitym tempie. Każdego dnia pojawiają się nowe strony WWW. Własne „miejsce” w sieci dawno już przestało być ekstrawagancją. Dziś serwis internetowy jest wyznacznikiem nowoczesnego przedsiębiorstwa i organizacji, miejscem, gdzie prezentuje się swoją ofertę, zdjęcia, dzieli opiniami dotyczącymi otaczającego nas świata, uczestniczy w dyskusjach i robi zakupy. Stworzenie własnej witryny WWW wbrew pozorom nie jest sprawą bardzo skomplikowaną. Gotowe kreatory i szablony dostępne u dostawców usług internetowych znacznie to ułatwiają, jednak prawdziwą kontrolę nad wyglądem i funkcjonalnością strony internetowej uzyskamy tylko dzięki własnoręcznemu jej utworzeniu. W tym celu niezbędne jest jednak opanowanie podstawowego „budulca” witryn WWW – języka HTML i technologii CSS.

Czytając książkę „HTML, XHTML oraz CSS. Praktyczne projekty”, opanujesz zagadnienia związane z projektowaniem witryny WWW od strony praktycznej. Dowiesz się, z jakich elementów składa się język HTML i czym różni się od niego język XHTML. Poznasz techniki formatowania tekstu i elementów graficznych za pomocą stylów CSS, nauczysz się umieszczać na stronie tabele, łącza, ramki i proste efekty specjalne. Przeczytasz także o planowaniu struktury witryny, korzystaniu z szablonów i projektowaniu stron WWW tak, aby mogły korzystać z nich osoby niepełnosprawne.

- Składnia języka XHTML
- Kaskadowe arkusze stylów
- Definiowanie krojów czcionek dla tekstów
- Elementy blokowe i tekstowe
- Tabele i listy
- Tworzenie łączy pomiędzy stronami
- Definiowanie tła dokumentu
- Efekty specjalne
- Projektowanie struktury witryny

Pokaż się w sieci. Stwórz własną witrynę WWW!



Spis treści

Rozdział 1. Wprowadzenie	13
Dla kogo jest ta książka?	13
Jak czytać tę książkę?	14
Testowanie stron WWW	14
Edycja kodu XHTML	15
Część I Elementarz	17
Rozdział 2. Składnia języka XHTML	19
Znaczniki i elementy	19
Wszystkie elementy języka XHTML	20
Elementy puste i niepuste	20
Znaczniki wymagane, opcjonalne oraz elementy puste	22
Wielkość liter w nazwach znaczników	23
Białe znaki wewnątrz znaczników	23
Białe znaki w treści elementów	24
Zagnieżdżanie elementów	24
Znaki specjalne	25
Atrybuty znaczników	27
Atrybuty logiczne, wyliczeniowe i inne	30
Komentarze w XHTML	32
Struktura dokumentu HTML	32
Pierwsza strona WWW	33
Rozdział 3. Znaki diakrytyczne i oznaczanie języka dokumentu	37
Polskie znaki diakrytyczne	37
Metody kodowania polskich znaków diakrytycznych	38
Fizyczne kodowanie pliku	38
Element meta ustalający kodowanie dokumentu XHTML	40
Pangramy	41
Atrybut lang	42
Szablony pustych polskich stron WWW	43
Znaki diakrytyczne w postaci encji	44
Kodowanie stron zawierających teksty w kilku językach	44
Jakiego kodowania używać?	45
Błędne wyświetlanie polskich znaków diakrytycznych	49
Ćwiczenia	49

Rozdział 4. XHTML poprawny składniowo	55
Czy poprawność składniowa jest ważna?	55
Obecny stan internetu	56
Metody sprawdzania poprawności składniowej	56
Rozdział 5. Tryb standardów	59
Quirks mode, standard mode — dwa tryby pracy przeglądarek internetowych	59
Problemy z trybami pracy	60
Które przeglądarki mają tryb standardów?	61
Jak stwierdzić tryb pracy przeglądarki?	61
W jaki sposób przeglądarka wybiera tryb pracy?	63
Stosuj DOCTYPE języka XHTML 1.0 strict	64
Rozdział 6. Podstawowe elementy XHTML	67
Akapit	68
Dzielenie wyrazów	73
Zakaz złamania wiersza	74
Złamanie wiersza	76
Znaki interpunkcyjne	78
Nagłówki	78
Wyróżnianie tekstu	81
Tekst preformatowany	83
Indeksy dolny i górny	85
Linia pozioma	86
Popularne znaki specjalne	87
Zestawienie	88
Rozdział 7. Kaskadowe arkusze stylów	89
Struktura a wygląd dokumentów HTML	89
Dołączanie stylów do dokumentu	90
Style zewnętrzne	90
Style wewnętrzne	91
Atrybut style	92
Domyślny język stylów	94
Ujmowanie stylów wewnętrznych w komentarz HTML	94
Rozdział 8. Składnia kaskadowych arkuszy stylów	97
Terminologia	97
Wielkość liter w selektorach	98
Wielkość liter w nazwach i wartościach właściwości	99
Białe znaki	99
Komentarze	100
Rozdział 9. Przykładowe właściwości CSS	101
Czcionki	101
Wysokość wiersza tekstu	103
Wyrównanie poziome tekstu	103
Marginesy	104
Kolory	105
Obramowanie	107
XHTML — struktura, CSS — wygląd	108
Zestawienie sumaryczne	108
Rozdział 10. Klasy i identyfikatory	111
Atrybut class	111
Selektory dotyczące klas	112

Stosowanie klas	112
Atrybut id	114
Selektory dotyczące identyfikatorów	114
Stosowanie identyfikatorów	115
Walka z classitis — selektory potomne	116
Rozdział 11. Projekty	119
Projekt 11.1. Zadania dla czwartoklasistów	120
Projekt 11.2. Zadania tekstowe z odpowiedziami	120
Projekt 11.3. Kolokwium z PHP	121
Projekt 11.4. Adam Mickiewicz: Oda do młodości	122
Projekt 11.5. Cyprian Kamil Norwid: Moja piosnka	123
Projekt 11.6. Charles Dickens: A Christmas Carol	124
Projekt 11.7. Jack London: The Call of The Wild	125
Projekt 11.8. Ignacy Krasicki: Bajki	127
Projekt 11.9. Adam Mickiewicz: Wiersze	128
Część II Czcionki na stronach WWW	131
Rozdział 12. Rodzaje czcionek	133
Czcionki szeryfowe i bezszeryfowe	133
Czcionki proporcjonalne i nieproporcjonalne	134
Inne podziały czcionek	135
Testowanie czcionek	137
Rozdział 13. Czcionki dla webmastera	139
Ogród Zen CSS	139
Core fonts for the Web	141
Czcionki dostępne na różnych platformach	141
Rozdział 14. Krój i rozmiar czcionki w CSS	143
Krój czcionki	143
Wielkość czcionki w CSS	146
Rozdział 15. Wszystkie atrybuty CSS 2.1 dotyczące czcionek i tekstu	149
font	150
font-family	151
font-size	151
font-style	153
font-weight	153
font-variant	153
text-align	153
text-decoration	154
text-indent	154
text-transform	154
word-spacing	155
letter-spacing	155
white-space	155
line-height	156
Rozdział 16. Projekty	157
Projekt 16.1. CSS — właściwości dotyczące czcionek i tekstu	157
Projekt 16.2. Specyfikacja aparatu fotograficznego Nikon D200	159
Projekt 16.3. Jan Kochanowski: Treny	160
Projekt 16.4. Adam Mickiewicz: Pan Tadeusz	161

Część III Elementy blokowe, tekstowe i ich pudełka	163
Rozdział 17. Elementy blokowe i liniowe	165
Elementy ogólne div i span	165
Użycie elementów div i span w połączeniu z klasami i identyfikatorami	166
Typowy przykład użycia elementów div	167
Rozdział 18. Pudełka	169
Obszar, jaki zajmie pudełko	171
Wyśrodkowanie elementu blokowego	173
Pudełka pływające (float)	174
Czyszczenie elementów pływających	176
Rozdział 19. Proste układy stałej szerokości	177
Układ nr 1	178
Układ nr 2	179
Układ nr 3	180
Układ nr 4	182
Układ nr 5	183
Układ nr 6	185
Układ nr 7	186
Układ nr 8	188
Rozdział 20. Proste układy zmiennej szerokości	191
Układ nr 9	191
Układ nr 10	191
Układ nr 11	192
Układ nr 12	193
Układ nr 13	193
Układ nr 14	194
Układ nr 15	195
Układ nr 16	195
Rozdział 21. Projekty	197
Projekt 21.1. Maria Konopnicka: Miłosierdzie gminy	197
Projekt 21.2. Leopold Staff: Pierwsza przechadzka	199
Projekt 21.3. Sławomir Mrożek: Słoń	200
Projekt 21.4. Stefan Żeromski: Rozdziobią nas kruki, wrony	201
Projekt 21.5. Bolesław Prus: Katarynka	203
Projekt 21.6. Planety układu słonecznego	205
Projekt 21.7. Funkcje file, count, explode oraz trim, czyli krojenie plików tekstowych w PHP	206
Część IV Elementy XHTML	209
Rozdział 22. Tekst	211
Elementy frazowe	211
Trudne wybory	214
Cytaty	214
Tekst na stronach WWW — podsumowanie	217
Rozdział 23. Listy	219
Wypunktowanie	219
Numerowanie	220
Lista definicji	221
Zagnieżdżanie list	222

Właściwości CSS list	224
Właściwość list-style-type	224
Właściwość list-style-image	225
Właściwość list-style-position	228
Właściwość list-style	228
Rozdział 24. Obrazy	229
Element img	230
Wymiary obrazów	232
Obrazy nieprostokątne	234
Style CSS elementu img	234
Opływanie	236
Dlaczego elementy pływające nie generują wysokości?	238
Osadzanie obrazów w kodzie XHTML	240
Rozdział 25. Tabele	243
Obramowanie i łączenie obramowania	244
Podstawowe formatowanie komórek i całej tabeli	246
Nagłówki kolumn i nagłówki wierszy	248
Podpis i opis tabeli	250
Tabele regularne i nieregularne	252
Nagłówek, stopka i treść tabeli	254
Kolumny tabeli	257
Tabele XHTML — podsumowanie	261
Rozdział 26. Odsyłacze	263
Spis treści w postaci listy numerowanej bądź wypunktowanej	265
Style CSS witryny z hiperłączami	266
Atrybut title	267
Odsyłacze do różnych typów plików	267
Odsyłacze wskazujące strony w internecie	268
Odsyłacze wewnętrzne	269
Obrazy jako odsyłacze	271
Style CSS odsyłaczy	272
Otwieranie nowych okien	274
Rozdział 27. Projekty	277
Projekt 27.1. Obrazy tworzone przy użyciu elementu pre	277
Projekt 27.2. ASCII Art	278
Projekt 27.3. Emotikony	279
Projekt 27.4. Cytaty ze specyfikacji HTML i XHTML	280
Projekt 27.5. Bibliografia	283
Projekt 27.6. Książka The Wonderful Wizard of Oz	284
Projekt 27.7. Instrukcja wykonywania zrzutów ekranu	285
Projekt 27.8. Jak pisać na klawiaturze w języku rosyjskim?	286
Projekt 27.9. Metody definiowania miejsc geometrycznych punktów	286
Projekt 27.10. Artykuł „HTML czy XHTML?”	288
Projekt 27.11. Zabytki Lublina	291
Projekt 27.12. Literackie Nagrody Nobla	293
Projekt 27.13. Tabela Orange Ekstraklasy w sezonie 2006/2007	295
Projekt 27.14. Kod paskowy rezystorów	296
Projekt 27.15. Tatry	297
Projekt 27.16. LaTeX — przykłady	298
Projekt 27.17. Janko Muzykant	299
Projekt 27.18. Favelety	300

Część V Tła	301
Rozdział 28. Właściwości CSS dotyczące tła	303
Rozdział 29. FIR — zamiana obrazów na teksty	315
Rozdział 30. Udawane kolumny	321
Rozdział 31. Przyciski rollover w CSS	331
Wymiana obrazu tła	331
Przycisk z etykietą tekstową	332
Pozycjonowanie względnie bezwzględne	334
Rozdział 32. Kafelkowanie	343
Rozdział 33. Projekty	349
Projekt 33.1. Wisława Szymborska: Miłość szczęśliwa	349
Projekt 33.2. Wiersze Marii Pawlikowskiej-Jasnorzewskiej	350
Projekt 33.3. Ogłoszenie	350
Projekt 33.4. Jack London: The Call of the Wild	351
Projekt 33.5. Fraszki	352
Część VI Zagadnienia dalsze	355
Rozdział 34. Struktura funkcjonalna witryny	357
Rozdział 35. Uzupełnienie wiadomości na temat CSS	361
Selektory	361
Importowanie stylów	366
Dziedziczenie	367
Style do druku	368
Projekt 35.1	370
Rozdział 36. Kolejność elementów w kodzie XHTML	371
Zmiana kolejności kolumn pionowych	372
Zmiana kolejności poziomych pasów	376
Rozdział 37. Atrybuty XHTML	379
Atrybuty zasadnicze	379
Atrybuty językowe	380
Zdarzenia	380
Atrybuty ogólne	381
Atrybuty dotyczące aktywnego punktu	382
Rozdział 38. Formularze	385
Atrybuty formularza	386
Kontrolki formularza	388
Atrybuty ogólne kontrolek formularza	389
Zdarzenia dotyczące kontrolek	390
Elementy input	391
Przyciski zatwierdzające i resetujące formularz	391
Wiersz wprowadzania danych	393
Pole hasła	393
Pola wyboru	394
Wykluczające pola wyboru	395
Kontrolki ukryte	396
Przyciski	396

Kontrolka wyboru pliku	397
Obraz	397
Element button	398
Listy	399
Pole tekstowe	401
Grupowanie i podpisywanie kontrolki formularza	402
Podsumowanie	404
Rozdział 39. Powiązania dokumentów	405
Ikona strony WWW	405
Następny, poprzedni oraz spis treści	408
Alternatywne wersje dokumentu	408
Projekt 39.1	409
Rozdział 40. Element meta — dodatkowe informacje na temat strony WWW	413
Składnia elementu meta	414
Znaczenie elementu meta	414
Dwa rodzaje elementów meta	415
Jakie metainformacje umieszczać w witrynach?	415
Kodowanie znaków	416
Język, w którym przygotowano dokument	416
Autor, prawa autorskie i firma	417
Słowa kluczowe i opis	417
Roboty	418
Data powstania i ważności dokumentu	419
Przechowywanie stron WWW przez pośredników	419
Skrypty i style — domyślny język	420
Metainformacje w kilku językach	420
Przekierowania	421
Różności	421
Rozdział 41. Dostępność strony WWW	423
Kilka prostych zasad	423
Stosuj atrybut lang	424
Definiuj skróty i skrótowce	424
Określaj powiązania między poszczególnymi podstronami witryny	424
Treść umieszczaj jako pierwszą	424
Pamiętaj o atrybutach alt	425
Określaj tytuły hiperłączy	425
Twórz czytelne tabele	425
Stosuj skalowalne czcionki	426
Etykietuj kontrolki formularzy	426
Nie otwieraj nowych okien	426
Nie używaj przekierowań meta refresh	426
Rozdział 42. HTML czy XHTML?	427
XHTML zgodny z HTML	427
Wielkość liter	427
Elementy puste i niepuste	429
Znaczniki opcjonalne	429
Cudzysłów obejmujący wartości atrybutów	430
Minimalizacja atrybutów logicznych	431
Identyfikator fragmentu	431
Style i skrypty	431
Encje	432

Dokumenty HTML/XHTML w sieci WWW	432
Content-type	433
Czy to HTML, czy XHTML?	435
Jak przeglądarka traktuje dokument HTML, a jak XHTML?	435
Po czym przeglądarka rozpoznaje język dokumentu?	437
Nagłówek Content-type dokumentu HTML oraz XHTML	437
Problemy z Internet Explorerem	437
Strona XHTML wysyłana jako application/xhtml+xml	438
Zmiana nagłówków wysyłanych przez serwer Apache	438
Wysyłanie nagłówka HTTP w PHP	438
Cztery proste zasady	438
HTML czy XHTML?	439
Rozdział 43. Semantyczny XHTML	441
Witryna WWW widziana oczami człowieka i robota	441
Semantyczny Web	442
Semantyka kodu XHTML	442
Problemy semantyczne XHTML i CSS	444
Dodatki	449
Dodatek A Instalacja oprogramowania	451
Skorowidz	465

Rozdział 7.

Kaskadowe arkusze stylów

Struktura a wygląd dokumentów HTML

Wygląd witryn WWW zależy od dwóch czynników: kodu strony napisanego w języku XHTML oraz formatu nadawanego przez przeglądarkę różnym elementom XHTML. Strukturę i zawartość dokumentu opisujemy, korzystając z elementów XHTML, m.in. akapitów (p) i nagłówków (h1, h2), a format elementów definiujemy w języku CSS (ang. *Cascading Style Sheets* — kaskadowe arkusze stylów).

Tworząc witrynę WWW, musimy więc opisać:

- ◆ strukturę i zawartość strony (język XHTML),
- ◆ format elementów (język CSS).

W przypadku braku stylów przeglądarka zastosuje style domyślne, które zazwyczaj są dość ubogie.

Oddzielając definicję formatu elementów od samej zawartości strony otrzymamy dokument, którego format będzie niezależny od zawartości. Główną korzyścią takiego rozwiązania jest to, że wygląd dokumentu możemy modyfikować, nie zmieniając jego treści (czyli kodu XHTML).

Dołączanie stylów do dokumentu

Style CSS możemy dołączyć do dokumentu XHTML na trzy sposoby:

- ♦ jako style zewnętrzne,
- ♦ jako style wewnętrzne,
- ♦ stosując atrybut style.

W pierwszym przypadku style są zapisane w osobnym pliku. W drugim style znajdują się w nagłówku strony WWW, a więc pomiędzy znacznikami `<head>` oraz `</head>`. W trzecim style pojawiają się przy konkretnych elementach XHTML w treści strony, czyli pomiędzy znacznikami `<body>` oraz `</body>`.

Style zewnętrzne

Style zewnętrzne zapisujemy w osobnym pliku. Plik ze stylami ma zazwyczaj rozszerzenie `.css`. Tak zdefiniowane style dołączamy do dokumentu XHTML, stosując element `link`:

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



Style zewnętrzne `<link ... />` — *Ctrl+B+Z*, *Ctrl+B+S*

Podany element dołącza do strony WWW style zapisane w pliku `style.css`.

Witryna składa się więc z dwóch plików: `index.html` oraz `style.css`. Listing 7.1 przedstawia przykładowy plik `index.html`, a 7.2 — zawartość pliku `style.css`. W kodzie XHTML pojawia się element `link` z atrybutem `href`. Wartością atrybutu `href` jest nazwa pliku ze stylami (w przykładzie: `style.css`).

Listing 7.1. Kod XHTML strony WWW stosującej style zewnętrzne (plik `index.html`)

```
<!DOCTYPE ...>
<html ...>
  <head>
    <title>Style zewnętrzne</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <link rel="stylesheet" type="text/css" href="style.css" />
  </head>
  <body>

  <h1>WITAJ</h1>

  </body>
</html>
```

Listing 7.2. *Style CSS (plik style.css)*

```
h1 {
  margin: 20px;
  background: blue;
  color: white;
  border: 4px solid black;
  text-align: center;
}
```

Zaletą takiego rozwiązania jest to, że w jednym miejscu możesz modyfikować wygląd wszystkich podstron witryny. Co więcej, tak wykonana witryna będzie zajmowała mniej miejsca i szybciej się wczyta. Style zostaną pobrane z serwera jeden jedyny raz. Kosztem dołączenia stylów do witryny jest jedynie element link dodany na każdej podstronie.

Ponadto style zawarte w zewnętrznym pliku mogą zawierać dowolne znaki, także <, > czy &., nie powodując żadnych komplikacji¹.

To jest najlepsza metoda formatowania wyglądu witryny. Należy ją stosować w odniesieniu do większości witryn przeznaczonych do publikacji w internecie.

Style wewnętrzne

Definicja stylów wewnętrznych wykorzystuje element `style`:

```
<style type="text/css">
...
tutaj definicja stylów
...
</style>
```

Element ten umieszczamy w nagłówku witryny WWW.



Style wewnętrzne `<style ...>...</style>` — *Ctrl+B+W*

Listing 7.3 przedstawia kod przykładowej strony WWW, która stosuje style wewnętrzne. W nagłówku witryny pojawia się element `style`, zawierający definicję stylu nagłówka `h1`. W treści witryny, pomiędzy znacznikami `<body>` oraz `</body>`, występuje element `h1`, którego wygląd zostanie zmieniony zgodnie ze stylami podanymi wewnątrz elementu `style`.

¹ Specyfikacja XHTML 1.0, punkt C.4.

Listing 7.3. *Style wewnętrzne*

```
<!DOCTYPE ...>
<html ...>
  <head>
    <title>Style wewnętrzne</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <style type="text/css">
      h1 {
        margin: 20px;
        background: blue;
        color: white;
        border: 4px solid black;
        text-align: center;
      }
    </style>
  </head>
  <body>

  <h1>WITAJ</h1>

  </body>
</html>
```

Przykład ten składa się z jednego pliku: *index.html*.

Zaletą tego pliku jest to, że w jednym miejscu pojawiają się style i kod XHTML. Jeśli wykonasz na takiej witrynie operację podglądu źródła², to ujrzysz i kod XHTML, i style CSS³. Ponadto tak wykonana witryna będzie poprawnie wyglądała (tj. będzie ozdobiona stylami), gdy zostanie otworzona wewnątrz aplikacji. Na przykład, jeśli spakujesz kilka tak wykonanych stron, po czym otworzysz spakowane archiwum programem archiwizującym, witryna będzie poprawnie wyglądała po otwarciu z wnętrza archiwum (bez wypakowywania).

Wadą tego rozwiązania jest rozmiar witryny: każda podstrona projektu będzie zawierała komplet stylów. Ponadto, jeśli zechcesz wprowadzić zmianę, która obejmie wszystkie podstrony witryny, będziesz musiał zmienić style w każdym pliku z osobna.

W praktyce style wewnętrzne stosują w odniesieniu do dokumentów, które są poje-
dynczymi plikami (np. opisy programów umieszczane w spakowanych archiwach).

Atrybut style

Trzecia metoda definiowania stylów wykorzystuje atrybut `style`. Atrybut ten może towarzyszyć niemal każdemu elementowi XHTML. Zmiana formatu akapitu ma postać:

```
<p style="width: 300px; margin: 20px; background: blue;">Witaj</p>
```

² Opcja *Widok/Źródło* w przeglądarce.

³ Usprawnia to m.in. prowadzenie ćwiczeń z języków XHTML oraz CSS. Nauczyciel przygotowuje przykład i umieszcza go w sieci. Uczniowie po wykonaniu operacji podglądu źródła ujrzą kompletny kod XHTML oraz CSS.

Element `style` może być przydatny w specyficznych okolicznościach. Na przykład wtedy, gdy nie masz uprawnień do modyfikowania plików CSS na serwerze, a możesz modyfikować fragment pliku XHTML. Sytuacja taka może pojawić się na przykład w systemie CMS. Użytkownik nie ma prawa modyfikować żadnych plików (ani XHTML, ani CSS), ale może wprowadzać do bazy danych wpisy zawierające oprócz kodu XHTML atrybut `style`.

W przypadku witryny, w której masz dostęp do pliku CSS, rozwiązanie takie nie ma sensu. Niektórzy twierdzą nawet, że atrybut `style` jest porównywalny z dawno wycofanym elementem `font`⁴.

Użycie stylów wewnętrznych w odniesieniu do elementu `h1` pokazuje listing 7.4.

Listing 7.4. *Atrybut style*

```
<!DOCTYPE ...>
<html ...>
  <head>
    <title>Atrybut style</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  </head>
  <body>

    <h1 style="margin: 20px; background: blue; color: white;
border: 4px solid black; text-align: center;">WITAJ</h1>

  </body>
</html>
```

Rysunek 7.1 przedstawia wygląd opisanej witryny. Trzy podane przykłady: pierwszy ze stylami zewnętrznymi, drugi stosujący `style` wewnętrzne i trzeci wykorzystujący atrybut `style`, mają identyczny wygląd.

Rysunek 7.1.
*Witryna
stosująca style*



⁴ Henri Sivonen: *HTML Syntax Checker in PHP*, <http://hsivonen.iki.fi/html-syntax-checker/>.

Domyślny język stylów

Obecnie style dokumentów XHTML są opisywane wyłącznie w języku CSS. Jest to język domyślny stosowany przez wszystkie przeglądarki.

O języku stylów mówi atrybut `type="text/css"` zawarty w elementach `link` (style zewnętrzne):

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

oraz `style` (style wewnętrzne):

```
<style type="text/css">
...
</style>
```

Wartość `text/css` ustala, że style są zapisane w języku CSS.

Korzystając z atrybutu `style`, nie wskazujemy w żaden sposób, w jakim języku opisano style:

```
<p style="...">Witaj</p>
```

Domyślnym językiem i w tym przypadku jest CSS. Jeśli jednak chcesz być dokładny, to język stylów możesz wskazać, umieszczając w nagłówku witryny element `meta`:

```
<meta http-equiv="Content-Style-Type" content="text/css" />
```

Ujmowanie stylów wewnętrznych w komentarz HTML

W okresie gdy implementacja stylów CSS zaczynała się pojawiać w przeglądarkach, tj. w latach 1995 – 2000, element `style` sprawiał pewien kłopot. Jeśli przeglądarka go nie rozumiała, to mogła podaną w nim zawartość wyświetlić na stronie wraz z tekstem. Zabezpieczeniem przed takim niepożądanym działaniem było stosowanie komentarzy w kodzie HTML. Całą zawartość elementu `style` umieszczano w komentarzu:

PRZYKŁAD NIEPOPRAWNY

```
<style type="text/css">
<!--
p {
  font-family: Georgia, serif;
}
-->
</style>
```

Obecnie takie postępowanie nie tylko nie przynosi żadnej korzyści, ale może powodować, że style nie będą działały⁵. Powyższy przykład należy zapisywać jako:

```
<style type="text/css">
p {
  font-family: Georgia, serif;
}
</style>
```

Ćwiczenie 7.1

Wykonaj witrynę WWW z tekstem *Lorem ipsum*. Stosując style zewnętrzne, sformatuj akapity tak, by miały duży margines oraz niebieską, wytłuszczoną czcionkę Georgia podwójnej wielkości. Wykorzystaj style z listingu 7.5.

Listing 7.5. Style do ćwiczenia 7.1

```
p {
  margin: 100px;
  font-family: Georgia, serif;
  font-size: 200%;
  font-weight: bold;
  color: blue;
}
```

Ćwiczenie 7.2

Wykonaj witrynę WWW z tekstem *Lorem ipsum*. Zdefiniuj style identyczne jak w ćwiczeniu 7.1. Przygotuj rozwiązanie zawarte w jednym pliku (style wewnętrzne stosujące element style).

Ćwiczenie 7.3

Wykonaj witrynę WWW z tekstem *Lorem ipsum*. Style podane na listingu 7.5 przypisz do elementu p, wykorzystując atrybut style.

⁵ Specyfikacja XHTML 1.0, rozdział C.4.