helion.pl
KSIĘGARNIA INTERNETOWA · WYDAWNICTWO

IDŹ DO
PRZYKŁADOWY ROZDZIAŁ
SPIS TREŚCI

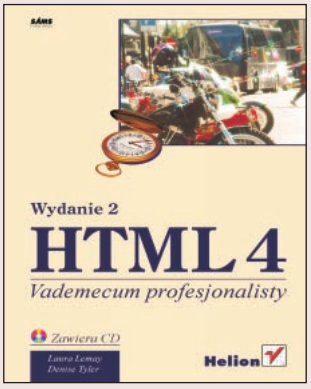
KATALOG KSIĄŻEK
KATALOG ONLINE
ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK
DODAJ DO KOSZYKA

CENNIK I INFORMACJE
ZAMÓW INFORMACJE O NOWOŚCIACH
ZAMÓW CENNIK

CZYTELNIA
FRAGMENTY KSIĄŻEK ONLINE

HTML4
Vademecum profesjonalisty
Wydanie II



Autorzy: Laura Lemay, Denise Tyler
Tłumaczenie: Piotr Rajca, Łukasz Orzechowski, Tomasz Jarzębowicz
ISBN: 83-7197-478-7
Tytuł oryginału: Sams Teach Yourself Web Publishing with HTML 4 in 21 Days Professional Reference Edition, Second Edition
Format: B5, stron: 1144, oprawa twarda
Zawiera CD-ROM

Szczegółowo poprawione i zaktualizowane wydanie tego łatwego do zrozumienia i wyczerpującego podręcznika zawiera omówienie wszystkich najnowszych technologii wykorzystywanych do publikowania informacji na WWW. W książce tej znajdziesz wszystkie zagadnienia związane z publikowaniem informacji na WWW, poczynając do podstaw języka HTML, a skończywszy na bardziej zaawansowanych zagadnieniach, takich jak:

- zastosowanie tabel,
- ramek,
- pisanie skryptów w języku JavaScript oraz skryptów CGI,
- wykorzystanie kaskadowych arkuszy stylów.

Dzięki tej książce zdobędziesz wszystkie umiejętności i wiedzę konieczną do zaprojektowania, stworzenia i utrzymania profesjonalnie wyglądającej witryny WWW.

Jest to w pełni poprawione wydanie obejmujące zagadnienia wykorzystania najnowszych technologii stosowanych podczas tworzenia WWW – w tym także XHTML. Odkryj, w jaki sposób tabele i ramki można wykorzystać do poprawienia układu stron WWW. Naucz się określać wygląd i położenie elementów stron dzięki zastosowaniu kaskadowych arkuszy stylów. Poznaj zasady wykorzystania obrazów, kolorów, animacji oraz dźwięków. Zapewnij interaktywność tworzonych witryn dzięki zastosowaniu: skryptów JavaScript, apletów pisanych w języku Java, formularzy oraz skryptów CGI. Zawiera wyczerpujące, 200-stronicowe omówienie języka HTML 4, wszystkich jego znaczników i atrybutów, ze szczególnym uwzględnieniem możliwości ich wykorzystania w poszczególnych przeglądarkach.

Laura Lemay jest jedną z najbardziej popularnych na świecie autorek książek dotyczących zagadnień publikowania informacji na stronach WWW. Oprócz książek z serii „Vademecum profesjonalisty", które stały się światowymi bestsellerami, jest także współautorką książki o Java 1.1 oraz autorką książki „Perl dla każdego".

Wydawnictwo Helion
ul. Chopina 6
44-100 Gliwice
tel. (32)230-98-63
e-mail: helion@helion.pl

Spis treści

O Autorach **29**

Wstęp **31**

Kto powinien przeczytać tę książkę 31
Co zawiera niniejsza książka 32
Czego potrzebujesz na początku 34
Konwencje przyjęte w książce 34
Elementy graficzne 34
Przykłady kodu źródłowego i wyników 35
Inna czcionka 35
CD-ROM 35
Strona internetowa oryginalnego wydania książki 35
Polskie litery 36

Część I Informacje wstępne **37**

Rozdział 1. Świat WWW **39**

Czym jest World Wide Web? 39
Sieć WWW jest hipertekstowym systemem informacyjnym 40
WWW jest systemem graficznym i łatwym w nawigacji 41
WWW jest siecią wieloplatformową 42
WWW jest siecią rozproszoną 43
WWW jest siecią dynamiczną 43
Przeglądarki sieciowe umożliwiają dostęp do wielu rodzajów informacji w Internecie 45
Sieć WWW jest interakcyjna 46
Przeglądarki sieciowe 47
Co robi przeglądarka? 48
Przegląd dwóch najpopularniejszych przeglądarek 48
Serwery WWW 50
URL 51
Podsumowanie 52
Warsztat 52
Pytania i odpowiedzi 52
Quiz 53
Odpowiedzi 54
Ćwiczenia 54

Rozdział 2. Grunt to organizacja 55
Anatomia witryn WWW 56
Do czego właściwie chcesz wykorzystać WWW? 57
Wyznacz swój cel 59
Podziel treść na główne tematy 60
Pomysły na organizację i nawigację 61
Struktura hierarchiczna 61
Struktura liniowa 64
Struktura liniowa z alternatywą 64
Struktura mieszana (liniowo-hierarchiczna) 65
Struktura sieciowa (pajęczyna) 67
Scenopis witryny WWW 69
Co to jest scenopis i do czego może się przydać? 69
Kilka wskazówek dotyczących scenopisów 70
Podsumowanie 71
Warsztat 72
Pytania i odpowiedzi 72
Quiz 72
Odpowiedzi 73
Ćwiczenia 73

Rozdział 3. Wprowadzenie do HTML 75
Czym jest, a czym nie jest HTML? 75
HTML opisuje strukturę strony 75
HTML nie opisuje układu strony 76
Dlaczego akurat tak? 77
HTML jest językiem znaczników 78
Krótka historia znaczników HTML 78
Przyszłość z XHTML 1.0 80
Jak wygląda plik HTML 80
Krótka uwaga na temat formatowania 84
Programy pomocne w tworzeniu plików HTML 85
Podsumowanie 86
Warsztat 86
Pytania i odpowiedzi 87
Quiz 87
Odpowiedzi 87
Ćwiczenia 88

Część II Tworzenie prostej strony WWW 91

Rozdział 4. Zaczynamy od podstaw 93
Struktura HTML-a 93
Znacznik <html> 94
Znacznik <head> 94
Znacznik <body> 95

Tytuł....95
Nagłówki....97
Akapity....98
Listy, listy i jeszcze raz listy....100
Znaczniki list....100
Listy numerowane....101
Formatowanie list numerowanych w HTML 3.2....102
Listy wypunktowane....105
Formatowanie list wypunktowanych w HTML-u 3.2....105
Listy definicji....106
Listy zagnieżdżone....108
Komentarze....109
Podsumowanie....111
Warsztat....111
Pytania i odpowiedzi....111
Quiz....113
Odpowiedzi....114
Ćwiczenia....114

Rozdział 5. Wszystko o połączeniach....115
Tworzenie hiperpołączeń....115
Znacznik <a>....115
Łączenie stron lokalnych przy użyciu ścieżek względnych i bezwzględnych....120
Ścieżki bezwzględne....122
Z którego rodzaju ścieżek powinieneś korzystać?....122
Połączenia do dokumentów w sieci WWW....123
Połączenia do określonych miejsc w dokumencie....127
Tworzenie połączeń i odnośników....128
Połączenia wewnątrz tego samego dokumentu....134
Anatomia adresu URL....134
Z czego składa się URL....135
Znaki specjalne w adresach URL....136
Rodzaje adresów URL....137
HTTP....137
Anonimowe FTP....137
FTP o ograniczonym dostępie....138
Mailto....139
Gopher....139
Usenet....139
File....140
Podsumowanie....141
Warsztat....142
Pytania i odpowiedzi....142
Quiz....144
Odpowiedzi....145
Ćwiczenia....145

Rozdział 6. Więcej o formatowaniu tekstu w HTML-u 147
Style znaków 148
Style logiczne 148
Style fizyczne 151
Tekst preformatowany 152
Linie poziome 155
Atrybuty znacznika <hr> 156
Łamanie linii 159
Adresy 160
Cytaty 160
Znaki specjalne 161
Kody znaków specjalnych 162
Kody znaków zarezerwowanych 163
Położenie tekstu na stronie 164
Kontrola położenia pojedynczych elementów 164
Kontrola położenia grup elementów 164
Czcionki i ich rozmiar 166
Zmiana rozmiaru czcionki 166
Zmiana rodzaju czcionki 167
Migający <blink> 168
<nobr> i <wbr> 168
Podsumowanie 178
Warsztaty 182
Pytania i odpowiedzi 182
Quiz 183
Odpowiedzi 183
Ćwiczenia 184

Część III Grafika na WWW 185

Rozdział 7. Wykorzystywanie obrazów, koloru i tła 187
Grafika w sieci WWW 187
Obrazy wewnętrzne na stronach WWW: znacznik <img> 189
Dodawanie tekstu alternatywnego 189
Obrazy i tekst 193
Wzajemne położenie tekstu i obrazu 194
Obraz otoczony tekstem 196
Określanie odległości obrazu od tekstu 199
Grafika i połączenia 200
Obrazy zewnętrzne 203
Inne sztuczki z wykorzystaniem obrazów 207
Wymiary i skalowanie obrazów 207
Kilka słów na temat ramek wokół obrazów 208
Podgląd obrazu 208
Kolory 209
Nazwy kolorów 209
Zmiana koloru tła 210

Zmiana koloru tekstu ... 211
Kolor pojedynczego znaku ... 212
Grafika w tle ... 213
Kilka pożytecznych wskazówek ... 214
Czy ten rysunek jest niezbędny? ... 214
Obrazy nie powinny być duże ... 215
Staraj się wielokrotnie wykorzystywać te same obrazy ... 215
Zamiast grafiki 216
Podsumowanie ... 216
Warsztat ... 217
Pytania i odpowiedzi ... 218
Quiz ... 219
Odpowiedzi ... 219
Ćwiczenia ... 220

Rozdział 8. Tworzenie obrazów na potrzeby sieci ... 221
Formaty obrazów ... 221
GIF ... 221
JPEG ... 222
Kolor ... 223
System barwny HSB ... 223
System barwny RGB ... 225
Formaty map bitowych i palety kolorów ... 226
Wyświetlanie kolorów ... 230
Bezpieczne palety kolorów ... 231
Kompresja obrazów ... 232
Podstawowe wiadomości o kompresji ... 233
Algorytmy kompresji formatu GIF i JPEG ... 234
Wyświetlanie skompresowanych plików ... 235
Przeplot obrazu i przezroczystość tła ... 238
Przezroczystość ... 238
Przeplot w formacie GIF ... 239
Wyświetlanie progresywne w formacie JPEG ... 242
Tworzenie obrazów o przejrzystym tle oraz obrazów z przeplotem ... 242
Animowane pliki GIF ... 243
Tworzenie i wykorzystanie obrazów ... 243
Własne projekty ... 244
Gotowe zestawy obrazów ... 245
Kliparty otrzymywane za pośrednictwem sieci ... 246
Obrazy z innych stron sieci ... 246
Format przyszłości: PNG ... 247
Źródła informacji ... 248
Podsumowanie ... 249
Warsztat ... 249
Pytania i odpowiedzi ... 250
Quiz ... 251
Odpowiedzi ... 251
Ćwiczenia ... 251

Rozdział 9. Tworzenie animowanych obrazów 253
Czym jest animacja? 253
Kiedy i gdzie używać animacji 254
Tworzenie przezroczystych obrazów GIF 256
Wybór przezroczystego koloru 257
Programy umożliwiające kompilację animowanych GIF-ów 258
GIF Construction Set (dla systemów Windows) 258
GifBuilder (dla systemu Macintosh) 259
Ograniczanie wielkości animacji 261
Narzędzia pomocne przy tworzeniu obrazów 264
Przydatne możliwości programów 266
Podsumowanie 272
Warsztat 272
Pytania i odpowiedzi 273
Quiz 273
Odpowiedzi 273
Ćwiczenia 274

Część IV Arkusze stylów, tabele i ramki 275

Rozdział 10. XHTML i arkusze stylów 277
Czym jest XHTML i dlaczego należy go używać? 277
Identyfikator <DOCTYPE> 278
Składnia XHTML 279
Idea arkuszy stylów 280
Dodatkowe informacje o arkuszach stylów 282
Księgarnia Bookworm 283
Sposoby tworzenia arkuszy stylów 286
Zewnętrzne arkusze stylów 286
Osadzone arkusze stylów 290
Kilka słów o kaskadzie stylów 293
Najczęściej stosowane właściwości stylów oraz ich wartości 293
Kontrola układu strony przy użyciu właściwości CSS 294
Tło, kolory i obrazy 296
Określanie wyglądu obramowań 300
Określanie czcionek i stylów 303
Właściwości CSS określające wyrównanie tekstu 305
Style lokalne 306
Źródła informacji o kaskadowych arkuszach stylów 314
Warsztat 314
Pytania i odpowiedzi 314
Quiz 315
Odpowiedzi 315
Ćwiczenia 316

Rozdział 11. Tabele 317
Definicja tabeli....318
Tworzenie tabel....318
Części tabeli....318
Znacznik <table>....319
Wiersze i komórki....320
Puste komórki....322
Podpisy....323
Określanie wielkości tabel, obramowań oraz komórek....326
Określanie szerokości tabeli....326
Zmiana obramowań tabeli....327
Odstępy między zawartością komórki a jej obramowaniem....329
Odstępy między komórkami....329
Szerokość kolumn....330
Wprowadzanie złamań wiersza....331
Kolor i wyrównanie tabeli oraz komórek....333
Modyfikacja koloru tła tabel i komórek....333
Zmiana koloru obramowań tabeli....335
Określanie wyrównania zawartości tabel....336
Wyrównywanie tabeli....336
Wyrównanie zawartości komórki....337
Wyrównywanie tytułu....340
Komórki rozpięte na kilku wierszach lub kolumnach....340
Bardziej zaawansowane możliwości ulepszania tabel....348
Grupowanie i określanie wyrównania kolumn....349
Grupowanie i określanie wyrównania wierszy....352
Atrybuty FRAME oraz RULES....354
Inne elementy i atrybuty tabel....356
Podsumowanie....356
Warsztat....360
Pytania i odpowiedzi....360
Quiz....361
Odpowiedzi....361
Ćwiczenia....361

Rozdział 12. Ramki i połączenia do nich....363
Czym są ramki i gdzie są obsługiwane....363
Połączenia do ramek....364
Znacznik <base>....368
Posługiwanie się ramkami....369
Znacznik <frameset>....370
Znacznik <frame>....372
Znacznik <noframes>....373
Zmiana obramowań ramek....374
Atrybuty dodatkowe....375
Tworzenie złożonych układów ramek....375
Magiczne nazwy target....386

Ramki lokalne ... 387
Podsumowanie ... 389
Warsztat ... 392
Pytania i odpowiedzi ... 392
Quiz ... 392
Odpowiedzi ... 392
Ćwiczenia ... 393

Część V Multimedia i aplety Javy ... 395

Rozdział 13. Multimedia: dodawanie dźwięków, obrazów wideo i innych elementów multimedialnych ... 397
Przedstawienie sposobów prezentacji dźwięków oraz obrazów wideo ... 398
Stare, lecz użyteczne rozwiązanie — dołączanie ... 399
Osadzanie dźwięków i obrazów wideo ... 407
Wykorzystanie elementu embed ... 408
Użycie elementu object ... 410
Łączenie elementów embed i object ... 412
Osadzanie animacji Flash ... 412
Osadzanie animacji Shockwave ... 413
Osadzanie RealAudio oraz RealVideo ... 415
Techniki multimedialne wykorzystywane w Microsoft Internet Explorerze ... 416
Typy plików dźwiękowych i wideo ... 419
O dodatkach i odtwarzaczach ... 424
Windows Media Player ... 425
Flash firmy Macromedia ... 426
Shockwave firmy Macromedia ... 426
QuickTime firmy Apple ... 426
Produkty RealNetworks ... 427
Inne dodatki ... 428
Podsumowanie ... 429
Warsztat ... 429
Pytania i odpowiedzi ... 430
Quiz ... 431
Odpowiedzi ... 431
Ćwiczenia ... 431

Rozdział 14. Dźwięk i obraz wideo ... 433
Dźwięk — wiadomości podstawowe ... 433
Fale dźwiękowe ... 434
Konwersja fali dźwiękowej do postaci cyfrowej ... 434
Odtwarzanie dźwięku cyfrowego w postaci analogowej ... 437
Popularne formaty zapisu dźwięku ... 437
Format μ-law (Mu-law), AU ... 438
Format AIFF/AIFC ... 438
Format SND systemu Macintosh ... 439
Format WAVE systemu Windows ... 439
Format MPEG Audio ... 439

Format RealAudio 440
Windows Media....440
Skąd brać pliki dźwiękowe? 440
Próbkowanie dźwięku....441
Próbkowanie na PC 441
Próbkowanie na Macintoshach 443
Próbkowanie w systemach uniksowych 444
Konwersja plików dźwiękowych....444
Dźwięk w sieci....444
Cyfrowy obraz wideo....445
Analogowy i cyfrowy obraz wideo 446
Kompresja i dekompresja obrazu wideo (Codec) 447
Formaty plików....448
Format QuickTime 448
Format Video for Windows (AVI) 448
Format MPEG Video....449
RealVideo....449
Windows Media....450
Kompresja obrazu wideo 450
Kodowanie symetryczne i asymetryczne....450
Porównywanie ujęć 451
Ujęcia kluczowe 452
Wspomaganie sprzętowe 453
Systemy kodowania 453
Algorytm CinePak 453
Algorytm Indeo 454
Algorytm JPEG 454
Algorytm MPEG 455
Digitalizacja obrazu wideo 455
Analogowy sygnał wideo 455
Obraz wideo na PC 456
Obraz wideo na Mac-ach....458
Obraz wideo w systemach uniksowych 458
Uzyskiwanie i konwersja plików wideo 458
Pliki wideo w sieci....459
Źródła informacji 461
Podsumowanie 461
Warsztat 462
Pytania i odpowiedzi 462
Quiz 463
Odpowiedzi 463
Ćwiczenia 463

Rozdział 15. Wykorzystywanie Javy 465

Czym jest Java? 465
Język Java....466
Aplety pisane w Javie....466
Do czego można użyć Javy?....467

Programowanie w Javie 469
Java Development Kit (JDK) 469
Dołączanie apletów do stron WWW 478
Znacznik <applet> 478
Znacznik <param> 479
Zastępowanie apletów kodem HTML 480
Rozbudowywanie przykładu Ticker 480
Stosowanie gotowych apletów 480
Podsumowanie 481
Warsztat 482
Pytania i odpowiedzi 482
Quiz 482
Odpowiedzi 483
Ćwiczenia 483

Część VI Mapy odsyłaczy i formularze 485

Rozdział 16. Tworzenie i stosowanie map odsyłaczy 487
Czym są mapy odsyłaczy 487
Mapy odsyłaczy działające po stronie serwera 488
Mapy odsyłaczy działające po stronie klienta 489
Mapy odsyłaczy a przeglądarki tekstowe 489
Tworzenie map odsyłaczy działających po stronie serwera 490
Wybór obrazu 490
Definiowanie koordynat 490
Tworzenie i zapisywanie pliku mapy odsyłaczy 495
Instalowanie programu obsługi map odsyłaczy 496
Łączenie wszystkiego razem 496
Tworzenie map odsyłaczy obsługiwanych po stronie przeglądarki 502
Znaczniki <map> i <area> 502
Atrybut usemap 503
Tworzenie stron WWW obsługujących oba typy map odsyłaczy 505
Podsumowanie 506
Warsztat 507
Pytania i odpowiedzi 507
Quiz 508
Odpowiedzi 508
Ćwiczenia 509

Rozdział 17. Projektowanie formularzy 511
Forma i funkcja formularzy 512
Podstawowe elementy formularzy 518
Zastosowanie znacznika <form> do tworzenia formularzy 518
Tworzenie elementów kontrolnych przy wykorzystaniu elementu input 520
Zastosowanie innych elementów kontrolnych 533

Dodatkowe elementy ... 537
Wyświetlanie etykiet ... 537
Grupowanie elementów przy wykorzystaniu znaczników <fieldset> oraz <legend> ... 538
Modyfikacja domyślnej kolejności poruszania się pomiędzy elementami ... 539
Stosowanie klawiszy dostępu ... 540
Tworzenie elementów wyłączonych i przeznaczonych tylko do odczytu ... 540
Planowanie formularzy ... 545
Podsumowanie ... 545
Warsztat ... 547
Pytania i odpowiedzi ... 547
Quiz ... 548
Odpowiedzi ... 548
Ćwiczenia ... 549

Rozdział 18. Skrypty CGI dla początkujących ... 551
Co to jest skrypt CGI? ... 552
Jak działają skrypty? ... 552
Prosty przykład ... 553
Czy mogę używać skryptów CGI? ... 555
Czy Twój serwer jest skonfigurowany w sposób pozwalający na wykonanie skryptów CGI? ... 555
Czy umiesz programować? ... 556
Jakich języków programowania należy używać? ... 556
Czy Twój serwer jest skonfigurowany prawidłowo? ... 556
A jeśli nie korzystasz z systemu Unix? ... 557
Anatomia skryptu CGI ... 558
Nagłówki odpowiedzi ... 558
Dane zwracane w odpowiedzi ... 559
Skrypty z parametrami ... 562
Przesyłanie innych informacji do skryptu ... 563
Generowanie specjalnych odpowiedzi ... 564
Odpowiedź polegająca na odesłaniu innego dokumentu ... 564
Brak odpowiedzi ... 565
Skrypty przetwarzające formularze ... 565
Formularze i skrypty je przetwarzające ... 566
GET i POST ... 566
Kodowanie URL ... 567
Najczęstsze problemy ... 570
Zmienne CGI ... 571
Programy dekodujące dane z formularzy ... 571
uncgi ... 571
CGI.pm ... 573
Samodzielne dekodowanie danych z formularzy ... 574
Skrypty bez przetwarzania nagłówków ... 575
Skrypty isindex ... 576
Podsumowanie ... 578
Warsztat ... 578
Pytania i odpowiedzi ... 578
Quiz ... 580

Odpowiedzi 580
Ćwiczenia 580

Część VII JavaScript i Dynamiczny HTML 581

Rozdział 19. Tworzenie skryptów w języku JavaScript....583
Przedstawienie języka JavaScript 584
Dlaczego miałbyś używać języka JavaScript? 585
Znacznik <script> 587
Struktura skryptów pisanych w języku JavaScript 587
Atrybut SRC 588
Podstawowe polecenia i struktura języka 588
Właściwości i metody 589
Zdarzenia i JavaScript 591
Zmienne 593
Operatory i wyrażenia 594
Podstawy programowania w języku JavaScript 595
Czym jest program? 595
Dowiedz się więcej o programowaniu w JavaScript 598
Podsumowanie 598
Warsztat 599
Pytania i odpowiedzi 599
Quiz 600
Odpowiedzi 600
Ćwiczenia 601

Rozdział 20. Praca z JavaScript....603
Tworzenie generatora losowych połączeń 603
Weryfikacja danych w formularzach 612
Tworzenie podmienianych obrazków 618
Podsumowanie 622
Warsztat 622
Pytania i odpowiedzi 622
Quiz 623
Odpowiedzi 623
Ćwiczenia 623

Rozdział 21. Użycie Dynamicznego HTML–a....625
Czym w zasadzie jest DHTML? 626
Prezentacja języka JavaScript 629
Wykorzystanie modelu obiektów dokumentu 638
Obsługa zdarzeń 642
Walka z rzeczywistością: techniki DHTML działające w wielu przeglądarkach 643
Określanie przeglądarki 644
Tworzenie odwołań DOM działających w wielu przeglądarkach 646
Dalsze poznawanie DHTML 681
Podsumowanie 681

Warsztat 682
Pytania i odpowiedzi 682
Quiz 683
Odpowiedzi 683
Ćwiczenia 684

Część VIII Projektowanie efektywnych stron WWW 685

Rozdział 22. Tworzenie i projektowanie stron WWW: zalecenia i przeciwwskazania 687

Stosowanie rozszerzeń HTML 688
Publikowanie w sieci 691
Pisz jasno i zwięźle 691
Organizuj strony w sposób przejrzysty 692
Twórz strony, będące samodzielnymi całościami 693
Ostrożnie stosuj wyróżnienia 694
Nie stosuj terminologii typowej dla danego typu przeglądarek 694
Sprawdzaj ortografię i składnię 695
Projektowanie i układ strony 695
Stosuj nagłówki jako nagłówki 696
Grupuj informacje wiążące się ze sobą 696
Stosuj spójny układ 697
Stosowanie połączeń 698
Dbaj o to, by tekst prezentacyjny połączeń zawartych w menu był opisowy 698
Umieszczaj połączenia w tekście 699
Unikaj używania słowa „tutaj” 700
Połączenia — tworzyć czy nie tworzyć 701
Obrazy 703
Nie przesadź z liczbą obrazów 704
Stosuj tekst alternatywny 705
Stosuj małe obrazy 705
Ostrożnie z założeniami 706
Nie przesadzaj z kolorami tła i połączeń 707
Inne dobre zwyczaje i podpowiedzi 708
Umieszczaj połączenie do strony głównej 708
Nie rozdzielaj tematu między strony 708
Wyważ właściwie liczbę stron 709
Podpisuj swoje strony 710
Udostępniaj wersje stron HTML nie korzystające z hipertekstu 711
Podsumowanie 712
Warsztat 713
Pytania i odpowiedzi 713
Quiz 714
Odpowiedzi 714
Ćwiczenia 715

Rozdział 23. Przykłady dobrych i złych stron WWW.....717
Przykład 1. Witryna firmowa.....717
Co nowego w firmie?.....718
Informacje techniczne.....719
Informacje o firmie.....721
Produkty.....721
Wsparcie dla produktów.....722
Możliwości pracy.....722
Cechy zaprezentowanej witryny i szczegóły projektowe.....724
Przykład 2. Encyklopedia multimedialna.....724
Strona przeglądowa.....725
Pozycja Norton.....725
Indeks marek.....728
Cechy zaprezentowanej witryny i szczegóły projektowe.....728
Przykład 3. Katalog wysyłkowy.....730
Przeglądanie katalogu.....731
Zamawianie.....734
Cechy zaprezentowanej witryny i szczegóły projektowe.....739
Podsumowanie.....741
Warsztat.....741
Pytania i odpowiedzi.....741
Quiz.....742
Odpowiedzi.....742
Ćwiczenia.....743

Rozdział 24. Projektowanie witryn do praktycznych zastosowań.....745
Czym są „zastosowania praktyczne”.....745
Uwzględnienie poziomu doświadczenia użytkownika.....747
Podaj listę stron zawierających opis witryny.....748
Opisz system nawigacyjny.....748
Dodaj informacje o zalecanych przeglądarkach.....751
Lista innych niezbędnych plików.....752
Określanie preferencji użytkowników.....752
Podejmowanie decyzji o użyciu języka HTML 4.0 bądź XHTML 1.0.....754
HTML 4.0 oraz XHTML 1.0 Transitional (pośredni).....754
HTML 4.0 i XHTML 1.0 Frameset (układ ramek).....757
HTML 4.0 i XHTML 1.0 Strict (ścisły).....759
Podsumowanie.....763
Warsztat.....763
Pytania i odpowiedzi.....763
Quiz.....764
Odpowiedzi.....764
Ćwiczenia.....765

Część IX Publikowanie na WWW 767

Rozdział 25. Publikowanie witryny 769
Jak działa serwer WWW? 769
Co jeszcze robią serwery? 770
Znajdowanie serwera WWW 772
Korzystanie z serwera WWW w szkole lub w pracy 772
Korzystanie z usług komercyjnych 772
Zakładanie własnego serwera 773
Organizowanie i instalowanie plików HTML 774
Pytania do Webmastera 774
Domyślny plik indeksu i poprawne nazwy plików 775
Instalowanie plików 776
Przemieszczanie plików pomiędzy systemami 777
Narzędzia zdalnego zarządzania plikami 778
Jaki jest mój adres URL? 779
Testuj, testuj i jeszcze raz testuj! 780
Usuwanie błędów 780
Nie mam dostępu do serwera 780
Nie mam dostępu do plików 781
Nie mam dostępu do obrazów 781
Połączenia nie działają poprawnie 781
Pliki nie są wyświetlane poprawnie 782
Podsumowanie 782
Warsztat 782
Pytania i odpowiedzi 783
Quiz 784
Odpowiedzi 784
Ćwiczenia 784

Rozdział 26. Rejestrowanie i reklamowanie stron WWW 785
Rejestracja i reklamowanie stron WWW 785
Listy witryn WWW 786
Yahoo! 786
W3 Virtual Library 788
Żółte Strony (Yellow Pages) 790
Prywatne serwisy katalogowe 791
Serwisy indeksujące i wyszukujące 791
AltaVista 791
Excite 792
Hotbot 793
Lycos 793
Go.com 794
Narzędzia automatycznej rejestracji 795
Reklamowanie witryny za pośrednictwem sieci Usenet 796
Pierścienie Internetowe (Ringi) 797
Wizytówki, papeteria firmowa i broszury 798

Informacje o czytelnikach ... 799
Raporty ... 799
Liczniki wizyt na stronie ... 800
Podsumowanie ... 801
Warsztat ... 801
Pytania i odpowiedzi ... 801
Quiz ... 802
Odpowiedzi ... 802
Ćwiczenia ... 802

Rozdział 27. Testowanie, poprawianie i aktualizowanie stron WWW ... 803
Test poprawności ... 804
Korekta kodu HTML ... 804
Testowanie w przeglądarkach ... 814
Sprawdzanie połączeń ... 815
Testowanie użyteczności strony ... 816
Studiowanie raportów ... 817
Uaktualnianie prezentacji i dodawanie nowych stron ... 817
Dodawanie nowej zawartości ... 817
Rewidowanie struktury ... 818
Podsumowanie ... 819
Warsztat ... 820
Pytania i odpowiedzi ... 820
Quiz ... 820
Odpowiedzi ... 821
Ćwiczenia ... 822

Część X Konfiguracja i administracja serwera WWW ... 823

Rozdział 28. Konfiguracja własnego serwera WWW ... 825
Zalety i wady posiadania własnego serwera WWW ... 826
Określenie odpowiedniego komputera ... 826
Określenie odpowiedniego połączenia z Internetem ... 827
Współpraca z dostawcą usług internetowych ... 828
Wykorzystanie istniejącego połączenia ... 828
Oprogramowanie ... 829
Serwery przeznaczone dla systemów Unix ... 830
NCSA HTTPD ... 830
Apache ... 830
W3 (CERN) HTTPD ... 831
Web Server firmy Netscape ... 832
Serwery przeznaczone dla systemu Windows ... 833
Microsoft Internet Information Server ... 833
Microsoft Personal Web Server ... 835
WebSite firmy O'Reilly ... 835
Serwery firmy Netscape przeznaczone dla systemu Windows ... 837
Apache for Windows ... 837

Serwery działające na komputerach Macintosh ... 838
Personal Web Sharing firmy Apple ... 838
Podpowiedzi związane z poprawnym zarządzaniem serwerem ... 839
Skojarz adres komputera z nazwą witryny — www.twoja_witryna.com ... 839
Stwórz adres poczty elektronicznej administratora ... 839
Stwórz stronę główną serwera ... 840
Określ wytyczne dotyczące zarządzania i projektu witryny ... 840
Podsumowanie ... 840
Warsztat ... 840
Pytania i odpowiedzi ... 841
Quiz ... 843
Odpowiedzi ... 843

Rozdział 29. Porady i wskazówki na temat serwera WWW ... 845
Mechanizm NCSA SSI ... 845
Konfiguracja serwera ... 846
Tworzenie plików z poleceniami SSI ... 848
Konfiguracja SSI ... 848
Włączanie innych plików do stron WWW ... 849
Dołączanie wartości zmiennych ... 850
Dołączanie informacji o pliku ... 852
Wyniki działania poleceń i skryptów jako część stron WWW ... 853
Przekierowanie pliku ... 856
Logi serwera WWW ... 858
Logi serwera oraz standardowy format logu ... 858
Kilka uwag na temat buforowania ... 859
Tworzenie statystyk na podstawie logów ... 861
User-agent i odnośniki ... 862
Własne dokumenty obsługi błędów ... 863
Programy CGI jako dokumenty obsługi błędów ... 864
Podsumowanie ... 864
Warsztat ... 865
Pytania i odpowiedzi ... 865
Quiz ... 866
Odpowiedzi ... 866
Ćwiczenia ... 867

Rozdział 30. Bezpieczeństwo serwera WWW i kontrola dostępu ... 869
Jak lepiej zabezpieczyć serwer WWW? ... 870
Uruchom serwer jako użytkownik nobody ... 871
Ogranicz dostęp do skryptów CGI ... 871
Ogranicz zastosowanie połączeń symbolicznych ... 872
Wyłącz SSI ... 872
Wyłącz wyświetlanie zawartości katalogów ... 873
Odetnij robotom sieciowym dostęp do swojego serwera ... 873
Jak pisać bezpieczne skrypty CGI ... 874

Kontrola dostępu do serwera WWW i autoryzacja — wprowadzenie 876
Co oznacza kontrola dostępu i autoryzacja?....877
Rodzaje kontroli dostępu....878
Na ile bezpieczny jest serwer WWW? 879
Kontrola dostępu i autoryzacja na serwerze NCSA HTTPD....880
Globalna i lokalna kontrola dostępu....880
Kontrola dostępu ze względu na nazwę komputera....882
Ustawianie pliku haseł....883
Kontrola dostępu na poziomie użytkownika....884
Ustawienia pliku grupy....886
Ograniczanie dostępu dla grupy 886
Opcje NCSA 887
Unieważnianie ustawień opcji i kontroli dostępu 888
Kontrola dostępu w serwerze Microsoft Internet Information Server 889
Bezpieczne połączenia i SSL....890
Jak działa SSL 890
Jak powstają połączenia SSL....893
Ustawienia SSL na serwerze 893
Więcej informacji o SSL 894
Podsumowanie....894
Warsztat....895
Pytania i odpowiedzi 895
Quiz....896
Odpowiedzi 896
Ćwiczenia....897

Dodatki....899

Dodatek A Inne źródła informacji....901
Analizatory składni plików Log....902
Dostawcy usług sieciowych....902
Dźwięk i obraz wideo 903
Edytory i konwertery HTML....903
Formularze i mapy odsyłaczy 904
Indeksy i wyszukiwarki sieci WWW....905
Interfejs Common Gateway (CGI) oraz skrypty CGI....906
Java, JavaScript i obiekty osadzone....906
Liczniki dostępu....907
Narzędzia i informacje dotyczące obrazów 907
Narzędzia sprawdzające HTML (walidatory), narzędzia sprawdzające łącza i proste pająki (roboty) sieciowe....908
Pozostałe....909
Przeglądarki....909
Przyszłość HTML i sieci WWW 910
Serwery i administrowanie siecią 910
Specyfikacje dla HTML, HTTP i URL....911
Zbiory informacji na temat tworzenia dokumentów HTML i WWW 912

Dodatek B Krótki opis języka HTML 4.0 915
Najczęściej używane atrybuty i zdarzenia 916
%coreattrs 916
%i18n 917
%events 917
Struktura 917
<bdo>…</bdo> 917
<body>...</body> 918
Komentarze <!-- … --> 918
… 919
<!DOCTYPE…> 919
<h1>…</h1> do <h6>…</h6> 919
<head>…</head> 920
<hr> 920
<html>…</html> 921
<meta>…</meta> 921
… 921
<title>…</title> 922
Wyrażenia tekstowe i akapity 922
<acronym>…</acronym> 922
<address>…</address> 922
<blockquote>…</blockquote> 923

 923
<cite>…</cite> 923
<code>…</code> 924
~~…~~ 924
<dfn>…</dfn> 924
… 925
<ins>…</ins> 925
<kbd>…<kbd> 925
… 925
<pre>…</pre> 926
<q>…</q> 926
<samp>…</samp> 926
… 927
_… 927
[…] 927
<var>…</var> 927
Elementy formatujące tekst 928
… 928
<basefont> 928
<big>…</big> 928
… 929
… 929
<s>…</s> 929
<small>…</small> 929
<strike>…</strike> 930

<tt>…</tt> 930
<u>…</u> 930
Listy 930
<dd>…</dd> 931
<dir>…</ dir > 931
<dl>…</dl> 931
<dt>…</dt> 932
… 932
<menu>…</menu> 932
… 933
… 933
Połączenia 933
<a>… 933
<base> 934
<link> 935
Tabele 935
<caption>…</caption> 935
<col> 936
<colgroup>…</colgroup> 936
<table>…</table> 937
<tbody>…</tbody> 937
<td>…</td> 938
<tfoot>…</tfoot> 938
<th>…</th> 939
<thead>…</thead> 939
<tr>…</tr> 940
Ramki 940
<frame> 941
<frameset>…</frameset> 941
<iframe>…</iframe> 942
<noframes>…</noframes> 942
Obiekty osadzone 942
<applet>…</applet> 943
<area>…</area> 943
<img>…</img> 944
<map>…</map> 944
<object>…</object> 945
<param>…</param> 946
Style 946
<style>…</style> 946
Formularze 947
<button>…</button> 947
<fieldset>…</fieldset> 947
<form>…</form> 947
<input> 948
<isindex> 949
<label>…</label> 950

<legend>…</legend>....950
<option>…</option>....950
<select>…</select>....951
<textarea>…</textarea>....951
Skrypty....952
<script>…</script>....952
<noscript>…</noscript>....953
Znaki....953

Dodatek C Porównanie możliwości przeglądarek....959
Tabela możliwości przeglądarek....959
Zdarzenia wbudowane....1008

Dodatek D Krótki przewodnik po kaskadowych arkuszach stylów (CSS) 1011
Jak korzystać z tego dodatku....1012
Właściwości bloku....1013
bottom, left, right, top....1013
direction....1014
display....1014
float....1014
position....1015
unicode-bidi....1015
z-index....1015
Właściwości tła i koloru....1016
background....1016
background-attachment....1016
background-color....1016
background-image....1017
background-position....1017
background-repeat....1017
color....1018
Właściwości modelu blokowego....1018
border....1018
border-bottom, border-left, border-right, border-top....1019
border-color....1019
border-bottom-color, border-left-color, border-right-color, border-top-color....1020
border-style....1020
border-bottom-style, border-left-style, border-right-style, border-top-style....1020
border-width....1021
border-bottom-width, border-left-width, border-right-width, border-top-width....1021
clear....1021
height, width....1022
margin....1022
margin-bottom, margin-left, margin-right, margin-top....1022
max-height, max-width....1023
min-height, min-width....1023
padding....1023
padding-top, padding-left, padding-bottom, padding-right....1024

Właściwości czcionek ... 1024
font ... 1024
font-family ... 1025
font-size ... 1025
font-size-adjust ... 1026
font-stretch ... 1026
font-style ... 1026
font-variant ... 1027
font-weight ... 1027
Właściwości list ... 1027
list-style ... 1028
list-style-image ... 1028
list-style-position ... 1028
list-style-type ... 1029
Właściwości tekstu ... 1029
letter-spacing ... 1029
line-height ... 1029
text-align ... 1030
text-decoration ... 1030
text-indent ... 1030
text-shadow ... 1031
text-transform ... 1031
vertical-align ... 1031
white-space ... 1032
word-spacing ... 1032
Właściwości efektów wizualnych ... 1032
clip ... 1032
overflow ... 1033
visibility ... 1033
Głosowe właściwości arkuszy stylów ... 1033
azimuth ... 1033
cue ... 1034
cue-after, cue-before ... 1034
elevation ... 1034
pause ... 1035
pause-after, pause-before ... 1035
pitch ... 1035
pitch-range ... 1035
play-during ... 1036
richness ... 1036
speak ... 1036
speak-header ... 1037
speak-numeral ... 1037
speak-punctuation ... 1037
speech-rate ... 1037
stress ... 1038
voice-family ... 1038
volume ... 1038
Właściwości zawartości wygenerowanej/automatycznego numerowania ... 1039
content ... 1039
counter-increment ... 1039

counter-reset 1040
marker-offset 1040
quotes 1040
Właściwości mediów stronicowanych 1040
marks 1041
orphans 1041
page 1041
page-break-after, page-break-before 1042
page-break-inside 1042
size 1042
widows 1042
Właściwości tabel 1043
border-collapse 1043
border-spacing 1043
caption-side 1043
column-span, row-span 1044
empty-cells 1044
table-layout 1044
Właściwości interfejsu użytkownika 1045
cursor 1045
outline 1045
outline-color 1045
outline-style 1046
outline-width 1046
Jednostki stosowane w kaskadowych arkuszach stylów 1046
<border-style> 1046
<border-width> 1047
<czas> 1047
<częstotliwość> 1048
<długość> 1048
<głos-ogólny> 1048
<kąt> 1049
<kolor> 1049
<konkretny-głos> 1049
<kształt> 1049
<liczba> 1050
<liczba_całkowita> 1050
<nazwa-rodziny> 1050
<rodzina-ogólna> 1050
<uri> 1050
<wartość_procentowa> 1051
<wielkość_bezwzględna> 1051
<wielkość_względna> 1051

Dodatek E Informacje o języku JavaScript 1053
Słów kilka o języku JavaScript 1.2 1053
Obiekt anchor [C|2|3|4|I] 1054
Obiekt applet [C|3] 1054
Obiekt area [C|3] 1054

Obiekt Array [C|3|I] 1055
Obiekt boolean [C|3|I] 1056
Obiekt button [C|2|3|I] 1056
Obiekt checkbox [C|2|3|I] 1057
Obiekt date [C|2|3|I] 1058
Obiekt document [C|2|3|I] 1059
Objekt event [C] 1062
Obiekt fileUpload [C|3] 1062
Obiekt form [C|2|3|I] 1063
Obiekt frame [C|2|3|I] 1064
Obiekt Fuction [C|3] 1066
Obiekt hidden [C|2|3|I] 1067
Obiekt history [C|2|3|I] 1067
Obiekt image [C|3] 1068
Obiekt layer [4] Tylko w Navigatorze 4.x 1069
Obiekt link [C|2|3|I] 1071
Obiekt location [C|2|3|I] 1072
Obiekt math [C|2|3|I] 1073
Obiekt mimeType [C|3] 1074
Obiekt navigator [C|2|3|I] 1075
Obiekt number [C|3|I] 1076
Obiekt object [C|2|3|I] 1077
Obiekt option [C|3] 1077
Obiekt password [C|2|3|I] 1078
Obiekt plugin 1078
Obiekt radio [C|2|3|I] 1079
Obiekt regExp 1080
Obiekt reset [C|2|3|I] 1081
Obiekt screen (Nowy w języku JavaScript 1.2) 1082
Obiekt select [C|2|3] 1082
Obiekt string [C|2|3|I] 1083
Obiekt submit [C|2|3|I] 1086
Obiekt text [C|2|3|I] 1086
Obiekt textarea [C|2|3|I] 1087
Obiekt window [C|2|3|I] 1088
Funkcje niezależne, operatory, zmienne i literały 1094
Funkcje 1094
Instrukcje 1095
Operatory 1096
Hierarchia operatorów 1099
Dodatek F Znaki 1101
Dodatek G Nazwy kolorów i ich odpowiedniki w systemie szesnastkowym 1107
Dodatek H Typy MIME i rozszerzenia plików 1111
Skorowidz 1115

Rozdział 10.
XHTML i arkusze stylów

W trzech ostatnich rozdziałach mogłeś zauważyć odwołania do znaczników HTML, które często określałam jako „przestarzałe w HTML-u 4.0 i zastępowane *arkuszami stylów*". Nadszedł czas, aby rozwiać tę tajemnicę i pokazać, jakie są tendencje rozwoju HTML-a. W tym rozdziale poznasz sposób na formatowanie i projektowanie stron opracowany przez World Wide Web Consortium. W rozdziale 3. — „Wprowadzenie do HTML" dowiedziałeś się, że HTML jest językiem znacznikowym, którego zadaniem jest opis struktury strony, a nie jej wyglądu. HTML nigdy nie został zaprojektowany w celu określania wyglądu strony (wyboru czcionek, kolorów czy też odległości pomiędzy wyrazami bądź literami), został on stworzony w celu opisu elementów tworzących stronę (nagłówków, teksu, obrazów, itp.). Rozszerzenia standardowych znaczników HTML (<font>, <color>, <margin>, itd.) udostępniły autorom stron WWW możliwości wykraczające poza początkowe zamierzenia twórców języka HTML.

W celu przywrócenia początkowej struktury języka HTML i zapewnienia autorom stron WWW tak poszukiwanej możliwości kontroli ich układu i wyglądu World Wide Web Consortium wprowadziło *kaskadowe arkusze stylów* (ang: *Cascading Style Sheets,* w skrócie: CSS) oraz XHTML (*Extensible HyperText Markup Language*).

W tym rozdziale poznasz:

- różnice pomiędzy językami HTML i XHTML,
- podstawowe informacje o kaskadowych arkuszach stylów,
- krótką historię kaskadowych arkuszy stylów,
- sposoby tworzenia i implementacji zewnętrznych, osadzonych oraz lokalnych arkuszy stylów,
- najczęściej wykorzystywane właściwości stylów oraz ich wartości,
- sposoby kontroli wyglądu strony, czcionek i kolorów przy wykorzystaniu właściwości kaskadowych arkuszy stylów.

Czym jest XHTML i dlaczego należy go używać?

World Wide Web Consortium (W3C) określa XHTML jako sposób wyrażenia HTML 4.0 w postaci aplikacji XML. Uważam, że to dobra wiadomość dla wszystkich autorów

stron WWW. Wraz z rozwojem Internetu coraz więcej firm będzie wchodziło na rynek, prezentując nowe przeglądarki dokumentów (na przykład, przeglądarki internetowe oraz inne programy). Jest całkiem prawdopodobne, że w niedalekiej przyszłości dzisiejsze przeglądarki zostaną zastąpione jakimiś zupełnie nowymi programami. Niewątpliwie chciałbyś mieć pewność, że Twoje strony WWW będzie można oglądać w tych nowych programach. Właśnie w tym celu został opracowany XHTML. Jest to możliwe dzięki wprowadzaniu wymogu, aby dokumenty spełniały pewne ściśle określone rygory.

Identyfikator <DOCTYPE>

Pamiętasz zapewne, że wszystkie dokumenty HTML muszą zawierać pewne elementy: `<html>`, `<head>`, `<body>` oraz odpowiednie znaczniki zamykające; elementy te przedstawiłam w poniższym przykładzie:

```
<html>
<head>
<title>Prosta strona HTML</title></head>
<body>
Najprostsza strona WWW zawierająca 3 znaczniki
</body>
</html>
```

XHTML dodaje jeszcze jeden wymagany element — identyfikator `<doctype>`. Znacznik ten określa typ tworzonego dokumenty HTML, który może być: `Transitional` (pośredni), ścisły `Strict` (ścisły) lub `Frameset` (układ ramek).

- Użyj znacznika <DOCTYPE> definiującego typ dokumentu jako pośredni (transitional), jeśli dokument jest w większości przypadków formatowany przy użyciu arkuszy stylów, jednak zawiera także formatujące znaczniki HTML (określające, na przykład, kolory i czcionki), dzięki którym dokument może być oglądany w starszych przeglądarkach.

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

- Użyj znacznika <DOCTYPE> definiującego typ dokumentu jako ścisły (strict), jeśli dokument jest formatowany wyłącznie za pomocą arkuszy stylów. Dokumenty tego typu będzie można oglądać tylko w przeglądarkach obsługujących kaskadowe arkusze stylów (CSS), takich jak Internet Explorer 4 lub Netscape Navigator 4.

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

- Użyj znacznika <DOCTYPE> definiującego typ dokumentu jako układ ramek, jeśli dokument zawiera układ ramek. Układy ramek nauczysz się tworzyć w rozdziale 12.

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



Przykłady przedstawione w tej książce są zgodne ze standardem XHTML Transitional.

Składnia XHTML

HTML to język znacznikowy, w którym można sobie pozwolić na popełnianie pewnych błędów bez żadnych konsekwencji, na przykład, HTML wie, że rozpoczynając kolejny znacznik <li> (element listy) wewnątrz znacznika <ol> (listy numerowanej), chcesz jednocześnie zamknąć poprzedni znacznik <li>, choć nie podałeś odpowiedniego znacznika zamykającego. Z punktu widzenia języka HTML poniższy przykład (wykorzystujący poprawną składnię):

```
<ol>
<li START="3">Hej ho, hej ho</li>
<li>do pracy by się...</li>
</ol>
```

nie różni się niczym od poniższego fragmentu kodu:

```
<OL>
<LI start="3">Hej ho, hej ho
<LI>do pracy by się...</LI>
</OL>
```

ani do kodu o następującej postaci:

```
<ol>
<Li start="3">Hej ho, hej ho</Li>
<LI>do pracy by się...</LI>
</OL>
```

Z punktu widzenia języka XHTML wszystkie te przykłady są odmienne. W dalszej części tego podrozdziału dowiesz się dlaczego:

- ***nazwy znaczników i atrybutów zapisuj małymi literami***, z punktu widzenia języka XHTML <LI>, <li> oraz <Li> to różne znaczniki, które należy traktować w całkowicie odmienny sposób, aby ułatwić przeglądarkom analizę i przetwarzanie kodu dokumentów, wszystkie znaczniki i atrybuty HTML będziemy zapisywali małymi literami,
- ***wartości atrybutów zapisuj wewnątrz cudzysłowów***, upewnij się, że wartości wszystkich atrybutów są zapisywane wewnątrz znaków cudzysłowu, w ten sposób ułatwisz przeglądarce określenie, że dany tekst jest wartością, a nie poleceniem,
- ***zakańczaj wszystkie znaczniki posiadające jakąś zawartość***, jak już wspominałam wcześniej, w języku HTML można pominąć znacznik zamykający, a przeglądarka, pomimo to będzie w stanie poprawnie wyświetlić dokument; XHTML wymaga, aby wszystkie znaczniki były poprawnie zamykane, ale jak należy zakańczać znaczniki, takie jak
 — przejście do nowego wiersza lub <hr> — poziomą linię, które nie mają żadnych znaczników zamykających? Wystarczy umieścić na końcu znacznika zmykający znak ukośnika (/), jak pokazałam w poniższym przykładzie:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/strict.dtd">
<html>
<head>
<title>Nowe wiersze i poziome linie</title></head>
<body>
```

```
<hr />                    <!-- otwiera i zamyka znacznik hr -->
Pierwszy wiersz<br />     <!-- otwiera i zamyka znacznik br -->
Drugi wiersz
<hr />                    <!-- otwiera i zamyka znacznik hr -->
</body>
</html>
```

- ***poprawnie zagnieżdżaj znaczniki i nie dopuszczaj, aby „zachodziły" na siebie***, w języku HTML oba przedstawione poniżej fragmenty kodu były identyczne; jednak w XHTML-u fragmenty te różnią się od siebie; pierwszy przedstawiony przykład jest poprawny, a drugi nie.

```
<b>Ten tekst jest pogrubiony.<i>Pogrubiona kursywa.</i></b><i>Tekst wyświetlony
kursywą.</i>
```

```
<b>Ten tekst jest pogrubiony.<i>Pogrubiona kursywa.</b>Tekst wyświetlony
kursywą.</i>
```

Idea arkuszy stylów

Teraz, kiedy już wiesz, jak należy pisać dokumenty HTML, używając poprawnej składni języka XHTML oraz, jak określać typ dokumentów przy użyciu znacznika <DOCTYPE>, zrozumienie idei arkuszy stylów nie powinno przysporzyć Ci żadnych problemów. W pierwszej kolejności autor tworzy standardową stronę WWW, wykorzystując przy tym standardowe znaczniki HTML (te same, które były używane do tej pory). Tak utworzona, standardowa strona WWW jest całkowicie niezależna, to znaczy, została zaprojektowana w taki sposób, aby można ją było poprawnie wyświetlić w przeglądarkach, które nie obsługują arkuszy stylów. Poniżej podałam prosty przykład takiej strony:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/strict.dtd">
<html>
<head>
<title>Zastosowanie arkuszy stylów</title></head>
<body>
<h1>Zastosowanie arkuszy stylów</h1>
<p> W tym prostym przykładzie, nagłówek zostanie wyświetlony
na niebiesko, a tekst akapitu będzie miał inną czcionkę.</p>
</body>
</html>
```

Powyższa strona WWW nie zawiera żadnych atrybutów określających jej wygląd. Zgodnie z jej treścią autor chce, aby nagłówek był niebieski, a do wyświetlenia tekstu akapitu użyje innej czcionki. Aby zrealizować te zamierzenia, tworzy arkusz stylów, który umożliwia sformatowanie zawartości strony w sposób określony przez jej autora.

Reguły stylów mogą formatować wszystkie znaczniki HTML, łącząc wybrane znaczniki (na przykład, h1) z właściwościami (takimi jak: color: blue). W *arkuszach stylów* znaczniki HTML używane są jako *selektory*. Reguły stylów mogą określać układ znacznika oraz wszelkie inne właściwości związane z typografią oraz jego wyglądem. Poniżej podałam kilka przykładów reguł stylów.

```
h1 { color: blue }
p { font-family: Arial, Helvetica, sans-serif; color: black }
```

Nowe pojęcie

Reguły stylów, używane w kaskadowych arkuszach stylów, składają się z dwóch części: *selektora*, którym może być znacznik HTML, na przykład, `p` lub `h1` oraz z *deklaracji* definiującej właściwości i wartości selektora, na przykład, `color: blue` (gdzie `color` jest właściwością, `blue`, wartością).

Reguła zapisana w pierwszym wierszu powyższego przykładu określa, że nagłówek `h1` na stronie będzie wyświetlany w kolorze niebieskim. Natomiast druga reguła określa, że tekst wszystkich akapitów (znaczników <p>) na stronie zostanie wyświetlony na czarno, czcionkami Arial, Helvetica lub dowolną inną czcionką bezszeryfową.

Teraz, gdy autor stworzył już zawartość strony (standardową stronę WWW) oraz określił reguły rządzące jej wyglądem, czas na kolejny etap, którym będzie dołączenie reguł stylów do standardowego dokumentu HTML. Można to zrobić na trzy sposoby: wykorzystując zewnętrzny arkusz stylów, osadzając arkusz stylów (jak pokazałam w kolejnym przykładzie) lub wpisując styl wewnątrz znacznika. W dalszej części rozdziału pt.: „Sposoby tworzenia arkuszy stylów", dowiesz się więcej na temat sposobów tworzenia arkuszy stylów.

Poniższy przykład pokazuje, jak nasza prosta strona WWW może zostać sformatowana za pomocą osadzonego arkusza stylów:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/strict.dtd">
<html>
<head>
<title>Zastosowanie arkuszy stylów</title>
<style type="text/css">
<!--
h1 { color: blue }
p { font-family: Arial, Helvetica, sans-serif; color: black }
-->
</style>
</head>
<body>
<h1>Zastosowanie arkuszy stylów</h1>
<p> W tym prostym przykładzie, nagłówek zostanie wyświetlony
na niebiesko, a tekst akapitu będzie miał inną czcionkę.</p>
</body>
</html>
```

W powyższym przykładzie, demonstrującym wykorzystanie osadzonego arkusza stylów, autor strony oddzielił style od standardowego dokumentu HTML. Kod określający wygląd strony został umieszczony pomiędzy otwierającym znacznikiem <style> oraz odpowiednim znacznikiem zamykającym </style>. Przeglądarki, które nie są w stanie obsługiwać arkuszy stylów, wciąż będą w stanie wyświetlić dokument, tak jakby był on standardowym dokumentem HTML, natomiast nowsze przeglądarki, obsługujące arkusze stylów, wyświetlą go zgodnie z podanymi regułami.

Dodatkowe informacje o arkuszach stylów

Pierwsza implementacja kaskadowych arkuszy stylów, znana jako CSS1, umożliwia określanie wielu aspektów wyglądu strony i znaczników, takich jak czcionki używane do prezentacji różnych elementów HTML, kolory czcionek, kolory oraz obrazy tła, marginesy, odstępy pomiędzy literami i wyrazami i wiele, wiele innych. Przeglądarki obsługujące arkusze stylów tego typu — Netscape Navigator 4 oraz Internet Explorer 4 oraz wersje późniejsze obu tych przeglądarek — określają ostateczną postać dokumentu na podstawie definicji stylów.

Kolejna generacja arkuszy stylów CSS2 stała się oficjalnym zaleceniem w maju 1998 roku. Wiele znaczników CSS2 jest obsługiwanych przez najnowsze wersje przeglądarek Internet Explorer oraz Netscape Navigator. W bardzo niedalekiej przyszłości arkusze stylów umożliwią wykonywanie jeszcze bardziej niesamowitych zadań. Dzięki wykorzystaniu CSS2, XHTML oraz odpowiednich przeglądarek będzie można tworzyć strony WWW wykorzystujące różne typy mediów. Na przykład, będzie można zaprojektować głosowy arkusz stylów, dźwiękowo odtwarzający elementy strony WWW, wykorzystując przy tym właściwości przestrzenne oraz stereo. Będzie także można podzielić stronę WWW na wiele stron, podobnie jak to się robi w edytorach tekstów oraz programach poligraficznych, kontrolować miejsca podziału stron, okna, itp. Możliwości są naprawdę fascynujące!

To zapowiada jeszcze jedną ogromną zaletę technologii arkuszy stylów. Otóż będą one umożliwiały przeglądanie dokumentów HTML w niestandardowy sposób, na przykład, przy wykorzystaniu odtwarzaczy audio, z czego będą mogły korzystać osoby mające problemy ze wzrokiem lub na jeszcze inne sposoby, gdzie wykorzystanie standardowej technologii przeglądarek nie będzie możliwe. Standardowy dokument HTML wciąż może być wyświetlany w normalny sposób, przy czym nie trzeba w nim stosować żadnych dodatkowych, rozszerzających znaczników HTML, przeznaczonych do udostępnienia niestandardowych możliwości i bardzo często wprowadzających niepotrzebne zamieszanie w kodzie strony.

Różne przeglądarki odmiennie obsługują język HTML, dotyczy to także kaskadowych arkuszy stylów. Firma Netscape stworzyła własną, alternatywną wersję arkuszy stylów. Netscape Navigator dysponuje także ramkami, alternatywnym rozwiązaniem służącym do umiejscawiania elementów stron WWW, przy czym obsługuje także pojawiające się standardowe rozwiązania. Przeglądarka ta obsługuje także arkusze stylów JavaScript, a jednocześnie można w niej stosować także kaskadowe arkusze stylów.

Ta sytuacja może się wydawać (i jest) nieco trudna, zwłaszcza wziąwszy pod uwagę, iż jest to nowa technologia. Jak to jednak bywa ze wszystkimi nowymi technologiami, sytuacja się w końcu wyjaśni i pojawi „standard VHS" dla arkuszy stylów, natomiast „standard Betamax" odejdzie w zapomnienie. Jak na razie należy jednak pamiętać, że przeglądarki obsługujące technologię CSS robią to w odmienny sposób.

W pozostałej części tego rozdziału poznasz podstawowe możliwości arkuszy stylów. Pamiętaj, że właściwości przedstawione w tym rozdziale to tylko niewielka, wybrana

cząstka wszystkich możliwości dostępnych w CSS1 oraz CSS2. Szczegółowe i precyzyjne omówienie tych zagadnień znacznie wykracza poza ramy niniejszej książki i bez wątpienia mogłoby podwoić jej objętość. Niemniej jednak, przedstawione pojęcia i informacje pomogą Ci zrozumieć i ocenić ogromne możliwości arkuszy stylów, ich właściwości i wartości. Dodatkowe źródła informacji dotyczące omawianych tu zagadnień, można znaleźć na końcu tego rozdziału.

Księgarnia Bookworm

Zamiast przytłaczać Cię samą teorią, przedstawię teraz kilka właściwości CSS i pokażę Ci, jak je wykorzystać na stronach WWW. W rozdziale 6. poznałeś różne polecenia służące do formatowania tekstu i określania czcionek, które przedstawiłam w opisie strony Księgarni Bookworm. Teraz, na przykładzie nowej wersji tej strony, dowiesz się, w jaki sposób można zastąpić te polecenia kaskadowymi arkuszami stylów.

Poniżej przedstawiłam nową wersję tej strony (bez żadnych wymyślnych dodatków umieszczonych na niej pod koniec rozdziału 6). Otwórz tę stronę lub stwórz ją od początku, gdyż w ćwiczeniach przedstawionych w dalszej części rozdziału będziemy konwertowali ją w taki sposób, aby korzystała z kaskadowych arkuszy stylów. Stronę zapisz w pliku o nazwie *bookwrm.html*. Początkowa jej część została przedstawiona na rysunku 10.1, można na nim zobaczyć wszystkie typy elementów występujących na całości strony.

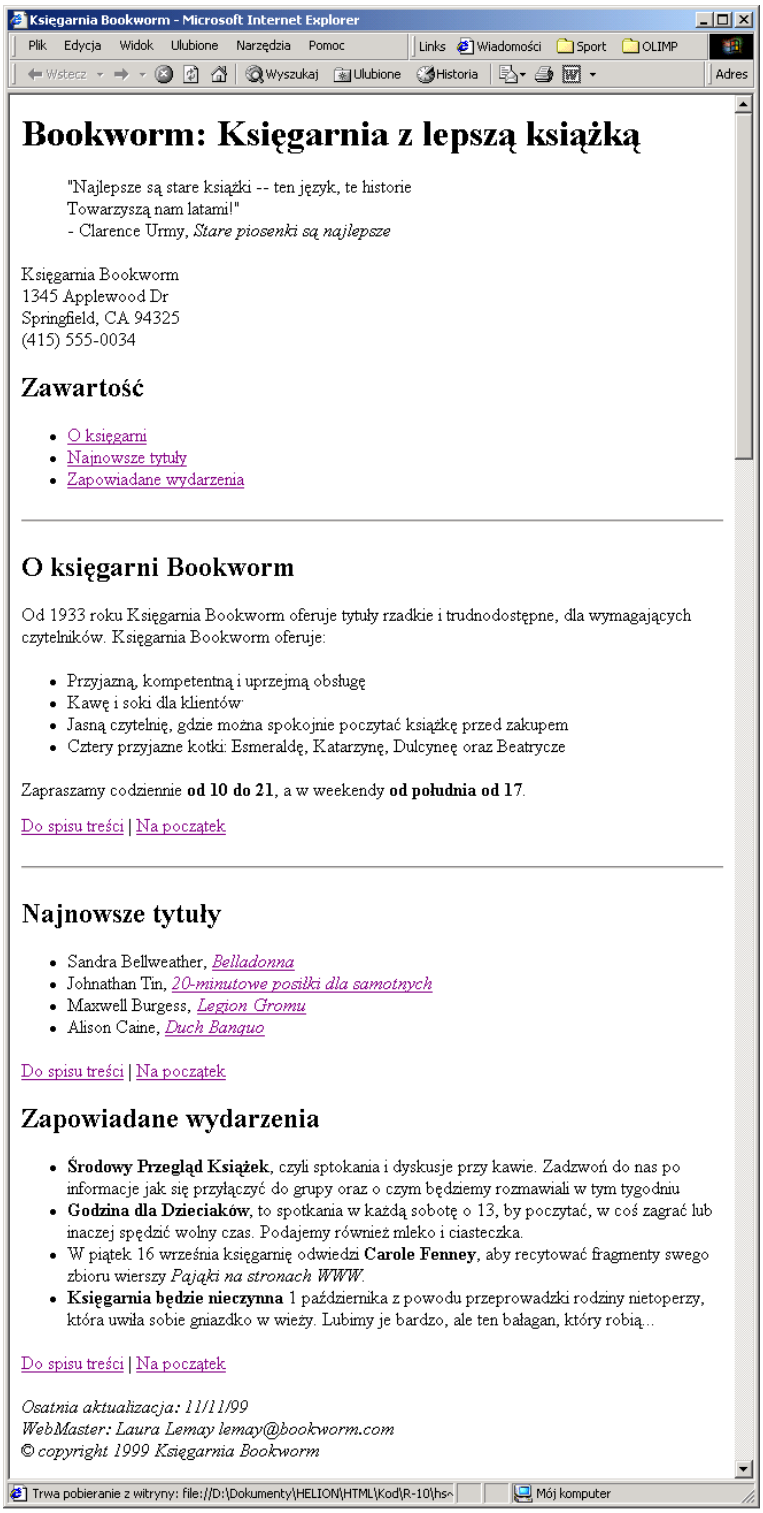
Źródło

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/strict.dtd">
<html>
<head>
<title>Księgarnia Bookworm</title>
</head>
<body>
<a name="top"><h1>Bookworm: Księgarnia z lepszą książką</h1></a>
<blockquote>
"Najlepsze są stare książki - ten język, te historie<br />
Towarzyszą nam latami!"<br />
- Clarence Urmy, <cite>Stare piosenki są najlepsze</cite>
</blockquote>
<p>Księgarnia Bookworm<br />
1345 Applewood Dr<br />
Springfield, CA 94325<br />
(415) 555-0034
</p>
<a name="contents"><h2>Zawartość</h2></a>
<ul>
  <li><a href="#about">O księgarni</a></li>
  <li><a href="#recent">Najnowsze tytuły</a></li>
  <li><a href="#upcoming">Zapowiadane wydarzenia</a></li>
</ul>
<hr />
<a name="about"><h2>O księgarni Bookworm</h2></a>
<p>Od 1933 roku księgarnia Bookworm oferuje tytuły rzadkie i trudnodostępne,
dla wymagających czytelników. Księgarnia Bookworm oferuje:</p>
```

```
<ul>
<li>Przyjazną, kompetentną i uprzejmą obsługę</li>
<li>Kawę i soki dla klientów</li>
<li>Jasną czytelnię, gdzie można spokojnie poczytać książkę przed zakupem</li>
<li>Cztery przyjazne kotki: Esmeraldę, Katarzynę, Dulcyneę oraz Beatrycze</li>
</ul>
<p>Zapraszamy codziennie <strong>od 10 do 21</strong>, a w weekendy
<strong>od południa od 17</strong>.</p>
<p><a href="#contents">Do spisu treści</a> | <a href="#top">
Na początek</a></p>
<hr />
<a name="#recent"><h2>Najnowsze tytuły</h2></a>
<ul>
<li>Sandra Bellweather, <a href="belladonna.html">
<cite>Belladonna</cite></a></li>
<li>Johnathan Tin, <a href="20minmeals.html">
<cite>20-minutowe posiłki dla samotnych</cite></a></li>
<li>Maxwell Burgess, <a href="legion.html">
<cite>Legion Gromu</cite></a></li>
<li>Alison Caine, <a href="banquo.html">
<cite>Duch Banquo</cite></a></li>
</ul>
<p><a href="#contents">Do spisu treści</a> | <a href="#top">
Na początek</a></p>
<a name="upcoming"><h2>Zapowiadane wydarzenia</h2></a>
<ul>
<li><b>Środowy Przegląd Książek</b>, czyli spotkania i dyskusje przy kawie. Zadzwoń
do nas po informacje, jak się przyłączyć do grupy oraz o czym będziemy rozmawiali w
tym tygodniu.</li>
<li><b>Godzina dla Dzieciaków</b>, to spotkania w każdą sobotę o 13, by poczytać,
w coś zagrać lub inaczej spędzić wolny czas. Podajemy również mleko i
ciasteczka.</li>
<li>W piątek 16 września księgarnię odwiedzi <b>Carole Fenney</b>, aby recytować
fragmenty swego zbioru wierszy <cite>Pająki na stronach WWW</cite>.</li>
<li><b>Księgarnia będzie nieczynna</b> 1 października z powodu przeprowadzki
rodziny nietoperzy, która uwiła sobie gniazdko w wieży. Lubimy je bardzo, ale ten
bałagan, który robią...</li>
</ul>
<p><a href="#contents">Do spisu treści</a> | <a href="#top">
Na początek</a></p>
<address>
Osatnia aktualizacja: 11/11/99<br />
WebMaster: Laura Lemay lemay@bookworm.com<br />
&#169; copyright 1999 Księgarnia Bookworm<br />
</address>
</body>
</html>
```

Na rysunku 10.1 przedstawiona została strona WWW, jaką, bez żadnych problemów, można wyświetlić w przeglądarce, która nie obsługuje kaskadowych arkuszy stylów. Widzimy, że strona nie jest szczególnie wymyślna. Zapewne możemy uatrakcyjnić ją nieco przy użyciu kolorów, różnych czcionek, marginesów, itp. Do tych celów, doskonale nadają się arkusze stylów.

Wynik



Rysunek 10.1. *Księgarnia Bookworm jako niezależna strona WWW*

Sposoby tworzenia arkuszy stylów

Zatem, w jaki sposób można wykorzystać technologię CSS na takiej standardowej stronie WWW? Istnieją trzy podstawowe sposoby zastosowania CSS do formatowania znaczników HTML. Najwspanialszą cechą arkuszy stylów jest ich ogromna elastyczność. Wykorzystanie znaczników HTML oraz atrybutów w celu określenia postaci dokumentów HTML sprawia, że autorzy stron oraz twórcy przeglądarek nie muszą stosować arkuszy stylów jednego typu.

World Wide Web Consortium zdefiniowało zbiór znaczników oraz atrybutów, które można wykorzystać do stworzenia definicji stylów (powiem o nich w dalszej części rozdziału). Z kolei definicji tych można użyć do określenia postaci dowolnego dokumentu lub znacznika HTML. Istniejące znaczniki dają możliwość tworzenia trzech rodzajów arkuszy stylów: zewnętrznych arkuszy stylów, osadzonych arkuszy stylów oraz stylów lokalnych. Pierwsze dwa typy zostaną przedstawione w dwóch kolejnych podrozdziałach. W dalszej części rozdziału przyjrzymy się bliżej stylom lokalnym.

Zewnętrzne arkusze stylów

Stosując zewnętrzne arkusze stylów, reguły stylów przechowane są w osobnych plikach, poza dokumentami HTML. Zaleta arkuszy stylów tego typu polega na tym, iż reguły stylów można wykorzystywać w wielu dokumentach HTML wchodzących w skład witryny. Dzięki temu można tworzyć strony WWW o spójnym, jednolitym wyglądzie. Dodatkową zaletą zewnętrznych arkuszy stylów jest umożliwienie szybkiej modyfikacji wyglądu witryny. Dzięki umieszczeniu reguł stylów w jednym dokumencie i połączeniu go z wieloma stronami WWW, zmiana sposobu ich prezentacji polega na zmianie tylko jednego pliku.

Podobnie jak dokument HTML, także i dokument zawierający arkusz stylów, to zwyczajne pliki tekstowe zapisane w kodzie ASCII i posiadające specjalne rozszerzenia. Strony WWW są zapisywane z rozszerzeniami *.html* lub *.htm*, natomiast arkusze stylów, z rozszerzeniem *.css*. Dokument ten definiuje wspólne reguły stylów stosowane we wszystkich stronach WWW. Zewnętrzny arkusz stylów dołącza się do dokumentów HTML za pomocą znacznika `<link>`.

Wszelkie szczegóły związane z tworzeniem dokumentów zawierających arkusze stylów przedstawię w kolejnym ćwiczeniu. Jak na razie załóżmy, że stworzyłeś arkusz stylów i zapisałeś go w pliku o nazwie *mystyle.css*, w tym samym folderze, w którym znajduje się strona Księgarni Bookworm. Przedstawiony poniżej fragment kodu pokazuje, w jaki sposób należy dołączyć ten arkusz stylów do nagłówka strony Księgarni Bookworm:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/strict.dtd">
<html>
<head>
<title>Księgarnia Bookworm</title>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>
```

Znacznik <link> kojarzy zewnętrzny arkusz stylów (*mystyle.css*) z aktualnym dokumentem HTML. Dzięki wstawieniu tego samego fragmentu kodu na każdej stronie witryny, jej administrator może nadać jednolity styl dokumentom tworzonym przez wszystkich pracowników firmy.

Istotną rolę pełni atrybut rel znacznika <link> (omawiany w rozdziale 5. — „Wszystko o połączeniach"). Aby efektywnie używać znacznika <link>, będziesz musiał poznać i zrozumieć różnice pomiędzy stylami trwałymi, domyślnymi oraz zamiennymi. Oto podstawowe informacje na ich temat:

- ***style trwałe*** są stosowane zawsze, niezależnie od lokalnych ustawień użytkownika,
- ***style domyślne*** są stosowane po pobraniu strony, jednak można je zastąpić stylami zamiennymi, wybieranymi przez użytkownika,
- ***style zamienne*** są tworzone, aby dać użytkownikom możliwość wyboru (w odróżnieniu od stylów domyślnych).

Atrybut rel kontroluje fragment tego procesu. Podanie atrybutu rel="stylesheet", jak w podanym przykładzie, wymusza zastosowanie stylów trwałych i wykorzystuje style podane w arkuszu, niezależnie od lokalnych ustawień użytkownika.

Dodając atrybut title do znacznika <link>, można określić styl domyślny. Oto przykład:

```
<link rel="stylesheet" title="mainstyle" href="mystyle.css">
```

Zmieniając atrybut rel="stylesheet" na rel="alternate stylesheet", definiujemy styl alternatywny o innym tytule.

W taki sposób można stworzyć stały styl zawierający definicje, które muszą być użyte niezależnie od ustawień użytkownika, jak również wybrać styl domyślny oraz jeden lub kilka stylów zamiennych, stanowiących alternatywę dla stylu trwałego.

Ćwiczenie 10.1: Tworzenie i dołączanie zewnętrznych arkuszy stylów

Jeśli jeszcze tego nie zrobiłeś, to otwórz stronę Księgarni Bookworm i zapisz ją w pliku o nazwie *bookwrm.html*. Oryginalny kod tej strony został przedstawiony w poprzednim podrozdziale, pt.: „Księgarnia Bookworm". Aby dołączyć do tej strony zewnętrzny arkusz stylów (który za chwilę utworzymy), umieść w nagłówku dokumentu następujący fragment kodu HTML, bezpośrednio po tytule strony:

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

Cały nagłówek strony ma teraz następującą postać:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/strict.dtd">
<html>
<head>
<title>Księgarnia Bookworm</title>
<link rel="stylesheet" href="mystyle.css">
</head>
```

Zapisz stronę zawierającą zmieniony nagłówek. A zatem połączyłeś już arkusz stylów ze stroną WWW, teraz należy stworzyć ten arkusz. Jeśli nie umieścisz w arkuszu stylów kilku stylów, nie będziesz w stanie określić, czy po podłączeniu arkusz stylów działa poprawnie. Dlatego, w dalszej części ćwiczenia, stworzysz nowy dokument zawierający kilka prostych stylów.

Jak na razie może Ci się wydawać, że przedstawiony poniżej przykład nie ma większego sensu; jednak w dalsze części rozdziału zrozumiesz jego przeznaczenie. Poniżej pokrótce opisałam działanie poniższego fragmentu kodu:

- tworzy arkusz stylów nadający tłu strony kolor jasnoniebieski,
- zapewnia, że tekst umieszczony wewnątrz strony zostanie wyświetlony czcionką Arial, Helvetica lub inną czcionką bezszeryfową, zależnie od czcionek dostępnych w systemie użytkownika oglądającego stronę,
- zapewnia, że tekst strony zostanie wyświetlony kolorem ciemnoniebieskim.

Dodatkowo arkusz stylów zmienia kolory połączeń w następujący sposób:

- nieodwiedzone połączenia będą fioletowe,
- odwiedzone połączenia będą ciemnobrązowe,
- aktywne połączenie będzie czerwone,
- gdy umieścisz wskaźnik myszy na połączeniu, zostanie ono wyświetlone w kolorze złotym.

Utwórz nowy dokument tekstowy i wpisz w nim fragment kodu podany poniżej. Wpisując poniższy kod, powinieneś zwrócić uwagę na jedną rzecz, otóż arkusze stylów są zapisywane wewnątrz nawiasów klamrowych ({}), a nie zwyczajnych nawiasów półokrągłych. Poza tym, pomiędzy regułami stylów oraz otwierającym i zamykającym nawiasem klamrowych, są zazwyczaj umieszczane pojedyncze znaki odstępu, na przykład: `H1 { color: blue }`. Po wpisaniu podanego poniżej kodu, zapisz plik w tym samym folderze, w którym umieściłeś stronę Księgarni Bookworm i nadaj mu nazwę *mystyle.css*. Zwróć uwagę, iż jest to ta sama nazwa, którą podałeś w nagłówku strony Księgarni Bookworm. Oto kod, który powinieneś wpisać w pliku *mystyle.css*:

```
body { background-color: #ccffff; font-family: Arial, Helvetica, sans-serif;
color: 330066 }
a:link { color: #ff00ff }
a:visited { color: #660000 }
a:hover { color: #ffcc00 }
a:active { color: #ff0000 }
```

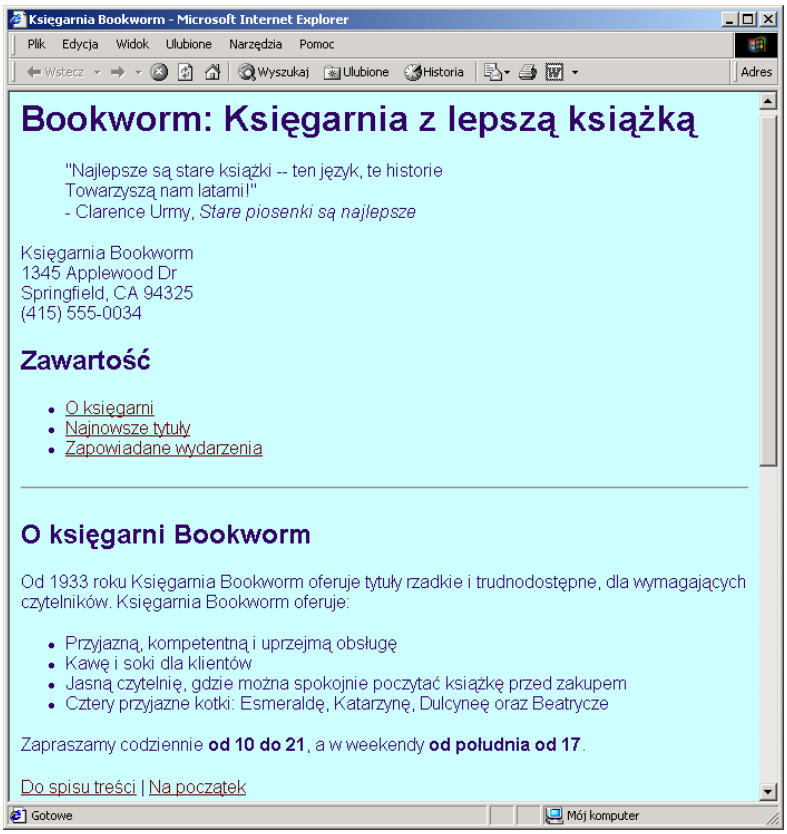
Pierwsza reguła stylu podana w powyższym przykładzie definiuje kolor tła strony (w naszym przypadku jest on jasnoniebieski — #ccffff). Tekst umieszczony na stronie zostanie wyświetlony czcionką Arial, Helvetica lub inną czcionką bezszeryfową, jaką będzie można znaleźć na dysku osoby oglądającej stronę. Tekst strony będzie ciemnoniebieski (#330066).

```
body { background-color: #ccffff; font-family: Arial, Helvetica, sans-serif;
color: 330066 }
```

Kolejne cztery linie kodu definiują cztery „pseudoklasy" określające kolor połączeń. Połączenia, które nie zostały jeszcze odwiedzone są formatowane przy użyciu pseudoklasy `a:link` (odpowiada jej atrybut HTML `link`). Postać połączeń odwiedzonych jest określana przy użyciu pseudoklasy `a:visited` (odpowiada jej atrybut `vlink`). Połączenie aktywne jest definiowane za pomocą pseudoklasy `a:active` (odpowiada jej atrybut HTML `alink`). Ostatnia pseudoklasa `a:hover` definiuje postać połączenia, na którym został umieszczony wskaźnik myszy (w HTML-u nie istnieje atrybut odpowiadający tej pseudoklasie):

```
a:link { color: #ff00ff }
a:visited { color: #660000 }
a:hover { color: #ffcc00 }
a:active { color: #ff0000 }
```

Już za chwilę zobaczysz, że przy użyciu zewnętrznego arkusza stylów zawierającego pięć linijek kodu, można stworzyć stronę WWW o całkowicie odmiennym wyglądzie. Otwórz stronę *bookwrm.html* w przeglądarce WWW obsługującej kaskadowe arkusze stylów, na przykład, w programie Internet Explorer 4 lub w Netscape Navigator 4. Strona, którą zobaczysz powinna przypominać tę, przedstawioną (częściowo) na rysunku 10.2.



Rysunek 10.2. *Strona Księgarni Bookworm po dołączeniu zewnętrznego arkusza stylów*

Osadzone arkusze stylów

Osadzone arkusze stylów to standardowe dokumenty HTML, wewnątrz których zostały umieszczone reguły stylów. Załóżmy, że chcesz, aby Twoja strona domowa miała czarne tło, umieszczony na niej tekst był wyświetlony czcionką o dużej wielkości w jasnym kolorze, a połączenia na stronie miały żywe kolory. Pozostałe strony witryny są bardziej stonowane (aby nie męczyły tak bardzo oczu oglądających je osób) i mają jaśniejszy kolor tła. Tworzenie nowego zewnętrznego arkusza stylów tylko na potrzeby strony domowej nie ma większego sensu. (No bo po co tworzyć dwie strony, kiedy można stworzyć tylko jedną?) A zatem, w takim przypadku możesz użyć osadzonego arkusza stylów, aby określić postać strony domowej oraz zewnętrznego arkusza stylów definiującego wygląd pozostałych stron witryny.

Być może pomyślisz sobie: „Zaraz, zaraz. Ale czy idea arkuszy stylów nie polega właśnie na tym, aby oddzielić style od zawartości strony?". W rzeczywistości style będą oddzielone od zawartości strony. Definiuje się je bowiem w nagłówku, umieszczanym przed początkiem treści dokumentu HTML. A zatem style pozostają odseparowane od zawartości strony. W poniższym przykładzie pokazałam, jak mógłby wyglądać nagłówek strony Księgarni Bookworm zawierający osadzony arkusz stylów. W tym przypadku zdefiniowałam w nim kolor tła strony oraz kolory tekstu i połączeń.

Aby umieścić na stronie WWW osadzony arkusz stylów, należy rozpocząć ją od standardowego nagłówka:

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

Teraz należy otworzyć obszar, w którym zostaną umieszczone reguły stylów. Są one zapisywane wewnątrz znaczników `<style>`. Atrybut `type` znacznika `<style>` określa, że strona wykorzystuje osadzone arkusze stylów i ma następującą postać:

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

Po otwierającym znaczniku `<style>` można podać reguły stylów. Przykład podany poniżej jest bardzo podobny do przykładu zewnętrznego arkusza stylów, przedstawionego we wcześniejszej części rozdziału.

Zwróć uwagę, iż w przypadku osadzonych arkuszy stylów reguły stylów są zapisywane pomiędzy znacznikami komentarzy HTML (`<!--` oraz `-->`). Jeśli nie zapiszesz reguł stylów wewnątrz komentarzy, to starsze typy przeglądarek, które nie obsługują kaskadowych arkuszy stylów, mogą je wyświetlić jako część strony WWW.

```
<!--
  body { color: #000000; background-color: #ffffff }
  a:active { color: #666699 }
  a:hover { color: #3366ff }
  a:link { color: #0066ff }
  a:visited { color: #9966cc }
-->
```

Teraz można zakończyć nagłówek strony zamykającymi znacznikami <style> oraz <head>:

```
</style>
</head>
```

Ćwiczenie 10.2: Tworzenie osadzonego arkusza stylów

Ponownie otwórz stronę Księgarni Bookworm, zapisaną w pliku *bookwrm.html*. Zanim zaczniesz modyfikację, stwórz jej kopię, zapisując stronę pod inną nazwą, na przykład bookwrm-kopia.html. *W tym ćwiczeniu zamienisz dokument* bookwrm.html *w stronę WWW wykorzystującą osadzony arkusz stylów.*

Chcesz oznaczyć dokument jako stronę używającą osadzonych arkuszy stylów. W tym celu konieczne będzie wprowadzenie niewielkich modyfikacji w nagłówku strony. Pamiętasz zapewne, że kod osadzonych arkuszy stylów jest umieszczany wewnątrz znaczników <style>, a same definicje stylów zapisywane w nawiasach klamrowych. W pierwszej kolejności usuń ze strony znacznik <link> odwołujący się do zewnętrznego arkusza stylów.

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

Następnie, w nagłówku strony umieść kod, który określi ją, jako stronę używającą osadzonego arkusza stylów. Nowy nagłówek strony powinien mieć następującą postać:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/strict.dtd">
<html>
<head>
<title>Księgarnia Bookworm</title>
<style type="text/css">
<!--
-->
</style>
</head>
```

Aby zaprezentować style, które zdefiniujesz w osadzonym arkuszu stylów, w niewielkim stopniu zmodyfikujemy style przedstawione w poprzednim ćwiczeniu. Zmienimy kolor tła strony z jasnoniebieskiego na jasnozielone (o wartości #ccffcc). Właściwości tekstu oraz połączeń pozostaną takie same. Tym razem jednak, reguły stylów zostaną zapisane *wewnątrz* strony Księgarni Bookworm, pomiędzy znacznikami komentarza: <!-- oraz -->.

Poniższy fragment kodu wpisz pomiędzy znacznikami komentarza HTML, umieszczonymi w nagłówku strony:

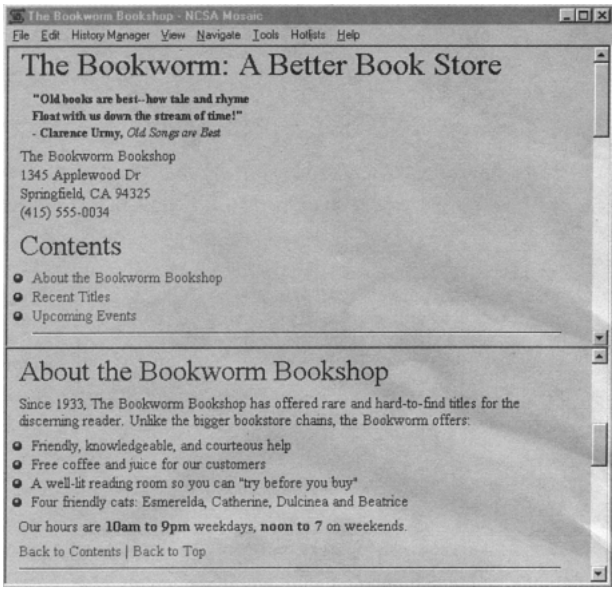
```
body { background-color: #ccffcc; font-family: Arial, Helvetica, sans-serif;
color: 330066 }
a:link { color: #ff00ff }
a:visited { color: #660000 }
a:hover { color: #ffcc00 }
a:active { color: #ff0000 }
```

A zatem, cały nagłówek strony powinien mieć teraz następującą postać:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/strict.dtd">
<html>
<head>
<title>Księgarnia Bookworm</title>
<style type="text/css">
<!--
body { background-color: #ccffcc; font-family: Arial, Helvetica, sans-serif;
color: 330066 }
a:link { color: #ff00ff }
a:visited { color: #660000 }
a:hover { color: #ffcc00 }
a:active { color: #ff0000 }
-->
</style>
</head>
```

Ponownie zapisz stronę *bookwrm.html*, a następnie otwórz ją w przeglądarce obsługującej kaskadowe arkusze stylów. Teraz strona ma jasnozielone tło. Poza tym, nowa wersja strony wygląda bardzo podobnie do strony wykorzystującej zewnętrzny arkusz stylów, przedstawionej na rysunku 10.2.

Dla celów porównawczych można otworzyć tę samą stronę w przeglądarce, która nie jest w stanie obsługiwać kaskadowych arkuszy stylów, na przykład, w programie NCSA Mosaic 3.0. Na rysunku 10.3 widać początkowy fragment strony z oryginalnego wydania niniejszej książki, wyświetlonej w przeglądarce NSCA Mosaic 3.0. Jak widać, wszystkie informacje umieszczone na stronie wciąż są na niej widoczne; wynika to z faktu, iż kod HTML definiujący zawartość strony nie jest zależny od kodu formatującego stronę. Niemniej jednak, czcionki oraz kolory zdefiniowane w osadzonym arkuszu stylów nie zostały w tej przeglądarce wyświetlone.



Rysunek 10.3. *Oryginalna strona Księgarni Bookworm wyświetlona w przeglądarce, która nie obsługuje kaskadowych arkuszy stylów*

Kilka słów o kaskadzie stylów

Ale dlaczego ten rodzaj arkuszy stylów został określony jako „kaskadowe"? Bez wątpienia nie chodzi tu o żadne wodospady.

Kaskadą stylów określa się możliwość łączenia informacji o stylach pochodzących z kilku źródeł. Już wiesz, że arkusze stylów można określać w różny sposób. Zewnętrzne arkusze stylów można dołączać do wielu stron, a umieszczone w nich reguły stylów posłużą do określenia postaci każdej z nich. Osadzone arkusze stylów są używane do określenia postaci tylko jednej strony. *Style lokalne* pozwalają na zdefiniowanie reguł stylów, odnoszących się do elementów strony. Jeśli chcesz, to na jednej stronie można wykorzystać wszystkie trzy typy arkuszy stylów.

O *kaskadzie stylów* mówimy dlatego, że istnieje ściśle określony porządek, w jakim są wykorzystywane reguły zdefiniowane w poszczególnych arkuszach stylów; otóż reguły zdefiniowane w arkuszu późniejszym mają wyższy priorytet, niż reguły zdefiniowane w arkuszach wcześniejszych. Poniżej, w najprostszy możliwy sposób, opisałam, co się dzieje, gdy na jednej stronie WWW zostaną zastosowane wszystkie trzy typy arkuszy stylów: zewnętrzne, osadzone oraz lokalne.

- W pierwszej kolejności wygląd strony jest modyfikowany zgodnie z regułami podanymi w zewnętrznych arkuszach stylów.
- W drugiej kolejności zostają zastosowane definicje stylów zapisane w osadzonych arkuszach stylów, które, w razie konieczności, zastępują definicje stylów opisane w zewnętrznych arkuszach stylów.
- Na końcu są stosowane style lokalne, które w razie konieczności zastępują definicje stylów podane w dwóch poprzednich rodzajach arkuszy stylów.

Najczęściej stosowane właściwości stylów oraz ich wartości

W tej części rozdziału poznasz najważniejsze właściwości stylów i dowiesz się, jak można ich używać. Jest ich jednak zbyt dużo, jak na jeden rozdział. Jeśli będziesz potrzebował dodatkowych informacji, to pełną specyfikację kaskadowych arkuszy stylów poziomu 1 (CSS1) można znaleźć na WWW, pod adresem *http://www.w3.org/ TR/REC-CSS1*.

Na WWW, pod adresem *http://www.w3.org/TR/REC-CSS2/*, można także znaleźć zalecenie CSS2. Najnowsza wersja tego dokumentu pochodzi z 12 maja 1998 roku. Oba te dokumenty, choć mają raczej techniczny charakter, dokładnie opisują znaczenie każdej z właściwości, zawierają informacje o tym, jakie zastosowania właściwości są poprawne, a które nie, jak również podają kilka przykładów, które ułatwią Ci tworzenie stron zgodnych ze starszymi oraz nowszymi przeglądarkami.

W kilku przedstawionych do tej pory przykładach zobaczyłeś jedynie niewielką próbkę możliwości, jakie dają style. CSS1 oraz CSS2 udostępniają bardzo wiele znaczników,

które umożliwiają kontrolę niemal wszystkich aspektów wyglądu strony WWW. CSS jest jednak technologią, która wciąż się rozwija. Z tego względu niektóre właściwości CSS są obsługiwane w przeglądarkach lepiej i bardziej niezawodnie od innych. Tak to już jest z WWW.

Choć przedstawiam tu niektóre właściwości CSS, musisz mieć świadomość, iż przeglądarki, obsługujące kaskadowe arkusze stylów, mogą wyświetlać te same przykłady w niespodziewany i zaskakujący sposób. Nie pozostaje Ci zatem nic innego, jak tylko próbować, próbować i jeszcze raz próbować, aż w końcu otrzymasz zadowalające rezultaty.

Kontrola układu strony przy użyciu właściwości CSS

W ćwiczeniach 10.1 oraz 10.2 dodałeś do strony kilka prostych reguł określających czcionki i kolory jej wybranych elementów. W tym celu wykorzystałeś właściwości `background-color`, `font-family` oraz `color`, których użyłeś do sformatowania zawartości strony (znacznika `<body>`). Kolory połączeń zostały określone dzięki wykorzystaniu czterech pseudoklas znacznika `<a>`: `a:link` (użytej do określenia koloru połączeń), `a:visited` (użytej do określenia koloru odwiedzonych połączeń), `a:hover` (użytej do określenia koloru połączenia, na którym został umieszczony wskaźnik myszy) oraz a:active (użytej do określenia koloru aktywnego połączenia).

Jednak istniej znacznie więcej możliwości modyfikowania wyglądu tekstu wyświetlanego na stronie. Arkusze stylów, podobnie jak edytory tekstów oraz programy poligraficzne, pozwalają na określanie wielkości marginesów oraz wypełnienia. Kilka najczęściej używanych właściwości zebrałam w tabeli 10.1.

Marginesy oraz wypełnienia są definiowane jako długość wyrażona wartością liczbową wraz z towarzyszącą jej *jednostką długości*, jako wartość procentowa lub jako wartość `auto`.

Nowe pojęcie

Jednostki długości są wyrażane w wartościach względnych lub bezwzględnych. Do względnych jednostek długości zaliczane są `em` (wielkość określonej czcionki), `ex` (wysokość określonej czcionki) oraz `px` (piksele, zależne od urządzenia, na którym strona będzie wyświetlana). Jednostki bezwzględne to: `pt` (punkty), `cm` (centymetry), `mm` (milimetry) oraz `pc` (pica).

Wartości procentowe są zawsze określane względem innej, wybranej wartości, takiej jak długość. Wartości procentowe mogą się rozpoczynać od opcjonalnego znaku + lub – bezpośrednio po nim należy podać liczbę i znak procentu (`%`).

Być może zwróciłeś uwagę na określenie *właściwość skrótowa* użyte w tabeli 10.1. Otóż wygląd niektórych właściwości opisywany jest przy użyciu kilku atrybutów. Właściwość skrótowa umożliwia połączenie i jednoczesne podanie kilku wartości, na przykład, używając jednej właściwości skrótowej *margin*, można określić postać wszystkich czterech marginesów: górnego, prawego, dolnego oraz lewego. To samo dotyczy właściwości *padding*, przy użyciu której można określać wszystkie cztery wypełniania (górne, prawe, itd.).

Tabela 10.1.
Przydatne właściwości określające postać strony

Właściwość	Opis
`margin-top`	Konfiguruje obszar pomiędzy elementem a jego sąsiadami (górny margines elementu). Można go definiować w jednostkach absolutnych, w procentach względem szerokości tekstu lub automatycznie (wartość `auto`).
`margin-right`	Konfiguruje obszar pomiędzy elementem a jego sąsiadami (prawy margines elementu). Można go definiować w jednostkach absolutnych, w procentach względem szerokości tekstu lub automatycznie (wartość `auto`).
`margin-bottom`	Konfiguruje obszar pomiędzy elementem a jego sąsiadami (dolny margines elementu). Można go definiować w jednostkach absolutnych, w procentach względem szerokości tekstu lub automatycznie (wartość `auto`).
`margin-left`	Konfiguruje obszar pomiędzy elementem a jego sąsiadami (lewy margines elementu). Można go definiować w jednostkach absolutnych, w procentach względem szerokości tekstu lub automatycznie (wartość `auto`).
`margin`	Właściwość skrótowa dająca możliwość łatwiejszego określenia właściwości `margin-top`, `margin-right`, `margin-bottom` oraz `margin-left`. Akceptowane wartości tej właściwości to długości wyrażone liczbowo, wartości procentowe lub wartość `auto`.
`padding-top`	Określa odległość pomiędzy górną granicą obszaru a elementami zawartymi w tym obszarze. Odległość może być podawana w dowolnych stosowanych jednostkach miary, w procentach lub automatycznie (wartość `auto`).
`padding-right`	Określa odległość pomiędzy granicą obszaru z prawej strony a elementami zawartymi w tym obszarze. Odległość może być podawana w dowolnych stosowanych jednostkach miary, w procentach lub automatycznie (wartość `auto`).
`padding-bottom`	Określa odległość pomiędzy dolną granicą obszaru a elementami zawartymi w tym obszarze. Odległość może być podawana w dowolnych stosowanych jednostkach miary, w procentach lub automatycznie (wartość `auto`).
`padding-left`	Określa odległość pomiędzy granicą obszaru z lewej strony a elementami zawartymi w tym obszarze. Odległość może być podawana w dowolnych stosowanych jednostkach miary, w procentach lub automatycznie (wartość `auto`).
`padding`	Określa odległości pomiędzy granicą obszaru a elementami zawartymi w tym obszarze. Odległość może być podawana w dowolnych stosowanych jednostkach miary, w procentach lub automatycznie (wartość `auto`).

We właściwościach *margin* oraz *padding* można podać od jednej aż do czterech wartości, które będą interpretowane w następujący sposób:

- jedna wartość — będzie dotyczyła wszystkich czterech marginesów lub wypełnień,
- dwie wartości — pierwsza z nich będzie dotyczyła górnego i dolnego wypełnienia lub marginesu, natomiast druga — prawego i lewego,
- trzy wartości — pierwsza wartość dotyczy górnego marginesu lub wypełnienia, druga — prawego i lewego, natomiast trzecia wartość — dotyczy dolnego marginesu lub wypełnienia.

Ćwiczenie 10.3: Określ marginesy oraz wypełnienie strony

W tym przykładzie określisz marginesy oraz wypełnienie na stronie Księgarni Bookworm. Do arkusza stylów dodasz następujące definicje:

- u góry oraz u dołu strony zdefiniujesz margines o wysokości 50 pikseli, natomiast prawy oraz lewy margines będzie miał szerokość 70 pikseli,
- każdy z nagłówków będzie miał wypełnienie górne oraz dolne o wysokości 15 pikseli.

W celu określenia marginesów oraz wypełnienia będziesz musiał dodać odpowiedni kod do zewnętrznego arkusza stylów (*mystyle.css*) lub umieścić go bezpośrednio na stronie *bookwrm.html* zawierającej osadzony arkusz stylów. Jeśli wcześniej nie stworzyłeś kopii strony zawierającej odwołanie do zewnętrznego arkusza stylów, to jedynym wyjście z sytuacji będzie wykorzystanie strony zawierającej osadzony arkusz stylów. Z tego względu w dalszych przykładach zastosuję to rozwiązanie.

Aby określić marginesy i wypełnienia, możesz posłużyć się skrótową właściwością `margin` z dwiema wartościami określającymi górny i dolny oraz lewy i prawy margines znacznika `<body>` oraz właściwościami `padding-top` oraz `padding-bottom` przypisanymi do znaczników `<h1>` oraz `<h2>` i określającymi odpowiednio górne i dolne wypełnienie. (W poniższym przykładzie zmieniłam sposób zapisu definicji stylu dla znacznika `<body>`, aby ułatwić analizę poszczególnych reguł stylów.)

```
<!--
body { background-color: #ccffcc;
       font-family: Arial, Helvetica, sans-serif;
       color: 330066;
       margin: 50px 70px }
a:link { color: #ff00ff }
a:visited { color: #660000 }
a:hover { color: #ffcc00 }
a:active { color: #ff0000 }
h1 { padding-top: 15px;
     padding-bottom: 15px }
h2 { padding-top: 15px;
     padding-bottom: 15px }
-->
```

Kiedy już zapiszesz nową wersję arkusza stylów, otwórz przeglądarkę obsługującą kaskadowe arkusze stylów i sprawdź rezultaty wprowadzonych modyfikacji. Na rysunku 10.4 przedstawiłam nową wersję strony, na której wyraźnie widać marginesy oraz wypełnienia.

Tło, kolory i obrazy

Zgodnie z tym, czego się dowiedziałeś w rozdziale 7. — „Wykorzystywanie obrazów, koloru i tła", dostępne są narzędzia zapewniające podstawową kontrolę nad wyglądem tła dokumentów HTML. Można użyć atrybutu `bgcolor`, aby określić kolor tła dokumentu lub atrybutu `background`, aby określić obraz tła. Dzięki tabelom niektóre przeglądarki rozszerzają dostępne możliwości i pozwalają autorom stron WWW na określanie tła poszczególnych komórek tabeli.

Rysunek 10.4. *Strona Księgarni Bookworm po określeniu marginesów oraz wypełnień*



Możliwości kontroli, jakie daje CSS są jeszcze większe. Istnieje sześć właściwości służących do kontrolowania tła. Co więcej, dzięki temu, iż style są stosowane do określania postaci poszczególnych elementów strony, istnieje możliwość określenia różnego tła dla różnych elementów. Wszystkie właściwości kontrolujące wygląd tła zostały przestawione w tabeli 10.2.

Tabela 10.2.
Właściwości CSS określające postać tła

Właściwość	Opis
`background-color`	Określa kolor tła dla elementu. Możliwymi wartościami są nazwy kolorów (lub odpowiadający im heksadecymalny „tryplet" `rgb`) lub słowo `transparent`.
`background-image`	Definiuje obraz jako tło elementu. Wartością powinien być adres URL pliku obrazu lub słowo `none`.
`background-repeat`	Jeśli konieczne jest złożenie tła z identycznych obrazów, ten atrybut umożliwi określenie sposobu wyświetlania tła. Wartości atrybutu to `repeat`, `repeat-x` i `repeat-y`. Wartość `repeat` wskazuje, że obraz powinien być powielany w sposób normalny. Zastosowanie wartości `repeat-x` spowoduje powtórzenie obrazu w pojedynczej linii poziomej, a zastosowanie wartości `repeat-y` — w linii pionowej.
`color`	Określa kolor elementu (w większości przypadków dotyczy to koloru tekstu wyświetlanego wewnątrz elementu strony WWW). Kolor można określić, podając jedną z 16 nazw lub używając kilku różnych sposobów zapisu wartości RGB.
`background-attachment`	Określa czy obraz tła jest stały (dołączony do dokumentu), czy też będzie przewijany wraz z nim. Dostępne wartości tej właściwości to: `scroll` oraz `fixed`.
`background-position`	Określa początkowe położenie obrazu tła. Dostępne wartości zostały opisane poniżej w tabeli.
`background`	Właściwość skrótowa umożliwiająca określenie jednej lub kilku powyższych właściwości w jednym miejscu arkusza stylów.

Według zaleceń W3C, definiując obraz tła strony WWW, należy jednocześnie określić kolor tła. Dzięki temu, jeśli obraz tła nie będzie dostępny, to strona przynajmniej będzie miała kolorowe tło.



W dodatku D pt.: „Krótki przewodnik po kaskadowych arkuszach stylów (CSS)" znajdziesz nazwy kolorów oraz opis sposobów zapisu wartości RGB.

Właściwość `background-position` jest dosyć złożona i dlatego wymaga krótkiego wyjaśnienia. Wymaga ona podania dwóch wartości oddzielonych od siebie znakiem odstępu. Wartości te określane są na jeden z poniższych sposobów:

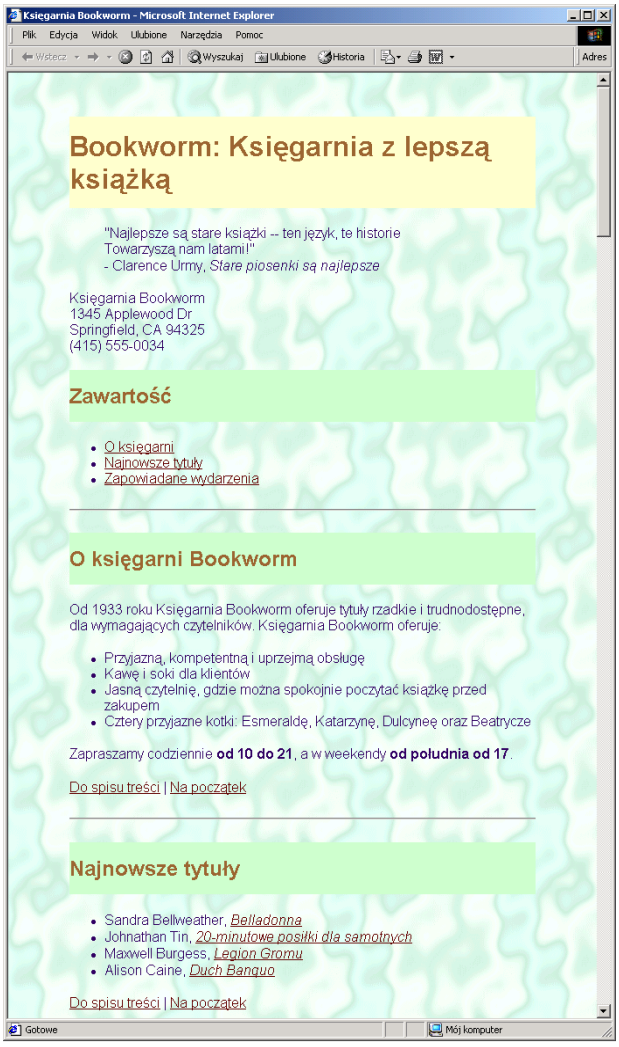
- jako *słowo kluczowe*. Słowa kluczowe: `top`, `center` oraz `bottom` określają pionowe położenie tła strony, natomiast słowa kluczowe: `left`, `center` oraz `right` określają położenie poziome tła. A zatem, aby umieścić obraz tła po środku strony, zarówno w pionie jak i poziomie, należy podać następującą kombinację wartości: `center center`;
- jako *długość*. W tym przypadku podane wartości to współrzędne x i y, przy czym x to współrzędna pozioma (odległość obrazu tła od lewej krawędzi strony) a y to współrzędna pionowa (czyli odległość obrazu tła od górnej krawędzi strony). A zatem, podanie wartości 20 25 spowoduje umieszczenie lewego, górnego wierzchołka obrazu tła w odległości 20 pikseli od lewej krawędzi strony i 25 pikseli od jej górnej krawędzi;

- jako *wartość procentową*. Domyślne położenie obrazu tła (czyli domyślna wartość właściwości `background-position`) to `0% 0%`. Odpowiada ona lewemu, górnemu wierzchołkowi strony WWW. Zastosowanie wartości `100% 100%` spowodowałoby umieszczenie obrazu tła w prawym, dolnym wierzchołku strony.

Ćwiczenie 10.4: Posługując się CSS, określ tło oraz kolory elementów strony

W poprzednich przykładach określiłeś już kolory niektórych elementów strony Księgarni Bookworm. W tym przykładzie dodatkowo rozbudujesz te ustawienia. Przedstawiony poniżej kod CSS spowoduje wyświetlenie na stronie obrazu tła. Stosując właściwość `background-image`, określisz URL obrazu tła i zażądasz od przeglądarki, aby był on powtarzany, jak pokazałam na rysunku 10.5.

Rysunek 10.5. *Strona Księgarni Bookworm z określonym obrazem tła oraz kolorami*





Plik zawierający obraz tła — *background.gif* — wykorzystany w tym przykładzie można znaleźć w witrynie niniejszej książki, pod adresem *http://www.tywebpub.com/*.

Cała strona WWW nie jest jedynym elementem, dla którego można określać tło i kolory. Różne tło i kolory możesz określać niezależnie dla każdego elementu strony. Dla zapewnienia ciekawego efektu, przedstawiony poniżej kod CSS definiuje także kolory nagłówków, na przykład, nagłówki poziomu 1 oraz 2 będą wyświetlane na brązowo.

Nagłówki poziomu 1 będą wyświetlane na jasnożółtym tle (określonym wartością #ffffcc), natomiast nagłówki poziomu 2 na tle jasnozielonym (określonym wartością #ccffcc).

Aby ułatwić Ci analizę reguł stylów podanych w kilku kolejnych przykładach, wszystkie wprowadzane w nich modyfikacje będą wyświetlane na szarym tle. Zmodyfikuj arkusz stylów na stronie bookwrm.html i dopisz do niego linie kodu wyświetlone w poniższym przykładzie na szarym tle:

```
<!--
body { background-color: #ccffcc;
       font-family: Arial, Helvetica, sans-serif;
       color: 330066;
       margin: 50px 70px;
       background-image: url(background.gif);
       background-repeat: repeat }
a:link { color: #ff00ff }
a:visited { color: #660000 }
a:hover { color: #ffcc00 }
a:active { color: #ff0000 }
h1 { color: #996633;
     padding-top: 15px;
     padding-bottom: 15px;
     background-color: #ffffcc }
h2 { color: #996633;
     padding-top: 15px;
     padding-bottom: 15px;
     background-color: #ccffcc }
-->
```

Określanie wyglądu obramowań

Kaskadowe arkusze stylów udostępniają wiele właściwości służących do kontroli postaci obramowań poszczególnych elementów strony. W HTML-u jedynie kilka elementów, takich jak komórki tabel oraz obrazy, mogło mieć obramowania. W CSS obramowanie może teoretycznie mieć każdy element strony.

Właściwości CSS określające postać obramowań elementów stron WWW przedstawiłam w tabeli 10.3.

Tabela 10.3.
Właściwości CSS określające postać obramowań

Właściwość	Opis
`border-style`	Określa styl wszystkich czterech obramowań elementu. Wartości tej właściwości są identyczne jak wartości właściwości `border-bottom-style`. Styl poszczególnych obramowań można także określać niezależnie, posługując się właściwościami `border-bottom-style`, `border-left-style`, `border-top-style` oraz `border-right-style`. Dostępne wartości, które można przypisać tej właściwości to: `none`, `dotted`, `dashed`, `solid`, `double`, `grove`, `ridge`, `inset` oraz `outset`.
`border-color`	Określa kolor wszystkich czterech obramowań elementu. Kolor poszczególnych obramowań można także określać niezależnie, posługując się właściwościami `border-bottom-color`, `border-left-color`, `border-top-color` oraz `border-right-color`. Wartością tych właściwości może być jedna z 16 nazw kolorów, wartość RGB zapisana w jednej z kilku dostępnych postaci lub też wartość `transparent`.
`border-width`	Określa grubość wszystkich czterech obramowań elementu. Grubość p szczególnych obramowań można także określać niezależnie, posługując się właściwościami `border-bottom-width`, `border-left-width`, `border-top-width` oraz `border-right-width`. Dostępne wartości to: `thin`, `medium`, `thick` lub też wartość określająca długość.
`border`	Właściwość skrótowa umożliwiająca jednoczesne określenie grubości, stylu oraz koloru wszystkich czterech obramowań elementu. Wygląd poszczególnych obramowań elementu (jego grubość, kolor oraz styl) można także określa niezależnie, przy użyciu właściwości `border-bottom`, `border-left`, `border-top` oraz `border-right`.

Ćwiczenie 10.5: Zastosowanie CSS do określania obramowań

Przedstawiony poniżej przykład demonstruje sposób zdefiniowania obramowań dla nagłówków wyświetlanych na stronie WWW. Dzięki obramowaniom mogą one nieco przypominać paski z reklamami. Wygląd każdego z nagłówków został zmodyfikowany dzięki zastosowaniu właściwości `border-style`, `border-color` oraz `border-width`. Górne oraz lewe obramowania nagłówków obu typów będą miały inny kolor niż obramowania prawe i dolne. Taki efekt można osiągnąć dzięki przypisaniu właściwości `border-color` dwóch wartości. Nagłówki poziomu 1 będą sprawiały wrażenie, jak gdyby były wypukłe, natomiast nagłówki poziomu 2, jak gdyby były wklęsłe. Ten efekt można z kolei uzyskać, przypisując właściwości `border-style` wartości `outset` oraz `inset` i wykorzystując tę właściwość do określenia postaci wybranych znaczników.

Aby zdefiniować obramowania oraz kolory nagłówków, dodaj do arkusza stylów osadzonego w stronie Księgarni Bookworm, kod wyświetlony w poniższym przykładzie na szarym tle:

```
<!--
body { background-color: #ccffcc;
       font-family: Arial, Helvetica, sans-serif;
       color: 330066;
       margin: 50px 70px;
       background-image: url(background.gif);
       background-repeat: repeat }
a:link { color: #ff00ff }
a:visited { color: #660000 }
a:hover { color: #ffcc00 }
```

```
a:active { color: #ff0000 }
h1 { color: #996633;
     padding-top: 15px;
     padding-bottom: 15px;
     background-color: #ffffcc;
     border-color: #cccc33 #cc9933;
     border-style: outset;
     border-width: thin }
h2 { color: #996633;
     padding-top: 15px;
     padding-bottom: 15px;
     background-color: #ccffcc;
     border-color: #9cc33 #996633;
     border-style: inset;
     border-width: thin }
-->
```

Po wyświetleniu strony w przeglądarce obsługującej kaskadowe arkusze stylów, powinna ona przypominać stronę przedstawioną na rysunku 10.6 (widoczna na nim strona została wyświetlona w Internet Explorerze). Netscape Navigator wyświetla obramowania i tło w nieco inny sposób. Obramowania wyświetlane w przeglądarce Internet Explorer są nieco grubsze, natomiast w przeglądarce Netscape, cieńsze i nieco przesunięte względem kolorowego tła elementu.

Wynik

Rysunek 10.6.

Strona wykorzystująca właściwości obramowań wyświetlona w Internet Explorerze



Określanie czcionek i stylów

Kaskadowe arkusze stylów dysponują obszerną grupą właściwości służących do określania postaci czcionek. W rzeczywistości kaskadowe arkusze stylów dają autorom strony WWW znacznie większą kontrolę nad wyglądem używanych czcionek niż ta, którą zapewniał bardzo prosty znacznik `<font>`.

Podstawowe właściwości służące do określania postaci czcionek zostały przedstawione w tabeli 10.4.

Tabela 10.4.
Właściwości CSS określające postać czcionek

Właściwość	Opis
`font-family`	Określa krój czcionki. Pozwala na podanie nazwy czcionki (na przykład: Arail, Times lub Palatino),bądź jednej z pięciu ogólnych nazw czcionek: `serif`, `sans-serif`, `cursive`, `fantasy` lub `monospace`.
`font-size`	Określa wielkość czcionki, wyrażoną w wielkościach bezwzględnych, względnych lub jako wartość procentową.
`font-style`	Określa styl czcionki, może to być jedna z wartości: `oblique`, `italic` lub `normal`.
`font-weight`	Określa grubość (wagę) czcionki, można ją ustalić przy użyciu wartości `normal`, `bold`, `bolder`, lub `lighter`.
`font-variant`	Określa modyfikację czcionki, może to być wartość `small-caps` lub `normal`.
`font`	Właściwość skrótowa umożliwiająca określenie wartości właściwości `font-weight`, `font-size`, `font-style`, `font-family` oraz `line-height` w jednym miejscu arkusza stylów. Właściwość `line-height` została szczegółowo opisana w następnym podrozdziale, pt.: „Właściwości CSS określające wyrównanie tekstu".

A oto jeszcze kilka dodatkowych informacji dotyczących powyższych właściwości o których warto wiedzieć:

- definiując rodzinę czcionek za pomocą właściwości `font-family`, możesz podać kilka nazw, oddzielając je przecinkami i wymieniając w porządku preferencji. Jeśli wybrana przez Ciebie czcionka nie jest dostępna, system sięgnie po następną nazwę na liście. W3C radzi, aby nazwa ogólna była umieszczana na ostatnim miejscu. Wynika to z faktu, iż każda przeglądarka dysponuje czcionką domyślną dla danej rodziny ogólnej;
- używając nazw ogólnych rodzin czcionek, należy uważać podczas stosowania rodzin `cursive` oraz `fantasy`. Wygląd tych czcionek jest w głównej mierze zależny od czcionek zainstalowanych na dysku osoby przeglądającej stronę. Czcionki `serif`, `sans-serif` oraz `monospace` są zazwyczaj instalowane wraz z systemem operacyjnym bądź przeglądarką, nie zawsze dotyczy to czcionek `cursive` oraz `fantasy`;
- w przypadku określania bezwzględnej wielkości czcionki za pomocą właściwości `font-size` stosowane są słowa kluczowe, takie jak `xx-smoll`, `x-small`, `samll`, `medium`, `large`, `x-large` oraz `xx-large`. Wartości te odpowiadają konkretnym wielkościom czcionek w przeglądarce. Wielkości względne wyrażane są w stosunku do wielkości czcionki elementu nadrzędnego i mogą być zdefiniowane jako większe lub mniejsze;

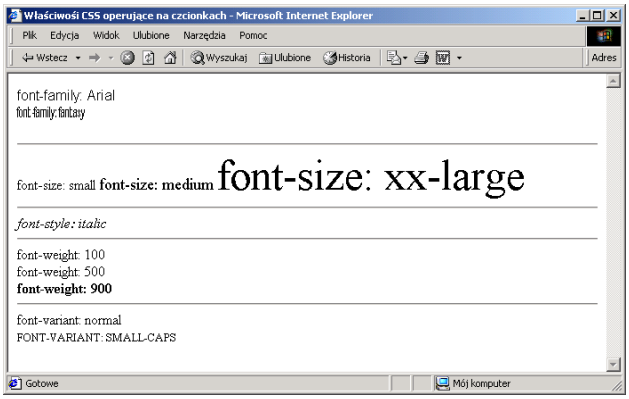
- **grubości czcionek określane są wartościami liczbowymi**: `100`, `200`, `300`, `400`, `500`, `600`, `700`, `800` **oraz** `900`. **Właściwości** `font-weight` **można przypisać jedną z tych liczb lub jedną z wartości** `normal`, `bold`, `bolder` **lub** `lighter`. **Wartość** `normal` **odpowiada wartości** `400`, **natomiast** `bold` **— wartości** `700` **na skali liczbowej. Wartości** `bolder` **oraz** `lighter` **określają grubość czcionki w zależności od elementu nadrzędnego.**

Przedstawiony poniżej fragment kodu demonstruje wykorzystanie kilku właściwości określających postać czcionek. Wygląd tej strony WWW został przedstawiony na rysunku 10.7.

Źródło

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/strict.dtd">
<html>
<head>
<title>Właściwości CSS operujące na czcionkach</title>
<style type="text/css">
<!--
BODY { background-color: #ffffff }
-->
</style>
</head>
<body>
<p><span style="font-family: Arial">font-family: Arial</span><br />
   <span style="font-family: fantasy">font-family: fantasy</span></p>
<hr />
<span style="font-size: small">font-size: small</span>
<span style="font-size: medium">font-size: medium</span>
<span style="font-size: xx-large">font-size: xx-large</span>
<hr />
<span style="font-style: italic">font-style: italic</span>
<hr />
<span style="font-weight: 100">font-weight: 100</span>
<br />
<span style="font-weight: 500">font-weight: 500</span>
<br />
<span style="font-weight: 900">font-weight: 900</span>
<hr />
<span style="font-variant: normal">font-variant: normal</span>
<br />
<span style="font-variant: small-caps">font-variant: small-caps</span>
</body>
</html>
```

Wynik



Rysunek 10.7. *Różne właściwości określające postać czcionek*

Właściwości CSS określające wyrównanie tekstu

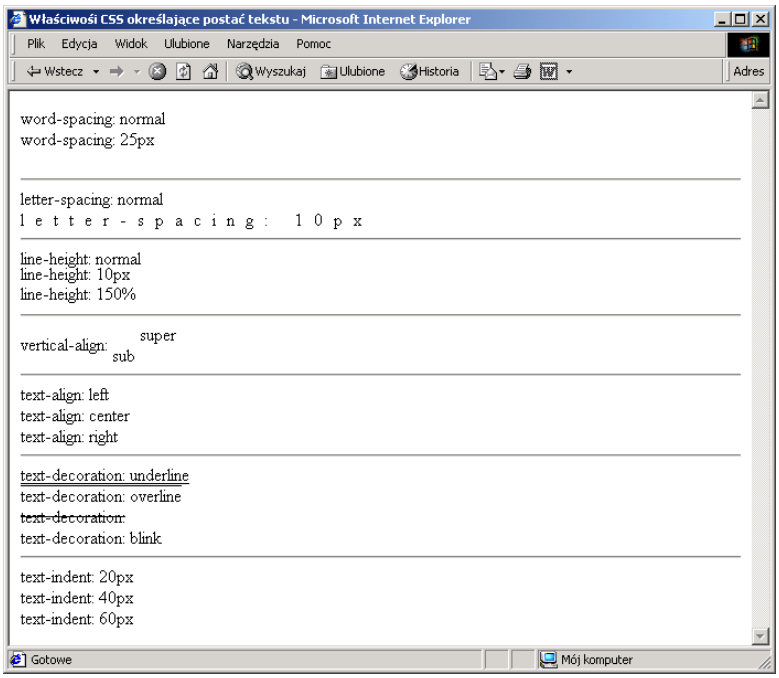
Oprócz właściwości określających postać czcionek, kaskadowe arkusze stylów posiadają także wiele cech pozwalających autorom stron WWW na kontrolowanie wyrównania tekstu. Właściwości te dają możliwość uzyskania dokładnej typograficznej kontroli nad tekstem, jaką posiadają edytory tekstu oraz programy poligraficzne. Taki stopień kontroli, przed opracowaniem CSS, nie był dostępny dla twórców stron WWW. Podstawowe właściwości służące do określania wyrównania tekstu zostały przedstawione w tabeli 10.5.

Tabela 10.5.
Właściwości CSS określające wyrównanie tekstu

Właściwość	Opis
`word-spacing`	Umożliwia zwiększenie domyślnych odstępów pomiędzy wyrazami. Odstępy są podawane w jednostkach absolutnych. Wartość domyślna to `normal`.
`letter-spacing`	Umożliwia zwiększenie domyślnych odstępów pomiędzy literami. Odstępy są podawane w jednostkach absolutnych. Wartość domyślna to `normal`.
`line-height`	Określa odległość pomiędzy liniami bazowymi dwóch kolejnych linii. Wartość liczbowa określa, że wysokość linii będzie wielkością czcionki pomnożoną przez podaną liczbę, na przykład, `line-height: 2` tworzy linię o wysokości równej dwukrotnej wielkości czcionki. Właściwości `line-height` można także przypisać wartość bezwzględną. Na przykład, `line-height: 15px` oznacza, że wysokość linii będzie wynosiła 15 pikseli. W przypadku podania wartości procentowej, wysokość linii jest obliczana na podstawie wysokości elementu. Na przykład, `line-height: 200%` jest równoważne podaniu wartości liczbowej: `line-height: 2`. Można także użyć wartości `normal`, która sprawi, że linia będzie miała domyślną wysokość. Aktualnie właściwość ta jest obsługiwana przez przeglądarkę Netscape Navigator 4.0.
`vertical-align`	Określa absolutną pozycję tekstu w pionie lub jego pozycję względem elementu nadrzędnego. Oto lista jego wartości: `baseline`, `middle`, `sub`, `super`, `text-top` i `text-bottom` (określają pozycję względem elementu nadrzędnego) oraz `top` i `bottom` (określają pozycję względem wiersza). Stosując procentowe wartości ujemne lub dodatnie, obniżamy lub podnosimy element ponad linię bazową elementu nadrzędnego.
`text-align`	Umożliwia nie tylko to, co standardowe wyrównanie HTML (wartości `left`, `right`, `center`), ale także wyrównanie do obu marginesów (`justify`). Wartość `justify` tej właściwości jest aktualnie obsługiwana przez przeglądarkę Netscape Navigator 4.0, Internet Explorer 4.0 jej nie obsługuje.
`text-decoration`	Umożliwia wprowadzanie dodatkowych ozdobników, takich jak podkreślenie, przekreślenie i miganie. Może przyjmować cztery wartości: `none`, `overline`, `line-through` i `blink`.
`text-indent`	Określa wcięcie pierwszej linii sformatowanego tekstu w elemencie. Wartość ta może być wartością bezwzględną lub wartością procentową, bazującą na szerokości elementu. Wartości procentowe określane są na podstawie szerokości elementu i w przeważającej większości przypadków działają najlepiej. Aktualnie właściwość ta jest obsługiwana w Netscape Navigatorze 4.0, Internet Explorer 4.0 nie obsługuje jej.
`white-space`	Atrybut określa odstępy i złamania wiersza w obrębie elementu. Wartości są następujące: `normal` — w tym przypadku dodatkowe odstępy są ignorowane, tak jak w przypadku zwykłego HTML, `pre` — jak w formatowanym wstępnie za pomocą znacznika `<pre>` tekście HTML lub `nowrap` — wiersze są łamane tylko za pomocą znacznika ` `.

Jak widać, przedstawione tu właściwości CSS są obsługiwane inaczej w programie Internet Explorer 4.0 oraz Netscape Navigatorze 4.0. Niektóre właściwości są odczytywane tylko w jednej z tych przeglądarek, natomiast inne, w obu. A zatem, jeśli w pewnym momencie stracisz rozeznanie co się dzieje, to wiedz, że wina niekoniecznie będzie leżała po Twojej stronie.

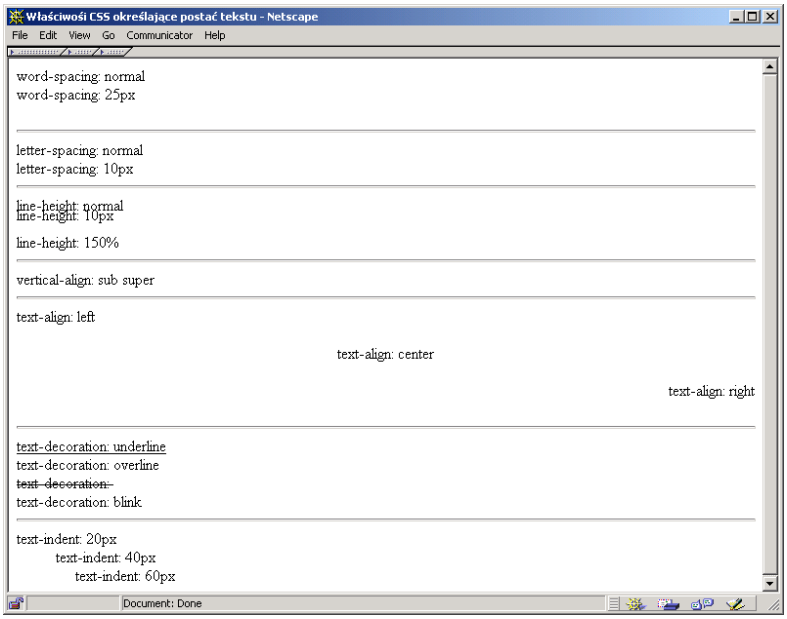
Fragment kodu przedstawiony poniżej demonstruje wiele właściwości służących do określania postaci tekstu i wykorzystuje je do sformatowania różnych elementów tekstowych. Zwróć uwagę, w jak odmienny sposób ten sam kod jest wyświetlany w dwóch najpopularniejszych przeglądarkach. Właśnie na tym przykładzie możesz się przekonać, dlaczego warto sprawdzać strony w wielu różnych przeglądarkach. Rysunki 10.8 oraz 10.9 ukazują, w jaki sposób poniższa strona wygląda w przeglądarkach Internet Exolorer 4 oraz w Netscape Navigator 4.



Rysunek 10.8. *Różne właściwości tekstu wyświetlone w przeglądarce Internet Explorer 4*

Style lokalne

Zanim wykorzystasz właściwości określające postać tekstu i czcionek do zmodyfikowania wyglądu strony Księgarni Bookworm, chciałam przedstawić ostatni typ arkuszy stylów — *style lokalne*. Metoda ta pozwala określić styl nie dla całej strony, lecz dla jej konkretnego elementu. Jeśli na stronie znajduje się nagłówek lub akapit, na który chcesz zwrócić uwagę czytelnika poprzez wykorzystanie odmiennego koloru lub wyrównania tekstu i jeśli chcesz wykorzystać ten szczególny sposób formatowania tylko na jednej stronie witryny, to możesz do tego celu użyć stylu lokalnego.



Rysunek 10.9. *Różne właściwości tekstu wyświetlone w przeglądarce Netscape 4*

HTML zawiera kilka atrybutów, które pozwalają dołączać reguły stylów do znaczników HTML. Podstawowymi atrybutami wykorzystywanymi w tym celu są atrybuty `style` oraz `class`. Dostępny jest także znacznik `<span>`, także wykorzystywany wraz z arkuszami stylów. Wszystkie te elementy zostaną opisane w dalszej części podrozdziału.

Atrybut STYLE

We wcześniejszej części rozdziału, a konkretnie w podrozdziale pt.: „Osadzone arkusze stylów" poznałeś znacznik `<style>`, który pozwalał na dołączenie do standardowej strony WWW zewnętrznego arkusza stylów. Atrybut `style` pozwala na dołączenie reguły stylu do konkretnego *elementu* strony.

Zewnętrzne oraz osadzone arkusze stylów umożliwiają rozdzielenie definicji stylów od zawartości dokumentu HTML. W odróżnieniu od nich, atrybut `style` jest umieszczany wewnątrz kodu treści dokumentu.



Atrybut `style` można przypisać każdemu znacznikowi HTML z wyjątkiem znaczników: `<base>`, `<basefont>`, `<head>`, `<html>`, `<meta>`, `<param>`, `<script>`, `<style>` **oraz** `<title>`.

Załóżmy, że chciałbyś zmienić kolor wybranego akapitu strony, posługując się w tym celu regułą stylu. Zazwyczaj tekst jest wyświetlany na stronie w kolorze czarnym i nie jest w żaden sposób wyróżniany. Ty chcesz, aby jeden akapit tekstu został pogrubiony i wyświetlony na czerwono.

Wartość przypisywana atrybutowi `style` (kolor `red` oraz grubość czcionki `bold`) zostają zapisane w cudzysłowach. Reguły stylów definiujące te atrybuty (`color: red` oraz `font-weight: bold`) zostają zapisane wewnątrz cudzysłowów i oddzielone od siebie średnikiem.

Przedstawiony poniżej fragment kodu demonstruje zastosowanie atrybutu `style` do określenia wyglądu akapitu:

```
<p style="color: red: font-weight: bold">Chcę, aby ten akapit był wytłuszczony
i wyświetlony na czerwono.</P>
```

Atrybut CLASS

Klasa jest zdefiniowanym stylem o szerokim zasięgu, określającym właściwości dla niektórych lub wszystkich elementów tworzących dokument. Klasy są definiowane w zewnętrznych arkuszach stylów lub w nagłówku standardowego dokumentu HTML, wykorzystującego osadzony arkusz stylów. Następnie specjalny atrybut `class` przypisuje określoną klasę jednemu lub kilku elementom strony WWW. Zaletą atrybutu `class` w porównaniu z atrybutem `style` jest możliwość wykorzystania tej samej reguły stylu do określenia postaci większej ilości elementów strony, bez konieczności jej wielokrotnego wpisywania.



Atrybut `class` można przypisać każdemu znacznikowi HTML z wyjątkiem znaczników: `<base>`, `<basefont>`, `<head>`, `<html>`, `<meta>`, `<param>`, `<script>`, `<style>` **oraz** `<title>`.

Przedstawiony poniżej fragment kodu CSS demonstruje dwie reguły określające wygląd akapitów. Pierwsza z nich określa wygląd „normalnego" akapitu strony, natomiast druga definiuje specjalną klasę, która zostanie użyta do określenia wyglądu wybranych akapitów. Ta specjalna klasa nosi nazwę `p.bigger`, gdzie `p` rozpoczyna definicję znacznika akapitu, a `bigger` oznacza nazwę klasy.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/strict.dtd">
<html>
<head>
<title>Stosowanie klas</title>
<style type="text/css">
<!--
p { font-family Arial, Helvetica, sans-serif; color: black }
p.bigger { font-family Arial, Helvetica, sans-serif; font-size: larger }
-->
</style>
</head>
```

Teraz, gdy już zdefiniowałeś specjalną klasę o nazwie `bigger`, należy dołączyć ją do wybranego akapitu strony, którego wygląd chcesz zmodyfikować. Aby użyć klasy `bigger` do określenia postaci akapitu, należy wykorzystać fragment kodu o następującej składni:

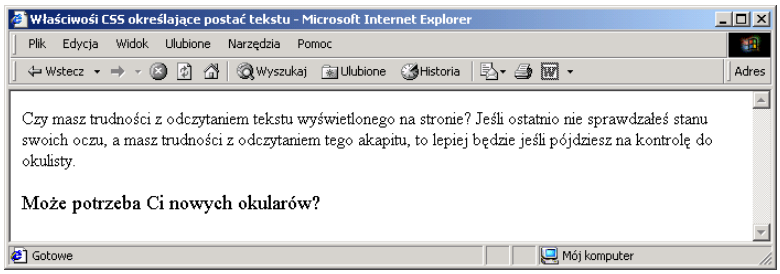
```
<p class="bigger">
```

Aby dokładniej zademonstrować sposób wykorzystania klasy, poniższy przykład przedstawia kod prostej strony WWW. Strona ta składa się z dwóch akapitów: pierwszy jest normalny, natomiast drugi został sformatowany przy użyciu stylu `bigger`.

Źródło

```
<body>
<p>Czy masz trudności z odczytaniem tekstu wyświetlonego na stronie?
Jeśli ostatnio nie sprawdzałeś stanu swoich oczu, a masz trudności
z odczytaniem tego akapitu, to lepiej będzie jeśli pójdziesz na
kontrolę do okulisty.</p>
<p class="bigger">Może potrzeba Ci nowych okularów?</p>
</body>
</html>
```

Wynik



Rysunek 10.10. *Zastosowanie atrybutu CLASS do określenia wyglądu akapitu*

Znacznik

Przykłady przedstawione w tym podrozdziale wykorzystują znacznik `<span>`. Umożliwia on sformatowanie wybranego fragmentu tekstu przy użyciu stylów a jednocześnie, w żaden sposób nie określa jego struktury (co oznacza, że tekst nie jest zapisywany w żadnym konkretnym, strukturalnym znaczniku HTML). Znacznika `<span>` można nawet użyć do określenia postaci pierwszej litery lub pierwszego słowa strony. Nie wywiera on żadnego wpływu na wygląd tekstu, co oznacza, iż jego postać będzie określał wyłącznie styl.

Ćwiczenie 10.6: Zastosowanie właściwości CSS określających postać czcionek i tekstu

W ostatnim ćwiczeniu przedstawionym w tym rozdziale zmodyfikujesz kod poprzednich przykładów, dodając do niego właściwości określające postać czcionek. Zwróć uwagę, iż w przykładzie wykorzystana została ogólna rodzina czcionek o nazwie `fantasy`. Zazwyczaj czcionki tego rodzaju są instalowane wraz z programami graficznymi lub poligraficznymi. Jeśli na swoim komputerze nie masz zainstalowanej żadnej czcionki tego typu, zamień wszystkie wystąpienia słowa `fantasy` słowem `fixed`. Pierwsza część strony bookwrm.html została przedstawiona poniżej:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/strict.dtd">
<html>
<head>
<title>Księgarnia Bookworm</title>
<style type="text/css">
<!--
body { background-color: #ccffcc;
       font-family: Arial, Helvetica, sans-serif;
       color: 330066;
       margin: 50px 70px;
```

```
        background-image: url(background.gif);
        background-repeat: repeat }
a:link { color: #ff00ff }
a:visited { color: #660000 }
a:hover { color: #ffcc00 }
a:active { color: #ff0000 }
h1 { color: #996633;
     padding-top: 15px;
     padding-bottom: 15px;
     background-color: #ffffcc;
     border-color: #cccc33 #cc9933;
     border-style: outset;
     border-width: thin
```

Teraz powinieneś dodać regułę stylu zmieniającą czcionkę, jaką będą wyświetlane nagłówki H1, na fantasy (lub fixed, jeśli nie masz zainstalowanych żadnych czcionek typu `fantasy`) oraz wyśrodkować treść nagłówka na stronie przy użyciu właściwości `text-align`. Możesz to zrobić, dopisując do arkusza stylu następujący kod:

```
font-family: "fantasy";
     text-align: center }
```

Oto dalsza część istniejącego kodu strony:

```
h2 { color: #996633;
     padding-top: 15px;
     padding-bottom: 15px;
     background-color: #ccffcc;
     border-color: #9cc33 #996633;
     border-style: inset;
     border-width: thin
```

Zawartość nagłówków H2 także chcesz wyświetlić czcionką typu fantasy i wyśrodkować na stronie:

```
font-family: "fantasy";
     text-align: center }
```

Kod stylu na stronie *bookwrm.html* kończy się definicją reguły stylu dla znacznika `<blockquote>` o następującej postaci:

```
blockquote { font-family: "Book Antiqua";
     line-height: 12pt;
     font-weight: normal;
     font-variant: normal;
     color: #996633;
     word-spacing: 2em;
     text-align: center }
```

Poniżej sekcji stylu dla znacznika `<blockquote>`, dodaj klasę o nazwie `fantasy` (lub `fixed`, jeśli na Twoim komputerze nie zostały zainstalowane żadne czcionki typu `fantasy`). Klasy tej użyjesz do wyświetlenia zawartości niektórych elementów strony czcionką `fantasy` (lub `fixed`). Aby zdefiniować klasę, wpisz do arkusza stylów poniższy fragment kodu, zastępując w razie konieczności słowo `fantasy` słowem `fixed`:

```
.fantasy { text-align: left;
     font-family: "fantasy";
     font-size: 16pt;
     color: #996600 }
```

Osadzony arkusz stylów kończy się w standardowy sposób:

```
-->
</style>
</head>
```

Ale to jeszcze nie wszystko! Wciąż pozostaje Ci wykorzystanie stylu lokalnego do określenia postaci fragmentu strony. Istnieją dwa fragmenty strony, których postać chcesz określić w taki sposób. Nazwa księgarni pojawia się w kilku miejscach na stronie i to właśnie te fragmenty chcesz sformatować stylem `fantasy` (lub `fixed`). Poniższe dwa odcinki kodu pokazują, gdzie i jakich zmian należy dokonać. Oto pierwszy fragment:

```
<p><span style="fantasy">Księgarnia Bookworm</span><br />
1345 Applewood Dr<br />
Springfield, CA 94325<br />
(415) 555-0034
</p>
```

I drugi fragment:

```
<p>Od 1933 roku <span style="fantasy">Księgarnia Bookworm</span> oferuje tytuły
rzadkie i trudnodostępne,
dla wymagających czytelników. Księgarnia Bookworm oferuje:</p>
```

Poniżej przedstawiłam pełny listing ostatecznej wersji kodu strony Księgarni Bookworm, jej początkowa część, wyświetlona w przeglądarce Internet Explorer, została także przedstawiona na rysunku 10.11:

Źródło

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/strict.dtd">
<html>
<head>
<title>Księgarnia Bookworm</title>
<style type="text/css">
<!--
body { background-color: #ccffcc;
       font-family: Arial, Helvetica, sans-serif;
       color: 330066;
       margin: 50px 70px;
       background-image: url(background.gif);
       background-repeat: repeat }
a:link { color: #ff00ff }
a:visited { color: #660000 }
a:hover { color: #ffcc00 }
a:active { color: #ff0000 }
h1 { color: #996633;
     padding-top: 15px;
     padding-bottom: 15px;
     background-color: #ffffcc;
     border-color: #cccc33 #cc9933;
     border-style: outset;
     border-width: thin;
     font-family: "fantasy";
     text-align: center }
h2 { color: #996633;
     padding-top: 15px;
     padding-bottom: 15px;
     background-color: #ccffcc;
     border-color: #9cc33 #996633;
```

```
     border-style: inset;
     border-width: thin;
     font-family: "fantasy";
     text-align: center }
blockquote { font-family: "Book Antiqua";
     line-height: 12pt;
     font-weight: normal;
     font-variant: normal;
     color: #996633;
     word-spacing: 2em;
     text-align: center }
.fantasy { text-align: left;
     font-family: "fantasy";
     font-size: 16pt;
     color: #996600 }
-->
</style>
</head>
<body>
<a name="top"><h1>Bookworm: Księgarnia z lepszą książką</h1></a>
<blockquote>
"Najlepsze są stare książki - ten język, te historie<br />
Towarzyszą nam latami!"<br />
- Clarence Urmy, <cite>Stare piosenki są najlepsze</cite>
</blockquote>
<p><span class="fantasy">Księgarnia Bookworm</span><br />
1345 Applewood Dr<br />
Springfield, CA 94325<br />
(415) 555-0034
</p>
<a name="contents"><h2>Zawartość</h2></a>
<ul>
  <li><a href="#about">O księgarni</a></li>
  <li><a href="#recent">Najnowsze tytuły</a></li>
  <li><a href="#upcoming">Zapowiadane wydarzenia</a></li>
</ul>
<hr />
<a name="about"><h2>O księgarni Bookworm</h2></a>
<p>Od 1933 roku <span class="fantasy">księgarnia Bookworm</span> oferuje tytuły
rzadkie i trudnodostępne, dla wymagających czytelników. Księgarnia Bookworm
oferuje:</p>
<ul>
<li>Przyjazną, kompetentną i uprzejmą obsługę</li>
<li>Kawę i soki dla klientów</li>
<li>Jasną czytelnię, gdzie można spokojnie poczytać książkę przed zakupem</li>
<li>Cztery przyjazne kotki: Esmeraldę, Katarzynę, Dulcyneę oraz Beatrycze</li>
</ul>
<p>Zapraszamy codziennie <strong>od 10 do 21</strong>, a w weekendy
<strong>od południa od 17</strong>.</p>
<p><a href="#contents">Do spisu treści</a> | <a href="#top">
Na początek</a></p>
<hr />
<a name="#recent"><h2>Najnowsze tytuły</h2></a>
<ul>
<li>Sandra Bellweather, <a href="belladonna.html">
<cite>Belladonna</cite></a></li>
<li>Johnathan Tin, <a href="20minmeals.html">
<cite>20-minutowe posiłki dla samotnych</cite></a></li>
<li>Maxwell Burgess, <a href="legion.html">
<cite>Legion Gromu</cite></a></li>
<li>Alison Caine, <a href="banquo.html">
<cite>Duch Banquo</cite></a></li>
</ul>
```

```
<p><a href="#contents">Do spisu treści</a> | <a href="#top">
Na początek</a></p>
<a name="upcoming"><h2>Zapowiadane wydarzenia</h2></a>
<ul>
<li><b>Środowy Przegląd Książek</b>, czyli spotkania i dyskusje przy kawie. Zadzwoń
do nas po informacje, jak się przyłączyć do grupy oraz o czym będziemy rozmawiali
w tym tygodniu.</li>
<li><b>Godzina dla Dzieciaków</b>, to spotkania w każdą sobotę o 13, by poczytać,
w coś zagrać lub inaczej spędzić wolny czas. Podajemy również mleko i
ciasteczka.</li>
<li>W piątek 16 września księgarnię odwiedzi <b>Carole Fenney</b>, aby recytować
fragmenty swego zbioru wierszy <cite>Pająki na stronach WWW</cite>.</li>
<li><b>Księgarnia będzie nieczynna</b> 1 października z powodu przeprowadzki
rodziny nietoperzy, która uwiła sobie gniazdko w wieży. Lubimy je bardzo, ale ten
bałagan, który robią...</li>
</ul>
<p><a href="#contents">Do spisu treści</a> | <a href="#top">
Na początek</a></p>
<address>
Osatnia aktualizacja: 11/11/99<br />
WebMaster: Laura Lemay lemay@bookworm.com<br />
&#169; copyright 1999 Księgarnia Bookworm<br />
</address>
</body>
</html>
```

Wynik



Rysunek 10.11. *Strona Księgarni Bookworm po dodaniu stylów określających postać czcionek i tekstu*



Jeśli wyświetlisz tę stronę w Netscape Navigatorze, to najprawdopodobniej zauważysz, że znacznik jest traktowany tak samo, jak znacznik <br>; czyli sprawia, że tekst umieszczony za znacznikiem zostanie wyświetlony w nowej linii. Wiedząc o tym, zapewne nie będziesz chciał umieszczać znaczników wewnątrz akapitu, tak jak zostały zapisane w powyższym przykładzie.

Źródła informacji o kaskadowych arkuszach stylów

W tym rozdziale bardzo pobieżnie przedstawiłam kaskadowe arkusze stylów. Wynika to z faktu, iż CSS jest bardzo rozległym zagadnieniem, a żadna przeglądarka nie implementuje stylów na tyle dobrze, aby można było szczegółowo opisywać ich możliwości.

Jeśli pragniesz zdobyć dodatkowe informacje dotyczące kaskadowych arkuszy stylów, możesz je znaleźć na witrynie W3C, gdzie przechowywane są specyfikacje CSS. Strona poświęcona CSS dostępna jest pod adresem *http://www.w3.org/Style/CSS/*.

Poniżej podałam listę innych witryn, na których można znaleźć informacje o kaskadowych arkuszach stylów.

- *Webreview.com*. Na tej witrynie można znaleźć przewodnik po CSS. Najbardziej przydatna część tego przewodnika zawiera informacje dotyczące znaczników CSS zaimplementowanych w poszczególnych przeglądarkach.
- *Cascading Style Sheets*. Ten dokładny opis opracowany przez Web Design Group zawiera informacje o właściwościach arkuszy stylów wprowadzonych w CSS1. Można go znaleźć pod adresem *http://www.htmlhelp.com/reference/css/*.
- *W3C CSS Home Page*. Na tej stronie możesz znaleźć obszerną listę połączeń do wszelkiego typu przykładów, historii oraz najnowszych trendów rozwoju arkuszy stylów oraz innych technologii związanych z WWW.

Warsztat

To już koniec tego długiego rozdziału wypełnionego przeróżnymi informacjami. Jeśli chcesz utrwalić omawiane w nim zagadnienia, to skorzystaj z pytań i ćwiczeń podanych w tej części rozdziału. Dzięki nim łatwiej zapamiętasz najistotniejsze informacje dotyczące kaskadowych arkuszy stylów, podane w tym rozdziale.

Pytania i odpowiedzi

P. W jaki sposób mam określić, którego typu arkuszy stylów należy użyć? Dokonanie tego wyboru przysparza mi wielu problemów.

O. Nie przejmuj się, nie jesteś jedyną osobą, która ma z tym problemy. Ogólnie rzecz biorąc, istnieje kilka metod określania typu arkuszy stylów, jakie należy zastosować. Jeśli masz zamiar wprowadzić jakiś unikalny efekt, który nie ma krytycznego znaczenia dla całej prezentacji, to możesz wybrać ten rodzaj arkusza stylów, który Ci najbardziej odpowiada. Jeśli chcesz, aby Twoje strony mogło oglądać jak najwięcej

osób, to najlepszym rozwiązaniem będzie wykorzystanie kaskadowych arkuszy stylów, gdyż są one najlepiej obsługiwane w przeglądarkach.

P. Mam problemy ze zrozumieniem właściwości związanych z kolorem i obrazem tła. Czy nie lepiej by było zamiast nich zastosować tabel i określać kolor lub obraz tła w poszczególnych komórkach?

O. Masz całkowitą rację, przy użyciu tabel można osiągnąć podobny efekt. Jednak jest pewien problem związany ze znacznikami tabel. Po pierwsze, nie są to w pełni strukturalne znaczniki, a co za tym idzie, niektóre przeglądarki nie są w stanie ich poprawnie wyświetlać. Chodzi tu o przeglądarki, takie jak Lynx lub prezentujące zawartość strony w formie dźwiękowej. Oznacza to, że tekst umieszczony w tabeli może zostać wyświetlony w nieodpowiedniej kolejności i wyglądać inaczej, niż zaplanował twórca strony. Uzyskując bardzo podobne wyniki przy zastosowaniu arkuszy stylów, można się skoncentrować na zapewnieniu strukturalnej postaci dokumentu. Dzięki temu, nawet w przeglądarkach, które nie obsługują arkuszy stylów, strona może być bardziej czytelna i przydatna.

Quiz

1. Jaka jest różnica pomiędzy dokumentami HTML typu ścisłego (`strict`) oraz pośredniego (`transitional`)?
2. Czym jest reguła stylu CSS?
3. Jakie są trzy podstawowe sposoby tworzenia kaskadowych arkuszy stylów?
4. Jakiej metody należy użyć, aby stworzyć arkusz stylów i przy jego użyciu określić postać wielu stron WWW? Jakie rozszerzenie pliku powinieneś zastosować?
5. Prawda czy fałsz: Na tej samej stronie WWW można użyć zewnętrznych arkuszy stylów, osadzonych arkuszy stylów oraz stylów lokalnych.

Odpowiedzi

1. W dokumentach HTML typu ścisłego (`strict`) formatowanie strony jest określane tylko i wyłącznie przy użyciu arkuszy stylów; dotyczy to także kolorów oraz wyróżniania tekstów. Dokumenty HTML typu pośredniego wykorzystują arkusze stylów oraz znaczniki i atrybuty HTML umożliwiające określanie sposobu formatowania dokumentu w starszych typach przeglądarek.
2. Reguła stylu CSS definiuje styl, który ma określać wygląd elementu HTML. Reguła składa się z selektora (jakim może być znacznik HTML), po którym podawana jest deklaracja określająca właściwości i wartości tego selektora.
3. Kaskadowe arkusze stylów można tworzyć jako zewnętrzne arkusze stylów, osadzone arkusze stylów oraz jako lokalne arkusze stylów.
4. W przypadku, gdy chcesz użyć arkusza stylów do określenia postaci wielu stron WWW, najlepszym rozwiązaniem jest stworzenie go jako arkusza zewnętrznego. Zewnętrzne arkusze stylów są zapisywane w plikach z rozszerzeniem *.css*.

5. To prawda. Właściwości zdefiniowane w osadzonych arkuszach stylów mają wyższy priorytet, niż właściwości zdefiniowane w arkuszach zewnętrznych. Podobnie, właściwości zapisane w arkuszach lokalnych mają wyższy priorytet, niż właściwości zdefiniowane w osadzonych arkuszach stylów.

Ćwiczenia

1. Stwórz własną stronę WWW i określ jej postać przy użyciu własnych reguł stylów. Style zapisz początkowo w zewnętrznym arkuszu stylów, który będziesz mógł wykorzystać do określenia wyglądu większej ilości stron.
2. Zmodyfikuj stronę stworzoną w poprzednim ćwiczeniu w taki sposób, aby wykorzystywała osadzony arkusz stylów. Jeśli lubisz przygody, to dołącz do tej strony także zewnętrzny arkusz stylów. Do osadzonego arkusza stylów dodaj style przesłaniające style zdefiniowane w arkuszu zewnętrznym i zobacz, jakie będą wyniki tych eksperymentów.