

IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK

DODAJ DO KOSZYKA

CENNIK I INFORMACJE

ZAMÓW INFORMACJE
O NOWOŚCIACH

ZAMÓW CENNIK

CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

HTML, XHTML i CSS. Biblia

Autorzy: Bryan Pfaffenberger, Steven M. Schafer,
Chuck White, Bill Karow

Tłumaczenie: Piotr Cieślak (rozdz. 33 – 34, dod. A, B),
Włodzimierz Gajda (rozdz. 8, 10), Marcin Jagodziński
(rozdz. 25, 29), Marek Pałczyński (rozdz. 6, 7, 9, 11 – 15),
Piotr Rajca (rozdz. 26 – 28), Marcin Samodulski (rozdz. 30 – 32),
Joanna Sugiero (wprowadzenie, rozdz. 1 – 5, 16 – 24)
ISBN: 83-7361-725-6

Tytuł oryginału: [HTML, XHTML, and CSS Bible](#)

Format: B5, stron: 816



Poznaj tajniki projektowania nowoczesnych stron WWW

- Napisz kod strony zgodny z najnowszymi standardami ogłaszanymi przez W3C
- Wykorzystaj na stronach technologie skryptowe, Dynamiczny HTML oraz mechanizmy języka XML
- Zastosuj nowoczesne narzędzia do tworzenia i testowania witryny

Sieć WWW, od swojego powstania do czasów obecnych, bardzo zmieniła swoje oblicze. Ilość włączonych w nią komputerów powiększa się w ogromnym tempie. Zmieniają się możliwości przeglądarek i sposoby przesyłania informacji. Stale ewoluuje również jeden z podstawowych „budulców” sieci WWW – język HTML. Jego możliwości muszą nadążać za coraz większymi oczekiwaniami użytkowników sieci i funkcjami oferowanymi przez urządzenia wykorzystywane do jej przeglądania. Organizacja koordynująca jego rozwój – konsorcjum W3C – co pewien czas ogłasza nowy standard języka. Ewolucja HTML-a mająca na celu uzyskanie możliwości wyświetlania witryn WWW nie tylko na monitorach komputerów, ale także na wyświetlaczach telefonów komórkowych i komputerów przenośnych oraz edycji treści strony niezależnie od jej wyglądu pociągnęła za sobą powstanie standardu CSS. CSS (Kaskadowe Arkusze Stylów) to technologia umożliwiająca dowolne formatowanie stron WWW oraz szybkie modyfikacje ich kolorystyki i układu elementów.

Książka „HTML, XHTML i CSS. Biblia” to dogłębny opis najnowszych standardów związanych z tworzeniem stron WWW – obowiązującej obecnie specyfikacji języka HTML, technologii CSS oraz najnowszego produktu W3C – języka XHTML. Przedstawia nowoczesne metodologie tworzenia i testowania witryn WWW oraz ich aktualizowania i modyfikowania. Wprowadza także w tematykę technologii skryptowych oraz języka XML. Jeśli chcesz zaistnieć w internecie – stwórz własną witrynę WWW. Dzięki wiadomościom zawartym w tej książce nie będzie to stanowiło dla Ciebie problemu.



Spis treści

O Autorach	23
Wprowadzenie	25
Część I Omówienie języka HTML i XHTML	29
Rozdział 1. Wprowadzenie do sieci WWW i języka HTML	31
Czym jest World Wide Web?	31
Jak działa sieć WWW?	32
Czym jest hipertekst?	33
Gdzie mieści się HTML?	33
Wynalezienie HTML-a	33
Krótka historia języka HTML	35
Kto zatem ustanawia reguły?	37
Czym jest CSS?	39
Koszmar utrzymania	39
Wstęp do CSS	42
Co oznacza słowo „kaskadowe”?	42
Czym jest XHTML?	43
Tworzenie dokumentu HTML	44
Pisanie w języku HTML	44
Nadawanie plikom rozszerzenia kojarzącego się z siecią WWW	44
Formatowanie tekstu	45
Nadawanie dokumentowi struktury	45
Czy nie potrzebuję serwera WWW?	46
Podsumowanie	46
Rozdział 2. Z czego składa się strona WWW?	47
Definiowanie typu dokumentu	48
Ogólna struktura — znaczniki HTML, Head oraz Body	48
Znacznik <html>	48
Znacznik <head>	48
Style	50
Elementy blokowe — znaczniki dla akapitów	52
Sformatowane akapity	52
Nagłówki	53
Wydzielony blok tekstu	54
Listy	55
Zachowanie pierwotnego formatowania tekstu	57
Grupowanie elementów	57

Elementy wstawiane — znaczniki formatujące znaki.....	58
Podstawowe znaczniki wstawiane.....	58
Elementy span	59
Znaki specjalne (encje).....	59
Elementy organizacyjne.....	60
Tabele.....	61
Formularze	62
Łączy do innych stron.....	64
Obrazy	65
Komentarze.....	66
Skrypty	66
Wszystko razem.....	67
Podsumowanie.....	68
Rozdział 3. Rozpoczęcie pracy nad stroną WWW.....	69
Podstawowe zasady dotyczące kodu HTML	69
Używanie swobodnych białych znaków.....	70
Używanie poprawnej formy języka HTML.....	70
Wstawianie w kodzie komentarzy.....	71
Tworzenie podstawowej struktury.....	72
Deklarowanie typu dokumentu.....	72
Definiowanie tytułu dokumentu	73
Dostarczanie informacji wyszukiwarkom.....	73
Definiowanie domyślnej ścieżki	74
Definiowanie automatycznego odświeżania i przekierowania.....	75
Kolor tła strony i obrazy w tle	76
Definiowanie koloru tła.....	76
Definiowanie obrazu w tle dokumentu.....	77
Podsumowanie.....	78
Część II Podstawy tworzenia stron w języku HTML i XHTML.....	79
Rozdział 4. Wiersze, podziały wierszy i akapity.....	81
Podziały wierszy.....	81
Akapity.....	83
Ręczne wstawianie podziałów wierszy	85
Twarde spacje.....	87
Łączniki opcjonalne.....	87
Zachowanie formatowania — element <pre>.....	88
Wcięcia.....	90
Nagłówki	92
Linie poziome.....	93
Grupowanie za pomocą elementu <div>	94
Podsumowanie.....	97
Rozdział 5. Listy	99
Omówienie list.....	99
Listy uporządkowane (numerowane).....	100
Listy nieuporządkowane (punktowane).....	105
Listy definicji	108
Listy zagnieżdżone	110
Podsumowanie.....	111

Rozdział 6. Rysunki	113
Formaty graficzne stosowane w dokumentach WWW	113
Kompresja obrazu	114
Opcje kompresji	114
Głębia kolorów	116
Zwiększanie szybkości pobierania stron	117
Przygotowanie plików graficznych.....	118
Najważniejsze funkcje.....	119
Darmowe oprogramowanie alternatywne.....	120
Progresywne obrazy JPEG i rysunki GIF z przeplotem	120
Wstawianie rysunków.....	122
Rozmieszczanie rysunków.....	123
Opis wyświetlany w przeglądarkach tekstowych	125
Wymiary i skalowanie rysunków	126
Obramowanie rysunków.....	128
Mapy obrazu.....	128
Definiowanie mapy obrazu	130
Definiowanie obszarów aktywnych.....	130
Połączenie poszczególnych rozwiązań	132
Animacje	133
Podsumowanie.....	134
Rozdział 7. Łączy	135
Czym są łączy?	136
Łączy do stron WWW	137
Łączy względne i bezwzględne.....	138
Docelowe okna łączy.....	140
Tekst podpowiedzi łączy.....	141
Skróty klawiaturowe i kolejność uaktywniania łączy	142
Skróty klawiaturowe.....	142
Kolejność uaktywniania łączy	142
Tworzenie kotwic	143
Dobór kolorów łączy	144
Parametry dokumentu docelowego.....	145
Znacznik <link>	147
Podsumowanie.....	147
Rozdział 8. Tekst	149
Metody formatowania tekstu	149
Znacznik 	150
Akcentowanie i inne znaczniki dotyczące tekstu.....	150
Formatowanie tekstu przy użyciu CSS.....	151
Pogrubienie i kursywa	152
Czcionka o stałej szerokości.....	153
Indeksy — górny i dolny	153
Skróty	154
Oznaczanie zmian w dokumencie — wstawianie i usuwanie tekstu.....	154
Grupowanie elementów tekstowych znacznikiem 	154
Podsumowanie.....	155

Rozdział 9. Znaki specjalne	157
Kodowanie znaków	158
Znaki specjalne.....	158
Znaki spacji i myślników.....	159
Symbol praw autorskich i znaku towarowego	160
Symbole walut.....	161
Rzeczywiste znaki cudzośliwu	161
Strzałki	162
Znaki zawierające symbole akcentu	162
Litery alfabetu greckiego i symbole matematyczne.....	165
Inne użyteczne symbole.....	168
Podsumowanie.....	170
Rozdział 10. Tabele	171
Części składowe tabeli HTML	171
Szerokość i wyrównanie tabeli	173
Odstępy i otaczanie komórek.....	176
Obramowanie i krawędzie	177
Obramowanie tabeli	177
Krawędzie tabeli.....	178
Wiersze.....	179
Komórki.....	180
Podpis tabeli	181
Grupowanie wierszy — nagłówek, treść i stopka tabeli	183
Kolor tła.....	185
Komórki rozciągające się na kilka wierszy lub kolumn	186
Grupowanie kolumn	189
Podsumowanie.....	190
Rozdział 11. Rozmieszczanie elementów strony za pomocą tabel.....	191
Podstawy formatowania z wykorzystaniem tabel	191
Przykłady stron dostępnych w internecie.....	194
Strony pływające	197
Niestandardowe połączenie grafiki i tekstu	199
Menu nawigacyjne i bloki dokumentów.....	202
Dokumenty wielokolumnowe.....	203
Podsumowanie.....	204
Rozdział 12. Ramki	205
Omówienie ramek.....	205
Zbiory ramek i zawartość ramek	208
Tworzenie zbioru ramek.....	208
Marginesy, obramowania i paski przewijania ramek.....	210
Modyfikacja rozmiaru ramek	211
Odsyłacze do ramek.....	213
Zagnieżdżone zbiory ramek.....	215
Ramki pływające	217
Podsumowanie.....	220

Rozdział 13. Formularze	221
Podstawowe informacje o formularzach.....	222
Wstawianie formularzy.....	224
Metoda GET.....	225
Metoda POST.....	225
Dodatkowe atrybuty znacznika <form>	226
Etykiety pól	226
Pola tekstowe.....	227
Pola haseł.....	227
Przyciski opcji	227
Pola wyboru.....	228
Listy wyboru.....	229
Obszary tekstowe.....	231
Pola ukryte.....	232
Przyciski	232
Rysunki.....	233
Pola plików	233
Przyciski przesyłania danych i czyszczenia pól.....	234
Kolejność uaktywniania kontrolek i skróty klawiaturowe	235
Uniemożliwienie wprowadzania zmian	236
Grupy pól i ich opis	237
Skrypty formularzy i usługi skryptowe.....	238
Pobranie programu obsługi formularza	239
Wykorzystanie usług skryptowych.....	239
Podsumowanie.....	240
Rozdział 14. Obiekty multimedialne	241
Wykorzystanie obiektów multimedialnych	241
Rodzaje obiektów multimedialnych	243
Umieszczanie obiektów multimedialnych na stronie WWW	243
Multimedialne pluginy i odtwarzacze.....	247
Flash	247
RealOne.....	247
Windows Media Player	248
QuickTime.....	249
Animacje	249
Tworzenie animowanych plików GIF	249
Zachowanie niewielkich rozmiarów plików.....	252
Tworzenie plików Flash.....	252
Wideoklipy	253
Pliki dźwiękowe	254
Pokazy slajdów	255
Eksport prezentacji przygotowanej w programie PowerPoint.....	255
Eksportowanie prezentacji przygotowanej w środowisku OpenOffice	261
Język SMIL	263
Podsumowanie.....	267

Rozdział 15. Skrypty	269
Skrypty serwerowe i skrypty klienckie.....	269
Skrypty klienckie.....	269
Skrypty serwerowe.....	270
Określanie domyślnego języka skryptowego.....	270
Dołączanie skryptów	271
Wywoływanie zewnętrznych skryptów	272
Wywoływanie skryptów za pomocą zdarzeń.....	272
Ukrywanie skryptów przed starszymi przeglądarkami	276
Podsumowanie.....	276
 Część III Kontrolowanie prezentacji za pomocą CSS	277
Rozdział 16. Wprowadzenie do kaskadowych arkuszy stylów	279
Co to jest CSS?.....	279
Reguły stylów.....	280
Określanie reguł stylu	281
Korzystanie z elementu <style>	281
Zewnętrzne arkusze stylów	281
Definicje stylów w poszczególnych znacznikach.....	282
Omówienie kaskadowych właściwości arkuszy stylów.....	283
Model formatowania obiektów w CSS	284
Rozmiary pojemnika	284
Odstęp	285
Ramka	286
Marginesy.....	287
1., 2. i 3. poziom CSS.....	287
Podsumowanie.....	287
 Rozdział 17. Tworzenie reguł stylów.....	289
Omówienie selektorów	289
Dopasowywanie elementów według nazwy	290
Korzystanie z selektora uniwersalnego	290
Dopasowywanie elementów według klasy	291
Dopasowywanie elementów przy użyciu identyfikatora	291
Dopasowywanie elementów, które zawierają określony atrybut.....	291
Korzystanie z elementów dzieci, potomków oraz elementów przystających	292
Omówienie dziedziczenia.....	294
Pseudoklasy	295
Definiowanie stylów łączy	295
Pseudoklasa :first-child	296
Pseudoklasa :lang	296
Pseudoelementy.....	297
Stosowanie stylów dla pierwszego wiersza tekstu w elemencie.....	297
Stosowanie stylów dla pierwszej litery elementu	298
Definiowanie przed danym tekstem i po nim	299
Wyrażenia skrótowe	300
Metryki wartości właściwości	302
Podsumowanie.....	304

Rozdział 18. Czcionki	305
Podstawy typografii w sieci WWW	305
Zła metoda opisywania czcionek	306
Dobra metoda opisywania czcionek	308
Praca z atrybutami stylów czcionek	310
Określanie rodzajów czcionek przy użyciu CSS	310
Omówienie zagadnienia rodziny czcionek	311
Omówienie czcionek oraz ich dostępności	312
Praca ze stylami czcionek	312
Określanie rozmiarów czcionek	313
Używanie (lub brak użycia) wariantów czcionek	314
Pogrubianie czcionek poprzez zmianę ich wagi	314
Poszerzanie i zwężanie czcionek za pomocą właściwości font-stretch	315
Wysokość wiersza oraz odstępy między wierszami	315
Automatyczne ładowanie czcionek	315
Standardy oraz opcje dynamicznych czcionek	316
Licencjonowanie	316
Używać wbudowanych czcionek czy raczej arkuszy stylów?	317
Jak dodawać do strony WWW czcionki, które będzie można załadować?	317
Składnia	317
Podsumowanie	318
Rozdział 19. Formatowanie tekstu	319
Wyrównywanie tekstu	319
Kontrolowanie wyrównania poziomego	319
Kontrolowanie wyrównania pionowego	321
Tworzenie wcięcia w tekście	324
Kontrolowanie znaków niewidocznych w tekście	324
Obiekty przestawne	325
Właściwość white-space	327
Kontrolowanie odstępów między literami i słowami	327
Definiowanie wielkich liter	329
Dekorowanie tekstu	330
Formatowanie list	331
Ogólne informacje na temat list	331
CSS — każdy element pasuje	331
Właściwość list-style-type	332
Pozycjonowanie markerów	334
Punktory rysunkowe	335
Tekst generowany automatycznie	335
Definiowanie znaków cudzysłowu	335
Automatyczne numerowanie elementów	336
Podsumowanie	340
Rozdział 20. Ramki, odstępy i marginesy	341
Omówienie modelu formatowania pojemnika	341
Definiowanie marginesów elementu	343
Dodawanie odstępu do elementu	345
Dodawanie ramki	345
Styl ramki	345
Kolor ramki	346
Szerokość ramki	347

Największy skrót: właściwość border	348
Dodatkowe właściwości ramki.....	348
Wykorzystywanie dynamicznego obramowania.....	348
Podsumowanie.....	349
Rozdział 21. Kolory i tło.....	351
Kolor pierwszoplanowy.....	351
Kolor tła.....	352
Rozmiar tła elementu.....	354
Obrazy w tle	354
Powtarzanie i przewijanie obrazów tła	355
Określanie pozycji obrazów tła.....	358
Podsumowanie.....	360
Rozdział 22. Tabele	361
Definiowanie stylów tabeli	361
Kontrolowanie atrybutów tabeli	362
Ramki tabeli	362
Odstępy w ramce tabeli	364
Pojedyncze obramowanie.....	365
Ramki wokół pustych komórek.....	365
Układ graficzny tabeli	366
Wyrównywanie i pozycjonowanie podpisów	367
Podsumowanie.....	368
Rozdział 23. Pozycjonowanie elementów	369
Omówienie pozycjonowania elementów	369
Pozycjonowanie statyczne.....	369
Pozycjonowanie względne	370
Pozycjonowanie bezwzględne.....	371
Pozycjonowanie stałe	372
Określanie pozycji elementu.....	372
Elementy dryfujące do lewej lub prawej strony.....	375
Definiowanie szerokości i wysokości elementu	376
Dokładne definiowanie rozmiarów.....	377
Definiowanie rozmiaru maksymalnego oraz minimalnego	377
Kontrolowanie przepełnienia elementu	378
Układanie elementów na stosie.....	379
Kontrolowanie widoczności elementu.....	383
Podsumowanie.....	383
Rozdział 24. Definiowanie stron do druku.....	385
Model formatowania pojemnika strony	386
Definiowanie rozmiaru strony	387
Definiowanie rozmiaru strony za pomocą właściwości size.....	388
Definiowanie marginesów za pomocą właściwości margin	388
Kontrolowanie podziałów stron.....	389
Właściwości page-break	390
Korzystanie z właściwości page-break-before oraz page-break-after.....	390
Właściwość page-break-inside	394
Zarządzanie wdowami i sierotami	395
Przygotowanie dokumentu do drukowania dwustronnego	395
Podsumowanie.....	396

Część IV Zaawansowane techniki tworzenia stron 397

Rozdział 25. JavaScript..... 399

Podstawy języka JavaScript.....	399
Pisanie programów w JavaScriptcie	401
Typy danych i zmienne	401
Obliczenia i operatory	402
Operacje na łańcuchach.....	402
Konstrukcje sterujące	403
Funkcje.....	405
Użycie obiektów.....	407
Obsługa zdarzeń w języku JavaScript.....	408
Stosowanie języka JavaScript w dokumentach HTML.....	408
Dodawanie skryptów za pomocą elementu script.....	408
Uruchamianie programów w JavaScriptcie	410
Praktyczne przykłady	411
Identyfikacja przeglądarki i zgodność z przeglądarkami.....	411
Data ostatniej modyfikacji.....	413
Aktywne przyciski.....	413
Umieszczanie grafiki w pamięci podręcznej	415
Weryfikacja formularzy.....	416
Określanie wielkości i położenia okna przeglądarki	419
Ramki	422
Wykorzystanie cookies.....	422
Podsumowanie.....	424

Rozdział 26. Dynamiczny HTML 425

Czemu służy dynamiczny HTML?	425
Jak działa DHTML?	426
DHTML a obiektowy model dokumentu.....	426
Stosowanie procedur obsługi zdarzeń	427
Chodzi o obiekty	428
Zagadnienia zgodności z różnymi przeglądarkami.....	428
Wykrywanie używanej przeglądarki	429
Wykrywanie obiektów	430
Przykłady rozwiązań DHTML.....	431
Wskaźnik położenia strony.....	431
Efekty podmieniania	436
Rozwijane menu.....	441
Podsumowanie.....	443

Rozdział 27. Dynamiczny HTML i CSS 445

DHTML i właściwości CSS	448
Określanie właściwości CSS przy użyciu języka JavaScript.....	448
Stosowanie zachowań w celu tworzenia efektów DHTML.....	451
Filtry Internet Explorera	452
Filtry.....	453
Elementy HTML pozwalające na stosowanie filtrów.....	453
Filtry wizualne	454
Podsumowanie.....	462

Rozdział 28. Wprowadzenie do zagadnień skryptów wykonywanych po stronie serwera	465
Sposób działania serwerów WWW	466
Najpopularniejsze serwery WWW	467
Apache	468
IIS	468
Zapotrzebowanie na skrypty działające po stronie serwera	469
Języki służące do tworzenia skryptów działających po stronie serwera	470
CGI	470
ASP, .NET oraz technologie firmy Microsoft	471
PHP	472
ColdFusion	474
Podsumowanie	474
Rozdział 29. Wprowadzenie do tworzenia serwisów opartych na bazach danych	475
Po co używać baz danych?	475
Jak działa integracja z bazą danych?	476
Metody publikowania w oparciu o bazę danych	477
Generowanie jednorazowe	477
Generowanie na żądanie	477
Studium przypadku — magazyn informacyjny	478
Metoda „ręczna”	478
Metoda z użyciem bazy danych	479
Uwierzytelnianie i bezpieczeństwo	486
Podsumowanie	488
Rozdział 30. Blog	489
Pojęcie blogu	489
Oprogramowanie i serwisy wspomagające blogowanie	491
Oprogramowanie firmy Userland	491
Oprogramowanie firmy Movable Type	491
Blosxom	492
Umieszczanie artykułów w blogu	492
Obsługa komentarzy i opinii	493
Stałe łącza do artykułów	493
Łącza zwrotne	494
Rozprowadzanie treści za pomocą RSS	495
Składnia RSS	495
Publikowanie elementu RSS	496
Przyciąganie czytelników	497
Podsumowanie	498
Rozdział 31. Wprowadzenie do języka XML	499
Zapotrzebowanie na XML	500
Związki pomiędzy XML, SGML i HTML	501
Jak działa XML?	502
Rozpocznij od parsera XML	503
Rozpocznij od prologu	503
O kodowaniu	504
Poprawna struktura dokumentu XML	505

Definicje typu dokumentu (DTD).....	508
Użycie elementów w definicji typu dokumentu	510
Definiowanie atrybutów w DTD	512
Definiowanie i użycie stałych tekstowych w definicji DTD	513
Użycie danych typu PCDATA i CDATA w definicji typu	513
Schematy XML	513
Użycie schematów	514
XML w internecie.....	516
Podsumowanie.....	516

**Rozdział 32. Przetwarzanie dokumentów XML
oraz implementacje standardu..... 517**

Przetwarzanie kodu XML.....	517
Język XPath.....	518
Arkusze stylów kodu XML — XSLT	523
Implementacje XML.....	536
XHTML	536
Usługi Web Services (SOAP, UDDI itd.)	538
XUL	538
WML.....	538
Podsumowanie.....	539

**Część V Testowanie i publikowanie witryny internetowej
oraz zarządzanie nią..... 541**

Rozdział 33. Testowanie i weryfikacja struktury dokumentów..... 543

Testowanie w różnych przeglądarkach	543
Testowanie w różnych rozdzielczościach ekranu	544
Weryfikacja kodu źródłowego	544
Określanie poprawnego typu dokumentu	545
Narzędzia do weryfikacji poprawności kodu	545
Jak weryfikować dokumenty?	545
Podsumowanie.....	547

Rozdział 34. Programy do projektowania stron internetowych..... 549

Edytory tekstowe	550
Proste edytory tekstowe.....	550
Inteligentne edytory tekstowe.....	550
Edytory HTML.....	551
Edytory HTML pracujące w trybie WYSIWYG	551
FrontPage firmy Microsoft	552
NetObjects Fusion	554
Dreamweaver firmy Macromedia.....	556
Inne narzędzia.....	557
Programy graficzne	558
Flash firmy Macromedia	559
Podsumowanie.....	560

Rozdział 35. Wybór dostawcy usług internetowych	561
Dostawcy usług internetowych	562
Usługi WWW w ramach opłat za dostęp do internetu	562
Hosting współdzielony	562
Hosting dedykowany	563
Kolokacja	564
Oszacowanie kosztów	564
Usługi i pomoc techniczna	564
Przepustowość i skalowalność	566
Umowy długoterminowe	567
Nazwy domen	567
Podsumowanie	568
Rozdział 36. Umieszczanie plików witryny na serwerze za pomocą protokołu FTP	569
Wprowadzenie do FTP	569
Programy-klienty FTP	570
Popularne programy-klienty FTP	573
Podstawowe zasady organizacji plików w obrębie witryny WWW	574
Podsumowanie	575
Rozdział 37. Publikowanie strony i zwiększanie jej oglądalności	577
Rozpowszechnianie adresu witryny	578
Wymiana adresów	578
Grupy dyskusyjne	578
Zgłaszanie witryny do wyszukiwarek internetowych	579
Pozycjonowanie i optymalizacja strony pod kątem wyszukiwarek	580
Zdobywanie łączy do Twojej witryny na innych stronach WWW	580
Zachęcanie do umieszczenia witryny w Ulubionych	581
Częsta aktualizacja strony	581
Dobór właściwych słów kluczowych i zwiększanie szansy na odnalezienie Twojej witryny	581
Jak zachęcić użytkowników do przedłużania pobytu na stronie WWW?	584
Udostępnienie zasobów	584
Fora i pogawędki internetowe	584
Czego na pewno robić nie należy?	585
Niechciana poczta	585
Wielokrotne zgłaszanie jednego adresu URL	585
Spam w grupach dyskusyjnych	585
Spam na forach dyskusyjnych i w pogawędkach internetowych	585
Podsumowanie	586
Rozdział 38. Zarządzanie witryną internetową	587
Szacowanie oglądalności witryny za pomocą dzienników serwera	587
Sprawdzanie ruchu na serwerze Apache	587
Sprawdzanie ruchu na serwerze IIS	590
Wybór właściwego programu do analizowania dzienników serwera	591
Weryfikacja pod kątem niedziałających łączy	592
Program sprawdzający W3C Link Checker	592
Programy sprawdzające w aplikacjach do tworzenia stron WWW	594
Inne, niezależne narzędzia	595
Sprawdzanie dzienników działania serwera WWW	595

Reakcja na opinie użytkowników	595
Tworzenie kopii zapasowych danych	596
Podsumowanie.....	596

Część VI Podstawowe zasady projektowania i tworzenia stron WWW 597

Rozdział 39. Proces tworzenia stron WWW..... 599

Zagadnienia związane z tworzeniem dużych serwisów internetowych	600
Podstawy zarządzania projektami.....	600
Konieczność zaprojektowania architektury danych.....	601
Ogólny opis procesu tworzenia stron WWW.....	602
Określenie celów	603
Określenie docelowej grupy odbiorców	603
Przeprowadzenie analizy rynku i konkurencji.....	603
Analiza funkcjonalności.....	603
Zaprojektowanie struktury witryny	604
Określenie zawartości projektu	606
Opracowanie szaty graficznej.....	607
Wykonanie projektu	607
Testowanie i ocena działania witryny.....	609
Promocja	610
Śledzenie liczby odwiedzin i wydajności działania witryny.....	611
Obsługa strony WWW	612
Podsumowanie.....	612

Rozdział 40. Projektowanie struktury strony 615

Podstawowe zasady dotyczące analizy odbiorców strony WWW	616
Systematyzowanie informacji.....	616
Dzielenie dokumentów na mniejsze części.....	617
W jaki sposób internauci czytają strony WWW?	617
Stylistyka i formatowanie tekstu.....	618
Opracowanie treści znaczników meta: tytułów i nagłówek oraz sloganów	619
Tytuły	619
Nagłówki	619
Slogany.....	620
Cechy doskonałej publikacji internetowej	620
Bądź zwięzły	621
Twórz łatwe w przeglądaniu strony.....	621
Zachowaj wiarygodność.....	621
Zachowaj obiektywność.....	622
Skup się na określonej tematyce i nie bądź wylewny.....	622
Od ogółu do szczegółu	622
Umieszczaj podsumowania	623
Jak pisać na potrzeby internetu?	623
Zastosowanie list wypunktowanych.....	623
Rozważne wykorzystanie słów kluczowych.....	623
Fachowa nomenklatura i slang marketingowy	624
Podstawowe składniki strony internetowej.....	624
Składamy wszystko w jedną całość	625
Podsumowanie.....	628

Rozdział 41. Zasady ergonomii i ułatwień dostępu.....629

Metody badania ergonomii	629
W jaki sposób użytkownicy korzystają z serwisów WWW?	630
Podstawowe zasady ergonomii stron WWW	631
Ergonomia	631
Ogłoszenia reklamowe	631
Animacje, multimedia i aplety.....	632
Kolory i łącza	632
Zachowanie spójności	633
Treść	633
Rozwijane menu	633
Kroje i rozmiary czcionek	633
Ramki	634
Elementy graficzne.....	634
Nagłówki	635
Przewijanie poziome	635
JavaScript	636
Czytelność	636
Wyszukiwanie	637
Mapa serwisu	637
Długość adresu URL	637
Slogany	638
Strona kodowa Windows 1252.....	638
Dlaczego warto projektować ułatwienia dostępu?	638
Zalecenia dotyczące dostępności	639
Americans with Disabilities Act.....	639
Rozporządzenia międzynarodowe.....	640
Inicjatywa Web Content Accessibility (W3C).....	640
Uwzględnienie wad wzroku	640
Uwzględnienie wad słuchu.....	641
Wsparcie dla użytkowników o ograniczonej sprawności motorycznej.....	641
Optymalizacja witryny pod kątem użytkowników mających problemy z przyswajaniem wiedzy i koncentracją	642
Przydatne narzędzia.....	642
Formularze i PDF	642
Testowanie dostępności za pomocą narzędzi sprawdzających	644
Podsumowanie.....	644

Rozdział 42. Projektowanie stron WWW dla obcokrajowców.....645

Podstawowe zasady lokalizacji serwisów	645
Wprowadzenie do projektowania serwisów wielojęzycznych	646
Tłumaczenie strony internetowej.....	646
Standard Unicode	647
Basic Latin (podstawowy łaciński, U+0000 – U+007F)	650
ISO-8859-1	650
Dodatek Latin-1 (U+00C0 – U+00FF)	654
Zestaw łaciński rozszerzony A (U+0100 – U+017F)	655
Zestaw łaciński rozszerzony B i łaciński rozszerzony dodatkowy	656
Polska norma kodowania znaków	656
Projektowanie serwisów wielojęzycznych.....	656
Podsumowanie.....	657

Rozdział 43. Bezpieczeństwo	659
Czyhające niebezpieczeństwa.....	659
Kradzież poufnych informacji.....	659
Chuligaństwo i wandalizm	660
Ataki typu DoS (ang. Denial of Service).....	660
Utrata danych	660
Straty finansowe.....	661
Utrata zaufania i reputacji	661
Spory i kwestie prawne	661
Zagadnienia związane z zabezpieczeniem serwera WWW.....	662
Uprawnienia dostępu do plików	662
Niewykorzystywane, lecz otwarte porty.....	662
Skrypty CGI	663
Przepelnienie bufora.....	663
Systemy, które uległy atakowi	664
Przegląd dostępnych metod zabezpieczeń	664
Spójna polityka bezpieczeństwa.....	665
Zasoby internetowe dotyczące zabezpieczeń	665
Blokada dostępu dla wyszukiwarek.....	666
Serwery o podwyższonym bezpieczeństwie.....	666
Podsumowanie.....	667
Rozdział 44. Poufność.....	669
Poufność danych.....	669
Regulacje prawne w USA.....	670
Ustawa „The Children’s Online Privacy Protection Act” (COPPA)	670
Ustawa „Electronic Communications Privacy Act” (ECPA).....	671
Ustawa „Patriot Act” z roku 2001	672
Ustawa „Fair Credit Reporting Act”.....	672
Regulacje prawne w Unii Europejskiej.....	672
Ochrona danych osobowych w Polsce.....	673
Dobrowolne rozwiązania na rzecz prywatności.....	676
Platform for Privacy Preferences Project (P3P)	676
Certyfikaty i programy legalizacyjne	679
Wzorcowe rozwiązania w zakresie prywatności.....	680
Podsumowanie.....	681
Dodatki	683
Dodatek A Znaczniki języka HTML 4.01	685
Dodatek B Właściwości elementów CSS	753
skorowidz	785

Rozdział 2.

Z czego składa się strona WWW?

W tym rozdziale:

- ♦ Definiowanie typu dokumentu
- ♦ Ogólna struktura — znaczniki HTML, Head oraz Body
- ♦ Style
- ♦ Elementy blokowe — znaczniki dla akapitów
- ♦ Elementy wstawiane — znaczniki formatujące znaki
- ♦ Znaki specjalne (encje)
- ♦ Elementy organizacyjne
- ♦ Łączy do innych stron
- ♦ Obrazy
- ♦ Komentarze
- ♦ Skrypty
- ♦ Wszystko razem

Język HTML przeszedł długą drogę od swoich skromnych początków. Jednak mimo tego, że obecne możliwości HTML-a (a także jego pochodnych) znacznie wykraczają poza statyczne przedstawianie dokumentów tekstowych, podstawowa struktura dokumentu HTML nie zmieniła się.

Omawianie elementów HTML-a rozpoczniemy krótkim podsumowaniem, z którego dowiesz się, z jakich elementów składa się dokument HTML, do czego służy każdy z nich i jak wpływa na inne elementy. W kolejnych rozdziałach znajdziesz szczegółowy opis każdego elementu oraz związanych z nim technologii.

Definiowanie typu dokumentu

Jednym z najczęściej pomijanych atrybutów dokumentu HTML jest definicja typu dokumentu (ang. *Document Type Definition*, DTD). Definicja ta poprzedza wszystkie umieszczone w dokumencie znaczniki i informuje przeglądarkę o formacie danego dokumentu — jakich znaczników można się spodziewać, jakie metody będzie obsługiwać i tak dalej.

Znacznik `<!DOCTYPE>` służy do określenia istniejącej definicji typu dokumentu. DTD zawiera wszystkie elementy, definicje, zdarzenia i inne rzeczy skojarzone z typem dokumentu. Znacznik DOCTYPE wygląda mniej więcej tak:

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

Znacznik ten zawiera następujące informacje:

- ♦ Poziom najwyższego znacznika w dokumencie to HTML (`html`).
- ♦ Dokument jest zgodny ze standardami formalnego identyfikatora publicznego (ang. *formal public identifier*, w skrócie FPI) „W3C HTML 4.01 Strict English” (`PUBLIC "-//W3C//DTD HTML 4.01//EN"`).
- ♦ Cała definicja DTD jest dostępna pod adresem URL <http://www.w3.org/TR/html4/strict.dtd>.

Ogólna struktura — znaczniki HTML, Head oraz Body

Wszystkie dokumenty HTML na poziomie dokumentu mają trzy identyczne znaczniki. Znaczniki te, `<html>`, `<head>` oraz `<body>`, określają poszczególne części dokumentu HTML.

Znacznik `<html>`

Znacznik `<html>` otacza cały dokument HTML. Informuje on przeglądarkę o tym, gdzie się rozpoczyna oraz kończy dokument.

```
<html>
... treść dokumentu...
</html>
```

W poprzednich wersjach HTML-a można było deklarować dodatkowe opcje wewnątrz znacznika `<html>`. Opcje te (zwłaszcza `lang` oraz `dir`) straciły jednak na znaczeniu z chwilą wprowadzenia HTML-a w wersji 4.0. Informacje kierunkowe oraz te, które dotyczą języka, są standardowo przechowywane w deklaracji typu dokumentu (`<!DOCTYPE>`).

Znacznik `<head>`

Znacznik `<head>` otacza nagłówek dokumentu HTML. Ta część dokumentu zawiera informacje dotyczące nagłówków umieszczonych w dokumencie. Jest w niej przechowywany tytuł dokumentu, metainformacje, a w większości przypadków także skrypty dokumentów. Typowy element `<head>` wygląda mniej więcej tak:

```

<head>
<link rel="stylesheet" type="text/css" href="/styles.css">
<title>Tytuł dokumentu</title>
<meta name="description" content="Przykładowa strona">
<meta name="keywords" content="przykład, nagłówek, strona">
<script language="JavaScript">
function NewWindow(url){
    fin=window.open(url, "", "width=800,height=600,scrollbars=yes,resizable=yes");
}
</script>
</head>

```



Większość znaczników poziomu <head> omówiono szczegółowo w rozdziale 3. Pisanie skryptów JavaScript zostało dokładnie opisane w rozdziałach 15. oraz 28.

Większość informacji znajdujących się w elemencie <head> nie będzie widoczna na stronie podczas jej wyświetlania przez przeglądarkę. Element <title> określa tekst, który przeglądarka wyświetli jako tytuł strony — w systemie Windows tytuł dokumentu pojawia się w pasku tytułu przeglądarki, tak jak widać to na rysunku 2.1.

Rysunek 2.1.

W systemie Windows tytuł dokumentu jest wyświetlany na pasku tytułu przeglądarki



Główna, wizualna zawartość dokumentu HTML znajduje się wewnątrz znaczników <body>.

Zwróć uwagę na to, że w wersji 4.0 języka HTML większość atrybutów znacznika <body> została pominięta, ponieważ preferuje się definiowanie atrybutów w formie stylów. W poprzednich wersjach HTML-a można było określać mnóstwo opcji dotyczących tła, tekstu czy też kolorów łączy. Wciąż można stosować atrybuty znacznika <body>: onload oraz onunload oraz atrybuty globalne, takie jak style. Jednak inne atrybuty należy już definiować nie wewnątrz znacznika <body>, a w stylach, tak jak pokazano w następującym przykładzie:

```
<html>
<head>
<title>Tytuł dokumentu</title>
<style type="text/css">
body { background: black; color: white}
a:link { color: red }
a:visited { color: blue }
a:active { color: yellow }
</style>
</head>
<body>
...treść dokumentu...
</body>
</html>
```



Szczegółowe omówienie stylów znajdziesz w rozdziałach 16. oraz 24.

Style

Style są stosunkowo nowym elementem języka HTML, który zrewolucjonizował sposób kodowania i wyświetlania dokumentów HTML. To właśnie one nadają językowi charakter rozszerzalny (słowo „rozszerzalny” jest jednym z elementów skrótu XHTML). Autorzy stron WWW mogą tworzyć nowe style, które umożliwią im odpowiednie zaprezentowanie treści dokumentu, nie odchodząc jednocześnie od stałych formatów.

Style są niczym innym, jak atrybutami definiującymi sposób wyświetlania strony w przeglądarce, połączonymi dla uzyskania określonego efektu. Osoby, które miały już kontakt ze stylami podczas pracy na programach edytujących tekst, nie będą miały problemów ze zrozumieniem stylów HTML.



Style są z reguły omawiane w kontekście kaskadowania jako standard CSS (kaskadowych arkuszy stylów). Standard CSS definiuje metodę, która sprawia, że w tym samym dokumencie można zastosować kilka arkuszy stylów (czyli list definicji stylów). Jeżeli tego samego stylu dotyczy kilka definicji, każda kolejna nadpisuje poprzednią na zasadzie *kaskadowania*. Więcej informacji na temat stylów, arkuszy stylów oraz CSS znajdziesz w rozdziale 16.

Założmy na przykład, że chcesz zaznaczyć w dokumencie fragmenty tekstu, które mają zostać usunięte. Tekst taki powinien być przekreślony i wyświetlony czerwoną czcionką. Mógłbyś w tym celu otoczyć wybrane fragmenty znacznikami `<font>` oraz `<del>`. Jednak ta metoda ma dwie poważne wady:

- ♦ HTML 4 dąży do rezygnacji ze stosowania znacznika `<font>`, dlatego nie powinien go używać.
- ♦ Jeżeli później zmienisz zdanie na temat koloru lub innego sposobu oznaczenia tekstu, będziesz musiał odnaleźć każdy zestaw znaczników i wprowadzić w nim odpowiednie zmiany.

Zamiast tego możesz zdefiniować style dla tych elementów, które zawierają określone atrybuty tekstu. Oto fragment kodu HTML, który definiuje taki styl, a później wykorzystuje go do zaznaczenia wybranego tekstu w dokumencie:

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
<style>
.redline { color: red; text-decoration: line-through; }
</style>
</head>
<body>
<h1>Wcześniejszy projekt Deklaracji Niepodległości</h1>
<p>Ilekoć wskutek biegu wypadków koniecznym się staje dla jakiegoś narodu, by zerwał
więzy polityczne łączące go z innym narodem i zajął wśród potęg ziemskich oddzielne
i równorzędne stanowisko, do którego upoważniają go prawa natury i jej Bóg, to właściwy
respekt wymaga, aby naród ten podał powody, które zmusiły go do oderwania się.<span
class="redline">W tym dokumencie zadeklarowano te powody.</span></p>
</body>
</html>

```

Efekt zastosowania tego kodu przedstawiono na rysunku 2.2.

Rysunek 2.2.
Styl „redline” został
zastosowany
w tekście dzięki
użyciu znacznika



W wielu znacznikach HTML style mogą być stosowane bezpośrednio, przy użyciu atrybutu `style`. Aby na przykład zaznaczyć cały akapit czerwoną przekreśloną czcionką, możesz użyć następującego kodu:

```
<p style="color: red; text-decoration: line-through;"> przykładowy akapit </p>
```

Jednak jeśli będziesz używać stylów w ten sposób, pozbawisz się możliwości łatwego ich modyfikowania, a więc jednej z ich największych zalet.

Jeżeli będziesz chciał później zmienić atrybuty tekstu, wystarczy, że odpowiednio zmodyfikujesz definicje zamieszczone w części `<style>`, a zmiany zostaną wprowadzone w całym dokumencie. A co, jeśli jeden styl wykorzystuje kilka dokumentów? W takiej sytuacji będziesz musiał dokonać odpowiednich zmian w każdym dokumencie. Na szczęście implementacja stylów HTML umożliwia stosowanie zewnętrznych arkuszy stylów w wielu dokumentach — wówczas wystarczy tylko zmienić styl w zewnętrznym arkuszu stylów.

Oto kod, który definiuje zewnętrzny arkusz stylów *site-styles.css* dla bieżącego dokumentu HTML:

```

<html>
<head>
<LINK rel="stylesheet" href="site-styles.css"
type="text/css">
</head>
<body> ...

```

Treść dokumentu *site-styles.css* będą stanowiły definicje, które normalnie znalazłyby się między znacznikami `<style>`. Dla przykładu z czerwoną przekreśloną czcionką treść tego dokumentu byłaby następująca:

```
.redline { color: red; text-decoration: line-through; }
```



Istnieje dużo więcej atrybutów, które można zastosować w tekście, a także w innych obiektach, przy użyciu stylów. Więcej informacji na temat stylów znajdziesz w rozdziale 16.

Elementy blokowe — znaczniki dla akapitów

Tak jak w przypadku większości edytorów tekstu, HTML zawiera kilka znaczników, które określają, a więc formatują poszczególne akapity tekstu. Są to następujące znaczniki:

- ♦ `<p>` — sformatowane akapity;
- ♦ `<h1>` do `<h6>` — nagłówki;
- ♦ `<blockquote>` — wydzielony blok tekstu;
- ♦ `<pre>` — zachowanie pierwotnego formatowania tekstu;
- ♦ `<ul>`, `<ol>`, `<dl>` — listy nienumerowane, uporządkowane oraz listy definicji;
- ♦ `<center>` — tekst wyśrodkowany;
- ♦ `<div>` — grupowanie elementów.

Zastosowanie dowolnego elementu blokowego powoduje wstawienie podziału wiersza oraz wyraźnego odstępu po zamykającym znaczniku. Jako takie, elementy blokowe działają tylko wtedy, gdy użyjesz ich do sformatowania fragmentów tekstu mających charakter akapitu — nie mogą być stosowane jako style wstawiane.

Kolejne punkty zawierają szczegółowe omówienie każdego z wymienionych wyżej znaczników.



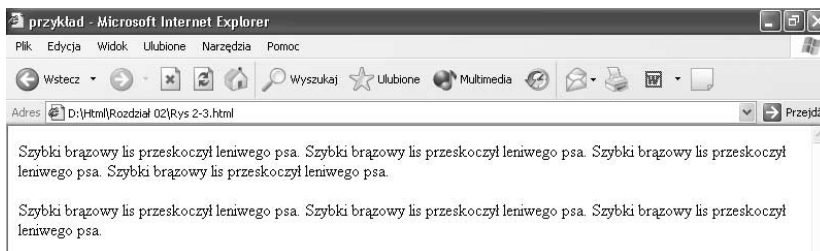
Więcej szczegółów na temat elementów blokowych oraz ich formatowania znajdziesz w rozdziale 4.

Sformatowane akapity

Znacznik akapitu (`<p>`) służy do określania całych akapitów w tekście. Oto przykładowy kod HTML oraz efekt jego zastosowania przedstawiony na rysunku 2.3:

```
<p> Szybki brązowy lis przeskoczył leniwego psa. Szybki brązowy lis przeskoczył leniwego psa. Szybki brązowy lis przeskoczył leniwego psa.</p>
<p> Szybki brązowy lis przeskoczył leniwego psa. Szybki brązowy lis przeskoczył leniwego psa. Szybki brązowy lis przeskoczył leniwego psa.</p>
```

Rysunek 2.3.
Znaczniki akapitu
dzielą tekst
na oddzielne akapity



Podobnie jak w przypadku większości znaczników, również w znaczniku `<p>` możesz definiować niektóre elementy formatowania, takie jak czcionka, wyrównanie, odstępy i tak dalej. Możesz na przykład wyśrodkować akapit, dodając do znacznika `<p>` atrybut `align`:

```
<p align="center"> Szybki brązowy lis przeskoczył leniwego psa. Szybki brązowy lis przeskoczył leniwego psa. Szybki brązowy lis przeskoczył leniwego psa.</p>
```

Jednak obecna tendencja zmierza ku rezygnacji z takiego formatowania na rzecz korzystania z arkuszy stylów. Następny przykład pokazuje, jak za pomocą arkuszy stylów osiągnąć taki sam efekt, jaki daje atrybut `align`:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
<style type="text/css">
p.center {text-align: center;}
</style>
</head>
<body>
<p class="center"> Szybki brązowy lis przeskoczył leniwego psa. Szybki brązowy lis przeskoczył leniwego psa. Szybki brązowy lis przeskoczył leniwego psa.</p>
</body>
</html>
```

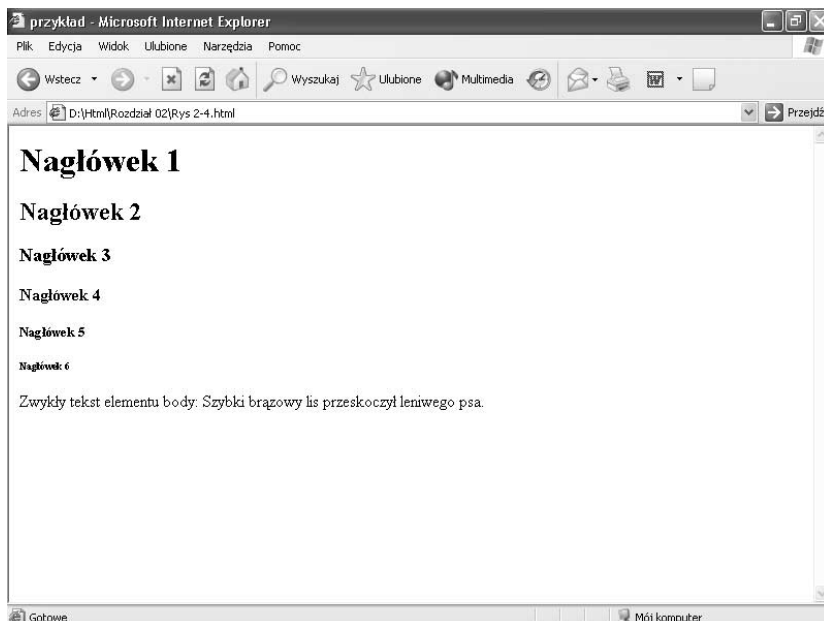
Użycie dowolnej z dwóch powyższych metod przynosi taki sam efekt: wyśrodkowanie akapitu w oknie przeglądarki.

Nagłówki

HTML obsługuje sześć poziomów nagłówków. Wykorzystywane przez nie style formatowania charakteryzują się dużą, najczęściej pogrubioną czcionką, która wyróżnia je spośród innych fragmentów tekstu. Oto przykładowy kod HTML oraz efekt jego zastosowania przedstawiony na rysunku 2.4:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<body>
<h1>Nagłówek 1</h1>
<h2>Nagłówek 2</h2>
<h3>Nagłówek 3</h3>
<h4>Nagłówek 4</h4>
<h5>Nagłówek 5</h5>
<h6>Nagłówek 6</h6>
<p>Zwykły tekst elementu body: Szybki brązowy lis przeskoczył leniwego psa.</p>
</body>
</html>
```

Rysunek 2.4.
HTML obsługuje sześć poziomów nagłówków



Sześć poziomów nagłówków rozpoczyna poziom 1., najwyższy i najważniejszy, po którym następują kolejne poziomy, aż do poziomu 6., najniższego i najmniej ważnego. Mimo że istnieje aż sześć predefiniowanych poziomów nagłówków, Ty prawdopodobnie będziesz używać w swoich dokumentach najwyżej trzech lub czterech. Nie ma żadnych ograniczeń w korzystaniu z nagłówków — możesz stosować wybrane poziomy niezależnie. Oznacza to, że nie musisz użyć nagłówków `<h1>` oraz `<h2>` po to, aby móc zastosować nagłówek `<h3>`. Pamiętaj, że za pomocą stylów możesz zmienić formatowanie narzucone przez każdy poziom nagłówków.



Tematyka wykorzystywania stylów została dokładniej omówiona w rozdziale 16.

Wydzielony blok tekstu

Znacznik `<blockquote>` służy do określenia wydzielonych bloków tekstu. Oto przykładowy kod, który definiuje pierwszy paragraf Deklaracji Niepodległości jako wydzielony blok:

```
<body>
<p>Deklarację Niepodległości rozpoczyna następujący akapit:</p>
<blockquote>
Ilekróć wskutek biegu wypadków koniecznym się staje dla jakiegoś narodu, by zerwał
więzy polityczne łączące go z innym narodem i zajął wśród potęg ziemskich oddzielne
i równorzędne stanowisko, do którego upoważniają go prawa natury i jej Bóg, to właściwy
respekt wymaga, aby naród ten podał powody, które zmusiły go do oderwania się.
</blockquote>
</body>
```

Znacznik `<blockquote>` powoduje wcięcie akapitu i wyróżnia go w ten sposób spośród innych fragmentów tekstu (zobacz rysunek 2.5).

Rysunek 2.5.

Znacznik
`<blockquote>`
 powoduje
 wcięcie akapitu

**Listy**

HTML umożliwia określenie trzech różnych rodzajów list:

- ♦ listy uporządkowane (zazwyczaj numerowane);
- ♦ listy nieuporządkowane (zazwyczaj punktowane);
- ♦ listy definicji (elementy listy ze zintegrowanymi definicjami).

Zarówno listy uporządkowane, jak i nieuporządkowane wykorzystują element `<li>` dla określenia poszczególnych elementów listy. Lista definicji posiada dwa znaczniki — jeden określa elementy listy (`<dt>`), a drugi definicje tych elementów (`<dd>`).

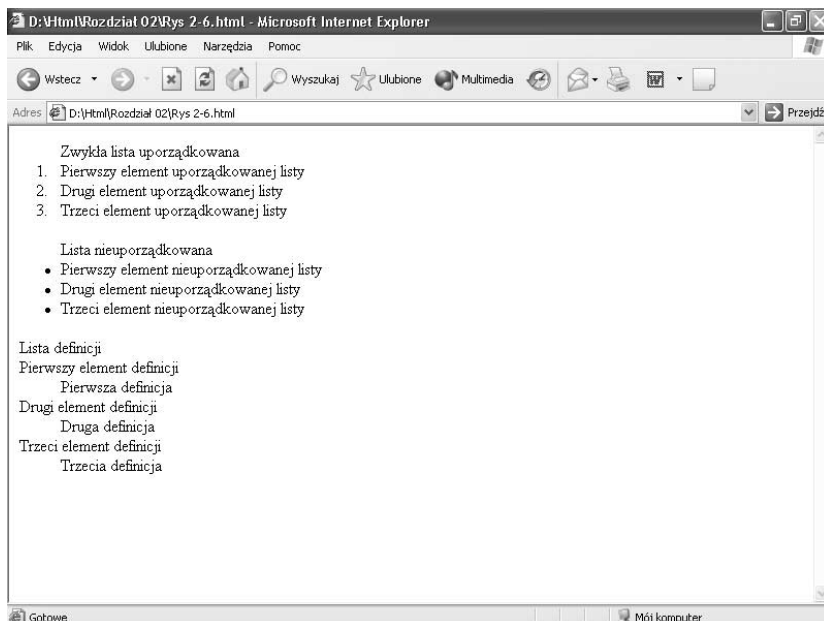
Oto przykładowy kod HTML oraz efekt jego zastosowania przedstawiony na rysunku 2.6.:

```
<html>
<body>
<ol>Zwykła lista uporządkowana
<li>Pierwszy element uporządkowanej listy
<li>Drugi element uporządkowanej listy
<li>Trzeci element uporządkowanej listy
</ol>
<ul>Lista nieuporządkowana
<li>Pierwszy element nieuporządkowanej listy
<li>Drugi element nieuporządkowanej listy
<li>Trzeci element nieuporządkowanej listy
</ul>
<dl>Lista definicji
<dt>Pierwszy element definicji
<dd>Pierwsza definicja

<dt>Drugi element definicji
<dd>Druga definicja

<dt>Trzeci element definicji
<dd>Trzecia definicja
</dl>
</body>
</html>
```

Rysunek 2.6.
Przykładowa lista
w HTML-u

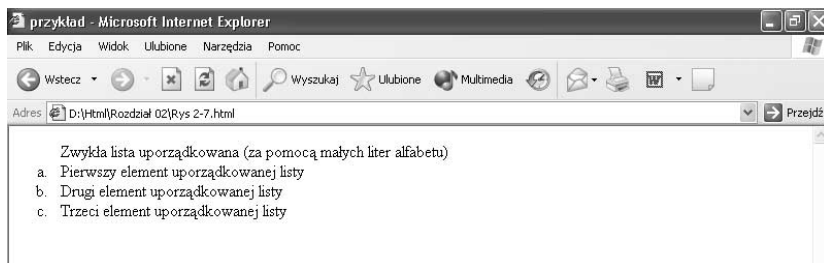


Każdy typ listy można dostosować do własnych potrzeb. Jeśli chcesz, możesz utworzyć wiele stylów dla każdego z typów. Możesz na przykład sprawić, aby kolejne elementy listy uporządkowanej rozpoczynały litery zamiast liczb. Oto kod HTML, który przynosi taki właśnie efekt (zobacz rysunek 2.7):

```

<html>
<body>
<ol style="list-style: lower-alpha;">Zwykła lista uporządkowana (za pomocą małych liter
alfabetu)
<li>Pierwszy element uporządkowanej listy
<li>Drugi element uporządkowanej listy
<li>Trzeci element uporządkowanej listy
</ol>
</body>
</html>
  
```

Rysunek 2.7.
Za pomocą różnych
stylów możesz
zmienić wygląd
każdej listy w Twoim
dokumencie.
Lista pokazana
na tym rysunku
korzysta ze stylu
listy lower-alpha



Starsze wersje HTML-a umożliwiały określanie różnego rodzaju opcji list za pomocą znaczników listy. Jednak formaty najnowszych wersji języków HTML oraz XHTML nakazują definiowanie tych opcji za pomocą stylów.

Zachowanie pierwotnego formatowania tekstu

Czasami chcesz mieć możliwość ręcznego sformatowania tekstu w dokumencie albo chcesz pozostawić formatowanie już zastosowane w danym fragmencie tekstu. Zazwyczaj tekst pochodzi z innego źródła — jest wycięty i wklejony do Twojego dokumentu — i może już mieć sformatowane odstępy, tabulatory i tak dalej. Zastosowanie znacznika `<pre>` sprawia, że przeglądarka traktuje białe znaki dosłownie i nie kondensuje ich tak, jak robi to zwykle.

Na przykład poniższa tabela będzie wyświetlona dokładnie w takiej samej formie, w jakiej została wpisana w kodzie:

```
<pre>
+-----+-----+
| nazwa           | wartość       |
+-----+-----+
| aktualizacja danych | 106009013    |
| aktualizacja zwolnienia | środa, 28.08, 20:18 |
| status           | 0             |
| aktualizacja zasilania | 1069009861    |
+-----+-----+
</pre>
```

Grupowanie elementów

Grupowanie elementów stanowi wyższy poziom formatowania bloków i stosuje się je zazwyczaj do grup powiązanych ze sobą akapitów, całych stron, a czasem tylko do jednego akapitu. Znacznik grupowania (`<div>`) umożliwia formatowanie większych części dokumentu w bardzo prosty sposób. Jeśli na przykład chcesz, aby określony fragment tekstu w dokumencie był otoczony ramką, możesz zdefiniować odpowiedni styl, a następnie umieścić ten fragment między znacznikami `<div>`, tak jak w przykładzie:

```
<html>
<head>
<style>
.bordered { border-style: solid; }
</style>
</head>
<body>
<p>To jest normalny akapit.</p>
<div class="bordered"><p>To jest akapit sformatowany za pomocą stylu div, który dodaje do niego ramkę.</p></div>
</body>
</html>
```

Efekt zastosowania tego kodu przedstawia rysunek 2.8.



Więcej informacji na temat wydzielania bloków tekstu za pomocą znacznika `<div>` znajdziesz w rozdziale 16.

Rysunek 2.8.
Znaczniki <div>
grupują duże
fragmenty tekstu



Elementy wstawiane — znaczniki formatujące znaki

Najwyższy możliwy poziom znaczników dostępny w HTML-u to poziom znaków — tak, jak w programach służących do edycji tekstu, i tu możesz określić formatowanie wybranych znaków. W tym punkcie znajdziesz omówienie elementów wstawianych.

Podstawowe znaczniki wstawiane

Do elementów wstawianych należą:

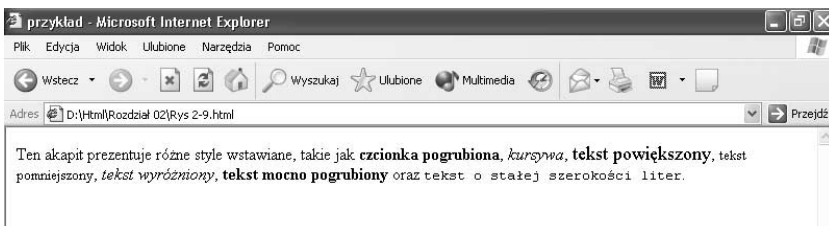
- ♦ czcionka pogrubiona ();
- ♦ kursywa (<i>);
- ♦ tekst powiększony (<big>);
- ♦ tekst pomniejszony (<small>);
- ♦ tekst wyróżniony ();
- ♦ tekst mocno pogrubiony ();
- ♦ tekst o stałej szerokości liter (<tt>).

Spójrz na przykładowy kod zawierający próbkę akapitu oraz na efekt jego zastosowania przedstawiony na rysunku 2.9.

```
<html>
<body>
<p>Ten akapit prezentuje różne style wstawiane, takie jak <b>czcionka pogrubiona</b>,
<i>kursywa</i>, <big>tekst powiększony</big>, <small>tekst pomniejszony</small>,
<em>tekst wyróżniony</em>, <strong>tekst mocno pogrubiony</strong> oraz <tt>tekst
o stałej szerokości liter</tt>.</p>
</body>
</html>
```

Zwróć uwagę na to, że w najnowszych specyfikacjach pominięto kilka znaczników wstawianych, takich jak przekreślenie (<strike>) oraz podkreślenie (<u>). Zrezygnowano nawet ze znacznika czcionki () na rzecz stylów span (zobacz punkt „Elementy span”

Rysunek 2.9.
Elementy wstawiane
mogą zmienić
wygląd wyrazów,
a nawet
pojedynczych
znaków



w dalszej części tego rozdziału). Znaczniki przekreślenia oraz podkreślenia zostały zastąpione znacznikami usunięcia (`<del>`) oraz wstawienia (`<ins>`), które służą do dokonywania korekty (`<del>` dla tekstu usuniętego i `<ins>` dla tekstu wstawionego).



Więcej informacji na temat elementów wstawianych znajdziesz w rozdziale 4.

Elementy span

Znaczniki `span` (`<span>`) grupują style wstawiane w większej liczbie znaków lub wyrazów, a co za tym idzie, umożliwiają definiowanie własnych stylów wstawianych. Jeśli na przykład chcesz, aby tekst był napisany czerwoną, pogrubioną czcionką i do tego jeszcze aby był podkreślony, możesz użyć takiego kodu:

```
<html>
<head>
<style>
.emphasis { color: red; text-decoration: underline; font-weight: bold; }
</style>
</head>
<body>
<p><span class="emphasis">Ten tekst jest wyróżniony</span>, a ten nie.</p>
</body>
</html>
```

Znacznik `<span>` umożliwia zastosowanie formatowania stylu dokładnie w tym miejscu, w którym chcesz.

Znaki specjalne (encje)

Do niektórych znaków specjalnych trzeba odwoływać się bezpośrednio zamiast wpisywać je w dokumencie. Niektóre z nich nie mogą być wpisane ze standardowej klawiatury, na przykład symbol znaku towarowego (™) lub praw autorskich (©); inne mogłyby wprowadzić zamieszanie podczas odczytywania dokumentu przez przeglądarkę (na przykład nawiasy trójkątne, `<` oraz `>`). Tego typu znaki są powszechnie nazywane encjami.

Odwołania do encji tworzymy w dokumencie za pomocą specjalnego kodu. Kod ten zaczyna się zawsze od znaku `&`, a kończy średnikiem. Istnieją trzy różne metody definiowania encji:

- ♦ kod mnemoniczny (na przykład słowo `copy` dla symbolu praw autorskich);
- ♦ wartość dziesiętna odwołująca się do danego znaku (na przykład wartość `#169` dla symbolu praw autorskich);
- ♦ wartość szesnastkowa odwołująca się do danego znaku (na przykład wartość `#xA9` dla symbolu praw autorskich);

Zwróć uwagę na to, że dziesiętne i szesnastkowe metody określania encji wymagają wstawienia przed wartością znaku (#).

Oto przykłady poprawnie zapisanych encji:

- ♦ ` `; — twarda spacja (zobacz później);
- ♦ `<`; — symbol „mniejszy niż” albo lewy nawias kwadratowy;
- ♦ `©`; — znak praw autorskich (©);
- ♦ `&`; — znak &;
- ♦ `—`; — myślnik.



Więcej informacji na temat encji znajdziesz w rozdziale 9.

Nieodpowiednie użycie encji

Jedna z encji, twarda spacja, jest często nadużywana w dokumentach HTML. Aby na przykład wstawić większy odstęp pomiędzy akapitami, twórcy stron często wpisują taki kod:

```
<p>&nbsp;</p>
```

Efektom zastosowania takiego kodu jest pusty akapit. Większość przeglądarek w ogóle go nie wyświetli.

Nie takie jest zamierzenie tej encji — jej zadaniem jest niedopuszczenie do rozdzielenia określonych słów między kolejnymi wierszami tekstu. Natomiast nie zaleca się wstawiania białych znaków w tekście za pomocą twardej spacji. Zamiast tego można skorzystać ze stylów, zgodnie z zaleceniami zaprezentowanymi w tej książce.

Elementy organizacyjne

HTML posiada dwa elementy, które ułatwiają organizowanie informacji w dokumencie: tabele i formularze. Tabele umożliwiają zaprezentowanie danych za pomocą kolumn i wierszy w sposób podobny jak w arkuszach kalkulacyjnych. Formularze służą do prezentowania (oraz wyszukiwania) danych za pomocą elementów wspólnych dla interfejsów GUI — takich jak pola tekstowe, pola wyboru czy listy.

Tabele

Tabele HTML są bardzo proste, ale odpowiednio użyte mogą przynieść naprawdę bardzo ciekawy efekt. Na najbardziej podstawowym poziomie tabele służą do organizowania danych w wiersze i kolumny. Na poziomie najwyższym tabele mogą prezentować bardzo skomplikowany układ strony — tak jak w czasopiśmie lub gazecie, gdzie w kolumnach umieszczany jest tekst, grafika, różnego rodzaju menu i tak dalej.

Tabele składają się z trzech podstawowych elementów, a co za tym idzie, posiadają trzy znaczniki:

- ♦ Sama definicja tabeli jest określana za pomocą znacznika `<table>`.
- ♦ Wiersze danych są definiowane za pomocą znaczników `<tr>`.
- ♦ Komórki tabeli (pojedyncze dane) są definiowane za pomocą znaczników `<td>`.
Komórki tabeli, umieszczone w równych wierszach, tworzą kolumny tabeli.

Spójrz na przykładowy kod, którego efekt zastosowania przedstawia rysunek 2.10.

```
<html>
<body>
<table border="1">
<tr><td>Imię</td><td>Wiek</td></tr>
<tr><td>Anna</td><td>35</td></tr>
<tr><td>Bartosz</td><td>29</td></tr>
<tr><td>Dariusz</td><td>23</td></tr>
<tr><td>Jan</td><td>31</td></tr>
<tr><td>Jerzy</td><td>34</td></tr>
<tr><td>Krzysztof</td><td>33</td></tr>
<tr><td>Marcin</td><td>39</td></tr>
<tr><td>Michał</td><td>25</td></tr>
<tr><td>Stefan</td><td>38</td></tr>
<tr><td>Szymon</td><td>40</td></tr>
</table>
</body>
</html>
```

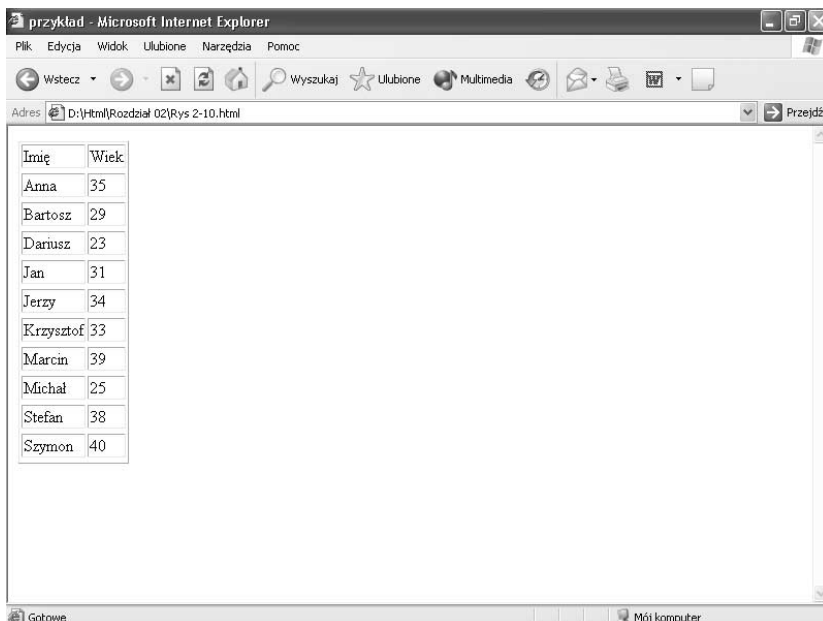
Jest to przykład bardzo prostej tabeli. HTML oferuje wiele opcji, które umożliwiają sformatowanie elementów tabeli, jak również zagnieżdżenie jednej tabeli wewnątrz drugiej. W ten sposób można znacznie poprawić wygląd tabeli, a także sprawić, aby była ona bardziej złożona. Porównaj ze sobą rysunki 2.11 oraz 2.12. Rysunek 2.11 przedstawia stronę w taki sposób, w jaki normalnie jest ona wyświetlana przez przeglądarkę. Jeśli jednak włączysz ramki tabeli (zobacz rysunek 2.12), przekonasz się, że do osiągnięcia odpowiedniego wyglądu dokumentu posłużono się tabelami (oraz tabelami zagnieżdżonymi).



Tabele zostały omówione szczegółowo w rozdziale 10., natomiast w rozdziale 11. znajdziesz informacje na temat tego, jak można wpływać na wygląd strony za pomocą tabel.

Rysunek 2.10.

*Jedenaście wierszy
i dwie kolumny
danych w tabeli*

**Rysunek 2.11.**

*W tym dokumencie
wykorzystano
dwie niewidoczne
tabele, aby nadać
mu odpowiedni
wygląd*



Formularze

W formularzach HTML można wyświetlać i gromadzić dane za pomocą standardowych elementów GUI. Formularze HTML oferują standardowy zestaw elementów GUI, takich jak pola tekstowe, pola wyboru, listy rozwijane i wiele innych. Oprócz podstawowych

Rysunek 2.12.

Tu wyświetlono ramki tabel, aby pokazać, ile tabel posłużyło do utworzenia strony i w jaki sposób wpłynęły one na jej wygląd



elementów GUI HTML oferuje również prostą metodę gromadzenia danych i przekazywania ich do programu obsługi danych, w którym można sprawdzić ich poprawność, a także przechowywać je, porównywać i tak dalej.

Oto kod typowego formularza HTML oraz efekt jego zastosowania przedstawiony na rysunku 2.13:

```
<html>
<body>
<form>
<!-- Pole tekstowe -->
<b>Imię:</b><input type="text" name="imię" size="40">
<br><br>
<!-- przełączniki-->
<b>Wiek:</b>
<input type="radio" name="wiek"> < 20
<input type="radio" name="wiek"> 21 -- 30
<input type="radio" name="wiek"> 31 -- 40
<input type="radio" name="wiek"> 41+
<br><br>
<!-- Lista wyboru -->
<b>Jaki jest Twój ulubiony smak lodów?</b>
<select name="lody">
<option name="czekoladowe">Czekoladowe
<option name="truskawkowe">Truskawkowe
<option name="waniliowe">Waniliowe
</select>
<br><br>
<!-- Pola wyboru -->
<b>Jak możemy się z Tobą skontaktować?</b><br>
<input type="checkbox" name="telefon">Telefon<br>
<input type="checkbox" name="poczta">Poczta<br>
```

Rysunek 2.13.

Elementy formularza to standardowe kontrolki GUI, które umożliwiają wyświetlanie i gromadzenie danych

```
<input type="checkbox" name="e-mail">E-mail<br>
<input type="checkbox" name="nie">Nie jestem zainteresowany<br>
</form>
</body>
</html>
```

Formularz przedstawiony w powyższym przykładzie jest bardzo prosty — prezentuje tylko kilka podstawowych elementów i nie zawiera żadnego programu obsługi danych, który mógłby je przetwarzać. W rzeczywistości formularze bywają bardzo skomplikowane i zazwyczaj wymagają skryptów sprawdzających poprawność zgromadzonych danych. Jednak nawet ten prosty formularz pokazuje, że przy wykorzystaniu HTML-a możesz uzyskać całkiem niezłą kontrolę nad danymi oraz ich formatem.



Formularze zostały szczegółowo omówione w rozdziale 13.

Łączy do innych stron

Główną zaletą sieci WWW jest możliwość przełączania się między różnymi dokumentami. Jeżeli na przykład tworzysz stronę, która dotyczy praw lokalnych, możesz umieścić w niej łącze do strony rządowej, na której użytkownik znajdzie dodatkowe informacje. Łącze jest zazwyczaj wyświetlane jako podkreślony tekst, a jego kolor różni się od normalnego tekstu dokumentu.

Przykładowo, łącze może wyglądać w przeglądarce tak:

Więcej informacji znajdziesz tutaj.

Słowo tutaj jest połączone z innym dokumentem — gdy użytkownik je kliknie, jego przeglądarka wyświetli określoną stronę WWW.

Łącza tworzy się za pomocą znacznika <a>. Na najprostszym poziomie znacznik ten przyjmuje jeden argument — stronę, z którą ma połączyć — i otacza tekst, który ma stanowić łącze. W poprzednim przykładzie można by użyć następującego kodu:

```
Więcej informacji znajdziesz <a href="http://www.sejm.gov.pl">tutaj</a>
```

Atrybut href (ang. *Hypertext REFerence* — odnośnik do hipertekstu) znacznika <a> definiuje protokół oraz docelową lokalizację łącza. W tym przykładzie zdefiniowano protokół http://, ponieważ strona docelowa jest dostarczana za pomocą protokołu HTTP. Można również korzystać z innych protokołów (takich jak ftp:// czy mailto:), jeżeli istnieje taka potrzeba.

Znacznik odnośnika może również zawierać dodatkowe atrybuty, które określają na przykład to, gdzie ma zostać otwarty nowy dokument (na przykład w nowym oknie przeglądarki), a także zdefiniują wzajemny stosunek między dokumentami i zestaw znaków czcionki użyty w nowym dokumencie.

Możesz również skorzystać z pewnej odmiany znacznika <a>, aby zaznaczyć wybrane miejsca w bieżącym dokumencie. Umieszczając łącze w innym miejscu tego samego dokumentu umożliwisz użytkownikowi przeniesienie się w określone miejsce. Spójrz na przykładowy kod HTML:

```
Więcej informacji znajdziesz w <a href="#Rozdz2">Rozdziale 2.</a>
...kod HTML...
<a name="Rozdz2">Rozdział 2.</a>
```

W tym przykładzie użytkownik może kliknąć łącze „Rozdział 2.,” aby przenieść się w miejsce, w którym znajduje się odnośnik do rozdziału 2. Zwróć uwagę na to, że łącze href musi zawierać symbol hash (#), który definiuje je jako odnośnik, a nie jako odrębną stronę.



Więcej informacji na temat łączy oraz odnośników znajdziesz w rozdziale 7.

Obrazy

Jedną z największych innowacji, jakie wniosły do internetu sieć WWW oraz protokół HTTP jest możliwość dostarczania multimediów do przeglądarki. Prekursorami filmów wideo i dźwięku o jakości takiej, jak na płytach CD, były obrazy graficzne w formacie GIF oraz JPEG.

Do dodawania obrazów do dokumentu HTML służy znacznik . Znacznik ten zawiera łącze do pliku z obrazem, a także przechowuje informacje niezbędne do tego, aby poprawnie wyświetlić obraz (na przykład określające jego rozmiar). Typowy znacznik obrazu wygląda mniej więcej tak:

```

```

Efektem zastosowania tego kodu byłoby wyświetlenie obrazu *tmoore.jpg* w tym miejscu dokumentu, w którym umieszczony jest znacznik. W tym przypadku obraz znajduje się w katalogu images bieżącego serwera i zostanie wyświetlony bez ramki, w rozmiarze 100 na 200 pikseli. Atrybut alt zawiera tekstowy odpowiednik obrazu dla przeglądarek, które nie mogą wyświetlić grafiki (lub pojawiający się w przypadku, jeśli sam użytkownik zdecydował się na taką konfigurację).

Obrazy mogą również pomagać w nawigacji po dokumencie. Możesz na przykład sprawić, aby ich kliknięcie przez użytkownika powodowało wykonanie jakiejś czynności, wyświetlenie innego dokumentu i tak dalej. Na przykład mapa Polski mogłaby służyć do tego,

aby wybrać określone województwo — kliknięcie województwa wywoływałoby wyświetlenie powiązanej tematycznie strony. Obrazy nawigacyjne są często nazywane *mapami obrazów* i z reguły wymagają oddzielnej mapy opartej na współrzędnych i zawierającej figury geometryczne, definiujące obszary, które można kliknąć na obrazie.



Więcej informacji na temat obrazów znajdziesz w rozdziale 6.

Komentarze

Dokumenty HTML są z reguły czytelne, ale czasem warto dodać do kodu komentarz. Najczęstszym powodem wstawiania komentarzy w kodzie HTML jest poprawa organizacji dokumentu, a także chęć zaznaczenia wybranych fragmentów po to, aby później się do nich odwołać.

HTML wykorzystuje znacznik `<!--` do rozpoczęcia komentarza oraz `-->` do jego zakończenia. Nawet jeśli komentarze rozciągają się na kilka linii, przeglądarka zawsze zignoruje wszystko, co znajduje się między znacznikami komentarza. Oto przykład dwóch komentarzy, które zostaną zignorowane przez przeglądarkę:

```
<!-- Ta część dokumentu wymaga lepszej organizacji. -->
```

oraz

```
<!-- W tej tabeli trzeba wstawić następujące kolumny:  
Wiek  
Stan cywilny  
Data zatrudnienia  
-->
```

Skrypty

HTML jest statyczną metodą rozmieszczania treści dokumentu — treść ta jest wysyłana do przeglądarki użytkownika, która ją odczytuje i wyświetla, ale zazwyczaj nie zmienia jej formy. Jednak czasem zachodzi potrzeba podjęcia pewnych decyzji, sprawdzenia poprawności formularzy, a także, w przypadku dynamicznego HTML-a (DHTML) — zmian atrybutów dynamicznych obiektów. W tych przypadkach (a także w wielu innych) można użyć skryptów uruchamianych po stronie klienta.



Więcej informacji na temat skryptów uruchamianych po stronie klienta znajdziesz w rozdziale 15.

Języki skryptów uruchamiane po stronie klienta, takie jak JavaScript, przekazują swój kod do przeglądarki, umieszczając go wewnątrz dokumentu HTML. Zadaniem przeglądarki jest zinterpretowanie kodu i podjęcie odpowiednich czynności. Większość skryptów uruchamianych po stronie klienta znajduje się w części `<head>` dokumentu HTML, wewnątrz znaczników `<script>`, tak jak w poniższym przykładzie:

```

<html>
<head>
<script language="JavaScript">
function MiscWindow(w,h,url){
  opts = "width="+w+",height="+h;
  opts = opts+",scrollbars=no,resizable=yes";
  fin=window.open(url,"",opts);
}
</script>
</head>...

```

W większości przypadków dokument musi zawierać zdarzenia, które uruchomią skrypty. Zdarzenia te mogą być osadzone w elementach (poprzez atrybut `onmouseover` lub podobne), powiązane z łączami, wywoływane za pomocą elementów formularza lub uruchamiane wraz z załadowaniem lub wyładowaniem dokumentu (poprzez atrybuty `onload` oraz `onunload` umieszczone w znaczniku `<body>`).



Istnieją metody umożliwiające automatyczne uruchamianie skryptów, czyli takie, które nie wymagają wystąpienia żadnego zdarzenia. Jednak nie zaleca się ich stosowania — dużo lepiej jest powiązać uruchomienie skryptu z jakimś zdarzeniem.

Wszystko razem

Jak widzisz, standardowy dokument HTML jest dość skomplikowaną bestią. Jeśli jednak przeanalizujesz kod fragment po fragmencie, okaże się, że w różnych dokumentach HTML znajdziesz wiele wspólnych elementów. Oto listing HTML, który łączy te wszystkie elementy w jednym kodzie.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta ...tutaj są umieszczone znaczniki meta...>
<title> tutaj znajduje się tytuł strony (dokumentu)</title>
<LINK rel="stylesheet" href="nazwa zewnętrznego arkusza stylów"
type="text/css">
<style>
... tutaj są umieszczone określone style...
</style>
<script>
... tutaj są umieszczone skrypty uruchamiane po stronie klienta...
</script>
<body>
...tutaj znajduje się treść zasadnicza dokumentu, akapity zmodyfikowane za pomocą elementów blokowych,
znaki, słowa i zdania zmodyfikowane przez elementy wstawiane...
</body>
</html>

```

Wszystkie dokumenty HTML powinny zawierać specyfikację `<DOCTYPE>`, znaczniki `<html>` oraz `<body>`, a także przynajmniej tytuł `<title>` oraz część `<head>`. Pozostałe elementy są opcjonalne, ale pomagają zdefiniować cel dokumentu, jego styl, a w rezultacie — jego użyteczność, o czym przekonasz się podczas czytania kolejnych rozdziałów.

Podsumowanie

Wiesz już, jakie podstawowe elementy składają się na dokument HTML. Ich ilość może na początku trochę przytłaczać, ale szybko nauczysz się, do czego służy każdy z nich i jaki jest jego wpływ na pozostałe elementy w dokumencie. Dzięki temu będziesz wiedział, jak najlepiej wykorzystać poszczególne elementy, aby utworzyć dokument HTML odpowiadający Twoim zamierzeniom. Kiedy będziesz znać już szczegółowe informacje na temat każdego z nich — znajdziesz je w kilku kolejnych rozdziałach — spróbuj dostosować ich możliwości do własnych potrzeb.

Teraz powinienes przeczytać rozdziały od 3. do 24., aby poszerzyć swoją wiedzę na temat różnych elementów HTML. Możesz również przejść do wybranych rozdziałów omawiających te elementy, które Cię najbardziej interesują lub takie, które sprawiają Ci największe trudności. (Kieruj się odnośnikami zamieszczonymi w każdym punkcie tego rozdziału, aby odnaleźć rozdział omawiający element, o którym chcesz się dowiedzieć czegoś więcej).