

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

Po prostu HTML, XHTML i CSS. Wydanie VI

Autor: Elizabeth Castro

Tłumaczenie: Piotr Rajca

ISBN: 978-83-246-0840-9

Tytuł oryginału: [HTML, XHTML, and CSS, Sixth Edition: Visual QuickStart Guide \(6th Edition\)](#)

Format: B5, stron: około 500



Stwórz własną witrynę WWW

- Poznaj znaczniki języka HTML
- Zdefiniuj kaskadowe arkusze stylów
- Przetestuj witrynę i opublikuj ją na serwerze

Język HTML i jego najnowsze wcielenie, określane nazwą XHTML, to podstawowy „budulec” witryn WWW. Stosując odpowiednie znaczniki, możemy umieszczać na stronach tekst i grafikę, tworzyć hiperłącza oraz osadzać elementy multimedialne i interaktywne. Technologia CSS (kaskadowych arkuszy stylów) pozwala na przypisanie do znaczników języka HTML definicji formatowania, dzięki czemu zmiana kolorystyki witryny bądź kroju czcionki wymaga jedynie modyfikacji pliku ze stylami. Poznanie tego języka i zasad korzystania z kaskadowych arkuszy stylów jest niezbędne do tworzenia własnych witryn WWW, ponieważ nawet tak bardzo popularne dziś „wizualne” narzędzia generują kod, który często wymaga ręcznych poprawek.

Książka „Po prostu HTML, XHTML i CSS. Wydanie VI” to kolejna edycja bestsellerowego poradnika, dzięki któremu tajniki tworzenia witryn WWW poznały tysiące czytelników na całym świecie. W wydaniu VI autorka uwzględniła najnowsze specyfikacje języka HTML oraz technologii CSS. Czytając tę książkę, dowiesz się, w jaki sposób wykorzystywać znaczniki do tworzenia elementów stron internetowych oraz osadzania na nich grafiki i obiektów interaktywnych. Nauczysz się definiować hiperłącza oraz wykorzystywać kaskadowe arkusze stylów do formatowania tekstu i tabel oraz określania kolorystyki stron. Przeczytasz także o testowaniu witryn, publikowaniu ich na serwerach oraz optymalizowaniu pod kątem wyszukiwarek internetowych.

- Struktura dokumentów HTML i witryn WWW
- Wprowadzanie i formatowanie tekstu
- Przygotowywanie grafiki na strony WWW
- Umieszczanie elementów graficznych w dokumentach HTML
- Tworzenie hiperłączy
- Definiowanie stylów CSS
- Formatowanie z wykorzystaniem stylów
- Tworzenie list i tabel
- Elementy formularzy HTML
- Osadzanie na stronie elementów multimedialnych
- Sprawdzanie poprawności kodu HTML
- Publikacja strony

Dołącz do grona webmasterów, którzy poznali język HTML dzięki tej książce



Wstęp	15
Internet, WWW oraz HTML	16
Otwarty, jednak nie identyczny	17
Wojny przeglądarek	18
Dążenie do wprowadzania standardów	19
CSS a możliwości przeglądarek	23
XHTML czy HTML: czego należy używać?	24
Kilka słów o niniejszej książce	26
Co się zmieniło w szóstym wydaniu?	27
Witryna WWW autorki	29
Rozdział 1. Elementy tworzące strony WWW	31
Znaki: elementy, atrybuty oraz wartości	32
Tekstowa zawartość stron WWW	36
Łąca, obrazy oraz inna zawartość nie będąca tekstem	37
Nazwy plików	38
Adresy URL	39
HTML a XHTML	42
Wersje, rodzaje oraz DOCTYPE	45
Domyślny sposób wyświetlania (X)HTML	48
Rozdział 2. Praca z dokumentami (X)HTML	49
Projektowanie witryny	50
Tworzenie nowej strony WWW	51
Zapisywanie stron WWW	52
Kilka słów o programie Microsoft Word i stronach WWW	54
Określanie strony domyślnej („domowej”)	55
Edycja stron WWW	56
Organizowanie plików	57
Wyświetlanie stron w przeglądarce	58
Czerpanie inspiracji od innych	59
Rozdział 3. Podstawowa struktura dokumentów (X)HTML	61
Rozpoczynanie strony	62
Tworzenie podstawowej struktury kodu	64
Określanie sposobu kodowania	65
Nadawanie tytułu	66
Tworzenie nagłówków sekcji	67

Rozpoczynanie nowego akapitu68
Nazywanie elementów69
Podział strony na działy70
Tworzenie obszarów wewnątrzwierszowych71
Tworzenie nowych wierszy72
Dodawanie komentarzy73
Nadawanie nazw elementom stron74
Rozdział 4. Podstawy formatowania tekstu	75
Tworzenie tekstu pogrubionego oraz kursywy76
Zmiana wielkości tekstu77
Stosowanie czcionki o stałej szerokości znaków78
Stosowanie tekstu preformatowanego80
Cytaty81
Tworzenie indeksów dolnych i górnych83
Oznaczanie zmodyfikowanego tekstu.84
Wyjaśnianie skrótów.85
Wyśrodkowywanie zawartości strony.86
Rozdział 5. Obrazy	87
O obrazach tworzonych na potrzeby stron WWW.88
Zdobywanie obrazów92
Wybór programu do edycji grafiki93
Polecenie Zapisz dla Webu94
Wstawianie ilustracji na stronę96
Tekst zastępczy.97
Określanie wymiarów obrazu w celu jego szybszego wyświetlenia99
Skalowanie ilustracji.	101
Zmniejszanie wymiarów obrazu	103
Otaczanie obrazów tekstem	105
Zakończenie otaczania tekstem	107
Zwiększanie odstępu wokół obrazów	108
Wyrównywanie obrazków	109
Poziome linie.	110
Dodawanie ikony witryny	111
Rozdział 6. Łąca	113
Tworzenie łączy do innych stron	114
Tworzenie odnośników	116
Łąca wykorzystujące odnośniki	117
Tworzenie łączy do wybranego okna	118
Określenie domyślnego okna	119
Tworzenie innych rodzajów łączy	120

Definiowanie klawiszy skrótów dla łączы	122
Określenie kolejności łączы dla klawisza TAB	123
Zastosowanie obrazów do tworzenia łączы	124
Łączenie miniatur z obrazami	125
Podział obrazka na obszary przypisane do różnych łączы	126
Tworzenie map odnośników obsługiwanych po stronie klienta	127
Rozdział 7. Elementy arkuszy stylów	129
Tworzenie reguły stylu	130
Dodawanie komentarzy do arkuszy stylów	131
Kaskada: gdy reguły kolidują ze sobą	132
Wartości właściwości	135
Rozdział 8. Stosowanie plików arkuszy stylów	139
Tworzenie zewnętrznego arkusza stylów	140
Dołączanie zewnętrznego arkusza stylów	141
Udostępnianie alternatywnych arkuszy stylów	142
Tworzenie wewnętrznego arkusza stylów	144
Importowanie zewnętrznych arkuszy stylów	145
Stosowanie arkuszy stylów dla różnych mediów	147
Lokalne stosowanie stylów	148
Znaczenie położenia	150
Czerpanie inspiracji od innych	152
Rozdział 9. Definiowanie selektorów	153
Tworzenie selektorów	154
Wybór elementów na podstawie nazwy	155
Wybieranie elementu na podstawie klasy lub identyfikatora	156
Wybieranie elementów na podstawie kontekstu	157
Wybieranie fragmentów elementu	160
Wybieranie łączы na podstawie ich stanu	162
Wybieranie elementów na podstawie atrybutów	163
Definiowanie grup elementów	164
Łączenie selektorów	165
Rozdział 10. Formatowanie przy wykorzystaniu stylów	167
Wybór czcionki	168
Określanie czcionek alternatywnych	169
Tworzenie czcionki pochylej (kursywy)	170
Pogrubianie czcionki	171
Określanie wielkości czcionki	172
Określanie wysokości linii	175
Jednoczesne określanie wszystkich parametrów czcionki	176

Definiowanie koloru tekstu	177
Zmiana koloru tła tekstu.	178
Kontrola odstępów pomiędzy wyrazami i literami	179
Dodawanie wcięć akapitowych	180
Parametry odstępów	181
Wyrównywanie tekstu	182
Zmiana wielkości liter	183
Wykorzystanie kapitalików	184
Dekorowanie tekstu	185

Rozdział II. Określanie układu strony za pomocą stylów **187**

Określanie struktury strony	188
Model pudełkowy	189
Modyfikowanie tła	190
Określanie wysokości i szerokości elementu	192
Określanie marginesów wokół elementu	195
Dodawanie wypełnienia wokół elementu	197
Przesuwanie elementów względem ich naturalnego położenia	199
Bezwzględne rozmieszczanie elementów	200
Określanie stałego położenia elementu w oknie przeglądarki	202
Otoczanie elementów tekstem	204
Kontrola sposobu otaczania elementów	205
Pozycjonowanie elementów w trzecim wymiarze	206
Tworzenie obramowań	207
Modyfikacja wskaźnika myszy	210
Określanie sposobu wyświetlania zawartości elementu	211
Wyrównywanie elementów w pionie	212

Rozdział 12. Tworzenie dynamicznych efektów przy użyciu stylów **213**

Wyświetlanie i ukrywanie elementów	214
Tworzenie podmienianych przycisków	216
Tworzenie wyskakujących elementów	217
Tworzenie rozwijalnych menu	218
Zastępowanie nagłówków obrazami	220

Rozdział 13. Style dla urządzeń kieszonkowych **223**

Mobilizacja kontra miniaturyzacja	224
Rzut oka na własną witrynę	225
XHTML i CSS dla komputerów kieszonkowych	226
Tworzenie arkuszy stylów dla komputerów przenośnych	228

Ukrywanie niepotrzebnych elementów	229
Tworzenie i stosowanie nagłówka graficznego	230
Dodawanie łączy prowadzących na początek strony	231
Poprawa sposobu prezentacji strony	232
Rozdział 14. Arkusze stylów przeznaczone do drukowania stron	233
Stosowanie arkuszy stylów przeznaczonych dla konkretnych rodzajów mediów	234
Czym różnią się style przeznaczone do drukowania	235
Kontrola dzielenia dokumentów na strony	237
Drukowanie adresów URL łączy	238
Kontrola wdów i sierot	239
Rozdział 15. Listy	241
Tworzenie list wypunktowanych i uporządkowanych	242
Określanie kształtu znaczników (punktów)	244
Określanie początkowej wartości numeracji punktów	245
Stosowanie niestandardowych znaczników	246
Określanie miejsca wyświetlania znaczników	248
Określanie wszystkich właściwości listy w jednym miejscu	249
Tworzenie list definicji	250
Określanie wyglądu list zagnieżdżonych	251
Rozdział 16. Tabele	253
Projektowanie układu strony	254
Tworzenie prostej tabeli	255
Dodawanie krawędzi tabeli	256
Określanie szerokości tabel	259
Wyrównywanie tabeli do środka strony.	262
Otoczanie tabeli tekstem	263
Łączenie tabel	264
Wyrównywanie zawartości komórek	266
Zmiana koloru tła	269
Kontrola odstępów pomiędzy komórkami i wewnątrz nich	271
Łączenie komórek leżących w sąsiednich kolumnach	274
Łączenie komórek w sąsiednich wierszach	275
Podział tabeli na grupy kolumn	276
Podział tabeli na poziome sekcje	278
Wybór linii do wyświetlania.	279
Kontrola łamania wierszy w komórce.	281
Przyspieszenie wyświetlania tabeli	282

Rozdział 17. Formularze	283
Tworzenie formularza	284
Przetwarzanie formularzy	286
Przesyłanie danych pocztą elektroniczną	288
Organizacja elementów formularzy	290
Tworzenie pól tekstowych	292
Tworzenie pól hasła	293
Formalne nadanie etykiet elementom formularzy	294
Tworzenie przycisków opcji	295
Rozwijalne listy	296
Tworzenie pól wyboru	298
Obszary tekstowe	299
Umożliwienie użytkownikom przesyłania plików	300
Dodawanie pól ukrytych do formularzy	301
Tworzenie przycisku przesyłającego	302
Czyszczenie zawartości formularza	304
Stosowanie obrazów do przesyłania danych	306
Określenie kolejności klawisza TAB w formularzach	308
Definiowanie klawiszy skrótów	309
Dezaktywacja elementów formularza	310
Uniemożliwienie modyfikacji elementów	311
 Rozdział 18. Multimedia	 313
Kilka słów o pluginach i odtwarzaczach	314
Zdobywanie odtwarzaczy dla użytkowników	315
Pobieranie plików multimedialnych	316
Tworzenie łączy do plików multimedialnych	317
Osadzanie w stronach filmów QuickTime	319
Osadzanie filmów QuickTime dla wszystkich przeglądarek oprócz IE	322
Osadzanie filmów QuickTime tak, by działały we wszystkich przeglądarkach	324
Zastosowania JavaScriptu do odtwarzania filmów	326
Skalowanie filmów QuickTime	328
Zmiana koloru tła filmów	329
Odtwarzanie filmów QuickTime w pętli	330
Udostępnianie zwiastuna filmu	331
Dodawanie atrybutów do kolejnych filmów	332
Udostępnianie kolejnych filmów	333
Umieszczanie plików MP3 na stronach WWW	334
Osadzanie plików Windows Media, część I	336
Osadzanie plików Windows Media, część II	338
Osadzanie plików Flash	339

Wyświetlanie klipów wideo z witryn Google i YouTube	340
Dodłączanie apletów	342
Osadzanie innych plików multimedialnych	343
Tworzenie automatycznego pokazu slajdów	344
Rozdział 19. Skrypty	345
Wstawianie „automatycznego” skryptu	346
Wywołanie zewnętrznego skryptu automatycznego	347
Wyzwalanie skryptu	348
Tworzenie przycisku, który wykonuje skrypt	350
Dodawanie informacji zastępczych	351
Ukrywanie skryptu przed starszymi przeglądarkami	352
Ukrywanie skryptów przed analizatorami składni XML	353
Definiowanie domyślnego języka skryptowego	354
Rozdział 20. Podstawy JavaScriptu	355
Dodawanie aktualnej daty i godziny	356
Określanie wielkości nowego okna przeglądarki	357
Podmianianie obrazków po wskazaniu ich myszką	359
Ładowanie obrazków do pamięci podręcznej	361
Rozdział 21. Symbole oraz inne znaki nie należące do alfabetu angielskiego	363
Kilka słów o sposobach kodowania	364
Deklarowanie sposobu kodowania strony	367
Deklarowanie sposobu kodowania arkusza stylów	369
Zapisywanie strony przy wykorzystaniu odpowiedniego sposobu kodowania	370
Edytowanie stron przy wykorzystaniu poprawnego sposobu kodowania	371
Dodawanie znaków nie należących do danego sposobu kodowania	372
Określanie języka strony	375
Rozdział 22. Testowanie i uruchamianie stron WWW	377
Kilka technik testowania	378
Najpierw rozwiąż najprostsze problemy	379
Sprawdzanie prostych błędów: HTML	380
Sprawdzanie prostych błędów: XHTML	381
Sprawdzanie prostych błędów: CSS	382
Walidacja kodu	384
Testowanie stron	385
Gdy przeglądarka wyświetla kod	387
Kiedy obrazki nie są wyświetlane	388
Różnice pomiędzy poszczególnymi przeglądarkami	389
Wciąż nie działa?	391

Rozdział 23. Publikowanie stron w sieci WWW	393
Jak znaleźć serwer dla swoich stron	394
Gdzie w Polsce opublikować strony WWW?	395
Onet.pl – Republika WWW.	395
Rejestracja nazwy domeny	404
Przesyłanie plików na serwer	405
Rozdział 24. Zdobywanie użytkowników	409
O słowach kluczowych	410
Jawne podawanie słów kluczowych.	411
Opis strony.	412
Zarządzanie innymi informacjami o stronie	413
Jak uniknąć odwiedzin	414
Zapobieganie archiwizacji strony	415
Tworzenie strony z adresami	416
Zastosowanie narzędzia Google Sitemaps	417
Dodawanie witryny do wyszukiwarki.	418
Jak zapewnić wysoką pozycję strony w wynikach wyszukiwania	419
Pisanie stron łatwych do indeksowania.	420
Inne sposoby reklamowania witryny	422
Rozdział 25. Kanaly informacyjne i podcasting	423
Jak wygląda kanał informacyjny.	424
Przygotowania do tworzenia kanału RSS.	426
Rozpoczynanie tworzenia kanału RSS	427
Zamieszczanie opisu witryny	429
Dodawanie wpisów do kanału	431
Dodawanie załączników.	433
Tworzenie podcastów obsługiwanych przez program iTunes	435
Weryfikacja poprawności kanału	441
Przesyłanie podcastu na serwer iTunes.	442
Publikowanie kanału RSS na własnej witrynie WWW	443
Subskrybowanie kanału RSS	445
Subskrybowanie podcastów w programie iTunes	446

Dodatek A	Elementy i atrybuty (X)HTML	447
	Elementy i atrybuty (X)HTML	448
Dodatek B	Właściwości i wartości CSS	457
	Właściwości i wartości CSS	458
Dodatek C	Zdarzenia wbudowane	465
	Zdarzenia wbudowane	466
Dodatek D	Symbole i znaki (X)HTML	467
Dodatek E	Wartości szesnastkowe	479
	Szesnastkowe odpowiedniki liczb w systemie dziesiętnym	480
Dodatek F	Narzędzia (X)HTML	481
	Edytory (X)HTML.	482
	Obrazy i grafika	484
	Programy graficzne	485

Podstawowa struktura dokumentów (X)HTML

3

W tym rozdziale przedstawione zostały najprostsze elementy dokumentów (X)HTML, konieczne do stworzenia podstawowej struktury dokumentu. Zawarto w nim też porady dotyczące tworzenia akapitów, nagłówków, nowych wierszy, wpisywania komentarzy i tworzenia kilku innych elementów stron.

Utworzenie przejrzystej i spójnej struktury znacznie ułatwia późniejsze określanie wyglądu stron za pomocą kaskadowych arkuszy stylów.

Rozpoczynanie strony

Kod strony należy rozpocząć od deklaracji DOCTYPE (patrz strona 53), określającej typ używanego języka HTML lub XHTML. Deklaracja ta informuje przeglądarki, czego mogą się spodziewać na stronie, i umożliwia poprawne działanie narzędzi sprawdzających poprawność kodu. Po wstawieniu tej deklaracji należy rozpocząć właściwy kod dokumentu, wpisując otwierający znacznik `html`.

Aby stworzyć stronę w przejściowej wersji języka HTML 4:

1. Wpisz `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">`, aby zadeklarować, że używana będzie przejściowa wersja języka HTML 4.01.
2. Wpisz `<html>`, aby rozpocząć faktyczny kod HTML strony.
3. Zostaw kilka wolnych wierszy na treść dokumentu.
4. Wpisz zamykający znacznik `</html>`.

Aby stworzyć stronę w przejściowej wersji języka XHTML:

1. Wpisz `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">`, aby zadeklarować, że używana będzie przejściowa wersja języka XHTML.
2. Wpisz `<html xmlns="http://www.w3.org/1999/xhtml">`, aby rozpocząć faktyczny kod XHTML strony.
3. Zostaw kilka wolnych wierszy na treść dokumentu.
4. Wpisz zamykający znacznik `</html>`.

Wskazówki

- Utwórz szablony zawierające odpowiednie deklaracje DOCTYPE i znaczniki `html` oraz używaj ich jako punktu wyjścia do tworzenia wszystkich stron

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

</html>
```

Rysunek 3.1. Oto deklaracja DOCTYPE dokumentu pisanego w przejściowej wersji języka HTML, wraz z otwierającym i zamykającym znacznikiem `html`. To straszny, niezrozumiały fragment tekstu. Sugeruję, aby przy tworzeniu nowych stron nie wpisywać go samodzielnie, lecz skopiować z wcześniej napisanego dokument

HTML czy XHTML

Bardziej wyczerpujące informacje o tym, czy powinieneś używać języka HTML, czy XHTML, możesz znaleźć na stronie 32, „XHTML czy HTML: Czego powinieneś używać”, oraz 50, „HTML a XHTML”. Na razie, najprościej rzecz ujmując, można powiedzieć, że o ile tylko zachowasz spójność i niczego nie pomylisz, wybór używanego języka nie ma większego znaczenia. Jeśli używałeś języka HTML, to używaj go dalej, a jeśli zdecydowałeś się na zastosowanie XHTML-a, stosuj się do jego reguł. Przykłady prezentowane w niniejszej książce zostały napisane w języku XHTML, gdyż sprzyja on stosowaniu standardów i zachowywaniu spójności, a to z kolei gwarantuje większe prawdopodobieństwo, że strony będą wyświetlane w jednolity, podobny sposób w wielu różnych przeglądarkach działających w wielu różnych platformach systemowych. Język HTML jest mniej rygorystyczny, dzięki czemu ułatwia nieco życie autorom stron. Jednak ta nieco większa prostota tworzenia kodu niesie ze sobą pewne zagrożenie, mianowicie może doprowadzić do niespójnego i nieoczekiwanego wyglądu strony oglądanej na różnych przeglądarkach, działających w różnych systemach operacyjnych.

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

</html>
```

Rysunek 3.2. Oto deklaracja DOCTYPE dla dokumentu tworzonego w przejściowej wersji języka XHTML, znacznik html wraz z wymaganą deklaracją przestrzeni nazw oraz zamykający znacznik html

strona 54). Bez wątpienia jest to bardzo poważny błąd. Na szczęście jednak dokumenty XHTML nie są udostępniane ani traktowane jako dokumenty XML, dzięki czemu deklarację tę można, a nawet należy pominąć (deklarowanie sposobu kodowania zamieściłam na stronie 73).

- W języku HTML zarówno deklaracja DOCTYPE, jak i element html są opcjonalne (dotyczy to nawet ścisłej wersji języka HTML). XHTML wymaga natomiast podania i deklaracji DOCTYPE, i elementu html (w którym dodatkowo należy podać deklarację przestrzeni nazw). Warto zwrócić uwagę, że nie istnieje żaden element xhtml.
- W przedstawionych przykładach pokazałam, jak tworzyć dokumenty pisane w przejściowej wersji języka HTML i XHTML. Listę wszystkich najczęściej wykorzystywanych deklaracji DOCTYPE można znaleźć na WWW, na mojej witrynie (patrz strona 37) oraz na witrynie <http://ucwv.w3.org/>. Informacje, które mogą pomóc w wyborze odpowiedniej deklaracji DOCTYPE, można znaleźć na stronie 53.
- Umieszczenie na początku stron deklaracji DOCTYPE, zawierającej adres URL, zazwyczaj powoduje przejście przeglądarek do pracy w *trybie standardów*. Dzięki temu można wykorzystać kod zgodny ze standardami w celu uzyskania większej kontroli nad wyglądem tworzonych stron (patrz strona 54).
- Jeśli używasz niestandardowych znaczników XHTML, to stosowanie deklaracji DOCTYPE nie ma większego sensu. W takim przypadku, umieść zawartość strony pomiędzy znacznikami html. Nowoczesne przeglądarki wyświetlają takie strony, działając w *trybie sztuczek* (ang. *quirks mode*).
- Zastosowanie deklaracji DOCTYPE informuje narzędzia sprawdzające poprawność kodu (tak zwane *walidatory*), z jaką specyfikacją należy je porównywać (patrz strona 392).
- Warto zauważyć, że samo słowo DOCTYPE (pochodzące z *jeszcze innego* języka, określanego mianem SGML) musi być zapisywane w całości dużymi literami — zarówno w dokumentach HTML, jak i XHTML.
- Strony XHTML są tak naprawdę pisane w języku XML, zatem, z technicznego punktu widzenia, powinny zaczynać się od *deklaracji XML* w następującej postaci: `<?xml version="1.0" encoding="ISO-8859-2"?>`. Jednak zastosowanie takiej deklaracji sprawi, że przeglądarka Internet Explorer będzie działał w trybie sztuczek (patrz

Tworzenie podstawowej struktury kodu

Większość stron WWW można podzielić na dwie części: *nagłówek* oraz *treść*. W nagłówku określany jest tytuł strony, informacje o stronie (wykorzystywane przez wyszukiwarki, takie jak Google), sposób kodowania. Definiowane są też style i umieszczane skrypty. Za wyjątkiem tytułu (patrz rozdział 3.) pozostała zawartość nagłówka strony nie jest bezpośrednio widoczna.

Aby utworzyć nagłówek strony:

1. Bezpośrednio za otwierającym znacznikiem `html` (patrz rozdział 3.) wpisz `<head>`.
2. Zostaw kilka wolnych wierszy, w których, w przyszłości, zostanie zapisana zawartość nagłówka.
3. Wpisz zamykający znacznik `</head>`.

Treść dokumentu (X)HTML zawiera tę część kodu strony, która będzie wyświetlany w przeglądarce, włącznie z tekstem i rysunkami.

Aby utworzyć część treści dokumentu:

1. Po zamykającym znaczniku `</head>` wpisz `<body>`.
2. Zostaw kilka wolnych wierszy z przeznaczeniem na treść strony (którą stworzysz, wykorzystując informacje zamieszczone w dalszej części niniejszej książki).
3. Wpisz `</body>`.

Wskazówki

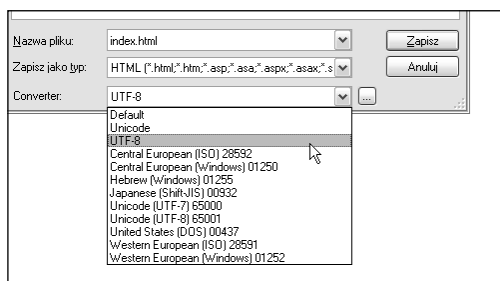
- W XHTML-u elementy `head` i `body` są wymagane. W HTML-u są one opcjonalne, jednak nawet jeśli zostaną pominięte, to przeglądarki będą działać tak, jak gdyby elementy te zostały zdefiniowane i pozwolą nam nawet na określanie ich wyglądu za pomocą stylów.
- Inną przyczyną, z jakiej warto stosować znaczniki `head` oraz `body`, jest określanie, kiedy mają być wykonywane skrypty (patrz strona 354).

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
</head>
<body>

</body>
</html>
```

Rysunek 3.3. Elementy `head` oraz `body` pozwalają nadać odpowiednią strukturę dokumentom (X)HTML



Rysunek 3.4. Tworzone pliki zapisywałam, wykorzystując sposób kodowania o nazwie UTF-8 (więcej informacji na temat sposobów kodowania można znaleźć w rozdziale 21., „Symbole oraz inne znaki nie należące do alfabetu angielskiego”)

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1
-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="content-type" content="text/html;
charset=utf-8" />
</head>
<body>

</body>
</html>
```

Rysunek 3.5. Gdy przeglądarka użytkownika odnajdzie ten znacznik meta, będzie „wiedzieć”, że strona została zapisana przy użyciu kodowania UTF-8 i wyświetli ją w odpowiedni sposób. Najważniejsze jest to, aby określenie kodowania podane w dokumencie odpowiadało sposobowi kodowania, wykorzystanemu podczas zapisywania pliku

Określanie sposobu kodowania

Wszystkie dokumenty tekstowe, nie wyłączając dokumentów (X)HTML, są zapisywane przy wykorzystaniu jakiegoś sposobu *kodowania*. Ponieważ na świecie jest wykorzystywanych aktualnie wiele różnych sposobów kodowania, bezpośrednio w kodzie dokumentu warto określić, który z nich został wykorzystany przy jego zapisywaniu. W ten sposób przeglądarki działające w systemach komputerowych o innym domyślnym systemie kodowania będą miały mniej problemów z poprawnym wyświetleniem strony.

Aby określić używany sposób kodowania:

W nagłówku dokumentu wpisz: `<meta http-equiv="content-type" content="text/html; charset=kodowanie" />`, gdzie *kodowanie* to nazwa sposobu kodowania, w jakim został zapisany dany dokument.

Sposób kodowania strony WWW zależy od sposobu, w jaki strona ta była zapisywana. Jeśli została ona zapisana w formie tekstowej, czyli bez jawnego określenia sposobu kodowania, to z dużą dozą prawdopodobieństwa można przyjąć, że został wykorzystany sposób kodowania domyślnie używany dla danego języka. Na przykład dla języka polskiego w systemie Windows domyślnie używany jest sposób kodowania `windows-1250`, dla języka angielskiego — `windows-1252`.

Wskazówki

- Chociaż z technicznego punktu widzenia specyfikacja języka nie wymaga określania sposobu kodowania, to jednak warto to zrobić.
- Jeśli podczas zapisywania pliku został wybrany określony sposób kodowania, to należy go także podać w dokumencie (X)HTML, w znaczniku meta.
- Listę zbiorów znaków, określających dostępne sposoby kodowania, można znaleźć pod adresem <http://www.w3.org/International/O-charset-lang.html>.
- XHTML wymaga, aby w dokumentach był określany sposób kodowania, jeśli jest on różny od UTF-8 lub UTF-16.
- Więcej informacji na temat kodowania można znaleźć w rozdziale 21., „Symbole oraz inne znaki nie należące do alfabetu angielskiego”.

Nadawanie tytułu

Każdy dokument (X)HTML musi zawierać element `title`. Element ten określa tytuł dokumentu, który powinien być krótki i opisowy. Większość przeglądarek wyświetla tytuł strony na pasku tytułu swojego okna (rysunek 3.7). Znacznie ważniejszy jest jednak fakt, iż tytułu używają wyszukiwarki, takie jak Google oraz Yahoo. Co więcej, jest on także wyświetlany w przeglądarkach użytkowników na liście odwiedzonych stron oraz w menu *Ulubione*.

Aby stworzyć tytuł:

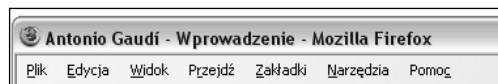
1. Ustaw kursor pomiędzy otwierającym a zamykającym znacznikiem `head` (patrz strona 72).
2. Wpisz `<title>`.
3. Wpisz tytuł strony.
4. Wpisz `</title>`.

Wskazówki

- Element `title` jest obowiązkowy.
- Tytuł nie może zawierać żadnych znaczników formatujących, obrazków ani łączy do innych stron WWW.
- Tytuł strony w bezpośredni sposób odpowiada za ocenę wyszukiwania, podawaną przez większość wyszukiwarek. Im dokładniej odpowiada on słowom podanym przez użytkownika — bez jakichkolwiek dodatkowych słów — tym bliżej początku listy wyników strona się pojawi. Poza tym tytuł strony jest wyświetlany na liście wyników (rysunek 3.8).
- Tytuły są także wyświetlane na listach — historii, ulubionych stron oraz zakładek (ang.: *bookmarks*) — rysunek 3.9.
- Jeśli w tytule znajdują się znaki specjalne, takie jak akcenty i litery narodowe, to muszą one należeć do używanego sposobu kodowania (patrz strona 73) bądź też należy je wpisać, wykorzystując symbole (patrz strona 380).

```
<head>
<meta http-equiv="content-type" content="text/html;
charset=utf-8" />
<title>Antonio Gaudi - Wprowadzenie</title>
</head>
<body>
```

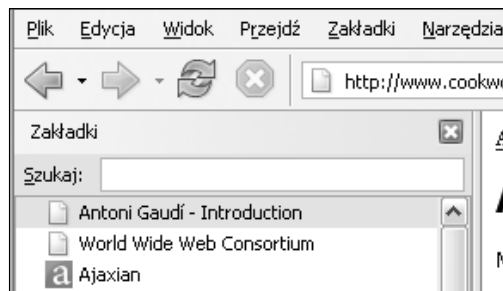
Rysunek 3.6. Element `title` powinien być umieszczony w sekcji nagłówka. Element ten jest wymagany



Rysunek 3.7. Tytuł strony jest wyświetlany na pasku tytułu okna przeglądarki



Rysunek 3.8. Prawdopodobnie najważniejsze jest to, że tytuł jest używany do opisywania stron w wyszukiwarkach, takich jak Google. Co więcej, stanowi on jeden z najważniejszych czynników podczas określania zgodności strony z zadanymi kryteriami oraz jej ostatecznego wyniku na liście wyszukiwania



Rysunek 3.9. Tytuł jest także wyświetlany w przeglądarkach na listach historii (pokazanej tutaj), ulubionych i zakładek

```

<head>
<meta http-equiv="content-type" content="text/html;
charset=utf-8" />
<title>Antonio Gaudi - Wprowadzenie</title>
</head>
<body>

<h1>Antonio Gaudi</h1>

<h2>La Casa Mila</h2>

<h2>La Sagrada Familia</h2>

</body>

```

Rysunek 3.10. Nagłówki można stosować w celu określenia struktury zawartości dokumentu



Rysunek 3.11. Nagłówki pierwszego stopnia są najczęściej wyświetlane 24-punktową, pogrubioną czcionką Times New Roman

Tworzenie nagłówków sekcji

(X)HTML daje możliwość stosowania aż sześciu różnych poziomów nagłówków, służących do dzielenia strony na fragmenty, którymi można łatwiej zarządzać.

Aby podzielić stronę przy wykorzystaniu nagłówków:

1. Wewnątrz treści dokumentu (X)HTML (po między znacznikami `body`) wpisz `<hn>`, gdzie `n` jest liczbą z zakresu od 1 do 6, określającą poziom tworzonego nagłówka.
2. Wpisz treść nagłówka.
3. Wpisz `</hn>`, gdzie `n` jest tą samą liczbą, która została użyta w kroku 1.

Wskazówki

- Traktuj nagłówki jak tytuły rozdziałów — tworzą one strukturę hierarchiczną. Używaj ich konsekwentnie.
- Jedyna oficjalna reguła odnosząca się do nagłówków nakazuje, aby nagłówek był tym wyraźniej wyświetlany na stronie, im wyższy jest jego poziom (czyli im niższa cyfra podana w znaczniku). Niemniej jednak wszystkie obecnie dostępne przeglądarki wyświetlają nagłówki tak samo — za pomocą pogrubionej czcionki Times New Roman o rozmiarach odpowiednio: 24, 18, 14, 12, 10 i 8 pikseli.
- Możesz skorzystać ze stylów, aby przypisać nagłówkom konkretną czcionkę, rozmiar, kolor itd. Więcej informacji na ten temat znajdziesz w rozdziale 10., „Formatowanie przy wykorzystaniu stylów”.
- Dodawaj do nagłówków nazwane odnośniki lub identyfikatory, aby móc tworzyć łącza, umożliwiające przejście bezpośrednio do danego nagłówka (patrz strona 124.).
- W razie potrzeby można wyrównać tekst nagłówka, dodając do znacznika `<hx>` atrybut `align= „kierunek”`, gdzie kierunek określa sposób wyrównania i może przybierać następujące wartości: `left` (do lewej), `right` (do prawej) lub `center` (wyrównany). Należy pamiętać, że atrybut `align` został odrzucony, a zamiast niego zaleca się stosowanie arkuszy stylów (patrz strona 190.).

Rozpoczynanie nowego akapitu

Język (X)HTML nie rozpoznaje znaków końca wiersza (*Return* lub *Enter*) ani dodatkowych odstępów, wprowadzonych w edytorze tekstu (patrz strona 44). Aby rozpocząć nowy akapit na stronie WWW, należy zatem wykorzystać znacznik `p`.

Aby rozpocząć nowy akapit:

1. Wpisz `<p>`.
2. Wprowadź treść akapitu.
3. Wpisz `</p>`, aby zakończyć akapit.

Wskazówki

- Zamykający znacznik `</p>` jest konieczny zarówno w języku XHTML, jak i w przypadku określania wyglądu strony przy użyciu arkuszy stylów. Dlatego radzę, aby zawsze go używać. W języku HTML znacznik ten jest opcjonalny.
- Postać akapitów można określać, wykorzystując arkusze stylów. W szczególności dotyczy to określania kroju czcionki, jej wielkości, koloru (oraz wielu innych atrybutów). Informacje na ten temat można znaleźć w rozdziale 10., „Formatowanie przy wykorzystaniu stylów”.
- Informacje na temat określania odstępów pomiędzy wierszami można znaleźć na stronie 183, „Określanie wysokości linii”. Informacje na temat określania wielkości obszaru poniżej akapitu można znaleźć na stronie 203, „Dodawanie wypełnienia wokół elementu” lub 205, „Określanie marginesów wokół elementu”.
- Jednym z szybkich i prostych (oraz zgodnych ze standardami) sposobów powiększenia odstępów pomiędzy akapitami jest wpisywanie ` ` pomiędzy kolejnymi, dodatkowymi znacznikami `p`. Lepszym rozwiązaniem jest jednak wykorzystanie arkuszy stylów (patrz strony 203 – 205).
- W razie potrzeby można wyrównać tekst akapitu, dodając do znacznika `<p>` atrybut `align="kierunek"`, gdzie kierunek określa sposób wyrównania i może przybierać następujące wartości: `left` (do lewej), `right` (do prawej) lub

```
</head>
<body>
<h1>Antonio Gaudi</h1>
<p>Wielu turystów przyjeżdża do Barcelony, aby
podziwiać niesamowitą architekturę Antonio Gaudiego. </p>
<p>W 2002 roku Barcelona obchodzi 150. rocznicę urodzin
Gaudiego.</p>
<h2>La Casa Milà</h2>
<p>Budynki projektowane przez Gaudiego są użyteczne.
La Casa Milà jest budynkiem mieszkalny, w którym żyją normalni
ludzie.</p>
<h2>La Sagrada Familia</h2>
<p>Ten niewykończony kościół pod wezwaniem Świętej Rodziny
jest najczęściej odwiedzaną budowlą w Barcelonie.</p>
</body>
```

Rysunek 3.12. *Tekst poszczególnych akapitów należy zapisywać pomiędzy otwierającym i zamykającym znacznikiem `p`. Jeśli element `p` nie zostanie zamknięty (co jest całkowicie poprawne w HTML, ale błędne w XHTML), przeglądarka może niewłaściwie zastosować style*



Rysunek 3.13. *Wielkości odstępów pomiędzy akapitami zależą od wielkości używanej czcionki*

`center` (wyśrodkuj) lub `justify` (do lewej i prawej). Należy pamiętać, że atrybut `align` został odrzucony, a zamiast niego zaleca się stosowanie arkuszy stylów (patrz strona 190).

```

</head>
<body>
<h1 id="gaudi">Antonio Gaudi</h1>
<p>Wielu turystów przyjeżdża do Barcelony, aby
podziwiać niesamowitą architekturę Antonio Gaudiego. </p>
<p>W 2002 roku Barcelona obchodzi 150. rocznicę urodzin
Gaudiego.</p>

<h2 class="building">La Casa Milà</h2>
<p>Budynki projektowane przez Gaudiego są użyteczne.
La Casa Milà jest budynkiem mieszkalny, w którym żyją normalni
ludzie.</p>

<h2 class="building">La Sagrada Família</h2>
<p>Ten niewykończony kościół pod wezwaniem Świętej Rodziny
jest najczęściej odwiedzaną budowlą w Barcelonie.</p>
</body>

```

Rysunek 3.14. Dodaj atrybut `id` do unikalnego elementu strony, aby potem móc określać jego wygląd przy użyciu stylów. Dodaj atrybut `class`, jeśli chcesz „zgrupować” elementy i za jednym razem określić wygląd całej grupy



Rysunek 3.15. Sam fakt dodania do elementu atrybutów `id` lub `class` nie zmienia w żaden sposób wyglądu tych elementów. Pełne możliwości, jakie zapewniają te atrybuty, uwidaczniają się dopiero po połączeniu ich ze stylami CSS (informacje na ten temat można znaleźć w rozdziałach 10. i 11.)

Nazywanie elementów

Elementom HTML można nadawać unikalne nazwy bądź też przypisywać je pewnym klasom. Następnie można określać wygląd elementów danej klasy przy użyciu stylów.

Aby określić unikalną nazwę elementu:

Wewnątrz otwierającego znacznika elementu wpisz `id="nazwa"`, gdzie nazwa określa w unikalny sposób dany element.

Aby określić nazwę grupy elementów:

Wewnątrz otwierającego znacznika elementu wpisz `class="nazwa"`, gdzie nazwa określa nazwę grupy elementów.

Wskazówki

- Każdy atrybut `id` w dokumencie (X)HTML musi mieć unikalną wartość. Innymi słowy, żadne dwa elementy nie mogą mieć tego samego identyfikatora.
- Do danej klasy może należeć dowolna ilość elementów — wartości atrybutu `class` różnych elementów mogą mieć te same wartości.
- Więcej informacji na temat przypisywania elementom stylów przy użyciu identyfikatora lub nazwy klasy, można znaleźć na stronie 156, „Wybieranie elementu na podstawie klasy lub identyfikatora”.
- Atrybutów `id` oraz `class` można używać w większości elementów (X)HTML, jednak są one szczególnie przydatne w elementach `div` oraz `span` (patrz strony 70 – 71).
- Atrybut `id` automatycznie zamienia element w odnośnik, do którego można tworzyć połączenia. Więcej informacji na ten temat można znaleźć na stronie 116, „Tworzenie odnośników”.
- Wreszcie, atrybut `id` może także służyć do wskazywania elementu, na którym mają operować skrypty pisane, na przykład, w języku JavaScript.

Podział strony na działy

Dzielenie strony na działy pozwala na definiowanie stylów, które będą określać wygląd całego, wybranego fragmentu strony. Rozwiązanie to jest szczególnie przydatne w przypadku określania układu strony za pomocą kaskadowych arkuszy stylów (patrz strona 195).

Aby podzielić stronę na działy:

1. Na początku działu wpisz `<div>`.
2. W razie potrzeby wpisz `id="nazwa"`, gdzie nazwa to unikalny identyfikator danego elementu `div`.
3. W razie potrzeby wpisz `class="nazwa"`, gdzie nazwa określa klasę, do której będzie należeć dany element.
4. Wpisz `>`, aby zakończyć otwierający znacznik `div`.
5. Stwórz zawartość danego działu strony.
6. Aby zakończyć dział, wpisz `</div>`.

Wskazówki

- Dział (element `div`) jest elementem blokowym, co oznacza, że jego zawartość jest automatycznie rozpoczynana w nowym wierszu.
- W istocie jedynymi znacznikami formatującymi, jakie powinny być wykorzystywane w elementach `div`, są znaczniki podziału wiersza. Wszelkie dodatkowe sposoby formatowania należy definiować za pośrednictwem arkuszy stylów, skojarzonych z klasą lub identyfikatorem danego działu, zgodnie z informacjami podanymi w rozdziałach od 7. do 14.
- Określanie nazwy klasy lub identyfikatora w elemencie `div` nie jest wymagane. Niemniej jednak zastosowanie tych atrybutów znacznie zwiększa możliwości działów.
- W jednym elemencie `div` można zdefiniować zarówno nazwę klasy, jak i unikalny identyfikator elementu, choć zapewne najczęściej stosowanym rozwiązaniem jest określenie tylko jednego z tych atrybutów. Podstawowa różnica pomiędzy nimi polega na tym, że atrybut `class` jest używany do grup, natomiast `id` do identyfikowania pojedynczych, unikalnych elementów.

```

</head>
<body>

<div id="gaudi">

<h1>Antonio Gaudi</h1>
<p>Wielu turystów przyjeżdża do Barcelony, aby
podziwiać niesamowitą architekturę Antonio Gaudiego. </p>
<p>W 2002 roku Barcelona obchodzi 150. rocznicę urodzin
Gaudiego.</p>

<div class="works">
<h2>La Casa Milà</h2>
<p>Budynki projektowane przez Gaudiego są użyteczne.
La Casa Milà jest budynkiem mieszkalny, w którym żyją normalni
ludzie.</p>
</div>

<div class="works">
<h2>La Sagrada Familia</h2>
<p>Ten niewykończony kościół pod wezwaniem Świętej Rodziny
jest najczęściej odwiedzaną budowlą w Barcelonie.</p>
</div>
</body>

```

Rysunek 3.16. Na tej stronie zdefiniowany jest jeden duży, zewnętrzny dział (rozpoczynający się przed nagłówkiem pierwszego poziomu i kończący bezpośrednio przed końcem treści dokumentu) oraz dwa działy wewnętrzne (zawierające odpowiednie nagłówki drugiego poziomu oraz towarzyszące im akapity tekstu)



Rysunek 3.17. Zazwyczaj rezultaty zdefiniowania działów będą widoczne dopiero po wykorzystaniu stylów (patrz strona 147). Dopiero wtedy dają one naprawdę wspaniałe efekty. Tę samą stronę ze stylami można zobaczyć w mojej witrynie WWW (patrz strona 37)

```

<body>
<div id="gaudi">
<h1>Antonio Gaudi</h1>
<p>Wielu turystów przyjeżdża do Barcelony, aby podziwiać niesamowitą architekturę Antonio Gaudiego. </p>
<p>W 2002 roku Barcelona obchodzi 150. rocznicę urodzin Gaudiego.</p>

<div class="works">
<h2>La Casa Milà</h2>
<p>Budynki projektowane przez Gaudiego są użyteczne. La Casa Milà jest budynkiem mieszkalny, w którym żyją <span class="emph">najczęściej</span> ludzie.</p>
</div>

<div class="works">
<h2>La Sagrada Família</h2>
<p>Ten niewykończony kościół pod wezwaniem Świętej Rodziny jest <span class="emph">najczęściej</span> odwiedzaną budowlą w Barcelonie. </p>
</div>

</div>
</body>
</html>

```

Rysunek 3.18. Znaczniki `span` służą zazwyczaj do oznaczania wybranego fragmentu zawartości wewnętrzzwerszowej, choć nie musi to być tekst. Zawartość taką można potem sformatować przy wykorzystaniu stylów (patrz rozdziały 10. i 11.)



Rysunek 3.19. Element `span` umożliwia także określanie wewnętrznej struktury dokumentu. Efekty wykorzystania tych elementów będą widoczne dopiero po przypisaniu im stylu (patrz strona 147.). Tę samą stronę ze stylami można zobaczyć na mojej witrynie WWW (patrz strona 37.)

Tworzenie obszarów wewnętrzzwerszowych

Główne fragmenty strony można organizować przy wykorzystaniu sekcji nagłówka i treści, a także przy użyciu działów, a nawet nagłówków (elementów `h1`, `h2`, itd.). Można także tworzyć *obszary wewnętrzzwerszowe*, zawierające tekst lub inne elementy wewnętrzzwerszowe, służące do ich grupowania, identyfikowania i określania wyglądu za pomocą stylów.

Aby utworzyć obszar wewnętrzzwerszowy:

1. Na samym początku, w miejscu, w którym ma się rozpocząć tworzony obszar wewnętrzzwerszowy, wpisz ``.
2. W razie potrzeby wpisz `id="nazwa"`, gdzie nazwa to unikalny identyfikator danego elementu `div`.
3. W razie potrzeby wpisz `class="nazwa"`, gdzie nazwa określa klasę, do której będzie należeć dany element.
4. Wpisz `>`, aby zakończyć otwierający znacznik `span`.
5. Wpisz zawartość tworzonego obszaru wewnętrzzwerszowego.
6. Aby zakończyć obszar, wpisz ``.

Wskazówki

- Więcej informacji na temat różnic pomiędzy elementami blokowymi oraz zawartością wewnętrzzwerszową można znaleźć na stronie 42, w sekcji „Elementy blokowe oraz wewnętrzzwerszowe”.
- Obszary wewnętrzzwerszowe nie mają żadnego charakterystycznego dla nich sposobu formatowania. Stają się one przydatne dopiero w przypadku wykorzystania ich wraz ze stylami (przy użyciu klas lub unikalnych identyfikatorów, zgodnie z informacjami podanymi w rozdziałach od 7. do 14.).
- W jednym elemencie `span` można zdefiniować zarówno nazwę klasy, jak i unikalny identyfikator elementu, choć zapewne znacznie częściej stosowanym rozwiązaniem jest podawanie tylko jednego z tych atrybutów. Podstawowa różnica pomiędzy nimi polega na tym, że atrybut `class` jest używany do grup, natomiast `id` do identyfikowania pojedynczych, unikalnych elementów.

Tworzenie nowych wierszy

Przeglądarki automatycznie przenoszą tekst do kolejnych wierszy, w zależności od szerokości elementu blokowego lub okna przeglądarki. Nowe akapity można tworzyć, wykorzystując znacznik `p` (patrz strona 76.), niemniej jednak istnieje także możliwość łamania wierszy w dowolnym miejscu.

Znacznik `br` doskonale nadaje się do przedstawiania poezji lub innych krótkich fragmentów tekstu, które powinny być wyświetlane bezpośrednio jeden poniżej drugiego, bez większych odstępów pomiędzy nimi.

Aby wstawić znacznik podziału wiersza:

Wpisz `
` w miejscu, w jakim tekst ma zostać przełamany. W przypadku tego znacznika nie jest stosowany żaden niezależny znacznik zamykający.

Wskazówki

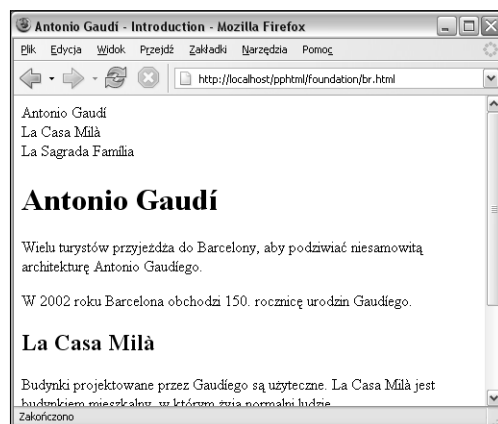
- Zamykający znak ukośnika (`/`) jest wymagany wyłącznie w XHTML, w celu spełnienia reguły wymuszającej prawidłowe zamykanie wszystkich elementów umieszczanych w dokumencie (patrz strona 50.). Należy się upewnić, że pomiędzy literami `br` oraz znakiem ukośnika jest odstęp. W dokumentach HTML znak ukośnika można pominąć, choć jego wstawienie w niczym nie przeszkadza.
- Można wstawiać kilka znaczników `br` jeden za drugim, aby stworzyć dodatkowy odstęp pomiędzy wierszami tekstu lub akapitami.
- Odległości pomiędzy kolejnymi wierszami tekstu wewnątrz akapitu oraz pomiędzy samymi akapitami można określać przy wykorzystaniu arkuszy stylów (patrz strona 183. oraz strony 203. i 205.).
- Znacznik `br` był często stosowany wraz z odrzuconym atrybutem `clear` do tworzenia tekstu otaczającego obrazek (patrz strona 115.). Rozwiązanie to zostało zastąpione atrybutem CSS `clear` (patrz strona 213.).
- Właściwość CSS `white-space` jest wspieranym narzędziem, umożliwiającym zachowanie oryginalnego sposobu zapisania dokumentu (patrz strona 189.).

```
<body>
<div id="toc">Antonio Gaudi<br />La Casa Mila<br />
La Sagrada Familia</div>
<div id="gaudi">
<h1>Antonio Gaudi</h1>
<p>Wielu turystów przyjeżdża do Barcelony, aby
podziwiać niesamowitą architekturę Antonio
Gaudiego. </p>
<p>W 2002 roku Barcelona obchodzi 150. rocznicę
urodzin Gaudiego.</p>
```

Rysunek 3.20. Na początku strony utworzyłam nowy element `div`, który może posłużyć do umieszczenia w nim spisu treści. Wewnątrz niego znajdują się trzy wiersze tekstu (utworzone przy wykorzystaniu znaczników `br`) o minimalnych odległościach pomiędzy nimi

```
<body>
<div id="toc">Antonio Gaudi
<br />La Casa Mila
<br />La Sagrada Familia</div>
<div id="gaudi">
<h1>Antonio Gaudi</h1>
<p>Wielu turystów przyjeżdża do Barcelony, aby
podziwiać niesamowitą architekturę Antonio
Gaudiego. </p>
<p>W 2002 roku Barcelona obchodzi 150. rocznicę
urodzin Gaudiego.</p>
```

Rysunek 3.21. Należy pamiętać, że znaki nowego wiersza umieszczane w kodzie źródłowym dokumentu są całkowicie ignorowane przez przeglądarkę. Ten kod jest równoważny temu z rysunku 3.20 (choć jest łatwiejszy do zrozumienia)



Rysunek 3.22. Jeśli użyjemy znacznika `br`, kolejne elementy będą wyświetlane w nowym wierszu

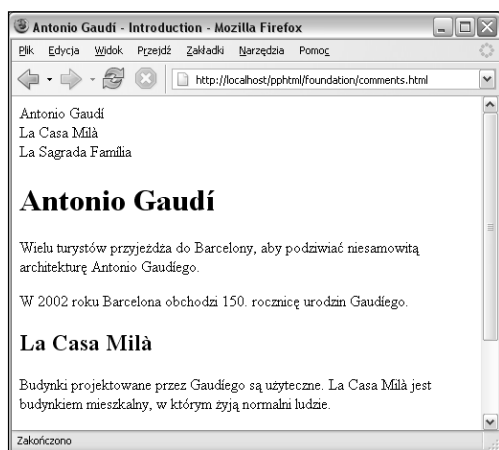
```

<body>
<!--Oto spis treści, który w normalnym dokumencie
mogłoby być o wiele dłuższy.-->
<div id="toc" title="toc">Antonio Gaudi<br />
La Casa Milà<br />La Sagrada Familia</div>

<div id="gaudi">
<h1>Antonio Gaudi</h1>
<p>Wielu turystów przyjeżdża do Barcelony, aby
podziwiać niesamowitą architekturę Antonio Gaudiego. </p>
<p>W 2002 roku Barcelona obchodzi 150. rocznicę
urodzin Gaudiego.</p>

```

Rysunek 3.23. *Komentarze to świetny sposób na wprowadzenie do swojego tekstu przypomnień. Możesz je również wykorzystać do zachowania informacji o modyfikacjach wprowadzanych w treści dokumentu*



Rysunek 3.24. *Komentarze są niewidoczne (dopóki nie wyświetli się kodu źródłowego strony — patrz strona 67.)*

Dodawanie komentarzy

W kodzie dokumentów (X)HTML można umieszczać komentarze, które będą przypominały Tobie (lub następnemu projektantowi), co chciałeś osiągnąć, wykorzystując konkretne znaczniki. Komentarze wyświetlane są jedynie w dokumencie HTML, gdy dokument taki zostanie wczytany do edytora tekstu lub edytora HTML. Są one zupełnie niewidoczne dla użytkownika.

Wpisanie komentarza na stronę WWW

1. Do swojego dokumentu HTML w miejscu, w którym chcesz wstawić komentarz, wpisz `<!--`.
2. Napisz treść komentarza.
3. Wpisz `-->`, by zakończyć komentarz.

Wskazówki

- Komentarze są szczególnie przydatne do opisywania, w jakim celu jest używany dany znacznik i jaki efekt chcemy dzięki niemu uzyskać.
- Innym przypadkiem, w którym warto stosować komentarze, to przypomnianie sobie (i ewentualnym następcom) o dodaniu, usunięciu lub uaktualnieniu poszczególnych sekcji strony.
- Obejrzyj w przeglądarce swój dokument z komentarzami, zanim go opublikujesz, aby uniknąć upublicznienia swych — być może osobistych — komentarzy.
- Miej się na baczności przed *zbyt* osobistymi komentarzami. Co prawda są one niewidoczne w przeglądarce, ale radośnie pojawią się, gdy użytkownik zachowa Twoją stronę w postaci źródłowej — jako plik HTML. Więcej informacji na temat zapisywania kodu przeglądanych stron znajdziesz w sekcji „Czerpanie inspiracji od innych” na stronie 67.
- Komentarze nie można zagnieżdżać — czyli umieszczać wewnątrz innych komentarzy.

Nadawanie nazw elementom stron

Za pomocą atrybutu `title` można tworzyć etykiety ekranowe niemal we wszystkich częściach witryny WWW. Etykiety te szczególnie przydają się przy wyświetlaniu podpowiedzi z informacją, jakie dane mają być wpisywane w poszczególnych polach formularzy, jednak można ich używać także w dowolnych innych elementach.

Aby dodać etykietę do elementu strony:

Wewnątrz znacznika (X)HTML, do którego chcesz dodać etykietę, wpisz: `title="etykieta"`, gdzie etykieta to tekst, jaki ma zostać wyświetlony w etykiecie ekranowej, gdy użytkownik umieści wskaźnik myszy w obszarze danego elementu.

Wskazówki

- Internet Explorer, przeznaczony do użycia w systemie Windows, wyświetla także etykiety ekranowe tworzone na podstawie atrybutu `alt`, definiowanego w znacznikach `img` (patrz strona 105.). W przypadku, gdy w elemencie zostaną zdefiniowane wartości obu atrybutów — `title` oraz `alt` — etykieta zostanie utworzona przy użyciu pierwszego z nich. A zatem jeśli nie chcesz, aby Internet Explorer wyświetlał zawartość atrybutu `alt` w etykiecie ekranowej, dodaj do elementu pusty atrybut `title` — `title=""`.

```
<body>
<!--Oto spis treści, który w normalnym dokumencie
mógłby być o wiele dłuższy.-->
<div id="toc" title="Spis treści">Antonio Gaudi<br />
La Casa Mila<br />La Sagrada Família</div>
<div id="gaudi">
<h1>Antonio Gaudi</h1>
<p>Wielu turystów przyjeżdża do Barcelony, aby
podziwiać niesamowitą architekturę Antonio Gaudiego. </p>
<p>W 2002 roku Barcelona obchodzi 150. rocznicę
urodzin Gaudiego.</p>

<div class="works">
<h2>La Casa Mila</h2>
<p>Budynki projektowane przez Gaudiego są użyteczne.
La Casa Mila jest budynkiem mieszkalnym, w którym żyją
<span class="emph">normalni</span> ludzie.</p>
</div>
```

Rysunek 3.25. Nazwę można nadać dowolnemu elementowi strony



Rysunek 3.26. Gdy użytkownik umieści wskaźnik myszy w obszarze takiego elementu, pojawi się etykieta ekranowa