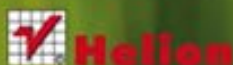


WYDANIE IV

# Projektowanie stron internetowych

PRZEWODNIK DLA POCZĄTKUJĄCYCH WEBMASTERÓW O HTML5, CSS3 I GRAFICE

Jennifer Niederst Robbins



Obojętnie, czy jesteś początkujący, czy na bieżąco podążasz za trendami, ta książka da Ci solidną porcję najważniejszych wiadomości o dzisiejszym przemyśle webowym. Każdy temat zobrazowałam wieloma ćwiczeniami, które pozwolą Ci w odpowiednim tempie wypróbować zdobyte umiejętności. Przeczytaj tę książkę i poczuj się, jakbyś siedział w mojej klasie!

— Jennifer Niederst Robbins

O'REILLY

Tytuł oryginału: Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics

Tłumaczenie: Aleksander Lamża (wstęp, rozdz. 1 – 11), Wojciech Moch (rozdz. 12 – 22, dodatki),  
z wykorzystaniem fragmentów książki „Projektowanie stron internetowych. Przewodnik dla  
początkujących webmasterów po (X)HTML, CSS i grafice” w tłumaczeniu Anny Trojan

ISBN: 978-83-246-6667-6

© 2014 Helion S.A.

Authorized Polish translation of the English edition *Learning Web Design, 4E*,  
ISBN 9781449319274

© 2012 Littlechair, Inc.

This translation is published and sold by permission of O'Reilly Media, Inc.,  
which owns or controls all rights to publish and sell the same.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means,  
electronic or mechanical, including photocopying, recording or by any information storage retrieval system,  
without permission from the Publisher.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji  
w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie  
książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi  
bądź towarowymi ich właścicieli.

Wydawnictwo HELION dołożyło wszelkich starań, by zawarte w tej książce  
informacje były kompletne i rzetelne. Nie bierze jednak żadnej odpowiedzialności  
ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw  
patentowych lub autorskich. Wydawnictwo HELION nie ponosi również  
żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania  
informacji zawartych w książce.

Wydawnictwo HELION

ul. Kościuszki 1c, 44-100 GLIWICE

tel. 32 231 22 19, 32 230 98 63

e-mail: [helion@helion.pl](mailto:helion@helion.pl)

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:

<ftp://ftp.helion.pl/przyklady/prsti2.zip>

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/prsti2>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

- Kup książkę
- Poleć książkę
- Oceń książkę

- Księgarnia internetowa
- Lubię to! » Nasza społeczność

# Spis treści

Przedmowa ..... xi

## Część I Podstawy

### Rozdział 1

Od czego zacząć? ..... 3

Od czego zacząć? ..... 4

Czym się zajmuje projektant stron internetowych? ..... 4

Jakich języków muszę się nauczyć? ..... 11

Co muszę kupić? ..... 14

Czego się nauczyłeś? ..... 19

Sprawdź się! ..... 20

### Rozdział 2

Jak działa internet? ..... 21

Internet a Web ..... 21

Dostarczanie informacji ..... 21

Kilka słów o przeglądarkach ..... 23

Adres strony internetowej (URL) ..... 24

Anatomia strony internetowej ..... 26

Składanie wszystkiego w całość ..... 30

Sprawdź się! ..... 32

### Rozdział 3

Kilka ważnych spraw, o których musisz wiedzieć .... 33

Od przybytku głowa boli ..... 34

Trzymanie się standardów ..... 36

Stopniowe ulepszanie ..... 36

Technika Responsive Web Design ..... 38

Web dla wszystkich, czyli dostępność .....	41
Need for Speed — liczy się szybkość .....	43
Sprawdź się! .....	45

## Część II Znaczniki HTML i struktura dokumentu

### Rozdział 4

Tworzenie prostej strony (przegląd języka HTML) ...	49
Strona internetowa krok po kroku .....	49
Przed rozpoczęciem należy uruchomić edytor tekstu .....	50
Krok 1. Zaczynamy od zawartości .....	53
Krok 2. Nadajemy dokumentowi strukturę .....	55
Krok 3. Oznaczamy elementy tekstowe .....	58
Krok 4. Wstawiamy obrazek .....	61
Krok 5. Zmieniamy wygląd za pomocą arkusza stylów .....	64
Kiedy dobre strony nie działają dobrze .....	65
Walidacja dokumentów .....	66
Sprawdź się! .....	67
Przegląd elementów HTML tworzących strukturę dokumentu .....	68

### Rozdział 5

Znaczniki tekstowe .....	69
Akapity .....	70
Nagłówki .....	70
Listy .....	73
Inne elementy stosowane do treści .....	76
Struktura zawartości strony .....	79
Przegląd elementów liniowych .....	84
Ogólne elementy div oraz span .....	94
Niektóre znaki specjalne .....	99
Zestawienie wszystkiego razem .....	100
Sprawdź się! .....	102
Przegląd elementów tekstowych .....	104

### Rozdział 6

Hiperłącza .....	105
Atrybut href .....	106
Tworzenie odsyłaczy do stron internetowych .....	107
Tworzenie odsyłaczy do stron z własnej witryny .....	108
Otwieranie stron docelowych w nowym oknie przeglądarki .....	118
Odsyłacze „pocztowe” .....	119

Odsyłacze „telefoniczne”	120
Sprawdź się!	120
Przegląd elementów — odsyłacze	122

## Rozdział 7

### Grafika ..... 123

Kilka słów o formatach obrazków	123
Element img	124
Okno w oknie	130
Sprawdź się!	131
Przegląd elementów — obrazki	132

## Rozdział 8

### Tabele ..... 133

Jak używać tabel	133
Podstawowa struktura tabeli	135
Nagłówki tabel	138
Tworzenie zakresów komórek	139
Dostępność tabel	142
Podsumowanie zagadnień związanych z tabelami	144
Sprawdź się!	146
Przegląd elementów HTML — tabele	146

## Rozdział 9

### Formularze ..... 147

Jak działają formularze	147
Element form	149
Zmienne oraz zawartość	151
Wielkie podsumowanie kontrolek	152
Dostępność formularzy	171
Projekt i układ formularza	173
Sprawdź się!	175
Formularze — przegląd elementów	176

## Rozdział 10

### Co nowego w HTML5? ..... 181

Co się przydarzyło w drodze do XHTML 2?	182
W świecie znaczników	185
Interfejsy API	189
Elementy video i audio	192
Element canvas	198

Podsumowanie .....	202
Sprawdź się! .....	203

### Część III CSS i prezentacja dokumentu

#### Rozdział 11

Kaskadowe arkusze stylów .....	207
Zalety CSS-a .....	207
Jak działają arkusze stylów .....	209
Najważniejsze koncepcje .....	214
Dalsza nauka CSS-a .....	221
Sprawdź się! .....	223

#### Rozdział 12

Formatowanie tekstu .....	225
Właściwości czcionek .....	225
Zmiana koloru tekstu .....	243
Więcej typów selektorów .....	244
Zmiana stylu wiersza tekstu .....	249
Podkreślenia oraz inne „dekoracje” .....	252
Zmiana wielkości liter .....	252
Odstępy .....	253
Cienie pod tekstem .....	254
Zmiany wypunktowania i numeracji list .....	259
Sprawdź się! .....	261
Przegląd CSS-a — właściwości dotyczące czcionki oraz tekstu .....	263

#### Rozdział 13

Kolory i tła .....	265
Określanie wartości koloru .....	265
Kolor pierwszego planu .....	272
Kolor tła .....	273
Zabawy z przezroczystością .....	275
Wprowadzenie do selektorów pseudoklas .....	276
Selektory pseudoelementów .....	279
Selektory atrybutów .....	281
Obrazki tła .....	284
Skrótowa właściwość background .....	293
Prawie tęcza (gradienty) .....	296
I wreszcie — zewnętrzne arkusze stylów .....	300
Sprawdź się! .....	303
Przegląd CSS-a — właściwości dotyczące koloru oraz tła .....	304

## Rozdział 14

<b>Model pojemnika</b>	<b>305</b>
Pojemnik elementu	305
Określanie wymiarów zawartości elementu	306
Dopełnienie	312
Obramowanie	316
Marginesy	328
Przypisywanie ról wyświetlania	333
Dodawanie cienia do elementów	335
Sprawdź się!	336
Przegląd CSS-a — podstawowe właściwości modelu pojemnika	338

## Rozdział 15

<b>Pływanie oraz pozycjonowanie</b>	<b>341</b>
Normalny układ dokumentu	341
Pływanie	342
Podstawy pozycjonowania	356
Pozycjonowanie względne	358
Pozycjonowanie bezwzględne	359
Pozycjonowanie sztywne	368
Sprawdź się!	370
Przegląd CSS-a — właściwości dotyczące pływania i pozycjonowania	371

## Rozdział 16

<b>Układ strony i CSS</b>	<b>373</b>
Strategie związane z układem strony	373
Szablony stron internetowych	380
Wielokolumnowe układy strony	
z wykorzystaniem elementów pływających	381
Układ pozycjonowany	392
Tła wypełniające kolumny od góry do dołu	395
Sprawdź się!	398

## Rozdział 17

<b>Przekształcenia, transformacje i animacje</b>	<b>399</b>
Przejścia CSS	399
Transformacje CSS	410
Animacja klatkowa	420
Sprawdź się!	423
Przegląd właściwości CSS-a — przejścia, transformacje i animacje	426

Rozdział 18

Techniki CSS427

Czysta strona (zerowanie stylów CSS)427

Techniki zastępowania tekstu obrazkiem429

Technika CSS Sprites430

Nadawanie stylów formularzom434

Nadawanie stylów tabelom441

Podstawy techniki Responsive Web Design444

Podsumowanie arkuszy stylów454

Sprawdź się!454

Przegląd CSS-a — właściwości dotyczące tabel oraz list456

Część IVJavaScript

Rozdział 19

Wprowadzenie do JavaScriptu459

Czym jest JavaScript?459

Dodawanie skryptów na stronę463

Anatomia skryptu463

Obiekt przeglądarki478

Zdarzenia478

Podsumowanie481

Sprawdź się!483

Rozdział 20

Korzystanie z JavaScriptu485

Poznaj model DOM485

Wypełniacze493

Javascriptowe biblioteki497

Wielkie zakończenie501

Sprawdź się!502

Część VTworzenie grafiki stron internetowych

Rozdział 21

Podstawy grafiki stron internetowych507

Źródła obrazków507

Poznaj formaty grafiki510

Rozmiar oraz rozdzielczość obrazka522

Praca z przezroczystością526



---

Wprowadzenie do formatu SVG .....	533
Podsumowanie informacji dotyczących grafiki stron internetowych ...	538
Sprawdź się! .....	539

**Rozdział 22**

<b>Optymalizacja grafiki stron internetowych .....</b>	<b>541</b>
Uniwersalne strategie optymalizacyjne .....	542
Optymalizacja plików GIF .....	543
Optymalizacja plików JPEG .....	547
Optymalizacja plików PNG .....	552
Optymalizacja pod kątem docelowego rozmiaru .....	553
Przegląd optymalizacji .....	554
Sprawdź się! .....	555

**Dodatek A**

<b>Odpowiedzi do ćwiczeń .....</b>	<b>557</b>
------------------------------------	------------

**Dodatek B**

<b>Selektory CSS3 .....</b>	<b>583</b>
-----------------------------	------------

<b>Skorowidz .....</b>	<b>587</b>
------------------------	------------



# Tworzenie prostej strony

## (przegląd języka HTML)

W pierwszej **części** książki został przedstawiony ogólny przegląd środowiska pracy projektantów stron internetowych. Po tym wprowadzeniu czas zakasać rękawy i przejść do tworzenia prawdziwej strony internetowej. Będzie ona prosta, jednak nawet najbardziej skomplikowane strony są oparte na opisanych tu zasadach.

W tym rozdziale utworzymy prostą stronę internetową, byś mógł się przekonać, jak w praktyce stosuje się znaczniki HTML. Przedstawione ćwiczenia pozwolą na dalszą samodzielną pracę.

Dzięki lekturze tego rozdziału:

- zrozumiesz, jak działają znaczniki HTML, w tym elementy oraz atrybuty,
- zobaczysz, w jaki sposób przeglądarki interpretują dokumenty HTML,
- poznasz podstawową strukturę dokumentu HTML,
- przyjrzyj się arkuszom stylów.

Na razie nie musisz się przejmować nauką poszczególnych elementów czy reguł arkuszy stylów, będzie na to czas w kolejnych rozdziałach. Na razie przyjrzyj się samemu procesowi, ogólnej strukturze dokumentu oraz nowej terminologii.

## Strona internetowa krok po kroku

Już w **rozdziale 2**, „Jak działa internet?”, mogłeś się przyjrzeć dokumentowi HTML. Teraz jednak dokument ten utworzysz samodzielnie i będziesz się nim bawić w przeglądanie. Cały proces został podzielony na pięć etapów ilustrujących podstawy tworzenia strony internetowej.

**Krok 1. Zaczynamy od zawartości.** Zaczniemy od wpisania do dokumentu zwykłego tekstu i sprawdzimy, co z nim zrobi przeglądarka.

**Krok 2. Nadajemy dokumentowi strukturę.** Zapoznamy się ze składnią elementów HTML oraz sposobem nadawania dokumentowi struktury.

**Krok 3. Oznaczamy elementy tekstowe.** Treść strony zostanie opisana za pomocą odpowiednich elementów tekstowych. Przy okazji dowiesz się, jak poprawnie używać HTML-a.

**Krok 4. Wstawiamy obrazki.** Dodając obrazek do strony, poznasz atrybuty oraz puste elementy.

### W TYM ROZDZIALE:

- Wprowadzenie do elementów oraz atrybutów
- Wstawianie znaczników w prostym dokumencie
- Elementy tworzące strukturę dokumentu
- Prosty arkusz stylów
- Rozwiązywanie problemów z niedziałającymi stronami internetowymi

## Praktyczna nauka języka HTML

Zdecydowanie najlepiej uczyć się języka HTML w starszowiecki sposób — poprzez *ręczne pisanie kodu*. Dzięki wpisywaniu znaczników jeden po drugim i obserwowaniu efektów tych zmian w przeglądarce masz największe szanse na zrozumienie zasad rządzących językiem. Wbrew pozorom nauczenie się poprawnego tworzenia dokumentów w HTML-u nie zabiera zbyt wiele czasu.

Zrozumienie języka HTML pomoże Ci sprawniej i wydajniej korzystać z narzędzi przeznaczonych do tworzenia stron internetowych. Poza tym pomyśl o satysfakcji, którą odczujesz, gdy spojrzysz na plik źródłowy i będziesz wiedzieć, o co w nim chodzi. Znajomość HTML-a jest także niezbędna do rozwiązywania problemów z niedziałającymi stronami internetowymi lub ulepszania domyślnego formatowania tworzonego przez różne narzędzia.

Musisz też wziąć pod uwagę fakt, że profesjonalści w zdecydowanej większości przypadków wpisują ręcznie kod, ponieważ daje im to największą kontrolę nad projektem i możliwość wyboru elementów, z których korzystają.

**Krok 5. Zmieniamy wygląd strony za pomocą arkusza stylów.** Dzięki temu ćwiczeniu dowiesz się, jak sformatować treść strony z wykorzystaniem kaskadowych arkuszy stylów (CSS).

Pod koniec rozdziału będziesz miał gotowy dokument źródłowy strony przedstawionej na [rysunku 4.1](#). Nie jest ona jakoś szczególnie rozbudowana, ale od czegoś trzeba zacząć.

W czasie pracy nad stroną będziesz bardzo często sprawdzać w przeglądarce wynik wprowadzonych zmian — prawdopodobnie częściej, niż robi się to zazwyczaj — ale ponieważ jest to wprowadzenie do HTML-a, dobrze będzie przyjrzeć się skutkowi każdej niewielkiej zmiany pliku źródłowego.

## Przed rozpoczęciem należy uruchomić edytor tekstu

Zarówno w tym rozdziale, jak i w całej książce dokumenty HTML będziemy tworzyć ręcznie, dlatego musisz zacząć od uruchomienia edytora tekstu. Edytor tekstu dostarczany z systemem operacyjnym, taki jak Notatnik (Windows) czy TextEdit (Mac OS X), powinien w zupełności wystarczyć<sup>1</sup>. Do naszych celów nadaje się każdy edytor tekstu, w którym można zapisywać zwykłe pliki tekstowe z rozszerzeniem *.html*. Jeśli do tworzenia stron internetowych używasz narzędzia typu WYSIWYG, takiego jak Dreamweaver, na razie zostaw je w spokoju. Najpierw musisz nabrać wprawy w samodzielnym tworzeniu dokumentów HTML (patrz [ramka „Praktyczna nauka języka HTML”](#)).

Z tego podrozdziału dowiesz się, jak otworzyć nowe dokumenty w programach Notatnik oraz TextEdit. Nawet jeśli korzystałeś z nich wcześniej, warto rzucić okiem na ten tekst pod kątem specjalnych ustawień, które sprawiają, że ćwiczenia pójdą gładko.

Rozpoczynamy od Notatnika; posiadacze komputerów Macintosh powinni przejść dalej.



**Rysunek 4.1.** W tym rozdziale krok po kroku napiszemy dokument źródłowy dla powyższej strony internetowej

1 Notatnik rzeczywiście umożliwia wprowadzanie tekstu, ale na tym jego funkcjonalność się kończy, więc przydatność tego narzędzia podczas pracy z kodem jest dyskusyjna. Zdecydowanie lepszym rozwiązaniem jest zainstalowanie edytora przeznaczonego dla programistów. Jedną z ważniejszych zalet takiego rozwiązania — zwłaszcza podczas nauki języka — jest funkcja kolorowania kodu, dzięki której dużo łatwiej wychwycić przypadkowe błędy. Dostępnych jest wiele bezpłatnych edytorów tego typu, z czego najbardziej popularne dla systemu Windows to: Notepad2, Notepad++, Bluefish czy PSPad. W systemie Mac OS jednym z lepszych edytorów jest TextMate — *przyp. tłum.*

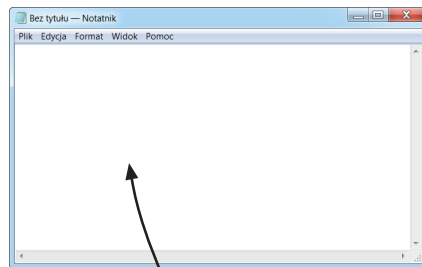
## Tworzenie nowego dokumentu w Notatniku (Windows)

Poniżej znajdują się kroki konieczne do utworzenia nowego dokumentu w programie Notatnik w systemie Windows 7 (rysunek 4.2).

1. Otwórz menu *Start* i wybierz pozycję *Notatnik* (w menu *Akcesoria*) ①.
2. Zostanie otwarte okno nowego dokumentu, w którym można rozpocząć wpisywanie tekstu ②.
3. Teraz zmienimy ustawienia, by były widoczne rozszerzenia nazw plików. Nie musisz tego robić, by tworzyć dokumenty HTML, jednak dzięki temu już na pierwszy rzut oka będziesz wiedzieć, z jakim plikiem masz do czynienia. W dowolnym oknie eksploratora z menu *Narzędzia* ③ wybierz *Opcje folderów*, a następnie przejdź na kartę *Widok* ④. Odszukaj opcję *Ukryj rozszerzenia znanych typów plików* i ją wyłącz ⑤. Kliknij przycisk *OK*, aby zapisać ustawienia. Rozszerzenia nazw plików będą już widoczne.

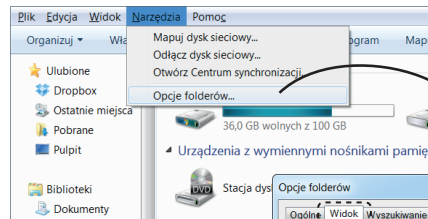
### UWAGA

*Aby w systemie Windows 7 wyświetlić menu i uzyskać dostęp do menu *Narzędzia*, wciśnij klawisz *Alt*. W systemie Windows Vista odpowiednikiem pozycji *Opcje folderów* jest *Opcje folderów i opcje wyszukiwania*.*



- ② Kliknięcie pozycji Notatnik spowoduje otwarcie nowego dokumentu

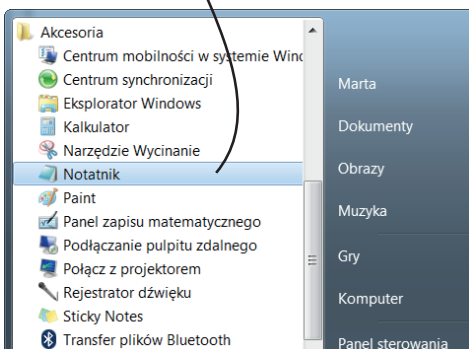
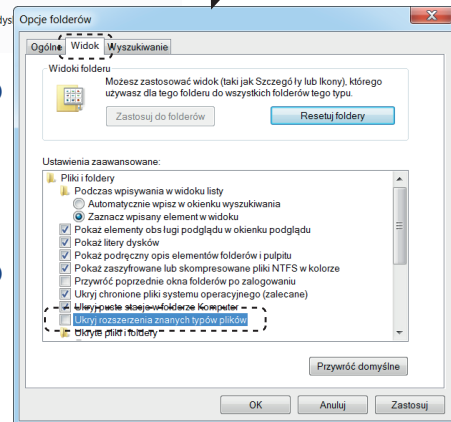
- ③ Aby wyłączyć wyświetlanie rozszerzeń nazw plików, musisz w oknie eksploratora przejść do menu *Narzędzia/Opcje folderów*



- ④ Przejdź na kartę *Widok*

⑤

- Odszukaj opcję *Ukryj rozszerzenia znanych typów plików* i ją wyłącz, a następnie kliknij przycisk *OK*, aby zapisać zmiany ustawień



- ① Otwórz menu *Start* i przejdź do Notatnika (Wszystkie programy/Akcesoria/Notatnik)

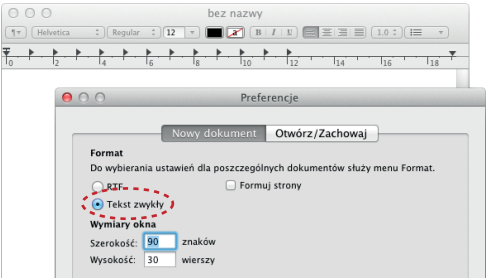
Rysunek 4.2. Tworzenie nowego dokumentu w Notatniku

## Tworzenie nowego dokumentu w programie TextEdit (Mac OS X)

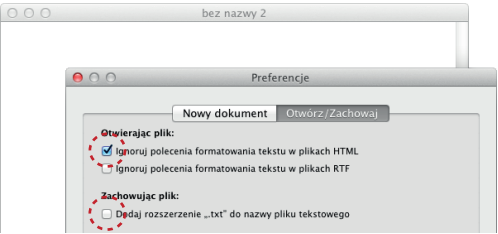
Domyślnie program TextEdit tworzy dokumenty w formacie tzw. bogatego tekstu (ang. *rich text*), czyli jako dokumenty z ukrytymi instrukcjami formatowania stylu pozwalającymi na przykład na pogrubianie tekstu czy ustalanie rozmiaru czcionki. Łatwo poznać, kiedy TextEdit jest w tym trybie, ponieważ na górze okna jest widoczny pasek narzędzi formatujących (nie ma go w trybie zwykłego tekstu). Dokumenty HTML muszą być zwykłymi dokumentami tekstowymi, dlatego należy zmienić format, jak to zostało przedstawione na [rysunku 4.3](#).

1. Skorzystaj z Findera, by w folderze *Programy* znaleźć program TextEdit. Następnie dwukrotnie kliknij jego nazwę lub ikonę, aby go uruchomić.
2. TextEdit otwiera nowy dokument. Na górze ekranu znajduje się pasek narzędzi formatujących, więc TextEdit jest w trybie bogatego tekstu. Poniżej znajdują się instrukcje zmiany trybu.
3. Otwórz okno *Preferencje* z menu *TextEdit*.
4. Musisz zmienić trzy ustawienia:  
Na karcie *Nowy dokument* zaznacz pole *Tekst zwykły* (*Plain text*).  
Na karcie *Otwórz/Zachowaj* (*Open and Save*) zaznacz pole *Ignoruj polecenia formatowania tekstu w plikach HTML* (*Ignore rich text commands in HTML files*) i wyłącz pole *Dodaj rozszerzenie „.txt” do nazwy pliku tekstowego* (*Append „.txt” extensions to plain text files*).
5. Po wprowadzeniu zmian kliknij czerwony przycisk znajdujący się w lewym górnym rogu.
6. Po utworzeniu nowego dokumentu zniknie pasek narzędzi formatujących, więc będziesz mógł zapisać dokument jako plik *.html*. Jeśli w przyszłości będziesz chciał przywrócić poprzedni tryb pracy edytora, wykonaj podobne czynności.

Widoczny pasek narzędzi formatujących wskazuje na tryb bogatego tekstu



W edytorze w trybie zwykłego tekstu ten pasek nie jest wyświetlany



Rysunek 4.3. Program TextEdit i okno ustawień z menu Preferencje

## Krok 1. Zaczynamy od zawartości

Kiedy już mamy nowy dokument, czas umieścić w nim jakąś treść. Tworzenie stron internetowych zawsze zaczynamy od ich zawartości, dlatego w naszym projekcie postąpimy tak samo. W [ćwiczeniu 4.1](#) pokazano, w jaki sposób wpisuje się zwykły tekst i zapisuje dokument w nowym folderze.

### Ćwiczenie 4.1.

#### Wprowadzanie treści strony

1. W nowym dokumencie otwartym w edytorze wpisz poniższą treść. Wprowadź ją w dokładnie takiej samej postaci, zachowując podział wierszy. Tekst ten jest dostępny w materiałach towarzyszących książce, które można pobrać ze strony <http://ftp.helion.pl/przyklady/prsti2.zip>.

Bistro "Pod Czarną Gąską"

Restauracja

Bistro "Pod Czarną Gąską" oferuje wybór popularnych dań obiadowych, którymi możesz się delektować w przyjaznej atmosferze. Menu jest często zmieniane, by zawsze serwować dania ze świeżych produktów.

Usługi cateringowe

Kiedy Ty będziesz się świetnie bawić, my będziemy dla Ciebie gotować. Catering "Pod Czarną Gąską" sprostą wszystkim wyzwaniom: czy będą to przekąski, spotkania klubowe, czy wystawne bankiety.

Lokalizacja i godziny otwarcia

Kuchary, ul. Smaczna 13

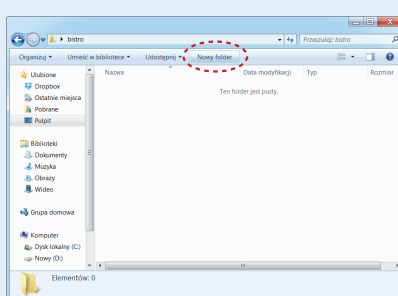
Od poniedziałku do czwartku w godzinach od 11 do 21, w piątki i soboty od 11 do 24

2. W menu *Plik* wybierz polecenie *Zapisz* lub *Zapisz jako*, aby zostało wyświetlone okno dialogowe *Zapisz jako* ([rysunek 4.4](#)). Zanim zapiszesz plik, musisz utworzyć nowy folder, który będzie zawierał wszystkie pliki tej strony (innymi słowy, **lokalny folder główny**).

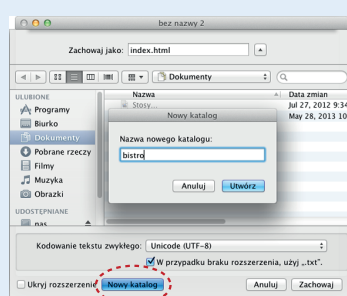
Windows: kliknij przycisk *Nowy folder* znajdujący się na górze okna.

Macintosh: kliknij przycisk *Nowy katalog*.

Windows 7



Mac OS X



**Rysunek 4.4.** Zapisywanie pliku *index.html* w nowym folderze o nazwie *bistro*

## Konwencje dotyczące nazewnictwa

Przy nazywaniu plików należy przestrzegać poniższych reguł oraz konwencji:

### Należy używać właściwych rozszerzeń nazw

**plików.** Pliki HTML i XHTML muszą mieć rozszerzenie *.html*. Pliki graficzne należy oznaczać zgodnie z ich formatem — *.gif*, *.png* lub *.jpg* (dopuszczalne jest również *.jpeg*).

### W nazwach plików nie należy używać spacji.

Często w celu wizualnego oddzielenia słów w nazwach plików używa się znaków podkreślenia ( ) lub łącznika (–), jak w *robbins\_bio.html* czy *robbins-bio.html*.

### Należy unikać znaków specjalnych, takich jak

?, %, #, /, :, ;, • i tym podobnych. Nazwy plików powinny zawierać litery, cyfry, znaki podkreślenia, łączniki oraz kropki.

### Wielkość liter w nazwach plików może

**mieć znaczenie** w zależności od konfiguracji serwera. Konsekwentne używanie w nazwach plików tylko małych liter — choć nie jest to konieczne — pozwala na łatwiejsze zarządzanie plikami i ich nazwami.

### Nazwy plików powinny być krótkie.

Dzięki krótkim nazwom łatwiej zapanować nad plikami. Jeśli plik naprawdę musi mieć długą, składającą się z kilku słów nazwę, poszczególne słowa można oddzielać za pomocą łączników, jak w *długi-tytuł-pliku.html*, dzięki czemu takie nazwy będą czytelne.

**Własne konwencje.** W dużych projektach warto wypracować własny, spójny system nazywania plików. Możesz na przykład założyć, że będziesz stosować tylko małe litery, a słowa będziesz oddzielać łącznikami. Eliminujesz w ten sposób sytuację, w której musisz zgadywać, jak kiedyś nazwałeś jakiś plik, do którego chcesz teraz utworzyć odsyłacz.



### Co ignorują przeglądarki?

Niektóre informacje zawarte w kodzie źródłowym dokumentu są ignorowane podczas wyświetlania strony.

**Powtarzające się spacje.** Kiedy przeglądarka napotka więcej niż jedną spację pod rząd, wyświetla pojedynczą spację. Jeśli zatem dokument zawiera tekst:

dawno,      dawno      temu  
przeglądarka wyświetli:  
dawno, dawno temu.

**Podziały wierszy (powroty karetki).** Przeglądarki zamieniają znaki podziału wiersza (powrotu karetki) na spację. Zgodnie z wcześniejszą zasadą (dotyczącą powtarzających się spacji) podziały wiersza umieszczone w dokumencie źródłowym nie mają wpływu na sposób wyświetlania w przeglądarce. Tekst oraz inne elementy będą zawijane, dopóki w tekście dokumentu nie zostanie napotkany nowy element blokowy, taki jak nagłówek (**h1**), akapit (**p**) lub podział wiersza (**br**).

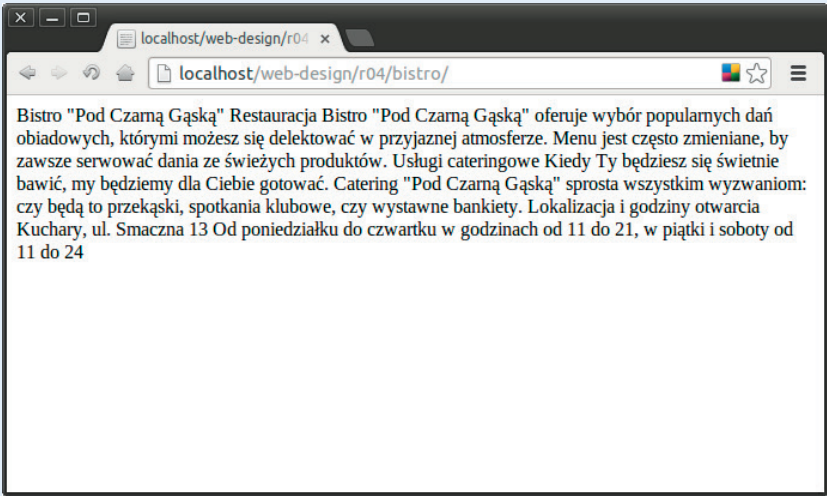
**Tabulatory.** Znaki tabulacji również są zamieniane na spację. Jaki stąd wniosek? Są bezużyteczne.

**Nierozpoznawane znaczniki.** Przeglądarka po prostu ignoruje wszystkie znaczniki, których nie rozumie lub które zostały zapisane niepoprawnie. W zależności od elementu oraz przeglądarki może to mieć różne skutki. Przeglądarka może nie wyświetlić nic lub może wyświetlić zawartość znacznika tak, jakby była ona normalnym tekstem.

**Tekst znajdujący się w komentarzach.** Przeglądarki nie wyświetlają tekstu znajdującego się pomiędzy specjalnymi znacznikami `<!--` oraz `-->`, służącymi do oznaczania komentarzy. Więcej informacji na ten temat znajduje się w ramce „Wstawianie komentarzy” w dalszej części rozdziału.

Nadaj folderowi nazwę *bistro*, a następnie zapisz w nim plik tekstowy *index.html*. Użytkownicy systemu Windows będą dodatkowo musieli wybrać pozycję *Wszystkie pliki* w polu *Zapisz jako typ*, żeby Notatnik nie dodał rozszerzenia *.txt* do wybranej przez nas nazwy pliku. Nazwa pliku musi się kończyć rozszerzeniem *.html*, by przeglądarka rozpoznała ten plik jako dokument HTML. Więcej informacji na temat nazw plików znajduje się w ramce „Konwencje dotyczące nazewnictwa”. Uwaga! Jako sposób kodowania wszystkich plików w książce przyjęto standard UTF-8, dlatego przy zapisywaniu plików należy również zamiast domyślnego ANSI wybrać z listy UTF-8.

3. Spróbuj obejrzeć plik *index.html* w przeglądarce. Uruchom ulubioną przeglądarkę i z menu *Plik* wybierz polecenie *Otwórz* lub *Otwórz plik*. Odszukaj plik *index.html* i go otwórz. Powinieneś zobaczyć coś podobnego do strony zaprezentowanej na rysunku 4.5.



Rysunek 4.5. Wygląd pierwszej wersji strony internetowej w przeglądarce

## Zrozumienie pierwszego kroku

Treść strony nie prezentuje się zbyt ciekawie (rysunek 4.5). Tekst napisany jest jednym ciągiem — nie tak to wyglądało w oryginalnym dokumencie. Można z tego wyciągnąć kilka wniosków. Pierwszy z nich — co jest widoczne na pierwszy rzut oka — jest taki, że przeglądarka ignoruje podział wierszy w dokumencie źródłowym (w ramce „Co ignorują przeglądarki?” wymieniono inne informacje zawarte w kodzie źródłowym, które nie są wyświetlane w oknie przeglądarki).

Po drugie, widać, że samo wpisanie treści strony i nazwanie dokumentu *.html* nie wystarczy. Choć przeglądarka może wyświetlić tekst z takiego pliku, nie oznaczyliśmy w żaden sposób *struktury* jego treści, a właśnie do tego służy HTML. Dodamy więc znaczniki, które zdefiniują strukturę dokumentu — najpierw w samym dokumencie HTML (krok 2.), a później do zawartości strony (krok 3.). Kiedy przeglądarka będzie znała strukturę zawartości, będzie w stanie wyświetlić stronę w pożądanym sposób.



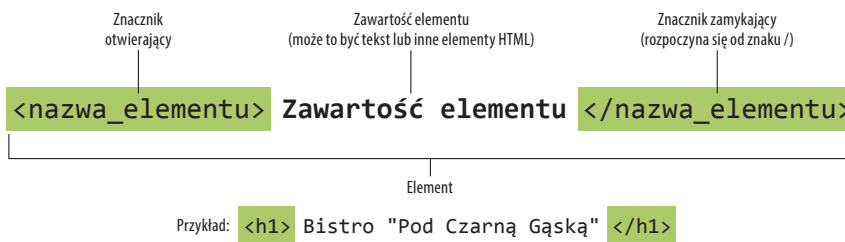
## Krok 2. Nadajemy dokumentowi strukturę

Zawartość strony zapisana jest w dokumencie `.html` — teraz czas dodać do niego znaczniki.

*Element składa się zarówno z zawartości, jak i obejmującego ją znacznika.*

### Wprowadzenie do elementów HTML

W rozdziale 2., „Jak działa internet?”, zostały pokazane przykłady elementów HTML ze znacznikami otwierającymi (jak na przykład `<p>` dla akapitu) oraz zamykającymi (jak `</p>`). Przed rozpoczęciem wstawiania znaczników do dokumentu warto się przyjrzeć strukturze elementu HTML i ustalić najważniejsze pojęcia. Ogólna budowa znacznika została przedstawiona na rysunku 4.6.



Rysunek 4.6. Elementy znacznika HTML

Poszczególne elementy strony są opisywane znacznikami znajdującymi się w kodzie źródłowym. Znacznik składa się z nazwy elementu (zazwyczaj będącej skrótem dłuższej nazwy opisowej w języku angielskim) znajdującej się w nawiasach ostrych (`<` `>`). Przeglądarka wie, że tekst znajdujący się pomiędzy takimi nawiasami nie może być wyświetlany w oknie przeglądarki.

Nazwa elementu pojawia się w **znaczniku otwierającym** (ang. *opening tag*, nazywanym również **znacznikiem początkowym**, ang. *start tag*) i ponownie w **znaczniku zamykającym** (ang. *closing tag*, nazywanym też **znacznikiem końcowym**, ang. *end tag*) poprzedzonym znakiem prawego ukośnika (ang. *slash*, `/`). Znacznik zamykający działa trochę jak „wyłącznik” dla elementu. Należy uważać, by w znacznikach nie użyć przez przypadek znaku lewego ukośnika (ang. *backslash*, `\`). Więcej na ten temat znajduje się w ramce „Ukośnik prawy a lewy”.

Znaczniki umieszczane wokół zawartości nazywa się w języku angielskim **markup** (stąd HTML — *HyperText Markup Language*). Należy zapamiętać, że **element** składa się zarówno ze swojej zawartości, jak i znaczników początkowych oraz końcowych. Nie wszystkie elementy posiadają jednak jakąś zawartość. Niektóre są z definicji **puste**, jak na przykład element `img` wykorzystywany do dodawania obrazków do strony. Elementy puste zostaną omówione w dalszej części rozdziału.

I jeszcze jedna sprawa — wielkość liter. W języku HTML wielkość liter stosowanych w znacznikach nie ma znaczenia, więc `<img>`, `<IMG>` i `<ImG>` są poprawne i oznaczają to samo. Z kolei w języku XHTML (który pod względem składni jest dużo bardziej wymagający) wszystkie znaczniki muszą być zapisane małymi literami. Wielu twórców stron przyjęło tę konwencję i warto się jej trzymać (w książce tak właśnie jest).

#### WSKAZÓWKA

#### Ukośnik prawy a lewy

W znacznikach HTML oraz adresach URL jest stosowany znak prawego ukośnika (`/`). Znak ten można znaleźć pod znakiem zapytania (`?`) na standardowej klawiaturze QWERTY.

Łatwo jest pomylić ten znak z lewym ukośnikiem (`\`), który na klawiaturze znajduje się pod znakiem `|`. Ukośnik lewy nie działa w znacznikach oraz adresach URL, dlatego należy pamiętać, by go tam nie używać.



## Podstawowa struktura dokumentu

Na [rysunku 4.7](#) została przedstawiona sugerowana podstawowa struktura dokumentu HTML5. Dlaczego „sugerowana”? Ponieważ jedynym *wymaganym* elementem jest **title**. Lepiej jednak od samego początku tworzyć dokumenty, które są w pełni poprawne. Gdybyś pisał w języku XHTML, wszystkie przedstawione elementy (z wyjątkiem *meta*) byłyby wymagane. Przyjrzyj się dokładniej [rysunkowi 4.7](#).

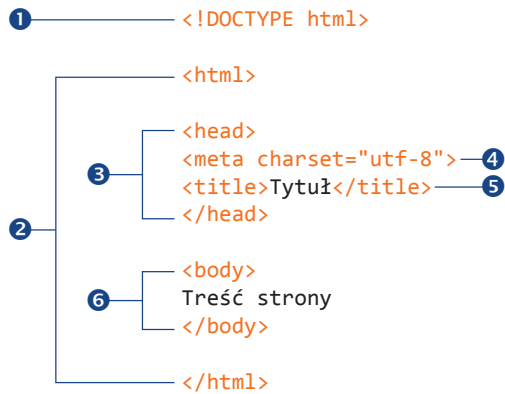
- 1 Tak się akurat składa, że w pierwszym wierszu kodu nie ma żadnego elementu. Jest to **deklaracja typu dokumentu** (znana także jako **deklaracja DOCTYPE**), która wskazuje, że jest to dokument HTML5. Temat ten jest szerzej omówiony w [rozdziale 10](#), „Co nowego w HTML5?”. W tej chwili wystarczy zapamiętać, że dzięki tej deklaracji nowoczesne przeglądarki będą wiedzieć, że mają do czynienia z dokumentem napisanym zgodnie ze specyfikacją HTML5.
- 2 Cały dokument znajduje się wewnątrz elementu **html**. Jest on nazywany **głównym elementem**, ponieważ zawiera wszystkie inne elementy, a poza tym nie można go umieścić w innym elemencie. Zasady te dotyczą zarówno języka HTML, jak i XHTML.
- 3 W elemencie **html** dokument jest podzielony na dwie sekcje: **head** i **body**. W elemencie **head** (nagłówek) są umieszczone informacje opisujące dokument, takie jak tytuł, stosowane arkusze stylów, skrypty i inne metadane.
- 4 Elementy **meta** w sekcji **head** dostarczają informacji o samym dokumencie. Istnieje wiele zastosowań tych elementów, ale my korzystamy tylko z jednego — określenia typu **kodowania znaków** (czyli standardu kodowania liter, cyfr i symboli) stosowanego w dokumencie. Nie ma sensu wdawać się teraz w szczegóły, ale musisz wiedzieć, że z wielu względów dobrze jest umieścić element **meta charset** (choćby ze względu na prawidłowe wyświetlanie polskich „ogonków”).
- 5 Ważnym elementem sekcji **head** jest **title**, ponieważ — zgodnie ze specyfikacją HTML — wszystkie dokumenty muszą posiadać tytuł.
- 6 W sekcji **body** umieszcza się wszystko to, co ma zostać wyświetlone przez przeglądarkę.

Jesteś gotowy na utworzenie struktury strony restauracji „Pod Czarną Gąską”? Otwórz dokument *index.html* i przejdź do ćwiczenia 4.2.

### UWAGA

Przed pojawieniem się specyfikacji HTML5 element **meta** służący do określenia kodowania znaków był trochę bardziej skomplikowany. Jeśli tworzysz dokumenty w standardzie HTML 4.01 lub XHTML 1.0, element **meta** powinien wyglądać tak:

```
<meta http-equiv="contenttype"
content="text/html;
charset=UTF-8">
```



Rysunek 4.7. Podstawowa struktura dokumentu HTML

## Ćwiczenie 4.2.

## Definiowanie podstawowej struktury

1. Otwórz dokument *index.html*, o ile nie zrobiłeś tego wcześniej.
2. Rozpocznij od dodania na samej górze deklaracji typu dokumentu HTML5:  

```
<!DOCTYPE html>
```
3. W kolejnym wierszu wstaw znacznik otwierający `<html>`, a na samym końcu — po całej zawartości pliku — znacznik zamykający `</html>`.
4. Teraz utwórz nagłówek dokumentu, w którym ma się znaleźć tytuł strony. W tym celu przed treścią umieść znaczniki `<head>` i `</head>`. Wewnątrz sekcji nagłówka wstaw znacznik `<meta charset="utf-8">` ustalający kodowanie znaków oraz wpisz tytuł **Bistro "Pod Czarną Gąską"**, umieszczając go między otwierającym i zamykającym znacznikiem `<title>`.

*Można również powiedzieć, że element `title` jest **zagnieżdżony** w elemencie `head`. Na temat zagnieżdżania dowiesz się więcej w kolejnych rozdziałach.*

5. Na koniec musisz zdefiniować ciało dokumentu. W tym celu całą treść dokumentu umieść między znacznikami `<body>` i `</body>`. Po wprowadzeniu zmian kod dokumentu powinien wyglądać jak poniżej (dopisane fragmenty zostały wyróżnione pogrubieniem):

```
<!DOCTYPE html>
<html>

<head>
<meta charset="utf-8">
<title>Bistro "Pod Czarną Gąską"</title>
</head>

<body>
Bistro "Pod Czarną Gąską"
```

## Restauracja

Bistro "Pod Czarną Gąską" oferuje wybór popularnych dań obiadowych, którymi możesz się delektować w przyjaznej atmosferze. Menu jest często zmieniane, by zawsze serwować dania ze świeżych produktów.

## Usługi cateringowe

Kiedy Ty będziesz się świetnie bawić, my będziemy dla Ciebie gotować. Catering "Pod Czarną Gąską" sprostą wszystkim wyzwaniom: czy będą to przekąski, spotkania klubowe, czy wystawne bankiety.

## Lokalizacja i godziny otwarcia

Kuchary, ul. Smaczna 13

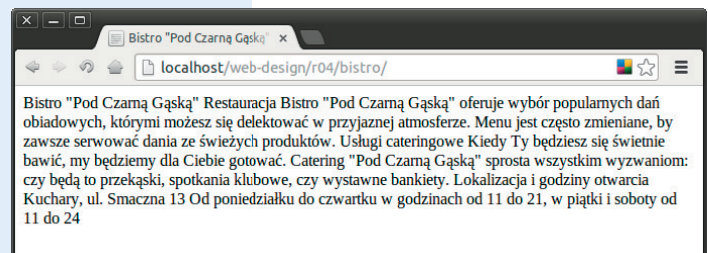
Od poniedziałku do czwartku w godzinach od 11 do 21, w piątki i soboty od 11 do 24

```
</body>
```

```
</html>
```

6. Zapisz zmiany w pliku, tak by została nadpisana starsza wersja. Otwórz dokument w przeglądarce lub — jeśli jest on już otwarty — odśwież stronę. Na [rysunku 4.8](#) został zaprezentowany aktualny wygląd dokumentu.

**Rysunek 4.8.** Wygląd strony w przeglądarce po zdefiniowaniu elementów struktury dokumentu



### Pamiętaj o dobrym tytule

Element `title` jest nie tylko wymagany w każdym dokumencie, ale także bardzo przydatny. Tytuł dokumentu jest tym, co jest wyświetlane na liście w Zakładkach bądź Ulubionych. Opisowe tytuły są także kluczowym narzędziem zwiększającym dostępność strony, ponieważ są pierwszą rzeczą, jaką słyszą użytkownicy strony korzystający z niej za pomocą czytnika ekranu. Również wyszukiwarki internetowe w dużej mierze polegają na tytułach dokumentów. Z tego powodu należy nadawać wszystkim dokumentom przemyślane oraz opisowe tytuły i unikać tytułów niejasnych typu „Witam” czy „Moja strona”. Należy się również zatroszczyć o odpowiednią długość tytułu, by mieścił się on w pasku tytułu przeglądarki. Dobrym rozwiązaniem jest też umieszczanie konkretnej informacji (na przykład nazwy firmy) na początku tytułu, tak by była ona widoczna nawet w sytuacji, gdy w przeglądarce jest otwartych wiele zakładek.

Wygląda na to, że po zdefiniowaniu struktury dokumentu niewiele się zmieniło — możemy tylko zauważyć, że przeglądarka wyświetla teraz tytuł strony. Gdybyś dodał tę stronę do Zakładek lub Ulubionych, tytuł pojawiłby się na liście (więcej informacji w ramce „Pamiętaj o dobrym tytule”). Treść strony nadal jest wyświetlana jako jeden blok tekstu, ponieważ nie wskazaliśmy przeglądarce, jak miałyby zostać podzielona. Zajmiemy się tym już niebawem.

## Krok 3. Oznaczamy elementy tekstowe

Nawet mając tak niewielkie doświadczenie z kodem HTML, nietrudno się domyślić, że teraz trzeba będzie uzupełnić treść strony odpowiednimi znacznikami definiującymi nagłówki (`h1` i `h2`), akapity (`p`), a także tekst zaakcentowany (`em`). Zajmiemy się tym w [ćwiczeniu 4.3](#), jednak przed jego rozpoczęciem warto poświęcić chwilę na omówienie tego, co można, a czego nie można robić ze znacznikami HTML.

### Wprowadzenie do znaczników semantycznych

Podstawowym celem języka HTML jest nadanie treści strony odpowiedniego znaczenia oraz struktury. Zapamiętaj, że HTML **nie służy** do ustalania wyglądu zawartości strony (czyli jej prezentacji).

Naszym zadaniem jest wybranie tych elementów HTML, które najlepiej oddają znaczenie poszczególnych fragmentów treści. Nazywa się to **semantycznym oznaczaniem treści** lub **dodawaniem znaczników semantycznych** (ang. *semantic markup*). Na przykład najistotniejszy nagłówek znajdujący się na początku treści powinien zostać oznaczony jako `h1`. Nie należy się przejmować tym, jak będzie wyglądał w przeglądarce — można to z łatwością zmienić za pomocą arkusza stylów. Ważne jest natomiast to, by wybrać element na podstawie tego, co ma w danym przypadku największy sens.

Poza dodawaniem znaczenia do treści strony znaczniki nadają jej również strukturę. Sposób następowania elementów po sobie lub zagnieżdżania ich wewnątrz innych tworzy pomiędzy nimi relacje. Strukturę traktuj jak konspekt dokumentu (którego technicznym odpowiednikiem jest **obiektywny model dokumentu**, czyli **DOM** — ang. *Document Object Model*). Dzięki ustaleniu hierarchii elementów przeglądarki wiedzą, jak traktować zawartość dokumentu. Poza tym struktura umożliwia definiowanie wyglądu (czyli prezentacji) stron za pomocą arkuszy stylów oraz zachowań — z wykorzystaniem JavaScriptu. Struktura dokumentu jest omówiona bardziej szczegółowo w trzeciej **części** książki, przy okazji omawiania kaskadowych arkuszy stylów, oraz w czwartej **części**, poświęconej językowi JavaScript.

Choć HTML miał być w zamierzeniach wykorzystywany do nadawania znaczenia oraz struktury, ta misja w początkowych latach istnienia internetu została nieco wypaczona. Ponieważ nie był dostępny mechanizm arkuszy stylów, rozszerzano HTML w taki sposób, by autorzy stron mogli zmieniać wygląd czcionek, kolory czy rozmieszczenie tekstu za pomocą znaczników i ich atrybutów. Tego typu dodatki prezentacyjne można znaleźć w kodzie starszych stron lub dokumentów utworzonych za pomocą starszych narzędzi. W tej książce skupiamy się jednak na wykorzystywaniu języka HTML w poprawny sposób, zgodny z obecnymi standardami.

Dość już tego wykładu — czas na [ćwiczenie 4.3](#), w którym popracujemy nad treścią dokumentu.

## Ćwiczenie 4.3.

## Definiowanie elementów tekstowych

1. Otwórz dokument *index.html* w edytorze, o ile jeszcze tego nie zrobiłeś.
2. Pierwszy wiersz tekstu: *Bistro „Pod Czarną Gąską”* jest głównym nagłówkiem strony, dlatego zostanie oznaczony jako element nagłówka poziomu pierwszego (**h1**). Na początku tego wiersza wstaw znacznik otwierający `<h1>`, a na jego końcu — zamykający `</h1>`:  
`<h1>Bistro "Pod Czarną Gąską"</h1>`
3. Dokument zawiera również trzy pomniejsze nagłówki. Oznacz je w podobny sposób, ale zastosuj elementy nagłówków poziomu drugiego (**h2**). Pierwszy został przedstawiony poniżej, a kolejnymi (*Usługi cateringowe* oraz *Lokalizacja i godziny otwarcia*) musisz się zająć sam:  
`<h2>Restauracja</h2>`
4. Po każdym z elementów **h2** następują krótkie fragmenty tekstu, które należy oznaczyć jako akapity (czyli elementy **p**). Poniżej znajduje się przykład pierwszego akapitu, a kolejnymi musisz się zająć sam:  
`<p>Restauracja Bistro "Pod Czarną Gąską" oferuje wybór popularnych dań obiadowych, którymi możesz się delektować w przyjaznej atmosferze. Menu jest często zmieniane, by zawsze serwować dania ze świeżych produktów.</p>`
5. Na koniec w sekcji *Usługi cateringowe* należy jakoś wyróżnić to, że odwiedzający restaurację powinni pozostawić gotowanie nam. Żeby zaakcentować tekst, należy umieścić go w elemencie **em**, jak poniżej:

`<p>Kiedy Ty będziesz się świetnie bawić, <em>my będziemy dla Ciebie gotować</em>. Catering "Pod Czarną Gąską" sprostą wszystkim wyzwaniom: czy będą to przekąski, spotkania klubowe, czy wystawne bankiety.</p>`

6. Po wstawieniu wszystkich znaczników należy, tak jak poprzednio, zapisać plik i otworzyć (lub odświeżyć) stronę w przeglądarce. Powinna wyglądać mniej więcej tak jak na [rysunku 4.9](#). Jeśli tak nie jest, sprawdź znaczniki pod kątem brakujących nawiasów lub ukośników w znacznikach zamykających.



**Rysunek 4.9.** Strona po oznaczeniu tekstu za pomocą elementów HTML

Powoli zaczynamy do czegoś dochodzić. Po oznaczeniu elementów przeglądarka może teraz wyświetlić tekst w poprawny sposób. Warto jednak poświęcić jeszcze chwilę na bardziej szczegółowe omówienie tego, co widzimy na [rysunku 4.9](#).

## Elementy blokowe oraz liniowe

Choć może się to wydawać oczywiste, warto podkreślić, że nagłówki oraz akapity rozpoczynają się od nowych wierszy i nie są zapisane ciągiem jeden po drugim, jak było wcześniej. Jest tak, ponieważ są one przykładami **elementów blokowych** (ang. *block element*). Przeglądarki traktują elementy blokowe tak, jakby znajdowały się w małych prostokątnych pojemnikach, ułożone jeden na drugim. Każdy element blokowy rozpoczyna się od nowego wiersza i zazwyczaj nad całym elementem oraz pod nim domyślnie dodawany jest jakiś odstęp. Na [rysunku 4.10](#) elementy blokowe są oznaczone na czerwono.

Wstawianie komentarzy

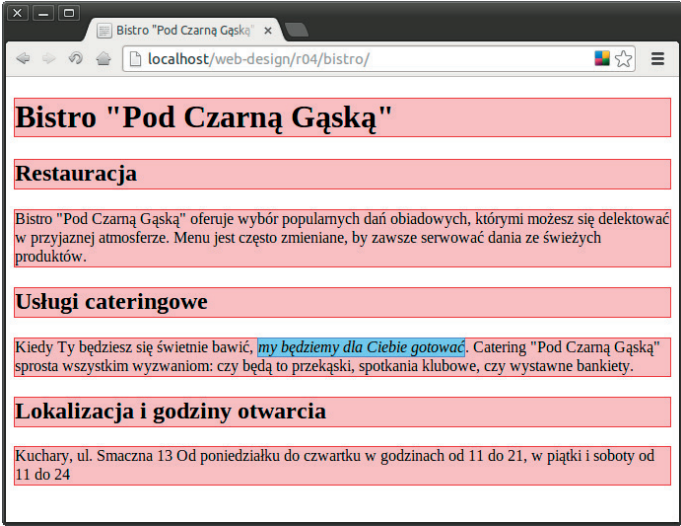
W dokumencie źródłowym można umieścić notatki dla siebie oraz innych osób, oznaczając je jako komentarze. Wszystko, co umieści się pomiędzy znacznikami komentarzy (`<!--` oraz `-->`), nie zostanie wyświetlone w przeglądarce i nie będzie miało wpływu na resztę źródła dokumentu.

```
<!-- To jest komentarz. -->
<!-- To jest komentarz
rozciągający się na kilka
wierszy.
kończy się tutaj. -->
```

Komentarze przydają się do opisywania oraz organizowania długich dokumentów, zwłaszcza jeśli pracuje nad nimi wieloosobowy zespół. W poniższym przykładzie komentarze są wykorzystywane do oznaczenia części strony zawierającej blok nawigacji.

```
<!-- początek nawigacji -->
<ul>
...
</ul>
<!-- koniec nawigacji -->
```

Należy pamiętać, że choć w oknie przeglądarki nie pojawiają się komentarze, są one widoczne w źródle dokumentu, które każdy użytkownik może wyświetlić, dlatego trzeba pozostawiać jedynie te komentarze, które wszyscy mogą zobaczyć. Najlepszym rozwiązaniem jest jednak usunięcie komentarzy przed opublikowaniem strony, co dodatkowo zmniejsza rozmiar plików.



Rysunek 4.10. Wyróżniona struktura elementów na stronie

Inaczej wygląda to w przypadku tekstu oznaczonego jako zaakcentowany (`em`). Nie rozpoczyna się on od nowego wiersza, ale pozostaje częścią akapitu. Jest tak, ponieważ element `em` jest **elementem liniowym** (ang. *inline element*). Elementy wewnętrzne nie rozpoczynają nowych wierszy, pozostają na swoim miejscu. Na **rysunku 4.10** element wewnętrzny `em` jest zaznaczony na jasnoniebiesko.

Domyślne style

Na **rysunkach 4.9** oraz **4.10** można również zauważyć, że przeglądarka nadaje elementom strony wygląd odzwierciedlający strukturę dokumentu — nagłówek pierwszego stopnia jest największy i najbardziej rzuca się w oczy, nagłówek drugiego stopnia jest nieco mniejszy i tak dalej.

W jaki sposób przeglądarka ustala, jak powinien wyglądać element `h1`? Wykorzystuje arkusz stylów! Wszystkie przeglądarki mają wbudowane własne arkusze stylów (w specyfikacji są one określane mianem *user agent style sheets* — arkusze stylów agenta użytkownika), które określają domyślny sposób wyświetlania poszczególnych elementów. Domyślny wygląd jest podobny w różnych przeglądarkach (na przykład elementy `h1` zawsze są duże i pogrubione), jednak istnieją pewne różnice (elementy `blockquote` mogą być wcięte lub nie).

Jeśli uważasz, że element `h1` wyświetlany przez przeglądarkę jest zbyt duży i niezgrabny, wystarczy zmienić to za pomocą arkusza stylów. Należy oprzeć się na pokusie oznaczenia nagłówka innym elementem tylko po to, by wyglądał on lepiej (na przykład używając `h3` w miejsce `h1`, by element ten nie był aż tak duży). W czasach kiedy obsługa arkuszy stylów nie była tak powszechna, w taki właśnie sposób nadużywano znaczników. Teraz, gdy istnieją arkusze stylów kontrolujące wygląd strony, zawsze powinno się wybierać elementy zgodnie z tym, jak opisują one zawartość, i nie należy się przejmować domyślnym wyglądem nadawanym przez przeglądarkę.



Prezentację strony poprawimy za chwilę za pomocą arkusza stylów, jednak najpierw warto dodać do strony obrazek.

## Krok 4. Wstawiamy obrazek

Strona internetowa bez obrazków wygląda średnio. Dlatego też w [ćwiczeniu 4.4](#) dodamy jeden obrazek, używając w tym celu elementu `img`. Obrazki zostaną omówione bardziej szczegółowo w [rozdziale 7.](#), „Grafika”, jednak na razie skupimy się na dwóch podstawowych zagadnieniach: elementach pustych oraz atrybutach.

### Elementy puste

Dotychczas wszystkie elementy wykorzystane na stronie internetowej Bistro „Pod Czarną Gąską” podlegały regułom składni przedstawionym na [rysunku 4.1](#) i składały się z tekstu otoczonego znacznikami otwierającymi oraz zamykającymi.

Spora liczba elementów nie ma zawartości tekstowej, ponieważ są one wykorzystywane jako proste instrukcje. O takich elementach mówi się, że są **puste** (ang. *empty*). Element obrazka (`img`, od angielskiego *image* — obrazek) jest przykładem tego typu elementu. Przekazuje on przeglądarce, że ma pobrać plik graficzny z serwera i wstawić go na stronę w miejscu występowania znacznika. Inne elementy puste to podział wiersza (`br`), linia pozioma (`hr`), a także elementy udostępniające informacje o dokumencie, jednak niewpływające na wyświetlaną treść, jak element `meta`.

Na [rysunku 4.11](#) została przedstawiona składnia pustego elementu (bardzo prosta w porównaniu z tą z [rysunku 4.4](#)). Jeśli dokument tworzysz w języku XHTML, składnia delikatnie się różni (patrz [ramka „Puste elementy w XHTML”](#)).

```
<nazwa-elementu>
```

Przykład: element `br` służy do wstawienia podziału wiersza.

```
<p>u1. Smaczna 13<br>Kuchary</p>
```

**Rysunek 4.11.** Składnia pustego elementu

### Atrybuty

Wróćmy teraz do wstawiania obrazka. Sam znacznik `<img>` w takiej postaci nie jest oczywiście zbyt przydatny, ponieważ nie wiadomo, jaki obrazek ma zostać wstawiony. Aby temu zaradzić, użyjemy atrybutów. **Atrybuty** to instrukcje określające lub modyfikujące element. Dla elementu `img` wymagany jest atrybut `src` (pochodzący od angielskiego wyrazu *source* oznaczającego źródło), który wskazuje lokalizację pliku obrazka za pomocą jego adresu URL.

#### Puste elementy w XHTML-u

W języku XHTML wszystkie elementy, włącznie z pustymi, muszą zostać zamknięte (**zakończone**). Elementy puste zamyka się, dodając na ich końcu, tuż przed nawiasem końcowym, ukośnik poprzedzony spacją, na przykład `<img />`, `<br />` oraz `<hr />`. Poniżej został zaprezentowany przykład wykorzystania składni XHTML:

```
<p>u1. Smaczna 13  
<br />Kuchary</p>
```

Składnia atrybutów jest następująca:

```
atribut="wartość"
```

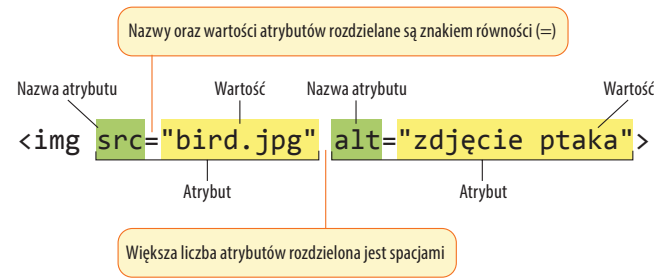
W znaczniku atrybuty umieszcza się po nazwie elementu. W niepustych elementach dodaje się je tylko w otwierającym znaczniku:

```
<element atribut="wartość">
<element atribut="wartość">Zawartość</element>
```

W znaczniku można umieścić więcej niż jeden atrybut — kolejność ich wpisywania nie ma znaczenia, trzeba jedynie oddzielać je spacjami.

```
<element atribut1="wartość" atribut2="wartość">
```

Na rysunku 4.12 został przedstawiony element `img` wraz z wymaganymi atrybutami.



Rysunek 4.12. Element wraz z atrybutami

Oto, co musisz wiedzieć na temat atrybutów:

- Atrybuty umieszczane są po nazwie elementu tylko w znaczniku otwierającym, nigdy w zamykającym.
- Do elementu można stosować większą liczbę atrybutów rozdzielanych spacjami w znaczniku otwierającym. Ich kolejność nie jest istotna.
- Atrybuty przyjmują wartości, które następują po znaku równości (=). W języku HTML niektóre atrybuty nie muszą mieć przypisanej wartości, np. atrybut `checked` (który służy do zaznaczania pola wyboru). Z kolei składnia języka XHTML wymaga przypisywania wartości w każdym przypadku (`checked="checked"`). Ten typ atrybutu jest nazywany **boolowskim**, ponieważ opisuje cechę, która może być albo włączona, albo wyłączona.
- W zależności od przeznaczenia atrybutu wartość może być liczbą, słowem, łańcuchem znaków, adresem URL lub miarą. W książce znajdziesz przykłady na każdy z tych atrybutów.
- Niektóre wartości nie muszą być umieszczane w cudzysłowie, ale dotyczy to tylko języka HTML — w XHTML-u cudzysłów jest obowiązkowy. Wielu twórców stron zawsze stosuje cudzysłowy, by kod był spójny i czytelny. Mimo że przyjętą konwencją jest stosowanie cudzysłówów, w ich miejsce można wstawiać apostrofy, ale trzeba pamiętać o zachowaniu konsekwencji. I jeszcze jedna uwaga — w kodzie HTML trzeba używać „prostych” cudzysłówów i apostrofów, czyli `"`, a nie `'`.
- W pewnych elementach niektóre atrybuty są wymagane, jak na przykład `src` oraz `alt` w elemencie `img`.



- Nazwy atrybutów dostępnych dla każdego elementu są zdefiniowane w specyfikacjach HTML. Nie można samemu wymyślić atrybutu dla elementu<sup>2</sup>.

Najwyższy czas na trochę praktyki. Przed nami [ćwiczenie 4.4](#), w którym do strony Bistro „Pod Czarną Gąską” dodasz element `img` wraz z atrybutami.

#### Ćwiczenie 4.4.

##### Wstawianie obrazka

1. Pierwsze, co musisz zrobić, to zdobyć kopię obrazka, który ma zostać wyświetlony na stronie. Plik ten znajduje się w materiałach do tego rozdziału, które można pobrać ze strony wydawnictwa (<ftp://ftp.helion.pl/przyklady/prsti2.zip>). Możesz też otworzyć stronę z tym przykładem, która również jest dostępna na stronie wydawnictwa (<http://helion.pl/pliki/prsti2/04/bistro>), i pobrać rysunek. W tym celu kliknij prawym przyciskiem myszy (w Macu kliknij, trzymając wciśnięty *Control*) obrazek przedstawiający gęś i z podręcznego menu (patrz [rysunek 4.13](#)) wybierz polecenie *Zapisz grafikę jako* (lub podobnie — to zależy od przeglądarki). W oknie dialogowym zapisywania przejdź do katalogu *bistro*, w którym znajduje się plik *index.html*, i zapisz w nim pobierany plik graficzny *blackgoose.png*.

2. Następnie na początku nagłówka pierwszego stopnia wpisz kod elementu `img` wraz z atrybutami:

```
<h1>Bistro "Pod Czarną Gąską"</h1>
```

W atrybucie `src` podajemy nazwę pliku graficznego, który ma zostać umieszczony na stronie. Z kolei w atrybucie `alt` wpisujemy tekst, który zostanie wyświetlony, jeśli obrazek nie jest dostępny. Oba atrybuty są wymagane w każdym elemencie `img`.



#### Windows:

aby wyświetlić podręczne menu, kliknij obrazek prawym przyciskiem myszy

#### Mac:

aby wyświetlić podręczne menu, kliknij obrazek, trzymając wciśnięty klawisz *Control*. Nazwy i liczba dostępnych opcji może się różnić w zależności od przeglądarki

**Rysunek 4.13.** Zapisywanie pliku obrazka ze strony internetowej



- 2 Zgodnie ze specyfikacją HTML5 istnieje możliwość definiowania własnych atrybutów. Tworzy się je poprzez uzupełnienie własnej nazwy prefiksem `data-`, np. `data-mojatrybut`

— *przyp. tłum.*  
Kup ksi k

- 3. Aby obrazek znalazł się nad tytułem, po elemencie `img` wstaw znacznik podziału wiersza (`<br>`).  
`<h1><br>Bistro "Pod Czarną Gąską"</h1>`
- 4. Ostatni akapit podzielimy na trzy wiersze, dzięki czemu stanie się bardziej czytelny, co widać na rysunku 4.14. Wstaw znaczniki `<br>` w odpowiednich miejscach, by uzyskać ten sam rezultat.
- 5. Po wprowadzeniu zmian zapisz plik `index.html`, a następnie otwórz go lub odśwież w oknie przeglądarki. Strona powinna wyglądać tak jak na rysunku 4.14. Jeśli tak nie jest, sprawdź, czy plik z rysunkiem (`blackgoose.png`) znajduje się w tym samym katalogu co strona `index.html`. Jeśli tak jest, upewnij się, że w znaczniku `img` nie brakuje jakichś znaków, na przykład zamykającego cudzysłowu czy nawiasu ostrego.



Rysunek 4.14. Wygląd strony internetowej z wstawionym obrazkiem logo

## Krok 5. Zmieniamy wygląd za pomocą arkusza stylów

W niektórych sytuacjach może się okazać, że domyślny wygląd nadawany stronie przez przeglądarkę jest całkowicie satysfakcjonujący. W przypadku strony internetowej Bistro „Pod Czarną Gąską” tak nie jest. Jeżeli chcemy, by robiła lepsze wrażenie na potencjalnych klientach, musimy popracować nad jej wyglądem i trochę ją upiększyć. To jest zadanie dla kaskadowych arkuszy stylów (CSS).

W [ćwiczeniu 4.5](#) za pomocą kilku prostych reguł arkuszy stylów zmienimy wygląd elementów tekstowych oraz tło strony. Nie martw się, jeśli wszystkiego nie rozumiesz — kaskadowe arkusze stylów są szczegółowo omówione w trzeciej [części](#) książki. Teraz zostanie jedynie uchylony rąbek tego, co można osiągnąć poprzez dodanie „warstwy” prezentacji do struktury dokumentu utworzonej za pomocą znaczników HTML.

## Ćwiczenie 4.5.

## Dodawanie arkusza stylów

1. Otwórz w edytorze plik `index.html`.
2. Arkusz stylów osadzimy w dokumencie za pomocą elementu `style` (to tylko jeden z możliwych sposobów dodawania arkuszy stylów; pozostałe są omówione w rozdziale 11., „Kaskadowe arkusze stylów”).

Element `style` wstaw wewnątrz elementu `head`, jak na poniższym listingu:

```
<head>
  <meta charset="utf-8">
  <title>Bistro „Pod Czarną Gąską”</title>
  <style>
```

```

</style>
</head>
```

3. Teraz wewnątrz elementu `style` wpisz poniższe reguły. Nie przejmuj się, jeśli nie wiesz, na czym to dokładnie polega (choć to całkiem intuicyjne). Reguły stylów zostaną omówione w trzeciej części książki.

```
<style>

body {
  background-color: #faf2e4;
  margin: 0 15%;
  font-family: sans-serif;
}

h1 {
  text-align: center;
  font-family: serif;
  font-weight: normal;
  text-transform: uppercase;
  border-bottom: 1px solid #57b1dc;
  margin-top: 30px;
}
```

```
h2 {
  color: #d1633c;
  font-size: 1em;
}
```

```
</style>
```

4. Czas zapisać stronę i przyjrzeć się jej w przeglądarce. Powinna wyglądać podobnie do strony z rysunku 4.15. Jeśli tak nie jest, przejrzyj kod arkusza stylów, żeby sprawdzić, czy nie pominąłeś jakiegoś średnika bądź nawiasu klamrowego.



**Rysunek 4.15.** Strona Bistro „Pod Czarną Gąską” po zastosowaniu reguł stylów

Strona Bistro „Pod Czarną Gąską” jest gotowa. Nie tylko udało Ci się napisać pierwszy dokument HTML oraz arkusz stylów, ale dowiedziałeś się też co nieco na temat elementów, atrybutów, elementów pustych, elementów blokowych i liniowych, podstawowej struktury dokumentu HTML oraz poprawnego stosowania znaczników.

## Kiedy dobre strony nie działają dobrze

Dotychczasowe ćwiczenia przebiegły dosyć gładko, jednak przy ręcznym wpisywaniu kodu HTML łatwo jest popełnić jakiś drobny błąd. Jeden źle wpisany znak potrafi niestety popsuć działanie całej strony. Za chwilę celowo wprowadzimy błędy w dokumencie, żeby można było zobaczyć, co się wtedy stanie.

### UWAGA

*Pominięcie prawego ukośnika w znaczniku zamykającym (i w efekcie pominięcie samego znacznika zamykającego) dla niektórych elementów blokowych, takich jak nagłówki czy akapity, może nie mieć aż tak dramatycznego efektu. Przeglądarki interpretują rozpoczęcie nowego elementu blokowego jako jednoczesne zakończenie poprzedniego.*

Co się stanie, gdy zapomni się wpisać ukośnik (/) w znaczniku zamykającym element `</em>`? Wystarczyło zapomnieć o jednym znaku, a spora część dokumentu została wyświetlona tekstem zaakcentowanym (kursywą), co widać na rysunku 4.16. Stało się tak, ponieważ z powodu braku ukośnika przeglądarka nie wyłączy ustawionego formatowania, a więc jest ono stosowane aż do końca dokumentu.

A co się stanie, jeśli przypadkowo pominiemy nawias znajdujący się na końcu pierwszego znacznika `<h2>` (jak na rysunku 4.17)?



Rysunek 4.16. Kiedy zostanie pominięty ukośnik, przeglądarka nie wie, gdzie kończyć się element, co ilustruje powyższy przykład



Rysunek 4.17. Brakujący nawias końcowy sprawia, że cała następująca po nim treść staje się częścią znacznika i tym samym nie zostaje wyświetlona

Jak widać, brakuje teraz nagłówka. Dzieje się tak, ponieważ bez nawiasu zamykającego znacznik przeglądarka zakłada, że cały następujący po nim tekst — aż do następnego nawiasu zamykającego (`>`) — jest częścią znacznika `<h2>`. Przeglądarki nie wyświetlają tekstu znajdującego się wewnątrz znacznika, dlatego nagłówków zniknęło. Przeglądarka zignorowała nieznaną nazwę elementu i przeszła do kolejnego.

Popełnianie błędów w pierwszych dokumentach HTML i ich samodzielne korygowanie jest świetną metodą nauki. Jeśli udało Ci się bezbłędnie napisać kod strony, spróbuj się nim pobawić, by sprawdzić, jak przeglądarka reaguje na różne zmiany. Może się to bardzo przydać, kiedy w przyszłości będziesz musiał rozwiązywać problemy z niedziałającymi stronami. Najczęściej spotykane kłopoty wymieniono w ramce „Masz problem?”. Warto zwrócić uwagę, że problemy te nie są typowe wyłącznie dla początkujących. Takie drobne błędy przydarzają się nawet profesjonalistom.

## Walidacja dokumentów

Jedną z metod, którą stosują profesjonaliści, by wyłapać błędy w kodzie dokumentów, jest ich walidacja. Co to oznacza? Walidacja polega na sprawdzeniu poprawności kodu względem specyfikacji używanej wersji języka HTML (a jest ich kilka, co jest omówione dokładniej w rozdziale 10., „Co nowego w HTML5?”). Powinneś zawsze poddawać swoje witryny walidacji, ponieważ zachowanie zgodności ze standardami zapewnia większą kompatybilność z różnymi przeglądarkami, przyspiesza ich działanie oraz zwiększa dostępność.

Tak naprawdę przeglądarki nie wymagają bezbłędnych dokumentów (starają się jak najlepiej je wyświetlić, ignorując drobne błędy). Może się jednak zdarzyć, że niezauważony w porę błąd da o sobie znać w innej przeglądarce lub na innym urządzeniu.

Jak w takim razie sprawdzić poprawność dokumentu? Mógłbyś sam go dokładnie przeanalizować (lub poprosić o to znajomego), ale weź pod uwagę to, że ludzie popełniają błędy. Poza tym nikt nie jest w stanie zapamiętać wszystkich szczegółów zawartych w specyfikacji. Powinieneś więc użyć **walidatora**, czyli programu sprawdzającego poprawność kodu HTML pod kątem zgodności ze standardami. Poniżej znajduje się lista najistotniejszych spraw, które sprawdza walidator:

- dołączenie deklaracji typu dokumentu (**DOCTYPE**) — bez niej walidator nie wie, której wersji języka HTML lub XHTML używasz;
- wskazanie kodowania znaków stosowanego w dokumencie;
- uwzględnienie wymaganych reguł i atrybutów;
- zastosowanie niestandardowych elementów;
- pomyłki w znacznikach;
- błędy zagnieżdżenia elementów;
- literówki i inne drobne błędy.

Programiści korzystają z wielu narzędzi służących do sprawdzania i poprawiania błędów w dokumentach HTML. Konsorcjum W3C udostępnia na swojej stronie walidator (<http://validator.w3.org>), z którego można korzystać online. Dokumenty HTML5 można sprawdzać też za pomocą walidatora ze strony [html5.validator.nu](http://html5.validator.nu). Możesz również skorzystać z walidatorów dostarczanych wraz z narzędziami programistycznymi przeglądarek (w Chrome i Safari), dodatkami (na przykład Firebug do Firefoksa), a nawet graficznymi edytorami stron, takimi jak Dreamweaver.

## Sprawdź się!

Pora sprawdzić, czy rozumiałeś podstawy stosowania znaczników. Odpowiedz na poniższe pytania, wykorzystując do tego wiedzę zdobytą w tym rozdziale. Odpowiedzi na pytania znajdziesz w **dodatku A**.

1. Jaka jest różnica między znacznikiem a elementem?
2. Napisz kod podstawowej struktury dokumentu HTML.

## Masz problem?

Poniżej znajduje się lista typowych problemów, które pojawiają się podczas tworzenia stron internetowych i oglądania ich w przeglądarkach.

**Zmieniłem swój dokument, ale kiedy odświeżam stronę w przeglądarce, wygląda dokładnie tak samo.**

Możliwe, że dokument nie został zapisany przed odświeżeniem lub też został zapisany w innym katalogu.

**Pół mojej strony zniknęło.**

Mogło się tak zdarzyć, jeśli brakuje gdzieś nawiasu zamykającego (>) lub cudzysłowu wewnątrz znacznika. Jest to często spotykany błąd przy ręcznym pisaniu kodu HTML.

**Za pomocą elementu `img` wstawiłem na stronę grafikę, jednak w przeglądarce pokazuje się tylko ikona wskazująca na nieistniejący obrazek.**

Taka ikona może oznaczać kilka rzeczy. Być może przeglądarka nie potrafi odnaleźć pliku graficznego. Upewnij się, że adres URL pliku obrazka jest poprawny (adresy URL zostaną omówione w **rozdziale 6**, „**Hiperłącza**”). Trzeba sprawdzić, czy plik obrazka naprawdę znajduje się w podanym katalogu. Jeśli tak jest, należy się upewnić, że jest zapisany w jednym z formatów, które przeglądarka potrafi wyświetlić (GIF, JPG lub PNG), a także że ma właściwe rozszerzenie (odpowiednio `.gif`, `.jpg` lub `.jpeg` oraz `.png`).

3.

Poniżej znajduje się kilka przykładowych nazw plików. Dla każdej z nich określ, czy jest to prawidłowa nazwa dokumentu webowego, zaznaczając odpowiedź „tak” lub „nie”. Jeśli uznasz, że jakaś nazwa nie jest poprawna, napisz, dlaczego tak uważasz.

a.

*Sunflower.html*

tak

nie

b.

*index.doc*

tak

nie

c.

*cooking home page.html*

tak

nie

d.

*Song\_Lyrics.html*

tak

nie

e.

*games/rubix.html*

tak

nie

f.

*%whatever.html*

tak

nie
4.

Wszystkie poniższe przykłady znaczników są niepoprawne. Opisz błędy popełnione w każdym z nich i podaj poprawne wersje znaczników.

a.

`<img "birthday.jpg">`

a.

`<i>Gratulacje!<i>`

a.

`<a href="file.html">tekst odsyłaacza</a href="file.html">`

a.

`<p>To jest nowy akapit<\p>`

5.

W jaki sposób można oznaczyć komentarz w dokumencie HTML, by nie był on wyświetlany w oknie przeglądarki?

tutaj zaczyna się lista produktów
- ## Przegląd elementów HTML tworzących strukturę dokumentu
- W tym rozdziale zostały opisane elementy ustanawiające strukturę dokumentu. Pozostałe elementy wprowadzone w ćwiczeniach zostaną omówione bardziej szczegółowo w kolejnych rozdziałach.
- | Element | Opis   |
|---------|--|
| body    | Określa ciało dokumentu, które przechowuje treść             |
| head    | Określa nagłówek zawierający informacje o dokumencie         |
| html    | Główny element dokumentu zawierający wszystkie inne elementy |
| meta    | Dostarcza informacje o dokumencie                            |
| title   | Nadaje stronie tytuł   |
- Kup książkę
- Poleć książkę
- 68 Część II. Znaczniki HTML i struktura dokumentu

# Skorowidz

## A

- absolute positioning, *Patrz:* pozycjonowanie bezwzględne
- ActionScript, 11
- adaptive layout, *Patrz:* strona układ adaptacyjny
- adjacent sibling selector, *Patrz:* selektor przylegającego rodzeństwa
- Adobe Dreamweaver, *Patrz:* Dreamweaver
- Adobe Fireworks, *Patrz:* Fireworks
- Adobe Flash, *Patrz:* Flash
- Adobe Illustrator, *Patrz:* Illustrator
- Adobe Photoshop, *Patrz:* Photoshop
- Adobe Photoshop Elements, *Patrz:* Photoshop Elements
- adres
  - IP, 22
    - IPv4, *Patrz:* IPv4
    - IPv6, *Patrz:* IPv6
  - plików obrazów, 115
  - URL, 24, 105, 106, 125, 149, 156, 284, 302, 478
    - bezwzględny, 106
    - długi, 107
    - względny, 106, 110, 111, 112, 125
- agent użytkownika
  - arkusze stylów, 60
  - identyfikator, 453
- Ajax, 497
- akapit, *Patrz:* element p
- akronim, 89
- Allsopp John, 297
- Alman Ben, 477
- animacja, 11, 399, 400, 402, 404, 416, 514, 515
  - automatyczna, 399
  - jawna, *Patrz:* animacja klatkowa
  - klatkowa, 420
  - niejawna, 420
  - właściwość, 422
- antialiasing, 531
- Apache, 22
- API, *Patrz:* interfejs API
- architekt informacji, 10
- arkusz stylów
  - agenta użytkownika, 60
  - kaskadowy, *Patrz:* CSS
- artykuł, 79, 80, *Patrz też:* element article
- ASP.NET, 9, 13, 150
- Asynchronous JavaScript and XML, *Patrz:* Ajax
- atrybut
  - 62, 184, 187
  - accesskey, 187
  - action, 149, 176
  - align, 344
  - alt, 42, 62, 124, 125, 132, 158
  - aria, 187
  - autoplay, 196, 198
  - boolowski, 62
  - cellpadding, 142
  - cellspacing, 142
  - checked, 62, 161
  - class, 95, 96, 97, 98, 187, 225, 246
  - colspan, 139, 140, 146
  - contenteditable, 187, 191
  - contextmenu, 187
  - controls, 195
  - data, 188
  - dir, 188
  - disabled, 155, 176
  - draggable, 188, 191
  - dropzone, 191
  - enctype, 176
  - for, 171
  - globalny, 187
  - hard, 155
  - headers, 137, 144, 146
  - height, 128
  - hidden, 188
  - href, 105, 106, 301
  - id, 95, 96, 97, 98, 116, 188, 225, 245
  - label, 164, 176
  - lang, 188
  - list, 157
  - loop, 196, 198
  - maxlength, 154, 156
  - mediagroup, 196
  - method, 150, 176
  - multiple, 164
  - muted, 196
  - name, 151, 152, 154, 156, 161, 176
  - placeholder, 154, 155
  - poster, 195
  - preload, 196
  - readonly, 155
  - rel, 301



## atrybut

role, 187, 188  
 rowspan, 139, 141, 146  
 scope, 137, 144, 146  
 selected, 164  
 size, 156  
 soft, 155  
 spellcheck, 188  
 src, 61, 62, 115, 124, 125, 132, 195, 463  
 start, 75  
 step, 169  
 style, 188  
 summary, 143  
 tabindex, 188  
 target, 119  
 title, 131, 155, 188  
 type, 153, 154, 155, 156, 158, 161, 162, 165, 166, 167, 168, 169, 176, 197, 213, 301, 463  
 usemap, 131  
 value, 154, 156, 176  
 width, 128, 132  
 wrap, 155  
 Audacity, 194  
 audio, 193  
 authoring, 8  
 backend, 9  
 bank fotografii, 508, 509  
 BBEdit, 17, 19  
 Berners-Lee Tim, 11, 21, 90, 182, 183  
 biblioteka  
   Dojo, 498  
   JavaScript, 497, 498  
   jQuery, 498, 499  
     pobieranie pliku, 499  
     tworzenie skryptów, 500  
   jQuery Mobile, 498  
   jQuery UI, 498  
   LESS, *Patrz:* LESS  
   Modernizr, 80, 495  
   MooTools, 498  
   Proietti Valerio, 498  
   Prototype, 498  
   Ruby on Rails, 150, 498

SASS CSS, *Patrz:* SASS  
 YUI, 498

## B

block element, *Patrz:* element blokowy  
 blog, 4  
 Blogger, 4  
 blok zawierający, *Patrz:* pozycjonowanie  
   blok zawierający  
 Bluefish, 50  
 border, *Patrz:* obramowanie  
 Bowman Doug, 397  
 box model, *Patrz:* model pojemnika  
 braille, *Patrz:* media  
 breakpoint, *Patrz:* punkt graniczny

## C

cache, *Patrz:* pamięć podręczna  
 Cagle Kurt, 538  
 Cailliau Robert, 22  
 Calzadilla Anthony, 420  
 Camen Kroc, 196  
 Cascading Style Sheet, *Patrz:* CSS  
 Catlin Hampton, 433  
 Cederholm Dan, 395, 410  
 checkbox, *Patrz:* przycisk pola wyboru  
 child, *Patrz:* dziecko  
 child selector, *Patrz:* selektor dziecka  
 Chisholm Wendy, 43  
 cień, 254, 263, 335  
   wewnętrzny, 336  
 Clark Keith, 495  
 Clarke Andy, 247  
 client-side, 23  
 client-side scripting language, *Patrz:*  
   język skryptowy działający po stronie  
   klienta  
 clipart, *Patrz:* klipart  
 Coda, 17, 19  
 collection, *Patrz:* kolekcja  
 ColorZilla, 300  
 comparison operator, *Patrz:* operator  
   porównania

Composer, 16  
 content area, *Patrz:* element zawartość  
 Content Strategist, *Patrz:* strateg  
   zawartości  
 contextual selector, *Patrz:* selektor  
   kontekstowy  
 Corel PaintShop Pro, 17, 19, 508  
 Coyier Chris, 134, 295, 299  
 CSS, 8, 11, 12, 29, 38, 64, 86, 123, 124, 172, 175, 205, 207, 213, 217, 305, 373  
   !important, 217, 218  
   historia, 221  
   komentarz, 213, 300  
   konflikt, 217  
   modularny, 302  
   narzędzia, 222  
   osadzony, 213, 218, 300  
   przeglądarki, 217  
   przejście, *Patrz:* przejście  
   regiony, *Patrz:* regiony CSS  
   reguła, 210, 211  
     @font-face, 227  
     @import, 300, 302  
     @keyframes, 421  
     kolejność, 219  
   transformacja, *Patrz:* transformacja  
   układ, 381  
   wewnętrzny, 214, 300  
   wykluczenia, *Patrz:* wykluczenia CSS  
   zalety, 207  
   zapytania, 445  
   zerowanie, 427  
   zewnątrzny, 213, 218, 300, 302  
 CSS Exclusions, *Patrz:* wykluczenia  
   CSS  
 CSS Regions, *Patrz:* regiony CSS  
 CSS Reset, *Patrz:* CSS zerowanie  
 Cygwin, 18, 19  
 cytat  
   długi, 76, 77, *Patrz też:* element  
   blockquote  
   krótki, 77, 89, *Patrz też:* element q  
 czcionka, *Patrz:* krój pisma  
 czynnik ekranu, 12, 23, 58, 70, 86, 87, 77, 126, 127, 134, 138, 171



## D

dane typ, 466

- ciąg znaków, 467
- liczbowy, 466
- logiczny, 467
- null, 466
- tablica, *Patrz:* tablica
- undefined, 466, 476

degradacja z wdziękiem, 37

deklaracja, 210, 211

- DOCTYPE, 56, 67, 184, 185
- wartość, 210, 212
- inherit, 239
- właściwość, *Patrz:* właściwość

descendant, *Patrz:* potomek

descendant selector, *Patrz:* selektor

- potomny

Deweloper, 8

DHTML, 13

display role, *Patrz:* rola wyświetlania

dithering, 520, 543, 545

DNS, 22

Document Object Model, *Patrz:* DOM

Document Type Definition, *Patrz:* DTD

dokument

- definicja typu, *Patrz:* DOCTYPE
- drzewo, *Patrz:* drzewo
- konspekt, 70, 81
- oznaczenie, 26
- struktura, 29, 55, 209, 215
- układ normalny, 341
- walidacja, 66
- zewnątrzny HTML, 130
- źródłowy, 26

DOM, 11, 13, 58, 461, 485

- trawersowanie, 486

Domain Name System, *Patrz:* DNS

domena, 22, 24

dopełnienie, 142, 292, 305, 306, 312

dostępność, 41, 42

- formularza, 153, 158, 161, 171
- tabeli, 142

Dreamweaver, 4, 16, 115, 222, 273

drzewo, 486

węzeł, 486, 489

- dodawanie, 490
- lista, 487
- metoda dostępu, 487
- usuwanie, 490

DTD, 185

Dunham Ethan, 228

dynamic pseudo-class, 276

dziecko, 215

dziedziczenie, 214, 216, 239, 274

## E

ECMAScript, 13, 460

edytor

- HTML, 16, 50
- WYSIWYG, 16

efekt halo, 531

Eich Brendan, 460

ekstranet, 23

elastic layout, *Patrz:* strona układ

- elastyczny

element, 184, 187

- a, 85, 105, 119, 276, 400
- abbr, 85, 86, 89, 104
- acronym, 86, 89, 189
- address, 84, 104
- applet, 86, 189
- article, 79, 80, 81, 82, 104, 187, 494
- aside, 79, 80, 81, 104, 187
- atrybut, *Patrz:* atrybut
- audio, 187, 190, 191
- autofocus, 176
- b, 85, 87, 88, 104, 172
- base, 97
- basefont, 86, 97, 189
- bdi, 85, 93, 104, 187
- bdo, 85, 93, 104
- big, 86, 189
- blockquote, 60, 76, 104
- blokowy, 59, 70, 76, 274, 306, 307
- plywający, 346
- body, 56, 68, 76, 215, 463
- br, 61, 85, 92, 104
- button, 158, 176
- canvas, 11, 187, 191, 198, 199, 200, 411
- caption, 137, 143, 146
- center, 86, 189
- cień, *Patrz:* cień
- cite, 85, 89, 104
- code, 85, 90, 104
- col, 136, 137, 146
- colgroup, 136, 137, 146
- command, 187
- data, 85, 92, 104
- datalist, 157, 170, 176, 187
- dd, 75, 76, 104
- del, 85, 92, 104
- details, 76, 187
- dfn, 85, 90, 104
- dir, 86, 189
- div, 79, 80, 94, 99, 104, 124, 149
- dl, 75, 76, 104, 134
- dopełnienie, *Patrz:* dopełnienie
- dt, 75, 76, 104
- dzielący na sekcje, 81
- em, 60, 66, 85, 86, 104, 215
- embed, 187, 196
- fieldset, 76, 161, 171, 172, 173, 176, 435
- figcaption, 76, 78, 104, 127, 143, 187
- figure, 76, 78, 104, 127, 143, 187
- font, 86, 189
- footer, 79, 80, 83, 104, 187
- form, 149, 150, 166, 176
- frame, 189
- frameset, 189
- główny, *Patrz:* element html
- grupujący treść, 76
- h1, *Patrz:* nagłówek
- head, 56, 68, 97, 215, 463
- header, 79, 80, 83, 104, 187
- hgroup, 73, 80, 104, 187
- hr, 61, 72, 76, 104
- html, 56, 68, 97
- i, 85, 87, 88, 104
- iframe, 130, 132
- img, 61, 115, 123, 124, 125, 127, 132, 215, 284, 411

## element

input, 153, 154, 155, 158, 161, 165, 166, 176, 187  
 ins, 85, 92, 104  
 isindex, 86, 189  
 isindex, 189  
 kbd, 85, 90, 104  
 keygen, 170, 176, 187  
 kolejność układania na stosie, 367  
 krawędź wewnętrzna, 305, 306  
 krawędź zewnętrzna, 305, 306  
 label, 153, 161, 164, 171, 173, 176  
 legend, 171, 172, 176, 435  
 li, 73, 74, 104, 107, 124, 244, 245, 259, 276  
 liniowy, 59, 60, 70, 77, 84, 85, 94, 124, 125, 274, 306, 307  
   margines, 330  
   pływający, 344, 345  
 link, 300, 301  
 margines, 305, 306, 328, 343  
   domyślny, 328  
   składanie, 330, 345  
   ujemny, 331, 388  
 mark, 85, 90, 104, 187  
 menu, 86, 187  
 meta, 56, 61, 68, 97, 445  
 meta charset, 56  
 meter, 170, 176, 187  
 nagłówek, *Patrz:* nagłówek  
 narożniki eliptyczne, 322, 323  
   zaokrąglone, 320, 323  
 nav, 79, 80, 81, 82, 104, 187, 494  
 niezastępowany, 125  
 noframes, 189  
 object, 196  
 ol, *Patrz:* lista uporządkowana  
 optgroup, 164, 176  
 option, 155, 157, 176  
 output, 170, 176, 187  
 p, 59, 70, 76, 104  
 param, 97, 196  
 pływający, 330, 341, 342, 344, 346, 349, 351, 354, 381  
 zrzućcie, 354

podział na kolumny, 381  
 pojemnik, 305, 306  
   typ, 334  
 porzucony HTML 4.01, 184, 189  
 pozycjonowany bezwzględnie, 330, 347, 356, 359, 381, 392  
 pre, 76, 104  
 progress, 170, 176, 187  
 przepełnienie zawartości, 311  
 przestarzały, 86  
 pusty, 61, 72, 301  
 q, 77, 85, 89, 104  
 rp, 85, 93, 104, 187  
 rt, 85, 93, 104, 187  
 ruby, 85, 93, 104, 187  
 s, 85, 87, 88, 104  
 samp, 85, 90, 104  
 script, 97, 462, 499  
 section, 79, 80, 81, 104, 187, 494  
 select, 163, 176  
 semantyczny wewnętrzny, 84,  
   *Patrz też:* element liniowy  
 small, 85, 87, 88, 104  
 source, 187, 196, 197  
 span, 85, 94, 95, 96, 99, 104, 172  
 strike, 86, 189  
 strong, 85, 86, 104  
 style, 65, 97, 213, 272  
 sub, 85, 90, 104  
 summary, 187  
 sup, 85, 90, 104  
 table, 135, 136  
 tbody, 137, 146  
 td, 76, 135, 136, 138, 140, 142, 146  
 textarea, 154, 155, 176, 434  
 tfoot, 137, 146  
 th, 135, 136, 137, 138, 140, 142, 146  
 thead, 137, 146  
 time, 85, 91, 104, 187  
 title, 56, 58, 68, 97  
 tr, 135, 146  
 track, 187  
 transformacja, *Patrz:* transformacja  
 tt, 86, 189  
 u, 85, 87, 88, 104

ul, *Patrz:* lista nieuporządkowana  
 var, 85, 104  
 video, 187, 190, 191, 195, 196, 198  
 wbr, 85, 93, 104, 187  
 wewnętrzny, *Patrz:* element liniowy  
 wymiary, 306, 307, 309, 310  
 wyświetlający czas, 42  
 zastępowany, 125  
   margines, 331  
   zawartość, 305, 306  
 element box, *Patrz:* element pojemnik,  
   model pojemnika  
 element type selector, *Patrz:* selektor  
   typu elementu  
 em, 234, 235, 313, 328, 356, 379  
 encja, 99  
 end tag, *Patrz:* znacznik zamykający  
 escape character, *Patrz:* znak ucieczki  
 event, *Patrz:* zdarzenie  
 event binding, *Patrz:* zdarzenie wiązanie  
 explicit animation, *Patrz:* animacja  
   klatkowa  
 eXtensible Markup Language, *Patrz:*  
   XML

## F

falsywe kolumny, 395, 396, 397  
 faux columns, *Patrz:* fałszywe kolumny  
 Ferraiuolo Eric, 272  
 Firefogg, 194  
 firewall, 23  
 Fireworks, 8, 17, 508, 514, 521, 534, 550, 553  
   kompresja, 549  
 fixed layout, *Patrz:* strona układ sztywny  
 fixed positioning, *Patrz:* pozycjonowanie  
   sztywne  
 Flanagan David, 463  
 Flash, 10, 11, 181, 191, 198  
 FlashCanvas, 199  
 Flexbox, 381, 395  
 Flickr, 509  
 float drop, 354  
 folder, 108

foreground, *Patrz:* pierwszy plan

format

audio, 193

BMP, 123, 510

Embedded Open Type, 228

EOT, 228

EPS, 123, 510

GIF, 123, 510, 511, 513, 514, 515, 520,

526, 528, 548

optymalizacja, 542, 543, 544, 545,  
546, 547

html, 53

JPEG, 123, 510, 511, 515, 520, 548

optymalizacja, 542, 547, 550

progresywny, 516

JPG, *Patrz:* format JPEG

OpenType, 228

OTF, 228

PNG, 123, 510, 511, 517, 518, 519,

520, 526, 528, 531, 548

optymalizacja, 552

strumieniowy, 11

SVG, 510, 532, 534, 535, 536

animacja, 537

XML, 534

TIFF, 123, 510

TrueType, 228

TTF, 228

Web Open Font Format, 228

webowy, 123

wideo, 192

WOFF, 228

formularz, 13, 76, 147, 171, 434, 436

aplikacja, *Patrz:* formularz skrypt

dostępność, *Patrz:* dostępność

formularza

kontrolka, 147, 151, 152, 153

grupa, 172

identyfikator, 171

ukryta, 166

menu rozwijane, *Patrz:* menu

rozwijane

pole tekstowe, *Patrz:* pole tekstowe

przycisk, *Patrz:* przycisk

skrypt, 147, 149

styl, 175, 434

układ, 173

zmienna, 151, 152

Friedman Vitaly, 247

frontend, 9

Frost Brad, 34, 369, 451, 454

FTP, 18, 19, 21

funkcja, 473, 474

addEventListener, 479, 480

alert, 464, 473, 478

argument, 473, 475

confirm, 464, 478

obsługi zdarzeń, 478

prompt, 464

window

close, 478

focus, 478

własna, 474

zwracanie wartości, 475

## G

gamma, 518

Gardner Lyza, 40, 453, 454

Garrett James, 497

general sibling selector, *Patrz:* selektor

dowolnego rodzeństwa

generated content, *Patrz:* zawartość

generowana

generic font family, *Patrz:* krój pisma

rodzina gatunkowa

geolokacja, 191

Getty Images, 509

GIF, *Patrz:* GIF

GIMP, 17, 19, 508

global scope, *Patrz:* zmienna zakres

globalny

gniazdo, 191

Google, 37

graceful degradation, *Patrz:* degradacja

z wdziękiem

gradient, 37, 272, 295

generator, 299, 300

liniowy, 296

projektowanie, 299

promienisty, 296, 297

przeglądarki, 298

grafika, 123, 132

bitmapowa, 522

clipart, *Patrz:* klipart

dostępność, 127

elastyczna, 447

format pliku, *Patrz:* format

optymalizacja, 541, 542, 543, 546, 547,

551, 552, 553, 554

rastrowa, *Patrz:* grafika bitmapowa

skalowanie, 128

w tle, 123, 124, 284, 286, 288, 292,  
294

paralaksa ruchu, *Patrz:* paralaksa

ruchu

webowa, 507

wektorowa, 11, 522, 533

wymiary, 128, 132

zewnętrzna, 123

źródła, 508

Graphic Interchange Format, *Patrz:*

format GIF

grid, *Patrz:* siatka

Griffiths Patrick, 378

Grigsby Jason, 40, 453

Gustafson Aaron, 38, 158, 396, 430

## H

Handbrake, 194

handheld, *Patrz:* media

Hawryluk Zoltan, 256, 336

Hay Stephen, 35, 381

Hayes Paul, 418

heksadecymalny,

Hickson Ian, 184

hiperłącze, 21, 22, 26, 105

do fragmentu w innym dokumencie,  
118

do katalogu wyżej w hierarchii, 112

do określonego miejsca na stronie, 116

do podkatalogu, 110

do strony internetowych, 107

pocztowe, 119

hiperłącze

- telefoniczne, 120
  - w nowym oknie przeglądarki, 118
  - w ramach jednego katalogu, 109
  - zewnętrzne, 107
- HSL, 269
- HSLa, 271
- HTML, 8, 12, 26, 49, 50, 55, 119, 485
- historia, 182
  - znacznik, *Patrz:* znacznik
- HTML5, 11, 12, 56, 80, 179, 180, 181, 185, 187
- HTML5 Shiv, 494
- HTTP, 11, 21, 24
- hybrid layout, *Patrz:* strona układ
- hybrydowy
- hypertext links, *Patrz:* hipertekst
- HyperText Markup Language, *Patrz:* HTML

I

- ID selector, *Patrz:* selektor
- identyfikatora
- identyfikator, 95
- agenta użytkownika, 453
- IIFE, 477
- IIS, 22
- Illustrator, 17, 19, 508
- ilustracja, 76, 78, *Patrz też:* element
- figure, element figcaption
  - źródła, 508
- image replacement technique, *Patrz:* tekst zastępowanie obrazkiem
- implicit animation, *Patrz:* animacja
- niejawna
- Independently Invoked Functional Expression, *Patrz:* IIFE
- informacje kontaktowe do autora dokumentu, 84
- Information Architect, *Patrz:* architekt informacji
- Inkscape, 536
- inline element, *Patrz:* element liniowy
- inline style, *Patrz:* CSS wewnętrzny

- inner edge, *Patrz:* element krawędź
- wewnętrzna
- instrukcja, 464
- if, 469
  - if/else, 469, 470
  - pętli, *Patrz:* pętla
  - warunkow, 469
- Interaction Design, *Patrz:* projektowanie
- interakcji
- interaktywność, 11, 13, 461
- interfejs, 461
- API, 184, 185, 189, 191, 485
  - Canvas API, 200
  - osi czasu, 421
  - użytkownika, 5, 6, 10, 498
  - widżet, 13
  - WYSIWYG, 16
- interlace, *Patrz:* przeplot
- interlinia, 235
- Internet Explorer 8, 80
- intranet, 23
- IPv4, 22
- IPv6, 22
- Irish Paul, 310, 494
- IStockphoto, 509
- IXD, *Patrz:* projektowanie interakcji

J

- Java, 13, 460
- JavaScript, 9, 11, 29, 37, 119, 459, 460
- biblioteka, *Patrz:* biblioteka
  - manifest, 470
- JavaScript Object Notation, *Patrz:* JSON
- JavaServer Pages, 150
- jednostka miary, 234
- bezwzględna, 234
  - względna, 234
- Jehl Scott, 38, 312, 496, 497
- Jensen Scott, 35
- język
- dynamiczny, 459, 460
  - o słabym typowaniu, 459, 460
  - skryptowy, 13
  - działający po stronie klienta, 459

- XML, *Patrz:* XML
- znaczników, *Patrz:* HTML
- Johansson Roger, 81
- JSON, 497

K

- kanal
- alpha, 271, 518, 526
  - RSS, *Patrz:* RSS
- katalog, 25, 108
- główny, 114
- Kellum Scott, 429, 523
- keyframe, klatka kluczowa
- klasa, 95, 98, 187, 244, 246, 400, 432, 500, 583
- klatka kluczowa, 420, 421
- Kleinfeld Sanders, 200
- klient, 23
- klip wideo, 78, 195
- klipart, 509
- klucz, 170
- Koblentz Thierry, 447
- kodek, 192
- kolekcja, 487
- kolor
- HSL, *Patrz:* HSL
  - indeksowany, 511, 513, 517
  - nazwa, 243, 266, 267
  - obramowania, *Patrz:* obramowanie
  - kolor
  - paleta, 273, 512, 513
  - pierwszego planu, 243, 272
  - RGB, *Patrz:* RGB
  - tekstu, *Patrz:* tekst kolor
  - tła, 243, 272, 273, 274
  - webowy, 273
  - wybieranie, 169
- kombinator, *Patrz:* selektor potomny
- komentarz, 54, 60, 213, 464
- CSS, *Patrz:* CSS komentarz
  - jednowierszowy, 465
  - warunkowy, 271, 272, 310
  - wielowierszowy, 465
- kompilator, 460

kompresja, 511, 513  
     bezstratna, 513, 517, 546  
     JPEG, 515, 547  
     stratna, 515  
 kontekst pozycjonowania, *Patrz:*  
     pozycjonowanie blok zawierający  
 koszyk na zakupy, 13, 14  
 kotwica, 105  
 krój pisma  
     bezszerzeryfowy, 71, 229, 230, 231  
     dekoracyjny, 230  
     format  
         Embedded Open Type, 228  
         konwersja, 228  
         OpenType, 228  
         TrueType, 228  
         Web Open Font Format, 228  
 kapitaliki, 241  
 o stałej szerokości znaków, 77, 90, 230, 231  
 osadzanie, 229  
 pisanka, 230  
 pochylenie, 239  
 rodzina, 226  
 rodzina gatunkowa, 229  
 stos, 226, 231  
 styl, 239  
 szeryfowy, 71, 229, 230, 231  
 udostępnianie, 228  
 waga, 239  
 wariant, 240  
 wielkość, 233, 234, 238  
 słowo kluczowe, 235

## L

Lawson Bruce, 182  
 Lennartz Sven, 431  
 LESS, 303, 433  
 licencja  
     Creative Commons, 509  
     rights-managed, *Patrz:* licencja  
         wyłączna  
     royalty-free, 509  
     wyłączna, 508

linia pozioma, 72  
 liquid layout, *Patrz:* strona układ płynny  
 lista  
     definicji, 73, 75  
     katalogowa, 86  
     nienumerowana, *Patrz:* lista  
         nieuporządkowana  
     nieuporządkowana, 73, 74, 82, 86, 93, 104, 259, 350, 351  
     numerowana, *Patrz:* lista  
         uporządkowana  
     uporządkowana, 73, 74, 104, 149  
 local scope, *Patrz:* zmienna zakres  
     lokalny  
 loop, *Patrz:* pętla  
 Lovitt Michael, 527

## Ł

łącze hipertekstowe, *Patrz:* hiperłącze

## M

MacDonald Matthew, 182  
 mailto, 119  
 Marcotte Ethan, 40, 376, 444  
 margin, *Patrz:* element margins  
 Marquis Mat, 454, 459, 485  
 maszyna wirtualna, 15  
 mathematical operator, *Patrz:* operator  
     matematyczny  
 May Matt, 43  
 media, 448  
     zapytanie, 448, 449  
 menu, 86  
     grupa opcji, 164  
     kontekstowe, 86  
     pasek, 42  
     poziome, 349  
     przewijane, 163  
     rozwijane, 13, 147, 151, 152, 163, 278  
 menubar, *Patrz:* menu pasek  
 metadane, 97  
 metajęzyk, 183  
 metoda

appendChild, 491  
 createElement, 490  
 createTextNode, 491  
 GET, 150, 151  
 getAttribute, 489  
 getElementById, 488  
 getElementsByClassName, 488  
 getElementsByTagName, 487  
 innerHTML, 490  
 insertBefore, 492  
 POST, 150, 151  
 querySelectorAll, 488  
 ready, 500  
 removeChild, 492  
 replaceChild, 492  
 setAttribute, 489  
 style, 490  
 Meyer Eric, 247, 299, 427  
 Microsoft Expression Web, 16, 19, 115, 222  
 Microsoft Internet Information Services, *Patrz:* IIS  
 miejsce docelowe, 116  
 mieszanica, 433  
 mikroformat, 96, 97  
 Miro Video Converter, 194  
 mixin, *Patrz:* mieszanica  
 Mobitest, 44  
 model  
     border-box, 309  
     content-box, 292, 307, 311  
     pojemnika, 220, 305  
     IE, 310, 311  
 Modernizr, *Patrz:* biblioteka Modernizr  
 modular style sheet, *Patrz:* CSS  
     modularny  
 Moll Cameron, 377  
 MP3/WMA/Ogg Converter, 194  
 MSDN, 381  
 multimedia, 10

## N

nagłówek, 59, 70, 76, 79, 83, 386, *Patrz też*: element header  
 grupa, 72  
 łączenie z treścią tabeli, 137  
 tabeli, 138  
 zapytanie o media, 449  
 nawigacja, 79, *Patrz też*: element nav  
 NCSA Mosaic, 22  
 niepełnosprawność, *Patrz*: użytkownik  
 niepełnosprawny  
 node, *Patrz*: drzewo węzeł  
 node list, *Patrz*: drzewo węzeł lista  
 non-replaced, *Patrz*: element  
 niezastępowany  
 Norman Donald, 5  
 notacja funkcjonalna, 284  
 Notatnik, 50, 51  
 Notepad, 50

## O

O Connor Joshue, 43  
 obiekt  
 document, 487  
 window, 478, 479  
 XMLHttpRequest, 497  
 obramowanie, 72, 202, 272, 289, 305,  
 306, 316  
 kolor, 202, 243, 272, 273, 319  
 styl, 316  
 szerokość, 318  
 obrazek, *Patrz*: grafika, ilustracja  
 obrys, *Patrz*: obramowanie  
 odsyłacz, *Patrz*: hiperłącze  
 offset, *Patrz*: pozycjonowanie  
 przesunięcie  
 okno w oknie, 130  
 operator  
 matematyczny, 469  
 porównania, 468, 469  
 outer edge, *Patrz*: element krawędź  
 zewnętrzna

## P

padding, *Patrz*: dopełnienie  
 paleta kolorów webowych, *Patrz*: kolor  
 webowy  
 pamięć podręczna, 126, 191  
 paralaksa ruchu, 295  
 parent, *Patrz*: rodzic  
 Parker Todd, 38  
 parsowanie, 28  
 pasek  
 menu, 42  
 nawigacyjny poziomy, 334, 353  
 postępu, 42  
 przewijania, 119, 130, 155, 295, 312  
 persona, *Patrz*:  
 Peter Beverloo, 299  
 pętla, 471  
 for, 471  
 Photoshop, 7, 17, 273, 507, 514, 525, 553  
 kompresja, 549  
 Próbnik kolorów, 268, 521  
 Photoshop Elements, 17, 19  
 PHP, 9, 13, 150, 459  
 pica, 234  
 pierwszy plan, 272  
 piksel, 234, 236, 313, 328, 363, 522, 523  
 referencyjny, 523  
 plik  
 dźwiękowy, 198  
 graficzny, 28  
 graficzny, 53  
 indeksu, 25  
 rozszerzenie, 53  
 wideo, 194  
 pływanie, *Patrz*: element pływający  
 PNG, *Patrz*: format PNG  
 PNGcrush, 519  
 podkładka, 462  
 pojemnik elementu, *Patrz*: element  
 pojemnik, model pojemnika  
 pole  
 daty i czasu, 152, 167  
 tekstowe, 147, 151, 152, 434  
 adresu e-mailowego, 156

hasło, 155  
 jednowierszowe, 154, 155  
 okalizacji, 156  
 numeru telefonu, 156  
 wielowierszowe, 154  
 wyspecjalizowane, 152, 155  
 wyszukiwania, 156  
 wartości liczbowych, 152, 168  
 wyboru, 152, 435  
 koloru, 152, 169  
 wartości liczbowej z danego  
 zakresu, 42  
 wyszukiwania, 86  
 polyfill, *Patrz*: wypełniacz  
 positioning context, *Patrz*:  
 pozycjonowanie blok zawierający  
 potomek, 215  
 Powers Shelley, 538  
 powiązanie  
 jawne, 171  
 niejawne, 171  
 pozycjonowanie, 341, 356, *Patrz też*:  
 element pozycjonowany  
 bezwzględne, 356, 359, 381, 392  
 blok zawierający, 360, 361, 364  
 przesunięcie, 357, 363  
 statyczne, 356  
 sztywne, 357, 368  
 względne, 356, 358  
 prawo autorskie, 88, 102, 507, 508  
 prezentacja, 12, 29  
 print, *Patrz*: media  
 program  
 do projektowania stron internetowych,  
 7, 17, 18, 19  
 do transferu plików, *Patrz*: FTP  
 progressbar, *Patrz*: pasek postępu  
 progressive enhancement, *Patrz*:  
 stopniowe ulepszanie  
 projection, *Patrz*: media  
 projektant informacji, *Patrz*: architekt  
 informacji  
 projektowanie, 5  
 doznań użytkownika, 5, 10  
 graficzne, 7



- interakcji, 5
- skoncentrowane na potrzebach użytkowników, 6
- wizualne, *Patrz.*: projektowanie graficzne
- protokół
  - HTTP, 11, 21, 24
  - HTTPS, 24
- przedrostek producenta przeglądarki, 298, 299
- przeglądarka, 18, 19, 23, 126, 127, 128, 478
  - błędy, 342
  - Chrome, 44
  - desktopowa, *Patrz.*: przeglądarka graficzna
  - graficzna, 22, 23
  - informacje ignorowane, 54, 77
  - mobilna, 18, 19, 23
  - nowe okno, 118, 119
  - obsługa formatu
    - audio, 194
    - wideo, 193
  - producent, 298, 299
  - Safari, 156, 157
  - w JavaScript, 478
  - wojna, 182, 493
- przejsięcie, 399, 400, 406, 407
- przeplot, 514, 518
- przezroczystość, 271, 275, 286, 514, 518, 526, 528
  - binarna, 526, 527
  - kanału alfa, 526, 527, 531
  - w IE, 271, 275, 276, 518, 527
- przodek, 215
  - pozycjonowany, 360
- przycisk, 147, 151, 152, 434
  - obrazkowy, 158
  - opcji, 161, 435
  - pola wyboru pliku, 161, 165
  - reset, 158
  - submit, 158, 434
  - uniwersalny, 158
- pseudoclass selector, *Patrz.*: selektor pseudoklasy

- pseudoelement, 279
  - :after, 280
  - :before, 280
  - :first-letter, 280
  - :first-line, 280
- pseudoklasa, 276, 279, 584, *Patrz też.*: selektor
  - akcji użytkownika, 277
  - dynamiczna, *Patrz.*: pseudoklasa dynamiczna
  - odnośnika, 276
- PSPad, 50
- punkt
  - graniczny, 452
  - typograficzny, 234
- PuTTY, 18, 19
- Python, 9, 13, 150

## Q

- Quartarolo Tony, 431

## R

- radio button, *Patrz.*: przycisk opcji
- ramka pływająca, 130
- reference pixel, *Patrz.*: piksel referencyjny
- regiony CSS, 381
- reklama interaktywna, 131
- relative positioning, *Patrz.*: pozycjonowanie względne
- replaced element, *Patrz.*: element zastępowany
- Resig John, 498
- Respond.js, 496
- Responsive Web Design, 38, 39, 40, 310, 374, 376, 444, 448
- Retina, 430, 449, 453, 517, 522, 523, 524, 550
- RGB, 169, 243, 265, 268, 515, 517, 520
  - wartości szesnastkowe, 266, 267, 269, 270
- RGBa, 271, 275
- Rieger Stephanie, 40
- Robinson Alex, 388

- Robinson Mike, 81
- rodzeństwo, 215
- rodzic, 215, 239
- rola wyświetlania, 333
- root directory, *Patrz.*: katalog główny
- rozdzielczość, 522
- roztrząsanie, *Patrz.*: dithering
- RSS, 14
- Ruby, 9, 13, 459
- Ruby on Rails, *Patrz.*: biblioteka Ruby on Rails
- rysunek, 78

## S

- SASS, 303, 433
- Scalable Vector Graphics, *Patrz.*: format SVG
- scope, *Patrz.*: zmienna zakres
- screen, *Patrz.*: media
- screen reader, *Patrz.*: czytnik ekranu
- SCSS, 433
- SeaMonkey, 16, 19
- sekcja, 76, 79, 80, 104, *Patrz też.*: element section
- Selectivizr, 279, 495
- selektor, 210, 211, 216, 276, 486, *Patrz też.*: pseudoelement, pseudoklasa
  - :active, 277, 400
  - :checked, 279
  - :disabled, 279
  - :empty, 279
  - :enabled, 279
  - :first-child, 279
  - :first-of-type, 279
  - :focus, 277, 400
  - :hover, 277, 400
  - :lang, 279
  - :last-child, 279
  - :last-of-type, 279
  - :link, 276
  - :not, 279
  - :nth-child, 279
  - :nth-last-child, 279
  - :nth-last-of-type, 279

## selektor

- :nth-of-type, 279
- :only-child, 279
- :only-of-type, 279
- :root, 279
- :target, 279
- :visited, 276
- atrybutów, 281, 282
- dowolnego rodzeństwa, 245
- dziecka, 245
- grupowanie, 220, 276
- identyfikatora, 244, 245, 247, 276
- interfejsu użytkownika, 279
- klasy, 244, 246, 247, 276
- kontekstowy, 244, 245, 247
- potomny, 244, 276
- przylegającego rodzeństwa, 245
- pseudoelementu, 279
- pseudoklasy, 276, 279
- strukturalny, 279
- typu elementu, 211, 218
- uniwersalny, 246, 276

separator treści, 28

server-side, 23

serwer, 21, 22

- Apache, *Patrz:* Apache
- IIS, *Patrz:* IIS
- przesyłanie danych
  - GET, 150, 151
  - POST, 150, 151

SGML, 183, 185

Sha Thomasa, 498

Shaefer Christian, 310

Sharp Remy, 80, 182, 393, 494

Shea David, 208

shim, *Patrz:* podkładka

siatka, 375, 381

Silverlight, 181, 192

skanowanie, 508, 509

skrót, 89

skrypt, 461, 463

- box-sizing, 310
- DOM, *Patrz:* DOM
- osadzony, 463
- Overthrow, 312
- po stronie serwera, 9, 11
- polyfill, 279
- tworzenie, 500
- zewnątrzny, 463

slider, *Patrz:* pole wyboru wartości liczbowej z danego zakresu

Sloppy, 44

Slowly, 44

słowo kluczowe

- return, 475
- var, 465, 476, 477

Souders Steve, 44

spam, 120

specyficzność, 218, 247

sprite, 430, 431

- generator, 432

Squarespace, 4

standard żyjący, 184

start tag, *Patrz:* znacznik otwierający

statement, *Patrz:* instrukcja

static positioning, *Patrz:* pozycjonowanie statyczne

Stephenson Sam, 498

stopka, 79, 83, 386, 392, *Patrz też:*

- element footer

stopniowe ulepszanie, 33, 36

storyboard, 7

strateg zawartości, 10

strona

- czysta, 427
- fałszywe kolumny, *Patrz:* fałszywe kolumny
- grafika, 61, 76
- kolumna, *Patrz:* fałszywe kolumny, strona układ wielokolumnowy
- tło, 395
- pełne powiększenie, 378
- prezentacja, 58
- szablon, 380
- szkic, 6
- tworzenie, 49
- tytuł, 58
- układ, 373, 379, 381
  - adaptacyjny, 446
  - elastyczny, 373, 377, 379
  - hybrydowy, 373, 379

- płynny, 373, 376, 377, 382, 386, 392, 396, 445, 446
- pozycjonowany, 392
- szywny, 373, 374, 375, 384, 385, 394, 395
- wielokolumnowy, 373, 380, 381, 382, 384, 385, 386, 392, 394, 396, 397
- źródło, 26

Style Tiles, 8

subdomena, 24

- m., 40

Sublime Text, 17, 19

syndykacja, 81

Syntactically awesome style sheets, *Patrz:* SASS

system siatki, *Patrz:* siatka

szeryfowa, czcionka,

## Ś

ścieżka, 108

- określana względem katalogu głównego, 114

## T

tabela, 78, 133, 324, 441

- dostępność, *Patrz:* dostępność tabel
- kolumny zakres, 140
- komórka, 76, 135
  - pusta, 443
  - rozmiar, 142
  - zakres, 139
- nagłówek, 138
- struktura, 135, 137
- styl, 138, 441
- wiersza zakres, 141

tablica, 468

- element, 468

technika

- CSS Sprites, 430
- Kelluma, 429
- zastępowania tekstu obrazkiem, *Patrz:* tekst zastępowanie obrazkiem
- zerowania stylów CSS, 427



teczka, 108  
 tekst  
   alternatywny, 42, 124, 126  
   anonimowy, 70  
   cień, 254, 263  
   formatowanie, 225, 263  
   goły, 70  
   kolor, 243  
   sformatowany, *Patrz:* element pre  
   wyrównanie, 251, 263  
   zastępowanie obrazkiem, 429  
   zawijanie, 347  
   zmiana stylu wiersza, 249, 250, 257  
 Telnet/SSH, 18, 19  
 test z udziałem użytkowników, 6  
 TextEdit, 52  
 TextMate, 50  
 TextPad, 16, 19  
 timer, *Patrz:* element wyświetlający czas  
 tło, *Patrz:* grafika w tle, kolor tła,  
   właściwość background, właściwość  
   background-image  
 Toland Patty, 38  
 tooltip, *Patrz:* wskazówka  
 touch event, *Patrz:* zdarzenie dotyku  
 transformacja, 399, 410, 412, 413, 414, 415  
   dwuwymiarowa, 410  
   kąta, 411  
   płynna, 416  
   trzywymiarowa, 418, 426  
   własna, 411  
 transformation, *Patrz:* transformacja  
 transition, *Patrz:* przejście  
 TRBL, 313, 314  
 tty, *Patrz:* media  
 Tuck Michael, 231  
 Tumblr, 4  
 tv, *Patrz:* media  
 tweening, *Patrz:* animacja automatyczna

## U

UCD, *Patrz:* projektowanie  
   skoncentrowane na potrzebach  
   użytkowników

UI, *Patrz:* interfejs użytkownika  
 URI, 106  
 URL, 106  
 URN, 106  
 urządzenie  
   dotykowe, 277, 278, 312  
   mobilne, 33, 34, 35, 39, 295, 336, 369,  
   445, 450  
 user agent style sheet, *Patrz:*  
   wyświetlanie domyślne  
 user agent style sheets, *Patrz:* arkusze  
   stylów agenta użytkownika  
 User Centered Design, *Patrz:*  
   projektowani skoncentrowane na  
   potrzebach użytkowników  
 User Experience, *Patrz:* projektowanie  
   doznań użytkownika  
 User Interface, *Patrz:* interfejs  
   użytkownika  
 UX, *Patrz:* projektowanie doznań  
   użytkownika  
 użytkownik, 5, 6, 7, 13  
   niepełnosprawny, 41, 42

## V

Veer, 509  
 viewport, *Patrz:* widok  
 Virtual Machine, maszyna wirtualna  
 Visscher Sjoerd, 494

## W

W3C, 11, 22, 36, 41, 42, 67, 69, 106, 137,  
   182, 183, 190, 376, 381  
 Wachs Maggie Costello, 38  
 WAI, 41  
 walidator, 67, 186  
 warstwa  
   prezentacji, 13, 29, 37, 58, 64, 81, 123,  
   124, 133, 142, 182, 185, 209  
   struktury, 13, 209  
   zachowania, 13  
 WaSP, 183  
 Web Accessibility Initiative, *Patrz:* WAI

Web Development, 8  
 Web Hypertext Application Technology  
   Working Group, *Patrz:* WHATWG  
 Web Standards Project, *Patrz:* WaSP  
 WebPagetest, 44  
 Weizenbaum Nathan, 433  
 Weyl Estelle, 420  
 WHATWG, 69, 91, 92, 93, 104, 184,  
   190  
 wideo, 192  
 widok, 445  
 widżet, 131  
 wiersz  
   długość optymalna, 374  
   poleczeń, 18, 19  
 witryna  
   diagram, 6  
   dla niepełnosprawnych, *Patrz:*  
   użytkownik niepełnosprawny  
   optymalizacja, 43, 44  
   wersja mobilna, 40  
 właściwość, 210, 211, 304  
   animation, 422, 426  
   background, 280, 293, 304  
   background-attachment, 291, 304  
   background-clip, 292, 304  
   background-color, 273, 285, 294, 304,  
   401  
   background-image, 284, 294, 296, 304  
   background-origin, 292, 304  
   background-position, 288, 292, 304,  
   401  
   background-repeat, 286, 289, 304  
   background-size, 292, 304  
   border, 280, 320, 338  
   border-collapse, 324, 441, 442, 456  
   border-color, 273, 319, 338, 401  
   border-image, 296, 323, 324, 325, 338  
   border-radius, 320, 321, 338  
   border-spacing, 442, 456  
   border-style, 316, 317, 338  
   border-width, 318, 338  
   box-shadow, 335, 338  
   box-sizing, 307, 310, 338  
   clear, 347, 371

właściwość

color, 243, 272, 273, 280, 304, 401  
column-count, 381  
display, 260, 333, 339, 411, 441  
    none, 334  
dziedziczenie, 216  
empty-cells, 443, 456  
float, 280, 342, 343, 371  
font, 241, 263, 280  
font-family, 226, 263  
font-size, 233, 234, 235, 237, 263, 401  
font-size-adjust, 257  
font-style, 240, 263  
font-variant, 241, 263  
font-weight, 239, 263, 401  
height, 306, 307, 339, 402  
letter-spacing, 253, 280, 401  
line-height, 249, 263, 280, 401  
list-style, 261  
list-style-image, 261, 296  
list-style-position, 260  
list-style-type, 259  
margin, 280, 328, 339, 402  
opacity, 275, 304, 401  
outline, 380, 401  
overflow, 311, 339  
padding, 280, 312, 313, 339, 402  
perspective, 426  
position, 356, 369, 371  
table-layout, 441  
text-indent, 250, 429  
text-align, 251, 263  
text-align-last, 251  
text-decoration, 252, 263, 280  
text-direction, 257, 263  
text-indent, 250, 263, 401  
text-justify, 251  
text-shadow, 254, 263, 299, 401  
text-transform, 252, 263, 280  
transform, 426  
transform-origin, 411  
transition, 406, 426  
transition-delay, 400, 404, 426  
transition-duration, 400, 401, 426  
transition-property, 400, 401, 426

transition-timing-function, 400, 402, 426  
unicode-bidi, 257, 263  
vertical-align, 257, 263, 280, 401  
visibility, 257, 263, 401  
white-space, 257, 263  
width, 306, 307, 339, 345, 346, 402  
word-spacing, 253, 263, 280, 401  
z-index, 367, 371, 402  
WordPress, 4  
World Wide Web Consortium, *Patrz:* W3C  
Wroblewski Luke, 35, 173  
wskazówka, 42  
wydajność, 6, 43, 44, 430  
wykluczenia CSS, 381  
wykres, 134  
wypełniacz, 462, 493, 494  
    HTML5 Shim, *Patrz:* HTML5 Shiv  
    HTML5 Shiv, *Patrz:* HTML5 Shiv  
    Modernizr, *Patrz:* biblioteka Modernizr  
    Respond.js, *Patrz:* Respond.js  
    Selectivizr, *Patrz:* Selectivizr  
wyszukiwarka, 37, 70  
wyświetlanie domyślne, 217

X

XHTML, 12, 14, 55, 183  
    składnia, 183  
XML, 14, 183, 485  
    serializacja dla HTML5, 185  
    SVG, 534

Y

Young Zebulon, 431  
YSlow, 44

Z

zagnieżdżanie, 57, 89  
zaporą sieciową, *Patrz:* firewall  
zawartość generowana, 280

zdarzenie, 478  
    atrybut HTML, 479  
    dotyku, 462  
    metoda przyłączona do elementu, 479, 480  
obsługa, 479, 480  
onblur, 479  
onchange, 479  
onclick, 478, 479  
onerror, 479  
onfocus, 479  
onkeydown, 479  
onkeypress, 479  
onkeyup, 479  
onload, 478, 479  
onmousedown, 479  
onmousemove, 479  
onmouseover, 478, 479  
onmouseup, 479  
onsubmit, 479  
wiązanie, 478  
Zeldman Jeffrey, 36, 429  
zmienna, 465  
    zakres, 476  
        globalny, 476, 477  
        lokalny, 476, 477  
znacznik, 26, 54, 55, 185  
    końcowy, *Patrz:* znacznik zamykający  
    otwierający, 55  
    początkowy, *Patrz:* znacznik otwierający  
    semantyczny, 58  
    tekstowy, 69  
    widoku meta, 445  
    zamykający, 55, 65, 66  
znak  
    #, 245, 270  
    \$, 433  
    &, 99, 100  
    \*, 246  
    ./, 112  
    ./, 115  
    /\*, 465  
    //, 465  
    :, 276, 279

---

::, 279  
„”, 99, 464  
@, 433  
£, 100  
¥, 100  
€, 100  
<, 99, 100  
>, 100  
©, 99, 100  
®, 100  
..., 100  
apostrofu, 100  
biały, 77, 211, 464  
cudzysłowu, 62, 89  
kropki, 246  
pauzy, 100  
półpauzy, 100  
spacji twardej, 100  
specjalny, 99  
™, 100  
ucieczki, 99  
wypunktowania, 73, 100

## Ż

żądanie HTTP, 44



# PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION

- 
1. ZAREJESTRUJ SIĘ
  2. PREZENTUJ KSIĄŻKI
  3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW  
w działający bankomat!

**Dowiedz się więcej i dołącz już dzisiaj!**

<http://program-partnerski.helion.pl>

GRUPA WYDAWNICZA

 **Helion SA**

# Przewodnik dla początkujących projektantów WWW!

Marzy Ci się własna strona internetowa lub profesjonalna witryna dla Twojej firmy? Jeżeli brak Ci odpowiednich umiejętności, to zadanie może być poza Twoim zasięgiem. Ale do czasu! Ta książka zawiera dawkę wiedzy, która nawet kompletnemu laikowi pozwoli zaprojektować i zbudować witrynę WWW. Dzięki licznym przykładom przebrniesz przez kolejne etapy tworzenia strony internetowej oraz odkryjesz w sobie pasję webmastera!

Kolejne wydanie tej cenionej książki zostało gruntownie przeredagowane i uzupełnione o nowe informacje, tak aby prezentowało obecnie wykorzystywane narzędzia i aktualne trendy w projektowaniu stron internetowych. W trakcie lektury poznasz składnię języka HTML, wykorzystywane znaczniki oraz nowości zawarte w HTML5. Ponadto zaznajomisz się z kaskadowymi arkuszami stylów (CSS) oraz przekonasz się, jaki potencjał ma język JavaScript. Nauczysz się także przygotowywać odpowiednią grafikę na potrzeby strony WWW. Książka ta jest doskonałym przewodnikiem wprowadzającym w świat projektowania!

## Dowiedz się:

- jak wykorzystać arkusze stylów CSS
- dlaczego należy odpowiednio przygotować grafikę na stronę WWW
- do czego wykorzystać język JavaScript
- jak zbudować pierwszą stronę WWW

**helion.pl**  
księgarnia  
internetowa

Nr katalogowy: 16909



Księgarnia internetowa:  
**<http://helion.pl>**



Zamówienia telefoniczne:

**0 801 339900**



**0 601 339900**

Informatyka w najlepszym wydaniu

O'REILLY



**Helion**

Sprawdź najnowsze promocje:

• <http://helion.pl/promocje>

Książki najchętniej czytane:

• <http://helion.pl/bestsellery>

Zamów informacje o nowościach:

• <http://helion.pl/nawosci>

Helion SA

ul. Kościuszki 1c, 44-100 Gliwice

tel.: 32 230 98 63

e-mail: [helion@helion.pl](mailto:helion@helion.pl)

<http://helion.pl>

sięgnij po WIĘCEJ



KOD KORZYŚCI

ISBN 978-83-246-6667-6



9 788324 666676

cena: 99,00 zł