



**SMASHING**  
MAGAZINE



**Bill Sanders**

# PODRĘCZNIK HTML5

**NIEZBĘDNY PODRĘCZNIK DLA WEBMASTERÓW I PROGRAMISTÓW,  
KTÓRZY CHCĄ SZYBKO ZACZAĆ PRACĘ Z HTML5!**

**SMASHING MAGAZINE**

- Gruntowny przegląd nowych znaczników, atrybutów i wartości w HTML5
- Projektowanie struktury i nawigacji strony oraz korzystanie z grafiki, plików audio i wideo
- Sposoby na poprawne wyświetlanie witryny w różnych przeglądarkach

## » Idź do

- Spis treści
- Przykładowy rozdział
- Skorowidz

## » Katalog książek

- Katalog online
- Zamów drukowany katalog

## » Twój koszyk

- Dodaj do koszyka

## » Cennik i informacje

- Zamów informacje o nowościach
- Zamów cennik

## » Czytelnia

- Fragmenty książek online

## » Kontakt

Helion SA  
ul. Kościuszki 1c  
44-100 Gliwice  
tel. 32 230 98 63  
e-mail: helion@helion.pl  
© Helion 1991–2011

## Podręcznik HTML5. Smashing Magazine

Autor: [Bill Sanders](#)

Tłumaczenie: Agata Smoczyńska

ISBN: 978-83-246-3317-3

Tytuł oryginału: [Smashing HTML5 \(Smashing Magazine Book Series\)](#)

Format: 168×237, stron: 352



- Gruntowny przegląd nowych znaczników, atrybutów i wartości w HTML5
- Projektowanie struktury i nawigacji strony oraz korzystanie z grafiki, plików audio i wideo
- Sposoby na poprawne wyświetlanie witryny w różnych przeglądarkach

### **Niezbędny podręcznik dla webmasterów i programistów, którzy chcą szybko zacząć pracę z HTML5!**

Po latach intensywnego rozwoju i niemal corocznego wprowadzania nowych, ulepszonych wersji nagle w 2000 roku dla języka HTML nastąpiły czasy nadzwyczajnego jak na warunki sieci zastoju. Stary, dobry... ale mocno już wysłużony HTML4 zdecydowanie potrzebował gruntownych zmian. Przełom nastąpił w 2008 roku, gdy organizacja W3C opublikowała pierwszy szkic nowego standardu HTML5. Spotkał się on z tak entuzjastycznym przyjęciem, że od samego początku otrzymał mocne wsparcie rynku - popularne przeglądarki, począwszy od Google Chrome, przez Firefoksa czy Safari już zaczęły na wyścigi go implementować. Wciąż rozwijany HTML5 ma do zaoferowania sporo nowych, fantastycznych możliwości, które dają ambitnym webmasterom i programistom spore pole do popisu! Czy jesteś gotów po nie sięgnąć?

Jeśli skorzystasz z tej niezwykłej książki, będziesz mieć w zasięgu ręki całą wiedzę, której potrzebujesz, by zacząć działać! Ten podzielony na cztery części podręcznik pokaże Ci, jak zacząć pracę z najnowszą wersją HTML5, używać znaczników tego języka oraz różnych atrybutów i wartości. Dowiesz się, jak pracować z CSS oraz zapewniać swoim stronom doskonały wygląd i poprawne wyświetlanie na wszelkich możliwych nośnikach - od monitora komputera po ekran telefonu komórkowego. Zobaczysz także, jak zoptymalizować media w sieci, i zaczniesz sprawnie korzystać z tekstu, grafiki, audio, wideo czy nawigacji w przeglądarkach kompatybilnych z HTML5. Nauczysz się używać ciekawego, służącego do rysowania elementu canvas, a żebyś mógł z nim efektywnie pracować, autor wprowadzi Cię także w podstawy JavaScriptu i obiektowego modelu dokumentu (DOM) w HTML5. Ponadto zaczniesz stosować PHP do przetwarzania informacji wprowadzanych w postaci formularzy oraz przeczytasz o nowych obiektach geolocation i ich zaskakujących właściwościach.

Dzięki tej książce nauczysz się m.in. jak:

- pracować ze znacznikami HTML5 i CSS3
- zaprojektować strukturę witryny
- pracować z wartościami kolorów
- stworzyć przejrzystą i intuicyjną nawigację strony
- umieszczać grafikę oraz pliki audio i wideo na stronach HTML5
- wykorzystywać moc elementu canvas w HTML5
- korzystać z formularzy HTML5
- tworzyć interaktywność, zagnieźdzać obiekty i przechowywać informacje

**Wykorzystaj wszystkie możliwości, jakie oferuje HTML5 –  
twórz świetnie wyglądające, interaktywne strony nowej generacji!**

# Spis treści

|  |           |
|--|-----------|
| <b>Wstęp</b>   | <b>13</b> |
| <b>CZĘŚĆ I JĘZYK SIECI</b>                                 | <b>17</b> |
| <b>Rozdział 1 Wprowadzenie do HTML5</b>                    | <b>19</b> |
| Tworzenie przy pomocy znaczników: przegląd                 | 20        |
| Wprowadzanie nowych elementów HTML5                        | 21        |
| Korzystanie z kontynuowanych znaczników z HTML4            | 23        |
| Pomijanie lub zastępowanie niekontynuowanych znaczników    | 27        |
| Wybieranie przeglądarki interpretującej HTML5              | 28        |
| Mozilla Firefox  | 29        |
| Google Chrome  | 30        |
| Opera  | 32        |
| Apple Safari   | 32        |
| Microsoft Internet Explorer 9                              | 33        |
| Przegląd różnych wyświetleń                                | 34        |
| Do dzieła!   | 35        |
| <b>Rozdział 2 Znajomość znaczników HTML5</b>               | <b>37</b> |
| Parsowanie kodu  | 38        |
| Znajomość HTML5 i związanych z nim plików                  | 38        |
| Dowiadujemy się, które pliki działają w sieci              | 40        |
| Dowiadujemy się, jak działają znaczniki                    | 41        |
| Zaczynamy od podstawowych znaczników HTML                  | 41        |
| Opisywanie strony za pomocą znaczników                     | 42        |
| Rozpoznawanie części znacznika                             | 43        |
| Znajomość roli znacznika komentarza                        | 45        |
| Zagnieżdżanie znaczników                                   | 48        |
| Do dzieła!   | 50        |
| <b>Rozdział 3 Znaczniki tekstowe i odrobina CSS3</b>       | <b>53</b> |
| Podstawy   | 54        |
| Trochę więcej organizacji                                  | 55        |
| Zastanówmy się nad strukturą                               | 57        |
| Dodajmy więcej struktury HTML5                             | 59        |
| Dodawanie stylu tekstowi za pomocą CSS3                    | 62        |
| Nadawanie stylu elementom HTML5 za pomocą właściwości CSS3 | 62        |
| Tworzenie klas CSS3 i identyfikatory                       | 69        |
| Do dzieła!   | 73        |

|                   |   |            |
|-------------------|---|------------|
| <b>Rozdział 4</b> | <b>Praca z wartościami kolorów</b>                    | <b>75</b>  |
|                   | Znajomość kolorów RGB                                 | 76         |
|                   | Używanie nazw   | 76         |
|                   | RGB i wartości procentowe HSL                         | 77         |
|                   | Ustawianie RGB za pomocą liczb całkowitych            | 80         |
|                   | Ustawienia heksadecymalne: myślenie jak Twój komputer | 81         |
|                   | Dodawanie kolorom przejrzystości                      | 84         |
|                   | Tworzenie schematu koloru                             | 86         |
|                   | Z koloru podstawowego                                 | 86         |
|                   | Z obrazu  | 87         |
|                   | Integrowanie Twojej palety kolorów z Twoją stroną     | 88         |
|                   | Do dzieła!  | 90         |
| <br>              |   |            |
| <b>CZĘŚĆ II</b>   | <b>STRONY, WITRYNY I PROJEKTY</b>                     | <b>93</b>  |
| <br>              |   |            |
| <b>Rozdział 5</b> | <b>Tworzenie strony</b>                               | <b>95</b>  |
|                   | Ustawianie bazy startowej                             | 96         |
|                   | Dodawanie witrynie charakteru za pomocą metadanych    | 97         |
|                   | Kiedy skrypt jest potrzebny                           | 98         |
|                   | Projektowanie za pomocą sekcji                        | 100        |
|                   | Porządkowanie zawartości Twojej strony                | 104        |
|                   | Akapity, podziały i listy                             | 104        |
|                   | Grupowanie bez dzielenia                              | 108        |
|                   | Rysunki i podpisy                                     | 111        |
|                   | Organizacja plików                                    | 113        |
|                   | Odwołania do obrazków i ich organizacja               | 113        |
|                   | Odniesienie bezwzględne                               | 113        |
|                   | Odniesienie względne                                  | 114        |
|                   | Do dzieła!  | 116        |
| <br>              |   |            |
| <b>Rozdział 6</b> | <b>Wyświetlanie danych w tabelach</b>                 | <b>117</b> |
|                   | Właściwości tabeli CSS3 dla HTML5                     | 118        |
|                   | Tabele i dane tabelaryczne                            | 121        |
|                   | Podstawowe elementy tabeli                            | 121        |
|                   | Nadawanie stylu tabeli                                | 122        |
|                   | Dodawanie obramowania za pomocą CSS3                  | 123        |
|                   | Oddzielanie danych za pomocą kolorów tła              | 125        |
|                   | Tabele złożone  | 127        |
|                   | Korzystanie z atrybutów rowspan i colspan             | 128        |
|                   | Praktyczne rozpiętości w tabelach                     | 130        |
|                   | Do dzieła!  | 133        |
| <br>              |   |            |
| <b>Rozdział 7</b> | <b>Wszystko o odnośnikach</b>                         | <b>135</b> |
|                   | Element link i jego główne atrybuty                   | 136        |
|                   | Alternatywne arkusze stylów                           | 136        |
|                   | Ikony odnośników                                      | 139        |
|                   | Wstępne pobieranie                                    | 140        |
|                   | Inne atrybuty odnośnika                               | 141        |

|  |            |
|--|------------|
| Odnosińki strony   | 141        |
| Więcej o atrybucie rel   | 142        |
| Kotwice strony i identyfikatory                                      | 145        |
| Cele   | 148        |
| Korzystanie z elementu iframe  | 150        |
| Zagnieżdżanie stron internetowych                                    | 151        |
| Do dzieła!   | 153        |
| <b>Rozdział 8 Strategie nawigacji</b>                                | <b>155</b> |
| Nawigacja w sieci  | 156        |
| Nawigacja projektanta i użytkownika                                  | 156        |
| Nawigacja globalna   | 157        |
| Korzystanie z JavaScriptu do wzywania strony z odnośnikiem           | 163        |
| Tworzenie spójności  | 165        |
| Nawigacja pionowa i pozioma  | 167        |
| Zastosowanie pseudoklas CSS3   | 167        |
| Znajomość mechanizmów HTML5 w nawigacji pionowej                     | 169        |
| Korzystanie z ikon grafiki w nawigacji                               | 172        |
| Jednostronicowe witryny internetowe z pływającymi ramkami            | 172        |
| Odwoływanie się do grafiki   | 173        |
| Tworzenie i wykorzystywanie miniatur                                 | 173        |
| Korzystanie z pływających ramek na urządzeniach przenośnych          | 175        |
| Do dzieła!   | 176        |
| <b>CZĘŚĆ III MEDIA W HTML5</b>                                       | <b>179</b> |
| <b>Rozdział 9 Grafika</b>  | <b>181</b> |
| Podstawy plików obrazu w HTML5                                       | 182        |
| Liczy się format i piksele   | 182        |
| Zachowywanie warstw w grafikach sieci                                | 184        |
| Praca nad rozmiarem plików graficznych                               | 185        |
| Korzystanie z programu graficznego do modyfikowania rozmiaru pliku   | 187        |
| Umieszczanie obrazków i tworzenie elastycznych stron internetowych   | 195        |
| Umieszczanie obrazka za pomocą atrybutu align                        | 195        |
| Elastyczny rozmiar obrazka z odrobiną JavaScriptu                    | 197        |
| Aplikacja dla plików dynamicznych SVG z plików Adobe Illustrator CS5 | 200        |
| Do dzieła!   | 202        |
| <b>Rozdział 10 Dźwięk</b>  | <b>205</b> |
| Podstawy audio w HTML5   | 206        |
| Atrybut autoplay   | 206        |
| Atrybut controls   | 206        |
| Atrybut preload  | 208        |
| Atrybut loop   | 209        |
| Obsługa audio przez przeglądarkę                                     | 209        |
| Uratowany przez źródło: plan B                                       | 210        |
| Atrybut type   | 210        |
| Parametr codec typu źródłowego                                       | 211        |

|  |     |
|--|-----|
| Tworzenie plików audio                     | 212 |
| Rejestrator dźwięku w systemie Windows 7   | 212 |
| Sound System w systemie Mac OS X           | 214 |
| Efekty dźwiękowe: FX na Twoim komputerze   | 215 |
| Dźwięk zmiany                              | 215 |
| Integrowanie efektów dźwiękowych ze stroną | 217 |
| Do dzieła!                                 | 220 |

## **Rozdział 11 Video** **221**

|  |     |
|--|-----|
| Tworzenie strony HTML5 z wideo                       | 222 |
| Wideo i kompatybilność przeglądarki                  | 224 |
| Tworzenie formatu WebM: program Miro Video Converter | 225 |
| Konwersja na 3GP: program Adobe Media Encoder CS5    | 226 |
| Tworzenie filmów do zamieszczania w sieci            | 228 |
| Kamery internetowe                                   | 228 |
| Małe kamery wideo                                    | 229 |
| Standardowe kamery wideo                             | 230 |
| Zrzuty wideo z ekranu                                | 230 |
| Wideo i atrybuty źródła                              | 231 |
| Atrybut type   | 231 |
| Atrybut poster                                       | 232 |
| Atrybut preload                                      | 233 |
| Atrybut loop   | 233 |
| Atrybut autoplay                                     | 234 |
| Atrybut controls                                     | 234 |
| Atrybut width i height                               | 235 |
| Do dzieła!   | 235 |

## **CZĘŚĆ IV DYNAMICZNE ZNACZNIKI HTML5 ORAZ ODROBINA JAVASCRIPTU I PHP** **237**

### **Rozdział 12 Dodajemy odrobinę języka JavaScript** **239**

|   |     |
|---|-----|
| Umieszczanie JavaScriptu na stronach HTML5        | 240 |
| JavaScript w plikach zewnętrznych                 | 240 |
| Funkcje   | 241 |
| Procedury obsługi zdarzeń                         | 242 |
| Korzystanie z obiektowego modelu dokumentu        | 246 |
| Jak DOM współdziała z Twoją stroną i JavaScriptem | 246 |
| Elementy HTML5 i DOM                              | 248 |
| Przechowywanie tymczasowych wartości              | 250 |
| Zmienne   | 250 |
| Tablice   | 254 |
| Obiekty   | 255 |
| Do dzieła!  | 258 |

### **Rozdział 13 Korzystanie ze znacznika canvas na stronie** **259**

|                                      |     |
|--------------------------------------|-----|
| Podstawy elementu canvas             | 260 |
| Prosta implementacja elementu canvas | 262 |
| Obrazki w elemencie canvas i cienie  | 269 |

|  |            |
|--|------------|
| Tworzenie złożonych rysunków za pomocą elementu canvas                         | 274        |
| Linie i ruch   | 275        |
| Krzywe   | 279        |
| Do dzieła!   | 285        |
| <b>Rozdział 14 Dodawanie formularzy</b>  | <b>287</b> |
| Dodawanie formularza   | 288        |
| Główne atrybuty formularza   | 289        |
| Formularz jako część DOM   | 294        |
| Wiele rodzajów danych wejściowych  | 296        |
| Atrybut list, rodzaj adresu URL i listy danych                                 | 298        |
| Pola typu radio i checkbox ułatwiające wybieranie elementów danych wejściowych | 300        |
| Wybieranie daty  | 304        |
| Do dzieła!   | 306        |
| <b>Rozdział 15 Zagnieżdżanie obiektów i przechowywanie informacji</b>          | <b>307</b> |
| Geolokalizacja   | 308        |
| Znajdywanie szerokości i długości geograficznej                                | 308        |
| Pobieranie mapy  | 309        |
| Praca z właściwościami geolokalizacji i wtyczka Google Earth                   | 312        |
| Przechowywanie w HTML5   | 314        |
| Przechowywanie w sesji   | 314        |
| Przechowywanie lokalne   | 318        |
| Dodawanie i dostosowywanie obiektów na stronach HTML5                          | 322        |
| Dodawanie obiektu  | 322        |
| Dostosowywanie obiektu   | 324        |
| Do dzieła!   | 324        |
| <b>Rozdział 16 Przechwytywanie danych interaktywnych</b>                       | <b>325</b> |
| Język programowania po stronie serwera   | 326        |
| Konfiguracja PHP na Twoim komputerze (któremu wydaje się, że jest serwerem)    | 327        |
| Testowanie PHP   | 327        |
| Podstawy PHP   | 328        |
| Przechwytywanie żądania post   | 330        |
| Weryfikacja danych   | 330        |
| Podstawowe struktury skryptu PHP   | 332        |
| Tymczasowe przechowywanie danych   | 332        |
| Główne operatory PHP   | 335        |
| Tworzenie aplikacji poczty elektronicznej                                      | 337        |
| Prosta aplikacja poczty elektronicznej   | 337        |
| Przechwytywanie i wysyłanie PHP  | 340        |
| Dodawanie nagłówka i automatycznej odpowiedzi w PHP                            | 341        |
| Do dzieła!   | 343        |
| <b>Skorowidz</b>   | <b>345</b> |



# 5

## TWORZENIE STRONY

**WIELE NOWYCH** znaczników HTML5 to znaczniki strukturalne. W poprzednich rozdziałach niektóre z nich były używane, ale niekoniecznie zostały wytłumaczone. W tym rozdziale przyjrzymy się z bliska budowaniu stron HTML5 za pomocą CSS3

i procesowi ich tworzenia. Niektóre elementy staną się zrozumiałe, jak tylko zaczniesz korzystać z JavaScriptu, jeśli jednak stworzysz swoją stronę według wytycznych HTML5, będzie ona gotowa do użycia po dodaniu choćby odrobiny JavaScriptu.



## GÓRNA CZĘŚĆ DOKUMENTU HTML5

W czterech pierwszych rozdziałach tej książki wyjaśniłem w dużym stopniu, jak wykorzystywane są informacje ponad znacznikiem `<body>`. Kod powyżej tego znacznika nie dodaje zawartości stronie, ale wpływa na jej wygląd i informuje przeglądarkę, że jest to strona internetowa i jaki jest jej rodzaj. Rysunek 5.1 przedstawia ogólną budowę pierwszej części strony internetowej.



Rysunek 5.1. Tworzenie górnej części strony

Znacznik `<html>` jest elementem bazowym i wewnątrz tego elementu możesz umieścić atrybut `language`. Wewnątrz kontenera `<head>` znajdują się elementy metadanych. Są w nim też elementy skryptowe; one także są pokrótce omówione w tym podrozdziale i dalej w całej części IV książki.

Poza skryptami CSS3 nasze dotychczasowe przykłady nie umieszczały w nagłówku dokumentu HTML5 wielu znaczników. Znacznik `<meta>` może być używany na wiele sposobów, ale do tej pory korzystaliśmy z niego tylko do określania zestawu znaków. W tym rozdziale pokażę więcej sposobów jego użycia.

### USTAWIANIE BAZY STARTOWEJ

Wewnątrz typowej witryny internetowej znajduje się zazwyczaj kilka różnych stron, do których odnośniki znajdują się na stronie głównej. Witryna internetowa działa jak system nawigacji, który ma odnośniki do różnych stron. Jeśli ustawisz w nagłówku swojej strony znacznik `<base>` z odnośnikiem do adresu URL, możesz odwoływać się do innych stron względem strony głównej. Na przykład poniższe dwa przykłady (pliki *Base.html* i *FirstBase.html* znajdujące się w folderze rozdziału 5. pod adresem `ftp://ftp.helion.pl/przyklady/tfanh5.zip`) mają odnośniki do siebie nawzajem, ale odnoszą się do bazy, która jest ustawiona w kontenerze nagłówka.

```
<!DOCTYPE HTML>
<html><head>
<base href="http://helion.pl/pliki/TFANH5/base/">
<style type="text/css">
body {
    background-color:#FCC;
```

```

}
</style>
<meta charset="utf-8">
<title>Baza startowa</title>
</head>
<body>
<h1>To jest baza startowa</h1>
<a href="FirstBase.html">Pierwsza baza</a>
</body></html>

<!DOCTYPE HTML>
<html><head>
<base href="http://helion.pl/pliki/TFANH5/base/">
<style type="text/css">
body {
    background-color:#FC0;
}
</style>
<meta charset="utf-8">
<title>Pierwsza baza</title>
</head>
<body>
<h1>To jest pierwsza baza</h1>
<a href="Base.html">Baza startowa</a>
</body>
</html>

```

Co się tutaj dzieje? Znacznik `<base>` dyktuje Twojej przeglądarce, jak rozwiązać problem odnośników do innych dokumentów HTML — takich jak znacznik kotwicy `<a href="Base.html">`. Twoja przeglądarka będzie wiedziała, że ma szukać dokumentu *Base.html* w miejscu określonym w znaczniku `<base>`, mianowicie pod adresem *http://helion.pl/pliki/TFANH5*.

## DODAWANIE WITRYNIE CHARAKTERU ZA POMOCĄ METADANYCH

Do tej pory korzystaliśmy ze znacznika `<meta>`, by stwierdzić, że Twoja witryna korzysta z zestawu znaków UTF-8, ale element `meta` może zrobić dużo więcej. Należy go postrzegać jako element charakteryzujący się wielozadaniowością. Najważniejsze atrybuty elementu `meta` to `name` i `content`. Za pomocą atrybutu `name` ustawionego na `keywords` możesz określić listę słów kluczowych dla swojej witryny. W ten sposób wyszukiwarki mogą znaleźć Twoją witrynę, gdy użytkownicy próbują znaleźć produkty lub usługi — lub nawet tylko tematy, które zawrzesz na swoich stronach. Przypuśćmy na przykład, że Twoja witryna ma odnośniki do blogów i innych stron na temat psych bud. Twój znacznik `meta` będzie wyglądał mniej więcej tak:

```
<meta name="keywords" content="pies, tresura, szkolenie, agility">
```

Wszystkie wartości zawartości muszą być oddzielone przecinkami. Te tokeny mogą być bezpośrednio powiązane z zawartością lub z tym, czego ktoś może szukać. Znaczniki zawartości `meta` łatwo się ustawić i możesz w ten sposób pomóc użytkownikom znaleźć drogę do Twojej witryny.

Kolejny bardzo fajny atrybut znacznika `<meta>` to `http-equiv` przyjmujący stan `refresh`. Korzystając z tego atrybutu, możesz automatycznie odświeżać stronę lub nawet zmieniać strony HTML. Na przykład część Twojej strony może wyświetlać automatyczny pokaz zdjęć z przyjęcia lub zdjęć znajomych w klubie. Format znacznika dla stanu `refresh` wygląda następująco:

```
<meta http-equiv="refresh" content="[secs]">
```

Na przykład poniższy znacznik odświeża (załadowuje ponownie) stronę co 30 sekund:

```
<meta http-equiv="refresh" content="30">
```

Możesz ładować ponownie nie tylko tę samą stronę, ale również inne strony. Jeśli chcesz ładować serię stron, możesz ustawić początkowy znacznik `meta` jak poniżej, by ustawić stronę przypisaną jako wartość URL po pół sekundy:

```
<meta http-equiv="refresh" content=".5; url=pg2.html">
```

Zwróć uwagę, że zawartość pól zarówno liczby sekund, jak i adresu URL jest w tym samym zestawie znaków cudzysłowu. Poniższy kod HTML5 uruchamia serię stron, które są odświeżane do momentu załadowania się strony głównej:

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8"> <meta http-equiv="refresh" content=".5; url=pg2.html">
<title>Obraz nr 1</title>
</head>
<body>

</body>
</html>
```

Zaraz po stronie początkowej nastąpi poniższa sekwencja — tylko jedna na stronę:

- strona nr 2: `<meta http-equiv="refresh" content=".5; url=pg3.html">`
- strona nr 3: `<meta http-equiv="refresh" content=".5; url=pg4.html">`
- strona nr 4: `<meta http-equiv="refresh" content=".5; url=pg5.html">`
- strona nr 5: `<meta http-equiv="refresh" content=".5; url=homeNow.html">`

Strona główna `homeNow.html` nie miałaby stanu `refresh` w znaczniku `<meta>`. Poza elementem `meta` z atrybutem `content-type` nie miałaby też żadnego innego znacznika `meta`. (Będzie się to powtarzało w nieskończoność, jeśli zapętlisz swoją stronę główną i pierwszą!)

### KIEDY SKRYPT JEST POTRZEBNY

Im więcej używasz HTML5, tym bardziej potrzebujesz skryptu, by tworzyć jak najlepsze strony. Najbardziej powszechnym językiem skryptowym używanym z HTML5 jest JavaScript. Twoja przeglądarka ma interpreter dla JavaScriptu, tak samo jak dla HTML5. Na szczęście JavaScript

jest łatwy do nauczenia się i działa w małych fragmentach — nie tylko programiści mogą go obsługiwać.

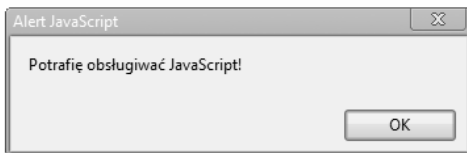
By włączyć do pracy JavaScript, potrzebujesz tylko dodać mały skrypt w nagłówku strony. Tutaj jest format znacznika:

```
<script type="text/javascript">
```

Program JavaScript przechodzi do reszty kontenera `<script>`. Poniższy kod HTML5 (plik *ScriptTag.html* znajdujący się w folderze rozdziału 5. pod adresem <ftp://ftp.helion.pl/przyklady/tfanh5.zip>) pokazuje, jak łatwy do nauczenia się jest JavaScript.

```
<!DOCTYPE HTML>
<html>
<head>
<script type="text/javascript">
  alert("Potrafię obsługiwać JavaScript!");
</script>
<meta charset="utf-8">
<title>Smak JavaScriptu</title>
</head>
<body>
Typowa strona internetowa...
</body>
</html>
```

Kiedy przetestujesz ten krótki program, zobaczysz pojawiające się okno ostrzeżenia (pokazane na rysunku 5.2).



Rysunek 5.2. JavaScriptowe okno ostrzeżenia

Tak na marginesie zauważysz, że JavaScriptowe okno ostrzeżenia ładuje się przed załadowaniem Twojej strony. To dlatego, że wszystko, co znajduje się w kontenerze nagłówka, ładuje się jako pierwsze. Jeśli masz bardziej złożony program JavaScript, który zostanie użyty na Twojej stronie HTML5, przetestuj go w różnych przeglądarkach i umieść go także w zewnętrznym pliku JavaScript. Rysunek 5.3 przedstawia to samo okno ostrzeżenia w przeglądarce Safari na iPhone; możesz wyraźnie zobaczyć, że strona skojarzona z kodem HTML5 nie załadowała się.

Jak tylko użytkownik kliknie *OK*, strona się załaduje. W międzyczasie możesz zobaczyć pliki z katalogu w tle na swoim urządzeniu przenośnym. Dodatkowo zwróć uwagę, że okno ostrzeżenia pokazuje domenę, w której mieści się JavaScript. Niektóre przeglądarki, takie jak Google Chrome, najpierw sprawdzają, czy użytkownik chce zaakceptować JavaScript z nazwanej witryny, zanim wyświetlą właściwe ostrzeżenie (podwójne ostrzeżenie!).



Rysunek 5.3. Okno ostrzeżenia ładujące się przed załadowaniem strony

Tak jak arkusze stylów, programy JavaScript mogą być ładowane z plików zewnętrznych. Zamiast jednak korzystać z elementu `link`, pliki JavaScript są ładowane za pomocą elementu `script`, tak jak pokazuje poniższy przykład:

```
<script type="text/javascript" src="smashingJS.js"></script>
```

Pliki JavaScript zapisujemy z rozszerzeniem `.js`, a pliki CSS3 z rozszerzeniem `.css`.

W części IV naszej książki zobaczysz dużo JavaScriptu podczas korzystania ze znacznika `<canvas>` i kilku innych znaczników HTML5. Znaczniki `<script>` i zawarty w nich kod JavaScript mogą być dodawane dokładnie w środku kodu HTML5. Zaletą umieszczania JavaScriptu w kontenerze nagłówka jest ładowanie się jego zawartości jako pierwszej, jeszcze przed załadowaniem się strony.

## PROJEKTOWANIE ZA POMOCĄ SEKCJI

Główna zmiana w HTML5 w porównaniu ze starszymi wersjami HTML dotyczy sekcji. Przed powstaniem HTML5 mogłeś myśleć o sekcjach jako o elemencie `body` i kilku znacznikach `<h...>`. W HTML5 możemy sobie wyobrażać stronę jako wiele sekcji i podsekcji. Większym elementem kontekstowym strony jest `article` i tak jak w artykule w gazecie, możesz znaleźć w nim różne sekcje, które tworzą blok tekstu. Rysunek 5.4 przedstawia przegląd sekcji strony HTML5.



Rysunek 5.4. Sekcje, które składają się na stronę

Na rysunku 5.4 widać różne bloki informacji, ale używane znaczniki nie mają generalnie zdolności, by wizualnie strukturyzować informacje. Znaczniki `<h>`, które są elementami sekcji, na pewno konfiguruje tekst w różnych rozmiarach, jednak inne znaczniki sekcji mają za zadanie nie tylko pomóc zorganizować stronę, ale też określić jej wygląd podczas wyświetlania.

Oto elementy sekcji:

- `body`,
- `section`,
- `nav`,
- `article`,
- `aside`,
- `h1 ... h6`,
- `hgroup`,
- `header`,
- `footer`,
- `address`.

Element `body` jest korzeniem sekcji, tak jak element `html` jest korzeniem strony. W poprzednich rozdziałach widziałeś kilka elementów sekcji, więc powinieneś je znać. Poniższy skrypt pomoże Ci zobaczyć, jak używa się kilku z nich naraz, i rozważyć ich użycie (plik *ArticleStructure.html* znajdujący się w folderze rozdziału 5. pod adresem <ftp://ftp.helion.pl/przyklady/tfanh5.zip>).

```

<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>Sekcje</title>
</head>
<body>
<article>
  <header>
    <h1>Piloci i Samoloty</h1>
    <p><q>Nigdy nie zostawiłem żadnego tam na górze</q>. <i>Ace Davis</i></p>
  </header>
  <nav><a href="#"> Bezpieczeństwo</a> | <a href="#">Wykaz czynności
  ↳ kontrolnych</a> | <a href="#">Lądowanie</a></nav>
  <section>
    <h2>Historie prawdziwych pilotów o lataniu</h2>
    <h3>...i inne lekarstwa na bezsenność.</h3>
    <section>
      <h4>Na krótko przed końcem</h4>
      <p>Krótko przed końcem pracy wieża kontrolna wezwała Maule do natychmiastowego
      ↳ startu, który wykonaliśmy około 4,5 metra od pasu startowego z prędkością
      ↳ 30 km/h. Coś pokryło mój wiatrochron, kiedy przyspieszyłem. Gdy wieża zdała
      ↳ sobie sprawę z popełnionego błędu, nakazała powrót i udało nam się
      ↳ bezpiecznie wylądować.</p>
    </section>
    <section>
      <h4>Prąd termiczny podczas startu</h4>
      <p>Kiedy wystartowaliśmy z Gila Bend w Arizonie przy temperaturze otoczenia 38 C,
      ↳ napotkaliśmy silny prąd termiczny na końcu pasa, który zabrał naszą Cessnę
      ↳ 177b na wysokość 4,500 metrów w równo 12 sekund, gdzie wyrównaliśmy
      ↳ i kontynuowaliśmy do Nowego Meksyku z pomocą strumienia odrzutowego,
      ↳ ustanawiając nowy rekord prędkości.</p>
    </section>
  </section>
  <aside>
    <h2>Prawdomówny pilot znaleziony!</h2>
    <p>Emily Rudders, pilot z Moose Bite w stanie Vermont, została uznana za
    ↳ jedyne istniejącego prawdomównego pilota. Zapytana o swoją najbardziej
    ↳ ekscytującą podniebną przygodę, Emily odpowiedziała: <q>"Nigdy nie leciałam
    ↳ żadnym samolotem. I tylko strzelam do nich, gdy przelatują i przeszkadzają
    ↳ łosiom".</q></p>
  </aside>
  <footer>
    <address>
      Kontakt:<a href="www.aopa.org">AOPA</a>
    </address>
  </footer>
</article>
</body>
</html>

```

Zadaniem sekcji jest podział strony na spójne części. Tworzą one organizacyjny zestaw elementów i chociaż mogą być używane do formatowania, nie jest to ich głównym zadaniem. Podczas nadawania akapitowi lub grupie akapitów formatowania należy, zgodnie z zaleceniami zawartymi w standardach W3C, używać znacznika `<div>`.

Rysunek 5.5 przedstawia wygląd strony. Może nie jest ona zbyt atrakcyjna, ale na pewno jest funkcjonalna. Artykuł jest o pilotach i lataniu. Nagłówek artykułu przedstawia tytuł artykułu (*Piloci i samoloty*) i zawiera przytoczoną wypowiedź pilota zapisaną z użyciem znacznika `<q>`. Zaraz za nagłówkiem znajduje się pierwsza sekcja zawierająca historie o lataniu. Wewnątrz niej zagnieżdżone są dwa znaczniki `<sect i on>`, które oddzielają od siebie dwie historie.

Nawiązująca do reszty treści sekcja o prawdziwości historii pilotów znajduje się w oddzielnym kontenerze elementu `aside`. Na rysunku 5.4 możesz zauważyć, że element ten został umieszczony w oddzielnym akapicie, ale sam w sobie stanowi on nawiązanie do sensu strony. Nie jest on elementem służącym do formatowania.



Rysunek 5.5. Strona zbudowana za pomocą elementów sekcji



## Dlaczego powinieneś w ogóle zwracać uwagę na organizację sekcji

Może Ci się wydawać, że da się stworzyć stronę bez kłopotania się ze znacznikami sekcji. To prawda. Choć pod przykrywką dobrze działającej strony znajduje się dudniący silnik, który może się odwoływać do różnych części Twojej strony, znany jako Obiektowy Model Dokumentu (DOM) — różne kombinacje, które ustawiłeś za pomocą elementów sekcji, mogą być w nim adresowane jako różne obiekty i ich pochodne w dobrze ponumerowanym strumieniu danych krążących po Internecie. Jeśli poświęcisz nieco uwagi modelowi organizacyjnemu używanemu w HTML5, Twoja strona będzie szczęśliwa, Internet będzie szczęśliwy i cała galaktyka będzie szczęśliwa.

Wreszcie na dole artykułu znajduje się stopka (<footer>). Elementy stopki mogą znajdować się gdziekolwiek, także wewnątrz indywidualnych kontenerów elementów `section` i `aside`. Stopki zachowują się jak elementy formalnie zamykające dla elementów sekcji. Wewnątrz stopki znajduje się element `address` z odnośnikiem adresu URL nawiązującym do artykułu.

Patrząc na stronę przedstawioną na rysunku 5.5 oraz jej kod, możesz zobaczyć sens strony opisany w znacznikach sekcji. Jak już wcześniej zaznaczyłem, nie służą one do formatowania, ale do organizowania sensu strony.

## PORZĄDKOWANIE ZAWARTOŚCI TWOJEJ STRONY

Kiedy będziesz mieć już swój plan organizacyjny, uporządkuj zawartość strony za pomocą różnych sekcji. Na rysunku 5.4 mogłeś zobaczyć, że kilka elementów sekcji zawiera elementy grupowania, takie jak znaczniki `<p>`. Elementy grupowania są najlepszym miejscem na dodawanie stylów CSS3, w przeciwieństwie do elementów sekcji. W tym podrozdziale dowiesz się, jakie są główne elementy pomocne w organizowaniu materiałów.

### AKAPITY, PODZIAŁY I LISTY

Znaczniki `<p>` i `<div>` są używane jako konie pociągowe stron HTML do grupowania i nadawania stylu. Oba są bardzo ważne, ale musisz pamiętać, że ich zadaniem nie jest już dzielenie zawartości Twojej strony na sekcje. Zamiast tego możesz myśleć o nich jak o elementach grupujących sekcje. Na przykład poniższy fragment kodu pokazuje stary sposób korzystania z tych dwóch znaczników:

```
<div>
  <h1>Wszystko o tym, co ważne</h1>
  <p>
  <h2>Szukanie prawdziwej miłości</h2>
  </p>
  <p>
  <h2>Wybór odpowiedniej kariery</h2>
  </p>
  <p>
  <h2>Znajdowanie miejsca parkingowego</h2>
  </p>
</div>
```

Ten kod działa świetnie w HTML5, ale zadziała jeszcze lepiej przy użyciu bardziej odpowiedniego elementu do tego zadania. Poprawiony kod wygląda następująco:

```
<header>
  <h1>Wszystko o tym, co ważne</h1>
</header>
<section>
  <h2>Szukanie prawdziwej miłości</h2>
  <h2>Wybór odpowiedniej kariery</h2>
  <h2>Znajdowanie miejsca parkingowego</h2>
</section>
```

Na Twojej stronie wygląda to tak samo, choć strony HTML5 będą praktyczniejsze, kiedy użyjesz nowych elementów sekcji.

Nasuwa się pytanie: „Gdzie mogą być użyte elementy `p` i `div`?”. Tak naprawdę nie chcesz polegać na żadnym z nich. Mogą być jednak przydatne, jeśli chcesz dodać element stylu lub inny atrybut w środku znacznika `<article>` lub `<section>`. Rozważ poniższy kod (plik *UseDiv.html* znajdujący się w folderze rozdziału 5. pod adresem <ftp://ftp.helion.pl/przyklady/tfanh5.zip>).

```
<!DOCTYPE HTML>
<html>
<head>
<style type="text/css">
body {
  font-family:"Comic Sans MS", cursive;
  color:#0C6;
  background-color:#FFC;
}
.dziewczynki {
  background-color:pink;
}
.chlopcy {
  background-color:powderblue;
}
</style>
<meta charset="utf-8">
<title>Imiona dla dzieci</title>
</head>
<body>
<article>
<header>
  <h1>Imiona dla dzieci</h1>
</header>
<section>
  <div class="dziewczynki">
    <h2>Dziewczynki</h2>
    <ul>
      <li>Oliwia</li>
      <li>Teresa</li>
      <li>Emilia</li>
    </ul>
```

```

        </div>
    </section>
    <section>
        <div class="chłopcy">
            <h2>Chłopcy</h2>
            <ul>
                <li>Jakub</li>
                <li>Ryszard</li>
                <li>Jan</li>
            </ul>
        </div>
    </section>
</body>
</html>
    
```

Rysunek 5.6 pokazuje wynik powyższego kodu; co ważne, znacznik `<div>` został wykorzystany tylko do nadania kolorów tła dwóm różnym elementom `<section>`.



Rysunek 5.6. Użycie znacznika `<div>` do nadania stylu

W widocznej na rysunku liście punktowanej element `div` umożliwił użycie dwóch różnych stylów tła w kontenerze `section` bez dodawania klas znacznikowi `<section>`. Pamiętaj, że oba znaczniki `<p>` i `<div>` są elementami bardziej ogólnymi, a zawsze powinieneś używać elementów, które najlepiej opisują Twój obiekt na stronie.

Poza grupowaniem i nadawaniem stylu z użyciem znacznika `<div>` listy służą też do przedstawiania danych. HTML5 nadal korzysta ze znaczników `<ul>` do grupowania na przykład imion dla dzieci — chłopców i dziewczynek. Mała, ale znacząca różnica polega na tym, czy tworzona lista jest numerowana (`<ol>`), czy nienumerowana (`<ul>`).

Użycie listy nienumerowanej lub numerowanej zależy od kontekstu. Na przykład w 2010 roku podczas rozgrywanych w Republice Południowej Afryki piłkarskich Mistrzostw Świata cztery najlepsze drużyny pochodziły z Niemiec, Holandii, Hiszpanii i Urugwaju. Jeśli tworzyłbyś ich listę na początku mistrzostw, mógłbyś skorzystać z listy nienumerowanej. Na koniec mistrzostw lepiej jednak byłoby użyć listy numerowanej, by pokazać ostateczne wyniki. Poniższa strona (plik *ol\_ul.html* znajdujący się w folderze rozdziału 5. pod adresem <ftp://ftp.helion.pl/przyklady/tfanh5.zip>) przedstawia różne listy punktowane zależne od kontekstu i znaczenia, które mu towarzyszy.

```
<!DOCTYPE HTML>
<html>
<head>
<style type="text/css">
/*20268C,0C080C,2F8C2B,F27507,F20505 */
body {
    background-color:#2F8C2B;
    color:#0C080C;
    font-family:Verdana, Geneva, sans-serif;
}
h2 {
    background-color:#F27507;
    color:#20268C;
    font-family:"Comic Sans MS", cursive;
}
h3 {
    font-family:"Comic Sans MS", cursive;
}
ol {
    background-color:#F27507;
}
ul {
    background-color:#F20505;
}
</style>
<meta charset="utf-8">
<title>Lista numerowana i nienumerowana</title>
</head>
<body>
<h2>Mistrzostwa Świata 2010</h2>
<h3>Początek</h3>
<ul>
    <li>Hiszpania</li>
    <li>Holandia</li>
    <li>Niemcy</li>
    <li>Urugwaj</li>
</ul>
<h3>Koniec</h3>
<ol>
    <li>Hiszpania</li>
    <li>Holandia</li>
    <li>Niemcy</li>
    <li>Urugwaj</li>
```

```
</ol>
</body>
</html>
```

Jak widzisz na rysunku 5.7, nie ma znaczenia, czy grupa na początku Mistrzostw Świata ułożona jest według hierarchii — lista składa się po prostu z czterech drużyn. Na koniec kolejność jest jednak bardzo ważna, więc lista numerowana jest bardziej odpowiednia.



Rysunek 5.7. Numerowane i nienumerowane listy oddają różne znaczenie

Mogłeś również zauważyć, że dwa różne rodzaje list mają odmienne kolory tła dodane za pomocą CSS3. Używając więc elementów grupowania, możesz też chcieć dalej pogrupować zawartość za pomocą koloru, co pokazano na rysunku 5.6 i 5.7.

### GRUPOWANIE BEZ DZIELENIA

Znacznik `<hr>` to element grupowania, którego nie powinieneś używać do niczego innego niż oddzielanie nagłówka od reszty strony (do tego też lepiej nie). Element `hr` (pozioma linia) to po prostu linia, ale powinna być używana rozsądnie i rzadko. Zwróć uwagę na przykład na poniższy fragment wiersza pt. „Kubla Chan” Samuela Taylora Coleridge’a (przeł. Jerzy Pietrkiewicz):

W Xanadu kazał Kubla Chan wznieść cud,  
 Pałac rozkoszy, gdzie przepych bez końca;  
 Płynęła święta rzeka Alf, jej chłód  
 Wśród niezmiernych pieczar trwał, aż z wód  
 Szum wlał się w morze mroczne bez krzty słońca.

Sto urodzajnych pól, za nimi wieże  
 I mury — jednym kręgiem jak pancerzem  
 Objęto; w środku sady i strumienie  
 Kręte, nad nimi kwitną wonne drzewa,  
 A tam znów bory stare jak wzgórz cienie,  
 Gdzie w plamach słońca zieleń się przelewa.

Ach, w romantyczną otchłań spójrz, jak spada  
 Z zielonej góry w dół — z cedrów zasłona  
 Ponad parowem. Dzikość. Czar się skrada  
 Święty — i straszy twarz księżycy blada,  
 Gdy lka z tęsknoty kochanka demona.

Trzy zwrotki są oddzielone zwykłą podwójną spacją, tak jak tytuł. Jeśli znaczniki `<hr>` zostaną jednak umieszczone tak jak w poniższym kodzie (plik *HR.html* znajdujący się w folderze rozdziału 5. pod adresem <ftp://ftp.helion.pl/przyklady/tfanh5.zip>), zobaczysz inny zintegrowany wygląd wiersza.

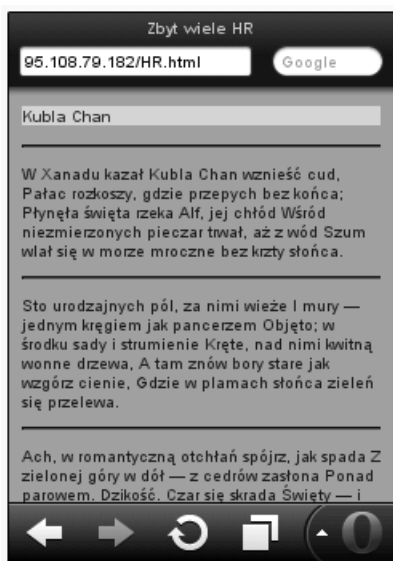
```
<!DOCTYPE HTML>
<html>
<head>
<style type="text/css">
/*A1A680,D9D7BA,D90404,8C0303,590202 */
body {
  background-color:#A1A680;
  color:#590202;
  font-family:"Palatino Linotype", "Book Antiqua", Palatino, serif;
  font-size:8px;
}
h4 {
  background-color:#D9D7BA;
  color:#8C0303;
  font-family:Tahoma, Geneva, sans-serif;
}
</style>
<meta charset="utf-8">
<title>Zbyt wiele HR</title>
</head>
<body>
<header>
  <h4>Kubla Chan</h4>
</header>
<article>
  <hr>
  W Xanadu kazał Kubla Chan wznieść cud,<br>
  Pałac rozkoszy, gdzie przepych bez końca;<br>
  Płynęła święta rzeka Alf, jej chłód<br>
  Wśród niezmiernych pieczar trwał, aż z wód<br>
```

```

Szum wlał się w morze mroczne bez krzty słońca.<br>
<hr>
Sto urodzajnych pól, za nimi wieże<br>
I mury – jednym kręgiem jak pancierzem<br>
Objęto; w środku sady i strumienie<br>
Kręte, nad nimi kwitną wonne drzewa,<br>
A tam znów bory stare jak wzgórz cienie,<br>
Gdzie w plamach słońca zieleń się przelewa.<br>
<hr>
Ach, w romantyczną otchłań spójrz, jak spada<br>
Z zielonej góry w dół – z cedrów zasłona<br>
Ponad parowem. Dzikość. Czar się skrada<br>
Święty – i straszy twarz księżycą blada,<br>
Gdy łąka z tęsknoty kochanka demona. </article>
</body>
</html>

```

Jak widzisz, wszystkie znaczniki `<hr>` są wewnątrz elementu `<article>`, podczas kiedy tytuł jest częścią elementu `header`. Na rysunku 5.8 strona jest jednak wyświetlona na urządzeniu przenośnym i poziome linie niczego nie rozjaśniają, natomiast fragmentują całość.



Rysunek 5.8. Linie poziome mogą podzielić znaczenie

Linia pozioma może się przydać, gdy Twoja strona dzieli się na wyraźne części. Nawet wtedy powinieneś jednak dodać CSS3 w celu rozjaśnienia elementu `hr`, by stał się on bardziej subtelny — powinno też pomóc dodanie przejrzystości. Dobrzy projektanci wiedzą, jak korzystać z linii poziomej oszczędnie, natomiast zwykli użytkownicy mogą łatwo narobić bałaganu na swoich stronach, nadużywając znaczników `<hr>`.

## RYSUNKI I PODPISY

Jedne z bardziej frustrujących elementów HTML5 to znaczniki `<figure>` i `<figcaption>`, szczególnie gdy są używane razem. Umieszczając element `figcaption` wewnątrz kontenera elementu `figure`, możesz przypuszczać, że tworzą one jeden obiekt dla układu i projektu. Element `figcaption` jest uważany za produkt elementu `figure`, kiedy element `figcaption` jest zagnieżdżony wewnątrz elementu `figure`. Nie znaczy to jednak, że pokazują się one razem na stronie. W rzeczywistości wyrównanie obrazka z podpisem może okazać się bardzo podchwytliwe.

W przypadku bardziej wyszukanego formatowania CSS3 obrazek i jego podpis mogą być traktowane jako obiekty o relacjach zbliżonych do relacji rodzica i dziecka. Z faktu, że elementy `figure` i `caption` należą do elementów grupowania HTML5, nie wynika, że są one formatowane razem na stronie; zamiast tego można się do nich odwoływać jako do pojedynczego przepływu głównej zawartości strony. W międzyczasie musisz ostrożnie pracować z obydwojema elementami razem, tak jak pokazano w poniższym kodzie HTML5 (plik *Figure\_n\_caption.html* znajdujący się w folderze rozdziału 5. pod adresem <ftp://ftp.helion.pl/przyklady/tfanh5.zip>), gdzie napis odwołuje się do ostylowanego obrazka.

```
<!DOCTYPE HTML>
<html>
<head>
<style type="text/css">
/* 732D3F,A66879,D9C3B0,260101,F2F2F2 */
body {
    background-color:#D9C3B0;
    color:#732D3F;
    font-family:Verdana, Geneva, sans-serif;
    font-size:11px;
}
aside {
    margin-left:260px;
}
h1 {
    font-family:"Trebuchet MS", Arial, Helvetica, sans-serif;
    background-color:#F2F2F2;
    color:#A66879;
    text-align:center;
}
figcaption {
    color:#A66879;
    background-color:#F2F2F2;
}
img {
    margin:5px;
}
</style>
<meta charset="utf-8">
<title>Grupowanie rysunków i podpisów</title>
</head>
<body>
```



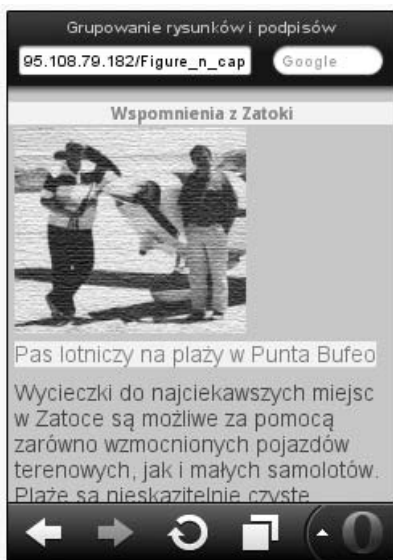
```

<header>
  <h1>Wspomnienia z Zatoki</h1>
</header>
<article>
  <figure> <br>
    <figcaption>Pas lotniczy na plaży w Punta Buefo</figcaption>
  </figure>
  <section>
    <p> Wycieczki do najciekawszych miejsc w Zatoce są możliwe za pomocą zarówno
    ↳wzmocnionych pojazdów terenowych, jak i małych samolotów. Plaże są
    ↳nieskazitelnie czyste, niezatłoczone i niezaśmiecone. Czas spędzony na
    ↳wędkowaniu jest wynagrodzony, kiedy ryby zostaną przyrządzone w rybnym tacos
    ↳- specjale, którego nie można przeoczyć. <i>Zatoka Kalifornijska</i> (znana
    ↳również jako <i>Morze Corteza</i> i <i>Morze Vermillona</i>) jest jasna
    ↳i czysto niebieska. Oczywiście plaże są niezatłoczone i wolne od odpadków
    ↳pozostawionych przez innych.</p>
  </section>
</article>
</body>
</html>

```

Zacznij myśleć o elementach i ich pochodnych. W naszym przypadku element `figcaption` jest elementem pochodnym elementu `figure`. Rysunek 5.9 przedstawia podpis pod obrazkiem, a oba są umieszczone w kontenerze `<figure>`.

Jak widać na rysunku 5.9, kontener `<figcaption>` ma nadany inny styl, mimo że pochodzi z kontenera `<figure>`. Nie możesz jednak zakładać, że element `figcaption` będzie odpowiednio ustawiony na rysunku 5.9 tylko dlatego, że pochodzi od elementu `figure`, któremu nadaje podpis.



Rysunek 5.9. Elementy `figure` i `figcaption` z grafiką

## ORGANIZACJA PLIKÓW

Organizacja plików prostej strony jest łatwa. Wraz ze wzrostem złożoności strony, szczególnie kiedy w jej tworzenie zaangażowanych jest wielu projektantów i programistów, pojawia się konieczność uporządkowania jej za pomocą oddzielnych katalogów lub czasami serwerów. W tym podrozdziale dowiesz się o kilku sprawach organizacyjnych i o tym, jak sobie radzić z organizacją i dostępem do plików.

### ODWOŁANIA DO OBRAZKÓW I ICH ORGANIZACJA

Typowa witryna internetowa ma jeden albo więcej folderów (katalogów) przeznaczonych na pliki grafiki lub ich typy. W większości przykładów w książce nie używano do tej pory oddzielnych folderów dla obrazów i stron HTML5, które je łądowały; zamiast tego wszystkie pliki były umieszczane w tym samym katalogu co pliki HTML5. Kiedy mamy więcej stron i obrazów do załadowania na stronach, bardziej wydajnym sposobem zarządzania naszą witryną jest używanie oddzielnych folderów dla różnych grup mediów. To, jak zorganizujesz obrazy, będzie zależało od wielu różnych czynników. Poniżej przedstawiam kilka możliwych katalogów i podkatalogów, z których będziesz korzystać:

- podział na grupy (*Zwierzęta > Ssaki > Gryzonie > Myomorpha > Mysz domowa > Mickey*),
- temat (*Wakacje > Dokąd pojechać > Gdzie się zatrzymać > Co spać*),
- proces (*Pieczenie > Robienie pączków > Przygotowywanie ciasta > Ustawianie piekarnika > Odmierzanie czasu*).

Niezależnie od planu organizacyjnego i od tego, jak są zorganizowane obrazy, musisz wiedzieć, jak mieć do nich dostęp.

### ODNIESIENIE BEZWZGLĘDNE

Wszystkie odniesienia do obrazów odbywają się za pomocą adresu URL w całości lub w części wraz z nazwą pliku. Adres bezwzględny zaczyna się od protokołu `http://` i zawiera całą ścieżkę dostępu do pliku HTML5. Na przykład poniżej widzimy adres bezwzględny pliku:

```
http://www.smashinghtml5.com/organization/graphics/faces.html
```

Bez względu na to, skąd wywołujemy adres URL, jest on rozpoznawany dzięki nazwie pliku znajdującej się na jego końcu. Tak samo jest ze źródłem (`src`) odniesienia do obrazu. Jeśli Twój kod ma poniższy odnośnik, to bez względu na to, gdzie znajduje się wzywana strona, załaduje się plik `nose.png`.

```

```

Wzywana strona może być na całkowicie innym serwerze i ukaże się pod adresem bezwzględnym.

Zaletą używania tych adresów jest to, że nie musisz się martwić, gdzie strona znajduje się w Twojej witrynie. Nie musisz się nawet przejmować, czy jest na tym samym serwerze. Pozostawia to jednak nadal wiele do życzenia, jeśli chodzi o organizację witryny, a długie adresy URL muszą być podawane bezbłądnie.

### ODNIESIENIE WZGLĘDNE

Odniesienie względne jest względne do pozycji wzywanej strony w witrynie lub jej zdefiniowanej bazy. Twoja strona na Twoim komputerze ma raczej pozycję `file`, a nie `http`. Na przykład poniższy adres to pozycja bezwzględna pliku *jakasStrona.html*:

```
"C:\Users\oem\Desktop\HTML5\jakasStrona.html"
```

Jeśli miałbym grafikę w folderze o nazwie *HTML5*, mógłbym użyć jej względnego adresu do wezwania jej z pliku *jakasStrona.html*. Jeśli miałbym na przykład plik *jakasGrafika.png* w folderze *HTML5*, mógłbym użyć tylko poniższego odniesienia względnego:

```

```

Jeśli chciałbym uporządkować swoje obrazy w oddzielnym folderze o nazwie *obrazy* wewnątrz folderu *HTML5*, użyłbym adresu względnego:

```

```

Możesz dodawać kolejne poziomy względne znajdujące się poniżej, rozdzielając je znakiem slash (/). Na przykład bardziej złożony zestaw grafiki wyglądałby tak:

```

```

Poza techniką „drążenia w dół” możesz też użyć techniki „drążenia w górę”. Korzystając z niej, masz dostęp do zawartości folderów swojej wzywanej strony. Przypuśćmy na przykład, że masz poniższą ścieżkę dostępu, a twoja strona HTML5 jest w folderze *podstawowyFolder*.

```
gornyFolder/srodkowyFolder/podstawowyFolder
```

Aby otworzyć plik z grafiką w folderze *srodkowyFolder*, musisz użyć poniższego formatu:

```

```

Jeśli grafika byłaby w folderze *gornyFolder*, należałoby skorzystać z poniższego formatu:

```

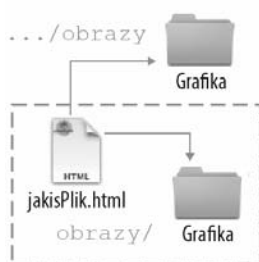
```

Przy używaniu techniki „drążenia danych w górę” nie podaje się nazwy docelowego folderu, który jest na wzywanej stronie; zamiast tego używa się znaków `../`, aż Twoja wzywana pozycja znajdzie się na takim poziomie, na jakim chcesz. Oznacza to, że możesz dotrzeć do dowolnego poziomu i wrócić do innej gałęzi. Na przykład poniżej otwieramy folder *gornyFolder*, a potem wewnątrz folderu *gornyFolder* schodzimy w dół przez folder obrazu do docelowej grafiki:

```

```

Rysunek 5.10 przedstawia pełną ilustrację graficzną dostępu do zawartości w wyższych i niższych poziomach folderach.



Rysunek 5.10. Względne ścieżki dostępu

Tak jak napisałem we wcześniejszym punkcie tego rozdziału, „Ustawianie bazy startowej”, Twoja pozycja względna może być inna niż położenie pliku. Na przykład spójrz na dwie poniższe strony (pliki *Earth.html* i *Alien.html* znajdujące się w folderze rozdziału 5. pod adresem <ftp://ftp.helion.pl/przyklady/tfanh5.zip>). Pierwsza wzywa drugą na innym serwerze, ale jako że baza pierwszej strony jest ustawiona na drugim serwerze, wezwanie jest względne. Pierwszy plik ma nazwę *Earth.html* i jest umieszczony na komputerze lokalnym lub dowolnym serwerze. A jej baza jest ustawiona na *helion.pl* w folderze pliki/TFANH5. Może więc korzystać ze względnego adresu URL, by mieć dostęp do pliku *Alien.html* na całkowicie innym serwerze.

---

### Baza ustawiona na innym serwerze

```
<!DOCTYPE HTML>
<html>
<head>
<base href="http://helion.pl/pliki/TFANH5">
<meta charset="utf-8"> <title>Ziemia</title>
</head>
<body>
<h1>To jest Ziemia</h1>
<a href="Alien.html">Startujemy!</a>
</body>
</html>
```

---

### Strona na innym serwerze

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8"> <title>Planeta Smashing</title>
</head>
<body>
<h1>Strona z serwera kosmitów</h1>
</body>
</html>
```

---

Mimo że domena pierwszej strony (*Earth.html*) to nie *helion.pl/pliki/TFANH5*, ustawiona baza to *helion.pl*. W rezultacie względny odnośnik do *Alien.html*, który mieści się w *helion.pl*, jest użyty bez potrzeby podawania adresu bezwzględnego.

## DO DZIEŁA!

W pierwszym podrozdziale dowiedziałeś się, jak korzystać ze stanu Refresh, by automatycznie zmieniać strony. Popatrz na odnośnik do prac Eadwarda Muybridge'a, by pobawić się trochę z animacją i stanem Refresh.

[http://138.23.124.165/collections/permanent/object\\_genres/photographers/muybridge/contents.html#](http://138.23.124.165/collections/permanent/object_genres/photographers/muybridge/contents.html#)

Co ciekawe, Muybridge już w roku 1878 był w stanie tworzyć filmy, używając do tego kilku zdjęć. Zanim więc Thomas Edison wynalazł ruchomy obraz, Muybridge robił krótkie filmy (około 12-klatkowe), prezentując animowany widok przeszłości. Uniwersytet Kalifornijski w Riverside zachował i ożywił prace Muybridge'a za pomocą animowanych plików GIF. Aby zobaczyć, jak możesz robić animacje, korzystając z odświeżania stron, pobierz jeden z animowanych plików GIF z kolekcji Muybridge'a z odnośnika powyżej — studia lokomocji — i wypakuj z niego 12 indywidualnych zdjęć. Możesz je wypakować za pomocą programu Adobe Photoshop, Adobe Fireworks i kilku innych. (Wpisz wyodrębnij obrazy z animowanego pliku GIF w wyszukiwarce, by znaleźć wiele sposobów na uzyskanie pojedynczych obrazów. Jeśli masz Maca, możesz użyć aplikacji Preview i tylko przeciągnąć pojedyncze obrazy z aplikacji Preview do oddzielnego folderu).

Kiedy już wypakujesz pojedyncze pliki GIF, ustaw swoją animację, korzystając ze stanu Refresh z elementem meta w sekcji <head> swojego skryptu. Aby zacząć, użyj poniższego skryptu HTML5 (plik *an1.html* znajdujący się w folderze rozdziału 5. pod adresem <ftp://ftp.helion.pl/przyklady/tfanh5.zip>).

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="refresh" content="0.1; url=an2.html">
<title>Obraz 1</title>
</head>
<body>

</body>
</html>
```

Pojedyncze pliki GIF zostały zapisane z rozszerzeniem *.png* i nazwane od *an1.png* aż po *an12.png* (część *an* ma oznaczać animację). Także 12 plików HTML zostało nazwanych z użyciem w nazwie liter *an*, od *an1.html* po *an12.html*. Jak tylko skończysz, uzyskasz animację spacerującego konia. Jeśli przełączysz się z dwunastej strony z powrotem do pierwszej, koń będzie szedł dalej.

# Skorowidz

!DOCTYPE, 23, 41, 54  
&gt;, 124  
&lt;, 124  
&nbsp;, 50  
\_blank, 148, 150, 293  
\_parent, 148  
\_self, 148  
\_top, 148  
3GP, 226

## A

a, 23, 136, 141  
    target, atrybut, 148  
abbr, 23  
accept-charset, atrybut, 290  
acronym, 27  
action, atrybut, 291  
addColorStop(), 284  
address, 23  
Adobe Browserlab, 30, 31  
Adobe Device Central, 227, 228  
Adobe Dreamweaver, 34  
Adobe Illustrator, 200  
Adobe Media Encoder CS5, 226, 227  
Adobe Photoshop, 188  
akapity, 104  
aliasing, 186  
align, atrybut, 58, 195, 196  
alt, atrybut, 58  
alternate stylesheet, 136  
Apache, 327  
Apple Quick Time Player, 231  
Apple Safari, 32  
applet, 27  
arc(), 280  
area, 23  
arkusze stylów, 63, 64, 136  
article, 22  
aside, 22

audio, 22, 206, 210  
    autoplay, atrybut, 206  
    controls, atrybut, 206  
    loop, atrybut, 209  
    preload, atrybut, 208, 209  
    src, atrybut, 206  
author, 142  
autocomplete, atrybut, 291  
autoplay, atrybut, 206, 234

## B

b, 23  
background-image, 85  
bajt, 82  
base, 23, 96, 97  
basefont, 27  
bdo, 23  
bgsound, 27  
big, 27  
bit, 82  
blink, 27  
blockquote, 23  
body, 23, 43  
br, 23, 55, 56  
Browserlab, 30, 31  
button, 23

## C

Camtasia, 230, 231  
canvas, 22, 259, 260, 261, 262, 263  
    gradienty, 283, 284, 285  
    height, atrybut, 263  
    koła, 283  
    krzywe, 279  
    łuki, 280, 282, 283  
    obrazki, 269  
    rysunki, 263, 265, 266, 267, 274  
    siatka, 262, 275  
    width, atrybut, 263

caption, 24, 111, 121  
 center, 27, 28  
 checkbox, pole, 300  
 Chrome, 30  
 ciasteczka, 318  
 cień, 270, 271, 272  
 cite, 24  
 clearRect(), 267  
 code, 24  
 codec, 211  
 col, 24  
 Coleridge, Samuel Taylor, 108  
 colgroup, 24  
 colspan, atrybut, 128, 130  
 command, 22  
 content, atrybut, 97  
 controls, atrybut, 206, 234  
 createLinearGradient(), 283  
 CSS3, 62
 

- dodawanie stylu, 62
- identyfikatory, 69, 71
- klasy, 69
- komentarze, 72
- pseudoklasy, 167, 168

 cudzysłów, używanie, 44, 45

## D

dane tabelaryczne, 117, 121  
 dane, weryfikacja, 330  
 datalist, 21, 22, 298  
 dd, 24  
 del, 24  
 details, 22  
 dfn, 24  
 dir, 27  
 display, 118  
 div, 24, 103, 104, 106  
 dl, 24  
 document, 258  
 document.write(), 240  
 DOM, 104, 246, 258
 

- elementy HTML5, 248
- współdziałanie z JavaScript, 246

 drawImage(), 270  
 Dreamweaver, 34  
 dt, 24  
 dźwięk, 205
 

- efekty dźwiękowe, 215, 217
- formaty, 209

konwersja plików, 215  
 tworzenie plików audio, 212

## E

em, 24  
 embed, 22  
 Emerson, Ralph Waldo, 165  
 enctype, atrybut, 290  
 event, 258

## F

fieldset, 24, 303  
 figcaption, 22, 111, 112  
 figure, 22, 111, 112  
 fill(), 264  
 fillRect(), 264  
 fillStyle, 264  
 filtry, 272, 274  
 Firefox, 29  
 first, 145  
 Flash, 322  
 FlashKit, aplikacja, 215  
 font, 27  
 footer, 22, 104  
 form, 24, 163, 249, 288
 

- accept-charset, atrybut, 290
- action, atrybut, 291
- autocomplete, atrybut, 291
- enctype, atrybut, 290
- id, atrybut, 292
- list, atrybut, 298
- method, atrybut, 291
- name, atrybut, 292
- novalidate, atrybut, 290
- target, atrybut, 293

 formnovalidate, atrybut, 291  
 formularze, 288, 289
 

- atrybuty, 289
- checkbox, pole, 300
- dane wejściowe, 296, 297
- daty, 304, 305
- DOM, 294
- radio, pole, 300

 frame, 27  
 frameset, 27

## G

geolocation, 308, 313  
 latitude, atrybut, 308  
 longitude, atrybut, 308  
 geolokalizacja, 308, 312  
 pobieranie mapy, 309  
 szerokość i długość geograficzna, 308  
 getContext(), 263  
 gęstość pikseli, 68  
 GIF, 183  
 zmiana rozmiaru, 191  
 Google Chrome, 30  
 Google Earth, 312, 313  
 Google Maps, 308  
 gradienty, 283, 284, 285  
 grafika  
 bitmapowa, 182  
 rozmiar, 182, 185, 186, 187, 188, 190, 191, 192  
 skala szarości, 194  
 warstwy, 184, 185  
 wektorowa, 182  
 wyświetlanie, 54  
 Graphic Information Format, *Patrz* GIF  
 grupowanie, 108

## H

h1 - h6, 24, 59  
 hash, 58  
 head, 24, 42, 43, 96  
 header, 22  
 height, atrybut, 235  
 hgroup, 22, 60, 61, 62  
 hierarchiczne typy odnośników, 144  
 hover, 90, 168  
 hr, 24, 108, 109, 110  
 href, atrybut, 54, 141  
 hreflang, atrybut, 141  
 HSL, 77, 78, 79  
 hsla, 84  
 html, 24, 96  
 lang, atrybut, 44  
 language, atrybut, 44  
 HTML, język, 20  
 HTML5, 37  
 kontynuowane znaczniki, 23, 24, 25  
 niekontynuowane znaczniki, 27  
 nowe elementy, 21, 22

przechowywanie, 314, 318  
 zdarzenia, 242  
 http-equiv, atrybut, 98

## I

i, 24  
 icon, 139  
 id, atrybut, 71, 292  
 identyfikatory, 145, 147  
 iframe, 24, 150, 151, 152, 173, 217  
 sandbox, atrybut, 152  
 seamless, atrybut, 152  
 srcdoc, atrybut, 152  
 ikony, 139, 140, 172  
 image, 258  
 img, 24, 54, 195  
 align, atrybut, 58, 195, 196  
 alt, atrybut, 58  
 href, atrybut, 54  
 src, atrybut, 54  
 indeks  
 dolny, 25  
 górny, 25  
 index, 144  
 input, 24  
 formnovalidate, atrybut, 291  
 required, atrybut, 291  
 ins, 24  
 Internet Explorer, 33  
 isindex, 27

## J

JavaScript, 98, 99, 100, 239  
 canvas, 263, 266, 267, 269  
 funkcje, 241  
 grafika, 197, 200  
 menu, 163  
 metody, 255, 256  
 obiekty, 255, 256  
 obsługa zdarzeń, 242, 243, 244, 245  
 pliki zewnętrzne, 240  
 przechowywanie danych, 250  
 tablice, 254  
 typy danych, 251, 252  
 umieszczanie na stronie, 240  
 wykrywanie zdarzeń, 242  
 zmienne, 250, 251



język skryptowy, 239  
 język znaczników hipertekstu, *Patrz* HTML, język  
 Joint Photographic Experts Group, *Patrz* JPEG  
 JPEG, 183, 184  
     zmiana rozmiaru, 188, 190

## K

kamery internetowe, 228, 229  
 kamery wideo, 229, 230  
 kbd, 24  
 keygen, 22  
 keywords, 97  
 kod, parsowanie, 38  
 kodek, 211  
 kolory, 76  
     bezpieczne dla przeglądarki, 183  
     HSL, 77, 78, 79  
     przejrzystość, 84, 85  
     RGB, 77, 78, 80, 81  
     schematy, 86, 87, 88  
     ustawienia heksadecymalne, 81, 83  
     używanie nazw, 76, 77  
     wartości procentowe, 77, 78  
 koła, 283  
 komentarze  
     CSS, 72  
     HTML, 23, 45, 46, 48  
 komórka, 128  
 kompresja  
     bezstratna, 183  
     stratna, 183  
 konteksty przeglądania, 148, 150, 151  
 kontener, 20  
 kotwice, 145, 147  
 krzywe, 279  
 Kuler, 86, 87

## L

label, 24  
 lang, atrybut, 44  
 language, atrybut, 44  
 last, 145  
 latitude, atrybut, 308  
 legend, 24, 303  
 li, 24  
 linia pozioma, 108, 110  
 link, 24, 41, 136  
     href, atrybut, 141  
     hreflang, atrybut, 141

media, atrybut, 141  
 rel, atrybut, 136, 142  
 sizes, atrybut, 141  
 title, atrybut, 141  
 type, atrybut, 141  
 list, atrybut, 298  
 listy, 104, 106  
     nienumerowane, 25, 107  
     numerowane, 25, 107  
     w nawigacji, 158, 159, 160  
 ilustrator, 200  
 longitude, atrybut, 308  
 loop, atrybut, 209, 233

## Ł

łuki, 280, 282, 283

## M

mail(), 337, 341  
 mailto, 142  
 map, 24  
 mark, 22, 24  
 marquee, 27  
 media, atrybut, 141  
 menu, 24  
 meta, 24, 54, 96, 97, 98  
     content, atrybut, 97  
     http-equiv, atrybut, 98  
     name, atrybut, 97  
 metadane, 97  
 meter, 22  
 method, atrybut, 291  
 Microsoft Internet Explorer, 33  
 miniatury, 173  
 Miro Video Converter, aplikacja, 225, 226  
 Mobile Safari, 32, 33  
 Mosaic, 29  
 Mozilla Firefox, 29  
 multicol, 27  
 Muybridge, Eadweard, 116

## N

name, atrybut, 97, 292  
 nav, 22, 155  
 navigator.appVersion, 197  
 navigator.platform, 253

nawigacja, 155, 156  
 globalna, 157, 158, 160  
 grafika, 172  
 iframe, 173, 175  
 listy, 158, 159, 160  
 pionowa, 167, 169  
 pozioma, 167  
 projekt, 156  
 pseudoklasy, 167, 168  
 rozwijane menu, 160  
 spójność, 165  
 testowanie, 157  
 Netscape Navigator, 29  
 next, 145  
 nobr, 27  
 noframes, 27  
 noscript, 27  
 novalidate, atrybut, 290

## 0

Obiektowy Model Dokumentu, *Patrz* DOM

obiekty

JavaScript, 255, 256  
 PHP, 334

object, 25, 324

obsługa zdarzeń, 242, 243, 244, 245

odniesienie bezwzględne, 113

odniesienie względne, 114, 115

odnośniki, 136

atrybuty, 141  
 ikony, 139, 140  
 strony, 141  
 typy, 144

odświeżanie strony, 98

ol, 25, 106

onionskin, 31

Opera, 32

Opera Mini, 32

optgroup, 25, 162, 163

option, 25, 160, 161

output, 22, 248, 250

## P

p, 25, 104

padding, 124

paleta kolorów, 87

integracja ze stroną, 88

paragraf, 25  
 param, 25  
 parsowanie kodu, 38  
 Photoshop, 188  
 PHP, 325, 326, 328, 330, 332  
 konfiguracja, 327  
 obiekty, 334  
 operatory, 335, 336  
 przechowywanie danych, 332  
 stałe, 333  
 tablice, 333  
 testowanie, 327  
 weryfikacja danych, 330  
 zmienne, 332

piksele, gęstość, 68

pliki

organizacja, 113  
 rozszerzenia, 38, 40  
 typy, 38, 40

pływające ramki, 150, 151, 152, 172

urządzenia przenośne, 175, 176

PNG, 184, 185

zmiana rozmiaru, 191

podpisy, 111

pop(), 255, 256

Portable Network Graphics, *Patrz* PNG

poster, atrybut, 232

pre, 25

prefetch, 140, 141

preload, atrybut, 208, 209, 233

prev, 145

progress, 22

przechowywanie

lokalne, 318  
 w sesji, 314

przeglądarki, 28

Apple Safari, 32

Google Chrome, 30

Microsoft Internet Explorer, 33

Mozilla Firefox, 29

Opera, 32

przejrzystość, 84, 85, 86

przycisk, 23

pseudoklasy, 167

## Q

q, 25

## R

radiansy, 280  
 radio, pole, 300  
 refresh, 98  
 Rejestrator dźwięku, aplikacja, 212, 213, 214  
 rel, atrybut, 136, 142  
 required, atrybut, 291  
 RGB, 77, 80, 81  
 rgba, 84, 273  
 rowspan, atrybut, 128, 130  
 rozszerzenia plików, 38, 40  
   zmiana ustawień, 39, 40  
 rozwijane menu, 160  
 rp, 22  
 rt, 22  
 ruby, 22  
 rysunki, podpis, 111

## S

s, 27  
 Safari, 32  
 samp, 25  
 sandbox, atrybut, 152  
 Scalable Vector Graphics, *Patrz SVG*  
 schematy kolorów, 86, 87, 88  
 script, 25, 41, 100  
 seamless, atrybut, 152  
 section, 22  
 sekcje, 100, 101, 103, 104  
 sekwencyjne typy odnośników, 144  
 select, 25, 160, 161, 163  
 sizes, atrybut, 141  
 skala szarości, 194  
 słowa kluczowe, 97  
 small, 25  
 Sound Studio, aplikacja, 214  
 source, 22, 210, 222, 231  
   type, atrybut, 210, 211, 231  
 spacer, 27  
 span, 25, 64  
 src, atrybut, 54, 206, 222  
 srcdoc, atrybut, 152  
 stopka, 104  
 stopnie, 280  
   zamiana na radiansy, 280  
 strike, 27  
 strokeRect(), 267  
 strona  
   efekty dźwiękowe, integracja, 217  
   elementy, 54

  porządkowanie zawartości, 104  
   struktura, 57  
   zagnieżdżanie, 151  
 strong, 25  
 style, 25, 62, 63  
 style wpisane, 62, 68, 69  
 sub, 25  
 summary, 22  
 sup, 25  
 SVG, 181, 182  
   konwersja z plików .ai, 200  
   zmiana rozmiaru, 192  
 systemy numeryczne, 82

## T

tabele, 25, 26, 117, 118, 121  
   elementy, 121  
   nadawanie stylu, 122  
   nagłówek, 24  
   obramowanie, 123, 124  
   tło, 125  
   złożone, 127, 128  
 table, 25, 117, 118, 121  
 table-cell, 118  
 tablice  
   JavaScript, 254  
   PHP, 333  
 target, atrybut, 148, 293  
 tbody, 25  
 td, 25, 121, 128  
   colspan, atrybut, 128, 130  
   rowspan, atrybut, 128, 130  
 tekst  
   cytowany, 23, 25  
   definicja, 24  
   kierunek wyświetlania, 23  
   kursywa, 24  
   mały, 25  
   nagłówki, 24  
   owijanie wokół obrazka, 58  
   pogrubienie, 23  
   usunięty, 24  
   wyróżniony, 24  
   wyświetlanie, 54  
   zaakcentowany, 25  
   zachowanie oryginalnego formatowania, 25  
   złamanie, 55, 56  
 textarea, 25  
 text-decoration, 90

TextEdit, ustawienia, 40  
 tfoot, 25  
 th, 25, 121  
 thead, 25  
 this, 257  
 thumbnails, 173  
 Tidwell, Jennifer, 156, 166  
 time, 22  
 title, 25, 42  
 title, atrybut, 141  
 tło, 85  
 tr, 25, 121  
 tt, 27  
 Tufte, Edward, 123, 156  
 twarda spacja, 50  
 type, atrybut, 141, 210, 211, 231  
 tytuł strony, 42

## U

u, 27  
 ul, 25, 106  
 Uniform Resource Locator, *Patrz* URL  
 up, 144  
 URL, 54

## V

var, 25  
 video, 22, 222, 231
 

- autoplay, atrybut, 234
- controls, atrybut, 234
- height, atrybut, 235
- loop, atrybut, 233
- poster, atrybut, 232
- preload, atrybut, 233
- src, atrybut, 222
- width, atrybut, 235

## W

W3C, 118  
 wbr, 22, 55, 56  
 WebM, 225  
 wideo, 221, 222
 

- formaty, 222, 223, 225
- kompatybilność, 224, 225
- tworzenie, 228, 229, 230

 width, atrybut, 235  
 wiersz, złamanie, 55, 56, 57  
 World Wide Web Consortium, *Patrz* W3C  
 współrzędne kartezjańskie, 262  
 wstępne pobieranie, 140, 141  
 wykrywanie zdarzeń, 242

## Z

zagnieżdżone arkusze stylów, 63  
 zagnieżdżony kontekst przeglądania, 151  
 zakomentowanie, 46  
 zdjęcia, oświetlenie, 191  
 zewnętrzne arkusze stylów, 62, 64  
 zmienne
 

- JavaScript, 250, 251
- PHP, 332

 znaczniki, 20, 21, 41
 

- atrybuty, 21
- części, 43
- elementy, 21
- końca, 41
- początku, 41
- podstawowe, 41
- zagnieżdżanie, 48

## WYKORZYSTAJ WSZYSTKIE MOŻLIWOŚCI, JAKIE OFERUJE HTML5. TWÓRZ ŚWIETNIE WYGLĄDAJĄCE, INTERAKTYWNE STRONY NOWEJ GENERACJI!

Po latach intensywnego rozwoju i niemal corocznego wprowadzania nowych, ulepszonych wersji nagle w 2000 roku dla języka HTML nastąpił czas zastoju. HTML4 zdecydowanie potrzebował gruntownych zmian. Przełom nastąpił w 2008 roku, gdy organizacja W3C opublikowała pierwszy szkic standardu HTML5. Spotkał się on z tak entuzjastycznym przyjęciem, że od samego początku otrzymał mocne wsparcie rynku. Wciąż rozwijany HTML5 ma do zaoferowania sporo nowych możliwości, które dają ambitnym webmasterom i programistom duże pole do popisu! Czy jesteś gotów po nie sięgnąć?

Ten podręcznik pokaże Ci, jak zacząć pracę z najnowszą wersją HTML5, używać znaczników tego języka oraz różnych atrybutów i wartości. Dowiesz się, jak pracować z CSS oraz zapewniać swoim stronom doskonały wygląd i poprawne wyświetlanie na wszelkich możliwych nośnikach. Zobacysz, jak zoptymalizować media w sieci, i zaczniesz sprawnie korzystać z tekstu, grafiki, audio, wideo czy nawigacji w przeglądarkach kompatybilnych z HTML5. Nauczysz się używać elementu canvas, autor wprowadzi Cię w podstawy JavaScriptu i obiektowego modelu dokumentu (DOM) w HTML5. Ponadto zaczniesz stosować PHP do przetwarzania informacji wprowadzanych w postaci formularzy oraz przeczytasz o nowych obiektach geolocation i ich zaskakujących właściwościach. Skorzystaj z tej niezwykłej książki, a całą wiedzę, której potrzebujesz, by zacząć działać, będziesz mieć w zasięgu ręki!

Dzięki tej książce nauczysz się m.in., jak:

- pracować ze znacznikami HTML5 i CSS3
- zaprojektować strukturę witryny
- pracować z wartościami kolorów
- stworzyć przejrzystą i intuicyjną nawigację strony
- umieszczać grafikę oraz pliki audio i wideo na stronach HTML5
- wykorzystywać moc elementu canvas w HTML5
- korzystać z formularzy HTML5
- tworzyć interaktywność, zagnieżdżać obiekty i przechowywać informacje

**Bill Sanders** jest zawodowym programistą, zajmującym się tworzeniem stron internetowych, projektowaniem informacji i interfejsów. Prowadzi także kursy z takich obszarów, jak HTML5, CSS3, Adobe Flash, ActionScript 3.0, ASP.NET, C# czy PHP. Pracuje na Uniwersytecie Hartford, a na swoim koncie ma już kilka bestsellerowych podręczników, m.in. na temat JavaScriptu i ActionScriptu 3.0.

nr katalogowy: 6627



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



Zamówienia telefoniczne:  
**0 801 339900**



**0 601 339900**



**Helion**

Sprawdź najnowsze promocje:

- <http://helion.pl/promocje>
- Książki najchętniej czytane:
- <http://helion.pl/bestsellery>
- Zamów informacje o nowościach
- <http://helion.pl/nowosci>

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>

**helion.pl**  
księgarnia  
internetowa

 **WILEY**

ISBN 978-83-246-3317-3



Cena 59,00 zł

9 788324 633173

Informatyka w najlepszym wydaniu