

Chuck Hudson
Tom Leadbetter


ADDISON-WESLEY

HTML5

Podręcznik programisty

Najlepsze przepisy na udane aplikacje internetowe!



Tytuł oryginału: HTML5 Developer's Cookbook

Tłumaczenie: Grzegorz Balcerek

ISBN: 978-83-246-4886-3

Authorized translation from the English language edition, entitled: HTML5 DEVELOPER'S COOKBOOK; ISBN 0321769384; by Chuck Hudson and Tom Leadbetter; published by Pearson Education, Inc, publishing as Addison Wesley.
Copyright © 2012 Pearson Education, Inc.

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 Pearson Education, Inc.

Polish language edition published by HELION S.A. Copyright © 2013.

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.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION nie ponoszą 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
WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
<http://helion.pl/user/opinie/html5p>
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:
<ftp://ftp.helion.pl/przyklady/html5p.zip>

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

Wstęp	13
Podziękowania	21
O autorach	23
Rozdział 1 Nowe elementy strukturalne w HTML5	25
PRZEPIS DLA POCZĄTKUJĄCYCH	
Budowa startowego dokumentu HTML5	26
doctype	26
Kodowanie znaków	26
JavaScript i łącza CSS	27
Styl używanej składni	27
Skąd pochodzą wszystkie nowe elementy?	28
PRZEPIS DLA POCZĄTKUJĄCYCH	
Wykorzystanie elementu header do utworzenia nagłówka witryny	29
PRZEPIS DLA POCZĄTKUJĄCYCH	
Wykorzystanie elementu hgroup do grupowania nagłówków	31
PRZEPIS DLA POCZĄTKUJĄCYCH	
Tworzenie nawigacji przy użyciu elementu nav	31
PRZEPIS DLA POCZĄTKUJĄCYCH	
Użycie elementu article	34
PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
Grupowanie treści przy użyciu elementu section	35
Którego elementu powinieneś używać: article czy section?	36
PRZEPIS DLA POCZĄTKUJĄCYCH	
Tworzenie paska bocznego za pomocą elementu aside	37
PRZEPIS DLA POCZĄTKUJĄCYCH	
Użycie elementu footer	39
PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
Wykorzystanie narzędzia HTML5 Outliner do utworzenia prawidłowej struktury dokumentu	41
PRZEPIS DLA ZAAWANSOWANYCH	
Wykorzystanie wszystkich nowych elementów do utworzenia strony z wiadomościami	43

PRZEPIS DLA ZAAWANSOWANYCH	
Wykorzystanie wszystkich nowych elementów do utworzenia strony z wynikami wyszukiwania	46
Podsumowanie	50
Rozdział 2 Grupowanie, poziom tekstu i zmiany semantyczne	51
PRZEPIS DLA POCZĄTKUJĄCYCH	
Oznaczenie rysunków i ich podpisów elementami figure i figcaption	51
PRZEPIS DLA POCZĄTKUJĄCYCH	
Oznaczenie daty i czasu za pomocą elementu time	53
PRZEPIS DLA POCZĄTKUJĄCYCH	
Tworzenie widżetu przełącznika za pomocą elementu details	54
PRZEPIS DLA POCZĄTKUJĄCYCH	
Użycie elementu address do danych kontaktowych	56
PRZEPIS DLA POCZĄTKUJĄCYCH	
Podświetlanie tekstu za pomocą elementu mark	57
PRZEPIS DLA POCZĄTKUJĄCYCH	
Użycie elementu s do pokazania niepoprawnej lub nieistotnej treści	58
Zmiany dotyczące istniejących elementów	58
Element cite	58
Element ol	59
Element dl	61
Element small	63
Elementy b oraz strong	63
Elementy i oraz em	64
Element abbr	64
Element hr	64
Elementy, których już nie ma	64
PRZEPIS DLA POCZĄTKUJĄCYCH	
Opakowywanie elementów łączami	65
PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
Dodawanie informacji semantycznych za pomocą mikro danych	66
PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
Stosowanie WAI-ARIA z HTML5	67
PRZEPIS DLA ZAAWANSOWANYCH	
Oznaczenie komentarzami strony z artykułem	69
Podsumowanie	71
Rozdział 3 Obsługa przeglądarek w HTML5	73
PRZEPIS DLA POCZĄTKUJĄCYCH	
Obsługa Internet Explorera	73
Użycie JavaScriptu do uzyskania kompatybilności z HTML5	73
Uzyskiwanie kompatybilności CSS-a	74
Szablony (boilerplates)	75
PRZEPIS DLA POCZĄTKUJĄCYCH	
Testowanie działania nowych możliwości HTML5	75
PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
Wykorzystanie jQuery do zastąpienia kalendarza	77

PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
Wykorzystanie biblioteki Modernizr do wykrywania możliwości	79
Polyfilling (wielowypełnianie)	82
Użyteczne strony sprawdzające HTML5	84
Podsumowanie	84
Rozdział 4 Nowe techniki dotyczące układu i stylizacji w CSS3	85
PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
Tworzenie dostosowującego się projektu za pomocą CSS3 Media Queries	85
Kiedy używać zapytań o media	90
Obsługa iPhone'a i urządzeń z systemem Android	91
PRZEPIS DLA POCZĄTKUJĄCYCH	
Korzystanie z własnych czcionek przy użyciu @font-face	92
Formaty plików i działanie czcionek w różnych przeglądarkach	93
Usługi związane z krojami pisma	94
PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
Tworzenie przycisków za pomocą gradientów CSS i wielu tła	95
PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
Upiększanie witryny za pomocą transformacji i przejść	98
PRZEPIS DLA ZAAWANSOWANYCH	
Tworzenie animacji za pomocą CSS-a	103
Podsumowanie	106
Rozdział 5 Formularze internetowe HTML5	107
Walidacja	107
Typy wejściowe HTML 4	107
PRZEPIS DLA POCZĄTKUJĄCYCH	
Tworzenie formularza do danych kontaktowych	109
input type="email"	109
input type="tel"	110
input type="url"	111
PRZEPIS DLA POCZĄTKUJĄCYCH	
Tworzenie formularza wyszukiwania przy użyciu input type="search"	112
PRZEPIS DLA POCZĄTKUJĄCYCH	
Tworzenie kontrolki kalendarza i czasu	113
input type="datetime"	113
input type="datetime-local"	113
input type="date"	114
input type="time"	114
input type="month"	115
input type="week"	115
Nakładanie ograniczeń na daty i godziny	115
PRZEPIS DLA POCZĄTKUJĄCYCH	
Tworzenie pola do wprowadzania liczby	116
PRZEPIS DLA POCZĄTKUJĄCYCH	
Tworzenie suwaka (bez potrzeby użycia JavaScriptu)	117
PRZEPIS DLA POCZĄTKUJĄCYCH	
Tworzenie próbnika kolorów	117

PRZEPIS DLA POCZĄTKUJĄCYCH	
Wyświetlanie wyników za pomocą elementu output	118
PRZEPIS DLA POCZĄTKUJĄCYCH	
Użycie tekstu zastępczego formularza	119
PRZEPIS DLA POCZĄTKUJĄCYCH	
Autouzupełnianie za pomocą atrybutu list i elementu datalist	120
PRZEPIS DLA POCZĄTKUJĄCYCH	
Śledzenie postępu wykonania zadania za pomocą elementu progress	121
PRZEPIS DLA POCZĄTKUJĄCYCH	
Wskazywanie wyniku pomiaru za pomocą elementu meter	122
PRZEPIS DLA POCZĄTKUJĄCYCH	
Przechodzenie do elementu form po załadowaniu strony	123
PRZEPIS DLA POCZĄTKUJĄCYCH	
Zezwalanie na wiele wartości	124
PRZEPIS DLA POCZĄTKUJĄCYCH	
Prosta walidacja przy użyciu atrybutu required	125
PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
Pisanie własnych reguł walidacji	126
PRZEPIS DLA POCZĄTKUJĄCYCH	
Nakładanie ograniczeń na wprowadzane dane	126
step	126
min i max	127
formnovalidate i novalidate	127
PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
Dostosowywanie formularza i nadawanie mu stylu	128
Komunikaty błędów	129
PRZEPIS DLA ZAAWANSOWANYCH	
Wszystko razem — tworzenie formularza rejestracyjnego	130
Podsumowanie	132
Rozdział 6 Rysowanie na płótnie	133
Element canvas — informacje ogólne	133
Początki	134
Współrzędne x i y	135
PRZEPIS DLA POCZĄTKUJĄCYCH	
Nałożenie siatki na płótno	135
Narzędzia płótna	139
PRZEPIS DLA POCZĄTKUJĄCYCH	
Tworzenie prostych kształtów i linii	139
Rysowanie i formatowanie prostokąta lub kwadratu	139
Stosowanie gradientów w kształtach	141
Rysowanie linii i ścieżek	142
PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
Rysowanie wielokątów za pomocą ścieżki	144
PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
Rysowanie łuków i okręgów	146
Rysowanie krzywych	147

PRZEPIS DLA POCZĄTKUJĄCYCH	
Dodawanie tekstu	147
PRZEPIS DLA POCZĄTKUJĄCYCH	
Wstawianie obrazka	148
PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
Prycinanie obrazka	149
PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
Animacja mapy sprite'ów	150
Przekształcenia płótna	153
PRZEPIS DLA ZAAWANSOWANYCH	
Animowanie obrazka	154
PRZEPIS DLA ZAAWANSOWANYCH	
Animacja pionowego wykresu słupkowego	157
Podsumowanie	162
Rozdział 7 Osadzanie wideo w HTML5	163
PRZEPIS DLA POCZĄTKUJĄCYCH	
Dołączanie filmów za pomocą elementu video	163
Obsługa w przeglądarkach i urządzeniach	164
HTML5 i kodeki wideo	164
Dlaczego kodeki powinny Cię interesować?	165
PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
Włączanie wideo we wszystkich przeglądarkach	166
Dodawanie zastępczej treści dla starszych przeglądarek	166
Nowe atrybuty wideo	169
PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
Tworzenie wideo z napisami	172
Inne możliwości formatowania napisów	174
API mediów	175
PRZEPIS DLA ZAAWANSOWANYCH	
Tworzenie niestandardowych kontroltek	176
Podsumowanie	183
Rozdział 8 Osadzanie dźwięku w HTML5	185
PRZEPIS DLA POCZĄTKUJĄCYCH	
Umieszczanie dźwięku za pomocą elementu audio	185
PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
Umieszczanie dźwięku we wszystkich przeglądarkach	186
Dodawanie zastępczej treści dla starszych przeglądarek	186
Nowe atrybuty elementu audio	187
API mediów	189
PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
Tworzenie miksera	190
PRZEPIS DLA ZAAWANSOWANYCH	
Dodawanie internetowego radia	194
Podsumowanie	201

Rozdział 9	Dokonywanie zmian w historii przeglądarki	203
	Podstawy historii	203
	Kompatybilność przeglądarek	204
	PRZEPIS DLA POCZĄTKUJĄCYCH	
	Dodawanie do historii wpisów za pomocą pushState	204
	PRZEPIS DLA POCZĄTKUJĄCYCH	
	Tworzenie przeglądarki obrazków	207
	PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
	Pobieranie stanu w przeglądarce obrazków	209
	PRZEPIS DLA POCZĄTKUJĄCYCH	
	Zmiana historii za pomocą replaceState	211
	PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
	Zmiana historii strony	213
	PRZEPIS DLA ZAAWANSOWANYCH	
	Używanie zaawansowanych obiektów danych stanu do przenoszenia informacji pomiędzy stronami	215
	PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
	Testowanie bezpieczeństwa historii	218
	Pomocne biblioteki	220
	Podsumowanie	221
Rozdział 10	Wykorzystanie Geolocation API do uzyskania informacji o położeniu geograficznym	223
	Omówienie Geolocation API	223
	Kompatybilność przeglądarek	224
	Gdzie na świecie się znajdujesz — getCurrentPosition	224
	PRZEPIS DLA POCZĄTKUJĄCYCH	
	Ustalanie położenia geograficznego za pomocą zwykłego wywołania getCurrentPosition	225
	Poufność informacji o położeniu	228
	PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
	Wyświetlanie na mapie informacji o lokalizacji przy użyciu getCurrentPosition	228
	PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
	Określanie odległości za pomocą opcji lokalizacji	233
	PRZEPIS DLA ZAAWANSOWANYCH	
	Podążanie za poruszającym się obiektem dzięki watchPosition	239
	Podsumowanie	245
Rozdział 11	Przechowywanie danych po stronie klienta	247
	Przechowywanie danych po stronie klienta — przegląd	247
	Bezpieczeństwo danych	248
	Klucze i wartości — sessionStorage i localStorage	249
	PRZEPIS DLA POCZĄTKUJĄCYCH	
	Pobieranie i ustawianie danych w pamięci sesji	250
	Narzędzia dla programistów Chrome służące do oglądania zawartości pamięci	252
	PRZEPIS DLA POCZĄTKUJĄCYCH	
	Nadawanie stylów z pamięci sesji	253

PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
Zapisywanie formularzy za pomocą lokalnego przechowywania danych	257
PRZEPIS DLA ZAAWANSOWANYCH	
Przechwytywanie zdarzeń w pamięci lokalnej	261
Web SQL Database API	267
PRZEPIS DLA ZAAWANSOWANYCH	
Użycie sieciowej bazy danych do stworzenia listy zakupów	270
Podsumowanie	277
Rozdział 12 Komunikacja i wątki	279
Przegląd WebSocket API	279
PRZEPIS DLA POCZĄTKUJĄCYCH	
Rozmawianie przez gniazda sieciowe	280
Realizacja wielowątkowości za pomocą wątków roboczych	283
PRZEPIS DLA POCZĄTKUJĄCYCH	
Tworzenie wątku roboczego	285
PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
Dodawanie dwukierunkowej komunikacji	288
PRZEPIS DLA ZAAWANSOWANYCH	
Wykorzystywanie współdzielonych wątków roboczych	291
Podsumowanie	296
Rozdział 13 Zachowanie przeglądarek w HTML5	297
Drag and Drop API	297
PRZEPIS DLA POCZĄTKUJĄCYCH	
Przeciąganie i upuszczanie pomiędzy elementami div	299
PRZEPIS DLA ZAAWANSOWANYCH	
Wykorzystanie zdarzeń i obiektu dataTransfer	302
Pamięć podręczna aplikacji i jej API	309
Pamięć podręczna przeglądarki a bezpieczeństwo	310
Odwołania do pliku manifestu	310
PRZEPIS DLA POCZĄTKUJĄCYCH	
Tworzenie pliku manifestu	310
CACHE	311
FALLBACK	311
NETWORK	312
Aktualizacja pamięci podręcznej poprzez manifest	312
PRZEPIS DLA POCZĄTKUJĄCYCH	
Korzystanie ze stron internetowych offline	312
API pamięci podręcznej aplikacji	315
Notification API	316
Uprawnienia dotyczące powiadomień	317
Kompatybilność przeglądarek	318
PRZEPIS DLA POCZĄTKUJĄCYCH	
Wyświetlanie prostego powiadomienia	318
PRZEPIS DLA ZAAWANSOWANYCH	
Tworzenie strony powiadomień o tweetach	322
Podsumowanie	329

Rozdział 14 Praca z plikami lokalnymi	331
Przegląd File API	331
Bezpieczeństwo File API	331
PRZEPIS DLA POCZĄTKUJĄCYCH	
Odczyt atrybutów pliku	332
PRZEPIS DLA POCZĄTKUJĄCYCH	
Przetwarzanie wielu plików za pomocą przeciągania i upuszczania	334
Interfejs FileReader	337
PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
Podgląd obrazków przy użyciu readAsDataURL	338
PRZEPIS DLA ZAAWANSOWANYCH	
Parsowanie pliku CSV za pomocą readAsText	342
Rozszerzone specyfikacje File API	349
PRZEPIS DLA ZAAWANSOWANYCH	
Tworzenie lokalnego pliku	350
Podsumowanie	355
Rozdział 15 Integracja z urządzeniami przenośnymi	357
Krótką historia API urządzeń	357
Contacts API	359
PRZEPIS DLA POCZĄTKUJĄCYCH	
Pobieranie wszystkich kontaktów i numerów telefonu	360
Messaging API	363
Network Information API	364
HTML Media Capture	365
PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
Przechwytywanie obrazu za pomocą elementu input typu file	365
DeviceOrientation Event Specification	368
PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH	
Tworzenie poziomiczy	369
Podsumowanie	373
Przepisy	375
Skorowidz	379

Nowe techniki dotyczące układu i stylizacji w CSS3

CSS poziomu 3 (CSS3) jest najnowszym wydaniem specyfikacji CSS. Zostało ono oparte na poprzednich wersjach, zawiera jednak nowe możliwości, które możesz zaimplementować w najnowszych przeglądarkach, by poprawić wygląd, łatwość użycia, dostępność i wydajność swoich stron internetowych. CSS to nie HTML5 — wiemy o tym i Ty także powinieneś to wiedzieć, ale dołączyliśmy ten rozdział, ponieważ często współdziałają one ręką w rękę, a my chcemy pokazać Ci tylko niektóre z możliwości dostępnych w CSS3, abyś mógł nadać Twoim witrynom HTML5 fantastyczny wygląd. W tym rozdziale powiemy sobie o witrynach reagujących na zmiany, projektowanych z użyciem Media Queries, zewnętrznych czcionek, gradientów, przejść i transformacji oraz, na końcu, animacji. Nie opisujemy wszystkich nowych możliwości CSS3; odkrywamy jedynie rąbek tego, co jest dostępne.

PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH Tworzenie dostosowującego się projektu za pomocą CSS3 Media Queries

Coraz częściej z internetu korzysta się przy użyciu urządzeń mobilnych, a niektórzy szacują, że taki sposób dostępu prześcignie przeglądanie z komputerów stacjonarnych w ciągu pięciu lat. CSS3 Media Queries¹ daje Ci możliwość dostosowania strony do określonych rozdzielczości, a nawet orientacji ekranu, tak że możesz obsługiwać smartfony, takie jak iPhone, albo telefony z systemem Android, jak również nowe tablety, takie jak iPad, i to wszystko za pomocą CSS-a. Tabela 4.1 pokazuje, które wersje przeglądarek obsługują zapytania o media w CSS3.

W przeszłości zapewne używałeś JavaScriptu do wykrywania telefonu komórkowego i dodawałeś osobny arkusz stylów albo przekierowywałeś użytkownika na specjalną witrynę dla urządzeń przenośnych. Mogłeś też korzystać z JavaScriptu do wykrywania, kiedy rozmiary okna przeglądarki zostały zmienione, i wtedy modyfikować style, by dostosować układ do okna. Jednak zawsze mieliśmy do dyspozycji podstawy zapytań o media, ponieważ zawsze mieliśmy możliwość ograniczenia stylów do ekranu lub wydruku:

¹ Zapytania o media — *przyp. tłum.*

Tabela 4.1. Obsługa zapytań o media z CSS3 w przeglądarkach i urządzeniach

Android	2.3+
Chrome	13.0+
Firefox	4.0+
Internet Explorer	9.0+
iOS Safari	4.0+
Opera	11.0+
Safari	5.0+

```
<link rel="stylesheet" href="screen.css" media="screen" />
<link rel="stylesheet" href="print.css" media="print" />
```

Teraz jednak możesz być bardziej wyrafinowany dzięki wykorzystaniu zapytań o media w CSS3 (listing 4.1).

Listing 4.1. Prosty przykład zapytań o media

```
<link rel="stylesheet" media="screen and
(max-device-width: 480px)" href="smartphone.css" />
<link rel="stylesheet" media="screen and
(min-width: 480px)" href="screen.css" />
```

Listing 4.1 odpytuje urządzenie, czy jego rozdzielczość pozioma to 480 pikseli albo mniej. Jeśli tak jest, możesz przypuszczać, że jest to smartfon, i ładujesz arkusz stylów *smartphone.css*. Następnie sprawdzasz, czy rozdzielczość to przynajmniej 480 pikseli; jeśli tak jest, możesz użyć innego arkusza stylów. Potencjalnie możesz mieć kilka różnych zapytań o media, jeśli chcesz obsługiwać rozmaite rodzaje urządzeń, rozdzielczości i orientacji ekranu. Możesz mieć arkusze stylów dla smartfonów, smartfonów z orientacją poziomą, ekranów androidowych, iPada, iPada z pionową orientacją, przeglądarek z rozdzielczością poziomą mniejszą od 800 pikseli albo przeglądarek z rozdzielczościami szerokoekranowymi. Tak naprawdę możesz być tak dokładny, jak chcesz.

Będziesz potrzebował dobrego sposobu organizacji wszystkich oddzielnych zapytań. Pamiętaj, że będziesz także miał wszystkie swoje główne arkusze stylów. Używanie podejścia z listingu 4.1 oznaczałoby dużą liczbę żądań HTTP w znaczniku head.

Zamiast tego możesz umieścić zapytania o media wewnątrz pliku CSS przy użyciu `@media`, jak pokazano na listingu 4.2.

Listing 4.2. Zapytania o media wewnątrz pliku CSS

```
body {background: black; color: #fff; font: normal 62.5%/1.5 tahoma, verdana, sans-serif;}
h1 {font-size: 2em;}
p {font-size: 1.4em;}
/* style dla smartfonów i bardzo małych rozdzielczości ekranu */
@media only screen and (min-width: 320px) and (max-width: 400px)
{
body {background: blue;}
}
/* style dla rozdzielczości ekranu większej od smartfonów, ale mniejszej lub równej 1024px */
@media only screen and (min-width: 401px) and (max-width: 1024px)
```

```
{
body {background: red;}
}
/* style dla ekranów o bardzo dużej rozdzielczości poziomej */
@media only screen and (min-width: 2000px)
{
body {background: green;}
}
```

W kodzie z listingu 4.2 zmienialiśmy kolor tła w zależności od rozdzielczości ekranu. Wszystkie zapytania o media są częścią tego samego dokumentu CSS, więc ważne jest, by był on odpowiednio zorganizowany — prawdopodobnie będziesz miał setki wierszy kodu Twojego projektu i dodatkowo CSS z zapytaniami o media. W zależności od konfiguracji Twojej witryny prostszym sposobem może być edytowanie zapytań o media, na przykład poprawienie `min-width: 2000px` na `min-width: 2500px`, wewnątrz pliku CSS zamiast na wszystkich stronach HTML.

Używając składni `and`, możesz łączyć różne zapytania. Możesz używać kilku właściwości, takich jak `width`, `height`, `device-width`, `device-height`, `orientation`, `aspect-ratio`, `device-aspect-ratio`, `color`, `color-index`, `monochrome`, `resolution`, `scan` i `grid`, wraz z rozpoznawanymi rodzajami mediów: `all`, `braille`, `embossed`, `handheld`, `print`, `projection`, `screen` i `speech`. Dzięki połączeniu ich ze sobą będziesz jednocześnie obsługiwać kilka różnych typów urządzeń i rozmiarów. Warto zauważyć, że `max-device-width` jest rozmiarem ekranu urządzenia (takiego jak iPhone), podczas gdy `max-width` jest tylko szerokością wyświetlającego obszaru (takiego jak okno przeglądarki); spójrz na różnice w poniższym kodzie:

```
/* urządzenia mniejsze od 480px; urządzenia przenośne, iPhone i zwykłe ekrany */
@media handheld and (max-width: 480px), screen and (max-device-width: 480px), screen and (max-width: 480px) {
/* tutaj umieszczasz style */
}
/* układ dla iPada w trybie poziomym */
@media only screen and (min-device-width: 768px) and (max-device-width: 1024px) and (orientation: landscape) {
/* tutaj umieszczasz style */
}
```

Wskazówka

Na kolejnych listingach użyliśmy kilku różnych rodzajów selektorów CSS. Pierwszym z nich jest selektor atrybutu użyty w przypadku `header[role=banner]`, a drugim — selektor pseudoklasy, `nth-xx()`, taki jak `section:nth-of-type(1)`, wykorzystany w kilku przypadkach. Selektory atrybutu nie są nowością w CSS3, ale w powiązaniu z nowymi rolami ARIA dają więcej możliwości używania selektorów wraz z CSS-em. Istnieje wiele nowych selektorów CSS3, włącznie z `nth-child()` i `nth-of-type()`. Korzystanie z tych nowych selektorów oznacza, że możesz zrezygnować z używania klas przy wielu elementach. Na przykład przy użyciu `nth-of-type()` możesz wybrać co drugi wiersz tabeli albo element listy. Więcej informacji na ten temat uzyskasz, czytając <http://quirksmode.org/css/nthchild.html>.

Wiele spośród tych nowych selektorów nie działa w starszych wersjach Internet Explorera, więc polecamy Selectivizr (<http://selectivizr.com/>) — łatwe w użyciu narzędzie javascriptowe, które powoduje, że Internet Explorer rozumie nowe selektory CSS.

Rysunek 4.1 pokazuje wynik wykorzystania przykładowego kodu HTML witryny z wiadomościami z rozdziału 1. Listing 4.3 zawiera skrócony HTML tej witryny. Kod CSS z listingu 4.4 jest przeznaczony dla domyślnego układu, jaki pokazano na rysunku 4.1.

Listing 4.3. *Skrócony HTML witryny z wiadomościami*

```

<body>

<header role="banner">
  <hgroup></hgroup>
</header>

<nav></nav>

<section> <!-- ten element section jest powtórzony jeszcze trzykrotnie -->
  <article>
    <header></header>
    
    <p></p>
  </article>
</section>

<aside></aside>

</body>

```

Listing 4.4. *CSS domyślnego układu ekranu*

```

<style>
* {margin: 0; padding: 0;}
body {background: #fff; color: #000; font: normal 62.5%/1.5 "Palatino Linotype", "Book Antiqua", Palatino,
↳serif; margin: 0 auto; width: 1260px}
header, nav, section, article, footer, aside {display: block;}
header[role=banner] {margin: 10px 0 20px; text-align: center;}
header[role=banner] h1 {background: url(images/logo.gif) top center no-repeat;font-size: 5em;
↳padding: 100px 0 0; text-transform: uppercase;}
header[role=banner] h2 {font-style: italic;}
header, nav {clear: both; width: 100%;}
nav {border-bottom: 1px dotted #ccc; padding-bottom: 20px; text-align: center;}
nav li {display: inline;}
nav li a {font-size: 1.4em; padding: .5em;}
section {float: left; margin: 0 0 10px; padding: 10px; width: 345px;}
section h1 {margin: 0 0 10px;}
section article {margin: 0 0 10px;}
section article header p {font-size: 1em; font-weight: bold; margin: 0 0 10px;}
section article img {float: left; margin: 0 5px 5px 0;}
section:nth-of-type(1) {clear: both; margin: 0 0 10px; min-height: 200px; padding: 1% 1% 1% 30%; position:
↳relative; width: 69%;}
section:nth-of-type(1) article {margin: 0;}
section:nth-of-type(1) article img {float: none; height: 200px; left: 0; position: absolute; top: 10px;
↳width: 360px}
section:nth-of-type(2) {margin-left: 0;}
aside {float: right; margin-left: 10px; width: 130px;}
aside img {border: 1px solid #ccc; display: block; margin: 0 auto 10px;}
footer {clear: both;}
h1 {font-size: 2em;}
p {font-size: 1.4em;}
</style>

```



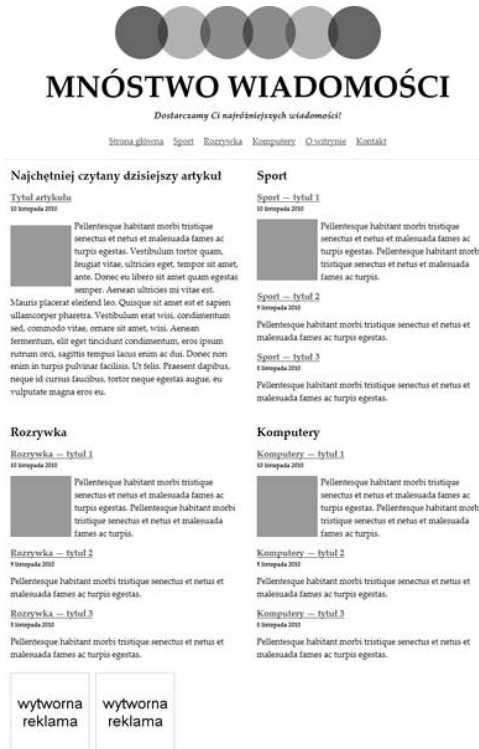
Rysunek 4.1. Witryna z wiadomościami z domyślnym CSS-em

Rysunek 4.2 pokazuje układ na mniejszym ekranie z wykorzystaniem zapytań o media z listingu 4.5.

Listing 4.5. CSS przeznaczony dla mniejszych ekranów

```
/* style dla rozdzielczości ekranu większych od smartfonów, ale mniejszych lub równych 1280px */
@media only screen and (min-width: 481px) and (max-width: 1259px)
{
  body {width: 800px;}
  section {margin: 0 0 0 10px;}
  section:nth-of-type(1), section {clear: none; float: left; padding: 10px; width: 375px;}
  section:nth-of-type(1) article img {float: left; height: auto; position: relative; width: auto;}
  section:nth-of-type(2) {margin-left: 10px;}
  section:nth-of-type(3) {clear: both; margin-left: 0;}
  aside {clear: both; float: left; width: 100%;}
  aside img {float: left; margin: 0 10px 0 0;}
}
```

Wreszcie, listing 4.6 zawiera zapytania o media w CSS-ie przeznaczone dla smartfonów albo znacznie zmniejszonego okna przeglądarki (rysunek 4.3). Zasadniczo tych kilka dodatkowych wierszy CSS-a sprawia, że w porównaniu z ekranem o mniejszych rozmiarach niektóre elementy są ukrywane lub zmniejszane.



Rysunek 4.2. Witryna z wiadomościami na mniejszym ekranie

Listing 4.6. CSS przeznaczony dla smartfonów

```

/* style dla smartfonów i bardzo małych rozdzielczości ekranu */
@media only screen and (max-width: 480px), only screen and (max-device-width: 480px) {
  body {width: 100%;}
  header[role=banner] h1 {background-image: url(images/logo-small.gif); font-size: 3em; padding: 50px 0 0;}
  section:nth-of-type(1), section {margin: 0 0 10px; min-height: inherit; padding: 0 1%; width: 98%;}
  header[role=banner] h2, img, section article p, aside {display: none;}
  section h2 a {border-bottom: 1px dotted #999; display: block; text-decoration: none;}
  nav, section article:last-of-type h2 a {border: none;}
}

```

Kiedy używać zapytań o media

Choć możesz tworzyć oddzielne style dla różnych urządzeń i rozmiarów ekranu, powstaje pytanie, czy zawsze powinieneś korzystać z zapytań o media. Bez wątplenia mogą być one użyteczne na komputerach stacjonarnych i laptopach, ale rynek urządzeń przenośnych od pewnego czasu się zmienia. Bardziej nowoczesne smartfony, takie jak iPhone i urządzenia z systemem Android, posiadają przeglądarki, które są prawie identyczne jak przeglądarki desktopowe i z większością rzeczy sobie radzą, a interaktywność urządzeń pozwala użytkownikowi stosunkowo łatwo wybierać i powiększać fragmenty witryn.



Rysunek 4.3. Witryna z wiadomościami na smartfonie

Korzystanie z zapytań o media oznacza na ogół pokazywanie lub ukrywanie treści w zależności od rozmiaru ekranu. Wiele witryn, takich jak <http://youtube.com/>, <http://facebook.com/>, <http://cnn.com/> oraz <http://nfl.com/>, wykrywa, czy użytkownik używa urządzenia przenośnego, i przekierowuje go na wersje przeznaczone dla takich urządzeń. Witryny te zawierają dużo treści z mnóstwem danych, zdjęć, filmów wideo, reklam, Flasha i innych rzeczy — gdyby korzystać tylko z zapytań o media, smartfon nadal musiałby pobierać wszystkie te dane, mimo że użytkownik nie mógłby ich zobaczyć. A zatem to, czy potrzebujesz jedynie nowych stylów, czy zupełnie oddzielnej witryny dla urządzeń przenośnych, zależy od treści witryny, ale jeśli masz zmienić jedynie układ i dodać kilka szczegółów, wtedy powinieneś prawdopodobnie użyć zapytań o media. Kilka znakomych sposobów ich użycia znajdziesz na <http://mediaqueri.es/>.

Obsługa iPhone'a i urządzeń z systemem Android

Jeśli używałeś zapytań o media do obsługi iPhone'a albo urządzeń z systemem Android, być może zauważyłeś, że nie uwzględniają one zmian CSS-a dotyczących małych ekranów. Jest tak dlatego, że nowoczesne przeglądarki na smartfonach mają duże możliwości i domyślnie wyświetlają stronę internetową tak samo jak przeglądarki desktopowe. Następnie przeglądarka zmniejsza witrynę, tak aby mieściła się ona na małym ekranie, co może często skutkować wyświetlaniem drobnego tekstu i obrazków, przez co użytkownik musi powiększać fragmenty witryny, by ją czytać i przeglądać. Możesz zmusić przeglądarkę do używania szerokości urządzenia jako szerokości obszaru, w którym przeglądarka pokazuje obraz, znanego również jako *viewport*. W tagu head należy umieścić następujący kod:

```
<meta name="viewport" content="width=device-width; initial-scale=1.0; " />
```

PRZEPIS DLA POCZĄTKUJĄCYCH

Korzystanie z własnych czcionek przy użyciu @font-face

Chociaż deklaracja @font-face nie pojawiła się dopiero w CSS3, została w nim uatrakcyjniona. Projektanci i twórcy stron internetowych mają teraz do dyspozycji wiele możliwości wyboru i sposobów implementacji czcionek w swojej witrynie. Tabela 4.2 pokazuje, które wersje każdej z przeglądarek obsługują @font-face.

Tabela 4.2. Obsługa @font-face w przeglądarkach i urządzeniach

Android	2.3+
Chrome	13.0+
Firefox	4.0+
Internet Explorer	6.0+
iOS Safari	4.0+
Opera	11.0+
Safari	5.0+

Uwaga

Deklaracja @font-face została włączona do specyfikacji CSS2 w 1998 roku, jednak była w zasadzie nieużywana z powodu słabej implementacji w przeglądarkach, zamieszania z rodzajami plików czcionek i obaw dotyczących problemów prawnych i licencyjnych związanych z wykorzystywaniem czcionek. Internet Explorer w wersji 4 obsługiwał nawet dodatkowe, zewnętrzne czcionki, choć tylko w formacie EOT.

Używając deklaracji @font-face, możesz osadzić na stronie swoje własne czcionki za pomocą jedynie kilku wierszy CSS-a. Na rysunku 4.4 pokazaliśmy czcionkę Anagram (dalej podamy informacje o tym, skąd zdobyć czcionki) w elemencie h1, chociaż może być ona użyta w dowolnym elemencie.



Dostarczamy Ci najróżniejszych wiadomości!

Rysunek 4.4. Własna czcionka wyświetlona przy użyciu @font-face

Listing 4.7 pokazuje kod potrzebny do uzyskania efektu z rysunku 4.4 i prezentuje przykład najprostszego użycia @font-face. Podajemy nazwę czcionki Anagram jako wartość właściwości font-family (można nazwać ją dowolnie), a później będziemy się ponownie do niej odwoływać; spójrz na wartości czcionek h1. W deklaracji @font-face właściwość src wykorzystuje ten sam katalog, w którym znajduje się strona HTML; jeśli chcesz, możesz umieścić czcionkę w innym katalogu.

Listing 4.7. Deklaracja @font-face w swojej najprostszej postaci

```
<style>
@font-face {
font-family: Anagram;
```

```

src: url('anagram.ttf');
}
h1 {
font-family: Anagram, Tahoma, Verdana, sans-serif;
font-size: 9em;
}
</style>
<header role="banner">
<hgroup>
<h1>Mnóstwo wiadomości</h1>
<h2>Dostarczamy Ci najróżniejszych wiadomości!</h2>
</hgroup>
</header>

```

Formaty plików i działanie czcionek w różnych przeglądarkach

Na listingu 4.7 użyliśmy tylko czcionek typu *.ttf* (TrueType/OpenType). Jak można się spodziewać, między przeglądarkami występują różnice implementacyjne. Istnieje wiele różnych formatów plików czcionek:

- Embedded OpenType (*.eot*),
- OpenType PS (*.otf*),
- TrueType/OpenType (*.ttf*),
- SVG (*.svg*),
- WOFF (*.woff*).

WOFF jest nowym sieciowym standardem krojów pisma. Internet Explorer w wersjach 8 i starszych wymaga czcionki *.eot*, ale Internet Explorer 9 obsługuje nowy format WOFF. A chociaż Safari obsługuje *.ttf* już od jakiegoś czasu, dopiero niedawno iPhone i iPad zaczęły wspierać ten format; wcześniej potrzebowały czcionki *.svg*. Powinieneś więc dostarczać tę samą czcionkę w różnych formatach. W tym celu musisz albo samodzielnie przekonwertować czcionki, albo użyć narzędzia, takiego jak Font Squirrel z <http://www.fontsquirrel.com/>. Narzędzie @font-face Generator pozwala Ci wgrać czcionki, a następnie przekształcić je na inny format. Generuje ono również odpowiedni CSS, który umożliwi działanie czcionki w różnych przeglądarkach, co pokazano na listingu 4.8.

Listing 4.8. Działający w różnych przeglądarkach @font-face

```

@font-face {
font-family: 'AnagramRegular';
src: url('type/anagram-webfont.eot');
src: url('type/anagram-webfont.eot?#iefix') format('embedded-opentype'),
url('type/anagram-webfont.woff') format('woff'),
url('type/anagram-webfont.ttf') format('truetype'),
url('type/anagram-webfont.svg#AnagramRegular') format('svg');
font-weight: normal;
font-style: normal;
}

```

Na listingu 4.8 odwołujemy się do czterech różnych formatów tej samej czcionki. Warto być może umieścić swoje czcionki w oddzielnych katalogach, tak jak robisz to z obrazkami. Kolejność czcionek

w powyższym kodzie jest związana z problemem dotyczącym działania Internet Explorera. Było już wiele wersji powyższego kodu, ale w miarę rozwoju przeglądarek zniknęło kilka problemów. Powyższy kod jest obecnie² dostarczany przez Font Squirrel, gdy generujesz pakiet z czcionkami, lecz oczywiście w przyszłości może się on zmienić.

Usługi związane z krojami pisma

Znalezienie odpowiedniego kroju pisma może być trudne, a znalezienie takiego, który może być użyty w sieci zgodnie z prawem, może być jeszcze trudniejsze. Font Squirrel pozwala Ci przekształcić czcionki na różne potrzebne formaty. By to zrobić, musisz zaznaczyć przycisk wyboru, przy którym jest napisane „Yes, the fonts I am uploading are legally eligible for web embedding”³. Jeśli go zaznaczysz, a nie jesteś w pełni przekonany, że tak jest, możesz mieć problemy prawne.

Jeśli potrzebujesz darmowej czcionki, sugerujemy przejrzanie biblioteki krojów na witrynie Font Squirrel, ponieważ ta witryna oferuje nie tylko generator czcionek — jej głównym celem jest udostępnianie czcionek, które są darmowe i które można wykorzystywać w zastosowaniach komercyjnych. Ponadto witryna nie zawiera po prostu jakichkolwiek darmowych czcionek, ale zawiera czcionki wysokiej jakości. Font Squirrel nie jest jedynym miejscem, w którym można pobrać darmowe czcionki do użytku w sieci, ale gdy będziesz szukał ich gdzie indziej, bądź świadomy aspektów prawnych.

Innym rozwiązaniem dotyczącym darmowych czcionek jest Google Font Directory (<http://www.google.com/webfonts>). Za pomocą dostępnego API bardzo łatwo skorzystać z oferowanych tam czcionek. Ponieważ czcionki znajdują się na serwerach Google, czasy ładowania są minimalne. Musisz tylko znaleźć czcionkę i wybrać „Quick-use”⁴. Narzędzie przygotowuje HTML i CSS, których potrzebujesz, a dodatkowo oferuje inne grubości czcionki, jeśli dana czcionka je posiada. Dostarczany przez to narzędzie kod HTML, który należy umieścić w znaczniku head, jest podobny do poniższego:

```
<link href='http://fonts.googleapis.com/css?family=Yanone+Kaffeesatz' rel='stylesheet' type='text/css'>
```

Google daje Ci również nazwę czcionki, którą możesz umieścić w swoim kodzie CSS:

```
h1 {font-family: 'Yanone Kaffeesatz', arial, serif;}
```

Tak, to naprawdę proste.

Wskazówka

W poprzednich przykładach mogłeś zauważyć, że po deklaracji dodatkowej czcionki podajemy listę niektórych „bezpiecznych” do stosowania w sieci czcionek, jak na przykład rodzinę czcionek 'Yanone Kaffeesatz', arial, serif. Rezerwową czcionkę podajemy, na wypadek gdyby cokolwiek poszło nie tak podczas pobierania dodatkowej czcionki. Może się pojawić problem z serwerem i przeglądarka może być niezdolna do pozyskania dodatkowej czcionki, którą chcemy wykorzystać, a poprzez zamieszczenie listy czcionek dajemy możliwość awaryjnego użycia czcionki, o której wiemy, że jest dostępna. Listy czcionek powinny być dodawane zawsze wtedy, kiedy używa się CSS-owej właściwości font-family.

² W kwietniu 2012 roku — *przyp. tłum.*

³ Tak, czcionki, które wgrzywam, mogą być zgodnie z prawem używane na witrynach internetowych — *przyp. tłum.*

⁴ Szybki wybór — *przyp. tłum.*

Jeśli Font Squirrel albo Google Font Directory nie mogą udostępnić czcionki, której szukasz, wypróbuj inne serwisy. Ponieważ licencjonowanie stanowi problem już od lat, wielu spośród głównych producentów czcionek zaczęło albo tworzyć swoje własne struktury pozwalające Ci zgodnie z prawem i bezpiecznie używać ich czcionek, albo współpracować z inną organizacją, która pomaga dostarczać czcionki.

Serwisy takie jak FontDeck i TypeKit umożliwiają Ci zarejestrowanie się i dokonanie wyboru spośród szeregu czcionek, z których wszystkie zostały poprawione w celu uzyskania lepszej czytelności na ekranie. FontDeck i TypeKit dostarczają czcionki na odrobinę różne sposoby, ale konfiguracja, wybór i aktywacja czcionek w obu serwisach nie powinny nikomu sprawiać problemów. Oba serwisy mają darmowe konta z pewnymi ograniczeniami i oba oferują różne pakiety cenowe.

PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH

Tworzenie przycisków za pomocą gradientów CSS i wielu tła

Gradientów CSS3 możesz używać do tworzenia miłych i subtelnych albo jakichś zwariowanych, pokracznych gradientów. Nie musisz tworzyć obrazków, a gradienty są skalowalne, więc kiedy już się z nimi oswoisz, sprawiają dużo mniej kłopotu w porównaniu z tworzeniem, edycją i ponownym tworzeniem obrazków, ponieważ możesz zrobić wszystko za pomocą kodu. Tabela 4.3 pokazuje, które wersje każdej z przeglądarek obsługują gradienty CSS3.

Tabela 4.3. Obsługa gradientów w przeglądarkach i urządzeniach

Android	2.3+
Chrome	13.0+
Firefox	4.0+
Internet Explorer	-
iOS Safari	4.0+
Opera	11.0+
Safari	5.0+

Listing 4.9 zawiera kod prostego gradientu od białego do czarnego koloru, który pokazano na rysunku 4.5.

Listing 4.9. Prosty liniowy gradient CSS

```
div {  
  height: 200px;  
  width: 200px;  
  background: url(gradient.gif); /* dla przeglądarek, które nie potrafią tworzyć gradientów */  
  background: -moz-linear-gradient(white, black);  
  background: -webkit-linear-gradient(white, black);  
  background: -linear-gradient(white, black);  
}
```



Rysunek 4.5. Prosty liniowy gradient CSS

Najpierw zostaje określony rodzaj gradientu (liniowy lub radialny), po czym następują nawiasy, w których zapisany jest początkowy i końcowy kolor gradientu. Zauważ, że kod ma cztery różne deklaracje. Pierwsza zawiera gradient w postaci pliku graficznego, na wypadek gdyby przeglądarka nie obsługiwała gradientów CSS3; `-moz-linear-gradient` jest przeznaczona dla przeglądarek Mozilla (Firefox); `-webkit-linear-gradient` dla przeglądarek opartych na WebKit (Safari i Chrome); ostatnia deklaracja `linear-gradient` jest oficjalną składnią gradientu CSS3, ale obecnie żadna przeglądarka jej nie obsługuje.

Uwaga

W 2008 roku WebKit był pierwszym silnikiem, który zaczął używać gradientów, lecz miał swoją własną składnię. W miarę jak gradienty CSS stawały się bardziej popularne, Mozilla zaimplementowała składnię znacznie bliższą tej z oficjalnej specyfikacji. Jednak w styczniu 2011 roku ogłoszono, że WebKit zmieni składnię gradientów i zacznie używać tego samego stylu co Mozilla i oficjalna specyfikacja, co jest wspaniałą wiadomością dla każdego, ponieważ oryginalna składnia WebKit była bardziej skomplikowana i nie była oparta na standardach. W tej książce używamy nowej składni WebKit.

Na listingu 4.10 używamy przycisków i kombinacji selektorów atrybutów, właściwości `box-shadow` i `border-radius`, wielu tła oraz gradientów CSS do uzyskania efektów z rysunku 4.6. We właściwości `background` oddzielamy poszczególne style tła przy użyciu przecinka. Wobec tego najpierw umieszczamy obrazek, z którego chcemy skorzystać, a potem ustawiamy style gradientu. Jeśli style gradientu byłyby użyte wcześniej, przykryłyby obrazek.

Listing 4.10. Efekty gradientu na przyciskach

```
input {
border: none;
-webkit-box-shadow: 0 1px 5px rgba(0, 0, 0, .4);
box-shadow: 0 1px 5px rgba(0, 0, 0, .4);
-webkit-border-radius: 10px;
border-radius: 10px;
cursor: pointer;
color: #fff;
font: bold 1.2em Arial, Helvetica, sans-serif;
margin: 0 10px 0 0;
padding: 10px 10px 10px 30px;
text-shadow: 0 2px 2px rgba(0, 0, 0, 0.25);
}
input[type="submit"] {
background: url(images/accept.png) 8px 55% no-repeat #91BD09;
background: url(images/accept.png) 8px 55% no-repeat, -webkit-linear-gradient(#91BD09, #578730);
background: url(images/accept.png) 8px 55% no-repeat, -moz-linear-gradient(#91BD09, #578730);
}
```

```
background: url(images/accept.png) 8px 55% no-repeat, -linear-gradient(#91BD09, #578730);
}

input[value="Anuluj"] {
background: url(images/cross.png) 8px 55% no-repeat #b53109;
background: url(images/cross.png) 8px 55% no-repeat, -webkit-linear-gradient(#b53109, #540303);
background: url(images/cross.png) 8px 55% no-repeat, -moz-linear-gradient(#b53109, #540303);
background: url(images/cross.png) 8px 55% no-repeat, -linear-gradient(#b53109, #540303);
}

input[type="reset"] {
background: url(images/error.png) 8px 55% no-repeat #f0bb18;
background: url(images/error.png) 8px 55% no-repeat, -webkit-linear-gradient(#f0bb18, #a46b07);
background: url(images/error.png) 8px 55% no-repeat, -moz-linear-gradient(#f0bb18, #a46b07);
background: url(images/error.png) 8px 55% no-repeat, -linear-gradient(#f0bb18, #a46b07);
}
```

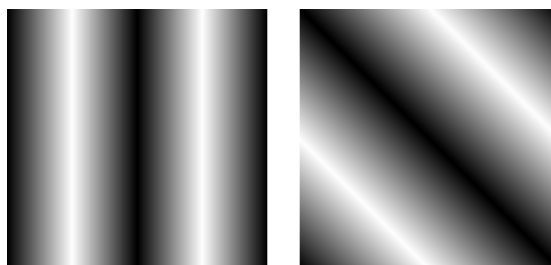


Rysunek 4.6. Gradienty CSS3 na przyciskach

Możesz uzyskać znacznie więcej niż tylko zwykłe przenikanie dwóch kolorów. Używając *zatrzymania*, możesz dodać wiele kolorów do gradientu, co pokazano na rysunku 4.7. Zatrzymania są dodawane po każdym przecinku i mogą zawierać pozycję, w której się zaczynają, jak pokazano na listingu 4.11.

Listing 4.11. Gradienty z wieloma zatrzymaniami

```
div {
height: 200px;
width: 200px;
background-color: #000;
background: -moz-linear-gradient(left, #000000 0%, #FFFFFF 25%, #000000 50%, #FFFFFF 75%, #000000 100%);
background: -webkit-linear-gradient(left, #000000 0%, #FFFFFF 25%, #000000 50%, #FFFFFF 75%, #000000 100%);
background: -linear-gradient(left, #000000 0%, #FFFFFF 25%, #000000 50%, #FFFFFF 75%, #000000 100%);
}
```



Rysunek 4.7. Gradienty CSS3 z wieloma zatrzymaniami

Listing 4.11 pokazuje kod użyty do utworzenia lewego obrazka z rysunku 4.7. Gradient staje się bardziej skomplikowany, ale zasadniczo każdy przecinek rozpoczyna nowy fragment informacji o gradientcie, w którym przekazujesz wartość koloru, a następnie procent oznaczający miejsce, w którym ten kolor się zaczyna.

Ponadto możesz obrócić gradient przez zmianę deklaracji `left`, na przykład na `45deg` albo `left top`, jak pokazano na listingu 4.12, który tworzy prawy obrazek z rysunku 4.7.

Listing 4.12. *Obrócone gradienty z wieloma zatrzymaniami*

```
div {
height: 200px;
width: 200px;
background-color: #000;
background: -moz-linear-gradient(45deg, #000000 0%, #FFFFFF 25%, #000000 50%, #FFFFFF 75%, #000000 100%);
background: -webkit-linear-gradient(45deg, #000000 0%, #FFFFFF 25%, #000000 50%, #FFFFFF 75%, #000000 100%);
background: -linear-gradient(45deg, #000000 0%, #FFFFFF 25%, #000000 50%, #FFFFFF 75%, #000000 100%);
}
```

Istnieje mnóstwo opcji związanych z gradientami CSS3, włączając w to tworzenie gradientów radialnych. Zalecamy wypróbowanie niektórych spośród generatorów gradientów, by nabrać wprawy:

- <http://colorzilla.com/gradient-editor/>,
- <http://display-inline.fr/projects/css-gradient/>.

PRZEPIS DLA ŚREDNIO ZAAWANSOWANYCH

Upiększanie witryny za pomocą transformacji i przejść

Dawniej twórcy stron internetowych musieli używać JavaScriptu albo wtyczek, takich jak Flash, żeby uzyskać na stronie internetowej efekty i animacje. Teraz jednak możesz robić takie rzeczy jak obracanie i zmiana rozmiaru elementów HTML przy użyciu jedynie CSS-a. Ponieważ żadna przeglądarka nie przyjęła jeszcze oficjalnej składni CSS-owej, musisz dodawać CSS dla każdego producenta. Oznacza to, że będziesz mieć zduplikowany CSS, ale na razie jest to konieczne, a nie wiąże się tak naprawdę z dużą ilością dodatkowej pracy. Tabela 4.4 pokazuje, które wersje każdej z przeglądarek obsługują transformacje i przejścia.

Tabela 4.4. *Obsługa transformacji i przejść w przeglądarkach i urządzeniach*

Android	2.3+
Chrome	13.0+
Firefox	4.0+
Internet Explorer	9.0+
iOS Safari	4.0+
Opera	11.0+
Safari	5.0+

Rysunek 4.8 pokazuje transformację obracającą zastosowaną do elementu `figure`.

Aby utworzyć efekt z rysunku 4.8, zastosuj właściwość `transform` z wartością `rotate` do elementów `figure` i `h1`. Jak pokazano na listingu 4.13, musisz użyć CSS-owych prefiksów kilku producentów, aby efekty działały w WebKit, Mozilli, Operze i Internet Explorerze.

To jest tytuł strony

Pellentesque habitant morbi tristique senectus et netus. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo. erat volutpat. Nam dui mi, tincidunt quis, accumsan porttitor, facilisis luctus, metus

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo. Quisque sit amet est et sapien ullamcorper pharetra. Vestibulum erat wisi, condimentum sed, commodo vitae, ornare sit amet, wisi. Aenean fermentum, elit eget tincidunt condimentum, eros ipsum rutrum orci, sagittis tempus lacus enim ac dui. Donec non enim in turpis pulvinar facilisis. Ut felis.



Rysunek 4.8. Obrazek i tekst obrócone za pomocą CSS-a

Listing 4.13. Transformacje CSS

```
* {
margin: 0;
padding: 0;
}
body {
font: normal .9em Arial, Helvetica, sans-serif;
position: relative;
padding: 30px 10px 10px 75px;
width: 600px;
}
h1 {
left: -100px;
position: absolute;
top: 160px;
text-align: left;
-webkit-transform: rotate(270deg);
-moz-transform: rotate(270deg);
-o-transform: rotate(270deg);
-ms-transform: rotate(270deg);
transform: rotate(270deg);
}
figure {
background: #fff;
border: 1px solid #BFBFBF;
-webkit-box-shadow: 2px 2px 4px rgba(0,0, 0, 0.3);
-moz-box-shadow: 2px 2px 4px rgba(0,0, 0, 0.3);
box-shadow: 2px 2px 4px rgba(0,0, 0, 0.3);
display: block;
float: right;
margin: 20px 20px 50px 50px;
padding: 5px;
text-align: center;
-webkit-transform: rotate(10deg);
```

```

-moz-transform: rotate(10deg);
-o-transform: rotate(10deg);
-ms-transform: rotate(10deg);
transform: rotate(10deg);
}
figcaption {
clear: both;
display: block;
font-weight: bold;
padding: .5em 0;
}
p {
margin: 0 0 1em;
}

<h1>To jest tytuł strony</h1>
<p>Pellentesque habitant morbi tristique sen.....</p>
<figure>

<figcaption>
Widok z naszego pokoju hotelowego
</figcaption>
</figure>
<p>Pellentesque habitant morbi tristique sen.....</p>

```

Na rysunku 4.8 widać efekt użycia funkcji `rotate` CSS-owej właściwości `transform`, lecz istnieją jeszcze inne, których możesz używać: `translate`, `scale` i `skew`. Listing 4.14 pokazuje przykład kodu potrzebnego do implementacji tych efektów (nie zapomnij dodać prefiksów specyficznych dla przeglądarek). Kod pokazuje efekty transformacji zastosowanych na tekście, ale mogą być one wykorzystane na dowolnym elemencie, włączając w to `body`.

Listing 4.14. Więcej możliwości transformacji CSS

```

h1 {transform: translate(100px, 200px);}
h1 {transform: skew(40deg);}
h1 {transform: scale(2);}
/* Możesz również połączyć je razem w pojedynczej deklaracji: */
h1{transform: translate(100px, 200px) skew(40deg) scale(2) rotate(40deg);}

```

Możesz również stosować przejścia i animacje. Istnieje różnica między przejściami i animacjami w CSS-ie. Przejście jest procesem zmiany między dwoma stanami, wywołanym przesunięciem myszą kursora nad obiekt (`:hover`) albo kliknięciem przyciskiem myszy (`:active`). Animacje CSS działają natomiast samodzielnie.

Przejścia były dostępne od dawna w WebKit, a Mozilla i Opera już je implementują; niestety, obecnie nie robi tego IE. Przejścia wykorzystują pokazaną wcześniej właściwość `transform` i mogą być zastosowane w przypadku pseudoklas `:hover` albo `:active`. Zamiast natychmiastowego zmieniania wartości elementu, gdy przesuwasz nad nim kursor myszą, możesz utworzyć animację przechodzącą do tych nowych wartości przy użyciu właściwości CSS `transition`. Rysunek 4.9 pokazuje prosty przykład zwykłego tekstowego łącza.

To jest łącze

Rysunek 4.9. Łącze tekstowe z efektem `hover`

Rysunek 4.10 pokazuje efekt w trakcie przechodzenia tła łącza do koloru czarnego i tekstu do koloru białego oraz ostateczny efekt przesunięcia kursora nad łącze. Trudno jest oddać ten efekt w drukowanej, czarno-białej książce, więc zachęcamy Cię do wypróbowania tego w przeglądarce.



Rysunek 4.10. Efekty przejścia

Listing 4.15 pokazuje kod potrzebny do uzyskania efektu związanego z przesunięciem kursora myszy. Docelowy kolor animacji jest zdefiniowany w deklaracji `:hover`. Bez żadnego przejścia kolory natychmiast zmieniłyby się na docelowe, ale dla uzyskania efektu stopniowej zmiany koloru używasz właściwości `transition` (oraz prefiksów producentów dla WebKit, Mozilli i Opery). Kluczowy w tym przypadku fragment CSS-a — `transition: all 1s ease-in` — przekazuje przeglądarce, by animowała wszystkie właściwości (`background` i `color`) i użyła funkcji czasowej `ease-in`, oraz informację, że całkowity czas trwania przejścia to jedna sekunda (1s). Możesz opcjonalnie umieścić wartość opóźnienia, tak jak w `transition: all 1s ease-in 1s`, opóźniając rozpoczęcie przejścia oraz moment, w którym element powraca do swojego domyślnego stanu.

Listing 4.15. Animowana zmiana koloru po najechaniu kursorem

```
a {
background: #fff;
border-radius: 5px;
display: block;
float: left;
padding: 5px;
text-align: center;
width: 125px;
-webkit-transition: all 1s ease-in;
-moz-transition: all 1s ease-in;
-o-transition: all 1s ease-in;
transition: all 1s ease-in;
}

a:hover {
background: #000;
color: #fff;
}
```

Za pomocą przejść uzyskujesz zatem drobne, subtelne efekty, ale możesz również używać ich do poprawienia ogólnych wrażeń użytkownika strony. Weź jako przykład galerię miniatur zdjęć z rysunku 4.11. Używając kodu z listingu 4.16, dzięki kombinacji `rotate` i `scale` stworzysz ładny efekt wyświetlenia obrazka w jego pełnym rozmiarze po najechaniu na niego kursorem, i to bez potrzeby użycia JavaScriptu, co pokazano na rysunku 4.12.

Listing 4.16. Animowane powiększenie obrazka po najechaniu kursorem

```
img {
background: #fff;
border: 1px solid #BFBFBF;
display: block;
```

```
float: left;
height: 125px;
margin: 0 10px 0 0;
padding: 5px;
width: 125px;
-webkit-box-shadow: 2px 2px 4px rgba(0,0, 0, 0.3);
-moz-box-shadow: 2px 2px 4px rgba(0,0, 0, 0.3);
box-shadow: 2px 2px 4px rgba(0,0, 0, 0.3);
-webkit-transition: all 1s ease-in-out;
-moz-transition: all 1s ease-in-out;
-o-transition: all 1s ease-in-out;
}
img:hover {
-webkit-transform: rotate(10deg) scale(2);
-moz-transform: rotate(10deg) scale(2);
-o-transform: rotate(10deg) scale(2);
-ms-transform: rotate(10deg) scale(2);
transform: rotate(10deg) scale(2);
}
```



Rysunek 4.11. Galeria zdjęć przed przejściami



Rysunek 4.12. Obrazek z galerii zdjęć po najechaniu na niego kursorem

PRZEPIS DLA ZAAWANSOWANYCH

Tworzenie animacji za pomocą CSS-a

W przeciwieństwie do przejść, animacje nie wymagają aktywacji poprzez najechanie kursorem lub kliknięcie. Obecnie animacje są dostępne tylko w przeglądarkach opartych na WebKit i w Firefoksie, ale wykorzystując różne techniki javascriptowe, możesz uzyskać animacje także w innych przeglądarkach. Tabela 4.5 pokazuje, które wersje każdej z przeglądarek obsługują animację CSS.

Tabela 4.5. Obsługa animacji CSS w przeglądarkach i urządzeniach

Android	2.3+
Chrome	13.0+
Firefox	5.0+
Internet Explorer	-
iOS Safari	4.0+
Opera	-
Safari	5.0+

W pierwszym, prostym przykładzie przesuń obrazek z lewej strony ekranu do prawej i jednocześnie go obróć. Pokazywanie zrzutu ekranu z tego przykładu nie ma większego sensu, więc otwórz przeglądarkę Chrome, Firefox albo Safari i uruchom kod z listingu 4.17.

Listing 4.17. Animacja CSS

```
/* CSS */
div {
float: left;
height: 100%;
position: relative;
width: 100%;
}
img {
position: absolute;
-webkit-animation-name: moveIt;
-webkit-animation-duration: 5s;
-webkit-animation-iteration-count: infinite;
-webkit-animation-timing-function: linear;
-moz-animation-name: moveIt;
-moz-animation-duration: 5s;
-moz-animation-iteration-count: infinite;
-moz-animation-timing-function: linear;
animation-name: moveIt;
animation-duration: 5s;
animation-iteration-count: infinite;
animation-timing-function: linear;
}
@-webkit-keyframes moveIt {
from {
left: 0;
-webkit-transform: rotate(0deg);
}
to {
left: 100%;
-webkit-transform: rotate(360deg);
}
```

```

}
}
@-moz-keyframes moveIt {
from {
left: 0;
-moz-transform: rotate(0deg);
}
to {
left: 100%;
-moz-transform: rotate(360deg);
}
}
keyframes moveIt {
from {
left: 0;
transform: rotate(0deg);
}
to {
left: 100%;
transform: rotate(360deg);
}
}
/* HTML */
<div>

</div>

```

Przy selektorze `img` znajdują się cztery właściwości CSS dotyczące animacji. Są powtórzone, ponieważ muszą się jawnie odwoływać do przedrostków `-webkit-` i `-moz-`. Na końcu umieść nazwy właściwości bez przedrostków z myślą o dniu, w którym wszystkie przeglądarki będą obsługiwać animacje CSS:

- `-webkit/moz-animation-name` — nazwa animacji, której chcesz użyć;
- `-webkit/moz-animation-duration` — jak długo animacja będzie trwać;
- `-webkit/moz-animation-iteration-count` — ile razy animacja będzie powtórzona;
- `-webkit/moz-animation-timing-function` — rodzaj animacji; wybierz spośród `ease`, `linear`, `ease-in`, `ease-out`, `ease-in-out` i `cubic-bezier`, którą można dostosowywać.

Teraz zdefiniuj animację `@-webkit/moz-keyframes MoveIt`. Jest to prosta animacja, więc zacznij od właściwości `from`, a skończ na wartości `to`. Wewnątrz tych właściwości użyj zwykłego CSS-a, by przemieścić oraz obrócić obrazek. W tych animacjach można zmienić prawie każdą właściwość CSS, masz więc dużo różnych możliwości.

Korzystanie z właściwości `from` i `to` jeszcze nie wszystko. Możesz również umieszczać klatki kluczowe, używając procentów, co pozwoli Ci na większą elastyczność. Rysunek 4.13 pokazuje baner, który znajduje się w prawym górnym rogu strony. Dzięki użyciu kodu z listingu 4.18 kolor tła zmienia się płynnie, a klatki kluczowe są umieszczone co 25 procent czasu animacji.

Listing 4.18. *Animowany baner*

```

p {
background: #000;
color: #fff;
font: bold 20px Tahoma, Geneva, sans-serif;
padding: 10px;

```

```
position: absolute;
right: -65px;
text-align: center;
top: 75px;
width: 300px;
-webkit-transform: rotate(45deg);
-moz-transform: rotate(45deg);
-o-transform: rotate(45deg);
-ms-transform: rotate(45deg);
transform: rotate(45deg);
-webkit-animation-name: glow;
-webkit-animation-duration: 5s;
-webkit-animation-iteration-count: infinite;
-webkit-animation-timing-function: ease-in;
-webkit-backface-visibility: hidden;
-moz-animation-name: glow;
-moz-animation-duration: 5s;
-moz-animation-iteration-count: infinite;
-moz-animation-timing-function: ease-in;
-moz-backface-visibility: hidden;
animation-name: glow;
animation-duration: 5s;
animation-iteration-count: infinite;
animation-timing-function: ease-in;
backface-visibility: hidden;
}
```

```
@-webkit-keyframes glow {
  0% {
    background: #F00;
  }
  25% {
    background: #06C;
  }
  50% {
    background: #000;
  }
  75% {
    background: #06C;
  }
  100% {
    background: #F00;
  }
}
```

```
@-moz-keyframes glow {
  0% {
    background: #F00;
  }
  25% {
    background: #06C;
  }
  50% {
    background: #000;
  }
  75% {
    background: #06C;
  }
  100% {
    background: #F00;
  }
}
```

```
}  
}  
  
keyframes glow {  
  0% {  
    background: #F00;  
  }  
  25% {  
    background: #06C;  
  }  
  50% {  
    background: #000;  
  }  
  75% {  
    background: #06C;  
  }  
  100% {  
    background: #F00;  
  }  
}
```



Rysunek 4.13. Animowany kolor tła

Wskazówka

Przejścia i animacje mają wiele zastosowań. Nie tylko pozwalają dodawać fajne efekty do stron internetowych, ale mogą być także w prosty sposób tworzone i poprawiane przez projektantów i programistów. Pokazaliśmy w tym rozdziale jedynie proste przykłady, by pobudzić Twój apetyt, a od Ciebie zależy, jak je wykorzystasz. Dużym możliwościom towarzyszy jednak duża odpowiedzialność, więc staraj się nie przesadzać z różnymi efektami, ponieważ mogą one szybko zamienić fajną witrynę w kiczowatą. Zbyt wiele przejść i animacji może również wpłynąć na wydajność przeglądarki. Sugerujemy spędzić trochę czasu na poszukiwaniu w internecie przykładów i badaniu sposobów, w jakie inni projektanci i programiści użyli CSS3 do ulepszenia witryn.

Podsumowanie

W tym rozdziale poznałeś kilka nowych możliwości CSS-a: zapytania o media, własne czcionki, gradienty, transformacje, przejścia oraz animacje. Istnieje o wiele więcej funkcji niż te, które opisaliśmy; dotknęliśmy jedynie powierzchni. Jeśli jesteś zainteresowany CSS-em, sugerujemy, żebyś kontynuował poszukiwania i naukę, ponieważ istnieje wiele innych technik i sugerowanych praktyk dotyczących użycia niektórych spośród tych nowych możliwości. A zatem zgłębiaj CSS i baw się nim.

Skorowidz

A

- adresy e-mail, 351
- aktualizacja pamięci podręcznej, 312
- Android, 91
- animacja, 103
 - mapy sprite'ów, 150
 - wykresu, 157–162
- animowana zmiana koloru, 101
- animowanie obrazka, 101, 154
- animowany baner, 104
- API dla multimediów, 175, 189
- API HTML5, 220
- arkusz stylów smartphone.css, 86
- artykuł z komentarzami, 71
- atrybut
 - accept, 365
 - address, 232
 - audio, 170
 - autofocus, 123
 - autoplay, 170, 188
 - capture, 365
 - controls, 171, 189
 - draggable, 297, 299
 - dropzone, 298, 301
 - formnovalidate, 127
 - kind, 172
 - list, 120
 - loop, 170, 188
 - manifest, 310
 - multiple, 124
 - novalidate, 127
 - pattern, 126
 - placeholder, 119
 - port, 295
 - poster, 169
 - preload, 170, 188
 - readyState, 280
 - required, 125

- src, 169, 188
- step, 126
- type, 364
- atrybuty
 - elementu audio, 187
 - interfejsu ContactField, 362
 - interfejsu Contacts, 360
 - obiektu adresowego, 232
 - pliku, 332
 - storageEvent, 261
 - wideo, 169
- autouzupełnianie, 120

B

- baza danych, 270
- bezpieczeństwo, 310
 - danych, 248
 - File API, 331
 - historii, 218
- biblioteka Modernizr, 79
- biblioteki, 220
- bloki try-catch, 253
- błąd
 - CONSTRAINT_ERR, 270
 - DATABASE_ERR, 269
 - PERMISSION_DENIED, 231
 - POSITION_UNAVAILABLE, 231
 - QUOTA_ERR, 270
 - SYNTAX_ERR, 270
 - TIMEOUT, 231
 - TIMEOUT_ERR, 270
 - TOO_LARGE_ERR, 269
 - UNKNOWN_ERR, 269
 - VERSION_ERR, 269
- błędy
 - interfejsu FileReader, 343
 - Messaging API, 364

C

- ciasteczka, 247
- Contacts API, 359
- CSS, 74
 - dla ekranów, 89
 - dla smartfonów, 90
- CSS3, 85
- CSS3 Media Queries, 85
- czcionki, 92, 95
- czyszczenie pamięci, 260

D

- dane formularza, 257
- deklaracja
 - @font-face, 92
 - dropzone, 298
- Device APIs Working Group, 357
- DeviceOrientation Event Specification, 368
- dodawanie
 - informacji semantycznych, 66
 - internetowego radia, 194
 - pliku .js, 294
 - tekstu, 147
 - treści zastępczej, 186
 - wątku roboczego, 290
 - wpisów do historii, 204
- DOM, 73, 284
- dostęp do danych, 248
- Drag and Drop API, 297
- DTD, Document Type Definition, 26

E

efekt hover, 100
 efekty, 140

- gradientu, 96
- przejścia, 101

 element

- abbr, 64
- address, 56
- article, 34, 36
- aside, 37
- audio, 185
- b, 63
- canvas, 76, 133, 153
- cite, 58
- datalist, 120
- del, 58
- details, 54
- div, 37
- dl, 61
- doctype, 26
- em, 64
- figcaption, 53
- figure, 51
- footer, 39
- form, 123, 128
- header, 29
- hgroup, 31
- hr, 64
- i, 64
- image, 365
- input, 77
- input typu file, 365
- link, 27
- mark, 57
- meter, 122
- nav, 31
- ol, 59
- output, 118
- progress, 121
- s, 58
- script, 27
- section, 34–37, 62
- small, 63
- source, 166
- strong, 64
- time, 53
- track, 172
- video, 163, 169, 171

 elementy

- listy, 108
- usunięte, 64

F

FIFO, first in, first out, 218
 File API, 331, 349, 355

- Directories and System, 349
- Writer, 349, 355

 film, 163
 filtrowanie plików, 334
 flaga trwałości, 354
 formatowanie

- kwadratu, 141
- napisów, 174

 formaty plików, 93
 formularz, 108

- kontaktowy, 109, 313
- rejestracyjny, 129–132
- z obliczeniami, 313

 funkcja

- animateSprite, 152
- blockHandleDragStart, 307
- calculateDistance, 234
- changeField, 260
- checkStorage, 260
- clearFile, 342
- clearRect, 146
- displayFiles, 337
- dragStartHandler, 301
- drawImage, 149
- fileAction, 351, 354
- geoSuccess, 231
- getGroceryItems, 276
- getResult, 329
- grabTweets, 328
- growBars, 162
- handleCapture, 367
- handleDragOver, 337
- handleFileDrop, 337
- initGraph, 161
- initSettings, 161
- loadPages, 214
- loadStations, 200
- moveGear, 156
- nextState, 211
- onLoadStartHandler, 347
- onOpen, 283
- onStorageEvent, 266
- remove, 355
- reverseGeoCode, 234
- sendNumber, 295
- setLocation, 234
- setPermission, 320
- setStation, 200

showFile, 342
 showGrid, 136, 138
 showImage, 207
 startWatch, 244
 startWorkerThread, 287
 swapCache, 316
 towerHandleDragOver, 308
 wywołania zwrotnego, 229

G

generator czcionek, 94
 geokodowanie, 224
 Geolocation API, 75, 223, 245
 gniazdo sieciowe, 279
 Google Font Directory, 94
 gra Wieże Hanoi, 303, 307
 gradient liniowy, 96
 gradienty, 95, 141
 gradienty z zatrzymaniami, 97
 grupowanie

- nagłówków, 31
- treści, 35

H

historia

- przeglądarki, 203
- sesji, 203
- strony, 213

 History API, 203
 HTML, Hypertext Markup Language, 13
 HTML Media Capture, 365
 HTML5, 15
 HTML5 Boilerplate, 75
 HTML5 Outliner, 41
 HTML5Shiv, 74

I

identyfikator stateInfo, 210
 implementacja historii, 205
 informacje o

- kontakcie, 359
- lokalizacji, 228
- położeniu geograficznym, 223
- przeglądarcze, 84

 interfejs

- BlobBuilder, 349
- DataTransfer, 298, 301

- device, 363
- DirectoryEntry, 349
- Entry, 349
- File, 331
- FileEntry, 349
- FileList, 331, 334
- FileReader, 331, 337
- FileReaderSync, 338
- FileSaver, 349
- FileSystem, 349
- FileWriter, 349
- Notification, 317
- NotificationCenter, 316
- SharedWorker, 291, 296
- Storage, 249
- Worker, 284

J

- JavaScript, 73
- jQuery, 77, 78
- JSON, 288

K

- kadry animacji, 157
- klasa
 - no-webgl, 80
 - no-sessionstorage, 81
 - sessionstorage, 81
- klucz
 - themeColor, 257
 - visits, 252
- kodek
 - MP3, 186
 - Ogg Vorbis, 186
- kodeki wideo, 164
- kodowanie znaków, 26
- kolor
 - motywu, 256
 - tła, 253
- komentarz w pliku, 312
- komentarze, 69
- kompatybilność przeglądarek, 318
- komunikacja
 - dwukierunkowa, 288
 - z serwerami, 279
- komunikat
 - błędu, 125, 200, 351
 - do klientów, 296
 - END, 328
 - TWEET, 328

- komunikaty
 - błędów, 129
 - walidacji, 110
- konfiguracja
 - biblioteki Modernizr, 80
 - elementu canvas, 134
- konspekt dokumentu, 42
- kontakty, 359
- kontrolka wprowadzania daty, 77
- kontrolki
 - domyślne, 182
 - formularzy, 108
 - niestandardowe, 176
 - wideo, 177
- kreje pisma, 94

L

- liczba
 - par klucz-wartość, 258
 - wizyt, 251
- licznik idxCounter, 211
- lista
 - JSON, 200
 - nieuporządkowana, 108
 - uporządkowana, 59
 - zakupów, 272
- listy zagnieżdżone, 60
- logo HTML5, 17
- lokalizacja, 223, 225, 233
- lokalne przechowywanie danych, 257, 262
- lokalny plik, 350

Ł

- ładowanie skryptów, 83
- łącza, 65

M

- manifest aplikacji obliczającej, 314
- mapa sprite'ów, 151
- Messaging API, 363
- metoda
 - abort, 343
 - cancel, 317
 - clear, 260, 267
 - clearWatch, 239
 - createNotification, 317, 320, 322
 - createPattern, 150

- dbPresent, 276
- deleteItem, 277
- drawImage, 149, 151
- executeSql, 269
- fillText, 138, 147
- find, 359
- fireNotification, 320
- getContext, 76, 134
- getCurrentPosition, 224, 227, 234
- getFile, 354
- getGroceryItems, 276
- history.pushState, 205
- importScripts, 329
- initShoppingDb, 276
- key, 258
- Modernizr.load, 83
- onaborthandler, 348
- onErrorHandler, 348
- openDatabase, 268, 276
- postMessage, 284, 288
- preventDefault, 301, 309
- pushState, 204
- readAsDataURL, 338–342
- readAsText, 342
- readTransaction, 276
- removeItem, 253, 260
- replaceState, 211
- requestFileSystem, 351
- requestPermission, 320
- rotate, 154
- scale, 154
- setItem, 250
- show, 317, 322
- showGroceryItems, 276
- strokeText, 147
- terminate, 284
- transaction, 269, 276
- translate, 154
- watchPosition, 239, 244
- webkitRequestFileSystem, 351, 354
- window.addEventListener, 136
- window.openDatabase, 276
- metody
 - DataTransfer, 298
 - File API, 350
 - FileReader, 338
- mikrodane, 66
- mikroformaty, 67
- mikser, 190, 194
- motyw domyślny, 253

N

nadawanie stylu, 141, 253
 nagłówek, 30
 CACHE, 311
 FALLBACK, 311
 NETWORK, 312
 napisy wideo, 172, 173
 narzędzia
 Google Chrome, 252
 płótna, 139
 narzędzie
 @font-face Generator, 93
 HTML5 Outliner, 41
 nawigacja, 31
 nazwy klas, 28
 Network Information API, 364
 notacja z kropką, 276
 notatki, 262
 Notification API, 316, 317

O

obiekt
 ApplicationCache, 315
 Blob, 351
 coords, 227
 danych stanu, 215
 dataTransfer, 302
 FileEntry, 355
 FileReader, 342
 fileWriter, 355
 localStorage, 249, 260
 lokalizacji, 227
 sessionStorage, 249
 storageEvent, 261
 sworker, 295
 timestamp, 227
 WebSocket, 279
 WindowDatabase, 268
 obrazek, 148
 obsługa
 @font-face, 92
 animacji, 103, 104
 atributu
 autofocus, 123
 formnovalidate, 127
 multiple, 124
 pattern, 126
 required, 125
 step, 127
 błędów, 231, 269, 276, 354

elementu

 canvas, 370
 datalist, 120
 output, 119
 progress, 122
 video, 164
 File API, 332
 gradientów, 95
 History API, 204
 Internet Explorera, 73
 iPhone'a, 91
 pamięci podręcznej, 309
 pola wprowadzania liczby, 116
 próbnika kolorów, 118
 przejść, 98
 transformacji, 98
 typu wejściowego
 datetime, 113
 email, 110
 search, 112
 tel, 111
 url, 111
 wątków roboczych, 285
 WebSocket API, 280
 zapytań o media, 86
 zdarzenia onmessage, 287
 zdarzenia popstate, 210
 zdarzeń, 266
 odczyt atrybutów pliku, 332
 odnajdywanie lokalizacji, 225
 odświeżanie pamięci podręcznej, 310
 odtwarzacz
 Flash Player, 186
 JW Playera, 186
 radia internetowego, 194
 odtwarzacze
 dźwięku, 185
 wideo, 183
 odwrócone geokodowanie, 224
 offline, 313
 określanie odległości, 233
 opakowywanie łączny, 65
 opcje
 CSS-a, 81
 lokalizacji, 233
 osadzanie
 dźwięku, 186
 wideo, 167
 oznaczanie
 daty i czasu, 53
 rysunków, 51
 wiadomości, 34

P

pamięć
 lokalna, 258, 261
 podręczna, 309, 312
 sesji, 250, 252
 para klucz-wartość, 248, 266
 parametry
 funkcji find, 359
 metody
 executeSql, 269
 getCurrentPosition, 225
 openDatabase, 268
 transaction, 269
 watchPosition, 239
 parsowanie
 pliku, 348
 pliku CSV, 342
 tweetów, 326
 pasek
 boczny, 37
 postępu, 122
 przewijania, 183
 plik
 CSV, 342, 347
 httpd.conf, 168
 lokalny, 351
 manifestu, 309, 310
 współdzielonego wątku
 roboczego, 294
 pliki
 .appcache, 310
 .eot, 93
 .htm, 27
 .js, 294
 .mov, 165
 .mp3, 185
 .mp4, 166
 .ogv, 166
 .otf, 93
 .svg, 93
 .ttf, 93
 .vtt, 172
 .webm, 166
 .woff, 93
 dźwiękowe, 189
 plótno, 135, 153
 pobieranie
 kontaktów, 360
 numerów telefonów, 361
 stanu z historii, 209
 tweetów, 328
 z historii, 213

podgląd obrazków, 338
 podświetlanie tekstu, 57
 pokazywanie motywu, 256
 polecenia SQL, 268
 polyfilling, 82
 połączenie z gniazdem, 279
 położenie geograficzne, 225, 232, 238
 port klienta, 296
 powiadomienia, 318
 powiadomienia o tweetach, 322
 poziomicca, 369
 poziomy uprawnień, 323, 328
 prezentacja slajdów, 215
 próbnik kolorów, 117
 przechowywanie
 danych, 247, 257
 w pamięci lokalnej, 257, 262
 przechwytywanie
 błędu, 253
 obrazu, 365
 zdarzeń, 261
 zdarzeń pamięci, 266
 przeciąganie
 obiektu, 297, 299, 302
 plików, 334
 przeglądarka obrazków, 207, 209
 przeglądarki, 73
 przejścia, 98
 przekształcenia płótina, 153
 przełącznik, 54
 przenoszenie informacji, 215
 przesyłanie wiadomości do wątku, 285
 przycinanie obrazka, 150
 przycisk Zastęp stan, 212

R

radio internetowe, 194
 rodzaje
 połączeń, 364
 składowania, 248
 role, 68
 rysowanie
 krzywych, 147
 linii, 142
 łuku, 146
 prostokąta, 139
 siatki płótina, 136
 ścieżek, 142
 trójkąta, 143
 wielokąta, 144

S

siatka, 135
 sieciowe bazy danych, 277
 składnia, 27
 skrypt
 Modernizr, 80
 MooTools, 172
 słownik, 61
 powiadomienia o tweetach, 322
 specyfikacja
 Contacts API, 359
 DeviceOrientation, 368
 File API, 349
 HTML5, 15
 Microdata, 66
 WAI-ARIA, 67, 69
 Web SQL Database API, 268
 Web Storage, 249
 strefa upuszczania, 309
 strona
 startowa, 28
 z wiadomościami, 43
 z wynikami wyszukiwania, 46, 49
 zastępcza, 314
 struktura dokumentu, 41
 styl formularza, 128
 szablon, 75

Ś

ścieżki, 144
 śledzenie
 położenia, 239, 240
 przepływu danych, 283
 tweetów, 326

T

tabela
 departments, 271
 groceryitems, 270
 tablica
 connections, 296
 zgodności przeglądarek, 84
 tekst, 147
 tekst zastępczy formularza, 119
 testowanie
 HTML5, 84
 obsługi geolokalizacji, 76
 transakcja, 269
 transformacje, 98

transformacje CSS, 99
 treści nieistotne, 58
 tweety, 322
 tworzenie
 animacji, 103
 bazy danych, 270
 formularza, 109
 formularza rejestracyjnego, 130
 formularza wyszukiwania, 112
 gradientów, 141
 konspektu dokumentu, 42
 kontrolki kalendarza, 113
 listy zakupów, 272
 lokalnego pliku, 350
 mikro danych, 67
 miksera, 190
 nawigacji, 31
 niestandardowych kontrolki, 176
 paska bocznego, 37
 pliku manifestu, 310
 pojemnika, 292
 połączenia, 279, 281
 powiadomień, 318, 323
 poziomiccy, 369, 370
 próbnika kolorów, 117
 przeglądarki obrazków, 207
 przycisków, 95
 słownika, 61
 strony klienta, 288, 292
 strony powiadomień, 322
 strony z wiadomościami, 43
 suwaka, 117
 wątku roboczego, 285
 wideo z napisami, 172
 widżetu przełącznika, 54
 zwijalnego spisu treści, 56
 typ

datetime-local, 113
 MIME, 168, 310, 347
 wejściowy
 date, 114
 datetime, 113
 email, 109
 month, 115
 number, 116
 range, 117
 search, 112
 tel, 110
 time, 114
 url, 111
 week, 115

U

układ

- domyślny ekranu, 88
- strony, 25, 45
- witryny z rolami, 69
- współrzędnych, 135

uprawnienia, 317, 323

upuszczanie

- obiektu, 299
- plików, 337

ustawianie

- elementów w CSS-ie, 74
- motywu, 253

W

WAI-ARIA, 67

walidacja, 107, 126

wątek główny, 323

wątek roboczy, 284

- dedykowany, 284
- współdzielony, 284

Web Notifications API, 316

Web SQL Database API, 267, 271

Web Storage API, 267

Web Workers API, 283

WebSocket API, 279

wiadomości, 35

wideo, 167

- z kontrolkami, 183
- z napisami, 172

widżet kalendarza, 82

wielokąt, 146

wielowypełniacz, 83

witryna Font Squirrel, 94

wizualizacja stanu przeglądarek, 84

własne reguły walidacji, 126

właściwości Modernizr, 79

właściwość

- transform, 100
- transition, 100

wpis do historii, 205, 214

wprowadzanie liczb, 116

współdzielony wątek roboczy, 291

wstawianie obrazka, 148

wtyczka VideoSub, 172

wykres słupkowy, 157, 161

wykrywanie

- CSS3, 79
- możliwości HTML5, 81
- typów, 79

wyłudzanie informacji, 219

wynik

- pomiaru, 122
- wyszukiwania, 46

wyrażenia regularne, 126

wysyłanie

- danych, 281
- komunikatów, 363
- obliczeń, 314
- wiadomości, 279

wyświetlanie

- adresu URL, 218
- atrybutów pliku, 332
- liczby wizyt, 251
- pliku CSV, 344
- powiadomień, 318–321
- tweetów, 322
- wyników, 118

wywołanie REST, 323

Y

YouTube, 167

Z

zapisywanie

- formularzy, 257
- w historii, 213
- w localStorage, 258

zapytania o media, 85–91

zarządzanie powiadomieniami, 320

zastępowanie stanu, 212

zatrzymania, 97

zawartość pamięci, 252

zbiór API, 358

zdarzenia

- addEventListener, 182
- API, 175, 189
- ApplicationCache, 315

Drag and Drop API, 302

durationchange, 201

FileReader, 338

wątków roboczych, 284

WebSocket, 280

zdarzenie

- compassneedscaleibration, 369
- devicemotion, 369
- deviceorientation, 368
- domReady, 209
- dragover, 337
- error, 288
- hashChange, 221
- message, 328
- onchange, 260, 341
- onclick, 191
- onconnect, 295
- ondragstart, 301
- onerror, 284
- onload, 339
- onmessage, 284
- popstate, 209, 214, 218
- pushState, 210
- SharedWorker, 291
- window.addEventListener, 250

zmiana

- historii, 211
- historii strony, 213
- manifestu, 315
- zawartości adresu URL, 218

zmienna dropzone, 337

znacznik

- body, 136
- canvas, 136
- czasu, 285

znaczniki nawigacyjne, 32

znak hash, 205

zwijalny spis treści, 56

Ż

żądanie REST, 329

żyroskop, 370

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>



**POZNAJ
MOŻLIWOŚCI
JĘZYKA HTML5
W PRAKTYCE!**

HTML5. Podręcznik programisty

Najlepsze przepisy na udane aplikacje internetowe!

HTML5 pozwala twórcom stron internetowych tworzyć oparte na standardach sieciowych, działające w przeglądarkach, niezwykle bogate aplikacje o dużych możliwościach. Aplikacje te są zaopatrzone w najróżniejsze funkcjonalności – od przeciągania i upuszczania do natywnego dźwięku i wideo – wszystko to oferują bez żadnych dodatkowych wycieczek. Krótko mówiąc, każdy twórca stron internetowych musi opanować HTML5, a im szybciej to zrobi, tym większą będzie miał przewagę. *HTML5. Podręcznik programisty* zawiera rady ekspertów i sprawdzony kod, potrzebne do tego, by od razu rozpocząć budowę wysokiej jakości aplikacji HTML5.

Autorzy tej książki, Chuck Hudson i Tom Leadbetter, prezentują przetestowane, modułarne przepisy na poziomach dla początkujących, średnio zaawansowanych i zaawansowanych. Dowiesz się konkretnie, jak zapewniać nowoczesne zachowania aplikacji poprzez integrację nowych i ulepszonych elementów HTML5 z CSS3, multimediami i API języka JavaScript. Zgodnie z obecnymi standardami książka skupia się w pierwszym rzędzie na funkcjach HTML5 obsługiwanych w przeglądarkach oraz określa dla nich poziom wsparcia.

Tematyka książki obejmuje:

- zrozumienie i stosowanie nowych strukturalnych elementów HTML5
- wykorzystanie grupowania, poziomu tekstu i zmienionej semantyki
- zarządzanie obsługą HTML5 w przeglądarkach
- używanie nowych technik dotyczących układu i stylizacji w CSS3
- maksymalizację interaktywności za pomocą formularzy internetowych HTML5
- osadzanie przy użyciu HTML5 dźwięku i wideo
- rysowanie na elemencie Canvas
- sterowanie historią przeglądarki
- integrowanie informacji o położeniu za pomocą API Geolocation
- implementowanie przechowywania danych po stronie klienta
- obsługę plików lokalnych
- zarządzanie komunikacją i wątkami
- optymalizację zachowań przeglądarki w HTML5
- integrację danych urządzeń

helion.pl
księgarnia
internetowa

Nr katalogowy: **11710**



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

<http://helion.pl>



ISBN 978-83-246-4386-3



Cena 67,00 zł

Informatyka w najlepszym wydaniu

9 788324 648863