

Laura Lemay
Rafe Colburn
Jennifer Kyrnin

Wydanie VII

HTML, CSS i JavaScript

Dla każdego

SAMS

Helion

Tytuł oryginału: SAMS Teach Yourself HTML, CSS & JavaScript Web Publishing in One Hour a Day, Seventh Edition

Tłumaczenie: Robert Górczyński

ISBN: 978-83-283-2780-1

Authorized translation from the English language edition: HTML, CSS & JAVASCRIPT WEB PUBLISHING IN ONE HOUR A DAY, SAMS TEACH YOURSELF: COVERING HTML5, CSS3, AND JQUERY; Seventh Edition; ISBN 0672336235; by Laura Lemay; and by Rafe Colburn; and Jennifer Kyrnin; published by Pearson Education, Inc, publishing as SAMS Publishing. Copyright © 2016 by 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 © 2017.

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)

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:
<ftp://ftp.helion.pl/przyklady/htcdk7.zip>

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/htcdk7>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

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

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

Spis treści

	O autorach	11
	Wprowadzenie	13
CZĘŚĆ I	ROZPOCZĘCIE PRACY	
Lekcja 1.	Co oznacza publikowanie treści w sieci WWW?	21
	Myślenie w kategoriach twórcy publikującego w sieci WWW	21
	Przeglądarki WWW	27
	Serwer WWW	31
	Adres URL	32
	Szeroka definicja publikacji treści w sieci WWW	33
	Podsumowanie	33
	Warsztaty	34
Lekcja 2.	Poznajemy narzędzia	37
	Anatomia witryny internetowej	37
	Przygotowanie komputera do publikacji treści w sieci WWW	39
	Narzędzia dla programistów wbudowane w przeglądarkę Google Chrome	42
	Co chciałbyś zrobić w sieci?	46
	Szkielet witryny internetowej	47
	Hosting	49
	Podsumowanie	51
	Warsztaty	52
Lekcja 3.	Wprowadzenie do HTML i CSS	55
	Czym jest HTML (i czym nie jest)?	55
	Jak wygląda plik HTML?	58
	Atrybuty HTML	62
	Użycie atrybutu style	63

Krótką historia standardów HTML	64
HTML5, czyli obecny i ewoluujący standard	66
Podsumowanie	66
Warsztaty	67

CZĘŚĆ II TWORZENIE STRON INTERNETOWYCH

Lekcja 4.	Poznajemy podstawy HTML	71
	Struktura kodu HTML	71
	Tytuł	73
	Nagłówki	75
	Akapity	77
	Komentarze	78
	Podsumowanie	80
	Warsztaty	81
Lekcja 5.	Organizacja informacji za pomocą list	83
	Ogólne omówienie list	83
	Lista numerowana	84
	Lista nieuporządkowana	89
	Lista definicji	92
	Listy zagnieżdżone	93
	Inne sposoby użycia list	95
	Podsumowanie	96
	Warsztaty	97
Lekcja 6.	Praca z łączami	101
	Utworzenie łącza	101
	Łączenie stron lokalnych przy użyciu względnych lub bezwzględnych ścieżek dostępu	106
	Łącza do innych dokumentów w sieci WWW	109
	Łącze do określonego położenia w dokumencie	114
	Anatomia adresu URL	120
	Rodzaje adresów URL	123
	Podsumowanie	126
	Warsztaty	127

CZĘŚĆ III JESZCZE WIĘCEJ ZA POMOCĄ HTML I CSS

Lekcja 7.	Formatowanie tekstu za pomocą HTML i CSS	133
	Elementy na poziomie znaku	133
	Formatowanie znaku za pomocą CSS	136
	Wstępnie sformatowany tekst	139
	Linia pozioma (inaczej tematyczna przerwa)	142
	Podział wiersza	144
	Adres	146
	Cytat	147
	Znaki specjalne	149
	Czcionka i jej wielkość	153
	Podsumowanie	163
	Warsztaty	165
Lekcja 8.	Użycie CSS do nadania stylu witrynie	169
	Dołączenie stylów na stronie	169
	Selektory	171
	Jednostki miary	176
	Edycja stylów za pomocą narzędzi dla programistów	180
	Użycie koloru	181
	Łączy	182
	Model pudełkowy	183
	Więcej selektorów	197
	Znacznik <body>	202
	Podsumowanie	202
	Warsztaty	203
Lekcja 9.	Użycie obrazów na stronie internetowej	205
	Obrazy w sieci WWW	205
	Formaty obrazów	206
	Obrazy śródliniowe w HTML — znacznik 	208
	Obrazy i tekst	212
	Obrazy i łączy	221
	Inne użyteczne sztuczki związane z obrazami	225
	Obrazy tła	226
	Użycie obrazów jako wypunktowania	230
	Czym jest mapa obrazu?	231

	Etykieta związana z obrazami	241
	Podsumowanie	242
	Warsztaty	242
Lekcja 10.	Tworzenie tabel	247
	Utworzenie tabeli	247
	Elementy składowe tabeli	248
	Wielkość tabeli, jej obramowanie i komórki	256
	Kolor tabeli i komórki	265
	Wyrównanie treści tabeli	267
	Łączenie wierszy i kolumn	271
	Bardziej zaawansowane usprawnienia tabel	279
	Jak używane są tabele?	285
	Podsumowanie	285
	Warsztaty	287
Lekcja 11.	Użycie CSS do pozycjonowania elementów na stronie	289
	Schematy pozycjonowania	289
	Pozycjonowanie bezwzględne	294
	Pozycjonowanie stałe	301
	Kontrolowanie stosu elementów	303
	Utworzenie rozwijanego menu	306
	Podsumowanie	310
	Warsztaty	310
Lekcja 12.	Projektowanie formularzy	313
	Poznajemy formularz i jego funkcje	313
	Użycie znacznika <form>	319
	Użycie znacznika <label>	322
	Utworzenie kontrolek formularza za pomocą znacznika <input>	323
	Użycie innych kontrolek formularza	339
	Grupowanie kontrolek za pomocą znaczników <fieldset> i <legend>	348
	Wyświetlanie uaktualnień za pomocą <progress> i <meter>	352
	Stosowanie właściwości CSS dla elementów formularza	357
	Planowanie formularza	362
	Podsumowanie	363
	Warsztaty	365

Lekcja 13.	Struktura strony w HTML5	369
	Krótką historia układu strony w HTML	369
	Układ strony w HTML5	371
	Znaczники strukturalne w HTML5	372
	Szkielet strony	377
	Użycie elementów strukturalnych w HTML5	378
	Podsumowanie	379
	Warsztaty	380
Lekcja 14.	Integracja multimediów — wideo i dźwięk	383
	Łatwe osadzanie plików wideo	383
	Samodzielny hosting plików wideo	389
	Osadzanie wideo za pomocą znacznika <video>	393
	Osadzenie treści Flash za pomocą znacznika <object>	397
	Znacznik <embed>	401
	Osadzanie filmów Flash za pomocą SWFObject	402
	Odtwarzacze Flash	404
	Osadzenie treści audio na stronie internetowej	409
	Podsumowanie	411
	Warsztaty	412
Lekcja 15.	Zaawansowany CSS — układ strony w CSS	415
	Przygotowanie układu strony	415
	Rola CSS w projektowaniu WWW	427
	Podsumowanie	432
	Warsztaty	432
Lekcja 16.	Użycie projektu responsywnego	435
	Czym jest responsywny projekt WWW?	435
	Najpierw należy zająć się obsługą urządzeń mobilnych	437
	Planowanie responsywnego projektu WWW	441
	Tworzenie zapytań o media	442
	Poznajemy mechanikę responsywnego projektu WWW	446
	Najlepsze praktyki responsywnego projektu WWW	459
	Podsumowanie	460
	Warsztaty	460

CZĘŚĆ IV UŻYCIE JAVASCRIPTU I JQUERY

Lekcja 17.	Wprowadzenie do języka JavaScript	465
	Dlaczego miałbyś używać języka JavaScript?	465
	Znacznik <script>	467
	JavaScript i oferowane przez Chrome narzędzia dla programistów	468
	Język JavaScript	470
	Środowisko JavaScript	483
	Zdarzenia	484
	Podsumowanie	487
	Warsztaty	487
Lekcja 18.	Użycie biblioteki jQuery	491
	Czym są biblioteki JavaScript?	491
	Rozpoczęcie pracy z jQuery	492
	Pierwszy skrypt jQuery	492
	Wybór elementów w dokumencie	494
	Dołączanie zdarzeń	495
	Modyfikacja stylów na stronie	496
	Modyfikacja treści strony	499
	Efekty specjalne	507
	AJAX i jQuery	510
	Podsumowanie	513
	Warsztaty	514
Lekcja 19.	Użycie JavaScriptu na stronach internetowych	517
	Weryfikacja formularza za pomocą JavaScriptu	517
	Ukrywanie i wyświetlanie treści	526
	Dodanie nowej treści na stronie	535
	Podsumowanie	539
	Warsztaty	540
Lekcja 20.	Praca z ramkami i powiązаныmi oknami	543
	Czym są ramki?	543
	Praca z powiązаныmi oknami	544
	Ramki śródliniowe	550
	Użycie kodu JavaScript do otwierania powiązanych okien	554
	Podsumowanie	557
	Warsztaty	558

CZĘŚĆ V PROJEKTOWANIE DLA KAŻDEGO

Lekcja 21.	Projektowanie dla urządzeń mobilnych	563
	Użytkownicy smartfonów inaczej korzystają z sieci	563
	Zgodność ze standardami i sieć mobilna	564
	Pisanie na potrzeby mobilnej sieci WWW	569
	Projekt i układ strony	572
	Użycie łączy	574
	Użycie obrazów i treści multimedialnej	578
	Maksymalne wykorzystanie CSS i JavaScriptu	581
	Wykorzystanie możliwości funkcji urządzeń mobilnych	583
	Inne dobre nawyki i podpowiedzi dotyczące projektu mobilnej wersji witryny	585
	Podsumowanie	587
	Warsztaty	588
Lekcja 22.	Projektowanie z myślą o użytkowniku	591
	Rozważania dotyczące poziomu doświadczenia użytkownika	591
	Ustalenie preferencji użytkownika	596
	Co oznacza dostępność?	597
	Alternatywne przeglądarki WWW	598
	Tworzenie kodu HTML zapewniającego dostępność	599
	Projektowanie z uwzględnieniem dostępności	603
	Sprawdzenie witryny pod kątem dostępności	605
	Podsumowanie	607
	Warsztaty	607

CZĘŚĆ VI PUBLIKACJA W INTERECIE

Lekcja 23.	Jak opublikować witrynę internetową?	613
	Na czym polega działanie serwera WWW?	613
	Jak znaleźć hosting WWW?	615
	Przygotowanie plików HTML do publikacji	618
	Publikacja plików	620
	Rozwiązywanie problemów	623
	Promowanie stron internetowych	625
	Ustalenie użytkowników Twoich stron internetowych	631
	Podsumowanie	635
	Warsztaty	635

Lekcja 24.	Wykorzystanie funkcji serwera	639
	Jak działa PHP?	639
	Instalacja PHP w komputerze lokalnym	641
	Język PHP	642
	Pętle	649
	Funkcje wbudowane	652
	Funkcje definiowane przez użytkownika	652
	Przetwarzanie formularzy	654
	Dołączanie plików za pomocą PHP	664
	Poszerzenie wiedzy z zakresu PHP	666
	Inne platformy aplikacji	668
	Podsumowanie	669
	Warsztaty	670
Lekcja 25.	Silniki wyszukiwarek i techniki SEO	673
	Co to jest SEO?	673
	W jaki sposób działają silniki wyszukiwarek internetowych?	675
	Techniki SEO	678
	Narzędzia do monitorowania i zarządzania SEO	683
	Płatne łącza	687
	Podsumowanie	687
	Warsztaty	687
	Skorowidz	691

Lekcja 14.

Integracja multimedialnych — wideo i dźwięk

Wideo i dźwięk stanowią ważne elementy nowoczesnej sieci WWW. Obecnie za pośrednictwem internetu możesz oglądać telewizję w witrynach takich jak Hulu, filmy na żądanie za pomocą witryn takich jak Netflix i Amazon, a także klipy wideo przekazane przez innych użytkowników za pomocą serwisów takich jak Vimeo i YouTube. W witrynach oferujących sprzedaż utworów muzycznych można posłuchać ich fragmentów przed zakupem. Z kolei serwisy takie jak Pandora i Last.fm pozwalają użytkownikom na tworzenie własnych stacji radiowych, do tego wystarczy tytuł zaledwie jednej piosenki.

Umiejętność dołączania plików wideo, audio i Flash na stronach internetowych jest bardzo ważna podczas budowania nowoczesnych witryn internetowych. W tej lekcji zostaną poruszone wymienione poniżej zagadnienia.

- ▶ Osadzanie na swoich stronach plików wideo pochodzących z serwisów Vimeo i YouTube.
- ▶ Konwersja plików wideo na najczęściej stosowane formaty sieciowe.
- ▶ Użycie znaczników HTML5 `<video>` i `<audio>`.
- ▶ Użycie znacznika `<object>` do osadzenia plików Flash na stronie internetowej.
- ▶ Użycie odtwarzaczy Flash dla audio i wideo.

Łatwe osadzanie plików wideo

Musisz nauczyć się sporo w zakresie osadzania wideo, poznać: znaczniki przeznaczone do osadzania klipów wideo na stronie, wprowadzone w HTML5 elementy przeznaczone do obsługi audio i wideo oraz niezgodności między przeglądarkami WWW. Jednak przede wszystkim warto zacząć od przeanalizowania sposobu, w jaki większość plików wideo jest obecnie osadzana na stronach internetowych. Użytkownicy przekazują klipy wideo do serwisów specjalizujących się w ich hostingu lub też wykorzystują klipy nakręcone przez inne osoby — kopiują kod z serwisu zawierającego pliki wideo i wklejają go na swoich stronach internetowych, tym samym osadzając klip wideo.

Dwie najpopularniejsze witryny internetowe specjalizujące się w hostingu plików wideo, czyli YouTube i Vimeo, podają dla każdego klipu kod pozwalający na osadzenie danego pliku wideo na własnej stronie internetowej. Na rysunku 14.1 pokazałem formularz pozwalający na wygenerowanie w serwisie YouTube kodu dla osadzonego pliku wideo.



RYСУNEK 14.1. Formularz pozwalający osadzić na własnej stronie internetowej klip z serwisu YouTube

Jeżeli chcesz na stronie dodać plik wideo pochodzący z YouTube'a, po prostu kliknij przycisk *Share*, a następnie wybierz opcję *Embed*, co spowoduje wyświetlenie pokazanego formularza. Jeżeli chcesz klip dostosować do własnych potrzeb, kliknij przycisk *Show More* i wybierz odpowiednie opcje. Następnie skopiuj wygenerowany kod HTML i umieść go na własnej stronie internetowej. Otrzymany efekt pokazałem na rysunku 14.2.



RYСУNEK 14.2. Pochodzący z serwisu YouTube klip osadzony na własnej stronie internetowej

Serwis YouTube (oraz inne tego rodzaju) automatycznie generuje kod znaczników, który można umieścić na własnej stronie internetowej i tym samym osadzić klip wideo. Nie ma w tym żadnej magii. W dalszej części lekcji omówię używane do tego celu znaczniki.

Wady i zalety hostingu plików wideo w zewnętrznych serwisach

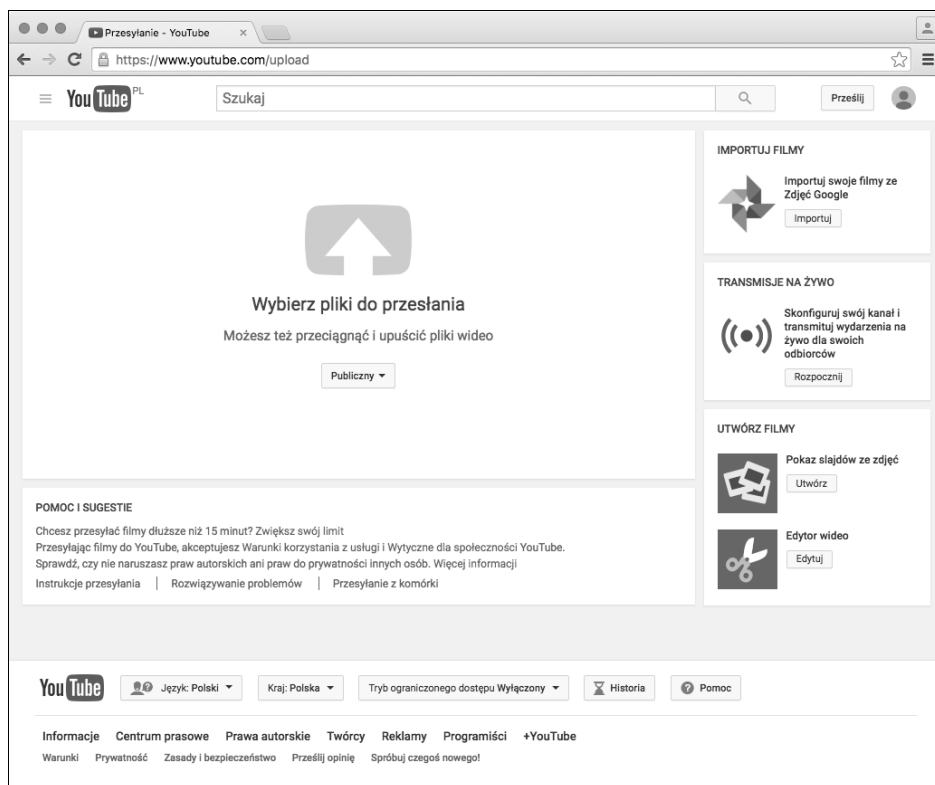
Istnieje możliwość przekazania plików wideo do YouTube'a, a następnie osadzania ich na własnych stronach internetowych. Inne serwisy, takie jak Vimeo (<https://vimeo.com/>), również oferują bezpłatny hosting klipów wideo. Mamy wiele zalet wynikających z hostingu plików wideo w zewnętrznych serwisach zamiast we własnym serwerze WWW. Przede wszystkim pliki wideo mogą być ogromne, więc ich hosting w YouTube lub Vimeo oznacza brak konieczności zastanawiania się nad tym, gdzie je umieścić. Ponadto zyskujesz przygotowany odtwarzacz wideo zapewniający obsługę wielu poziomów jakości materiału wideo oraz jego odtwarzanie na pełnym ekranie. Ten odtwarzacz jest używany przez miliony użytkowników na całym świecie, więc można go uznać za doskonale przetestowany. Ponadto istnieją aplikacje dla platform mobilnych, takich jak Apple iOS i Google Android, co pozwala na wyświetlanie plików wideo z serwisów YouTube i Vimeo. Odtwarzacze w formacie Flash nie działają na platformach mobilnych. Jak się wkrótce przekonasz, kolejną zaletą jest łatwe rozpoczęcie pracy z klipami oferowanymi przez serwisy YouTube i Vimeo. Wystarczy po prostu przekazać plik, przejść na stronę tego klipu, skopiować kod HTML i umieścić ten kod na własnej stronie — tylko to jest wymagane, aby osadzić klip wideo.

Kolejną zaletą hostingu wideo w zewnętrznym serwisie jest możliwość przyciągnięcia także innych użytkowników wymienionych serwisów, a nie jedynie użytkowników własnej strony internetowej. Po przekazaniu klipu wideo do YouTube'a i publicznym udostępnieniu ten plik będzie pojawiał się w wynikach wyszukiwania oraz na listach powiązanych plików wideo, gdy użytkownicy oglądają inne wideo w serwisie. Dlatego też użycie serwisu YouTube lub Vimeo do hostingu plików może prowadzić do przyciągnięcia kolejnych odwiedzających do przygotowanej przez Ciebie witryny internetowej.

Wadą umieszczenia wideo w zewnętrznym serwisie jest mniejsza kontrola nad plikiem wideo oraz sposobem jego prezentacji. Wprawdzie odtwarzacz YouTube'a działa doskonale, ale dla użytkowników strony internetowej będzie oczywiste, że to odtwarzacz YouTube'a — to samo dotyczy Vimeo. Do tego pod uwagę trzeba wziąć także ograniczenia dotyczące materiału, który można umieścić w wymienionych serwisach, na przykład co do czasu trwania klipu. Wspomniane tutaj serwisy mają zautomatyzowane systemy sprawdzające pliki pod kątem naruszenia praw autorskich. Oczywiście tego rodzaju automatyczny system może się pomylić i usunąć plik wideo, który nie narusza niczych praw. Wprawdzie poszanowanie praw autorskich jest bardzo ważne (w serwisie nie wolno umieszczać filmów i muzyki chronionych prawami autorskimi), ale męczące będzie udowadnianie, że masz prawo do pewnej treści, która jest automatycznie usuwana przez serwisy YouTube i Vimeo. Rozważ samodzielny hosting plików wideo, jeśli nie chcesz rozpraszania użytkowników przez YouTube'a, gdy masz potrzebę dostarczania naprawdę ogromnych plików wideo lub jeżeli chcesz wykorzystać własny odtwarzacz treści multimedialnych.

Przekazywanie plików do YouTube'a

YouTube oferuje kilka metod przekazania wideo do serwisu. Jedną z nich jest możliwość nagrywania wideo z kamery sieciowej i bezpośredniego przekazania do serwisu. W ten sposób można wysłać nawet klip nakręcony za pomocą smartfona. W omawianym tutaj przykładzie pokażę, jak przekazać plik, wykorzystując do tego interfejs sieciowy. W celu zainicjowania procesu należy przejść pod adres <https://www.youtube.com/upload>, a następnie przeciągnąć na wskazane miejsce plik wideo (patrz rysunek 14.3).



RYСУNEK 14.3. Strona serwisu YouTube pozwalająca na przekazanie klipu wideo do serwisu

Uwaga

YouTube obsługuje wiele różnych formatów wideo, między innymi używane w większości cyfrowych kamer filmowych. Do obsługiwanych formatów zaliczamy między innymi MP4, MOV, AVI, WMV i FLV.

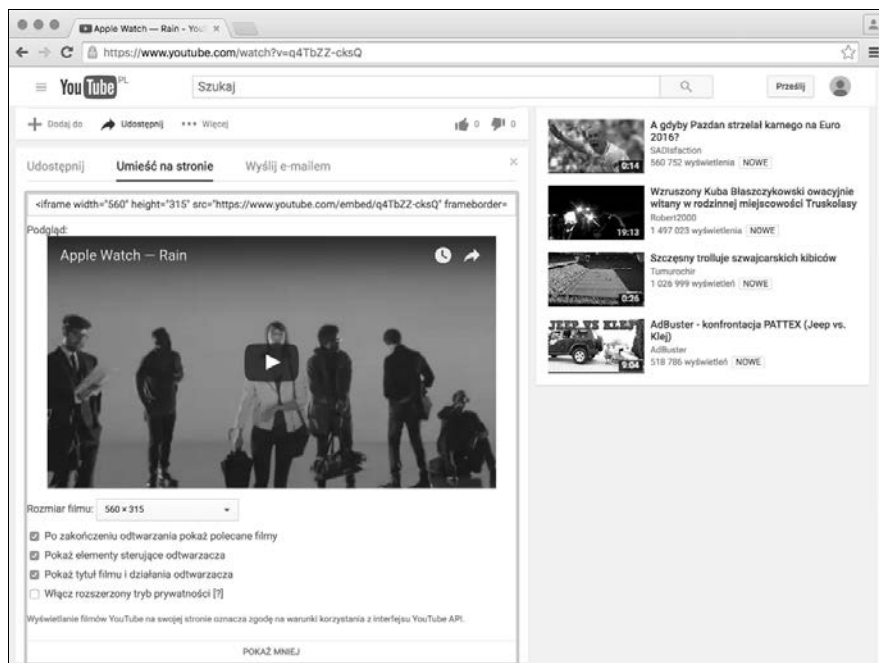
Po wybraniu pliku zostaniesz natychmiast przeniesiony na stronę pozwalającą przetworzyć klip wideo. Zobaczysz wyświetlony pasek postępu informujący, ile czasu potrzeba na przekazanie pliku wideo do serwisu, a także możesz wprowadzić dane dotyczące tego klipu. Za pomocą omawianego formularza podajesz tytuł, opis, kategorię oraz tagi dla wideo. Te wszystkie informacje mają znaczenie, jeśli chcesz, aby inni użytkownicy YouTube'a byli w stanie znaleźć to wideo.

Jak możesz zobaczyć na rysunku, istnieje również możliwość zdefiniowania ustawień prywatności dla przekazywanego klipu. Określenie pliku jako publicznego pozwala innym użytkownikom serwisu na jego znalezienie w YouTube, natomiast klip prywatny będzie dostępny jedynie dla wskazanych osób. Trzecia możliwość to publiczne udostępnienie wideo, ale jedynie dla użytkowników znających adres URL klipu. To jest dobre rozwiązanie, jeśli chcesz osadzić wideo we własnej witrynie internetowej, choć nie chcesz, aby mogło być znalezione przez użytkowników przeglądających zasoby YouTube'a. Opcja związana z harmonogramem pozwala na określenie daty i godziny, kiedy wideo ma zostać udostępnione publicznie w serwerach serwisu YouTube.

YouTube podaje adres URL i kod pozwalający na osadzenie wideo jeszcze przed zakończeniem jego przekazywania, więc łącznie możesz dodać natychmiast.

Dostosowanie odtwarzacza do własnych potrzeb

Po przekazaniu wideo do serwisu YouTube możesz je osadzić na stronach internetowych. Osadzenie własnego klipu niczym nie różni się od osadzania innych plików wideo znalezionych w YouTube — wystarczy wybrać opcję *Embed*, a następnie skopiować kod i wkleić go na stronie. Jednak istnieje kilka możliwości w zakresie dostosowania do własnych potrzeb osadzonego odtwarzacza. Wszystkie dostępne opcje możesz zobaczyć na rysunku 14.4.



RYSUNEK 14.4. Opcje pozwalające na dostosowanie do własnych potrzeb osadzonych klipów z YouTube'a

Po zmodyfikowaniu ustawień strona automatycznie uaktualnia kod osadzający wideo, aby odzwierciedlić wprowadzone zmiany. Dostępne są cztery pola wyboru. Pierwsze powoduje wyłączenie wyświetlania listy powiązanych klipów, która przez serwis YouTube jest standardowo wyświetlana po zakończeniu odtwarzania danego wideo. Możesz skorzystać z możliwości

wyłączenia tej listy, jeśli chcesz, aby użytkownicy pozostali w Twojej witrynie po obejrzeniu wideo, a nie przechodzili do innych klipów w YouTube. Drugie pole powoduje dodanie widocznych kontrolki do odtwarzacza serwisu YouTube (oraz odpowiednie dostosowanie wysokości i szerokości, aby uwzględnić te kontrolki i nie naruszać wymiarów samego wideo). To użyteczna opcja, jeśli użytkownicy mają problemy z klipami, gdy odtwarzacz ma domyślnie ukryte kontrolki. Trzecie pole wyboru pozostawia tytuł wyświetlony na górze ekranu wraz z przyciskami pozwalającymi na obejrzenie klipu później lub podzielenie się informacją o tym wideo. Z kolei ostatnie pole wyboru uniemożliwia serwisowi YouTube przechowywanie informacji identyfikujących użytkownika, jeśli nie kliknął on odtwarzacza.

Po wybraniu odpowiednich opcji dostosowujących odtwarzacz do Twoich potrzeb możesz skopiować kod służący do osadzenia klipu na własnej stronie.

Inne serwisy

YouTube to najpopularniejszy serwis zajmujący się hostingiem plików wideo, choć oczywiście nie jedyny. Vimeo (<https://vimeo.com/>) to inny popularny serwis oferujący hosting klipów wideo, bardzo podobny do YouTube'a. O ile YouTube oferuje możliwość nieograniczonego przekazywania plików, to jednocześnie ogranicza ich długość do 15 minut. Vimeo oferuje konto profesjonalne (płatne) pozwalające subskrybentom na przekazywanie klipów o dowolnej długości. Przykład odtwarzacza Vimeo pokazałem na rysunku 14.5.



RYSUNEK 14.5. Odtwarzacz wideo w serwisie Vimeo

Proces przekazania wideo do serwisu Vimeo odbywa się w praktycznie taki sam sposób jak w YouTube. Wskazujesz plik do przekazania i podajesz informacje, takie jak tytuł i opis. Oba wymienione serwisy przeprowadzą konwersję klipu z praktycznie każdego formatu wideo na używany w stosowanych przez nie odtwarzaczach.

Poniżej przedstawiłem listę innych popularnych serwisów zajmujących się hostingiem plików wideo.

- ▶ Dailymotion — <http://www.dailymotion.com/pl>
- ▶ Flickr — <https://www.flickr.com/>
- ▶ SmugMug — <https://www.smugmug.com/>
- ▶ Viddler — <http://www.viddler.com/>
- ▶ VideoPress — <https://videopress.com/>

Wybór konkretnego serwisu to właściwie kwestia osobistych preferencji. Każdy z nich oferuje własny odtwarzacz wideo oraz społeczność. Wybierz ten, który najbardziej Ci odpowiada. Upewnij się o sprawdzeniu ograniczeń w zakresie długości klipu oraz jego maksymalnej rozdzielczości. Na przykład maksymalna długość klipu w serwisie Flickr to 90 sekund i jedynie użytkownicy kont Pro mogą dodawać wideo w formacie HD. Nie ma żadnej reguły, która zabraniałaby przekazywania plików wideo do więcej niż tylko jednego serwisu. Dlatego też możesz umieścić klip w Vimeo, aby osadzić go na stronie internetowej, i jednocześnie przekazać do YouTube'a, aby udostępnić go użytkownikom tego serwisu.

Samodzielny hosting plików wideo

Z wielu różnych powodów możesz się zdecydować na samodzielny hosting plików wideo, zamiast opierać się na zewnętrznych serwisach, takich jak Vimeo lub YouTube. Jedną z zalet samodzielnego hostingu jest możliwość wykorzystania własnego odtwarzacza zamiast dostarczanego przez serwis. Ponadto możesz nie mieć ochoty na promowanie marki serwisu i stosowanie reklam we własnej witrynie, a także rozpraszanie użytkowników łączami prowadzącymi do YouTube'a. Podobnie jak jest w przypadku większości zewnętrznych usług w sieci WWW, samodzielny hosting daje większą kontrolę nad wynikiem końcowym, choć jednocześnie wymaga większego nakładu pracy i posiadania nieco większej wiedzy.

Wcześniej istniało wiele różnych metod przeznaczonych do osadzania klipów na stronach internetowych, a każda z nich wymagała oddzielnej wtyczki dla przeglądarki WWW oraz formatu pliku. Obecnie najczęściej stosowane są dwie metody. Pierwsza polega na wykorzystaniu technologii Flash do odtwarzania wideo, natomiast druga opiera się na znaczniku HTML5 <video> w celu odtworzenia klipu przez przeglądarkę. Omówię oba stosowane podejścia i pokażę, jak ich połączenie pozwala na zapewnienie obsługi maksymalnej możliwej liczby przeglądarek WWW i platform.

Zanim przejdziemy do poznawania znaczników używanych do publikacji wideo w sieci WWW, warto zacząć od wyjaśnienia sposobu tworzenia plików wideo przeznaczonych do odtwarzania w przeglądarce WWW. Poznanie mechanizmu przygotowania klipów dla sieci to pierwszy krok na drodze przeniesienia wideo z kamery bądź smartfona na stronę internetową.

Formaty wideo i kontenerów

Przed omówieniem sposobu osadzania wideo na stronie internetowej ważne jest poznanie formatów plików wideo. Wszystkie tego rodzaju pliki są kompresowane za pomocą tak zwanego kodeka. Po zakodowaniu wideo konieczne jest umieszczenie go w pliku kontenera. Podobnie jak istnieje wiele różnych kodeków, tak samo mamy wiele różnych kontenerów formatu

pliku. W celu odtworzenia pliku wideo aplikacja musi umieć obsługiwać plik kontenera i potrafić dekodować to wideo. Na przykład H.264 to jeden z najpopularniejszych kodeków wideo, obsługiwany przez wiele kontenerów, między innymi FLV (Flash Video) i MP4.

Dość często zdarza się sytuacja, gdy odtwarzacz wideo potrafi otworzyć kontener zawierający klip, ale jednocześnie nie rozpoznaje kodeka użytego do kompresji pliku wideo. Podobnie jeśli odtwarzacz wideo nie rozpozna pliku kontenera użytego do zapisania wideo, nie będzie w stanie go odtworzyć, niezależnie od rodzaju kodeka. Wprawdzie istnieje naprawdę dużo kodeków wideo i formatów kontenera, ale zaledwie kilka ma znaczenie w przypadku klipów dla sieci WWW. Rozszerzenie pliku wideo wskazuje na format kontenera, a nie kodeka użytego w wideo. Na przykład rozszerzenie dla formatu kontenera Apple QuickTime to *.mov*, niezależnie od kodeka użytego do zakodowania wideo.

H.264 to format komercyjny obsługiwany natywnie przez przeglądarki Microsoft Internet Explorer 9, Apple Safari i Google Chrome. Ponadto jest obsługiwany przez technologię Flash. Problem z H.264 polega na jego opatentowaniu, co oznacza istnienie opłat za wykorzystanie patentu. Firmy implementujące ten kodek muszą płacić za licencję, podobnie jak firmy wykorzystujące ten kodek w celu dostarczania użytkownikom treści zakodowanych w H.264. Z powodu wspomnianych opłat licencyjnych fundacja Mozilla przez długi czas wstrzymywała się z dodaniem obsługi H.264 w przeglądarce Firefox, ale mniej więcej w okolicach wersji 20. zdecydowała się jednak na wprowadzenie obsługi tego kodeka. Jak dotąd jest to najpopularniejszy format przeznaczony do dostarczania treści wideo w sieci WWW. Ponadto jest wykorzystywany przez telewizję satelitarną i kablową, a także do kodowania wideo na dyskach Blu-Ray.

Wideo H.264 jest najczęściej powiązane z kontenerami MP4 (rozszerzenie *.mp4*), czasami z kontenerami Flash Video (*.flv*). Pliki MP4 są obsługiwane przez odtwarzacze Flash oraz wszystkie przeglądarki WWW zapewniające obsługę wideo H.264. Z tego powodu MP4 jest najlepiej obsługiwanym kontenerem przeznaczonym do dystrybucji wideo w sieci WWW.

Theora to otwarty i pozbawiony opłat licencyjnych kodek wideo wydany przez fundację Xiph.org. Przeglądarki Mozilla Firefox i Google Chrome obsługują ten format, natomiast firmy Apple i Microsoft nie mają planów wprowadzenia jego obsługi. Wideo Theora jest najczęściej powiązane z kontenerem Ogg, a pliki są nazywane Ogg Theora. Pliki Ogg zawierające wideo zwykle mają rozszerzenie *.ogv*. Dostępny jest również kodek Vorbis przeznaczony do kodowania audio. Pliki audio Ogg Theora mają rozszerzenie *.oga*.

W roku 2010 firma Google wydała nowy format kontenera o nazwie WebM. Tego rodzaju pliki wykorzystują kodek wideo o nazwie VP8 oraz kodek audio Vorbis. Kodek VP8 został opracowany przez firmę On2 przejętą później przez Google, która następnie udostępniła ten kodek publicznie bez żadnych wymagań licencyjnych. Format WebM jest obsługiwany przez Google Chrome, a będzie obsługiwany także przez przeglądarki Mozilla Firefox, Microsoft Internet Explorer i Adobe Flash.

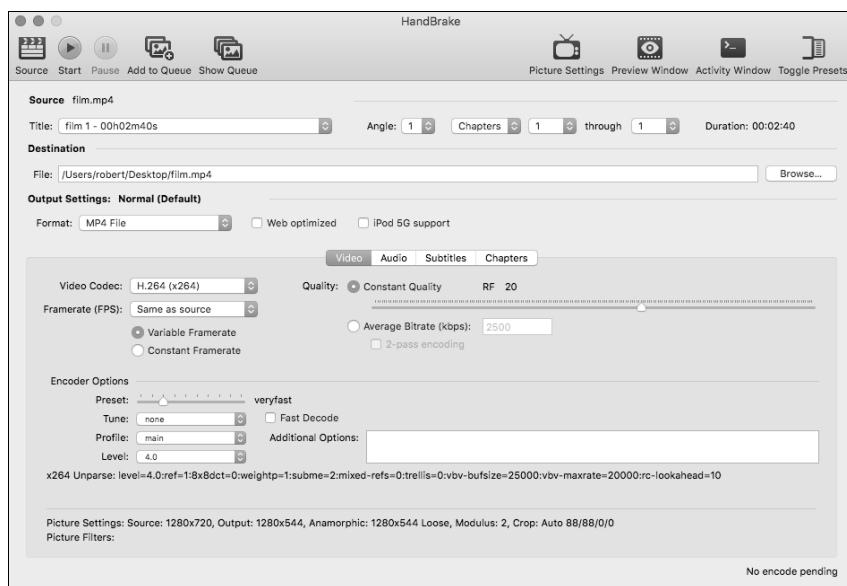
Obecnie, jeśli chcesz tylko jednokrotnie kodować wideo, najlepszym rozwiązaniem jest duet H.264 i MP4, a tego rodzaju pliki będą natywnie odtwarzane przez przeglądarki WWW za pomocą znacznika `<video>`. Wszystkie nowoczesne przeglądarki WWW obsługują ten format. Internet Explorer 8 to jedyna przeglądarka, która tego nie potrafi. Jeżeli musisz zapewnić obsługę Internet Explorera 8, ten sam klip wideo możesz odtworzyć za pomocą technologii Flash.

Konwersja wideo do H.264

Jedną z zalet serwisów zajmujących się hostingiem wideo jest brak konieczności zajmowania się kodekami i formatami kontenerów, ponieważ konwersja jest przeprowadzana automatycznie. Od Ciebie zależy rozdzielczość materiału wideo, natomiast wybór kodeka i kontenera należy już do serwisu. Jeżeli samodzielnie zajmujesz się hostingiem wideo, wówczas musisz je skonwertować na format MP4 i prawdopodobnie Flash.

Do pracy z wideo dostępnych jest wiele narzędzi. W zakresie konwersji różnych formatów na H.264 właściwie interesuje Cię tylko jedno narzędzie — Handbrake. To jest bezpłatna aplikacja typu open source przeznaczona do konwersji wideo w praktycznie dowolnym formacie na H.264. Dostępne są wersje programu dla systemów Windows, OS X i Linux, wszystkie działają w zasadzie tak samo. Aplikację Handbrake możesz pobrać z witryny <https://handbrake.fr/>.

Jeżeli chcesz po prostu skonwertować plik wideo na H.264, otwórz go w programie Handbrake i kliknij przycisk *Start*. Prawdopodobnie należy przeprowadzić pewną modyfikację ustawień, aby zoptymalizować wideo na potrzeby użycia w sieci WWW. Interfejs użytkownika aplikacji Handbrake pokazałem na rysunku 14.6. Przedstawię teraz opcje, które pomagają w optymalizacji wideo umieszczanego w sieci.



RYСУNEK 14.6. Interfejs programu Handbrake

Przed wszystkim w sekcji *Destination* należy podać nazwę dla pliku wyjściowego wideo. Pozostań przy domyślnym formacie danych wyjściowych, czyli MP4. Cztery karty znajdujące się w dolnej części okna pozwalają na optymalizację pliku wideo. Jednak zacznij od zaznaczenia pola wyboru *Web Optimized*. Pozwala to na rozpoczęcie odtwarzania wideo w trakcie jego pobierania, a ponadto odtwarzaczom łatwiej jest poruszać się po takim klipie. Jedynym kosztem jest nieco dłuższy czas kodowania pliku. Jeżeli przewidujesz, że plik będzie odtwarzany w starszej (piątej) generacji urządzeń iPod, powinieneś zaznaczyć też pole wyboru *iPod 5G Support*.

W opcjach na karcie *Video* mamy domyślny kodek H.264. Nie zmieniaj go. W rozwijanym menu *Framerate* możesz pozostać z wartością użytą w pliku wideo bądź wybrać inną. Im większa wartość *Framerate*, tym większy plik wynikowy. Jeżeli zmienisz tę wartość, możesz włączyć kodowanie w dwóch przejściach, co wydłuża czas kodowania (przez dodanie drugiego przejścia), choć jednocześnie wynikiem jest wyższa jakość wideo przy zachowaniu tej samej wielkości pliku.

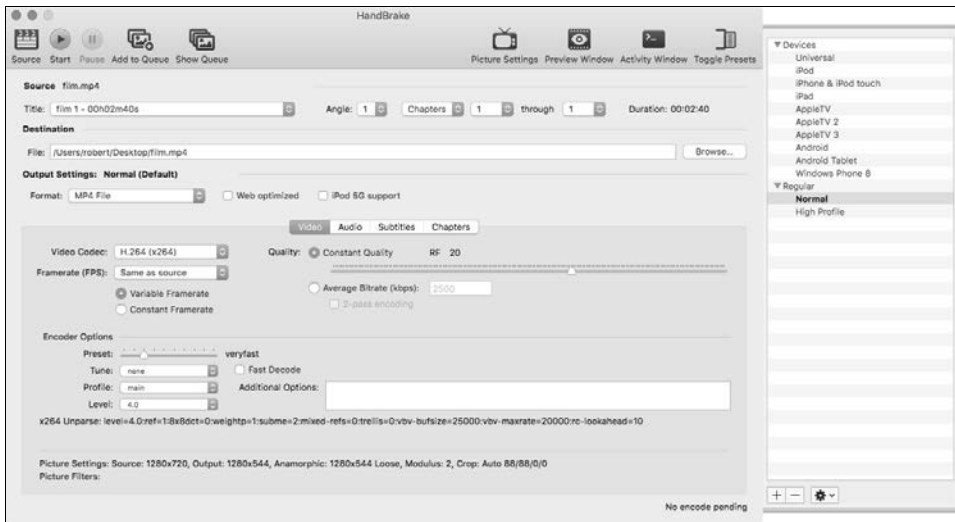
Dochodzimy do ustawień w grupie *Quality*. Kodowanie wideo zawsze wiąże się z pewnymi kompromisami. Im wyższa jakość obrazu, tym większy plik wynikowy. Większy plik wymaga więcej miejsca na serwerze, a jego pobranie będzie trwało dłużej. Z drugiej strony, wideo będzie wyglądało lepiej. Masz do dyspozycji trzy zmienne wpływające na ogólną wielkość pliku: wysokość i szerokość wideo (klip w rozdzielczości 320×240 pikseli będzie znacznie mniejszy niż w rozdzielczości 640×480 pikseli), wartość *Framerate* i jakość. Jeżeli wideo ma być odtwarzane w niewielkim pudełku osadzonym na stronie, można sobie pozwolić na zmniejszenie wymienionych wcześniej wartości i wygenerować mniejszy plik wynikowy. Z kolei jeśli wideo ma być odtwarzane na ekranie o przekątnej 42 cale, wówczas trzeba prawdopodobnie zwiększyć jakość obrazu. Pamiętaj, że strony internetowe są wyświetlane na ekranach o różnych wielkościach, więc unikaj zbyt małego lub zbyt dużego wideo.

Mamy trzy sposoby określania jakości materiału wideo, ich poznanie wymaga nieco wiedzy na temat wartości *bit rate*. W przypadku wideo *bit rate* wskazuje na ilość danych używanych do przedstawienia sekundy klipu. Im większa ta wartość, tym więcej miejsca wymaga dane wideo. Domyślnie jest wybrana opcja o nazwie *Constant Quality*. Oznacza ona, że całe wideo będzie skompresowane z użyciem tego samego współczynnika. Kodek H.264 jest pod pewnymi względami podobny do formatu obrazu JPEG — pewne informacje są tracone w trakcie kompresji wideo. Opcja *Constant Quality* powoduje zastosowanie tego samego współczynnika kompresji podczas kodowania całego wideo. Po wybraniu tej opcji wideo będzie stosować wartość *bit rate* niezbędną do zachowania wybranej jakości. Dlatego też *bit rate* ulega zmianie w zależności od tego, jak można skompresować wideo na podanym poziomie jakości.

Opcja *Average Bit Rate* powoduje zmianę jakości wideo w celu zachowania podanej wartości *bit rate*. Wymienione opcje oraz opcja pozwalająca na podanie wielkości pliku wynikowego są znacznie bardziej przewidywalne — po skonwertowaniu wideo będzie miało oczekiwaną wielkość.

Zamiast samodzielnie modyfikować ustawienia, można wykorzystać jedno z wbudowanych w programie. Jeżeli chcesz wyświetlić dostępne opcje, kliknij przycisk *Toggle Presets* w prawym górnym rogu okna. Zostanie wyświetlona lista opcji (patrz rysunek 14.7). Na rysunku pokazałem w pełni rozwiniętą listę. Te ustawienia są przygotowane dla konkretnych sytuacji. W przypadku wideo na potrzeby sieci WWW najlepiej sprawdza się ustawienie zatytułowane *iPhone & iPod touch*. Jeżeli dopiero zaczynasz pracę z przygotowaniem wideo na stronę internetową, najprostsze podejście będzie polegać na wybraniu wymienionego ustawienia i zaznaczeniu pola wyboru *Web Optimized*.

Możesz również kliknąć przycisk *Picture Settings* widoczny na pasku i podać wielkość skonwertowanego wideo. Rozdzielczość 320×240 pikseli jest standardową wielkością dla małych klipów, natomiast 640×480 to dość powszechnie stosowana wielkość. W przypadku klipów HD należy wybrać wielkość przynajmniej 1280×720 pikseli. W wyświetlonym oknie ustawień obrazu można również przyciąć klip lub zastosować filtry. Więcej informacji dotyczących ustawień filtrów znajdziesz w dokumentacji programu HandBrake.



RYSUNEK 14.7. Lista ustawień wbudowanych w Handbrake

Jedną z możliwości jest przygotowanie wielu kopii pliku oraz wyświetlanie ich za pomocą zapytań o media. W ten sposób można utworzyć responsywne wersje wideo. Więcej informacji na ten temat znajdziesz w lekcji 16., zatytułowanej „Użycie projektu responsywnego”.

Po ustawieniu żądanych opcji wystarczy kliknąć przycisk *Start*, co rozpocznie kodowanie wideo jako H.264. Po zakończeniu kodowania sprawdź klip za pomocą tego samego odtwarzacza, który będziesz stosować na stronie WWW, i upewnij się, że jakość materiału jest wystarczająca. Jeżeli nie, ponownie zakoduj wideo, ale z innymi ustawieniami. Gdy otrzymasz plik większy, niż oczekiwałeś, też będziesz musiał ponownie przeprowadzić kodowanie z innym współczynnikiem kompresji. Po zakończeniu pracy obejrzyj wideo i upewnij się, że wygląda dobrze.

Osadzanie wideo za pomocą znacznika <video>

Metody używane do osadzania wideo na stronach internetowych uległy dużej zmianie na przestrzeni lat. We wczesnych latach istnienia sieci WWW w celu przedstawienia wideo najlepszym rozwiązaniem było podanie łącza do pliku, aby użytkownicy mogli pobrać klip i odtworzyć go w aplikacji innej niż przeglądarka WWW. Kiedy w przeglądarkach pojawiła się obsługa wtyczek za pomocą znacznika <embed>, wówczas stało się możliwe osadzanie wideo bezpośrednio na stronach internetowych. Aby odtworzyć tego rodzaju wideo, użytkownik musiał mieć zainstalowaną odpowiednią wtyczkę.

Znacznik używany do osadzania wtyczek na stronie uległ zmianie z <embed> na <object>, ale podejście pozostało takie samo. Wtyczki pozwalały (i nadal pozwalają) na odtwarzanie osadzonych plików wideo, choć to nie najłatwiejsze zadanie z powodu istnienia wielu formatów wideo i co za tym idzie — wielu dostępnych wtyczek. Rozwiązanie sprawdzające się w przypadku wszystkich lub przynajmniej części użytkowników nadal było problemem.

W roku 2002 firma Adobe dodała do technologii Flash obsługę wideo. Ponieważ niemal każdy miał zainstalowane oprogramowanie odtwarzające pliki Flash, więc osadzenie wideo w tym formacie okazało się najłatwiejszym rozwiązaniem w zakresie osadzania wideo na stronach internetowych. Później możliwe stało się wskazanie odtwarzacza wideo Flash w prawidłowo zakodowanym wideo i jego odtworzenie. Jak zobaczysz w dalszej części lekcji, istnieją odtwarzacze Flash, które można wykorzystać do odtwarzania samodzielnie hostowanych plików wideo. Wraz z nadejściem specyfikacji HTML5 przeglądarki WWW otrzymały natywną obsługę odtwarzania plików wideo za pomocą znacznika `<video>`.

Obecna generacja urządzeń mobilnych, które potrafią odtwarzać pliki wideo (na przykład smartfony iPhone oraz oparte na opracowanym przez Google systemie Android), obsługuje znacznik HTML5 `<video>` oraz nie obsługuje technologii Flash. Dlatego też najlepsze podejście w celu udostępnienia treści jak największej grupie użytkowników polega na użyciu znacznika `<video>` oraz odtwarzacza Flash. Po wprowadzeniu do użycia znacznika `<video>` wyjaśnię, jak można go zastosować wraz z wideo Flash w taki sposób, aby użytkownicy widzieli tylko jeden odtwarzacz odpowiedni dla ich środowiska.

Znacznik `<video>`

Znacznik `<video>` jest nowością w HTML5. Powoduje osadzenie pliku wideo na stronie internetowej oraz wykorzystuje natywne możliwości przeglądarki WWW w zakresie odtwarzania klipów, co jest przeciwieństwem technologii Flash oraz innych wtyczek. Poniżej przedstawiłem najprostszą postać znacznika `<video>`.

```
<video src="mojewideo.mp4">
```

Jeżeli przeglądarka potrafi odtworzyć wideo znajdujące się pod adresem URL wskazywanym przez atrybut `src`, odtworzy ten plik. Ewentualnie może odtworzyć wideo, jeśli będzie inny sposób nakazania tego przeglądarce. W omawianym przykładzie wideo nie zawiera żadnych kontroltek, a jego odtwarzanie nie rozpocznie się automatycznie. Jeżeli chcesz zająć się tego rodzaju szczegółami, powinieneś wykorzystać dostępne atrybuty znacznika `<video>`, które wymieniałem w tabeli 14.1.

TABELA 14.1. Atrybuty znacznika `<video>`

Atrybut	Opis
<code>src</code>	Adres URL odtwarzanego pliku wideo.
<code>height</code>	Wysokość elementu.
<code>width</code>	Szerokość elementu.
<code>controls</code>	Atrybut boolowski wskazujący, czy przeglądarka ma dostarczyć własne kontrolki przeznaczone do obsługi wideo. Wartością domyślną jest brak tego rodzaju kontroltek.
<code>autoplay</code>	Atrybut boolowski wskazujący, czy wideo ma zostać odtworzone natychmiast po wczytaniu strony.
<code>loop</code>	Atrybut boolowski. Jeśli zostanie użyty, wideo będzie odtwarzane w nieskończoność lub do chwili zatrzymania odtwarzania przez użytkownika.

TABELA 14.1. Atrybuty znacznika <video> (ciąg dalszy)

Atrybut	Opis
preload	Atrybut boolowski. Jeśli zostanie użyty, przeglądarka rozpocznie pobieranie wideo w trakcie wczytywania strony, aby było gotowe do odtworzenia. Atrybut ten jest ignorowany w przypadku użycia atrybutu autoplay.
poster	Obraz, który będzie wyświetlany, zanim rozpocznie się odtwarzanie wideo.
muted	Atrybut boolowski. Jeśli zostanie użyty, wideo będzie odtwarzane bez dźwięku.

Ponieważ w przedstawionym powyżej przykładzie nie użyliśmy atrybutu controls lub autoplay, nie ma możliwości rozpoczęcia odtwarzania wskazanego wideo. Na rysunku 14.8 możesz zobaczyć ten klip osadzony na stronie za pomocą znacznika <video> z atrybutem controls.

```
<video src="mojewideo.mp4" controls>
```



RYСУNEK 14.8. Wideo osadzone za pomocą znacznika <video> z atrybutem controls

Ostrzeżenie

Jeżeli zdecydujesz się na osadzenie wideo, upewnij się o zapewnieniu użytkownikom możliwości kontroli jego odtwarzania. Ponadto zachowaj powściągliwość w użyciu atrybutów autoplay i loop. Wielu użytkowników nie chce, aby wideo było odtwarzane natychmiast po przejściu na daną stronę. Jeżeli użyjesz atrybutu loop, ale nie wyświetlisz kontrolki, użytkownik będzie musiał opuścić stronę, aby zatrzymać odtwarzanie wideo.

Domyślnie element <video> ma taką samą wielkość jak rozdzielczość klipu. Do zmiany wielkości elementu na stronie możesz użyć atrybutów height i width, choć przeglądarka WWW zachowa proporcje wideo i pozostawi puste miejsce, gdy zajdzie potrzeba. Na przykład moje wideo ma proporcje 3:2. Dlatego też po utworzeniu elementu <video> o proporcjach 9:5 klip będzie wyśrodkowany w elemencie, jak pokazałem na rysunku 14.9.


```
<video style="background: black;" src="http://www.yo-yo.org/mp4/yu.mp4" controls
width="675" height="375">
```



RYСУNEK 14.9. Znacznik `<video>` tworzący element o innych proporcjach niż osadzone w nim wideo

Dla elementu `<video>` zdefiniowałem czarny kolor tła, aby wyraźnie pokazać, gdzie przeglądarka WWW umieszcza dodatkowe miejsce, gdy proporcje filmu nie odpowiadają proporcjom elementu.

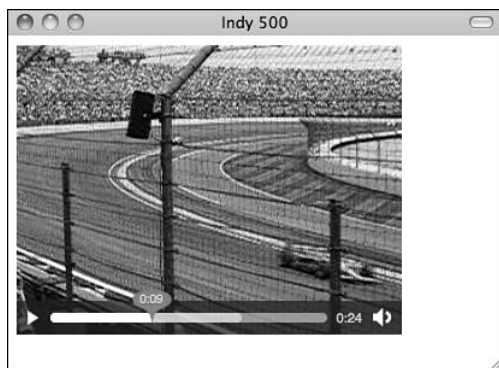
Jeżeli jesteś niemal pewien, że większość osób wchodzących na daną stronę chce obejrzeć wideo, a jednocześnie chcesz im pozwolić na samodzielne rozpoczęcie odtwarzania klipu, skorzystaj z atrybutu `preload`, który nakazuje przeglądarce pobieranie wideo podczas wczytywania strony. Przeglądarka nie rozpocznie automatycznie odtwarzania wideo, ale poczeka na decyzję użytkownika. Zwykle oznacza to, że użytkownik nie będzie musiał długo czekać na rozpoczęcie odtwarzania wideo, gdy spróbuje je włączyć. Wadą takiego podejścia jest pobieranie pliku wideo dla każdego użytkownika, niezależnie od tego, czy faktycznie chce odtworzyć klip, czy nie.

Użycie elementu `<source>`

Mamy jedną poważną wadę związaną z użyciem znacznika `<video>`. Nie wszystkie przeglądarki WWW obsługują te same formaty kontenerów i kodeki. Jak mogłeś zobaczyć, rozwiązanie tego problemu wymaga zakodowania wideo w wielu formatach obsługiwanych przez większość przeglądarek. Na szczęście znacznik `<video>` pozwala na obsługę tego rodzaju sytuacji, a użytkownik nawet nie będzie o tym wiedział.

W celu osadzenia pojedynczego pliku wideo można użyć atrybutu `src` znacznika `<video>`. Natomiast aby dostarczyć wideo w wielu formatach, wykorzystaj element `<source>` zagnieżdżony w znaczniku `<video>`. Poniżej przedstawiłem przykład takiego rozwiązania, wynik możesz zobaczyć na rysunku 14.10.

```
<video width="320" height="240" preload controls>
  <source src="mojewideo.mp4" type='video/mp4; codecs="avc1.42E 01E, mp4a.40.2"'>
  <source src="mojewideo.ogv" type='video/ogg; codecs="theora, vorbis"'>
</video>
```

RYSUNEK 14.10. Wideo osadzone za pomocą znacznika <video> wraz z osadzonymi znacznikami <source> i kontrolkami

Jak możesz zobaczyć, w omawianym przykładzie pominąłem atrybut <src> w znaczniku <video>. Zamiast tego mamy dwa zagnieżdżone elementy <source>. Atrybut src znacznika <source> zawiera adres URL pliku wideo, natomiast atrybut type dostarcza przeglądarce informacje o formacie danego pliku. Następnie przeglądarka analizuje typy poszczególnych plików wideo i wybiera obsługiwany.

Na początku składnia atrybutu type może być nieco niezrozumiała z powodu znaków przestankowych. Poniżej przedstawiłem przykładową wartość.

```
video/ogg; codecs="theora, vorbis"
```

Część pierwsza to typ MIME kontenera wideo. Część druga wymienia kodeki użyte do kodowania ścieżek audio i wideo danego pliku. W omawianym przykładzie typ kontenera to video/ogg, kodek wideo to theora, natomiast kodek audio to vorbis. Jeżeli przeglądarka WWW obsługuje oba typy plików i kodeków, wówczas użyje tego pliku wideo. Poniżej przedstawiłem wartości dla atrybutu type.

- ▶ **MP4/H.264** — video/mp4; codecs="avc1.42E01E, mp4a.40.2"
- ▶ **Ogg Theora** — video/ogg; codecs="theora, vorbis"
- ▶ **WebM** — video/webm; codecs="vp8, vorbis"

Osadzenie treści Flash za pomocą znacznika <object>

Znacznik <object> jest używany do osadzania na stronach internetowych wszelkiego rodzaju plików multimedialnych. Wprawdzie najczęściej jest wykorzystywany do osadzania plików Flash, ale może być używany również w przypadku plików audio, wideo, obrazów oraz innych typów wymagających specjalnych odtwarzaczy. W przeciwieństwie do wszelkich pozostałych znaczników HTML, które dotąd poznałeś, znacznik <object> działa bardzo odmiennie w poszczególnych przeglądarkach. Problem polega na tym, że przeglądarki stosują różne metody do ustalenia, które wtaczki powinny być użyte do wyświetlenia treści wskazywanej przez znacznik <object>.

Wersja znacznika <object> działająca ze starszymi wydaniem przeglądarki Internet Explorer przedstawia się następująco:

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" width="780" height="420">  
  <param name="movie" value="mojewideo.swf" />  
</object>
```

Atrybuty height i width są niezbędne do zdefiniowania ilości miejsca zajmowanego przez <object>. Z kolei atrybut classid identyfikuje kontrolkę ActiveX, która będzie wykorzystana do wyświetlenia treści w przeglądarce WWW. Ten długi, wyglądający na losową kolekcję liter i cyfr ciąg tekstowy jest adresem kontrolki ActiveX w rejestrze Windows. Najlepszym rozwiązaniem jest znalezienie przykładu osadzenia pliku Flash i skopiowania kodu.

Wskazówka

Wskazówka

Kiedy podajesz szerokość i wysokość filmu w formacie Flash, upewnij się o uwzględnieniu także wielkości odtwarzacza. Niektóre odtwarzacze zawierają obramowania, a niemal wszystkie mają również kontrolki pozwalające na sterowanie odtwarzaniem treści. Konieczne jest uwzględnienie tych elementów okna, aby wyświetlane wideo miało oczekiwaną rozdzielczość.

Element <param> jest używany wraz z <object> w celu dostarczenia dodatkowych informacji o osadzonej treści wtyczce, do której odwołuje się kod w znaczniku <object>. Element <param> ma dwa atrybuty: name i value. W omawianym przykładzie element <param> dostarcza odtwarzacz Flash wraz z adresem URL filmu przeznaczonego do odtworzenia.

Przedstawiony wcześniej przykład będzie działał także w przeglądarce Internet Explorer 8, a film Flash został osadzony w pliku o nazwie *mojewideo.swf*. Poniżej przedstawiłem kod znacznika <object> przeznaczony dla pozostałych przeglądarek WWW.

```
<object type="application/x-shockwave-flash" data="mojatresc.swf" width="780"  
  height="420">  
</object>
```

W przypadku przeglądarek WWW innych niż Internet Explorer wtyczka do użycia jest podawana za pomocą atrybutu type, natomiast adres URL filmu do odtworzenia — za pomocą atrybutu data. Jak możesz zobaczyć, atrybuty height i width również zostały użyte. Atrybut type jest stosowany w celu podania rodzaju pliku (typu treści). Przeglądarka „wie”, jak poszczególne typy treści są mapowane na wtyczki, więc potrafi ustalić, czy została zainstalowana prawidłowa wtyczka. Jeżeli tak, można wczytać i wygenerować treść znajdującą się pod adresem URL wskazywanym przez atrybut data. W poniższej ramce dokładnie wyjaśniłem, czym są internetowe typy treści.

Internetowe typy treści

Internetowe typy treści, określane mianem typów treści lub typów MIME, są używane do opisu formatu pliku bądź zasobu. To rodzaj bardziej niezawodnej wersji rozszerzenia pliku. Na przykład obraz w formacie PNG zwykle ma rozszerzenie pliku *.png*. Typ MIME dla pliku PNG to *image/png*. Dokument Microsoft Word ma rozszerzenie pliku *.doc* (w ostatnich wersjach to *.docx*), natomiast jego typ MIME to *application/msword*.

Wymienione typy były pierwotnie wykorzystywane do identyfikacji typów załączników poczty elektronicznej — MIME to skrót od *multipurpose internet mail extensions* — ale obecnie są używane także w innych przypadkach, gdy zachodzi potrzeba wymiany informacji o typach plików.

W przypadku znacznika <object> należy podać internetowy typ treści, aby przeglądarka mogła ustalić najlepszy sposób wygenerowania treści wskazywanej przez ten znacznik. Internetowy typ dla treści w formacie Flash to `application/x-shockwave-flash`, więc po podaniu tego typu przeglądarka powinna użyć wtyczki Flash.

Mamy jeszcze jeden ważny przypadek użycia omawianych typów w zakresie plików audio i wideo. Kiedy serwer WWW wysyła zasób do sieci, dołącza typ `content`. Przeglądarka następnie sprawdza ten typ w celu ustalenia, co powinna zrobić z danym zasobem. Jeśli na przykład typ zasobu został określony jako `text/html`, wówczas przeglądarka traktuje go jak stronę internetową.

Gdy serwer WWW wysyła plik do użytkownika, internetowy typ treści ustala na podstawie rozszerzenia pliku. Dlatego też jeżeli użytkownik zażąda pliku `index.html`, serwer ustali, że rozszerzenie `.html` wskazuje na plik, w którym typem treści powinno być `text/html`. W dalszej części lekcji dowiesz się, jak mieć pewność, że serwer WWW definiuje odpowiedni typ treści dla używanych przez Ciebie plików audio i wideo.

W przypadku większości znaczników można po prostu połączyć wszystkie atrybuty i przygotować znacznik <object>, który będzie działał we wszystkich popularnych przeglądarkach WWW. W przypadku znacznika <object> nie działa to w ten sposób, jednak można znaleźć rozwiązanie problemu. Poniżej przedstawiłem wersję, która będzie działała.

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" width="780" height="420">
  <param name="movie" value="mojewideo.swf" />
  <object type="application/x-shockwave-flash" data="mojatresc.swf" width="780"
    height="420">
  </object>
</object>
```

W powyższym przykładzie mamy do czynienia z zagnieżdżonymi znacznikami <object>. Takie rozwiązanie się sprawdza, ponieważ przeglądarki ignorują nieznane im znaczniki. Dlatego też przeglądarki inne niż Internet Explorer zignorują zewnętrzny znacznik <object>. Z kolei Internet Explorer zignoruje wewnętrzny znacznik <object>, choć jednocześnie przetworzy <param>. To jest najprostsze rozwiązanie w zakresie użycia znacznika <object> w sposób zapewniający jego działanie we wszystkich przeglądarkach. Jeszcze lepsze podejście polega na wykorzystaniu znacznika <object> jako rozwiązania awaryjnego dla starszych przeglądarek (takich jak Internet Explorer 8), które nie rozpoznają znacznika <video>.

Znacznik <object> obsługuje wiele atrybutów, jak możesz zobaczyć w tabeli 14.2.

TABELA 14.2. Atrybuty znacznika <object>

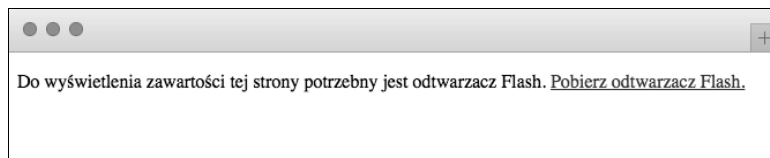
Atrybut	Opis
data	Adres URL dla danych prezentowanych w elemencie <object>. Natomiast Flash używa do tego celu elementu <param>.
form	Umożliwia powiązanie elementu z odpowiednim formularzem.
height	Wysokość elementu.
name	Nazwa elementu.
type	Typ MIME treści wyświetlanej przez obiekt.
usemap	Adres URL dostępnej po stronie klienta mapy obrazu, która będzie zastosowana w obiekcie.
width	Szerokość elementu.

W standardzie HTML będziesz dla plików wideo używać znacznika <video> zamiast <object>, natomiast znacznik <object> nadal będzie stosowany dla plików Flash oraz innej treści multimedialnej, takiej jak Microsoft Silverlight.

Alternatywna treść dla znacznika <object>

Co się stanie w sytuacji, gdy użytkownik nie ma zainstalowanej wtyczki wymaganej przez treść w znaczniku <object>? W takim przypadku przeglądarka WWW wyświetli komunikat błędu lub po prostu nie wygeneruje żadnej treści dla danego znacznika. Jednak istnieje możliwość dostarczenia alternatywnej treści przeznaczonej do wyświetlenia, gdy przeglądarka nie znajduje odpowiedniej wtyczki. Twoje zadanie sprowadza się jedynie do podania alternatywnej treści wewnątrz znacznika <object>. Jeżeli znacznik <object> zadziała, treść alternatywna będzie zignorowana. Natomiast jeśli znacznik <object> nie zadziała, wyświetlona będzie treść alternatywna. Poniżej przedstawiłem zagnieżdżony znacznik <object> wraz z pewną treścią alternatywną, która zostanie wyświetlona przez przeglądarkę niezawierającą zainstalowanej wtyczki Flash (patrz rysunek 14.11). Poniżej przedstawiłem kod źródłowy znacznika.

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" width="780" height="420">
  <param name="movie" value="mojewideo.swf" />
  <object type="application/x-shockwave-flash" data="mojatresc.swf" width="780"
    height="420">
    <p>Do wyświetlenia zawartości tej strony potrzebny jest odtwarzacz Flash.
    <a href="http://get.adobe.com/flashplayer/">Pobierz odtwarzacz Flash.</a></p>
  </object>
</object>
```



RYSUNEK 14.11. Treść alternatywna wyświetlona przez przeglądarkę nieobsługującą treści w formacie Flash

Wskazówka

Często dobrym rozwiązaniem jest zdefiniowanie dla treści alternatywnej takiej samej wielkości jak w przypadku znacznika <object>, aby zachować układ strony. Tej treści alternatywnej można nadać styl za pomocą CSS lub obraz o takiej samej wielkości wykorzystać jako miejsce zarezerwowane dla <object>.

Pamiętaj, że lepszym rozwiązaniem jest dołączenie treści alternatywnej będącej odpowiednikiem prezentowanej za pomocą elementu <object> niż po prostu zasugerowanie pobrania odtwarzacza Flash lub innego programu do wyświetlenia danej treści. Naprawdę rzadko się zdarza, aby użytkownik zdecydował się na pobranie i instalację nowego programu z powodu komunikatu w witrynie internetowej informującego o jej nieprawidłowym działaniu na skutek braku tego programu. Poniżej przedstawiłem znacznie lepszą wersję treści alternatywnej.

<p>Ta strona działa najlepiej z odtwarzaczem

Flash, ale jeśli nie możesz użyć technologii Flash, zawsze możesz pobrać to wideo w formacieMOV i odtworzyć ten plik w komputerze lokalnym.</p>

Wyświetlenie użytkownikowi informacji, że używa nieprawidłowej przeglądarki WWW, oprogramowania itd., jest nieakceptowalne i po prostu irytujące.

Znacznik <embed>

Znacznik <embed> został dodany do specyfikacji HTML5 przede wszystkim ze względu na fakt, że jest używany od chwili jego utworzenia przez Netscape po dodaniu obsługi wtyczek w wymienionej przeglądarce WWW. Przeglądarki nadal obsługują ten znacznik, ponieważ wciąż jest stosowany na wielu stronach internetowych.

Zaczynamy od przeglądu wymaganych atrybutów znacznika <embed>.

<embed src="a01607av.avi" height="120" width="160" type="application/x-shockwave-flash">

Atrybut src zawiera informacje o położeniu pliku multimedialnego, który ma być osadzony na stronie internetowej. Atrybut type wskazuje typ treści. (To jest ten sam atrybut typu jak w przypadku znacznika <object>). Atrybuty height i width podają wyrażone w pikselach wymiary osadzonego pliku.

W tabeli 14.3 znajdziesz wymienione atrybuty znacznika <embed> stosowane w HTML5.

TABELA 14.3. Atrybuty znacznika <embed>

Atrybut	Opis
height	Wysokość elementu.
src	Adres URL pliku multimedialnego.
type	Typ MIME pliku multimedialnego wskazanego przez atrybut src.
width	Szerokość elementu.

Mamy jeszcze kilka innych atrybutów obsługiwanych jedynie przez przeglądarkę Internet Explorer. Jeżeli musisz zapewnić obsługę starszych wersji przeglądarki Internet Explorer, wówczas możesz wykorzystać niezgodne z HTML5 atrybuty wymienione w tabeli 14.4, aby poprawić obsługę osadzonych plików wideo we wspomnianych wydaniach przeglądarki.

TABELA 14.4. Atrybuty znacznika `<embed>` obsługiwane tylko przez przeglądarkę Internet Explorer

Atrybut	Opis
<code>allowfullscreen</code>	Określa, czy osadzony element może zajmować cały ekran. Dozwolone wartości tego atrybutu to <code>true</code> i <code>false</code> .
<code>allowscriptaccess</code>	Określa, czy osadzony obiekt może się komunikować z zewnętrznymi skryptami albo zawierać łącza do zewnętrznych stron. Dozwolone wartości tego atrybutu to <code>always</code> , <code>samedomain</code> i <code>never</code> .
<code>flashvars</code>	Atrybut służący do przekazania parametrów konfiguracyjnych do odtwarzacza plików Flash. Używany jedynie w przypadku, gdy osadzonym elementem jest Flash.
<code>plug-inpage</code>	Adres URL strony, na której można pobrać wtyczkę przeznaczoną do wyświetlenia zawartości tego obiektu.

Ogólnie rzecz biorąc, nie powinieneś używać znacznika `<embed>`. Zdecydowałem się na jego omówienie tutaj, ponieważ nadal jest spotykany na różnych stronach internetowych. Jednak pamiętaj, że istnieją lepsze sposoby osadzania treści multimedialnych na stronach internetowych.

Osadzanie filmów Flash za pomocą SWFObject

SWFObject to połączenie kodu znaczników i JavaScriptu w sposób pozwalający na osadzenie filmów w formacie Flash na stronach internetowych za pomocą zgodnego ze standardami kodu znaczników wciąż obsługującego wszystkie aktualnie używane przeglądarki WWW. JavaScript to język programowania działający w kontekście strony internetowej. Jego działanie szczegółowo omówię w lekcji 17., zatytułowanej „Wprowadzenie do języka JavaScript”. Nie musisz umieć programować w języku JavaScript, aby móc wykorzystywać SWFObject. Wystarczy po prostu skopiować i wkleić pewien kod, a następnie podać kilka szczegółów. W celu pobrania SWFObject i zapoznania się z dokumentacją przejdź na stronę <https://github.com/swfobject/swfobject>.

Pomijając zapewnienie niezawodnego sposobu na przedstawienie filmu w formacie Flash za pomocą zgodnego ze standardami sieciowymi kodu znaczników, SWFObject działa również jako rozwiązanie problemu, który trudno pokonać, używając jedynie kodu znaczników. Jeśli zainstalowana przez użytkownika wersja Flash jest zbyt przestarzała i nie może odtworzyć treści, wówczas film nie będzie wyświetlony, podobnie jak zdefiniowana treść alternatywna. Przeglądarka WWW wykorzysta odtwarzacz Flash i przyjmie założenie o jego prawidłowym działaniu, nawet jeśli tak nie jest. SWFObject wykorzystuje język JavaScript do wychwylenia tego rodzaju błędów oraz wyświetlenia wówczas zdefiniowanej treści alternatywnej.

SWFObject oferuje dwa podejścia w zakresie osadzania treści. Pierwsze wykorzystuje kod znaczników nieco zmodyfikowany za pomocą JavaScriptu (to jest statyczna metoda publikacji),

natomiast drugie opiera się wyłącznie na kodzie JavaScript (to jest dynamiczna metoda publikacji). Wykorzystanie kodu znaczników charakteryzuje się lepszą wydajnością i oferuje pewien poziom funkcjonalności, gdy obsługa JavaScriptu została wyłączona w przeglądarce lub treść jest ponownie publikowana w środowisku pozbawionym obsługi JavaScriptu. Wersja dynamiczna jest nieco bardziej elastyczna, ponieważ pozwala na konfigurację „w locie” osadzonego odtwarzacza.

Użycie SWFObject z kodem znaczników wymaga trzech kroków: dodania znaczników <object>, dołączenia pliku *swfobject.js* na stronie oraz rejestracji odtwarzacza w bibliotece SWFObject.

Przed wszystkim należy zacząć od dodania pliku JavaScript w sekcji <head> dokumentu.

```
<script src="swfobject.js"></script>
```

W ten sposób nastąpi wczytanie na stronie zewnętrznego kodu JavaScript. Trzeba się upewnić, że atrybut *src* prowadzi do odpowiedniego połączenia zawierającego plik *swfobject.js*. Więcej informacji na temat języka JavaScript oraz jego zewnętrznych skryptów znajdziesz w lekcji 17.

Teraz można już osadzić odtwarzacz Flash. Poniżej przedstawiłem odpowiedni kod.

```
<object id="myId" classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
width="780" height="420">
  <param name="movie" value="mojewideo.swf" />
  <!--[if !IE]>-->
    <object type="application/x-shockwave-flash" data="mojewideo.swf" width="780"
      height="420">
<!--<![endif]>-->
<p>Treść alternatywna</p>
  <!--[if !IE]>-->
    </object>
  <!--<![endif]>-->
</object>
```

Trzy kolejne wiersze to pewien kod JavaScript osadzony na stronie.

```
<script>
  swfobject.registerObject("myId", "9.0.115", "mojewideo.swf");
</script>
```

Tekst pochylony przedstawia miejsce zarezerwowane dla wartości, które należy podać w zarejestrowanym SWFObject. Jak możesz zobaczyć, SWFObject wymaga identyfikatora znacznika <object> (tutaj *myId*), informacji o wersji Flasha wymaganej przez film (9.0.115) oraz adresu URL treści przeznaczonej do odtworzenia (*mymovie.swf*).

Inną możliwością jest wykorzystanie publikacji dynamicznej. Przed wszystkim w znaczniku <head> dokumentu umieść znacznik wczytujący kod JavaScript, podobnie jak wcześniej.

```
<script src="swfobject.js"></script>
```

Teraz można już osadzić plik SWF i zdefiniować treść alternatywną.

```
<script>
  swfobject.embedSWF("mojatresc.swf", "myContent", "300", "120", "9.0.115");
</script>
<div id="myContent">
<p>Treść alternatywna</p>
</div>
```

Jak możesz zobaczyć, największa różnica między przedstawionymi podejściami polega na tym, że w tym drugim zupełnie pozbyliśmy się znacznika `<object>`. Zamiast tego mamy znacznik `<div>` działający w charakterze kontenera dla filmu w formacie Flash. Treść alternatywna będzie wyświetlona wtedy, gdy odtwarzacz Flash nie jest dostępny lub też nie spełnia wymagań w zakresie wersji. Wywołanie JavaScriptu w celu dynamicznego publikowania filmów Flash jest nieco inne niż stosowane w statycznej metodzie publikacji.

```
swfobject.embedSWF(adres URL filmu, ID docelowego elementu <div>, szerokość, wysokość, wymagana wersja Flash);
```

Wiele filmów Flash pozwala na przekazanie im konfiguracji za pomocą parametru o nazwie `FlashVars`. Do tego celu można wykorzystać znacznik `<param>`.

```
<param name="FlashVars" value="controls=on">
```

Dostępne zmienne konfiguracyjne zależą całkowicie od używanego filmu Flash. Istnieje możliwość konfiguracji filmu całkowicie z zastosowaniem podejścia publikacji dynamicznej, choć to wymaga nieco większej wiedzy z zakresu JavaScriptu. Więcej informacji na ten temat znajdziesz w dokumentacji `SWFObject` po lekcjach poświęconych językowi JavaScript.

W kolejnej sekcji zajmiemy się pewnymi odtwarzaczami Flash, które mogą być osadzone na stronie za pomocą `SWFObject`.

Odtwarzacze Flash

Dowiedziałeś się już, jak osadzać wideo na stronie za pomocą znacznika `<video>` oraz jak osadzać treść w formacie Flash. Teraz przechodzimy do odtwarzaczy plików Flash, za pomocą których można odtwarzać także pewne klipy przygotowane do użycia w znaczniku `<video>`. Omówione tutaj odtwarzacze są użyteczne, ponieważ pozwalają na wyświetlanie wideo każdemu użytkownikowi, który ma zainstalowaną wtyczkę Flash. Wprawdzie dostępnych jest wiele tego rodzaju odtwarzaczy, ale zajmiemy się tylko dwoma z nich: `JW Playerem` i `Flowplayerem`.

JW Player

`JW Player` to popularny odtwarzacz treści w formacie Flash, zarówno wideo, jak i audio. Jest dostępny na licencji Creative Commons do użytku niekomercyjnego, więc możesz korzystać z niego bezpłatnie w celach niezarobkowych. Kiedy używasz tego odtwarzacza, konieczne jest podanie jego twórcy. Innymi słowy, musisz umieścić łącze prowadzące do witryny `JW Playera`, jeśli korzystasz z osadzonego odtwarzacza. Istnieje również komercyjna wersja `JW Playera`, której można używać w dowolnych celach bez konieczności dodawania łącza prowadzącego do witryny internetowej `JW Playera`. Jeżeli na swojej witrynie internetowej zamieszczasz reklamy, musisz korzystać z komercyjnej wersji `JW Playera`.

W celu pobrania `JW Playera` przejdź na stronę <https://github.com/jwplayer/jwplayer>.

Po pobraniu odtwarzacza będziesz potrzebować pliku wideo przeznaczonego do udostępnienia, jak i samego odtwarzacza w postaci pliku *player.swf*. `JW Player` zawiera przykładowy plik używający znaczników `<object>` i `<embed>`, choć powinieneś skorzystać z podejścia zgodnego ze standardami. Istnieje możliwość użycia `SWFObject` do osadzenia `JW Playera`, jak pokazałem na rysunku 14.12, stosując do tego celu poniższy fragment kodu.


```

<script src="swfobject.js"></script>
<script>
    swfobject.registerObject("myId", "9", "expressInstall.swf");
</script>
<div>
    <object id="myId" classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
        width="400" height="315">
        <param name="movie" value="player.swf" />
        <param name="flashvars" value="file=mojewideo.mp4" />
        <!--[if !IE]>-->
            <object type="application/x-shockwave-flash" data="player.swf"
                width="400" height="315">
                <param name="flashvars" value="file=mojewideo.mp4" />
            <!--

```



RYSUNEK 14.12. Video odtwarzane za pomocą JW Playera

Kod zaczyna się od znacznika `<script>` użytego w celu dołączenia skryptu `SWFObject` oraz zarejestrowania odtwarzacza. Jeżeli spojrzysz na wiersz rejestrujący odtwarzacz, zobaczysz, że zarejestrowałem znacznik `<object>` z identyfikatorem `myId` oraz podałem konieczność użycia odtwarzacza Flash w wersji co najmniej 9, ponieważ to pierwsza wersja obsługująca MP4 i H.264. Na końcu mamy odwołanie do `expressInstall.swf`, czyli pliku Flasha dołączanego do `SWFObject` i pozwalającego użytkownikowi na uaktualnienie przestarzałej wersji odtwarzacza Flash.

Znacznik `<object>` — zarówno dla przeglądarki Internet Explorer, jak i zagnieżdżony dla pozostałych przeglądarek WWW — odwołuje się do pliku JW Playera o nazwie `player.swf`.

W obu przypadkach używany jest parametr `flashvars` do wskazania pliku wideo *mojewideo.mp4*. Dodałem także pewną treść alternatywną prowadzącą użytkownika na stronę pozwalającą pobrać odtwarzacz Flash, jeśli jeszcze nie ma on zainstalowanej tej wtyczki.

Odtwarzacz JW Player charakteryzuje się dużymi możliwościami w zakresie dostosowania go do własnych potrzeb. Szczegółowe omówienie parametrów konfiguracyjnych znajdziesz w dokumentacji dostępnej w witrynie JW Playera. Wszystkie opcje konfiguracyjne są podane w parametrze `flashvars`. Na rysunku 14.13 przedstawiłem przykład, w którym pasek kontrolny jest wyświetlany na górze.

```
<param name="flashvars" value="file=mojewideo.flv&controlbar=top">
```



RYСУNEK 14.13. JW Player z paskiem kontrolnym przeniesionym na górę

Mamy dwa parametry konfiguracyjne w znaczniku `<param>`: `file` i `controlbar`. Każdy z nich jest oddzielony od wartości za pomocą znaku równości, natomiast poszczególne parametry rozdziela zakodowany ampersand (`&`). Zmienna `flashvars` jest formatowana w dokładnie taki sam sposób, jak ciąg tekstowy zapytania w adresie URL, stosuje ten sam sposób jak podczas kodowania parametrów przekazywanych do serwera. Więcej informacji na temat formatowania ciągu tekstowego zapytania znajdziesz w artykule Wikipedii na stronie https://en.wikipedia.org/wiki/Query_string.

Użycie Flowplayera

Flowplayer to inny popularny odtwarzacz wideo oparty na technologii Flash. Wersja podstawowa jest bezpłatna, dostępna jako open source i może być używana w komercyjnych witrynach internetowych. Jednak na koniec filmu wyświetlane jest logo Flowplayera, a maksymalna długość filmu wynosi 4 minuty. Jeżeli chcesz pozbyć się wspomnianego logo lub wyświetlić własne, musisz zakupić komercyjną wersję Flowplayera. Cena zależy od liczby domen, na których ma być używany odtwarzacz. Sam odtwarzacz możesz pobrać z witryny <https://flowplayer.org/>.

Flowplayer, podobnie jak przy publikacji dynamicznej, jest używany za pomocą `SWFObject`. W celu osadzenia na stronie wideo z zastosowaniem Flowplayera konieczne jest dołączenie własnego pliku JavaScript z odtwarzaczem za pomocą znacznika `<script>`.

```
<script src="path/to/the/flowplayer-3.2.2.min.js"></script>
```

Następnym krokiem jest dodanie do strony kontenera, w którym zostanie wyświetlone wideo.

```
<a href="mojewideo.mp4"
  style="display: block; width: 425px; height: 300px;"
  id="player">Pobierz plik wideo</a>
```

Na końcu można zainstalować odtwarzacz w elemencie docelowym.

```
<script>
flowplayer("player", "path/to/the/flowplayer-3.2.2.swf");
</script>
```

Zamiast używać znacznika `<object>` lub `<div>` jako kontenera dla odtwarzacza, Flowplayer zachęca do wykorzystania znacznika `<a>`. W takim przypadku zostanie odtworzony plik wideo wskazany w atrybucie `href` znacznika `<a>`. Poniżej przedstawiłem pełny kod przykładowej strony, efekt osiągany w przeglądarce możesz zobaczyć na rysunku 14.14.

```
<!DOCTYPE html>
<html>
<head>
  <script src="flowplayer-3.2.2.min.js"></script>
</head>
<body>
  <a href="http://elh13.simplecdn.net/flowplayer/flowplayer.flv"
    style="display:block; width:520px; height:330px" id="player"></a>
  <script>
    flowplayer("player", "flowplayer-3.2.2.swf");
  </script>
</body>
</html>
```



RYSUNEK 14.14. Wideo odtwarzane za pomocą Flowplayera

Dostępnych jest wiele opcji pozwalających na dostosowanie odtwarzacza Flowplayer do własnych potrzeb. Najłatwiejszym sposobem ich zmiany jest użycie aplikacji Setup, którą można pobrać ze strony <https://flowplayer.org/designer/>. Istnieje również możliwość samodzielnej konfiguracji Flowplayera za pomocą języka JavaScript. Pełną listę opcji konfiguracyjnych odtwarzacza Flowplayer znajdziesz na stronie <http://flash.flowplayer.org/documentation/configuration/>.

Użycie znacznika <object> ze znacznikiem <video>

Znacznik <object> może być użyty jako alternatywa dla przedstawiania wideo w przeglądarkach, które nie obsługują znacznika <video>, takich jak Internet Explorer 8. Twoje zadanie sprowadza się jedynie do umieszczenia prawidłowego znacznika <object> wewnątrz <video>. W ten sposób obsługa oferowana przez HTML będzie stosowana w przeglądarkach WWW, które rozpoznają znacznik <video>. Ponieważ przeglądarki ignorują nieznanie im znaczniki, więc Internet Explorer pominie znacznik <video>. Z kolei przeglądarki obsługujące <video> zignorują osadzony w nim znacznik <object>, uznając, że został dostarczony jako treść alternatywna dla osadzonego wideo.

Dlatego też po zebraniu wszystkiego w całość otrzymujemy kod znaczników podobny do przedstawionego poniżej.

```
<video width="320" height="240" controls>
  <source src="ścieżka/do/mojewideo.mp4"
    type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"'>
  <source src="ścieżka/do/mojewideo.ogv"
    type='video/ogg; codecs="theora, vorbis"'>
  <object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
    width="320" height="240">
    <param name="movie" value="/mojewideo.mp4">
  </object>
</video>
```

Znacznik <video> wraz z jego elementami <source> wyświetli wideo w przeglądarkach obsługujących znacznik <video>. Natomiast dla pozostałych jest przeznaczony element <object>, który wyświetli wideo z użyciem technologii Flash.

Można jeszcze bardziej zagnieźdzać znaczniki i dołączyć elementy <object> używane zarówno przez Internet Explorera, jak i inne przeglądarki WWW. Wymienione elementy będą potomnymi znacznika <video>.

```
<video width="320" height="240" controls>
  <source src="ścieżka/do/mojewideo.mp4"
    type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"'>
  <source src="ścieżka/do/mojewideo.ogv"
    type='video/ogg; codecs="theora, vorbis"'>
  <object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
    width="320" height="240">
    <param name="movie" value="/mojewideo.mp4">
    <object type="application/x-shockwave-flash" data="mojatresc.swf"
      width="780" height="420">
      <p>Do wyświetlenia zawartości strony potrzebny jest odtwarzacz Flash.
      <a href="http://get.adobe.com/flashplayer/">Pobierz odtwarzacz Flash.</a></p>
    </object>
  </object>
</video>
```

W każdym przypadku przeglądarka natywnie obsługująca znacznik <video> po prostu go użyje. Z kolei Internet Explorer wykorzysta zagnieźdżony znacznik <object>. Natomiast wszystkie pozostałe przeglądarki użyją wewnętrznego znacznika <object>.

Osadzenie treści audio na stronie internetowej

Zaletą osadzania treści audio na stronie internetowej jest to, że proces przypomina osadzanie wideo. Specyfikacja HTML5 dostarcza znacznik `<audio>`, który działa niemal identycznie jak znacznik `<video>`. Omówiony wcześniej znacznik `<embed>` może być używany także z audio, choć zamiast z niego należy korzystać ze znaczników `<audio>` i `<object>`.

Cztery najważniejsze formaty plików i kodeków są używane dla treści audio w sieci WWW: MP3, Ogg Vorbis, AAC i WAV. Format MP3 jest natywnie obsługiwany przez przeglądarki Internet Explorer 9+, Firefox, Opera, Safari i Chrome oraz w systemach iOS i Android, a ponadto może być odtwarzany przez rozwiązania oparte na technologii Flash. Format WAV jest obsługiwany przez wszystkie przeglądarki poza Internet Explorerem. Ogg Vorbis, czyli format otwarty, jest obsługiwany przez przeglądarki Firefox, Opera i Chrome oraz w systemie Android. Z kolei AAC to format używany przez program iTunes podczas konwersji płyt CD na postać plików audio. Jest obsługiwany natywnie przez wszystkie przeglądarki poza jedną — Firefox, a także można go odtwarzać za pomocą rozwiązań opartych na technologii Flash. Firefox obsługuje go jedynie w kontenerze MP4.

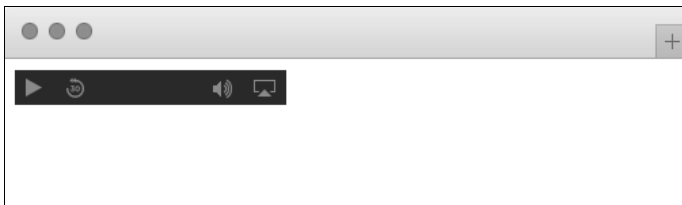
Najlepsze rozwiązanie pozwalające na dotarcie do jak największej grupy odbiorców polega na użyciu znacznika `<audio>` wraz z plikami MP3 dla obsługujących takie podejście przeglądarek, w tym także przeglądarek na platformach mobilnych, które obsługują standard HTML5, ale nie technologię Flash. Z kolei odtwarzacz oparty na technologii Flash można wykorzystać do odtwarzania plików MP3 w przeglądarkach nieobsługujących specyfikacji HTML5 lub plików w formacie MP3.

Znacznik `<audio>`

Znacznik `<audio>` jest podobny do znacznika `<video>`. Próbuje wykorzystać natywne możliwości przeglądarki WWW w celu odtworzenia treści audio. Jego atrybuty są takie same jak w przypadku znacznika `<video>`, z wyjątkiem nieużywanych atrybutów `height` i `width`. Poniżej przedstawiłem przykład użycia omawianego znacznika.

```
<audio src="piosenka.mp3" controls>
```

Jeżeli przeglądarka potrafi odtworzyć treść audio znajdującą się pod adresem URL podanym w atrybucie `src`, wówczas zrobi to oraz wyświetli kontrolki pozwalające na sterowanie odtwarzaniem treści. Wygląd odtwarzacza w przeglądarce pokazałem na rysunku 14.15.



RYSUNEK 14.15. Odtwarzacz osadzonej treści audio

Atrybuty stosowane w znaczniku `<audio>` wymieniałem w tabeli 14.5.

TABELA 14.5. Atrybuty znacznika `<audio>`

Atrybut	Opis
<code>src</code>	Adres URL pliku audio przeznaczonego do odtworzenia.
<code>controls</code>	Atrybut boolowski wskazujący, czy przeglądarka ma dostarczyć własne kontrolki przeznaczone do obsługi pliku audio. Wartością domyślną jest brak tego rodzaju kontrolek.
<code>autoplay</code>	Atrybut boolowski wskazujący, czy plik audio ma zostać odtworzony natychmiast po wczytaniu strony.
<code>loop</code>	Atrybut boolowski. Jeśli zostanie użyty, plik audio będzie odtwarzany w nieskończoność lub do chwili zatrzymania odtwarzania przez użytkownika.
<code>mediagroup</code>	Grupuje elementy multimedialne, wyraźnie wykorzystując kontroler multimediów.
<code>muted</code>	Atrybut boolowski. Jeśli zostanie użyty, wówczas plik audio będzie wyciszony.
<code>preload</code>	Atrybut boolowski. Jeśli zostanie użyty, przeglądarka rozpocznie pobieranie pliku audio w trakcie wczytywania strony, aby był gotowy do odtworzenia. Atrybut ten jest ignorowany w przypadku użycia atrybutu <code>autoplay</code> .

Jeżeli na stronie internetowej chcesz zapewnić odtwarzanie muzyki w tle, do znacznika możesz dodać atrybuty `autoplay` i `loop`. Istnieje spore prawdopodobieństwo, że jeśli użyjesz `autoplay` lub jeszcze gorzej `autoplay` i `loop`, a nie dostarczysz kontrolek dla treści audio, użytkownicy będą w pośpiechu uciekać z takiej strony.

Odtwarzacze audio oparte na technologii Flash

Osadzanie opartych na technologii Flash odtwarzaczy audio przypomina osadzanie opartych na technologii Flash odtwarzaczy wideo. Możesz wykorzystać znacznik `<object>`, osadzone znaczniki `<object>` i `SWFObject` w celu osadzenia treści Flash. Ponadto można zagnieźdźać znaczniki `<object>` w `<audio>` w celu zapewnienia obsługi w jak największej liczbie przeglądarek WWW.

Odtwarzacze JW Player i Flowplayer obsługują także treść audio. W celu jej odtworzenia należy podać ścieżkę dostępu do pliku MP3 zamiast do pliku wideo. Istnieje również pewna liczba odtwarzaczy Flash przeznaczonych jedynie dla treści audio.

Jednym z popularniejszych opartych na Flashu odtwarzaczy treści audio jest WordPress Audio Player. Samodzielnie działającą wersję możesz pobrać z witryny <http://wpaudioplayer.com/>. Dostępna jest również wtyczka dla WordPress, upewnij się, że jej nie pobierasz.

Po pobraniu odtwarzacza pliki *audio-player.js* i *player.swf* umieść w witrynie internetowej. Następnie pozostało już tylko skonfigurowanie odtwarzacza audio, jak pokazałem w poniższym przykładzie strony. Efekt otrzymany w przeglądarce WWW możesz zobaczyć na rysunku 14.16.

```
<!DOCTYPE html>
<html>
  <head>
```

```
<title>Odtwarzacz audio</title>
<script src="audio-player.js"></script>
<script>
  AudioPlayer.setup("player.swf", {
    width: 325
  });
</script>
</head>
<body>
  <p id="myaudio"><a href="song.mp3">Pobierz plik MP3</a></p>
  <script>
    AudioPlayer.embed("myaudio", {soundFile: "piosenka.mp3"} );
  </script>
</body>
</html>
```



RYSUNEK 14.16. Odtwarzacz osadzonej treści audio

Przedstawiony powyżej odtwarzacz audio używa kodu JavaScript do osadzenia odtwarzacza w elemencie docelowym, wykorzystując w tym celu technikę podobną do publikacji dynamicznej znanej z SWFObject. W nagłówku strony należy umieścić plik JavaScript dostarczający odtwarzacz Audio Player, a następnie skonfigurować ten odtwarzacz w kolejnym znaczniku `<script>`. Skopiuj przedstawiony kod i wartość właściwości `width` zastąp odpowiadającą treści audio przeznaczonej do odtworzenia. (W omawianym przykładzie odtwarzacz ma szerokość 325 pikseli).

Teraz w treści strony umieszczamy znacznik `<p>`, który będzie zawierał odtwarzacz. Treść alternatywna znajdzie się w tym znaczniku. Znacznik zostanie zastąpiony właściwym odtwarzaczem przez dodany wcześniej kod JavaScript. W omawianym przykładzie dodałem łącze do pliku MP3, który zostanie odtworzony. Na końcu osadzamy odtwarzacz za pomocą kodu JavaScript. Aby rozwiązanie działało, podałem identyfikator elementu, który będzie zawierał odtwarzacz, oraz wskazałem położenie pliku audio. Tutaj to plik o nazwie *piosenka.mp3*, znajdujący się w katalogu razem ze stroną internetową.

Podsumowanie

W tej lekcji dowiedziałeś się sporo o znacznikach, kodekach i formatach plików przeznaczonych do publikowania treści audio i wideo w sieci WWW. Najpierw zobaczyłeś, jak przebiega proces przekazania pliku wideo do YouTube'a oraz jego publikacja w tym serwisie. Później

poznałeś alternatywne serwisy, takie jak Vimeo, które również oferują hosting wideo. Następnie przeszliśmy do formatów plików powiązanych z wideo w sieci WWW oraz kwestii konwersji treści na zapisaną w omówionych formatach. Dowiedziałeś się dużo o ograniczeniach poszczególnych przeglądarek w zakresie obsługiwanych znaczników i formatów, a także nauczyłeś się pokonywać te ograniczenia w celu dostarczania treści audio i wideo jak największej grupie odbiorców.

Poruszyliśmy także temat samodzielnego hostingu plików wideo. Zobaczyłeś, jak osadzić wideo na stronach internetowych za pomocą znaczników `<video>` i `<object>`, a także jak je połączyć i tym samym zapewnić obsługę treści przez wiele przeglądarek. Dowiedziałeś się o istnieniu narzędzia SWFObject ułatwiającego osadzanie treści w technologii Flash na stronie w sposób zgodny ze standardami sieciowymi. W tej lekcji przedstawiłem także dwa odtwarzacze przeznaczone do obsługi treści audio i wideo na stronach internetowych. Omówiłem także osadzanie treści audio za pomocą znacznika `<audio>`.

W tabeli 14.6 przedstawiłem podsumowanie znaczników wprowadzonych w tej lekcji.

TABELA 14.6. Znaczniki przeznaczone do obsługi osadzonych treści audio i wideo

Znacznik	Opis
<code><audio></code>	Osadza pliki audio na stronach internetowych, aby mogły być obsługiwane natywnie przez przeglądarkę WWW.
<code><embed></code>	Osadza obiekty na stronie internetowej.
<code><object>...</object></code>	Osadza obiekty na stronie internetowej.
<code><param>...</param></code>	Określa parametry przekazywane do osadzonych obiektów. Używany w przypadku znacznika <code><object></code> .
<code><source></code>	Wskazuje plik audio lub wideo, który ma być odtworzony przy użyciu znacznika <code><audio></code> lub <code><video></code> .
<code><video></code>	Osadza pliki wideo na stronach internetowych, aby mogły być obsługiwane natywnie przez przeglądarkę WWW.

Warsztaty

W przedstawionych tutaj warsztatach znajdziesz pytania, które mogą pojawić się w kwestii osadzania audio i wideo na stronach internetowych. Ponadto przedstawiłem pytania sprawdzające Twoją wiedzę oraz trzy ćwiczenia pozwalające na utrwalenie materiału omówionego w lekcji.

Pytania i odpowiedzi

P: Jaki jest najszybszy sposób na rozpoczęcie dodawania wideo na stronie internetowej?

O: Najszybsze rozwiązanie polega na użyciu serwisu takiego jak YouTube lub Vimeo, który niezwykle ułatwia przekazywanie plików wideo do serwisu, a następnie osadzanie ich za pomocą podanego kodu. Dla większości osób wykorzystanie tego rodzaju witryny to naprawdę wszystko, co jest potrzebne, aby dostarczyć wideo użytkownikom. Dodatkowa praca wymagana

do samodzielnego hostingu plików wideo prawdopodobnie jest niepotrzebna w przypadku większości aplikacji, tym bardziej gdy można wykupić usługę subskrypcji na przykład w Vimeo i publikować stamtąd wideo bez konieczności tworzenia łączy zwrotnych.

P: Czy podczas obsługi treści audio i wideo powinienem się przejmować możliwościami przeglądarek w tym zakresie oraz standardami sieciowymi?

O: Niestety tak. W przypadku treści audio i wideo można bardzo łatwo przygotować kod znaczników niezgodny ze standardami sieciowymi lub też wyeliminować pewną grupę użytkowników przez zastosowanie znaczników nieobsługiwanych przez ich przeglądarki WWW. Na szczęście jeśli skorzystasz z technik przedstawionych w tej lekcji, będziesz potrafił osadzać na stronach internetowych treść audio lub wideo w sposób zgodny ze standardami sieciowymi i zapewniający obsługę wszystkich aktualnie używanych przeglądarek WWW.

P: Jaka jest różnica między H.264 i Ogg Theora?

O: H.264 i Ogg Theora to kodeki wideo. Nieco się różnią w kategoriach wydajności, ale podstawowa różnica jest związana z kwestiami licencji. Ogg Theora to technologia otwarta, która bez żadnych przeszkód może być implementowana przez każdego. W celu użycia znacznika `<video>` i dotarcia do jak największej grupy odbiorców pliki wideo powinny być udostępniane w obu wymienionych formatach.

Quiz

1. Jak można dostarczyć treść użytkownikom, których przeglądarki WWW nie obsługują znacznika `<video>` oraz nie mają zainstalowanej wtyczki Flash?
2. Dlaczego SWFObject to bardziej niezawodne podejście podczas osadzania treści Flash niż po prostu użycie znacznika `<object>`?
3. Dlaczego potrzebne są dwa znaczniki `<object>` do osadzenia na stronie treści Flash w sposób zapewniający jej dostępność w większości przeglądarek WWW?
4. Który format wideo jest obsługiwany przez wszystkie przeglądarki rozpoznające znacznik `<video>`?

Odpowiedzi

1. W przypadku użytkowników, którzy nie mogą obejrzeć wideo ze względu na niedostateczną obsługę ze strony przeglądarki WWW lub z powodu braku odpowiedniej wtyczki, najlepszym rozwiązaniem jest wykorzystanie treści alternatywnej. Treść umieszczona wewnątrz znacznika `<video>` lub `<object>` będzie zignorowana przez przeglądarki obsługujące te znaczniki i wyświetlona przez pozostałe przeglądarki. Możesz dodać łącze do odpowiedniej wtyczki, innej przeglądarki WWW lub nawet bezpośrednie łącze do pliku wideo, aby użytkownik mógł go pobrać i odtworzyć za pomocą innej aplikacji w komputerze.
2. Dwie największe zalety SWFObject to możliwość utworzenia prawidłowego kodu znaczników nadal obsługiwanego przez wiele przeglądarek WWW oraz eleganckie zachowanie, gdy użytkownik nie ma zainstalowanej wtyczki lub dostępna wersja odtwarzacza Flash jest zbyt przestarzała i nie może wyświetlić danej treści.

3. Dwa znaczniki `<object>` są wymagane, ponieważ jeden zestaw atrybutów działa w przeglądarce Internet Explorer, natomiast dla pozostałych przeglądarek, takich jak Firefox i Safari, jest przeznaczony inny zestaw atrybutów.
4. To jest trudne pytanie. Nie istnieje jeden format kontenera lub kodeka, który byłby obsługiwany przez wszystkie przeglądarki WWW i każdą z ich wersji. Jeżeli chcesz dotrzeć do jak największej grupy odbiorców, powinieneś zakodować wideo w wielu formatach oraz wykorzystać technologię Flash dla przeglądarek nieobsługujących natywnie znacznika `<video>`.

Ćwiczenia

1. Przekaż plik wideo do serwisu YouTube lub Vimeo, a następnie utwórz stronę internetową i osadź na niej to wideo.
2. Wykorzystaj jeden z dwóch odtwarzaczy omówionych w lekcji (JW Player i Flowplayer) do osadzenia wideo na stronie internetowej. Spójrz na opcje konfiguracyjne i spróbuj zmienić wygląd odtwarzacza.
3. Spróbuj zmodyfikować kod osadzonego wideo z serwisu YouTube na postać zgodną ze standardami sieciowymi. Naprawdę nie ma żadnego powodu, abyś używał kodu niezgodnego ze standardami YouTube'a do osadzania na swoich stronach filmów pochodzących z tego serwisu.

Skorowidz

A

adres, 146
adres URL, 24, 32, 120, 592
 anonimowy FTP, 123
 budowa, 120
 HTTP, 123
 nieanonimowy FTP, 124
 typ file, 125
 typ mailto, 125
 znaki specjalne, 121
AJAX, 510
akapity, 77
analiza
 ikony, 45
 składniowa, 57
 strony głównej, 45
anatomia
 adresu URL, 120
 witryny internetowej, 37
anonimowy FTP, 123
API, application programming interface, 583
aplikacje
 CRUD, 660
 do zarządzania treścią, 50
Apple Safari, 30
argument, 471
arkusz stylów, 175
artykuły, 376
ASCII, 150
atak typu XSS, 655
atrybut, 59, 62
 action, 315, 320
 border, 287
 cellpadding, 261
 class, 63, 173
 classid, 398
 clear, 145
 colspan, 287
 height, 398
 high, 355

href, 102
id, 63, 114, 173
kind, 603
low, 355
method, 315, 321
min, 355
name, 115
optimum, 355
rel, 122
rowspan, 287
sandbox, 553, 558
span, 287
src, 468
srcdoc, 553
style, 63
target, 544, 545
usemap, 237

atrybuty
 globalne, 63
 selektora, 201
 tabeli, 287
 znacznika
 <audio>, 410
 <embed>, 401
 <hr>, 143, 164
 <iframe>, 551
 <object>, 400
 <video>, 394

B

baza danych, 667
bezpieczeństwo, 615
 formularza, 351
bezwzględna ścieżka dostępu, 107
białe znaki, 62
biblioteka jQuery, 491
biblioteki JavaScript, 491
blok cytatu, 148
breadcrumb, 594
budowa adresu URL, 120

C

CDN, content delivery network, 514
ciasteczka, cookies, 668
ciąg tekstowy, 471, 646
 zapytania, 121
CRUD, create-update-delete, 660
CSS, 50, 169–204
 definiowanie kolorów, 178
 dla urządzeń mobilnych, 581
 elementy formularza, 357
 formatowanie tekstu, 133
 formatowanie znaku, 136
 kontrolowanie stosu elementów, 303
 model pudełkowy, 183
 pływający układ strony, 426
 pozycjonowanie elementów, 289
 projektowanie WWW, 427
 punkty graniczne, 443
 schematy pozycjonowania, 289
 tła, 229
 układ graficzny strony, 421
 układ strony, 415, 419
 zapytania o media, 444
 zasięg witryny, 432
 zmiana kolejności sekcji, 424
 zmiana właściwości, 498
CSS3, 446
cytat, 147
czcionka, 153, 422, 604
 właściwości, 137
czcionki sans-serif, 64

D

dane
 analityczne, 441
 tabeli, 248
definiowanie
 kolorów, 178
 kolumn, 446
 okręgu, 234
 prostokąta, 234
 publikacji treści, 33
 wielokąta, 234
deklaracja stylu, 64
document root, 108
dodanie
 obrazów do strony, 209
 silnika wyszukiwania, 592

 spisu treści, 156
 tekstu do obrazów, 209
 treści, 155, 504, 535
dokument HTML, 40, 61
dokumenty sieciowe, 38
dołączanie
 plików, 664
 stylów, 169
 zdarzeń, 495
DOM, document object model, 494, 531
dopełnienie, 186, 188
 komórki, 261
dostawca usług hostingowych, 51
dostęp
 do modelu DOM, 532, 538
 do obrazów, 624
 do plików, 623
 do serwera, 623
dostępność, 597, 599, 603, 605
dostosowanie
 odtwarzacza, 387
 układu, 446
DTD, document type definition, 71
działanie
 PHP, 639
 przeglądarki, 28
 serwera WWW, 613
dźwięk, 383

E

edycja stylów, 180
edytor tekstu, 40
 Coda, 41
 HTML-Kit, 40
 Komodo Edit, 40
 Notepad++, 40
 TextEdit, 41
 TextWrangler, 40
 typu WYSIWYG, 55
efekty specjalne, 507
element, 59
 <div>
 pływający po prawej stronie, 195
 wewnętrzny, 187
 wyśrodkowany, 189
 zagnieżdżony, 187
elementy
 na poziomie bloku, 133
 na poziomie znaku, 133

- pływające, 193
- strukturalne, 378
- śródliniowe, 183
- tabeli, 286
- w dokumencie, 56
- z własnymi szkieletami, 378

encje dla znaków, 149

- specjalnych, 151
- zarezerwowanych, 152

F

Facebook, 628

Flash, 397

Flowplayer, 406

format

- ASCII, 620
- GIF, 206, 620
- HTML, 620
- JPEG, 207, 620
- MPEG, 620
- PDF, 620
- PNG, 207, 620
- QuickTime, 620
- Shockwave Flash, 620
- SVG, 207, 620
- WAV, 620

formatowanie

- tekstu, 62, 133
- znaku, 136

formaty

- obrazów, 206
- wideo i kontenerów, 389

formularz rejestracyjny, 657

formularze, 313, 605

- biblioteka jQuery, 502
- elementy, 363
- grupowanie kontrolek, 348
- kontrolki, 323, 343
- kontrolki wyłączone, 350
- nawigacja, 349
- osadzanie klipów, 384
- PHP, 654
- planowanie, 362
- poła ukryte, 337
- stosowanie właściwości CSS, 357
- wysyłanie, 332
- wyświetlenie, 660
- zapewnienie bezpieczeństwa, 351
- zerowanie, 333

framework Bootstrap, 45

FTP, file transfer protocol, 51, 120–124, 622

funkcja, 479

- celsiusToFahrenheit(), 666
- require_once(), 665
- strip_tags(), 655

funkcje

- mediów, 442
- serwera, 639
- urządzeń mobilnych, 583
- wbudowane PHP, 652

G

geolokalizacja, 583

GIF, graphics interchange format, 206

Google Analytics, 632

Google Chrome, 29, 42

- Narzędzia dla programistów, 42

gramatyka, 572

grupowanie

- kolumn, 279
- kontrolek, 348
- wierszy, 282

H

hasło, 331, 615

hipertekst, 22

hosting, 49, 615

- bezpłatny, 617
- plików wideo, 385, 388
- samodzielny, 389

HTML, hypertext markup language, 55, 71

- formatowanie tekstu, 133

HTML 2.0, 64

HTML 3.2, 65

HTML 4.0, 65

HTML5, 66, 369

- układ strony, 371
- znaczniki strukturalne, 372

HTTP, hypertext transfer protocol, 120, 123

I

identyfikator, 173

- DOCTYPE, 71, 72

inspektor, 44

instalacja

- Google Analytics, 633
- PHP, 641

integracja multimediów, 383
interaktywność, 26
interfejs programowania aplikacji, API, 583
internetowe typy treści, 398

J

Java EE, 669
JavaScript, 465, 470, 517
 deklarowanie tablic, 481
 dla urządzeń mobilnych, 581
 dodawanie nowych funkcji, 535
 funkcje, 479
 obiekty, 482
 obsługa zdarzeń, 484
 operatory, 473
 operatory porównania, 476
 otwieranie powiązanych okien, 554
 pętle, 477
 słowa zarezerwowane, 475
 struktury sterujące, 475
 typy danych, 480
 ukrywanie treści, 526
 weryfikacja formularza, 517
 wyrażenia, 473
 wyszukiwanie elementów, 531
 wyświetlanie treści, 526
 zmiennne, 473
jednostka
 cm, 176
 em, 176
 ex, 176
 in, 176
 mm, 176
 pc, 176
 pt, 176
 px, 176
 rem, 176
 vh, 176
 vw, 176
język
 HTML, 55
 JavaScript, 465, 470
 PHP, 639, 642
 SQL, 667
 XML, 65
JPEG, joint photographic experts group, 207
jQuery, 491
 dołączanie zdarzeń, 495
 efekty specjalne, 507

 modyfikacja formularza, 502
 modyfikacja stylów, 496
 modyfikacja treści strony, 499, 504
 operacje na atrybutach, 503
 wybór elementów, 494
 wybór obiektów, 492
 zmiana właściwości CSS, 498

K

kanal alfa, 178
kaskadowe arkusze stylów, CSS, 50, 133, 174
katalog, 619
klasa, 173
 headline, 64
klawisze dostępu, 350
klient FTP, 622
kod źródłowy strony, 43
kodek, 389
 H.264, 390, 391
kodowanie znaków, 149
kolor, 177, 422, 603
 tabeli, 265
kolumna, 263, 446
komentarze, 78, 642
komórka, 261
 tabeli, 248
konfiguracja
 hostingu WWW, 50
 własnego serwera, 617
 wyskakujących okien, 555
konkatenacja, 646
konsorcjum W3C, 34
konstrukcje warunkowe, 647
kontekst, 594
kontrolka
 hasła, 331
 typu date, 330
 typu file, 337
 typu range, 329
kontrolki
 formularza, 343
 HTML5, 328
 tekstowe, 324
 wyłączone, 350
kontrolowanie wykonywania pętli, 651
konwersja wideo, 391
kotwica, 101, 114
kreatory witryn, 617

L

linia pozioma, 142
lista, 83
 definicji, 92
 nieuporządkowana, 89
 style, 90
 numerowana, 84
 uporządkowana, 84
 alternatywny styl numerowania, 87
 style, 86
 wypunktowana, 89
 zagnieżdżona, 93
listy
 elementy, 96
 właściwości, 97
logika, 660

Ł

łącza, 22, 101, 182, 221, 601
 do elementów, 119
 do innych dokumentów, 109
 do określonego położenia, 114
 menu, 111
 między sekcjami, 115
 nieodwiedzone, 182
 odwiedzone, 182
 stan aktywny, 182
 stan hover, 182
 śródliniowe, 300
 typu mailto, 125
 w postaci przypisów, 578
łączenie
 stron lokalnych, 106
 wierszy i kolumn, 271

M

mapa obrazu, 231
margines, 186, 188
marketing treści, 626
menu rozwijane, 95, 306
metoda, 471
 appendChild(), 538
 event.preventDefault(), 520
 fadeOut(), 508
 GET, 315, 321
 getElementById(), 532
 getElementByName(), 532

 getElementsByTagName(), 532
 hasClass(), 501
 insertBefore(), 538
 load(), 513
 POST, 315, 321, 655
 ready(), 493
 removeAttribute(), 538
 removeChild(), 538
 replaceChild(), 538
 setAttribute(), 538
 toggle(), 497
 toggleClass(), 500
Microsoft ASP.NET, 669
Microsoft Internet Explorer, 29
MIME, 399
mobilna wersja witryny, 563, 585
 CSS i JavaScript, 581
 organizacja stron, 569
 spójne elementy stron, 574
 spójne formy nawigacji, 574
 układ strony, 572
 użycie łącz, 574
 użycie obrazów, 578
 weryfikacja stron, 566
model pudełkowy, 183
modyfikacja
 listy uporządkowanej, 85
 stylów, 496
 treści strony, 499
Mozilla Firefox, 30

N

nadawanie stylu
 witrynie, 169–204
 znacznika <meter>, 357
nagłówki, 75, 373
 tabeli, 248
nakładanie dynamiczne, 298
narzędzia, 37
 Narzędzia dla programistów, 42, 180, 468
narzędzie Color Schemer, 179
nawiązywanie połączeń telefonicznych, 583
nawigacja, 375, 594
 po formularzu, 349
nazwa domeny, 51
nazwane encje, 151
nazwy plików, 619
nieanonimowy FTP, 124
numerowane encje, 151

O

obiekt, 471, 482
obramowanie, 183
 tabeli, 256
obrazy, 205, 578, 602
 dodanie tekstu, 209
 dostosowanie ilości miejsca, 219
 formaty, 206
 i łączy, 221
 i tekst, 212
 jako przyciski, 335
 pływające, 216
 pobranie, 232
 śródliniowe, 208
 współrzędne, 238
 wymiary, 225
 wyrównanie, 213
 zawijanie tekstu, 216
obsługa
 parametrów, 655
 urządzeń mobilnych, 437
odległość między komórkami, 262
odtwarzacz
 Flowplayer, 406
 JW Player, 404
 osadzonej treści audio, 409, 411
odtwarzacze
 audio, 410
 Flash, 404
okna, 543, 545
okno nadrzędne, 546
określanie koloru, 177
okrąg, 234
opcje pól tekstowych, 326
operacje
 na atrybutach, 503
 na klasach, 499
 na wartościach formularza, 502
operator, 472
 konkatenacji, 646
operatory
 porównania, 476
 warunkowe, 648
optymalizacja dla wyszukiwarek
 internetowych, SEO, 630
organizacja
 arkusza stylów, 429
 plików, 619
ortografia, 572

osadzanie

 treści audio, 409
 treści Flash, 397, 402
 plików wideo, 383, 393
osadzenie stylów w znacznikach, 64

P

parametry z wieloma wartościami, 655
parkowanie domeny, 616
paski boczne, 377
PEAR, php extension and application
 repository, 667
pętla, 477
 do-while, 651
 for, 477, 650
 foreach, 649
 REPL, 469
 while, 478, 651
PHP, 639, 642
 ciągi tekstowe, 646
 dołączanie plików, 664
 funkcje użytkownika, 652
 funkcje wbudowane, 652
 komentarze, 642
 konstrukcje warunkowe, 647
 obsługa parametrów, 655
 operatory warunkowe, 648
 połączenie z bazą danych, 667
 przetwarzanie formularzy, 654
 tablice, 644
 weryfikacja formularza, 656
 wysyłanie wiadomości e-mail, 668
 wyświetlenie formularza, 660
 zmiennne, 643
planowanie
 formularza, 362
 responsywnego projektu, 441
 strony internetowej, 154
plik HTML, 58
pliki
 .gif, 620
 .htm, 620
 .html, 620
 .jpeg, 620
 .jpg, 620
 .mov, 620
 .mp3, 620
 .mp4, 620
 .pdf, 620

- .png, 620
- .svg, 620
- .swf, 620
- .txt, 620
- .wav, 620
- dołączanie za pomocą PHP, 664
- dzienników zdarzeń, 631
- przekazywanie, 622
- przenoszenie, 621
- publikacja, 620
- wideo, 383
- pływające kolumny, 426
- pływanie elementów, 193
- PNG, portable network graphics, 207
- pobranie
 - obrazu, 232
 - współrzędnych obrazu, 238
- podpis strony, 160, 586
- podział wiersza, 144
- pole
 - tekstowe, 339
 - wyboru, 333
- polecenie if, 476
- połączenia telefoniczne, 583
- połączenie
 - dwóch stron, 103
 - strony z siecią, 109
 - z bazą danych, 667
- położenie względne elementu, 290
- poprawa wydajności, 466
- posty gościnne, 626
- pozycjonowanie
 - bezwzględne, 294, 421
 - elementów, 289
 - stałe, 301
- preferencje użytkownika, 596
- problem
 - dostępu do obrazów, 624
 - dostępu do plików, 623
 - dostępu do serwera, 623
 - z łączami, 624
 - z wyświetlaniem plików, 625
- procedura obsługi zdarzeń, 485, 533
- procesor tekstu, 60
- program Handbrake, 391
- programowanie zorientowane obiektowo, 668
- projektowanie
 - dla urządzeń mobilnych, 563
 - formularzy, 313
- promowanie witryny, 625, 631
- Google Analytics, 632
- gościnne posty, 626
- indeksy witryn, 629
- łącza zewnętrzne, 626
- marketing treści, 626
- serwisy społecznościowe, 627
- prostokąt, 234
- protokół, 120
 - FTP, 51, 120
 - HTTP, 120, 123
 - SCP, 51
 - SFTP, 51
- przeglądarka WWW, 27, 42
 - Apple Safari, 30
 - Google Chrome, 29, 42
 - Microsoft Internet Explorer, 29
 - Mozilla Firefox, 30
- przeglądarki
 - alternatywne, 598
 - w urządzeniach mobilnych, 31
- przekazywanie plików, 622, 668
 - do YouTube'a, 386
 - na serwer, 51
- przenoszenie
 - plików, 621
 - sekcji, 424
- przepełnienie pudełka, 193
- przepływ normalny, 289
- przetwarzanie
 - formularzy, 654
 - pliku, 615
- przezroczystość, 178
- przycisk, 332–336
- przygotowanie plików, 618
- pseudoklasa, 198
 - a:active, 181
 - a:hover, 181
 - a:link, 181
 - a:visited, 181
 - first-letter, 198
 - first-line, 198
 - focus, 199
 - hover, 199
- publikacja, 613
 - przygotowanie plików, 618, 620
 - treści, 21, 33, 39
- punkt
 - graniczny CSS, 443
 - kontrolny, 522

R

ramki, 543, 605
 śródliniowe, 543, 550
reguła #sitenav, 421
rejestracja domeny, 51
reklama, 631
REPL, read-eval-print-loop, 469
responsywne
 obrazy i wideo, 452
 tabele, 454
responsywny projekt WWW, 435
rodzaje
 adresów URL, 123
 kontrolek formularza, 343
rozszerzalny język znaczników, 65
Ruby on Rails, 669

S

schemat pozycjonowania
 absolute, 289, 295
 fixed, 289, 302
 relative, 289, 292, 295
SCP, secure copy, 51, 621
sekcja, 372
 #main, 421
 508, 598
 z łączem, 119
sekcje
 nadchodzących zdarzeń, 159
 strony, 417
selektory, 171, 197
 jQuery, 494
 kontekstowe, 172
SEO, search engine optimization, 630
serwer
 HTTP, 613
 WWW, 23, 31, 38, 613
serwis
 Vimeo, 388
 YouTube, 386
serwisy społecznościowe, 627
SFTP, secure ftp, 51
SGML, standard generalized markup
 language, 55
sieci
 dynamiczność, 24
 interaktywność, 26
 niezależność od platformy, 23
 rozproszenie, 23

sieć

 CDN, 514
 mobilna, 564
 WWW, 22
silnik wyszukiwania, 592
skalowanie, 225
skrypty, 614
 jQuery, 492
 typu polyfill, 379
słowa
 kluczowe kontekstu przeglądania, 545
 zarezerwowane, 475
słownik, 644
słowo kluczowe this, 486
smartfon, 563
spójne
 elementy stron, 574
 formy nawigacji, 574
sprawdzenie
 dostępności witryny, 605
 gramatyki i ortografii, 572
SQL, structured query language, 667
stopka, 374
stos elementów, 303
strona
 główna, 38
 internetowa, 38
struktura
 kodu HTML, 71
 skryptu, 467
 strony, 55
 strony w HTML5, 369
struktury sterujące, 475
style
 listy uporządkowanej, 86
 listy wypunktowanej, 90
 pozycjonowania, 424
SVG, scalable vector graphics, 207
SWFObject, 402
szerokość kolumny, 263
szkielet
 strony, 48, 377
 witryny internetowej, 47

Ś

ścieżka dostępu, 107, 121
 bezwzględna, 107
 względna, 107
środowisko JavaScript, 483

T

tabela, 140, 247
 kolorów, 256
tabele, 600
 atrybuty, 287
 elementy, 286
 elementy składowe, 248
 kolor, 265
 komórki, 249, 256
 obramowanie, 256
 podpis, 248, 252
 puste komórki, 252
 rozmiar, 256
 tworzenie, 253
 usprawnienia, 279
 wiersze, 249
 wyrównanie treści, 267
tablica, 481, 644
 \$_POST, 655
tablice hash, 644
technologia AJAX, 510
tekst, 212
 elementy, 164
 właściwości, 165
testowanie wyniku, 163
tła, 226
treść, 46
 multimedialna, 602
tworzenie
 formularza, 314
 frameworka, 154
 kontrolerek formularza, 323
 kontrolerek tekstowych, 324
 kontrolki hasła, 331
 kontrolki pola wyboru, 333
 łączy, 101
 łączy i kotwic, 114
 ogólnego przycisku, 336
 opisu, 156
 pliku HTML, 60, 79
 przycisku
 opcji, 334
 wysyłającego formularz, 332
 zerującego formularz, 333
 responsywnych tabel, 454
 rozwijanego menu, 306
 sekcji, 158
 stylów
 na poziomie strony, 169
 na poziomie witryny, 170

 szkieletu, 48
 tabel, 247, 253
 zapytań o media, 442
typ
 file, 125
 mailto, 125
typy
 danych, 480
 mediów, 442
 plików, 620
tytuł, 73

U

układ
 graficzny strony, 421
 oparty na tabelach, 416
 strony, 56, 369, 371, 415, 419
ukośnik, 59
ukryte pola formularza, 337
ukrywanie
 elementów, 496
 treści, 526
umieszczenie łącza, 536
URL, uniform resource locator, 24, 120
urządzenia mobilne, 438, 563
ustalenie
 użytkowników, 631
 współrzędnych, 233
usuwanie treści, 504
użycie
 atrybutu style, 63
 biblioteki jQuery, 491
 CSS, 169–204
 JavaScriptu, 517
 klawiszy dostępu, 350
 koloru, 181
 list, 95
 łączy, 222, 574
 obrazów, 205, 578
 obrazów jako wypunktowania, 230
 projektu responsywnego, 435
użytkownik, 591
 ustalenie preferencji, 596

V

Vimeo, 388

W

W3C, world wide web consortium, 34
warstwa prezentacyjna, 660
wartości atrybutu sandbox, 554
wartość zwrotna, 653
WCAG, 605
weryfikacja
 formularza, 517, 524, 656
 stron, 566
weryfikatory HTML, 149
wewnętrzny element <div>, 187
węzeł, 534
WHATWG, 66
video, 383
wielkość elementu, 191
wielokąt, 234
witryna internetowa, 24, 37
właściwości
 CSS, 97, 165
 czcionki, 137, 153
 pozycjonowania, 294
 skrót, 184
 węzła, 534
właściwość
 background-image, 226
 border-spacing, 262
 border-style, 184
 clear, 195
 float, 193
 text-decoration, 137
współrzędne, 233
WWW, world wide web, 21
wybór
 elementów, 494
 funkcji dołączającej plik, 665
wydajność, 466
wypunktowanie, 230
wyrażenia, 472
 regularne, 667
wyrównanie
 kolumn, 279
 komórki, 268
 obrazu i tekstu, 213
 podpisu tabeli, 268
 tabeli, 267
 wierszy, 282
WYSIWYG, 55
wyskakujące okna, 554
wysyłanie wiadomości e-mail, 668

wyszukiwanie, 592
 elementów, 531
wyszukiwarki internetowe, 629
wyśrodkowany element <div>, 189
wyświetlanie
 elementów, 191, 496
 treści, 526
 uaktualnień, 352
 formularza, 660
 wyniku, 61
względna ścieżka dostępu, 107
względne położenie elementu, 290

X

XHTML 1.0, 65
XML, extensible markup language, 65
XSS, cross-site scripting, 655

Y

YouTube, 384, 386

Z

zagnieżdżanie
 elementów pozycjonowanych
 bezwzględnie, 297
 znaczników, 106
 znacznika <a>, 104
 znacznika <div>, 187
zamykanie pustych elementów, 142
zapytania o media, 442
zarządzanie
 plikami, 614
 treścią, 50
zatrzymanie zawijania tekstu, 217
zawijanie tekstu, 216, 217
zdalne strony, 109
zdarzenia, 484
zmiana
 kolejności sekcji, 424
 obramowania tabeli, 258
 wielkości elementu, 191
 właściwości arkusza stylów, 498
zmiennne, 473, 643
znacznik, 58
 <!DOCTYPE>, 60
 <a>, 101, 115, 544
 <abbr>, 135

<address>, 146, 164, 586
<area>, 234
<article>, 376, 380
<aside>, 377, 380
<audio>, 409, 412
, 136, 164, 604
<base target="window">, 557
<base>, 549
<blackquote>, 147, 370
<blockquote>, 164
<body>, 80, 73, 202

, 144, 165
<button>, 339, 363
<caption>, 252, 286
<cite>, 135, 148, 164
<code>, 134, 164
<col>, 286
<colgroup>, 279, 286
<datalist>, 326
<dd>, 84, 92, 97
<dfn>, 135, 164
<dir>, 83
<div>, 173, 187
<dl>, 92, 97
<dt>, 84, 92, 97
, 134, 164
<embed>, 393, 401, 412
<fieldset>, 348, 364
, 416
<footer>, 374, 380
<form>, 319
<frameset>, 543
<h1>, 64, 80
<h6>, 81
<head>, 72, 80
<header>, 373, 380
<hr>, 142, 143, 164
<html>, 72, 80
<i>, 136, 164, 604
<iframe>, 544, 551, 557
, 208, 225, 229
<input>, 323, 331, 363
<kbd>, 134, 164
<label>, 322, 364
<legend>, 348, 364
, 84
<map>, 234
<menu>, 83
<meter>, 355, 364
<nav>, 307, 375, 380

<object>, 397, 399, 400, 408, 412
, 84, 87, 96
<option>, 340, 364
<p>, 77, 81, 134
<param>, 398, 412
<pre>, 62, 140, 164
<progress>, 352, 364
<q>, 148, 164
<samp>, 134, 164
<script>, 467, 493
<section>, 373, 380
<select>, 340, 363
<small>, 136, 164
<source>, 396, 412
, 164, 198
, 134, 164
<style>, 170
<sub>, 136, 164
<sup>, 136, 164
<table>, 248, 286, 416
<tbody>, 282, 286
<td>, 250, 286
<textarea>, 339, 363
<tfoot>, 282, 286
<th>, 250, 286
<thead>, 282, 286
<title>, 74, 80
<tr>, 286
<track>, 603
<u>, 136, 164
, 84, 89, 97
<var>, 164
<video>, 393, 408, 412

znaczniki

atrybuty, 59, 287
osadzenie stylów, 64
otwierające, 59
semantyczne, 134
strukturalne, 372
strukturalne HTML5, 417
zamykające, 59

znak

#, 115, 118, 174
\$, 493
&, 151
%, 125

znaki

specjalne, 149, 151
zarezerwowane, 152

zwijanie marginesów, 189

PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



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

Zmień swoją stronę WWW
w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

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

GRUPA WYDAWNICZA

 **Helion SA**

HTML, CSS i JavaScript

Dla każdego

Za dwadzieścia pięć lekcji staniesz się projektantem!

Internet to nasza codzienność. Znakomita większość firm ma swoje strony internetowe. Coraz więcej ludzi prowadzi blogi czy profile w mediach społecznościowych. I choć żeby publikować w internecie, nie musisz być profesjonalnym projektantem, to aby mieć kontrolę nad wyglądem i funkcjonalnością witryny, warto znać podstawy HTML, CSS i JavaScriptu. To wcale nie takie trudne!

Dzięki tej książce nauczysz się pisać poprawne, sprawnie działające strony WWW. Znajdziesz tu dwadzieścia pięć lekcji — każda zajmie Ci około godziny. W jednym miejscu zebrano i uporządkowano wszystkie informacje niezbędne do budowania witryn internetowych. Nie tylko pokazano tu, jak tworzyć świetne witryny internetowe, ale również zawarto wskazówki, jak zachęcić użytkowników do ich odwiedzania. Poszczególne zagadnienia zostały zaprezentowane w przejrzysty sposób, a przyswojenie materiału dodatkowo ułatwiają zamieszczone w książce pytania sprawdzające, ćwiczenia i quizy.

Najważniejsze zagadnienia ujęte w książce:

- podstawy pisania stron w HTML i CSS
- formatowanie treści, tworzenie tabel, list, dodawanie grafiki i formularzy do stron WWW
- przygotowywanie układu strony i zapewnienie jej responsywności
- język JavaScript i biblioteka jQuery
- prawidłowe projektowanie stron WWW i publikowanie ich w internecie

Laura Lemay jest jedną z najpoczytniejszych na świecie autorek książek dotyczących programowania sieciowego. **Rafe Colburn** jest programistą sieciowym z przeszło piętnastoletnim doświadczeniem w dziedzinie budowania witryn internetowych. Napisał kilka książek o różnych technikach tworzenia aplikacji sieciowych. **Jennifer Kyrynin** jest projektantką witryn WWW. Technologiami internetowymi zajmuje się od 1995 roku. Autorka wielu popularnych podręczników z tej dziedziny.

Helion	
księgarnia internetowa	
http://helion.pl	
zamówienia telefoniczne	
	0 801 339900
	0 601 339900
Informatyka w najlepszym wydaniu	

SAMS

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
<http://helion.pl>

Sprawdź najnowsze promocje:
• <http://helion.pl/promocje>
Książki najchętniej czytane:
• <http://helion.pl/bestsellery>
Zamów informacje o nowościach:
• <http://helion.pl/novosci>



cena: 99,00 zł

