



HTML i CSS

Zaprojektuj i zbuduj witrynę WWW
Podręcznik Front-End Developera

JON DUCKETT

Tytuł oryginału: HTML and CSS: Design and Build Websites

Tłumaczenie: Piotr Rajca

Skład: Ewa Galczak

ISBN: 978-83-283-4481-5

©2011 by John Wiley & Sons, Inc., Indianapolis, Indiana

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior permission of the publisher.

Wiley and the Wiley logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

Translation copyright © 2014 by Helion S.A.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiekolwiek postaci jest zabronione.

Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

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

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

Wydawnictwo HELION

ul. Kościuszki 1c, 44-100 GLIWICE

tel. 32 231 22 19, 32 230 98 63

e-mail: helion@helion.pl

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

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:

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

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

http://helion.pl/user/opinie/htcswv_ebook

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

- [Poleć książkę na Facebook.com](#)
- [Kup w wersji papierowej](#)
- [Oceń książkę](#)

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

EXECUTIVE EDITOR

Carol Long

MARKETING MANAGER

Ashley Zurcher

PRODUCTION MANAGER

Tim Tate

PRODUCTION EDITOR

Daniel Scribner

VICE PRESIDENT**AND EXECUTIVE GROUP****PUBLISHER**

Richard Swadley

VICE PRESIDENT AND**EXECUTIVE PUBLISHER**

Barry Pruett

ASSOCIATE PUBLISHER

Jim Minatel

PRODUCTION COORDINATOR,**COVER**

Katie Crocker

AUTHOR

Jon Duckett

COVER DESIGNER

Emme Stone

DESIGN AND LAYOUT

Jon Duckett

Emme Stone

TECHNICAL EDITOR

Chris Mills

TECHNICAL REVIEWERS

Andy Stone

Angela Shimell

Donna Watson

Martin Callanan

Rob Jacoby

Tony Berry

PHOTOGRAPHY

John Stewardson

johnstewardson.com

ADDITIONAL PHOTOGRAPHY

Hesperian

Joe Robertson

[flickr.com/photos/](http://flickr.com/photos/mindfire)

mindfire

Jules Clancy

thestonesoup.com

Kylie Gusset

gusset.net

Michael Stillwell

beebo.org

SPIS TREŚCI

Wprowadzenie	1
Rozdział 1. Struktura	10
Rozdział 2. Tekst	38
Rozdział 3. Listy	60
Rozdział 4. Łącza	72
Rozdział 5. Obrazy	92
Rozdział 6. Tabele	124
Rozdział 7. Formularze	142
Rozdział 8. Dodatkowe elementy HTML	174
Rozdział 9. Flash, wideo i audio	198
Rozdział 10. Prezentacja CSS	224
Rozdział 11. Kolor	244
Rozdział 12. Tekst	262
Rozdział 13. Pudełka elementów	298
Rozdział 14. Listy, tabele i formularze	328
Rozdział 15. Układ	356
Rozdział 16. Obrazy	404
Rozdział 17. Układ w HTML5	426
Rozdział 18. Proces i projekt	450
Rozdział 19. Informacje praktyczne	474
Skorowidz	493

WPROWADZENIE

- ▶ O książce.
- ▶ Jak działa WWW?
- ▶ Nauka poprzez analizę innych stron.

Przede wszystkim dziękuję za wybranie tej książki. Została ona napisana z myślą o dwóch całkowicie odmiennych typach odbiorców:

- tych, którzy chcą się nauczyć projektowania i tworzenia witryn WWW od podstaw;
- wszystkich, którzy mają własną witrynę (być może stworzoną przy użyciu systemu zarządzania treścią, systemu blogowego albo platformy do handlu elektronicznego) i chcą mieć większą kontrolę nad wyglądem strony.

Jedyną rzeczą niezbędną do korzystania z tej książki jest komputer z przeglądarką WWW oraz edytorem tekstów (takim jak Notatnik w systemie Windows lubTextEdit na komputerach Mac).

Przykłady

Wszystkie przykłady zamieszczone w niniejszej książce można pobrać z serwera FTP wydawnictwa Helion — <ftp://ftp.helion.pl/przykłady/htcsww.zip>. Znajdują się w niej także odwołania do witryny poświęconej tej publikacji — <http://www.htmlandcssbook.com/>. Witryna ta została stworzona jako dodatek do książki — można tam przejrzeć wszystkie przedstawione w niej przykłady, jak również znaleźć wiele przydatnych informacji dotyczących np. przygotowywania treści multimedialnych przeznaczonych do umieszczenia na stronach WWW, popularnych programów graficznych, których można w tym celu używać, sposobów kodowania materiałów wideo czy implementacji mechanizmów wyszukiwania na własnych stronach WWW. Trzeba jednak pamiętać, że zarówno przykłady, jak i wszystkie informacje podane w tej witrynie są napisane w języku angielskim.



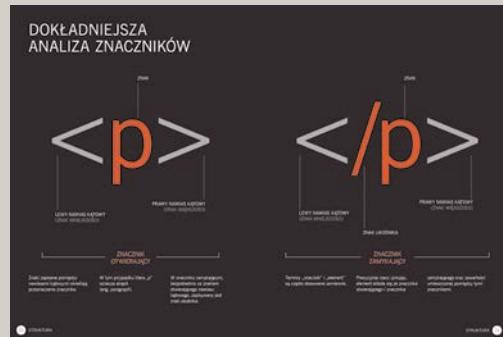
Na początku każdego rozdziału znajduje się **wprowadzenie**, które przedstawia zagadnienia opisane w danej części książki.

Strony z **dodatkowymi informacjami** mają białe tło; wyjaśniają kontekst zagadnień opisanych w danym rozdziale.



Przykład jest podsumowaniem informacji poznanych w danym rozdziale, demonstruje, w jaki sposób można je zastosować w praktyce.

Strony **informacyjne** przedstawiają kluczowe elementy kodu HTML i CSS. Kod HTML jest w kolorze niebieskim, CSS — w kolorze różowym.



Diagramy oraz infografiki są przedstawiane na ciemnym tle. Stanowią prostą, wizualną prezentację opisywanych zagadnień.

Na końcu każdego rozdziału znajduje się **podsumowanie**. Przypomina ono kluczowe zagadnienia przedstawione w danym rozdziale.

CZY NAUKA HTML I CSS JEST TRUDNA?

Wiele książek prezentujących język HTML oraz CSS przypomina nudne podręczniki. Aby ułatwić Czytelnikowi zadanie, odrzuciliśmy standardowe szablony używane zazwyczaj przez wydawców i od podstaw przeprojektowaliśmy tę książkę.

W pracy, kiedy ktoś spogląda na ekran mojego monitora i widzi wiersze kodu, często słyszę komentarze, że wygląda on na skomplikowany, a ja muszę być bardzo mądry, skoro go rozumiem. Prawda jest taka, że bez trudu można się nauczyć tworzenia stron WWW oraz analizy ich kodu — i wcale nie trzeba być „programistą”.

Poznanie i zrozumienie języków HTML i CSS może pomóc wszystkim, których praca jest w jakiś sposób związana z WWW; projektanci będą mogli tworzyć bardziej atrakcyjne i użyteczne witryny, redaktorzy — formułować lepsze treści, specjaliści od reklamy — efektywniej komunikować się z odbiorcami, a menadżerzy — zlecać doskonalsze witryny i w optymalny sposób wykorzystywać możliwości swoich zespołów.

Skoncentrowałem się na kodzie, którego Czytelnik będzie używać w 90% przypadków, pomijając taki, z którym spotka się sporadycznie, nawet jeśli będzie pracować nad tworzeniem stron WWW na cały etat. Choćby Czytelnik napotkał kod, który został tu pominięty, to po przeczytaniu tej książki będzie w stanie szybko i łatwo, korzystając z wyszukiwarki, dowiedzieć się, co on oznacza.

W książce zamieściłem praktyczne informacje dotyczące zagadnień, które w niej opisałem, a o które jestem często pytany, np.: w jaki sposób przygotowywać obrazy, wideo oraz audio do wykorzystania na stronach WWW, jak zabrać się do projektowania i tworzenia nowej witryny, jak poprawić jej położenie w wynikach wyszukiwarki (SEO) oraz jak korzystać z usługi Google Analytics, by dowiedzieć się czegoś o osobach odwiedzających stronę.

STRUKTURA KSIĄŻKI

Aby nauczyć Czytelnika sposobów tworzenia stron WWW, książkę podzielono na trzy części:

1. HTML

W pierwszym rozdziale mowa o tym, w jaki sposób język HTML jest używany do tworzenia stron WWW. Czytelnik przekona się, że wszystko rozpoczyna się od zapisania słów, które mają się pojawić na stronie. Dopiero później dodaje się do nich znaczniki i elementy, dzięki którym przeglądarka wie, co jest nagłówkiem, gdzie mają się zaczynać i kończyć poszczególne akapity itd.

Kolejne rozdziały pierwszej części książki przedstawiają znaczniki używane do tworzenia witryn WWW. Zostały one pogrupowane tematycznie i dotyczą następujących zagadnień: tekstu, list, łączy, obrazów, tabel, formularzy, video, audio oraz animacji Flash, a także wybranych innych elementów.

Z góry uprzedzam, że przykłady zamieszczone w pierwszych rozdziałach książki nie są szczególnie atrakcyjne wizualnie, niemniej stanowią podstawę wszystkich stron WWW. Kolejne rozdziały poświęcone CSS pokazują, w jaki sposób sprawić, by tworzone strony wyglądały znacznie atrakcyjniej.

2. CSS

Ta część książki rozpoczyna się od rozdziału, który wyjaśnia, jak CSS używa reguł, by zapewnić twórcom strony możliwość określania jej wyglądu i układu. Następnie przedstawia właściwości CSS, których można używać podczas tworzenia arkuszy stylów. Generalnie właściwości te można zaliczyć do dwóch głównych kategorii:

Prezentacja: jak kontrolować takie aspekty wyglądu jak kolor tekstu, używane czcionki oraz ich wielkość, jak określić kolor tła strony (lub jej fragmentów) oraz jak stosować obrazy tła.

Układ: jak kontrolować rozmieszczenie poszczególnych elementów na ekranie. Czytelnik pozna przy okazji kilka technik, których używają profesjonalisci, by poprawić wygląd tworzonych stron WWW.

3. INFORMACJE PRAKTYCZNE

W tej części książki zamieszczonych zostało nieco praktycznych informacji, które mogą Czytelnikowi pomóc w tworzeniu lepszych witryn.

Przyjrzymy się nowym znacznikom wprowadzonym w języku HTML5, ułatwiającym opisywanie struktury stron. HTML5 to najnowsza wersja języka HTML (która wciąż jest rozwijana i tworzona). Jednak przed poznaniem tych elementów Czytelnik musi dobrze poznać zasady stosowania CSS do określania i kontroli układu stron. Jeden z rozdziałów przedstawia proces projektowania witryny, według którego można postępować podczas tworzenia nowych witryn.

Na końcu książki znalazł się opis zagadnień, które mogą się przydać już po utworzeniu witryny, takich jak umieszczenie jej w internecie, optymalizacja pod kątem wyszukiwarek (SEO) oraz stosowanie narzędzi analitycznych do zbierania informacji o tym, kto ogląda witrynę i czego w niej szuka.

RÓŻNE SPOSOBY UZYSKIWANA DOSTĘPU DO WWW

Przed spojrzeniem na kod używany do tworzenia witryn WWW warto zastanowić się nad różnymi sposobami uzyskiwania dostępu do sieci oraz wyjaśnić terminologię związaną z tymi zagadnieniami.

PRZEGŁĄDARKI

Witryny są oglądane przy użyciu programów nazywanych **przeglądarkami WWW**. Do najpopularniejszych programów tego typu należą Mozilla Firefox, Internet Explorer, Safari, Chrome oraz Opera.

Aby obejrzeć stronę WWW, użytkownik musi podać w przeglądarce jej adres, kliknąć łącze umieszczone na innej stronie lub skorzystać z adresu zapisanego wcześniej w przeglądarce.

Producenci oprogramowania regularnie udostępniają kolejne wersje swoich przeglądarek, wyposażone w nowe możliwości i obsługujące nowsze wersje języków. Trzeba jednak pamiętać, że wielu właścicieli komputerów nie będzie miało najnowszych wersji przeglądarek. Dlatego nie można zakładać, że osoby odwiedzające witrynę będą w stanie skorzystać z udostępnianych przez nią możliwości.

W rozdziale 19. jest mowa o tym, jak sprawdzić, których przeglądarek używają osoby oglądające witrynę.

SERWERY WWW

Kiedy prosimy przeglądarkę o wyświetlenie strony, wysyła ona przez internet żądanie do specjalnego komputera nazywanego **serwerem WWW**, na którym jest umieszczona witryna.

Serwery WWW to specjalne komputery, które są bezustannie podłączone do internetu i zostały zoptymalizowane pod kątem wysyłania stron WWW do wszystkich, którzy o nie poproszą.

Niektóre duże firmy posiadają własne serwery, jednak znacznie częściej korzysta się z **usług specjalnych firm**, które odpłatnie umożliwiają umieszczenie witryn na swoich serwerach.

URZĄDZENIA

Do oglądania witryn WWW używanych jest coraz więcej urządzeń: od komputerów stacjonarnych i laptopów, poprzez tablety, po telefony komórkowe. Trzeba pamiętać, że różne urządzenia są wyposażone w ekranы o różnych rozmiarach, a połączenia z internetem, jakimi dysponują, mogą mieć różne szybkości.

CZYTNIKI EKRANOWE

Czytniki ekranowe to programy, które są w stanie odczytać na głos zawartość ekranu komputera. Są one stosowane przez osoby z upośledzeniem wzroku.

Prawo w wielu krajach wymaga, by budynki publiczne były łatwo dostępne dla osób z różnymi upośledzeniami. Podobnie jest z witrynami WWW — prawo wymaga, by także były łatwo dostępne dla takich osób.

W niniejszej książce można znaleźć kilka wzorzeców o czytnikach ekranowych. Informacje te ułatwiają Czytelnikowi zadbanie o to, by tworzone witryny były łatwo dostępne dla osób korzystających z takiego oprogramowania.

Warto zauważyć, że technologie podobne do tych wykorzystywanych w czytnikach ekranowych są stosowane również w innych przypadkach — gdy użytkownicy nie są w stanie czytać informacji prezentowanych na ekranie urządzenia, ponieważ np. prowadzą samochód lub biegają.

JAK SĄ TWORZONE WITRYNY WWW?

Wszystkie witryny są tworzone przy użyciu języków HTML i CSS, jednak systemy zarządzania treścią, platformy blogowe oraz platformy do prowadzenia handlu elektronicznego często uzupełniają tę parę o dodatkowe technologie.

TO, CO WIDZIMY

Kod HTML i CSS wyświetlany przez przeglądarkę w witrynie najprawdopodobniej pochodzi z serwera, na którym ona działa. Przeglądarka WWW interpretuje kod HTML i CSS, tworząc na ich podstawie strony, które oglądamy.

Większość witryn WWW zawiera dodatkowe materiały, takie jak obrazy, klipy audio i wideo, animacje; w niniejszej książce Czytelnik znajdzie informacje o tym, jak przygotowywała takie materiały oraz umieszczać je na stronach.

Niektóre witryny przesyłają do przeglądarki także skrypty JavaScript lub animacje Flash. Niniejsza książka pokazuje, jak je dodawać do tworzonych stron. Obie te technologie są zagadnieniami zaawansowanymi, które być może Czytelnik będzie chciał poznać po opanowaniu języków HTML i CSS.

JAK TO JEST STWORZONE

Niewielkie witryny WWW czasami tworzy się, korzystając wyłącznie z kodu HTML i CSS.

Jednak te większe — a zwłaszcza te, których zawartość jest regularnie aktualizowana i które korzystają z systemów zarządzania treścią (CMS), platform blogowych lub oprogramowania do prowadzenia handlu elektronicznego — często wymagają użycia znacznie bardziej złożonych technologii; choć są one w ostateczności stosowane do wygenerowania kodu HTML i CSS przesyfanego do przeglądarki użytkownika. A zatem jeśli tworzona witryna używa tych technologii, Czytelnik będzie w stanie wykorzystać zdobytą tu nową wiedzę o językach HTML i CSS, by mieć większą kontrolę nad wyglądem swoich stron.

Takie większe i bardziej złożone witryny mogą korzystać z baz danych oraz języków programowania wykonywanych po stronie serwera, takich jak PHP, ASP.Net, Java lub Ruby, jednak nie trzeba ich stosować, by poprawić wygląd stron oglądanych przez użytkowników. Nowe umiejętności, które Czytelnik zdobędzie w trakcie lektury tej książki, powinny mu w zupełności do tego wystarczyć.

HTML5 i CSS3

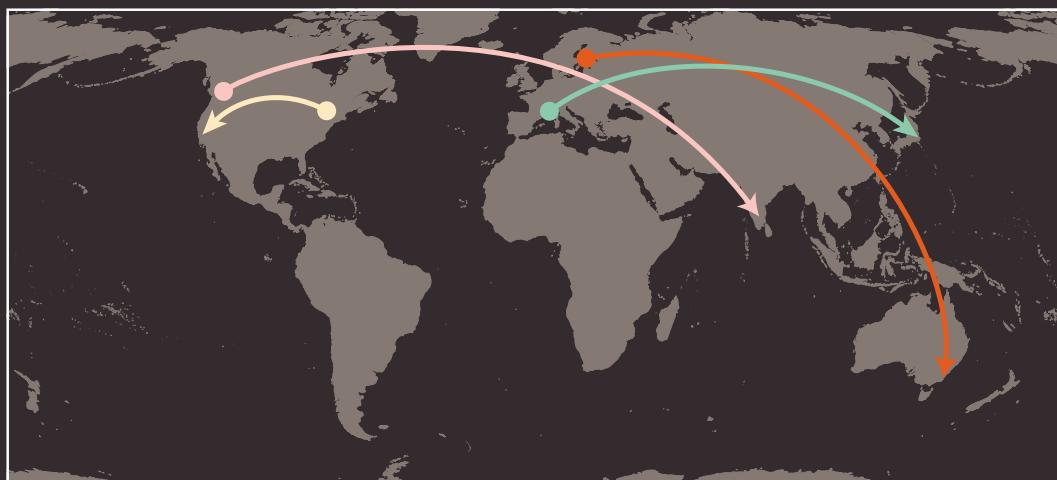
Od czasu powstania WWW pojawiło się już kilka wersji języków HTML i CSS — z których każda miała być usprawnieniem poprzednich.

W czasie gdy powstawała ta książka, HTML5 i CSS3 wciąż były w fazie rozwoju. Choć ich potencjał nie został jeszcze do końca określony, sporo przeglądarek już obsługuje niektóre spośród nowych możliwości tych języków, a wiele osób korzysta z nich w swoich witrynach. Dlatego zdecydowałem się przedstawić Czytelnikowi możliwości najnowszych wersji obu tych języków.

Oba języki — HTML5 i CSS3 — bazują na swoich poprzednich wersjach, co oznacza, że ucząc się ich, Czytelnik będzie w stanie zrozumieć ich wcześniejsze wersje. W tekście książki zamieściłem notatki w jasny sposób informujące, który kod jest nowy oraz który może nie działać w starszych wersjach przeglądarek.

JAK DZIAŁA WWW?

Serwer WWW, na którym działa witryna, może być umieszczony w dowolnym miejscu świata. Aby określić jego położenie, przeglądarka najpierw łączy się z serwerem systemu nazw domen (DNS).



Rysunek zamieszczony na tej stronie pokazuje, że serwery WWW, na których działają odwiedzane przez nas witryny, mogą się znajdować na całym świecie. To serwer DNS informuje przeglądarkę, jak odnaleźć daną witrynę.

- Użytkownik z Marsylii odwiedza stronę *sony.jp* na serwerze w Tokio.
- Użytkownik z Nowego Jorku odwiedza stronę *google.com* na serwerze w San Francisco.
- Użytkownik ze Sztokholmu odwiedza stronę *quantas.com.au* na serwerze w Sydney.
- Użytkownik z Vancouver odwiedza stronę *airindia.in* na serwerze w Bangalore.

Rysunek zamieszczony na następnej stronie pokazuje, co się stanie, gdy internauta z Anglii spróbuje odwiedzić stronę muzeum w Luwrze, dostępną pod adresem *www.louvre.fr*. Najpierw przeglądarka użytkownika mieszkającego w Cambridge kontaktuje się z serwerem DNS w Londynie. Serwer ten przekazuje jej położenie serwera, na którym działa witryna i który jest zlokalizowany w Paryżu.



1

STRUKTURA

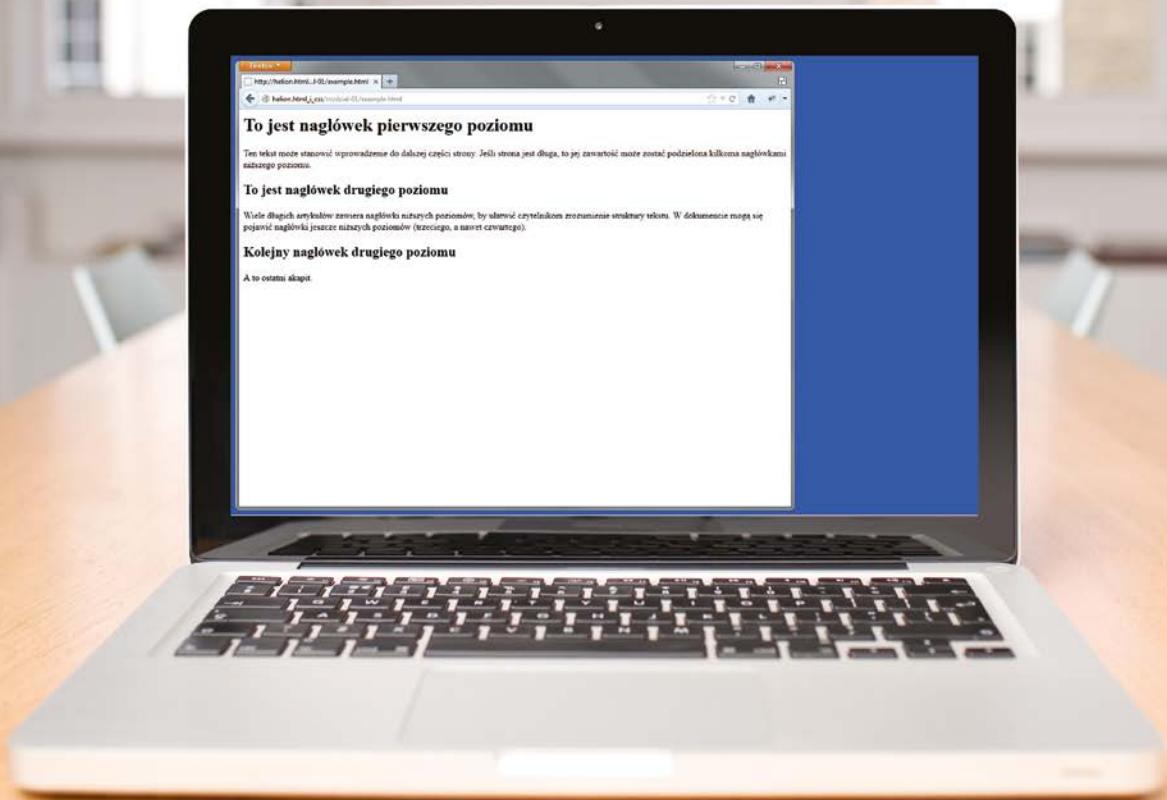
- ▶ Prezentacja struktury.
- ▶ Nauka kodu znacznikowego.
- ▶ Znaczniki i elementy.

Każdego dnia stykamy się z przeróżnymi dokumentami: gazetami, formularzami ubezpieczeniowymi, katalogami sklepów... Tę listę można by dłucho rozwijać.

Wiele stron WWW pełni funkcję elektronicznych wersji tych dokumentów. Gazety prezentują te same artykuły w wersjach elektronicznych i wydaniach papierowych, ubezpieczenia można kupować przez internet, a sklepy udostępniają internetowe katalogi i mechanizmy pozwalające na zakupy w sieci.

We wszystkich rodzajach dokumentów struktura jest ważnym czynnikiem ułatwiającym odbiorcy poruszanie się po ich zawartości oraz zrozumienie informacji przekazywanych przez autorów. A zatem by nauczyć się pisania stron WWW, trzeba poznać sposoby określania ich struktury. Oto co zrobimy w tym rozdziale:

- Dowiemy się, w jaki sposób język HTML opisuje strukturę strony WWW.
- Zobaczmy, jak dodawać znaczniki lub elementy do dokumentów.
- Napiszemy pierwszą stronę WWW.



SPOSOBY STOSOWANIA STRUKTURY PRZY TWORZENIU STRON WWW

Przyjrzyjmy się artykułom publikowanym w gazetach: każdy ma nagłówek, tekst oraz, być może, jakieś obrazy. Jeśli artykuł jest długi, mogą się w nim pojawić dodatkowe nagłówki dzielące go na mniejsze fragmenty oraz cytaty wyróżnione z tekstu. Struktura pomaga czytelnikom zrozumieć treści publikowane w gazetach.

Struktura artykułów publikowanych na stronach WWW jest bardzo podobna (choć mogą się w nich także pojawiać klipy dźwiękowe oraz wideo). Pokazano to na rysunku z prawej strony — widać na nim zdjęcie gazety oraz ten sam artykuł na stronie WWW czasopisma.

A teraz wyobraźmy sobie dokument innego typu — formularz ubezpieczeniowy. Jego poszczególne części są zazwyczaj poprzedzone odrębnymi nagłówkami, a każda z nich zawiera listę pytań oraz obszary przeznaczone na odpowiedzi lub pola wyboru, które można zaznaczać. Także w tym przypadku struktura formularza na stronie WWW jest bardzo podobna.

Read more on
MediaGuardian.co.uk

Digital economy or bust
Part 3: In which the team turn up the volume with inside track on The X Factor - and get a glimpse of the future

Coming up this week
Monday: Shortlists for Student Media Awards announced
Wednesday to Friday: Coverage of the RTS Cambridge Convention

Interview Rio Caraeff

Vevo revolutionary

Universal's former mobile chief is leading the music industry's fight to shake up online video. He reveals his frustration with MTV, and says why no one need own music if his site succeeds. Interview by **Mark Sweeney**

If Rio Caraeff succeeds, perhaps only diehard fans will need to own music. His online music video site, part-owned by the two largest record labels, aims to have the same impact as MTV and to be an answer to YouTube. Chuck those goals in with that of making the industry less dependent on the purchase of recordings, and for Caraeff there is clearly plenty to do.

Caraeff, a youthful chief executive of Vevo - launched in late 2009 with the backing of three of the four major groups - Sony Music, Universal Music and EMI, which have since joined the consortium with a roll-out starting in the UK and continental Europe. "See, music and sports are the only entertainment categories on the planet that people can buy and sell to global audiences at the scale of billions of people," he says. "I'm in the business of connecting billions of people to music," he muses, starting to sound like a guru.

With global CD sales plummeting by \$1.5bn last year, Caraeff's mission is clear.

"We would like to have created a new digital MTV," he says. "The industry felt it was necessary. If MTV was doing a great job paying royalties, if YouTube wasn't there would be no need for us. We have tried to get ourselves to be responsible for our own destiny. We can't sit back and say 'I hope Apple or whoever figures this out'."

Caraeff's relationship with Google, the owner of the world's largest video-sharing platform YouTube, is clearly a sticking point. "YouTube is a parasite" and Sir Martin Sorrell described it as a "frenemy". Despite the cumulative relationship with the two major players in the digital space, Caraeff prefers to characterise Vevo's dealings with YouTube as "symbolic", although "declaration of independence" would be more accurate.

"We said 'let's figure out how to work with them,'" he explains. "There are no duplicate content laws in the US. YouTube has more views than ours, the official versions are only available from us. They don't threaten us. YouTube is a place where people can upload their video in the world, we're not trying to compete." Caraeff points out that 50% of Vevo's traffic comes from YouTube search, and adds: "We're not trying to compete with videos that users might like to watch that appear on the side of the YouTube when that appears when a user is viewing clips."

Free access

Vevo's business model is all about providing music videos that fans can access free. Caraeff adds: "It's not a portal in another way; it's consumers as an alternative to owning songs." He believes the future is access, not ownership, not that's as it is today - he says: "It's not that we sell people music, our customers are not the small amount of people that want to buy music. We are about providing access; it is the only way to monetise for us, for me, including the question is, how do you do that and make money?"

Which raises the question of how well Vevo is actually doing. Caraeff doesn't want to say, but it's too much commercially but says it is already making "hundreds of millions of dollars" in revenue, although there are hosting costs to pay for that. Half of the revenue comes from content owners - the label, artist or licensor - with the remainder being kept by Vevo or paid to partners such as YouTube. Vevo's original business plan - about 40% ahead to be precise - and is on track to achieve profitability "in the very early part of next year".

Yet there is a problem. Caraeff's business is dependent on advertising, and he is frustrated by the low rates that companies pay to run campaigns on the site. His conclusion is that advertising must treat content as inferior and that Vevo's role is to "own" the prime content and then be able to position it as a premium product. In the end, the premium equivalent of BSkyB owning Premier League football.

"The audience that loves music is vast and promising; it should be treated as



Video vexations ... Rio Caraeff says: 'If MTV was doing a

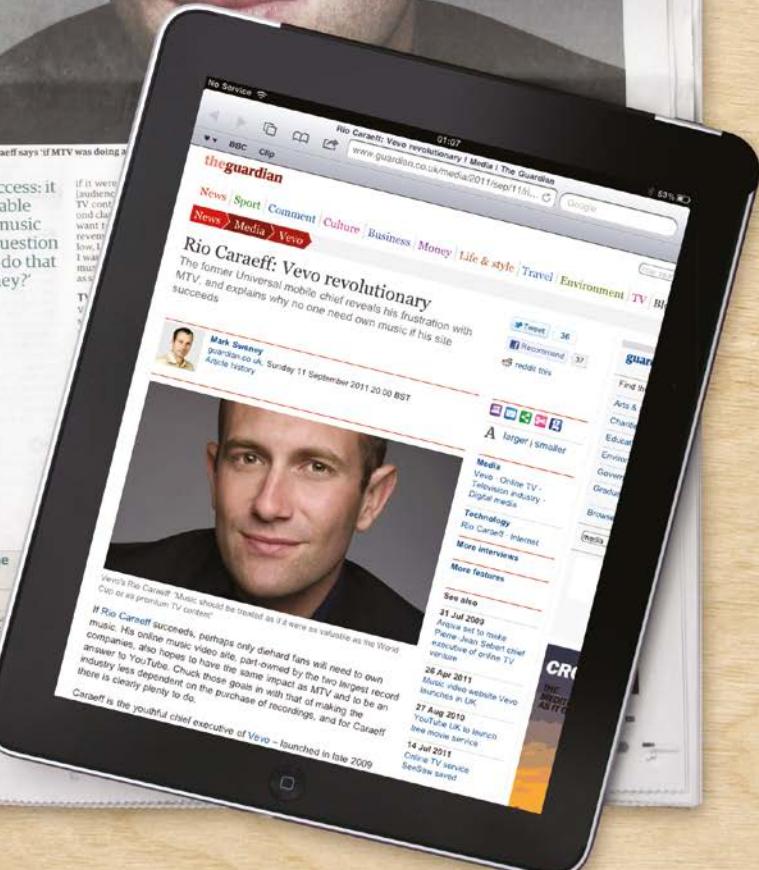
We are about access; it is the only scalable model for the music industry; the question is, how do you do that and make money?

Curriculum vitae

Age 36

Education Did not go to university because "I started my first company when I was 18"

Career 2004 vice-president, Pictures 2005 general manager, Pictures 2006 managing director, Unilever, responsible and new technology, Vevo



OKREŚLANIE STRUKTURY DOKUMENTÓW PROGRAMU WORD

Zastosowanie nagłówków różnych poziomów w dokumencie zazwyczaj odpowiada hierarchii zamieszczonych w nim informacji. Dokument może się zaczynać np. od dużego nagłówka, po którym jest umieszczone wprowadzenie oraz najważniejsze informacje.

Informacje mogą być rozwijane w sekcjach umieszczonych pod nagłówkami niższego poziomu, znajdującymi się w dalszej części strony. W razie tworzenia dokumentu w programie Word strukturę tekstu można także określić, oddzielając od siebie jego poszczególne fragmenty. Każdy temat może być umieszczony w osobnym akapicie; każda sekcja może mieć nagłówki opisujący jej treść.

Z prawej strony widać prosty dokument wyświetlony w programie Word. Widoczna z boku rozwijalna lista zawiera różnorodne rodzaje stylów, takich jak nagłówki różnych poziomów. Osoby korzystające z tego programu mogą określać style także przy użyciu paska narzędzi lub specjalnej palety.

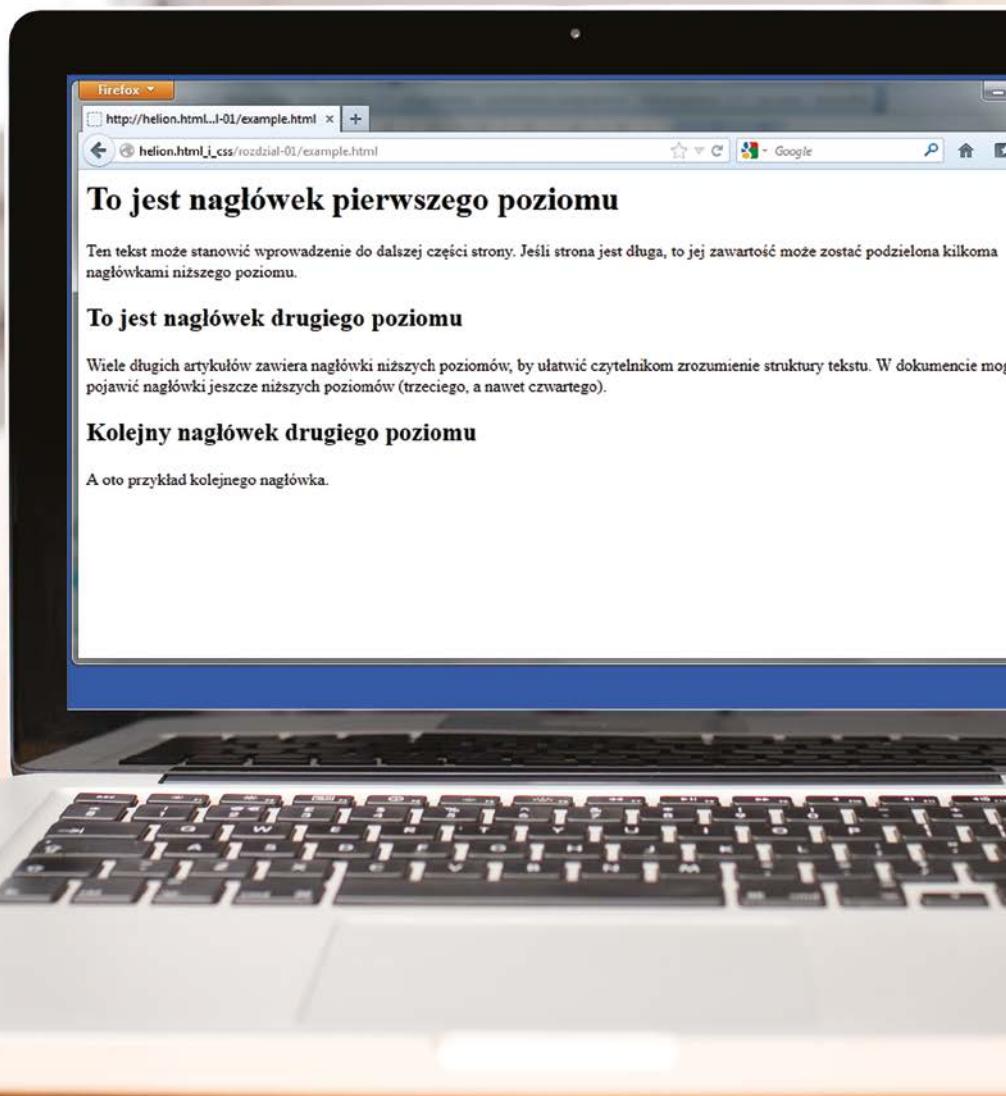
Nagłówek 1
Nagłówek 2
Nagłówek 3
Normalny
Więcej...

Nagłówek pierwszego poziomu
Innować wprowadzenie do dalszej części strony. Jeśli strona jest dłużna, to jej można podzielić na kilka części, aby łatwiej było ją przewieźć.

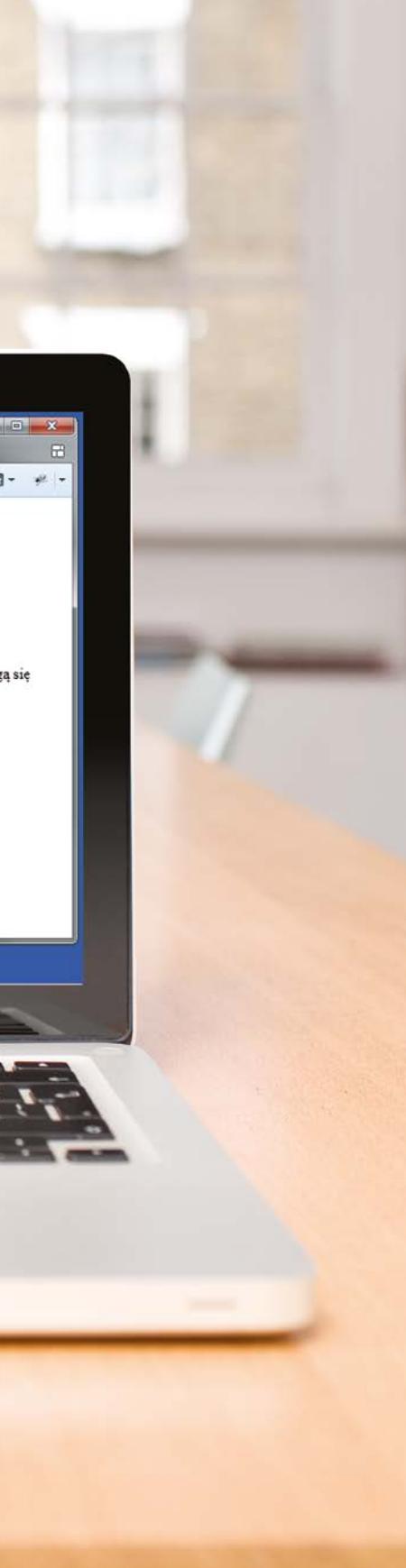
Nagłówek drugiego poziomu
Wiele długich artykułów zawiera nagłówki niższych poziomów, aby ułatwić czytelnikom zrozumienie struktury tekstu. W dokumencie mogą się pojawić nagłówki jeszcze niższych poziomów (trzeciego, a nawet czwartego).

Kolejny nagłówek drugiego poziomu
A oto przykład kolejnego nagłówka.

Na poprzedniej stronie została przedstawiona struktura określona w dokumencie programu Word. W identyczny sposób używamy struktury podczas tworzenia stron WWW.



STRUKTURĘ STRON OPISUJE KOD HTML



W oknie przeglądarki można zobaczyć stronę WWW o takiej samej treści, jaką miał dokument programu Word przedstawiony na stronie 17. Aby opisać strukturę strony WWW, do słów stanowiących jej zawartość dodawane są specjalne kody.

Poniżej widać kod HTML przykładowej strony. Na razie nie musimy się przejmować jego znaczeniem — na następnej stronie będziemy go dokładniej analizować. Należy zwrócić uwagę na to, że kod HTML został wyróżniony niebieskim kolorem, a tekst widoczny na ekranie jest pokazany na czarno.

```
<html>
  <body>
    <h1>To jest nagłówek pierwszego poziomu</h1>
    <p>Ten tekst może stanowić wprowadzenie do dalszej części strony. Jeśli strona jest długa, to jej zawartość może zostać podzielona kilkoma nagłówkami niższego poziomu.<p>
    <h2>To jest nagłówek drugiego poziomu</h2>
    <p>Wiele długich artykułów zawiera nagłówki niższych poziomów, by ułatwić czytelnikom zrozumienie struktury tekstu. W dokumencie mogą się pojawić nagłówki jeszcze niższych poziomów (trzeciego, a nawet czwartego).<p>
    <h2>Kolejny nagłówek drugiego poziomu</h2>
    <p>A oto przykład kolejnego nagłówka.<p>
  </body>
</html>
```

Kod HTML (wyróżniony niebieskim kolorem) składa się ze znaków zapisanych pomiędzy nawiasami kątowymi — sekwencje te tworzą tzw. **elementy** HTML. Elementy te zazwyczaj składają się z dwóch **znaczników**: otwierającego oraz zamykającego. (Wewnątrz znacznika zamykającego umieszczany jest dodatkowo znak ulkoznika). Każdy z tych znaczników HTML przekazuje przeglądarce WWW pewne informacje o swojej zawartości umieszczonej pomiędzy jego znacznikiem otwierającym i zamykającym.

HTML UŻYWA ELEMENTÓW DO OPISYWANIA STRUKTURY STRON

Przeanalizujmy nieco dokładniej kod HTML przedstawiony na poprzedniej stronie. Zostało w nim umieszczonych kilka różnych elementów. Każdy z nich składa się ze znacznika otwierającego oraz zamykającego.

KOD

The diagram illustrates the structure of an HTML document. It starts with the root element <html>, which contains the <body> element. The <body> element contains several <p> elements and two <h2> elements. The first <h2> element contains a nested <p> element. The second <h2> element also contains a nested <p> element. The <body> element is enclosed in a red box, and the entire <html> element is enclosed in a green box.

```
<html>
  <body>
    <h1>To jest nagłówek pierwszego poziomu</h1>
    <p>Ten tekst może stanowić wprowadzenie do dalszej części strony. Jeśli strona jest długa, to jej zawartość może zostać podzielona kilkoma nagłówkami niższego poziomu.<p>
    <h2>To jest nagłówek drugiego poziomu</h2>
    <p>Wiele długich artykułów zawiera nagłówki niższych poziomów, by ułatwić czytelnikom zrozumienie struktury tekstu. W dokumencie mogą się pojawić nagłówki jeszcze niższych poziomów (trzeciego, a nawet czwartego).<p>
    <h2>Kolejny nagłówek drugiego poziomu</h2>
    <p>A oto przykład kolejnego nagłówka.<p>
  </body>
</html>
```

Elementy pełnią funkcję pojemników. Niosą ze sobą informację o zawartości umieszczonej pomiędzy ich znacznikami otwierającymi i zamykającymi.

OPIS

..... Otwierający znacznik **<html>** informuje, że wszystko, co się znajdzie pomiędzy nim i zamykającym znacznikiem **</html>**, jest kodem strony.

..... Otwierający znacznik **<body>** informuje, że wszystko pomiędzy nim i zamykającym znacznikiem **</body>** powinno zostać wyświetcone w głównym oknie przeglądarki.

Słowa umieszczone pomiędzy znacznikami **<h1>** i **</h1>** stanowią najważniejszy nagłówek strony.

Akapity tekstu są zapisywane pomiędzy znacznikami **<p>** i **</p>**.

Słowa umieszczone pomiędzy znacznikami **<h2>** i **</h2>** stanowią nagłówek drugiego poziomu.

A tutaj znajduje się kolejny akapit, także zapisany pomiędzy znacznikami **<p>** i **</p>**.

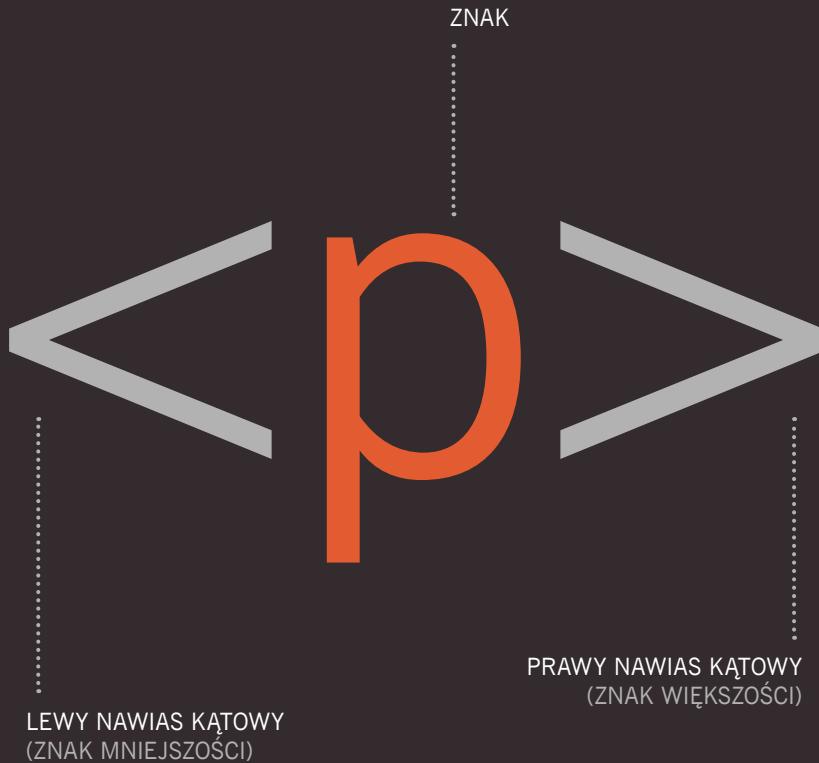
I kolejny nagłówek drugiego poziomu, zapisany pomiędzy znacznikami **<h2>** i **</h2>**.

Kolejny akapit zapisany między znacznikami **<p>** i **</p>**.

..... Zamykający znacznik **</body>** określa koniec zawartości, która powinna zostać wyświetlona wewnętrz okna przeglądarki.

..... Zamykający znacznik **</html>** określa koniec kodu HTML.

DOKŁADNIEJSZA ANALIZA ZNACZNIKÓW

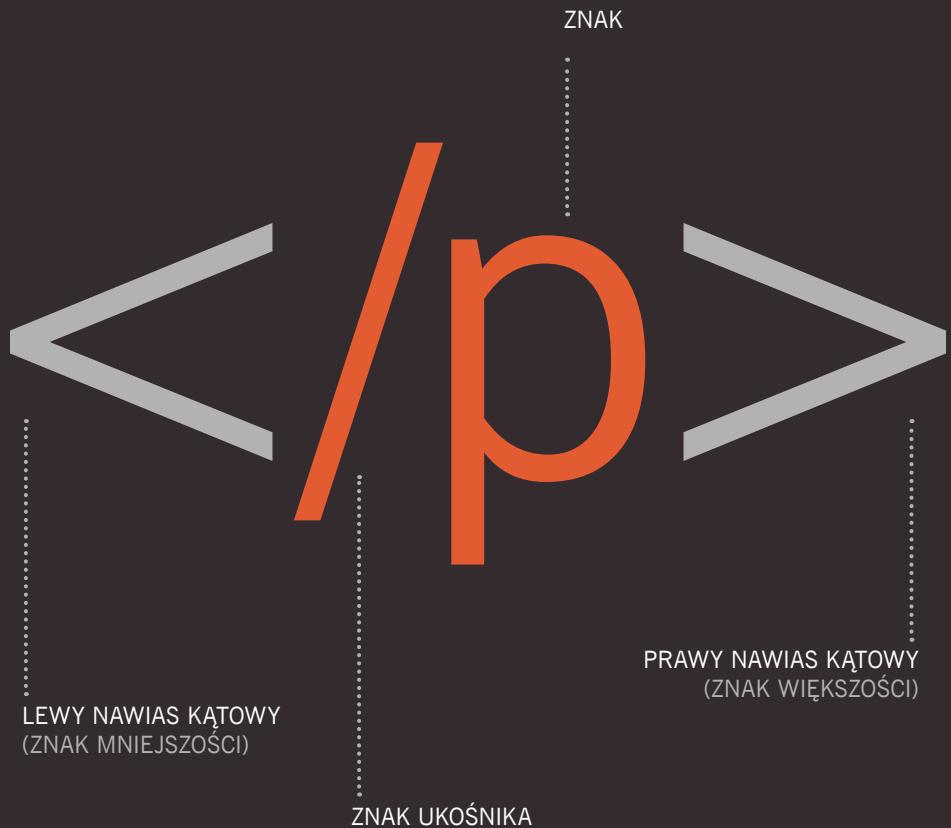


ZNACZNIK OTWIERAJĄCY

Znaki zapisane pomiędzy nawiasami kątowymi określają przeznaczenie znacznika.

W tym przypadku litera „p” oznacza akapit (ang. *paragraph*).

W znaczniku zamykającym, bezpośrednio za znakiem otwierającego nawiasu kątowego, zapisywany jest znak ukośnika.



ZNACZNIK ZAMYKAJĄCY

Terminy „znacznik” i „element” są często stosowane zamiennie.

Precyjnie rzecz ujmując, element składa się ze znacznika otwierającego i znacznika

zamykającego oraz zawartości umieszczonej pomiędzy tymi znacznikami.

ATRYBUTY PRZEKAZUJĄ DODATKOWE INFORMACJE O ELEMENTACH

Atrybuty dostarczają dodatkowych informacji o zawartości elementów. Są umieszczane w znaczniku otwierającym elementu i składają się z dwóch części: **nazwy** oraz **wartości**, oddzielonych od siebie znakiem równości.



Nazwa atrybutu określa, jakiego rodzaju dodatkowe informacje dotyczące treści elementu chcemy podać. Należy ją zapisywać małymi literami.

Wartość to informacje lub ustawienia określane przez atrybut. Wartość tę należy zapisywać wewnątrz cudzysłowów. Różne atrybuty mogą mieć różne wartości.

W powyższym przykładzie atrybut o nazwie `lang` został zastosowany w celu określenia języka, w jakim zapisano zawartość elementu. Wartość tego atrybutu wskazuje, że tekst umieszczony wewnątrz elementu został zapisany w języku polskim.

Język HTML5 pozwala pisać nazwy atrybutów wielkimi literami i pomijać znaki cudzysłou przy podawaniu wartości atrybutów, niemniej nie jest to zalecane.



W przeważającej większości dany atrybut może być umieszczony wyłącznie w wybranych elementach, istnieje jednak kilka atrybutów (takich jak `lang`), które mogą występować w dowolnych elementach języka HTML.

W większości przypadków wartości atrybutów są predefiniowane lub zapisywane w ścisłe określony sposób. Dopuszczalne wartości zostaną przedstawione podczas opisywania konkretnych atrybutów.

Wartość atrybutu `lang` jest skróconym, rozumianym przez wszystkie przeglądarki sposobem określania języka, w jakim została zapisana treść elementu.

BODY, HEAD I TITLE

<body>

Element <body> pojawił się już w pierwszym przykładzie prezentowanym w tej książce. Wszystko, co zostanie umieszczone wewnątrz niego, będzie widoczne w głównym oknie przeglądarki.

<head>

Bardzo często przed elementem <body> widnieje element <head>. Zawiera on informacje dotyczące strony (w odróżnieniu od informacji wyświetlanego w głównej części okna przeglądarki, zaznaczonej na niebiesko na rysunku na następnej stronie). Zazwyczaj wewnątrz elementu <head> umieszczany jest element <title>.

<title>

Zawartość elementu <title> zwykle jest wyświetlana albo na samej górze okna przeglądarki, nad polem służącym do podawania adresów URL stron, które chcemy oglądać, albo na karcie prezentującej stronę (jeśli przeglądarka udostępnia karty i umożliwia jednoczesne przeglądanie większej liczby stron).

/rozdział-01/body-head-title.html

HTML

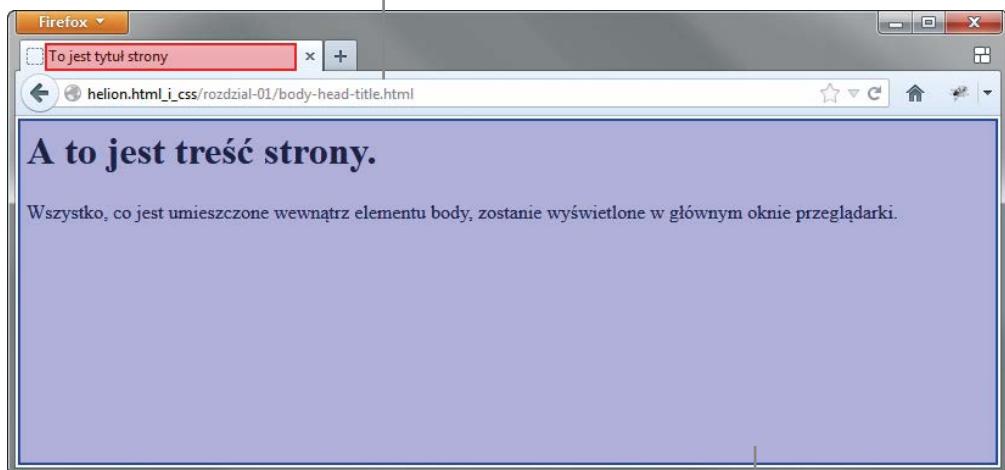
```
<html>
  <head>
    <title>To jest tytuł strony</title>
  </head>
  <body>
    <h1>A to jest treść strony.</h1>
    <p>Wszystko, co jest umieszczone wewnątrz elementu
       body, zostanie wyświetcone w głównym oknie
       przeglądarki.</p>
  </body>
</html>
```

WYNIK

A to jest treść strony.

Wszystko, co jest umieszczone wewnątrz elementu body, zostanie wyświetcone w głównym oknie przeglądarki.

Cały tekst umieszczony wewnątrz znaczników <title> zostanie wyświetlony na pasku tytułowym na samej górze okna przeglądarki lub na kartach. Na tym rysunku tytuł został zaznaczony czerwoną ramką.



Wszystko, co jest umieszczone wewnątrz elementu body, zostanie wyświetlone w głównym oknie przeglądarki.

Wszystko, co jest umieszczone pomiędzy znacznikami <body>, zostanie wyświetlone w głównym oknie przeglądarki. Na tym rysunku zawartość ta została zaznaczona niebieską ramką i niebieskim tłem.

Być może Czytelnik wie, że HTML to skrót od angielskiego *HyperText Markup Language* (hipertekstowy język znaczników). Słowo „hipertekstowy” odnosi się do tego, że HTML pozwala tworzyć łańcza dające

użytkownikom możliwość szybkiego i łatwego przenoszenia się pomiędzy różnymi stronami. Język znaczników umożliwia opisywanie tekstu, a opisy te dostarczają dodatkowych informacji o znaczeniu treści dokumentu. Przeanalizujmy

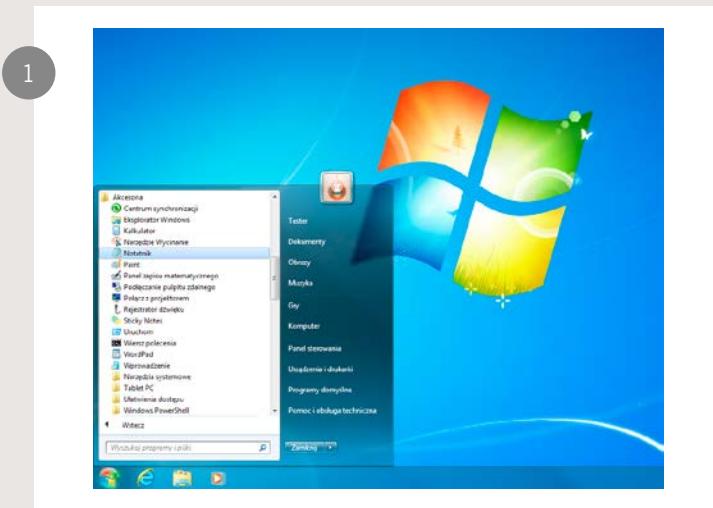
przedstawioną wcześniej przykładową stronę WWW: do jej początkowej zawartości, którą chcieliśmy wyświetlić, dodaliśmy fragmenty kodu, a przeglądarka używa ich do jej prawidłowego wyświetlania. A zatem te dodatkowe fragmenty to znaczniki.

TWORZENIE STRON WWW NA KOMPUTERACH PC

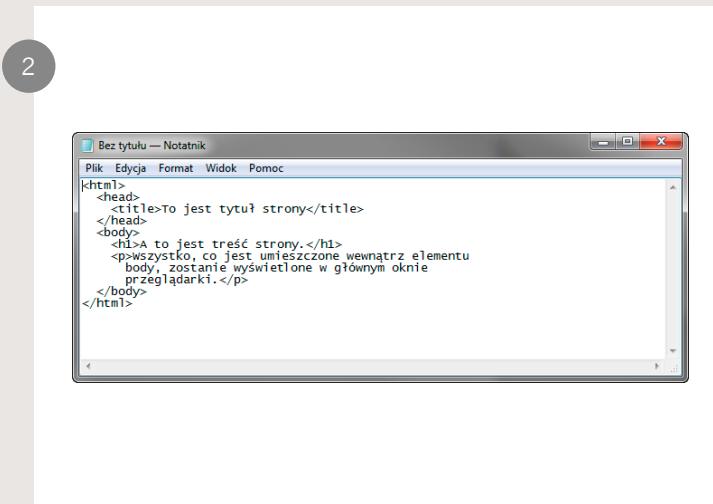
Aby stworzyć nową stronę WWW na komputerze PC z systemem Windows, należy uruchomić program Notatnik. Można go znaleźć po wybraniu następujących opcji:

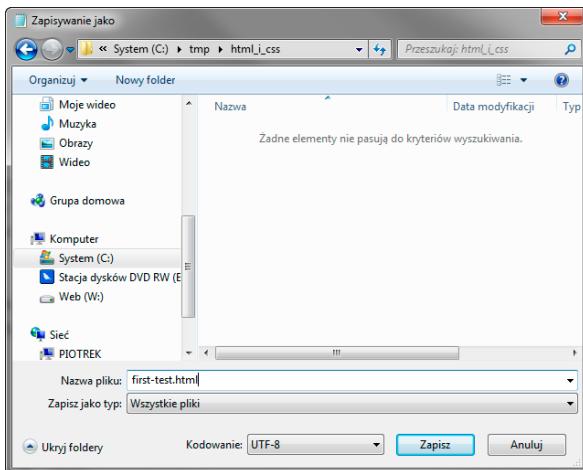
*Start/Wszystkie programy/
Akcesoria/Notatnik*

Można także rozważyć pobranie darmowego edytora o nazwie Notepad++ ze strony notepad-plus-plus.org.



Wpisz kod widoczny z prawej strony.

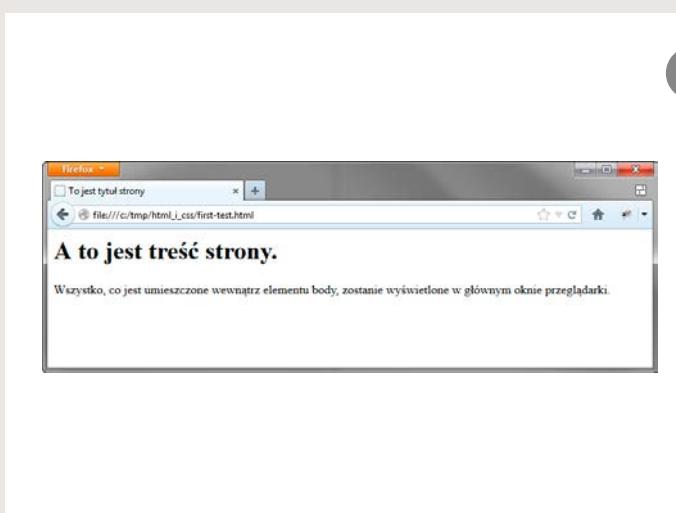




3

Z menu głównego należy wybrać opcję *Plik*, a następnie *Zapisz jako...*. Plik warto zapisać w takim miejscu, by później łatwo go odnaleźć. Można także stworzyć katalog, do którego trafią wszystkie przykłady tworzone w trakcie lektury niniejszej książki.

Kolejną czynnością będzie zapisanie pliku pod nazwą *first-test.html*. Należy się przy tym upewnić, że na rozwijalnej liście *Zapisz jako typ* została wybrana opcja *Wszystkie pliki*.



4

Teraz można uruchomić przeglądarkę, a po jej wyświetleniu wybrać z menu głównego opcję *Plik/Otwórz plik*. W wyświetlonym oknie dialogowym należy odszukać i zaznaczyć zapisany wcześniej plik, a następnie kliknąć przycisk *Otwórz*. Uzyskany wynik powinien przypominać okno przeglądarki przedstawione na zrzucie obok.

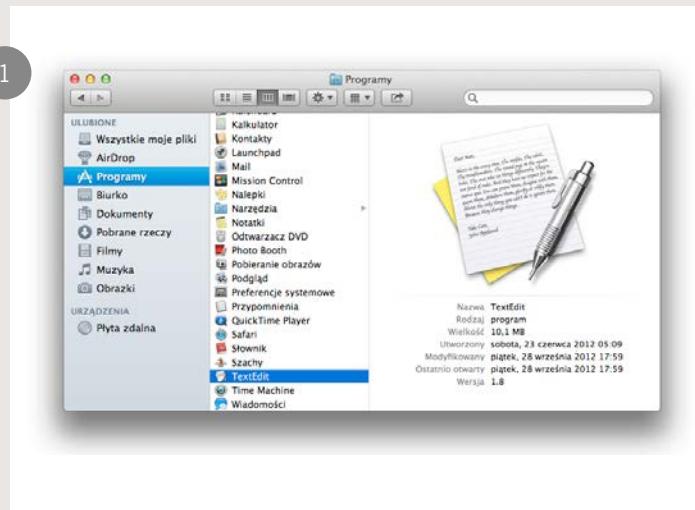
Jeśli strona nie jest podobna, to należy odszukać utworzony wcześniej plik i upewnić się, że ma on rozszerzenie *.html* (jeśli ma rozszerzenie *.txt*, należy ponownie przejść do Notatnika i jeszcze raz zapisać, podając jego nazwę w cudzysłowach: *"first-test.html"*).

TWORZENIE STRON WWW NA KOMPUTERACH MAC

Aby utworzyć swoją pierwszą stronę na komputerze Mac, należy uruchomić program TextEdit. Powinieneś znajdować w katalogu *Programy*.

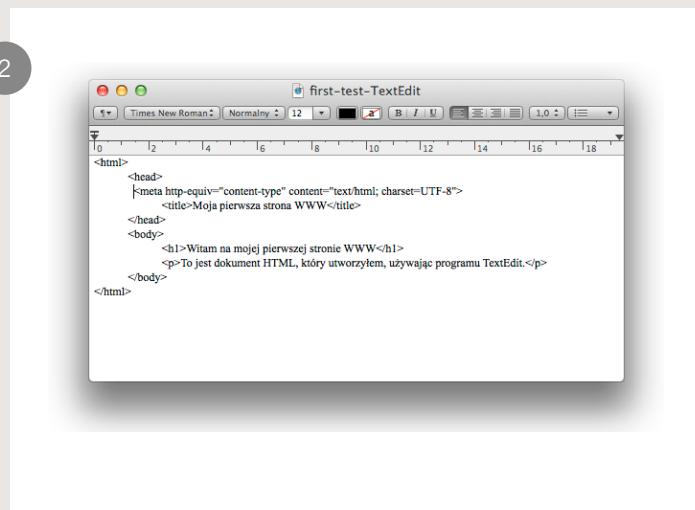
Można także rozważyć pobranie darmowego edytora TextWrangler służącego do tworzenia stron WWW. Można go znaleźć na stronie barebones.com.

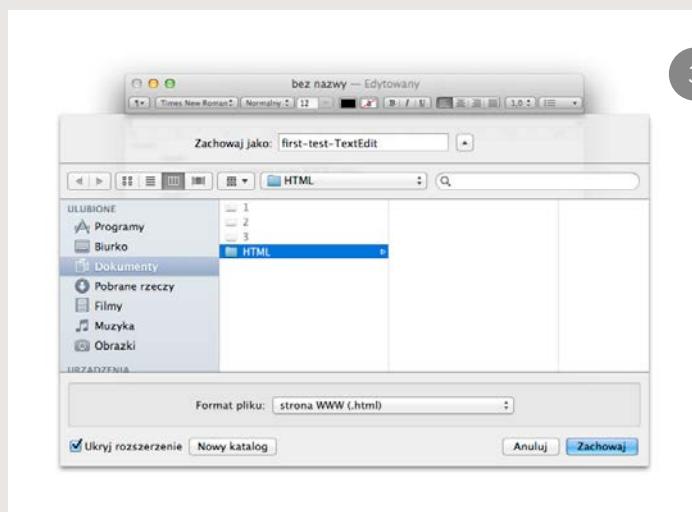
1



W edytorze należy wpisać kod przedstawiony z prawej strony.

2





Teraz należy przejść do menu *Plik* i wybrać opcję *Zachowaj jako...*. Plik należy zapisać w miejscu, w którym go później znajdziemy.

Można rozważyć utworzenie katalogu, w którym będą umieszczane wszystkie przykłady tworzone podczas lektury tej książki. Zapisywanej plikowi należy nadać nazwę *first-testTextEdit.html*.

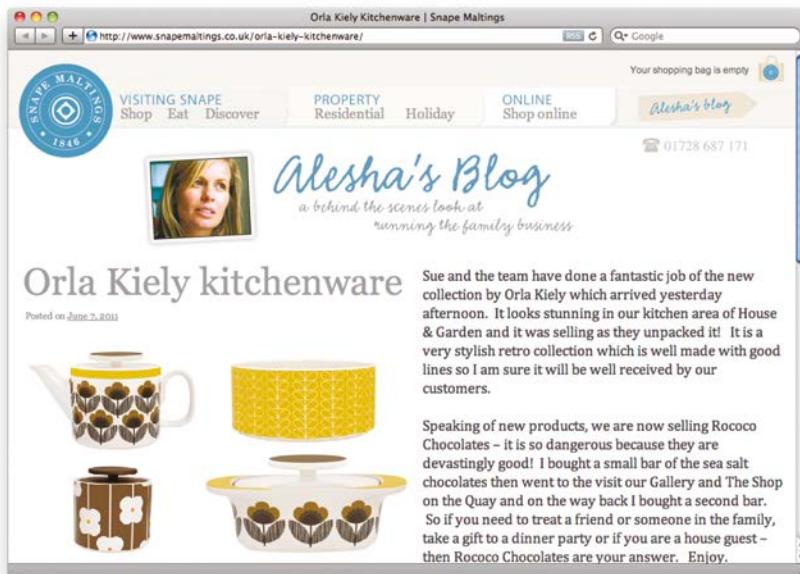
Jeśli na ekranie pojawi się okienko dialogowe dające możliwość wyboru rozszerzenia, z którym zostanie zapisany plik, to należy w nim kliknąć przycisk *Użyj .html*.



Kolejną czynnością będzie uruchomienie przeglądarki i wybranie z jej menu głównego opcji *Plik* oraz *Otwórz*. Następnie należy odszukać i zaznaczyć zapisany wcześniej plik oraz kliknąć przycisk *Otwórz*. Wyświetlone wyniki powinny przypominać te przedstawione z lewej strony.

Jeśli wyświetlona strona wygląda inaczej, to zapewne konieczne będzie wprowadzenie zmian w ustawieniach programu *TextEdit*. W tym celu z menu głównego programu należy wybrać opcję *Preferencje...* Następnie na karcie preferencji *Otwórz/Zachowaj* należy zaznaczyć pole *Wyświetlaj pliki HTML jako kod HTML zamiast tekstu sformatowanego*; po czym można ponownie spróbować zapisać plik.

PISANIE STRON W SYSTEMIE ZARZĄDZANIA TREŚCIĄ



W przypadku korzystania z jakiegoś systemu zarządzania treścią, platformy blogowej lub aplikacji do handlu elektronicznego w celu utworzenia nowych treści koniecznie będzie zalogowanie się do specjalnej części administracyjnej. Narzędzia dostępne w tych częściach administracyjnych zazwyczaj pozwalają na edycję jedynie fragmentów, a nie całych stron, dlatego rzadko kiedy zobaczymy tam elementy <html>, <head> lub <body>.

Jeśli spojrzeć na system zarządzania treścią przedstawiony na kolejnej stronie, można zobaczyć pole

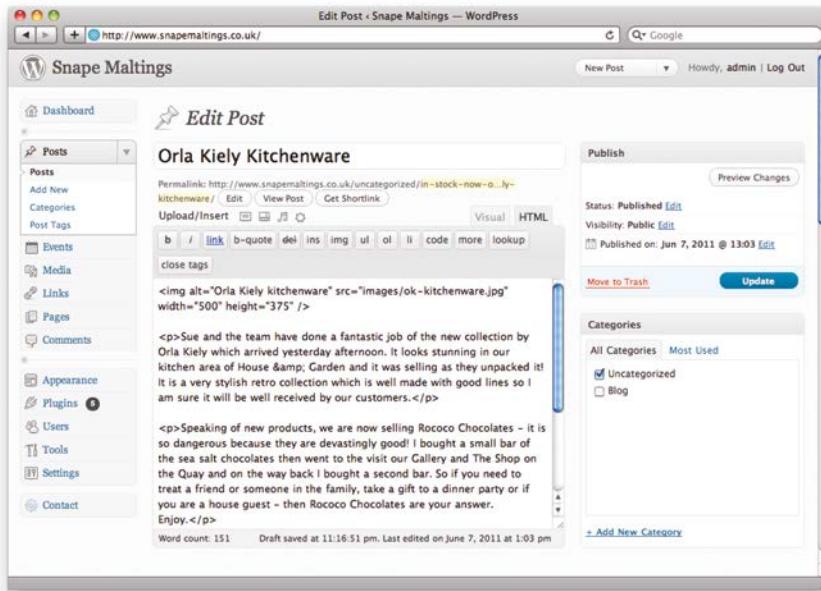
przeznaczone do podania tytułu strony, pole służące do wpisania głównej treści artykułu i kolejne — do określenia daty publikacji oraz wyboru sekcji witryny, w której artykuł powinien się pojawić.

W przypadku sklepów internetowych dostępne będą zapewne pola pozwalające na podanie nazwy produktu, zamieszczenie jego opisu, ceny oraz liczby dostępnych sztuk.

W tym przypadku tworzenie treści wygląda tak dlatego, że aplikacje te używają jednego „szablonu” określającego wygląd wszystkich stron danej sekcji witryny. (Sklep internetowy może np. korzystać

z tego samego szablonu do prezentacji wszystkich dostępnych towarów). Podawane przez nas informacje są następnie wyświetlane w tym szablonie.

Takie rozwiązanie ma tę zaletę, że nawet osoby, które nie wiedzą, jak tworzyć strony WWW, mogą dodawać do nich informacje. Oprócz tego czasami można wprowadzać zmiany w sposobie prezentacji szablonu, zostaną one automatycznie uwzględnione na wszystkich stronach, na których jest on używany. Jeśli sklep internetowy oferuje 1000 różnych produktów, w takim przypadku zmiana



jednego szablonu jest znacznie prostsza niż modyfikowanie stron poszczególnych produktów. W takich systemach, gdy udostępniana jest możliwość edycji większych fragmentów tekstu, jak najnowsze doniesienia prasowe, wpisy na blogach lub opisy towarów w sklepach internetowych, często stosowane są specjalne edytory tekstów.

Takie internetowe edytory tekstów mają zazwyczaj kontrolki odpowiadające tym dostępnym w normalnych edytorech i zapewniające możliwość stosowania różnych stylów, dodawania odnośników lub wstawiania obrazków. W sposób niewidoczny dla użytkownika edytory te

dodają do tekstu kod HTML, taki sam jak ten, który przedstawiliśmy w przykładach zamieszczonych wcześniej w tym rozdziale. Wiele spośród tych edytorów ma także opcję pozwalającą na wyświetlenie (oraz edycję) wygenerowanego kodu HTML.

Kiedy już się dowiemy, jak rozumieć oraz tworzyć taki kod, uzyskamy większą kontrolę nad wyglądem tych fragmentów witryny, które możemy edytować.

W przykładzie przedstawionym na powyższym rysunku widać, że edytor udostępnia odrebną zakładkę prezentującą wygląd treści wprowadzonych przez użytkownika oraz reprezentujący je kod HTML.

W innych systemach może być dostępny przycisk (często prezentujący nawiasy kątowe) umożliwiający wyświetlenie kodu źródłowego.

Niektóre systemy zarządzania treścią udostępniają także narzędzia pozwalające na edycję szablonów. Jeśli zdecydujemy się z nich skorzystać, warto zapoznać się z dokumentacją konkretnego systemu, ponieważ w każdym z nich szablony tworzy się w inny sposób. Podczas edycji szablonów należy zachować ostrożność, gdyż jeśli przez przypadek usuniemy niewłaściwy fragment kodu lub dodamy coś w nieodpowiednim miejscu, cała witryna może przestać działać.

ANALIZA INNYCH WITRYN WWW

W czasach, gdy sieć dopiero zaczynała zdobywać popularność, najczęstszym sposobem poznawania języka HTML oraz odkrywania różnych sztuczek i technik była analiza kodu istniejących stron.

Aktualnie dostępnych jest znacznie więcej książek oraz internetowych samouczków poświęconych językowi HTML, niemniej jednak wciąż można analizować kod przesyłany do przeglądarki przez serwery WWW. Aby się przekonać, jak to działa, wystarczy otworzyć w przeglądarce plik `view-source.html` umieszczony w katalogu z przykładami do pierwszego rozdziału książki.

Po wyświetleniu wybranej strony należy poszukać w menu przeglądarki opcji *Widok* i wybrać z niego *Źródło* lub *Pokaż źródło strony*. (Te nazwy mogą się różnić w zależności od używanej przeglądarki).

Po wybraniu tej opcji na ekranie powinno się wyświetlić nowe okno prezentujące kod źródłowy oglądanej strony.

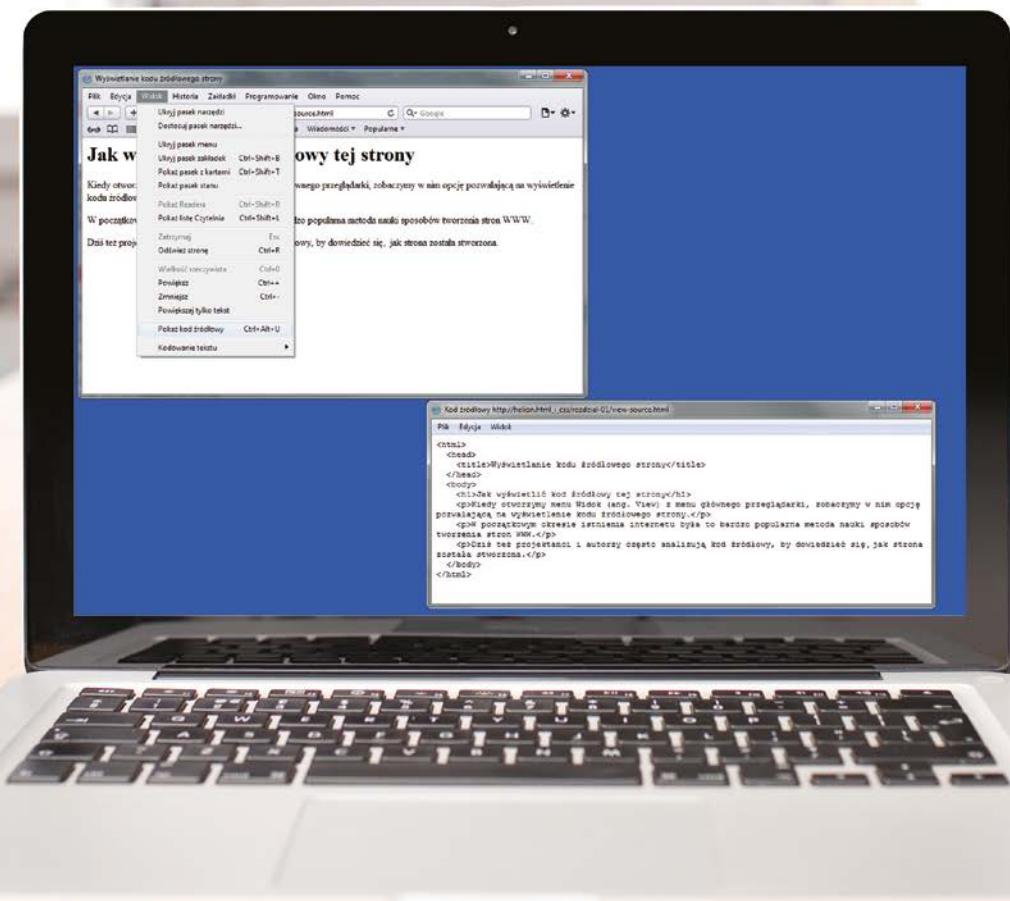
Efekty można zobaczyć na zrzucie przedstawionym na następnej stronie. W górnym oknie widoczna jest strona WWW wyświetlona w przeglądarce, a u dołu — jej kod źródłowy.

Początkowo kod analizowanych stron WWW może się wydawać złożony, jednak nie należy się tym zniechęcać. Po przeczytaniu następnego rozdziału tej książki Czytelnik będzie w stanie go zrozumieć.

Wszystkie przykłady prezentowane w tej książce są dostępne na poświęconej jej stronie WWW — www.htmlandcssbook.com¹ — a opisaną tu prostą technikę można zastosować do sprawdzenia, jak jest stworzony każdy z nich.

Oprócz tego można pobrać plik zawierający wszystkie przykłady zamieszczone w książce, jest on dostępny na serwerze FTP wydawnictwa Helion — [ftp://ftp.helion.pl/przykłady/htcsww.zip](ftp://ftp.helion.pl/przykladы/htcsww.zip).

¹ Trzeba jednak pamiętać, że są to oryginalne przykłady w języku angielskim — przyp. tłum.



PODSUMOWANIE

STRUKTURA

- ▶ Strony HTML są dokumentami tekstowymi.
- ▶ HTML używa znaczników (znaków zapisanych wewnętrz nawiasów kątowych), by nadawać dodatkowe znaczenie tekstem, które są umieszczane wewnętrz nich.
- ▶ Znaczniki często są nazywane elementami.
- ▶ Znaczniki zazwyczaj występują parami. Znacznik otwierający oznacza początek pewnego fragmentu zawartości, natomiast znacznik zamykający — jego koniec.
- ▶ W znacznikach otwierających mogą być umieszczane atrybuty przekazujące dodatkowe informacje o zawartości elementów.
- ▶ Atrybuty składają się z nazwy oraz wartości.
- ▶ Aby poznać język HTML, należy się dowiedzieć, jakie znaczniki są dostępne, jakie jest ich przeznaczenie oraz gdzie można je umieszczać.

2

TEKST

- ▶ Nagłówki i akapity.
- ▶ Wytluszczenia, kursywa i emfaza.
- ▶ Znaczniki strukturalne i semantyczne.

Podczas tworzenia stron WWW do ich treści dodajemy znaczniki. Nadają one dodatkowe znaczenie treści strony i pozwalają przeglądarkom w odpowiedni sposób przedstawić jej strukturę.

W tym rozdziale skoncentrujemy się na tym, w jaki sposób dodawać znaczniki do tekstu wyświetlanego na stronie. A konkretnie zostaną opisane:

- **Znaczniki strukturalne:** elementy, których można używać do opisywania zarówno nagłówków, jak i akapitów.
- **Znaczniki semantyczne:** czyli takie, które dostarczają dodatkowych informacji, np. nadają większe znaczenie cytatowi zapisanemu w znakach cudzysłowu, akronimom itd.



NAGŁÓWKI

<h1>
<h2>
<h3>
<h4>
<h5>
<h6>

Język HTML udostępnia sześć „poziomów” nagłówków.

Nagłówki najwyższego, pierwszego, poziomu są tworzone przy użyciu elementu <h1>.

Do tworzenia nagłówków drugiego poziomu służy element <h2>.

Jeśli wewnątrz fragmentu strony poprzedzonego nagłówkiem <h2> znajdują się kolejne nagłówki niższego poziomu, to tworzy się je przy użyciu elementów <h3> itd.

Przeglądarki wyświetlają zawartość nagłówków czcionką o różnej wielkości. Zawartość nagłówków <h1> jest prezentowana największą czcionką, a nagłówków <h6> — najmniejszą. Konkretnie wielkości czcionek stosowanych

rozdział-02/headings.html

HTML

<h1>To jest nagłówek pierwszego poziomu</h1>
<h2>To jest nagłówek drugiego poziomu</h2>
<h3>To jest nagłówek trzeciego poziomu</h3>
<h4>To jest nagłówek czwartego poziomu</h4>
<h5>To jest nagłówek piątego poziomu</h5>
<h6>To jest nagłówek szóstego poziomu</h6>

WYNIK

To jest nagłówek pierwszego poziomu

To jest nagłówek drugiego poziomu

To jest nagłówek trzeciego poziomu

To jest nagłówek czwartego poziomu

To jest nagłówek piątego poziomu

To jest nagłówek szóstego poziomu

przez poszczególne przeglądarki mogą się nieznacznie różnić. Oprócz tego użytkownicy mogą zmieniać wielkość czcionek używanych w nagłówkach poszczególnych poziomów, modyfikując ustawienia przeglądarki. W dalszej części

książki, kiedy zajmiemy się arkuszami stylów — CSS — przekonamy się, w jaki sposób można kontrolować wielkość tekstu, jego kolor oraz czcionkę.

AKAPITY

HTML

rozdział-02/paragraphs.html

`<p>`Akapit zawiera jedno zdanie lub kilka zdań stanowiących jedną całość. Początek akapitu jest zawsze wyświetlany w nowym wierszu.

`<p>`Tekst jest łatwiejszy w odbiorze, jeśli zostanie podzielony na mniejsze fragmenty. Na przykład książka może się składać z rozdziałów. Zawartość rozdziału może być podzielona na fragmenty nagłówkami. A treść umieszczona pomiędzy nagłówkami może być podzielona na wiele akapitów.

WYNIK

Akapit zawiera jedno zdanie lub kilka zdań stanowiących jedną całość. Początek akapitu jest zawsze wyświetlany w nowym wierszu.

Tekst jest łatwiejszy w odbiorze, jeśli zostanie podzielony na mniejsze fragmenty. Na przykład książka może się składać z rozdziałów. Zawartość rozdziału może być podzielona na fragmenty nagłówkami. A treść umieszczona pomiędzy nagłówkami może być podzielona na wiele akapitów.

`<p>`

Aby utworzyć akapit tekstu, należy umieścić tworzące go słowa pomiędzy otwierającym znacznikiem `<p>` i zamykającym znacznikiem `</p>`.

Domyślnie przeglądarka wyświetlająca zawartość każdego akapitu zaczyna go od nowego wiersza i dodaje nieco wolnego miejsca pomiędzy nim a następnym akapitem.

POGRUBIENIE ORAZ KURSYWA

Jeśli umieścimy słowa pomiędzy znacznikami **** i ****, sprawimy, że będą one wyświetcone pogrubioną czcionką.

Element **** reprezentuje także te fragmenty tekstu, które mają być przedstawione w wizualnie odmienny sposób (np. kluczowe słowa akapitu), choć jego zastosowanie nie nadaje wyróżnianym słowom żadnego dodatkowego znaczenia.

Można także zająrzeć na stronę 50, gdzie zostały podane informacje na temat elementu ****.

rozdział-02/bold.html

HTML

```
<p>Oto w jaki sposób słowo zostanie pogrubione.</p>
<p>Na stronie opisującej produkt niektóre jego
kluczowe cechy mogą być pogrubione.</p>
```

WYNIK

Oto w jaki sposób słowo zostanie **pogrubione**.

Na stronie opisującej produkt niektóre jego **kluczowe cechy** mogą być pogrubione.

<i>

Jeśli umieścimy słowa pomiędzy znacznikami *<i>* i *</i>*, sprawimy, że zostaną one wyświetcone kursywą.

Element **<i>** reprezentuje także te fragmenty, które mają się odróżniać od otaczającego je tekstu, takie jak terminy techniczne, nazwy statków, słowa zapisane w innym języku, myśli oraz inne terminy, które normalnie zostałyby zapisane kursywą.

Można także zająrzeć na stronę 50, gdzie zostały podane informacje na temat elementu ****.

rozdział-02/italic.html

HTML

```
<p>Oto w jaki sposób można wyświetlić słowo
kursywą.</p>
<p>Oto ziemniak Solanum teberosum.</p>
<p>Kapitan Cook popłynął do Australii na swoim
statku Endeavour.</p>
```

WYNIK

Oto w jaki sposób można wyświetlić słowo *kursywą*.

Oto ziemniak *Solanum teberosum*.

Kapitan Cook popłynął do Australii na swoim statku *Endeavour*.

INDEKS GÓRNY I DOLNY

HTML

rozdział-02/superscript-and-subscript.html

`<p>Na początku września poznamy wzór`

`E = mc2</sup>.`

`</p>`

`<p>W pierwszym kwartale 2009 roku ilość CO2₂`
`w atmosferze wzrosła o 2 ppm.</p>`

WYNIK

Na początku września poznamy wzór $E = mc^2$.

W pierwszym kwartale 2009 roku ilość CO_2 w atmosferze wzrosła o 2 ppm.

<sup>

Element `<sup>` jest stosowany do oznaczania znaków, które powinny zostać wyświetcone jako indeksy górne, tak jak w niektórych pojęciach matematycznych (np. potęgowaniu: 2^2).

<sub>

Element `<sub>` służy do oznaczania znaków, które powinny zostać wyświetcone jako indeksy dolne. Powszechnie takie znaki są stosowane we wzorach chemicznych, np. H_2O .

ODSTĘPY

Aby ułatwić sobie tworzenie kodu stron WWW, ich autorzy często dodają do nich znaki odstępu lub umieszczają początek niektórych elementów w nowym wierszu.

Jeśli przeglądarka WWW napotka w kodzie strony dwa znaki (lub większą ich liczbę) odstępu umieszczone jeden za drugim, to wyświetli tylko jeden z nich. Podobnie gdy napotka znak nowego wiersza, wyświetli go jako odstęp. Działanie to nazywamy **pomijaniem odstępów**.

Ten sposób działania przeglądarki jest często wykorzystywany przez autorów stron do formatowania kodu źródłowego, tak by jego analiza była łatwiejsza.

rozdział-02/white-space.html

HTML

```
<p>Księzyc krąży wokół Ziemi.</p>
<p>Księzyc      krąży wokół Ziemi.</p>
<p>Księzyc krąży wokół Ziemi.</p>
```

WYNIK

Księzyc krąży wokół Ziemi.
Księzyc krąży wokół Ziemi.
Księzyc krąży wokół Ziemi.

PODZIAŁ WIERSZA I POZIOMA LINIA

HTML

rozdział-02/line-breaks.html

`<p>Ze względu
 na opadający kosmiczny pył

każdego dnia ciężar Ziemi powiększa się
o setki ton.</p>`

WYNIK

Ze względu
na opadający kosmiczny pył
każdego dnia ciężar Ziemi powiększa się o setki ton.

`
`

Jak już się przekonaliśmy, przeglądarka automatycznie wyświetla każdy akapit oraz nagłówek od nowego wiersza. Jeśli jednak chcemy zakończyć wiersz i wyświetlić dalszą część akapitu w nowym, możemy to zrobić, używając znacznika `
`.

HTML

rozdział-02/horizontal-rules.html

`<p>Wenus jest jedyną planetą obracającą się zgodnie
z ruchem wskazówek zegara.</p>`
`<hr />`
`<p>Jowisz jest większy od wszystkich pozostałych
planet Układu Słonecznego razem wziętych.</p>`

WYNIK

Wenus jest jedyną planetą obracającą się zgodnie z ruchem wskazówek zegara.

Jowisz jest większy od wszystkich pozostałych planet Układu Słonecznego razem wziętych.

`<hr />`

Aby utworzyć przerwę pomiędzy zagadnieniami — sygnalizującą np. przejście do innej kwestii w książce albo sceny w sztuce — można dodać poziomą kreskę tworzoną przy użyciu znacznika `<hr />`.

Istnieje kilka elementów, w których pomiędzy znacznikiem otwierającym i zamykającym nie umieszcza się żadnych słów. Są to tzw. **elementy puste**; zapisuje się je w nieco inny sposób.

Elementy puste zazwyczaj składają się z tylko jednego znacznika. W takich elementach przed zamykającym nawiasem kątowym często jest zapisywany znak odstępu oraz ukośnik. Niektórzy autorzy stron WWW pomijają te dodatkowe znaki, jednak warto wyrobić sobie nawyk ich stosowania.

EDYTORY WIZUALNE I PODGLĄD KODU

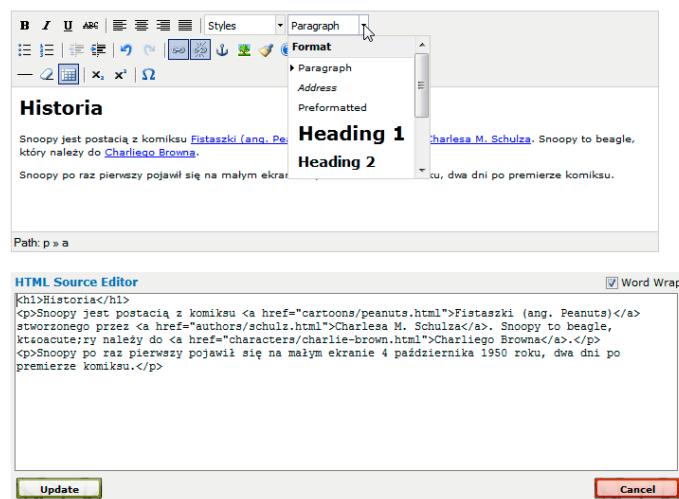
Systemy zarządzania treścią oraz edytory HTML, takie jak Dreamweaver, zazwyczaj udostępniają dwa sposoby prezentacji (tzw. widoki) tworzonych stron: pierwszym to edytor wizualny, a drugi — podgląd kodu źródłowego.

Edytory wizualne często przypominają procesory tekstu. Choć każdy z takich edytorów nieco się różni do pozostałych, większość z nich daje użytkownikom podobne możliwości kontroli sposobu prezentacji tekstu.

- Nagłówki tworzy się poprzez zaznaczenie tekstu i wybranie rodzaju nagłówka z rozwijalnej listy.
- Pogrubienie oraz kursywę tworzy się poprzez zaznaczenie wybranego fragmentu tekstu i kliknięcie przycisku *b* (ang. *bold* — pogrubienie) lub *i* (ang. *italic* — kursywa).
- Nowe akapity tworzy się, naciskając klawisz *Enter* lub *Return*.
- Aby utworzyć nowy wiersz, należy nacisnąć kombinację klawiszy *Shift+Enter*.
- Poziome linie tworzy się, naciskając przycisk przedstawiający poziomą kreskę.

Jeśli skopiujemy i wkleimy do edytora wizualnego tekst pochodzący z jakiegoś programu pozwalającego na edycję i formatowanie tekstu (takiego jak Word), może on dodać do wklejanego tekstu odpowiednie znaczniki. Aby uchronić się przed takim niepożądany efektem, należy wkleić tekst do prostego edytora tekstu (takiego jak program Notatnik w systemie Windows lub TextEdit na komputerach Mac), a następnie jeszcze raz go skopiować i wkleić do edytora wizualnego.

Widok kodu źródłowego przedstawia kod utworzony przez edytor wizualny; pozwala także na jego ręczną edycję lub dodawanie nowych treści. Widok ten często jest wyświetlany po naciśnięciu przycisku z napisem *HTML* lub z parą nawiasów kątowych. Edytor może dodawać odstępy do kodu źródłowego, by poprawić jego przejrzystość.



ZNACZNIKI SEMANTYCZNE

Istnieje pewna grupa elementów, których przeznaczeniem nie jest określanie struktury strony, lecz nadawanie jej dodatkowego znaczenia. Są to tzw. znaczniki semantyczne.

W dalszej części rozdziału poznamy kilka kolejnych elementów, które mogą nam pomóc podczas umieszczania tekstów na stronach WWW. Zostanie przedstawiony np. element `` pozwalający położyć większy nacisk na wyróżnione słowa oraz element `<blockquote>`, który służy do oznaczania bloków tekstu będących cytatami.

Przeglądarki wyświetlają zawartość tych elementów na różne sposoby, np. zawartość elementów `` jest przedstawiana kursywą, natomiast teksty umieszczone w elementach `<blockquote>` są wcięte. Niemniej elementów tych nie należy używać do modyfikacji wyglądu tekstów — służą one raczej do bardziej precyzyjnego opisywania zawartości stron.

Elementy te są stosowane dlatego, że inne programy, takie jak czytniki ekranowe lub wyszukiwarki, mogą korzystać z dostarczanych przez nie dodatkowych informacji. I tak głos generowany przez czytnik ekranowy może położyć większy nacisk na zawartość umieszczoną wewnątrz elementu ``, w razie zastosowania elementu `<blockquote>` wyszukiwarka może ten fakt odnotować.

WZMOCNIENIE I NACISK

Zastosowanie elementu `` oznacza, że jego zawartość ma większe znaczenie. Słowa umieszczone w tym elemencie można wypowiedzieć z dodatkowym naciskiem.

Domyślnie przeglądarki wyświetlają zawartość elementów `` pogrubioną czcionką.

rozdział-02/strong.html

HTML

```
<p><strong>Uwaga!</strong> W tej dzielnicy jest dużo kieszonkowców.</p>
<p>Ta zabawka składa się z niewielkich części i <strong>nie nadaje się dla dzieci poniżej piątego roku życia</strong>.</p>
```

WYNIK

Uwaga! W tej dzielnicy jest dużo kieszonkowców.

Ta zabawka składa się z niewielkich części i nie nadaje się dla dzieci poniżej piątego roku życia.

Element `` oznacza dodatkowy nacisk, który nieznacznie zmienia znaczenie zdania.

Domyślnie przeglądarki wyświetlają zawartość elementów `` kursywą.

rozdział-02/emphasis.html

HTML

```
<p><em>Myślę</em>, że Ewa była pierwsza.</p>
<p>Myślę, że <em>Ewa</em> była pierwsza.</p>
<p>Myślę, że Ewa była <em>pierwsza</em>.</p>
```

WYNIK

Myślę, że Ewa była pierwsza.

Myślę, że Ewa była pierwsza.

Myślę, że Ewa była pierwsza.

CYTATY

HTML

rozdział-02/quotations.html

```
<blockquote cite="http://pl.wikipedia.org/wiki/  
Kubuś_Puchatek">  
<p>Kubuś Puchatek usiadł sobie pod tym dębem,  
podparł głowę na łapkach i zaczął rozmyślać.</p>  
</blockquote>  
<p>A.A. Milne powiedział: <q>Niektórzy ludzie  
rozmawiają ze zwierzętami. Jednak niewielu ich  
słucha. I w tym tkwi problem</q>.</p>
```

WYNIK

Kubuś Puchatek usiadł sobie pod tym dębem, podparł głowę na łapkach i zaczął rozmyślać.

A.A. Milne powiedział: "Niektórzy ludzie rozmawiają ze zwierzętami. Jednak niewielu ich słucha. I w tym tkwi problem".

Oba te elementy udostępniają atrybut **cite** pozwalający określić źródło cytatu. Jego wartością powinien być adres URL strony zawierającej więcej informacji dotyczących źródła cytatu.

Dostępne są dwa elementy powszechnie używane do oznaczania cytatów: **<blockquote>** i **<q>**.

<blockquote>

Element **<blockquote>** służy do tworzenia dłuższych cytatów prezentowanych jako osobne akapity tekstu. Należy przy tym zwrócić uwagę, że wewnętrz nich i tak są umieszczane elementy **<p>**.

Wyświetlając zawartość elementu **<blockquote>**, przeglądarki zazwyczaj umieszczają wcięcie z ich lewej strony. Nie należy jednak używać go wyłącznie po to, by wciąć jakiś fragment tekstu — w takich przypadkach lepiej skorzystać z możliwości CSS.

<q>

Element **<q>** służy do oznaczania krótszych cytatów umieszczanych wewnętrz akapitów tekstu. Przeglądarki powinny wyświetlać zawartość tych elementów wewnętrz cudzysłowów, niemniej Internet Explorer tego nie robi — z tego powodu wiele osób unika stosowania tego elementu.

SKRÓTY I AKRONIMY

<abbr>

Jeśli w tekście ma się pojawić skrót lub akronim, to można go zapisać, używając elementu `<abbr>`. Do określenia rozwinięcia skrótu służy atrybut `title` znacznika otwierającego.

W języku HTML 4 dostępny był odrębny element `<acronym>` służący do oznaczania akronimów. Także w nim pełną postać akronimu można było podać, używając atrybutu `title`. W języku HTML5 zarówno skróty, jak i akronimy są tworzone przy użyciu elementu `<abbr>`.

rozdział-02/abbreviations.html

HTML

```
<p><abbr title="Profesor">Prof.</abbr> Stephen  
Hawking jest fizykiem teoretykiem  
i kosmologiem.</p>  
<p>W <acronym title="National Aeronautics and Space  
Administration">NASA</acronym> zajmują się  
wszelkimi zwariowanymi rzeczami związanymi  
z lotami kosmicznymi.</p>
```

WYNIK

Prof. Stephen Hawking jest fizykiem teoretykiem i kosmologiem.

W NASA zajmują się wszelkimi zwariowanymi rzeczami związanymi z lotami kosmicznymi.

National Aeronautics and Space
Administration

CYTATY I DEFINICJE

HTML

rozdział-02/citations.html

`<p>Książka <cite>Krótka historia czasu</cite> napisana przez Stephena Hawkinga została sprzedana na całym świecie w milionach egzemplarzy.</p>`

WYNIK

Czarna dziura to obszar kosmosu, z którego nic, nawet światło, nie jest w stanie się wydostać.

HTML

rozdział-02/definitions.html

`<p><dfn>Czarna dziura</dfn> to obszar kosmosu, z którego nic, nawet światło, nie jest w stanie się wydostać.</p>`

WYNIK

Czarna dziura to obszar kosmosu, z którego nic, nawet światło, nie jest w stanie wydostać.

<cite>

W przypadku odwoływania się do jakiejś pracy — książki, filmu lub opublikowanych wyników badań — w celu określenia źródła prezentowanych informacji można wykorzystać element `<cite>`.

W języku HTML5 elementu `<cite>` nie należy używać do podawania imion i nazwisk osób; w języku HTML 4 było to dozwolone, dlatego większość autorów stron wciąż tak robi.

Przeglądarki wyświetlają zawartość elementu `<cite>` kursywą.

<dfn>

Kiedy w dokumencie będziemy po raz pierwszy wyjaśniać znaczenie jakiegoś nowego terminu (pojęcia naukowego lub wyrażenia potocznego), określmy jego definicję.

Do podawania definicji stuży element `<dfn>`.

Niektóre przeglądarki wyświetlają zawartość elementu `<dfn>` kursywą, jednak programy Safari i Chrome nie zmieniają wyglądu tekstu stanowiącego definicję.

SZCZEGÓŁOWE INFORMACJE O AUTORZE

<address>

Element <address> ma bardzo precyzyjne określone zastosowanie: służy do podawania szczegółowych informacji kontaktowych autora strony.

W tym elemencie można (choć nie trzeba) podać adres pocztowy. Można w nim także umieścić numer telefonu oraz adres poczty elektronicznej.

Przeglądarki często wyświetlają zawartość elementów <address> kursywą.

Warto się zainteresować tzw. mikroformatem hCard służącym do dodawania informacji adresowych do kodu HTML strony.

Więcej w sieci

Więcej informacji na temat mikroformatu hCard można znaleźć na stronie poświęconej tej książce.

rozdział-02/address.html

HTML

```
<address>
  <p><a href="mailto:homer@simsonowie.org.pl">
    homer@simsonowie.org.pl</a></p>
  <p>al. Kreskówkowa 742, Warszawa</p>
</address>
```

WYNIK

homer@simsonowie.org.pl

al. Kreskówkowa 742, Warszawa

ZMIANY TREŚCI

HTML

rozdział-02/insert-and-delete.html

```
<p>To był <del>najgorszy</del> <ins>najlepszy</ins>  
pomysł, jaki kiedykolwiek przyszedł jej do głowy.</p>
```

<ins>

WYNIK

To był najgorszy najlepszy pomysł, jaki kiedykolwiek przyszedł jej do głowy.

Elementu `<ins>` można używać do przedstawiania treści, które zostały dodane do dokumentu, natomiast elementu `` — do oznaczania treści, które zostały z niego usunięte.

Zawartość elementów `<ins>` jest zazwyczaj podkreślana, natomiast zawartość elementów `` — przekreślana.

HTML

rozdział-02/strikethrough.html

```
<p>Nowoczesny laptop</p>  
<p><s>Kosztował 2599 zł</s></p>  
<p>A teraz tylko 1899 zł</p>
```

<s>

Element `<s>` oznacza informacje, które nie są już wystarczająco dokładne lub się zdezaktualizowały (których jednak nie należy usuwać ze strony).

Zazwyczaj zawartość elementów `<s>` jest wyświetlana przekreślona czcionką.

WYNIK

Nowoczesny laptop

~~Kosztował 2599 zł~~

A teraz tylko 1899 zł

W starszych wersjach języka HTML dostępny był element `<u>`, którego zawartość była wyświetlana podkreślona czcionką; obecnie element ten nie jest stosowany.



Firefox

Tekst helion.html_i_css/rozdzial-02/example.html

Opowieść w książce

Rozdział 1.

Mariola wyglądała przez okno przez prawie godzinę. Na biurku, pomiędzy magazynami *Nature*, *New Scientist* oraz różnymi innymi czasopismami naukowymi, których lektury wymagała jej praca, leżał podpiszczony egzemplarz *W drodze*. Była to jej ulubiona powieść jeszcze z czasów studiów. Im dłużej Mariola przebywała w tych czterech ścianach, tym bardziej czuła potrzebę odzyskania wolności.

Spędziła w tym pokoju ostatnich dziesięć lat, siedząc pod plakatem z cytatem Oscara Wilde'a: "Praca jest ucieczką dla osób, które nie mają niczego lepszego do roboty". Choć wielu uważało, że jej pionierskie prace nad odkryciem tajemnic DNA to niezwykłe osiągnięcie, Mariola wiedziała, że ma coś ciekawszego do zrobienia.

MacBook Pro

PRZYKŁAD

TEKST



To bardzo prosta strona HTML przedstawiająca zastosowanie poznanych znaczników.

Do zastosowanych tu znaczników strukturalnych należą: `<h1>`, `<h2>` i `<p>`.

Natomiast informacje semantyczne są zapewnione przez takie elementy jak `<cite>` i ``.

```
<html>
  <head>
    <title>Tekst</title>
  </head>
  <body>
    <h1>Opowieść w książce</h1>
    <h2>Rozdział 1.</h2>
    <p>Mariola wyglądała przez okno przez prawie godzinę.
      Na biurku, pomiędzy magazynami
      <i>Nature</i>, <i>New Scientist</i> oraz różnymi
      innymi czasopismami naukowymi, których lektury
      wymagała jej praca, leżała podniszczony egzemplarz
      <cite>W drodze</cite>. To była jej ulubiona
      powieść jeszcze z czasów studiów. Im dłużej
      Mariola przebywała w tych czterech ścianach, tym
      bardziej czuła potrzebę odzyskania wolności.</p>
    <p>Spędziła w tym pokoju ostatnich dziesięć
      lat, siedząc pod plakatem z cytatem
      z Oscara Wilde'a:
      <q>Praca jest ucieczką dla osób,
      które nie mają niczego lepszego do roboty</q>.
      Choć wielu uważało, że jej pionierskie prace
      nad odkryciem tajemnic <abbr
      title="Deoxyribonucleic acid">DNA</abbr> to
      niezwykłe osiągnięcie, Mariola
      <em>wiedziała</em>, że ma coś ciekawszego
      do zrobienia.</p>
  </body>
</html>
```


PODSUMOWANIE

TEKST

- ▶ Elementy HTML służą do opisu struktury stron WWW (nagłówków różnych stopni, akapitów itd.).
- ▶ Elementy dostarczają także informacji semantycznych (np. które fragmenty tekstu mają większe znaczenie, co jest rozwinięciem skrótu albo które fragmenty tekstu są cytatami).

3

LISTY

- ▶ Listy numerowane.
- ▶ Listy wypunktowane.
- ▶ Listy definicji.

Sytuacje, w których konieczne jest zastosowanie list, zdarzają się dosyć często. Język HTML udostępnia trzy różne rodzaje list:

- **Listy uporządkowane**, czyli takie, których kolejne elementy są oznaczone liczbami. Przykładem takiej listy może być procedura składająca się z kilku czynności, które należy wykonać w określonej kolejności, lub kontrakt, którego poszczególne punkty muszą być oznaczone liczbami.
- **Listy nieuporządkowane**, czyli takie, których poszczególne elementy są oznaczone znacznikami (a nie znakami określającymi porządek).
- **Listy definicji** tworzone przez zbiór par składających się z terminu i jego opisu.

Jajecznica

Potrawy z jajek należą do moich ulubionych. Oto przepis na wspaniałą, bogatą jajecznicę.

Składniki

- 2 jajka
- 1 łyżka masła
- 2 łyżki śmietany

Sposób przyrządzania

1. Rozepiąć jajko na półki na średnim ogień.
2. W rosolek delikatnie rozprowadzić jajka w kierunku.
3. Po rozepięciu masła dodać do narożnika jajka z kierunkiem.
4. Utrzywać lopatki, unierzą jajka z boczów; patrzeć i przeklinać je na jej środku przez około 20 s (tak jak byśmy robili ośle).
5. Szanować do uzyskania odpowiedniej konsystencji, pamiętać, że po zdobyciu z ognia jajecznica jeszcze lekko stwardniaje.

LISTY UPORZĄDKOWANE

Listy uporządkowane tworzy się przy użyciu elementu .

Każdy element listy jest umieszczany pomiędzy otwierającym znacznikiem a zamykającym znacznikiem . (Nazwa elementu — li — pochodzi od pierwszych liter angielskich słów *list item* — element listy).

Przeglądarki domyślnie wcinają zawartość list.

W niektórych przypadkach w elemencie można umieścić atrybut **type** umożliwiający określenie używanego sposobu numeracji (liczby, litery, liczby rzymskie itd.). Jednak lepszym rozwiązaniem jest skorzystanie z właściwości **list-style-type** CSS przedstawionej na stronach 332 – 334.

rozdział-03/ordered-lists.html

HTML

```
<ol>
  <li>Ziemniaki obrać i pokroić w ćwiartki.</li>
  <li>Wrzucić do wody i gotować do miękkości
    przez 15-20 min.</li>
  <li>Podgrzać mleko z masłem i gałką muszkatołową.
    </li>
  <li>Ziemniaki odcedzić i utłuc.</li>
  <li>Dolać do nich mleko z masłem i gałką,
    wymieszać, doprawić do smaku.</li>
</ol>
```

WYNIK

1. Ziemniaki obrać i pokroić w ćwiartki.
2. Wrzucić do wody i gotować do miękkości przez 15-20 min.
3. Podgrzać mleko z masłem i gałką muszkatołową.
4. Ziemniaki odcedzić i utłuc.
5. Dolać do nich mleko z masłem i gałką, wymieszać, doprawić do smaku.

LISTY NIEUPORZĄDKOWANE

HTML

rozdział-03/unordered-lists.html

```
<ul>
  <li>1 kg ziemniaków,</li>
  <li>100 ml mleka,</li>
  <li>50 g solonego masła,</li>
  <li>świeżo zmielona gałka muszkatołowa,</li>
  <li>sól i pieprz do smaku.</li>
</ul>
```

WYNIK

- 1 kg ziemniaków,
- 100 ml mleka,
- 50 g solonego masła,
- świeżo zmielona gałka muszkatołowa,
- sól i pieprz do smaku.

Listy nieuporządkowane są tworzone przy użyciu elementu ``.

Każdy element listy jest umieszczany pomiędzy otwierającym znacznikiem `` a zamykającym znacznikiem ``. (Nazwa elementu — `li` — pochodzi od pierwszych liter angielskich słów *list item* — element listy).

Przeglądarki domyślnie wcinają zawartość list.

W niektórych przypadkach w elemencie `` można umieścić atrybut `type` umożliwiający określenie typu używanych znaczników (kółka, kwadraty, romby itd.). Jednak lepszym rozwiązaniem jest skorzystanie z właściwości `list-style-type` CSS przedstawionej na stronach 332 – 334.

LISTY DEFINICJI

<dl>

Listy definicji tworzy się przy użyciu elementu `<dl>`. Zwykle zawierają one serię terminów i ich opisów.

Wewnątrz elementu `<dl>` zazwyczaj umieszczane są pary elementów `<dt>` i `<dd>`.

<dt>

Ten element służy do podania definiowanego terminu (ang. **definition term**).

<dd>

Tego elementu używa się do podania definicji.

Można znaleźć listy, w których dwa terminy są opisywane przy użyciu jednej definicji albo do jednego terminu podano dwa wyjaśnienia.

rozdział-03/definition-lists.html

HTML

```
<dl>
  <dt>Sashimi</dt>
  <dd>Pokrojona surowa ryba podawana
      z dodatkami, takimi jak tarta rzodkiew daikon,
      korzeń imbiru, wasabi oraz sos sojowy.</dd>
  <dt>Waga</dt>
  <dd>Urządzenie służące do precyzyjnego odmierzania
      składników.</dd>
  <dd>Potoczne określenie ciężaru ciała.</dd>
  <dt>Scamorze</dt>
  <dt>Scamorzo</dt>
  <dd>Włoski ser wyrabiany zazwyczaj z pełnotłustego
      mleka krowiego (choć tradycyjnie do jego
      wyrobu było używane mleko bawole).</dd>
</dl>
```

WYNIK

Sashimi

Pokrojona surowa ryba podawana z dodatkami, takimi jak tarta rzodkiew daikon, korzeń imbiru, wasabi oraz sos sojowy.

Waga

Urządzenie służące do precyzyjnego odmierzania składników.
Potoczne określenie ciężaru ciała.

Scamorze

Scamorzo

Włoski ser wyrabiany zazwyczaj z pełnotłustego mleka krowiego (choć tradycyjnie do jego wyrobu było używane mleko bawole).

LISTY ZAGNIEŻDŻONE

HTML

rozdział-03/nested-lists.html

```
<ul>
  <li>Musy</li>
  <li>Ciasta
    <ul>
      <li>Rogaliki</li>
      <li>Szarlotka</li>
      <li>Karpatka</li>
      <li>Sernik wiedeński</li>
    </ul>
  </li>
  <li>Tarty</li>
</ul>
```

Wewnątrz elementu `` można umieścić kolejną listę i w ten sposób utworzyć listę zagnieżdżoną.

Przeglądarki, wyświetlając listy zagnieżdżone, wcinają je bardziej niż listy, w których zostały umieszczone. Co więcej, w zagnieżdżonych listach nieuporządkowanych przeglądarka zapewne zmieni rodzaj używanych znaczników.

WYNIK

- Musy
- Ciasta
 - Rogaliki
 - Szarlotka
 - Karpatka
 - Sernik wiedeński
- Tarty

Firefox

helion.html_i_css/rozdzial-03/example-03.html

Jajecznica

Potrawy z jajek należą do moich ulubionych. Oto przepis na wspaniałą, bogatą jajecznicę.

Składniki

- 2 jajka
- 1 łyżka masła
- 2 łyżki śmietany

Sposób przyrządzania

1. Roztopić masło na patelni na średnim ogniu.
2. W rondelku delikatnie rozemieszać jajka ze śmietaną.
3. Po roztopieniu masła dodać do niego jajka ze śmietaną.
4. Używając łyżki, usuwać jajka z brzegów patelni i przekladać je na jej środek przez około 20 s (tak jak byśmy robili omlet).
5. Smazyc do uzyskania odpowiedniej konsystencji, pamiętając, że jeszcze po zdjęciu z ognia jajecznica lekko stwardnieje.

PRZYKŁAD

LISTY

W poniższym przykładzie widać nagłówek poziomu pierwszego oraz umieszczony poniżej akapit tekstu stanowiącego wprowadzenie. Na stronie została umieszczona także lista nieuporządkowana opisująca składniki oraz lista uporządkowana opisująca sposób przyrządzenia potrawy.

```
<html>
  <head>
    <title>Listy</title>
  </head>
  <body>
    <h1>Jajecznica</h1>
    <p>Potrawy z jajek należą do moich ulubionych.  
      Oto przepis na wspaniałą, bogatą  
      jajecznicę.</p>
    <h2>Składniki</h2>
    <ul>
      <li>2 jajka</li>
      <li>1 łyżka masła</li>
      <li>2 łyżki śmietany</li>
    </ul>
    <h2>Sposób przyrządzenia</h2>
    <ol>
      <li>Roztopić masło na patelni na średnim  
          ogniu.</li>
      <li>W rondelku delikatnie rozmieszać jajka  
          ze śmietaną.</li>
      <li>Po roztopieniu masła dodać do niego  
          jajka ze śmietaną.</li>
      <li>Używając łyżki, usuwać jajka z brzegów  
          patelni i przekładać je na jej środek  
          przez około 20 s (tak jak byśmy  
          robili omlet).</li>
      <li>Smażyć do uzyskania odpowiedniej konsystencji,  
          pamiętając, że po zdjęciu z ognia  
          jajecznica jeszcze lekko stwardnieje.</li>
    </ol>
  </body>
</html>
```


PODSUMOWANIE

LISTY

- ▶ Język HTML udostępnia trzy rodzaje list: uporządkowane, nieuporządkowane oraz listy definicji.
- ▶ Do oznaczania elementów listy uporządkowanej używa się liczb.
- ▶ W listach nieuporządkowanych elementy są oznaczane przy użyciu znaczników.
- ▶ Listy definicji są używane do wyjaśniania terminologii.
- ▶ Listy można zagnieźdzać.

4

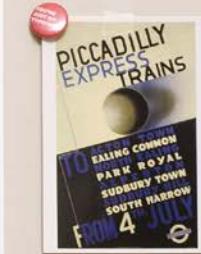
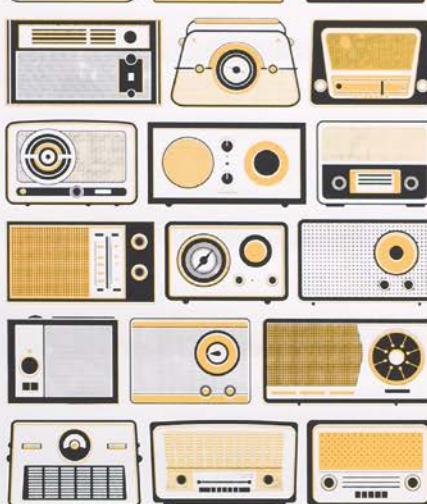
ŁĄCZA

- ▶ Tworzenie łączy pomiędzy stronami.
- ▶ Łącza do innych witryn.
- ▶ Łącza służące do wysyłania wiadomości poczty elektronicznej.

Łącza są jednym z kluczowych elementów tworzących WWW, gdyż pozwalają przenosić się pomiędzy stronami — zapewniając tym samym możliwość swobodnego poruszania się po zasobach sieci.

Powszechnie spotyka się następujące rodzaje łącz:•

- Łącza pomiędzy odrębnymi witrynami.
- Łącza pomiędzy różnymi stronami należącymi do tej samej witryny.
- Łącza pomiędzy różnymi fragmentami tej samej strony.
- Łącza wyświetlające nowe okna przeglądarki.
- Łącza, które pozwalają uruchomić klienta poczty elektronicznej i utworzyć nową wiadomość do podanej osoby.



Now style sheet
— update
— a very long

Then play
Then for fun

Festiwalowe

Kalendarium festiwalowe

Oto niektóre spośród tegorocznych festiwali filmowych, które nas interesują. Proszę się z nimi [skontaktować](#) w celu uzyskania dodatkowych informacji.

Wrzesień

XVII PRZEGŁAD FILMÓW GÓRSKICH
Lądek-Zdrój
18 - 23 września 2012 r.

Listopad

[20. Międzynarodowy Festiwal Sztuki Autorów Zdjęć Filmowych Plus Camerimage](#)
Bydgoszcz
24 listopada - 1 grudnia 2012 r.

[19. Międzynarodowy Festiwal Filmowy Etude & Animus](#)
Kraków
23 - 29 listopada 2012 r.

[10. Krakowski Festiwal Górska](#)
Kraków
7 - 9 grudnia 2012 r.

[O nas](#)

[Początek strony](#)



TWORZENIE ŁĄCZY

Łącza tworzy się przy użyciu elementu `<a>`. Użytkownik może kliknąć dowolną zawartość umieszczoną pomiędzy znacznikami `<a>` i ``. Strona docelowa, z którą chcemy się połączyć, jest określana za pomocą atrybutu `href`.



Tekst umieszczony pomiędzy otwierającym znacznikiem `<a>` i zamykającym znacznikiem `` jest nazywany tekstem łącza. Zawsze, o ile to możliwe, tekst ten powinien informować, gdzie użytkownik zostanie przeniesiony po kliknięciu łącza (nie należy zatem tworzyć łączy z lakonicznym „kliknij tutaj”). Poniżej przedstawiono przykład łącza do witryny IMDB utworzonego na poprzedniej stronie.

Wiele osób porusza się po witrynach, przeglądając ich zawartość w poszukiwaniu łączy. Przejrzysty tekst łącza ułatwia użytkownikowi odnalezienie poszukiwanych informacji. W ten sposób możemy poprawić jego doznanie związane z korzystaniem z aplikacji i sprawić, by spędził w niej więcej czasu. (Dodatkowo może to ułatwić przeglądanie witryny osobom korzystającym z czytników ekranowych).

Przed sformułowaniem odpowiedniego tekstu łącza należy się zastanowić, jakich słów mogą użyć inne osoby podczas poszukiwania strony, do której dane łącze ma prowadzić (np. zamiast „zatrzymać się w” można napisać „hotele w Krakowie”).



ŁĄCZA DO INNYCH WITRYN

<a>

Łącza są tworzone przy użyciu elementu `<a>` posiadającego atrybut `href`. Wartością tego atrybutu jest adres strony, którą chcemy wyświetlić, gdy użytkownik kliknie łącze.

Użytkownicy mogą kliknąć dowolną treść umieszczoną pomiędzy otwierającym znacznikiem `<a>` i zamkającym znacznikiem ``, w efekcie zostaną przeniesieni na stronę podaną w atrybutie `href`.

W przypadku tworzenia łączy prowadzących do innej witryny wartością atrybutu `href` musi być jej pełny adres internetowy, nazywany także **bezwzględnym** adresem URL.

Przeglądarki wyświetlają tekst łącza niebieską czcionką z podkreśleniem.

rozdział-04/linking-to-other-sites.html

HTML

```
<p>Recenzje i przeglądy filmów:<br><ul><li><a href="http://www.empireonline.com">Empire</a></li><li><a href="http://www.metacritic.com">Metacritic</a></li><li><a href="http://www.rottentomatoes.com">Rotten Tomatoes</a></li><li><a href="http://www.variety.com">Variety</a></li></ul></p>
```

WYNIK

Recenzje i przeglądy filmów:

- [Empire](http://www.empireonline.com)
- [Metacritic](http://www.metacritic.com)
- [Rotten Tomatoes](http://www.rottentomatoes.com)
- [Variety](http://www.variety.com)

Bezwzględne adresy URL

URL to skrót od angielskiego *Uniform Resource Locator* — ujednolicony lokalizator zasobów. Jest to adres, który wpisujemy w przeglądarce, by wyświetlić jakąś stronę.

Bezwzględne adresy URL rozpoczynają się od nazwy domeny, a w ich końcowej części może się znaleźć ścieżka do konkretnej strony. Jeśli ścieżka nie zostanie podana, przeglądarka wyświetli stronę główną danej witryny.

ŁĄCZA DO INNYCH STRON TEJ SAMEJ WITRYNY

HTML

rozdział-04/linking-to-other-pages.html

```
<p>
  <ul>
    <li><a href="index.html">Strona główna</a></li>
    <li><a href="about-us.html">O nas</a></li>
    <li><a href="movies.html">Filmy</a></li>
    <li><a href="contact.html">Kontakt</a></li>
  </ul>
</p>
```

WYNIK

- [Strona główna](#)
- [O nas](#)
- [Filmy](#)
- [Kontakt](#)

Względne adresy URL

Podczas tworzenia łączy do innych plików tej samej witryny można stosować względne adresy URL.

Przypominają one nieco skróconą postać adresów bezwzględnych — nie trzeba w nich podawać nazwy domeny.

Więcej informacji o względnych adresach URL znajduje się na stronach 82 – 83, istnieje bowiem kilka użytecznych, skrótowych sposobów zapisywania adresów do stron we własnej witrynie.

Względne adresy URL ułatwiają pracę podczas tworzenia stron WWW na własnym komputerze, gdyż pozwalają definiować łącza bez konieczności konfigurowania nazwy domeny oraz wykupywania usługi hostingowej.

<a>

Tworząc łącza prowadzące do innych stron tej samej witryny, w adresach URL nie trzeba podawać nazwy domeny. W tym przypadku można zastosować skrócony zapis, nazywany **względnym** adresem URL.

Jeśli wszystkie strony witryny są umieszczone w tym samym katalogu, to wartością atrybutu href może być nazwa pliku.

Jeśli inne strony witryny znajdują się w odrębnych katalogach, do określenia położenia strony docelowej względem bieżącej trzeba będzie użyć nieco bardziej złożonego formatu. Zostanie on przedstawiony na stronach 80 – 83.

Jeśli Czytelnik przyjrzy się kodowi stron *index.html* umieszczonej w poszczególnych katalogach z przykładami do tej książki, przekona się, że zawierają one właśnie takie — względne — adresy URL.

STRUKTURA KATALOGÓW

W przypadku dużych witryn WWW warto zorganizować ich zawartość, umieszczając w osobnych katalogach strony należące do różnych działów. Katalogi w witrynach WWW są czasami nazywane folderami.

STRUKTURA

Diagram przedstawiony z prawej strony pokazuje strukturę katalogów fikcyjnej witryny o nazwie *ExampleArts*, zawierających informacje i doniesienia związane z kulturą i rozrywką.

Główny katalog witryny nosi nazwę *korzeń* (ang. *root*). (W tym przykładzie katalog główny nosi nazwę *examplearts*). To właśnie w nim umieszczane są wszystkie pozostałe pliki i katalogi tworzące witrynę.

Każdy dział witryny został umieszczony w odrębnym katalogu; takie rozwiązanie ułatwia organizację plików.

W przypadku korzystania z systemu zarządzania treścią, oprogramowania dla blogerów lub systemu do prowadzenia handlu elektronicznego może się zdarzyć, że nie będzie odrębnych plików dla poszczególnych stron witryny.

POWIĄZANIA

Wzajemne powiązania pomiędzy plikami i katalogami w witrynie WWW są opisywane przy użyciu tej samej terminologii określającej relacje hierarchiczne.

Niektóre z tych powiązań zostały zaznaczone na rysunku przedstawionym z prawej strony.

Katalog *examplearts* jest nadzędny względem katalogów *movies*, *music* oraz *theater*. Z kolei trzy wspomniane wcześniej katalogi są podrzędne względem katalogu *examplearts*.

Zamiast tego systemy te często używają jednego szablonu do prezentacji stron konkretnego typu (przy czym typów tych może być kilka, np. doniesienia, wpisy na blogach, opisy produktów).

STRONY GŁÓWNE

Strona główna całej witryny oraz strony główne poszczególnych jej działów są pisane w języku HTML i zazwyczaj noszą nazwę *index.html*.

Serwery WWW są zazwyczaj konfigurowane w taki sposób, by zwracały stronę *index.html*, jeśli w żądaniu nie zostanie określona żadna nazwa pliku.

Dlatego jeśli w przeglądarce wpiszemy adres *examplearts.com*, serwer zwróci stronę *examplearts.com/index.html*, natomiast po wpisaniu adresu *examplearts.com/music* — stronę *examplearts.com/music/index.html*.

Poprzez zmianę pliku szablonu możemy zmienić wygląd wszystkich stron generowanych na jego podstawie. Należy jednak uważać, by zmieniać wyłącznie kod HTML, w przeciwnym razie modyfikacje mogą spowodować wystąpienie problemów.

KATALOG NADRZĘDNY

Katalog *examplearts* jest katalogiem nadrzędnym dla katalogu *music*.

KATALOG PODRZĘDNY

Katalog *music* jest katalogiem podrzędnym katalogu *examplearts*.

```
..... └── examplearts
      .....
```

- └── examplearts
 - index.html
 - └── images
 - logo.gif
 - └── movies
 - └── cinema
 - index.html
 - listings.html
 - reviews.html
 - └── dvd
 - index.html
 - reviews.html
 - index.html
 - └── music
 - index.html
 - listings.html
 - reviews.html
 - └── theater
 - index.html
 - listings.html
 - reviews.html

DALSZY KATALOG NADRZĘDNY

Katalog *examplearts* jest dalszym katalogiem nadrzędnym katalogu *dvd*.

DALSZY KATALOG PODRZĘDNY

Katalog *dvd* jest dalszym katalogiem podrzędnym katalogu *examplearts*.

Każda strona i każdy obraz w witrynie ma własny adres **URL** (ujednolicony lokalizator zasobów). Składa się on z nazwy domeny oraz umieszczonej za nim **ścieżki** do strony lub obrazu.

Ścieżka do strony głównej tej witryny ma postać: www.examplearts.com/index.html. Z kolei ścieżka do pliku z logo witryny może mieć następującą postać: www.examplearts.com/images/logo.gif.

Adresy URL są używane podczas tworzenia łączy prowadzących do innych stron oraz umieszczania obrazów w witrynach WWW. Na następnej stronie zostanie przedstawiony skrócony sposób tworzenia łączy do plików wchodzących w skład tej samej witryny.

Katalog główny zawiera:

- Plik o nazwie *index.html* będący stroną główną całej witryny.
- Odrębne katalogi dla działów witryny poświęconych filmom, muzyce oraz teatrowi.

Każdy podkatalog zawiera:

- Plik o nazwie *index.html* stanowiący stronę główną danego działu.
- Stronę z recenzjami o nazwie *reviews.html*.
- Stronę z listą zawartości o nazwie *listings.html* (z wyjątkiem działu filmów DVD).

Dział filmów zawiera:

- Katalog o nazwie *cinema*.
- Katalog o nazwie *DVD*.

WZGLĘDNE ADRESY URL

Względnych adresów URL można używać podczas tworzenia łączy do stron należących do tej samej witryny. Zapewniają one krótszy i prostszy sposób poinformowania przeglądarki, gdzie należy odnaleźć stronę docelową.

W łączach, które mają prowadzić do stron wchodzących w skład tej samej witryny, nie trzeba podawać nazwy domeny. Można skorzystać ze **względnych adresów URL** stanowiących skrócony sposób informowania przeglądarki o tym, gdzie, w stosunku do aktualnie przeglądanej strony, znajduje się strona docelowa.

To rozwijanie jest szczególnie użyteczne w przypadku tworzenia nowych witryn lub nauki języka HTML, gdyż pozwala na tworzenie łączy pomiędzy stronami przechowywanymi na lokalnym komputerze (jeszcze zanim uzyskamy własną domenę i prześlemy pliki do internetu).

Dodatkowo, jako że w takich łączach nie trzeba podawać nazwy domeny, umieszczanie ich w kodzie jest szybsze.

Jeśli wszystkie pliki tworzące witrynę znajdują się w jednym katalogu, w łączach wystarczy podać tylko nazwy plików.

Jeśli jednak zawartość witryny jest umieszczona w kilku odrębnych katalogach, trzeba poinstruować przeglądarkę, w jaki sposób może dostać się z *aktualnie* prezentowanej strony na *stronę docelową* łącza.

Dlatego przy tworzeniu dwóch łączy prowadzących z różnych stron witryny do tej samej strony może się zdarzyć, że trzeba będzie podać dwa różne względne adresy URL.

Także w takich łączach stosowane jest to samo nazewnictwo, które zostało przedstawione na poprzedniej stronie, w opisie struktury katalogów.

TYP ŁĄCZA WZGLĘDNEGO

PRZYKŁAD

(na podstawie schematu z poprzedniej strony)

TEN SAM KATALOG

Aby stworzyć łącze do pliku znajdującego się w tym samym katalogu, wystarczy podać jego nazwę (nie potrzeba nic więcej)

Oto łącze pozwalające przejść ze strony głównej działu muzycznego do recenzji utworów muzycznych:

`Recenzje`

KATALOG PODRZĘDNY

W przypadku łącza do plików z katalogu podrzędnego należy podać nazwę tego katalogu, znak ukośnika oraz nazwę pliku

Oto łącze prowadzące ze strony głównej witryny do listy utworów muzycznych:

`Listings`

KATALOG W KATALOGU PODRZĘDNYM

W tym przypadku należy podać nazwę katalogu podrzędnego, znak ukośnika, nazwę kolejnego katalogu umieszczonego wewnątrz katalogu podrzędnego, kolejny znak ukośnika i, w końcu, nazwę pliku

Oto łącze prowadzące ze strony głównej witryny do recenzji filmów DVD:

`Recenzje`

KATALOG NADRZĘDNY

Aby odwołać się do strony w katalogu nadzędnym względem tego, w którym znajduje się aktualna strona, należy użyć sekwencji znaków ../, a za nią podać nazwę strony docelowej

Oto łącze prowadzące ze strony recenzji w dziale muzycznym do strony głównej:

`Strona główna`

KATALOG NADRZĘDNY WZGLĘDEM

KATALOGU NADRZĘDNEGO

Należy dwukrotnie użyć sekwencji ../, by zaznaczyć, że należy przejść dwa poziomy wyżej (a nie tylko jeden), a następnie podać nazwę strony docelowej

Oto łącze prowadzące ze strony recenzji filmów DVD do strony głównej:

`Strona główna`

Po opublikowaniu witryny (czyli skopiowaniu jej zawartości na serwer WWW) można zastosować kilka innych sposobów tworzenia łączy, niedostępnych podczas pracy z plikami znajdującymi się na lokalnym komputerze.

Można np. podać jedynie nazwę katalogu podrzędnego bez określania konkretnego pliku, o który nam chodzi. W takim przypadku serwer WWW zazwyczaj spróbuje pobrać stronę główną danego działu witryny.

Użycie samego znaku ukośnika jako całego adresu URL spowoduje zwrócenie strony głównej całej witryny; natomiast znak ukośnika, po którym została podana nazwa pliku, zwróci wskazaną stronę, o ile znajduje się ona w katalogu głównym witryny.

ŁĄCZA Z ADRESAMI POCZTY ELEKTRONICZNEJ

mailto:

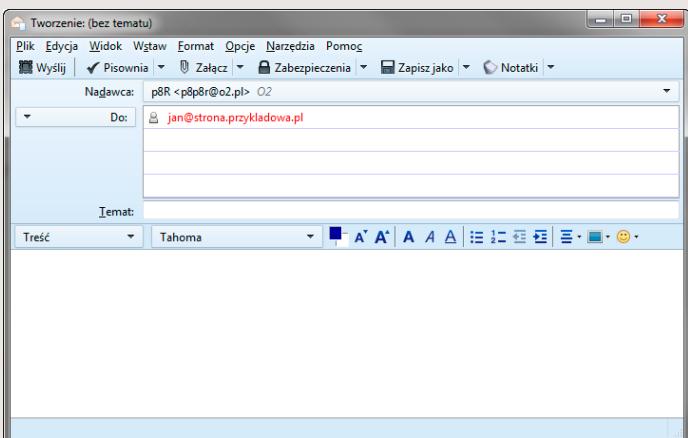
Elementu <a> można użyć, by utworzyć łącze, którego kliknięcie spowoduje uruchomienie programu do obsługi poczty elektronicznej i utworzenie wiadomości zaadresowanej do konkretnego odbiorcy. W tym przypadku wartość atrybutu href musi się zaczynać od ciągu znaków mailto:, po którym trzeba podać adres poczty elektronicznej, na jaki ma być wysłana wiadomość.

Jak pokazują zrzuty zamieszczone obok, takie łącze wygląda dokładnie tak samo jak każde inne, jednak po jego kliknięciu zostanie uruchomiony program do obsługi poczty elektronicznej, a w nim utworzone okno nowej wiadomości zaadresowanej do osoby podanej w łączu.

rozdział-04/email-link.html

HTML

```
<a href="mailto:jan@strona.przykładowa.pl">Wyślij e-mail do Jana</a>
```



OTWIERANIE STRON W NOWYM OKNIE PRZEGŁĄDARKI

HTML rozdział-04/opening-links-in-a-new-window.html

```
<a href="http://www.imdb.com" target="_
blank">Internetowa
baza filmów</a> (wyświetlana w nowym oknie)
```

WYNIK

[Internetowa baza filmów \(wyświetlana w nowym oknie\)](http://www.imdb.com)

target

Jeśli chcemy, by strona docelowa łącza została wyświetlona w nowym oknie, to do otwierającego znacznika `<a>` powinniśmy dodać atrybut `target` o wartości `_blank`.

Jednym z najczęstszych powodów, dla których autor witryny mógłby chcieć, by strona została wyświetlona w nowym oknie, jest sytuacja, gdy łącze wskazuje na inną witrynę. W takich przypadkach autor witryny ma nadzieję, że użytkownik, po zakończeniu przeglądania strony docelowej, wróci do jego witryny.

Ogólnie rzecz biorąc, należy unikać otwierania łączy w nowym oknie przeglądarki; jeśli jednak się na to zdecydujemy, to dobry zwyczaj nakazuje poinformować o tym użytkownika.

ŁĄCZA DO OKREŚLONYCH MIEJSC NA TEJ SAMEJ STRONIE

Na samej górze długiej strony WWW można umieścić listę zawartości z łączami do jej odpowiednich części znajdujących się poniżej. Oprócz tego pośrodku takiej strony można umieścić łącze prowadzące na jej początek, by uchronić użytkowników przed koniecznością przewijania zawartości.

Przed utworzeniem łącza do określonego miejsca strony konieczne będzie jego wskazanie. W tym celu należy skorzystać z atrybutu id (który można dodawać do wszystkich elementów języka HTML). Jak widać na przykładzie przedstawionym obok, we wszystkich elementach `<h1>` i `<h2>` zostały podane wartości atrybutu id identyfikujące poszczególne części strony.

Wartość atrybutu id powinna zaczynać się od litery lub znaku podkreślenia (lecz nie od cyfry ani innego znaku). Co więcej, na jednej stronie nie powinno być dwóch atrybutów id o tej samej wartości.

rozdział-05/linking-to-a-specific-part.html

HTML

```
<h1 id="top">Terminologia filmowa</h1>
<a href="#arc_shot">Ujęcie okrężne</a><br />
<a href="#interlude">Przerywnik</a><br />
<a href="#prologue">Prolog</a><br /><br />
<h2 id="arc_shot">Ujęcie okrężne</h2>
<p>Ujęcie, w którym tworzona jest sekwencja zdjęć obiektu, a kamera przesuwa się wokół niego.</p>
<h2 id="interlude">Przerywnik</h2>
<p>Krótka scena lub sekwencja umieszczona w filmie, niekoniecznie ściśle związana z jego głównym wątkiem.</p>
<h2 id="prologue">Prolog</h2>
<p>Przemowa, wstęp lub krótka scena, bezpośrednio poprzedzające główną akcję lub wątek filmu; przeciwnieństwo epilogu.</p>
<p><a href="#top">Na początek</a></p>
```

Także w przypadku tworzenia łącza do elementu z określona wartością atrybutu id należy posłużyć się elementem `<a>`, lecz wówczas w jego atrybucie href należy podać wartość atrybutu id elementu

docelowego poprzedzoną znakiem `#`. W tym przykładzie element `` tworzy łącze do elementu `<h1>` z atrybutem id o wartości `top`, umieszczonego na samej górze strony.

ŁĄCZA DO KONKRETNYCH MIEJSC NA INNYCH STRONACH

WYNIK

Terminologia filmowa

[Ujęcie okrężne](#)
[Przerywnik](#)
[Prolog](#)

Ujęcie okrężne

Ujęcie, w którym tworzona jest sekwencja zdjęć obiektu, a kamera przesuwa się wokół niego.

Przerywnik

Krótką scenę lub sekwencja umieszczona w filmie, niekoniecznieściśle związana z jego głównym wątkiem.

Prolog

Przemowa, wstęp lub krótka scena, bezpośrednio poprzedzające główną akcję lub wątek filmu; przeciwieństwo epilogu.

[Na początek](#)

Bardzo podobną technikę stosuje się przy tworzeniu łączy do konkretnych miejsc innych stron WWW (niezależnie do tego, czy należą one do naszej, czy do dowolnej innej witryny).

O ile strona docelowa zawiera element z atrybutem id identyfikującym miejsce, od którego chcemy przejść, wystarczy utworzyć łącze, wykorzystując dokładnie tę samą opisaną wcześniej składnię.

A zatem w tym przypadku atrybut href będzie zawierał adres strony docelowej (czyli bezwzględny lub względny adres URL), następnie znak # oraz wartość atrybutu id wybranego elementu tej strony.

Aby np. utworzyć łącze do pierwszego elementu <h2> na naszej przykładowej stronie, możemy użyć następującego elementu: .



New style sheet
— type faces
— ie box bug

Burn project
files for Emanu



Łacza

helion.html_i_css/rozdzial-04/example-04.html

Festiwalowe filmowe

Kalendarium festiwalowe

Oto niektóre spośród tegorocznych festiwali filmowych, które nas interesują. Proszę się z nami [skontaktować](#) w celu uzyskania dodatkowych informacji.

Wrzesień

[XVII PRZEGŁAD FILMÓW GÓRSKICH](#)
Lądek-Zdrój
18 - 23 września 2012 r.

Listopad

[20. Miedzynarodowy Festiwal Sztuki Autorów Zdjęć Filmowych Plus Camerimage](#)
Bydgoszcz
24 listopada - 1 grudnia 2012 r.

[19. Miedzynarodowy Festiwal Filmowy Etuda & Anima](#)
Kraków
23 - 29 listopada 2012 r.

[10. Krakowski Festiwal Górski](#)
Kraków
7 - 9 grudnia 2012 r.

[O nas](#)

[Początek strony](#)



PRZYKŁAD

ŁĄCZA

W tym przykładzie prezentujemy stronę WWW poświęconą filmom.

Na samej górze strony został umieszczony element `<h1>` z atrybutem `id`, dzięki czemu w jej dalszej części będzie można umieścić łącza pozwalające użytkownikom szybko wrócić na początek witryny. Na stronie znajduje się także łącze z adresem poczty elektronicznej, pozwalające użytkownikowi wysłać wiadomość do jej autora. Dodatkowo na stronie znalazły się łącza do innych stron określone przy użyciu bezwzględnych adresów URL. Są to łącza do stron różnych festiwali filmowych. Poniżej, na samym dole strony, znajduje się łącze korzystające ze względnego adresu URL, prowadzące do strony „O nas” umieszczonej w tym samym katalogu.

```
<html>
  <head>
    <title>Łącza</title>
  </head>
  <body>
    <h1 id="top">Festiwale filmowe</h1>
    <h2>Kalendarium festiwalowe</h2>
    <p>Oto niektóre spośród tegorocznych festiwali filmowych, które nas interesują.<br />Proszę się z nami
      <a href="mailto:filmfolk@strona.przykladowa.org">
        skontaktować</a> w celu uzyskania dodatkowych informacji na ich temat.</p>
    <h3>Wrzesień</h3>
    <p><a href="http://przeglad.ladek.pl">
      XVII PRZEGŁĄD FILMÓW GÓRSKICH</a><br />
      Lądek-Zdrój<br />
      18 - 23 września 2012 r.</p>
    <h3>Listopad</h3>
    <p><a href="http://www.pluscamerimage.pl/">
      20. Międzynarodowy Festiwal Sztuki Autorów Zdjęć
      Filmowych Plus Camerimage</a><br />
      Bydgoszcz<br />
      24 listopada - 1 grudnia 2012 r.</p>
    <!-- dalsza zawartość strony -->
    <p><a href="about.html">O nas</a></p>
    <p><a href="#top">Początek strony</a></p>
  </body>
</html>
```



PODSUMOWANIE

ŁĄCZA

- ▶ Łącza są tworzone przy użyciu elementu `<a>`.
- ▶ Strona docelowa łącza jest określana przy użyciu atrybutu `href` elementu `<a>`.
- ▶ Jeśli tworzone łącze ma prowadzić do strony należącej do tej samej witryny, najlepszym rozwiązaniem będzie zastosowanie względnego adresu URL (a nie bezwzględnego).
- ▶ Można tworzyć łącza, których kliknięcie spowoduje uruchomienie programu do obsługi poczty elektronicznej, utworzenie nowej wiadomości i wypełnienie pola adresata.
- ▶ Dzięki zastosowaniu atrybutu `id` można odwoływać się do konkretnych elementów stron docelowych.

5

OBRAZY

- ▶ Sposoby umieszczania obrazów na stronach WWW.
- ▶ Wybór odpowiedniego formatu.
- ▶ Optymalizacja obrazów pod kątem wykorzystania na stronach WWW.

Istnieje wiele powodów, by na tworzonych stronach umieszczać obrazy. Może to być chcę wyświetlenia logo, fotografii, ilustracji, diagramu lub wykresu.

Przy wybieraniu i przygotowywaniu obrazów do umieszczenia na stronie należy wziąć pod uwagę kilka czynników, jednak czas poświęcony na to sprawi, że strona będzie wyglądała atrakcyjniej i bardziej profesjonalnie. W tym rozdziale powiemy, jak:

- Umieszczać obrazy na stronach WWW, korzystając z możliwości języka HTML.
- Odpowiednio dobierać format obrazów.
- Wyświetlać obrazy o odpowiednich rozmiarach.
- Optymalizować obrazy pod kątem wykorzystania w witrynie WWW, tak by strona ładowała się szybciej.

Obrazy można wyświetlać na stronach WWW, posługując się właściwością **background-image** arkuszy stylów CSS; zagadnienia te zostaną opisane w dalszej części książki, na stronach 412 – 419.



WYBIERANIE OBRAZÓW DO PREZENTACJI W WITRYNIE

Jeden obraz jest wart więcej niż tysiąc słów, a doskonale dobrane obrazy mogą stanowić o różnicę pomiędzy witryną wyglądającą przeciętnie a tą naprawdę zachwycającą.

Obrazy są w stanie określić charakter witryny znacznie szybciej niż jej opis. Jeśli nie dysponujemy zdjęciami, które moglibyśmy wykorzystać w witrynie, to jest sporo firm udostępniających **zdjęcia komercyjne** — takie, które można kupić i umieścić

na tworzonych stronach (poniżej lista witryn udostępniających zdjęcia). Trzeba pamiętać, że wszystkie obrazy podlegają prawom autorskim, więc wykorzystanie zdjęcia z innej witryny może przysporzyć nam problemów.

Jeśli strona ma przedstawiać kilka zdjęć (fotografii produktów lub członków zespołu), to umieszczenie ich na jednolitym tle sprawi, że, jako grupa, będą wyglądały lepiej.

OBRAZY POWINNY...

- Mieć znaczenie.
- Dostarczać informacji.
- Tworzyć właściwy klimat.
- Wyróżniać się na pierwszy rzut oka.
- Pasować do palety barw.

WITRYNY WWW SPRZEDAJĄCE ZDJĘCIA

- <http://www.istockphoto.com>
<http://www.gettyimages.com>
<http://www.veer.com>
<http://www.sxc.hu>
<http://www.fotolia.com>

WIĘCEJ W SIECI

Na stronie internetowej poświęconej tej książce dostępna jest galeria, która może pomóc w wyborze odpowiednich obrazów do tworzonych witryn. Można ją znaleźć w dziale *Tools*.

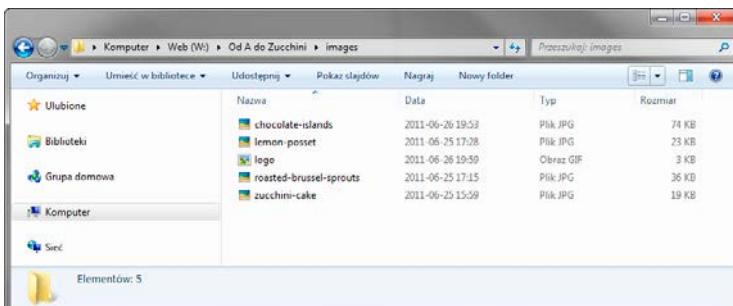
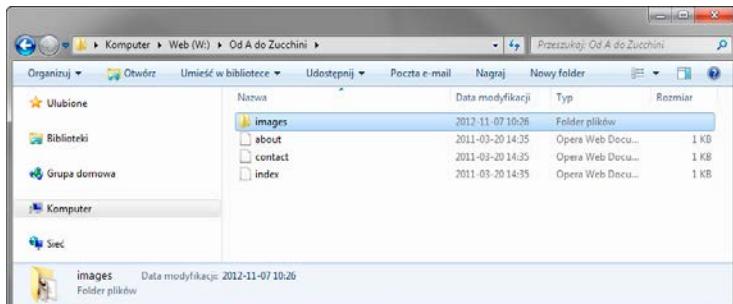
PRZECHOWYWANIE OBRAZÓW WE WŁASNEJ WITRYNIE

Jeśli tworzymy witrynę od samego początku, to dobrym rozwiązaniem będzie utworzenie katalogu, w którym umieścimy wszystkie obrazy używane na stronie.

Kiedy witryna zacznie się powiększać, przechowywanie obrazów w odrębnym katalogu może ułatwić zrozumienie jej struktury i organizacji. Poniżej przedstawiono przykład plików i katalogów witryny; wszystkie obrazy zostały umieszczone w katalogu o nazwie *images*. W przypadku dużych witryn można się także zdecydować na utworzenie

dodatkowych podkatalogów w katalogu *images*. I tak wszystkie logo oraz przyciski mogą być umieszczone w katalogu *interface*, zdjęcia produktów — w katalogu *products*, a obrazy związane z najnowszymi doniesieniami — w katalogu *news*. W przypadku korzystania z systemu zarządzania treścią lub platformy blogowej

wystarczy sięgnąć do ich paneli administracyjnych — tam zazwyczaj dostępne są narzędzia pozwalające na przesyłanie obrazu na serwer, a sama aplikacja będzie posiadała odrębny katalog do przechowywania plików obrazów oraz wszelkich innych plików umieszczanych na serwerze.



DODAWANIE OBRAZÓW

Aby umieścić obraz na stronie WWW, należy użyć elementu . Jest to element pusty (to oznacza, że nie ma znacznika zamykającego). W elemencie należy określić wartości następujących atrybutów:

src

Atrybut ten informuje przeglądarkę, gdzie może znaleźć plik obrazu. Zazwyczaj wartością tego atrybutu będzie względny adres URL wskazujący lokalizację obrazu w tej samej witrynie. (Na przykładzie widać, że obrazy zostały umieszczone w katalogu podrzędnym o nazwie *images*; względne adresy URL zostały opisane na stronach 82 – 83).

alt

Ten atrybut zawiera tekst stanowiący opis obrazu, z którego można skorzystać, gdy obraz nie będzie widoczny.

title

Można także skorzystać z atrybutu title elementu , by przekazać dodatkowe informacje o obrazie. Większość przeglądarek wyświetla zawartość tego znacznika w formie etykiety ekranowej po wskazaniu obrazka przyciskiem myszy.

rozdział-05/adding-images.html

HTML

```

```

WYNIK



Tekst umieszczany w atrybucie **alt** często jest nazywany **tekstem alternatywnym**. Powinien przedstawać dokładny opis zawartości obrazu, tak by mógł on zostać zrozumiany przez czytniki ekranowe (używane przez osoby mające problemy ze wzrokiem) oraz wyszukiwarki.

Jeśli obraz służy wyłącznie poprawie wizualnej atrakcyjności strony (i nie ma żadnego innego znaczenia, tak jak kreska oddzielająca jakieś fragmenty strony), to i tak w znaczniku należy umieścić atrybut **alt**, choć jego wartość powinna być pusta.

WYSOKOŚĆ I SZEROKOŚĆ OBRAZÓW

HTML

rozdział-05/height-and-width-of-images.html

```

```

WYNIK



Kolejnymi atrybutami, często stosowanymi w elementach ``, są dwa atrybuty określające wymiary obrazu prezentowanego na stronie:

height

Atrybut ten określa wysokość obrazu wyrażoną w pikselach.

width

Z kolei ten określa szerokość obrazu; także jest wyrażany w pikselach.

Często się zdarza, że pobranie obrazów zajmuje więcej czasu niż wczytanie kodu HTML strony. Właśnie z tego powodu podawanie wymiarów obrazu jest bardzo dobrym rozwiązaniem — dzięki temu przeglądarka będzie w stanie wyświetlić pozostałą zawartość strony, pozostawiając odpowiednio dużo miejsca dla obrazka.

Wymiary obrazów są coraz częściej podawane nie w kodzie HTML strony, lecz w arkuszach stylów CSS; więcej informacji na ten temat można znaleźć na stronach 408 – 409.

W KTÓRYCH MIEJSCACH KODU UMIESZCZAĆ OBRAZY?

To, w którym miejscu kodu umieścimy obraz, będzie miało wpływ na to, w jaki sposób zostanie on później wyświetlony. Poniżej przedstawiono trzy przykłady, w których wybór miejsca decyduje o sposobie wyświetlania obrazu.

1. PRZED AKAPITEM TEKSTU

W tym przypadku akapit tekstu rozpoczęcie się od nowego wiersza poniżej obrazu.

2. WEWNĄTRZ AKAPITU, NA JEGO POCZĄTKU

W tym przypadku pierwszy wiersz tekstu w akapicie zostanie wyrównany w pionie do dolnej krawędzi obrazu.

3. WEWNĄTRZ TEKSTU AKAPITU

W tym przypadku obraz zostanie wyświetlony pomiędzy słowami, wewnątrz treści akapitu.

rozdział-05/where-to-place-images.html

HTML

```

<p>Obecnie na Ziemi żyje ponad 10 tysięcy gatunków ptaków zamieszkujących przeróżne ekosystemy, zaczynając od Arktyki, a na Antarktyce kończąc. Wiele spośród tych gatunków podejmuje trud długich, corocznych migracji, a znacznie więcej odbywa o wiele krótsze, nieregularne podróże.</p>
<hr />
<p>Obecnie na Ziemi żyje ponad 10 tysięcy gatunków ptaków zamieszkujących przeróżne ekosystemy, zaczynając od Arktyki, a na Antarktyce kończąc. Wiele spośród tych gatunków podejmuje trud długich, corocznych migracji, a znacznie więcej odbywa o wiele krótsze, nieregularne podróże.</p>
<hr />
<p>Obecnie na Ziemi żyje ponad 10 tysięcy gatunków ptaków zamieszkujących przeróżne ekosystemy, zaczynając od Arktyki, a na Antarktyce kończąc.  Wiele spośród tych gatunków podejmuje trud długich, corocznych migracji, a znacznie więcej odbywa o wiele krótsze, nieregularne podróże.</p>
```

WYNIK



Obecnie na Ziemi żyje ponad 10 tysięcy gatunków ptaków zamieszkujących przeróżne ekosystemy, zaczynając od Arktyki, a na Antarktyce kończąc. Wiele spośród tych gatunków podejmuje trud długich, corocznych migracji, a znacznie więcej odbywa o wiele krótsze, nieregularne podróże.



Obecnie na Ziemi żyje ponad 10 tysięcy gatunków ptaków zamieszkujących przeróżne ekosystemy, zaczynając od Arktyki, a na Antarktyce kończąc. Wiele spośród tych gatunków podejmuje trud długich, corocznych migracji, a znacznie więcej odbywa o wiele krótsze, nieregularne podróże.

Obecnie na Ziemi żyje ponad 10 tysięcy gatunków ptaków zamieszkujących przeróżne



Wiele

ekosystemy, zaczynając od Arktyki, a na Antarktyce kończąc. Wiele spośród tych gatunków podejmuje trud długich, corocznych migracji, a znacznie więcej odbywa o wiele krótsze, nieregularne podróże.

Miejsce, w którym umieścimy obraz, ma znaczenie, gdyż przeglądarki wyświetlają elementy HTML na dwa sposoby.

Elementy blokowe zawsze są wyświetlane w nowym wierszu. Przykładami takich elementów są `<h1>` i `<p>`.

Jeśli bezpośrednio za elementem `` zostanie umieszczony element blokowy (taki jak akapit), to element blokowy zostanie wyświetlony w nowym wierszu, poniżej obrazka; co pokazuje pierwszy przykład na zamieszczonym obok rysunku.

Elementy wewnętrzwerszowe są umieszczane wewnętrz elementów blokowych i nie zaczynają się od nowego wiersza.

Przykładami elementów wewnętrzwerszowych są: ``, `` i ``.

Jeśli element `` zostanie umieszczony wewnętrz elementu blokowego, to tekst oraz wszelkie inne elementy wewnętrzwerszowe będą otaczały obrazek, tak jak pokazano na drugim i trzecim przykładzie.

Elementy blokowe oraz wewnętrzwerszowe zostały opisane bardziej szczegółowo w dalszej części książki, na stronach 184 – 185.

WYRÓWNYWANIE OBRAZÓW W POZIOMIE — STARY SPOSÓB

align

Atrybut align był niegdyś powszechnie używany w celu określenia, w jaki sposób inna zawartość strony powinna otaczać obraz. Atrybut ten jednak został usunięty z języka HTML5, a tworzone aktualnie witryny powinny kontrolować wyrównywanie obrazów wyłącznie przy użyciu kaskadowych arkuszy stylów (informacje na ten temat można znaleźć na stronach 410 – 411).

Zdecydowałem się zamieścić tu informację na temat tego atrybutu, gdyż Czytelnik na pewno spotka się z nim, analizując kody istniejących stron WWW, oraz dlatego, że niektóre wizualne edytory HTML wciąż korzystają z niego w celu określania sposobu wyrównywania obrazów.

Atrybut align może przyjmować następujące wartości określające sposób wyrównania obrazu w poziomie:

[rozdział-05/aligning-images-horizontally.html](#)

HTML

```
<p>Obecnie na Ziemi żyje ponad 10 tysięcy gatunków ptaków zamieszujących przeróżne ekosystemy, zaczynając od Arktyki, a na Antarktyce kończąc. Wiele spośród tych gatunków podejmuje trud długich, corocznych migracji, a znacznie więcej odbywa o wiele krótsze, nieregularne podróże.</p>
<hr />
<p>Obecnie na Ziemi żyje ponad 10 tysięcy gatunków ptaków zamieszujących przeróżne ekosystemy, zaczynając od Arktyki, a na Antarktyce kończąc. Wiele spośród tych gatunków podejmuje trud długich, corocznych migracji, a znacznie więcej odbywa o wiele krótsze, nieregularne podróże.</p>
```

left

Użycie tej wartości sprawi, że element zostanie wyrównany do lewej (przez co tekst będzie go otaczał z prawej strony).

right

Użycie tej wartości sprawi, że element zostanie wyrównany do prawej (przez co tekst będzie go otaczał z lewej strony).

WYNIK



Obecnie na Ziemi żyje ponad 10 tysięcy gatunków ptaków zamieszkujących przeróżne ekosystemy, zaczynając od Arktyki, a na Antarktyce kończąc. Wiele spośród tych gatunków podejmuje trud długich, corocznych migracji, a znacznie więcej odbywa o wiele krótsze, nieregularne podróże.

Obecnie na Ziemi żyje ponad 10 tysięcy gatunków ptaków zamieszkujących przeróżne ekosystemy, zaczynając od Arktyki, a na Antarktyce kończąc. Wiele spośród tych gatunków podejmuje trud długich, corocznych migracji, a znacznie więcej odbywa o wiele krótsze, nieregularne podróże.



Taki układ zawartości wygląda znacznie lepiej niż przedstawiony na poprzedniej stronie, gdzie obok obrazka wyświetlał się tylko jeden wiersz tekstu.

Kiedy atrybutowi `align` przypiszemy wartość `left`, obrazek zostanie wyświetlony z lewej strony, a tekst będzie go otaczał.

Z kolei gdy atrybutowi `align` przypiszemy wartość `right`, obrazek zostanie wyświetlony z prawej strony, a tekst, podobnie jak poprzednim przypadku, będzie go otaczał.

Jeśli tekst będzie się stykał bezpośrednio z krawędzią obrazu, to jego lektura może być nieco utrudniona. Na stronach 312 – 313 będzie mowa o dodawaniu odstępu pomiędzy obrazkiem i tekstem z wykorzystaniem właściwości CSS: `padding` oraz `margin`.

WYRÓWNYWANIE OBRAZÓW W PIONIE — STARY SPOSÓB

Na poprzedniej stronie wspomniano o tym, że w języku HTML5 atrybut align nie jest używany, niemniej przedstawiam go tutaj, gdyż można go jeszcze znaleźć w kodzie starych witryn i wciąż jest używany przez niektóre wizualne edytory stron WWW.

Na stronach 284 – 285 znajdują się wskazówki, jak uzyskać identyczne efekty, korzystając z możliwości CSS. Atrybut align udostępnia trzy kolejne wartości, pozwalające określić sposób wyrównania obrazu w pionie względem otaczającej go zawartości strony:

top

Użycie tej wartości sprawi, że pierwszy wiersz tekstu otaczającego obraz zostanie wyrównany z jego górną krawędzią.

middle

Użycie tej wartości sprawi, że pierwszy wiersz tekstu otaczającego element zostanie wyrównany do połowy wysokości obrazu.

bottom

Użycie tej wartości sprawi, że pierwszy wiersz tekstu otaczającego obraz zostanie wyrównany z jego dolną krawędzią.

[rozdział 05/aligning-images-vertically.html](#)

HTML

```
<p>Obecnie na Ziemi żyje ponad 10 tysięcy gatunków ptaków zamieszkujących przeróżne ekosystemy, zaczynając od Arktyki, a na Antarktyce kończąc. Wiele spośród tych gatunków podejmuje trud długich, corocznych migracji, a znacznie więcej odbywa o wiele krótsze, nieregularne podróże.</p>
```

```
<hr />
```

```
<p>Obecnie na Ziemi żyje ponad 10 tysięcy gatunków ptaków zamieszkujących przeróżne ekosystemy, zaczynając od Arktyki, a na Antarktyce kończąc. Wiele spośród tych gatunków podejmuje trud długich, corocznych migracji, a znacznie więcej odbywa o wiele krótsze, nieregularne podróże.</p>
```

```
<hr />
```

```
<p>Obecnie na Ziemi żyje ponad 10 tysięcy gatunków ptaków zamieszkujących przeróżne ekosystemy, zaczynając od Arktyki, a na Antarktyce kończąc. Wiele spośród tych gatunków podejmuje trud długich, corocznych migracji, a znacznie więcej odbywa o wiele krótsze, nieregularne podróże.</p>
```

Wartość **top** sprawi, że pierwszy wiersz tekstu zostanie umieszczony na wysokości górnej krawędzi obrazu, a kolejne wiersze znajdują się poniżej niego.

Wartość **middle** sprawi, że pierwszy wiersz tekstu zostanie umieszczony w połowie wysokości obrazu, a kolejne wiersze znajdują się poniżej niego.

Wartość **bottom** sprawi, że pierwszy wiersz tekstu zostanie umieszczony na wysokości dolnej krawędzi obrazu, a kolejne wiersze znajdują się poniżej niego.

Jeśli tekst będzie się stykał bezpośrednio z krawędzią obrazu, to jego lektura może być nieco utrudniona. Na stronach 312 – 313 będzie mowa o dodawaniu odstępu pomiędzy obrazkiem i tekstem z wykorzystaniem właściwości CSS: **padding** oraz **margin**.

Jeśli chcemy, by nie tylko pierwszy wiersz, lecz znacznie większe fragmenty tekstu otaczały obraz, możemy skorzystać z właściwości **float** arkuszy stylów; została ona dokładniej opisana na stronach 369 – 371.

W starszym kodzie można znaleźć przykłady wykorzystania atrybutu **align** z wartościami **left** lub **right** w celu otoczenia obrazu tekstem (opisano to na poprzedniej stronie), niemniej takie rozwiązania nie są już zalecane.

WYNIK



Obecnie na Ziemi żyje ponad 10 tysięcy gatunków ptaków

zamieszczających przeróżne ekosystemy, zaczynając od Arktyki, a na Antarktyce kończąc. Wiele spośród tych gatunków podejmuje trud długich, corocznych migracji, a znacznie więcej odbywa o wiele krótsze, nieregularne podróże.



Obecnie na Ziemi żyje ponad 10 tysięcy gatunków ptaków

zamieszczających przeróżne ekosystemy, zaczynając od Arktyki, a na Antarktyce kończąc. Wiele spośród tych gatunków podejmuje trud długich, corocznych migracji, a znacznie więcej odbywa o wiele krótsze, nieregularne podróże.



Obecnie na Ziemi żyje ponad 10 tysięcy gatunków ptaków

zamieszczających przeróżne ekosystemy, zaczynając od Arktyki, a na Antarktyce kończąc. Wiele spośród tych gatunków podejmuje trud długich, corocznych migracji, a znacznie więcej odbywa o wiele krótsze, nieregularne podróże.

TRZY ZASADY TWORZENIA OBRAZÓW

Poniżej zostały opisane trzy zasady, o których należy pamiętać w trakcie przygotowywania obrazów przeznaczonych do użycia na stronach WWW. Każda z zasad zostanie dokładniej opisana na kolejnych dziewięciu stronach tego rozdziału.

1

ZAPISUJ OBRAZY W ODPOWIEDNIM FORMATIE

W witrynach WWW zazwyczaj używa się obrazów zapisanych w formatach: JPEG, GIF i PNG. W efekcie wyboru nieodpowiedniego formatu obrazy mogą utracić niezbędną ostrość albo ich zbyt duża wielkość może wydłużyć czas wyświetlania stron.

2

ZAPISUJ OBRAZY O ODPOWIEDNICH WYMIARACH

Zapisywane obrazy powinny mieć dokładnie takie wymiary jak te, w których będą widoczne na stronie. Jeśli rzeczywiście wymiary obrazu będą mniejsze od tych używanych na stronie, zostanie on rozciagnięty i zniekształcony. Z kolei jeśli obraz będzie większy, niż podano, przyczyni się to do wydłużenia czasu wyświetlania strony.

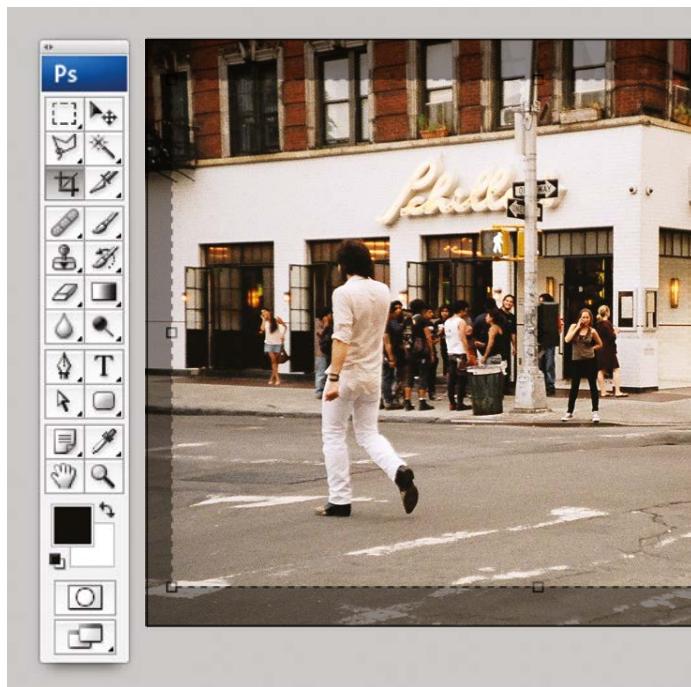
3

UŻYWAJ ODPOWIEDNIEJ ROZDZIELCZOŚCI

Ekrany komputerów składają się z kropek nazywanych pikselami. Także obrazy używane na stronach WWW składają się z takich kropek. Rozdzielcość to termin określający liczbę kropek na cal — w większości monitorów komputerowych wynosi ona jedynie 72 piksele na cal. A zatem utworzenie obrazów o wyższej rozdzielcości sprawi, że będą one niepotrzebnie duże, przez co wydłużą się czas wyświetlania strony.

NARZĘDZIA DO EDYCJI I ZAPISU OBRAZÓW

Dostępnych jest kilka programów służących do edycji i zapisywania obrazów, gwarantujących, że będą one miały odpowiednie wymiary, format oraz rozdzielcość.



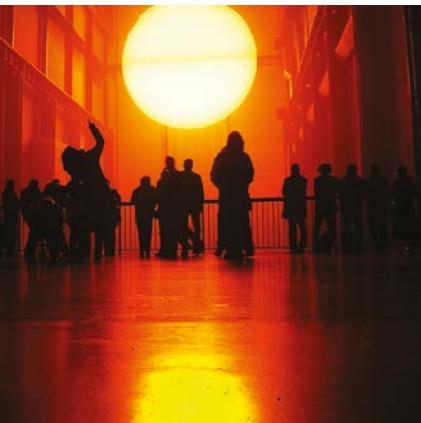
Najpopularniejszym programem używanym przez profesjonalnych grafików jest **Adobe Photoshop**. (W rzeczywistości profesjonalni projektanci stron często używają tego programu, by projektować całe witryny). Jego pełna wersja jest droga, dostępna jest jednak tańsza — Photoshop Elements — która powinna zaspokoić potrzeby większości początkujących twórców stron.

**INNE
OPROGRAMOWANIE**
Adobe Fireworks
Pixelmator
PaintShop Pro
Paint.NET

**EDYTORY
INTERNETOWE**
www.photoshop.com
www.pixlr.com
www.splashup.com
www.ipiccy.com

WIĘCEJ W SIECI
Warto obejrzeć filmy demonstrujące, w jaki sposób, korzystając z obu tych programów, modyfikować wielkość obrazów oraz zapisywać je w odpowiednim formacie.

FORMATY OBRAZÓW: JPEG

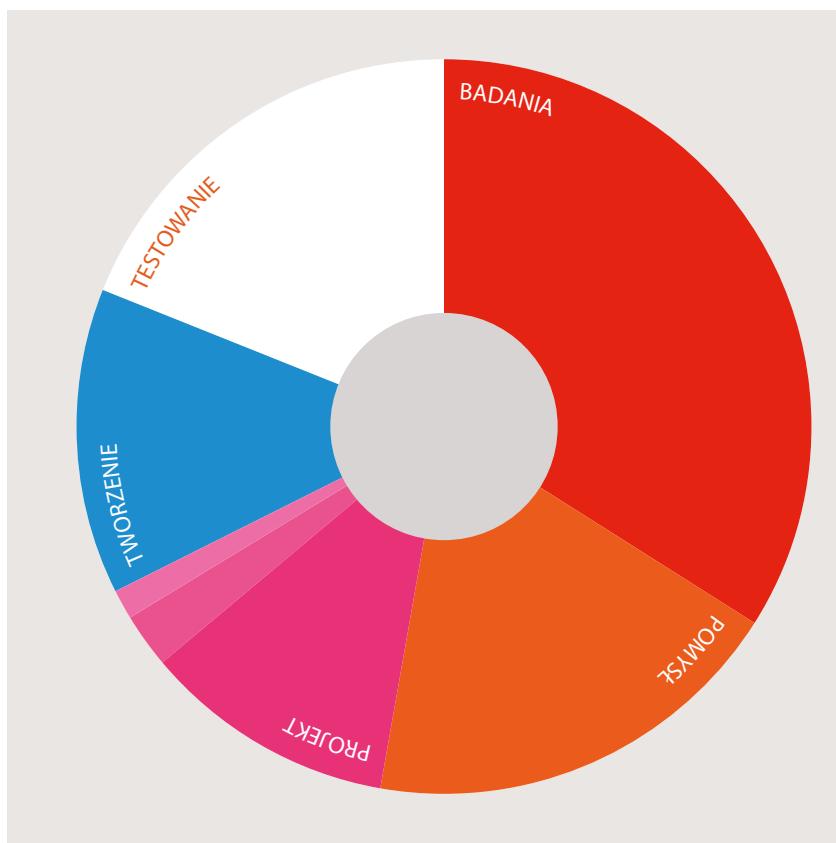
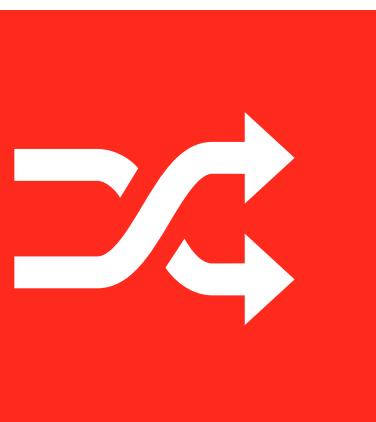


Zawsze, gdy obraz ma wiele kolorów, należy używać formatu JPEG. Zdjęcia przedstawiające zaśnieżony krajobraz lub zachmurzone niebo mogą zawierać duże obszary białego lub szarego koloru, zazwyczaj jednak można na nich wyróżnić bardzo wiele kolorów, które nieznacznie się od siebie różnią.

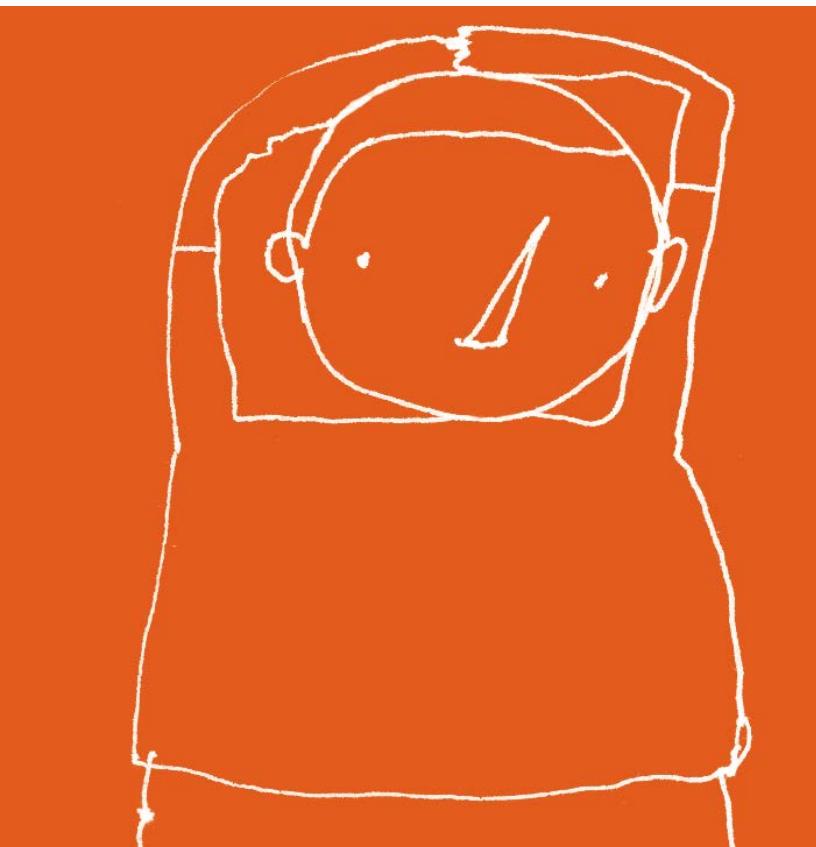




FORMATY OBRAZÓW: GIF



Formatów GIF lub PNG należy używać do zapisywania obrazów zawierających niewiele kolorów lub duże obszary tego samego koloru.



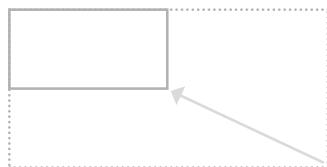
Kiedy na obrazie występują duże obszary tego samego koloru, mówi się, że są to fragmenty jednolitego koloru. Takie obszary jednolitego koloru występują często w logo, ilustracjach oraz diagramach. (Trzeba zauważyć, że zdjęcia śniegu, nieba czy trawy nie zawierają takich jednolitych kolorów, tworzą je obszary składające się z bardzo wielu, nieznacznie się od siebie różniących odcienni jednego koloru. Takie obrazy nie nadają się do zapisu w formacie GIF ani PNG).

WYMIARY OBRAZÓW

Przy zapisywaniu obrazów przeznaczonych do użycia na stronie WWW należy zadbać, by ich wymiary odpowiadały szerokości i wysokości danego obrazu wyświetlanego na stronie.

Jeśli np. zaprojektowaliśmy stronę, na której ma być umieszczony obraz o szerokości 300 i wysokości 150 pikseli, to obraz, którego faktycznie użyjemy, powinien mieć 300 pikseli szerokości i 150 pikseli wysokości. Być może konieczne będzie użycie programu graficznego w celu odpowiedniej zmiany wielkości i przycięcia obrazu.

Zmniejszanie wymiarów obrazu
Można ograniczyć wymiary obrazu, by stworzyć jego pomniejszoną wersję.



Przykład: jeśli obraz ma 600 pikseli szerokości i 300 pikseli wysokości, to można go zmniejszyć o 50%.

Wynik: utworzymy obraz, którego pobranie zajmie mniej czasu.

Obrazów należy szukać ze świadomością, w jaki sposób będzie można zmieniać ich wymiary; wyobraźmy sobie, że na zaprojektowanej przez nas stronie ma się znaleźć obraz o szerokości 300 i wysokości 150 pikseli.

Zwiększenie wymiarów obrazu
Nie można znacząco zwiększyć wymiarów cyfrowych zdjęć bez utraty ich jakości.



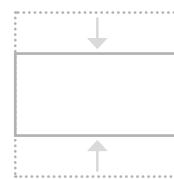
Przykład: jeśli obrazek będzie miał 100 pikseli szerokości i 50 pikseli wysokości, to powiększenie go o 300% spowoduje znaczne pogorszenie jego jakości.

Wynik: obraz będzie rozmarzony lub będą na nim widoczne duże, prostokątne piksele.

WIĘCEJ W SIECI

W dziale *Tools* na stronie internetowej poświęconej tej książce znajdują się filmowe poradniki pokazujące, jak zmieniać wymiary obrazów przy użyciu programu GIMP.

Zmiana kształtu
Tylko niektóre obrazy można przycinać bez utraty widocznych na nich cennych informacji (więcej na ten temat na następnej stronie).



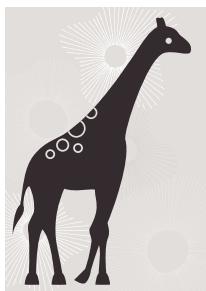
Przykład: jeśli dysponujemy obrazem, który jest kwadratem o krawędziach długości 300 pikseli, możemy usunąć jego fragmenty, jednak robiąc to, narażamy się na utratę pewnych cennych informacji.

Wynik: tylko niektóre obrazy można przycinać tak, by ich późniejsze zastosowanie wciąż miało sens.

PRZYCINANIE OBRAZÓW

Przy przycinaniu obrazów ważne jest, by nie stracić żadnych ważnych informacji. Jeśli to możliwe, warto korzystać z obrazów, które od razu mają właściwy kształt.

UKŁAD PIONOWY

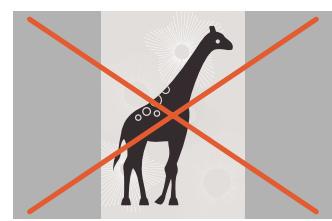


Oto obraz przedstawiający żyrafę — najlepiej prezentować go w **układzie pionowym**.

UKŁAD POZIOMY

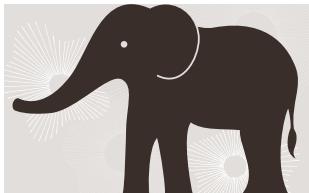


Jeśli **przytniemy** ten obraz, by przedstawić go w układzie poziomym, stracimy głowę i dolną część nóg żyrafy.



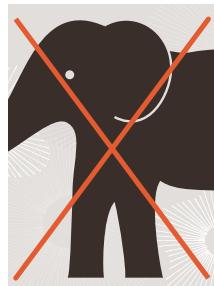
Jeśli **dodamy dodatkowe obszary** z lewej oraz prawej strony obrazu, będą się one wyraźnie odróżniać od jego oryginalnego tła.

UKŁAD POZIOMY



A to obraz przedstawiający słonia — najlepiej prezentować go w **układzie poziomym**.

UKŁAD PIONOWY



Jeśli **przytniemy** ten obraz, by przedstawić go w układzie pionowym, stracimy trąbę słonia oraz tylną część jego ciała.



Jeśli **dodamy dodatkowe obszary** powyżej oraz poniżej oryginalnego obrazu, będą się one wyraźnie odróżniać od jego oryginalnego tła.

OKREŚLANIE WYMIARÓW OBRAZÓW I ROZDZIELCZOŚĆ

Wymiary obrazu, który ma być wyświetlany na ekranie, zawsze należy podawać w pikselach (a nie w centymetrach ani calach).

Obrazy wyświetlane na ekranach komputerów składają się z miniaturowych prostokątów zwanych **pikselami**. **Rozdzielcością** (ang. *resolution*) ekranu nazywamy liczbę wyświetlanych na nim pikseli; większość komputerów pozwala na jej zmniejszanie i powiększanie.

Jeśli rozdzielcość ekranu została ustawiona na 640×480 pikseli, a następnie zostanie powiększona

do 1024×768 pikseli, to obrazy oraz tekst będą się wydawały mniejsze, gdyż na ekranie o tych samych wymiarach będzie prezentowanych znacznie więcej pikseli (liczba pikseli na cal będzie większa).

Obrazy tworzone z przeznaczeniem do druku najlepiej zapisywać w rozdzielcości 300 punktów na cal (dpi — ang. *dots per inch*) lub wyższej (by zapewnić, że będą ostre).

W przypadku stron WWW rozdzielcość obrazu nie ma znaczenia — tu wielkość obrazu odpowiada jego **wymiarom** wyrażonym w pikselach.

Można się spotkać z opinią, że obrazy tworzone na potrzeby stron WWW powinny być zapisywane w rozdzielcości 72 pikseli na cal (ppi — ang. *pixels per inch*), jednak użytkownicy mogą zmieniać rozdzielcość monitora (a tym samym liczbę pikseli wyświetlanych w jednym calu ekranu). Zatem same ekranie determinują wielkość obrazów w zależności od wysokości i szerokości wyświetlanych pikseli.

Obraz, który ma wymiary 300×300 pikseli i został zapisany w rozdzielcości 72 ppi, będzie wyglądał dokładnie tak samo jak obraz o wymiarach 300×300 pikseli, zapisany w rozdzielcości 500 ppi. Zastosowana rozdzielcość bowiem ma znaczenie wyłącznie w przypadku drukowania obrazu.



OBRAZY WEKTOROWE

Obrazy wektorowe różnią się od obrazów rastrowych i są niezależne od rozdzielczości. Obrazy tego typu zazwyczaj tworzy się w takich programach graficznych jak Adobe Illustrator.

Kiedy obrazy w znacznej mierze składają się z linii (tak jest w przypadku logo, ilustracji, diagramów), projektanci często decydują się na zapisanie ich w formacie wektorowym. Takie obrazy ogromnie różnią się od obrazów rastrowych.

Obraz wektorowy tworzy się, umieszczając w jego obszarze punkty, które są następnie łączone liniami. Następnie można określić kolor, który posłuży do wypełnienia obszarów pomiędzy liniami.

Zaletą tworzenia rysunków w formacie wektorowym jest możliwość powiększenia ich rozmiarów bez utraty jakości.

Aktualny sposób korzystania z obrazów wektorowych na stronach WWW polega na zapisaniu oryginalnego obrazu w formacie rastrowym, który następnie będzie wyświetlany na stronie.

SVG (*Scalable Vector Graphics*) to stosunkowo nowy format używany do wyświetlania obrazów wektorowych bezpośrednio na stronach WWW (bez uprzedniego zapisywania ich w formie obrazów rastrowych); nie jest on jednak jeszcze powszechnie obsługiwany.



ANIMOWANE OBRAZY GIF

Animowane obrazy GIF prezentują sekwencję składającą się z kilku klatek, dzięki czemu można ich używać do tworzenia prostych animacji.

Poniżej pokazano kolejne klatki tworzące animowany obraz GIF przedstawiający kręczącą się w kółko pomarańczową kropkę — podobny do animacji, które często są stosowane podczas czytывания zawartości strony.

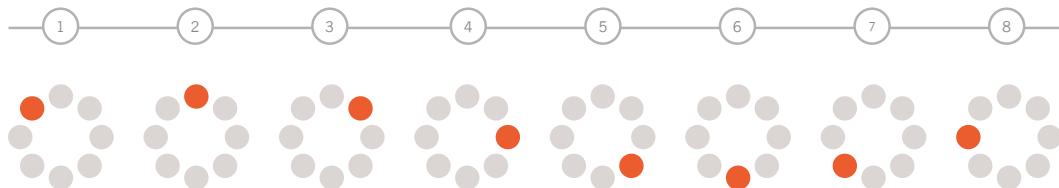
Niektóre programy graficzne, takie jak Adobe Photoshop, pozwalają na tworzenie animowanych obrazów GIF. Na WWW można znaleźć kilka poradników opisujących, jak to zrobić. Dostępne są także witryny pozwalające na przesyłanie obrazów reprezentujących poszczególne klatki animacji i utworzenie na ich podstawie jednego, animowanego obrazu GIF.

KONIECZNIE ZAPAMIĘTAJ

Każda kolejna klatka animacji zwiększa wielkość pliku, tym samym wydłuża czas jego pobierania z serwera (a internauci nie lubią dugo czekać na wyświetlenie strony).

Ponieważ GIF nie jest optymalnym formatem do wyświetlania fotografii, dlatego animowane obrazy GIF nadają się w rzeczywistości wyłącznie do prezentowania prostych ilustracji.

Niektórzy projektanci nie przepadają za animowanymi obrazami GIF, gdyż doskonale pamiętają, że w latach 90. ubiegłego wieku były nadużywane przez początkujących twórców stron.



PRZEZROCZYSTOŚĆ

Jeśli tworzymy częściowo przezroczyste (prześwitujące) obrazy do wykorzystania na stronach WWW, mamy do wyboru jeden z dwóch formatów.

PRZEZROCZYSTE OBRAZY GIF

Jeśli przezroczysty obszar obrazka ma ostre krawędzie i jest przezroczysty w 100% (a nie częściowo), to taki obraz można zapisać w formacie GIF (z włączoną opcją przezroczystości).

PNG

Jeśli przezroczysty obszar obrazka ma skośne lub zaokrąglone krawędzie albo jeśli jest tylko częściowo przezroczysty lub rzuca cień, to obrazek trzeba zapisać w formacie PNG.

Przezroczyste obrazy PNG nie są w pełni obsługiwane w starszych przeglądarkach, przy czym dotyczy to przede wszystkim Internet Explorera 6. Można jednak użyć odpowiedniego kodu JavaScript, by rozwiązać ten problem. Szczegółowe informacje na ten temat znajdują się w dziale *Tools* witryny poświęconej tej książce — www.htmlandcssbook.com/.

PROSTY



PRZEKĄTNE



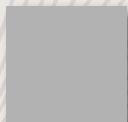
ZAOKRĄGLONY



PÓŁPRZEZROCZYSTY



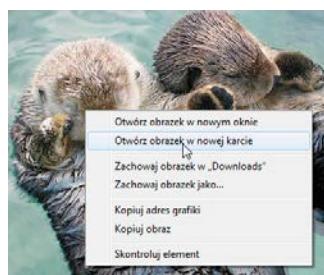
CIEŃ



SPRAWDZANIE WYMIARÓW OBRAZÓW ZNAJDOWANYCH W INTERNECIE

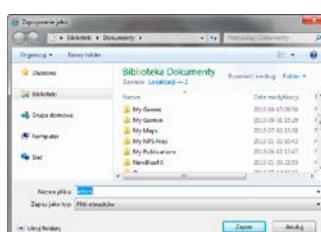
SPRAWDZANIE WYMIARÓW OBRAZÓW

Podczas aktualizacji witryny może się zdarzyć, że zanim zastąpimy starszą wersję obrazu nową, będziemy musieli sprawdzić jego rozmiar. Aby to zrobić, wystarczy kliknąć prawym przyciskiem myszy na obrazie i wybrać odpowiednią opcję z wyświetlnego menu kontekstowego. (Użytkownicy komputerów Mac będą musieli w tym celu nacisnąć klawisz *Control* i kliknąć).



POBIERANIE OBRAZÓW

Jeśli zechcemy pobrać jakiś obraz ze strony WWW, także to będziemy mogli zrobić, wybierając opcję z menu kontekstowego. (Trzeba przy tym pamiętać, że wszystkie obrazy zamieszczane na stronach podlegają prawom autorskim, a ich wykorzystanie wymaga uzyskania odpowiedniego pozwolenia).



Z lewej strony pokazano, w jaki sposób można sprawdzić wymiary obrazków oraz jak je pobierać w przeglądarce Safari. Poniżej przedstawione zostały opcje, które w tym samym celu należy wybrać z menu kontekstowego w innych przeglądarkach.

CHROME

Wymiary: *Otwórz grafikę w nowej karcie*
Wymiary zostaną wyświetlane na nowej karcie.
Pobranie: *Zapisz grafikę jako...*

FIREFOX

Wymiary:
Pokaż informacje o obrazku
Wymiary zostaną wyświetlane w oknie dialogowym.
Pobranie: *Zapisz obrazek jako...*

INTERNET EXPLORER

Wymiary: *Właściwości*
Wymiary zostaną wyświetlane w oknie dialogowym.
Pobranie: *Zapisz obraz jako...*

SAFARI

Wymiary: *Otwórz obrazek w nowej karcie...*
Wymiary zostaną wyświetlane na pasku tytułu przeglądarki.
Pobieranie:
Zachowaj obrazek jako...

HTML5: ELEMENTY FIGURE ORAZ FIGCAPTION

HTML

rozdział-05/figure-and-figure-caption.html

```
<figure>
  
  <br />
  <figcaption>Podczas snu wydry morskie trzymają
  się za łapki, dzięki czemu nie odpływają
  od siebie.</figcaption>
</figure>
```

WYNIK



Podczas snu wydry morskie trzymają się za łapki, dzięki czemu nie odpływają od siebie.

<figure>

Do obrazów prezentowanych na stronach WWW zazwyczaj są dodawane podpisy. W języku HTML5 wprowadzony został nowy element `<figure>` pozwalający na wyświetlenie obrazu i podanie jego podpisu oraz na powiązanie ze sobą obu tych elementów.

Wewnątrz elementu `<figure>` można umieścić więcej niż jeden obraz, o ile tylko podany podpis odnosi się do nich wszystkich.

<figcaption>

Element `<figcaption>` został dodany do języka HTML5, aby zapewnić autorom stron możliwość dodawania podpisów do wyświetlanych obrazów.

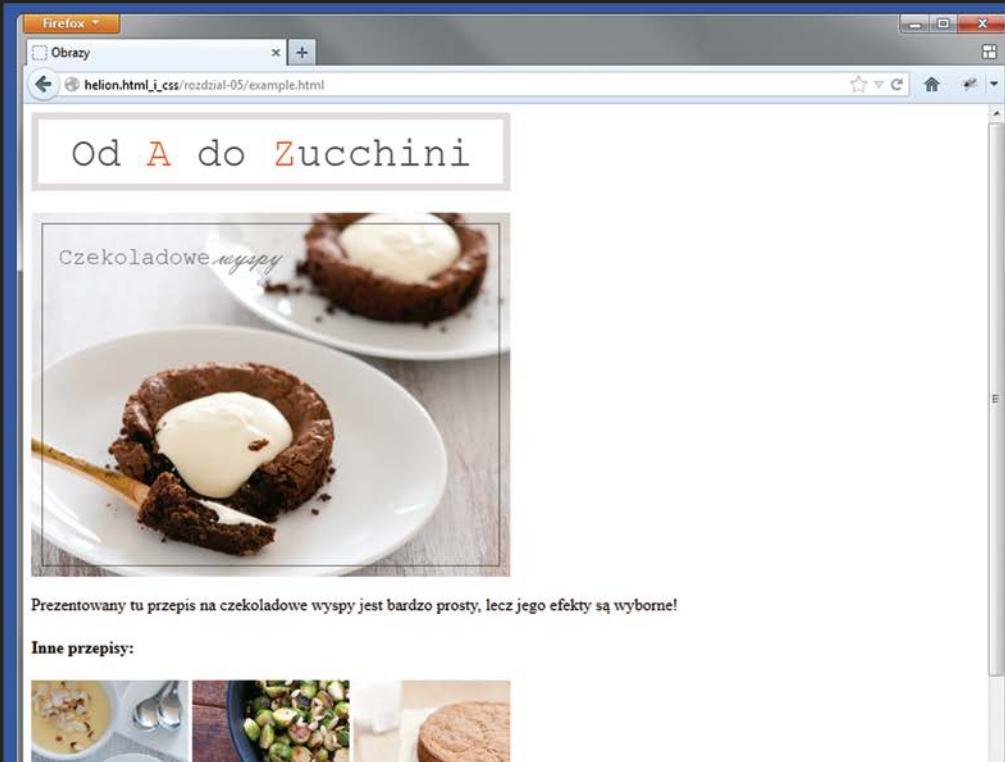
Przed jego wprowadzeniem nie był dostępny żaden sposób pozwalający na powiązanie elementu `` z podpisem obrazka.

Starsze przeglądarki, które nie obsługują języka HTML5, pominą nowe elementy i wyświetwią jedynie ich zawartość.

W tym przykładzie logo jest obrazem w formacie GIF, gdyż składa się z jednolitych kolorów; natomiast zdjęcia są zapisane w formacie JPEG. Główne zdjęcie zostało umieszczone w elemencie <figure> i ma podpis.

Atrybut alt każdego obrazu podaje jego opis, z którego mogą skorzystać użytkownicy postępujący się czytnikami ekranowymi. Dodatkowych informacji o poszczególnych obrazach (wyświetlanych w formie etykiet ekranowych) dostarczają atrybuty title.

W tym przykładzie nie są używane atrybuty height, width ani align, gdyż uznaje się je za przestarzałe; zamiast nich należy stosować odpowiednie właściwości arkusza stylów CSS.



PRZYKŁAD

OBRAZY

```
<html>
  <head>
    <title>Obrazy</title>
    <style type="text/css">
      figure {margin: 0 0;}
    </style>
  </head>
  <body>
    <h1>
      
    </h1>
    <figure>
      
      <p>
        <figcaption>
          Prezentowany tu przepis na czekoladowe
          wyspy jest bardzo prosty, lecz jego
          efekty są wyborne!
        </figcaption>
      </p>
    </figure>
    <h4>Inne przepisy:</h4>
    <p>
      
      
      
    </p>
  </body>
</html>
```


PODSUMOWANIE

OBRAZY

- ▶ Do wyświetlania obrazów na stronach WWW służy element ``.
- ▶ Zawsze należy podać atrybut `src` określający położenie źródła obrazu oraz atrybut `alt` zawierający jego krótki opis.
- ▶ Zapisywane obrazy powinny mieć takie same wymiary, jakie będą miały po wyświetleniu na stronie; dodatkowo należy wybrać właściwy format.
- ▶ Fotografie najlepiej zapisywać w formacie JPEG, a ilustracje lub logo zawierające jednolite kolory — w formacie GIF.

6

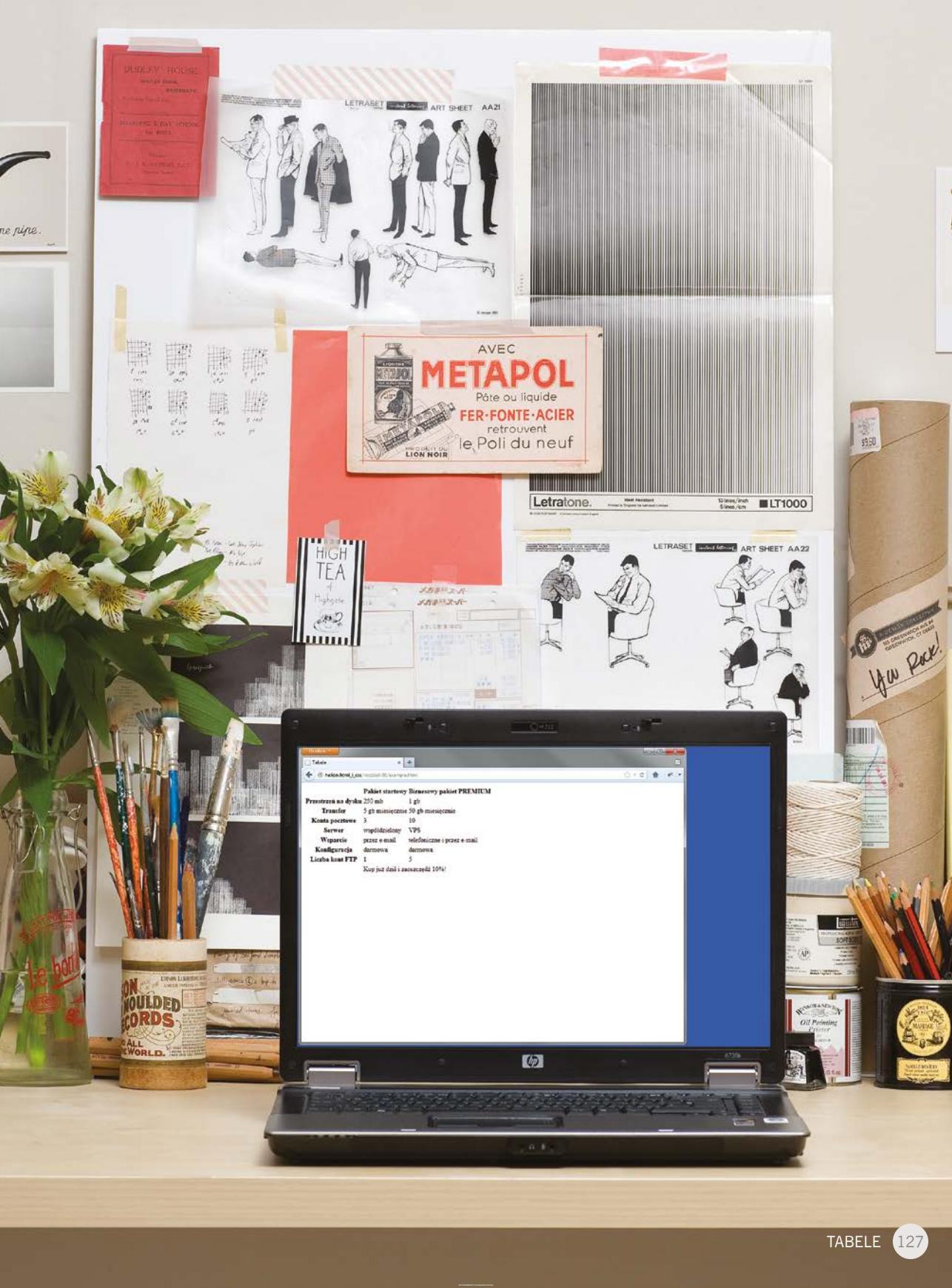
TABELE

- ▶ Sposoby tworzenia tabel.
- ▶ Jakie informacje nadają się do przedstawienia w tabelach?
- ▶ Jak przedstawiać w tabelach złożone dane?

Można wskazać kilka typów informacji, które trzeba prezentować w siatce lub tabeli, np. wyniki sportowe, dane giełdowe, rozkład jazdy pociągów.

Przed zaprezentowaniem informacji w tabeli należy je sobie wyobrazić w formie wierszy i kolumn (przypominających arkusz kalkulacyjny). W tym rozdziale powiemy, jak:

- Tworzyć tabele przy użyciu czterech kluczowych elementów.
- Przedstawiać w tabelach złożone dane.
- Dodawać podpisy do tabel.



CZYM JEST TABELA?

Tabela prezentuje informacje zapisane w układzie siatki. Przykładami tabel mogą być raporty finansowe, program telewizyjny oraz wyniki sportowe.

Dzięki wykorzystaniu dwóch osi tabele pozwalają zrozumieć złożone dane.

Każdy blok w siatce jest nazywany **komórką tabeli**. Kod HTML tabeli jest tworzony wiersz po wierszu.

The screenshot shows the Reuters homepage with a focus on the 'Commodities' section. At the top, there's a navigation bar with 'REUTERS' logo, 'EDITION: U.S.', 'News & Markets', 'Sectors & Industries', 'Analysis & Opinion', 'Register', 'Sign In', and social media links. Below the navigation is a search bar with 'Search News & Quotes' and a 'SEARCH' button.

Commodities

Related Topics: [MARKETS](#) [BUSINESS](#) [ECONOMY](#) [GREEN BUSINESS](#) [HOT STOCKS](#) [MORE TOPICS ▾](#)

THOMSON REUTERS/JEFFERIES CRB INDEX(TR/J CRB)

Change	Open	High	Low	Times
▼ 359.42	-3.36	360.92	361.19	357.99
04/18 14:58				

Data as of 3:45pm EDT (Delayed at least 20 minutes).

COMMODITY FUTURES

- Energy:** Oil, Natural Gas, Electricity
- Metals:** Base Metals, Precious Metals
- Grains:** Corn/Maize, Wheat, Barley, Rice
- Oilseeds:** Soybeans, Rapeseed, Palm Oil
- Softs:** Sugar, Coffee, Cocoa, Rubber, Citrus, Cotton
- Livestock:** Lean Hogs, Live Cattle

THOMSON REUTERS EQUAL WEIGHT CONTINUOUS COMMODITIES INDEX (CCI)

Position of price in relation to its moving average. This chart is desined to identify cyclical turns. CCI works well in ranging markets and typically fluctuates between + 100 and - 100.

Commodity	Currency	Last	Change	% Change	Trade Date/Time
Hogs, Lean Pit CME Jun11	USD	101.28	+0.20	+0.20%	04/18 14:13
Oil, Heating New York No. 2 NYMEX May11	USD	3.19	-0.04	-1.18%	04/18 15:14
Crude Oil Light Sweet May11	USD	107.21	-2.45	-2.29%	04/18 15:14

MARKETS

U.S. EUROPE ASIA SECTORS

SPONSORED BY **Trade forex with Citi.**

Market Indices [» All Indices](#)

Enter company name or Symbol	SEARCH		
DOW	12,189.14	▼ -152.69	-1.24%
S&P 500	1,305.75	▼ -13.93	-1.06%
NASDAQ	2,731.64	▼ -33.01	-1.19%
TR US INDEX	119.44	▼ -1.47	-1.22%

Currencies [» All Currencies](#)

EUR/USD	1.4234	-1.35%
GBP/USD	1.6262	-0.40%
USD/JPY	82.600	-0.63%

Commodities [» All Commodities](#)

GOLD	1,496.20	+10.20	+0.68%
------	----------	--------	--------

PODSTAWOWA STRUKTURA TABEL

<table>

Do tworzenia tabel służy element `<table>`. Zawartość tabel jest zapisywana jako kolejne wiersze.

<tr>

Początek nowego wiersza tabeli jest oznaczany przy użyciu otwierającego znacznika `<tr>` (to skrót od angielskiego *table row* — wiersz tabeli).

Po znaczniku otwierającym umieszczany jest jeden element lub kilka elementów `<td>` (po jednym dla każdej komórki tabeli).

Na końcu wiersza umieszczana się zamykający znacznik `</tr>`.

<td>

Każda komórka tabeli jest reprezentowana przez element `<td>` (to skrót od angielskiego *table data* — dane tabeli).

Koniec każdej komórki jest oznaczany przy użyciu zamykającego znacznika `</td>`.

rozdział-06/basic-table-structure.html

HTML

```
<table>
  <tr>
    <td>15</td>
    <td>15</td>
    <td>30</td>
  </tr>
  <tr>
    <td>45</td>
    <td>60</td>
    <td>45</td>
  </tr>
  <tr>
    <td>60</td>
    <td>90</td>
    <td>90</td>
  </tr>
</table>
```

WYNIK

15	15	30
45	60	45
60	90	90

Niektóre przeglądarki automatycznie rysują ramkę wokół tabeli oraz obramowują jej poszczególne komórki. Sposoby kontroli obramowania tabel przy użyciu stylów CSS zostaną przedstawione na stronach 308 – 311 oraz 336 – 339.

NAGŁÓWKI TABEL

<th>

HTML

rozdział-06/table-headings.html

```
<table>
  <tr>
    <th></th>
    <th scope="col">Sobota</th>
    <th scope="col">Niedziela</th>
  </tr>
  <tr>
    <th scope="row">Liczba sprzedanych biletów:</th>
    <td>120</td>
    <td>135</td>
  </tr>
  <tr>
    <th scope="row">Wartość sprzedaży:</th>
    <td>600 zł</td>
    <td>675 zł</td>
  </tr>
</table>
```

WYNIK

	Sobota	Niedziela
Liczba sprzedanych biletów:	120	135
Wartość sprzedaży:	600 zł	675 zł

Element <th>, którego używa się tak samo jak elementu <td>, jest stosowany do oznaczania nagłówków — zarówno kolumny, jak i wierszy („th” to skrót od angielskiego *table heading* — nagłówek tabeli).

Nawet jeśli komórka tabeli nie ma żadnej zawartości, to i tak należy umieścić w kodzie strony element <td> lub <th>, by zaznaczyć obecność pustej komórki, w przeciwnym razie tabela może zostać wyświetlona w nieprawidłowy sposób. (W przedstawionym obok przykładzie taka pusta komórka znajduje się w pierwszym wierszu i w pierwszej kolumnie).

Stosowanie elementów <th> do oznaczania nagłówków stanowi ułatwienie dla osób korzystających z czytników ekranowych, pomaga wyszukiwarkom w indeksowaniu zawartości stron, jak również ułatwia określanie wyglądu tabel przy użyciu CSS.

Za pomocą atrybutu scope elementu <th> można określić, czy reprezentuje on nagłówek wiersza, czy kolumny. Atrybut ten może przyjmować następujące wartości: `row` — jeśli dany element jest nagłówkiem wiersza, lub `col` — jeśli jest on nagłówkiem kolumny.

ŁĄCZENIE KOMÓREK Z SĄSIADUJĄCYCH KOLUMN

Czasami pojawia się konieczność połączenia kilku komórek tabeli umieszczonych w sąsiadujących kolumnach.

Do tego celu można użyć atrybutu `colspan` elementów `<td>` i `<th>`. Jego wartością jest liczba określająca, ile kolumn ma zajmować dana komórka tabeli.

Zamieszczony obok przykład przedstawia plan zajęć będący tabelą złożoną z pięciu kolumn. Pierwsza kolumna zawiera nagłówek wiersza (reprezentującego dzień), natomiast pozostałe cztery kolumny reprezentują kolejne godziny.

Jeśli przyjrzymy się komórce zawierającej słowo geografia, zauważymy, że w jej elemencie został umieszczony atrybut `colspan` o wartości 2. Oznacza on, że komórka ta ma rozciągać się na dwie kolumny. W trzecim wierszu komórka z tekstem WF zajmuje aż trzy kolumny.

Analizując kod przykładu, można zauważyc, że w drugim i trzecim wierszu umieszczonych jest mniej elementów `<td>` i `<th>`, niż tabela ma kolumn.

rozdzial-06/spanning-columns.html

HTML

```
<table>
  <tr>
    <th></th>
    <th>9:00</th>
    <th>10:00</th>
    <th>11:00</th>
    <th>12:00</th>
  </tr>
  <tr>
    <th>Poniedziałek</th>
    <td colspan="2">geografia</td>
    <td>matematyka</td>
    <td>fizyka</td>
  </tr>
  <tr>
    <th>Czwartek</th>
    <td colspan="3">WF</td>
    <td>historia</td>
  </tr>
</table>
```

WYNIK

	9:00	10:00	11:00	12:00
Poniedziałek	geografia		matematyka	fizyka
Czwartek	WF			historia

Dzieje się tak, ponieważ kiedy komórka rozciąga się na więcej niż jedną kolumnę, elementy reprezentujące dodatkowe komórki, których miejsce zostało zajęte, nie są uwzględniane w kodzie.

W tym przykładzie zostały zastosowane dodatkowe style CSS, by wyraźnie pokazać, że komórki rozciągają się na kilka kolumn. Informacje o tym, jak stworzyć takie style, znajdują się na stronach 249 oraz 336 – 339.

ŁĄCZENIE KOMÓREK W SASIADUJĄCYCH WIERSZACH

HTML

rozdział-06/spanning-rows.html

```
<table>
  <tr>
    <th></th>
    <th>TVN</th>
    <th>Polsat</th>
    <th>TVP1</th>
  </tr>
  <tr>
    <th>19:30 - 20:00</th>
    <td rowspan="2">Film</td>
    <td>komedia</td>
    <td>dziennik</td>
  </tr>
  <tr>
    <th>20:00 - 20:30</th>
    <td>sport</td>
    <td>serial</td>
  </tr>
</table>
```

WYNIK

	TVN	Polsat	TVP1
19:30 - 20:00	Film	komedia	dziennik
20:00 - 20:30		sport	serial

Może się zdarzyć, że komórki tabeli będą musiały zająć kilka wierszy.

Do określenia liczby wierszy, jakie ma się zajmować dana komórka, można użyć atrybutu rowspan elementów <td> i <th>.

W przykładzie przedstawionym obok widać, że w programie TVN jest film, który trwa od 18:00 do 20:00, natomiast w dwóch pozostałych stacjach — Polsacie oraz TVP1 — w tym samym czasie pokazane zostaną dwa różne programy (każdy z nich będzie trwał pół godziny).

Jeśli przeanalizujemy kod przykładu, zauważymy, że ostatni element <tr> zawiera jedynie trzy komórki tabeli, choć w rzeczywistości w tym ostatnim wierszu są widoczne cztery kolumny. Dzieje się tak dlatego, że w komórce z napisem film w wierszu powyżej został umieszczony atrybut rowspan informujący, że komórka ta rozciąga się na dwa wiersze.

W tym przykładzie zostały zastosowane dodatkowe style CSS, by wyraźnie pokazać, że komórki rozciągają się na kilka wierszy. Informacje o tym, jak stworzyć takie style, znajdują się na stronach 249 oraz 336 – 339.

DŁUGIE TABELE

Język HTML posiada trzy elementy, które umożliwiają odróżnienie głównej zawartości tabeli od jej pierwszego i ostatniego wiersza (które mogą zawierać inne treści). Elementy te stanowią ułatwienie dla osób korzystających z czytników ekranowych, jak również umożliwiają nadawanie tym fragmentom tabeli odmiennego wyglądu niż jej pozostała zawartość (o czym się przekonamy w dalszej części książki, poświęconej CSS).

<thead>

Nagłówki tabeli należy umieszczać wewnątrz elementu <thead>.

<tbody>

Główna zawartość tabeli powinna być umieszczona wewnątrz elementu <tbody>.

<tfoot>

Stopkę tabeli należy umieszczać wewnątrz elementu <tfoot>.

Domyślnie przeglądarki rzadko traktują zawartość tych elementów inaczej niż zawartość jakichkolwiek innych elementów, niemniej projektanci często zmieniają ich wygląd przy wykorzystaniu CSS.

rozdział-06/long-tables.html

HTML

```
<table>
  <thead>
    <tr>
      <th>Data</th>
      <th>Przychód</th>
      <th>Wydatki</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>1 stycznia</th>
      <td>250</td>
      <td>36</td>
    </tr>
    <tr>
      <th>2 stycznia</th>
      <td>285</td>
      <td>48</td>
    </tr>
    <!-- kolejne wiersze tabeli -->
    <tr>
      <th>31 stycznia</th>
      <td>129</td>
      <td>64</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td></td>
      <td>7824</td>
      <td>1241</td>
    </tr>
  </tfoot>
</table>
```

WYNIK

Data	Przychód	Wydatki
1 stycznia	250	36
2 stycznia	285	48
3 stycznia	260	42
4 stycznia	290	38
5 stycznia	310	115
6 stycznia	168	14
7 stycznia	226	20
8 stycznia	253	37
9 stycznia	294	33
10 stycznia	216	46
11 stycznia	244	29
12 stycznia	297	32
13 stycznia	328	86
14 stycznia	215	38
15 stycznia	254	30
16 stycznia	256	27
17 stycznia	311	68
18 stycznia	212	39
19 stycznia	234	36
20 stycznia	221	43
21 stycznia	259	38
22 stycznia	246	31
23 stycznia	248	17
24 stycznia	229	45
25 stycznia	263	34
26 stycznia	258	41
27 stycznia	283	22
28 stycznia	256	30
29 stycznia	278	47
30 stycznia	251	15
31 stycznia	129	64
	7824	1241

Niektóre edytory stosowane w systemach zarządzania treścią udostępniają narzędzia ułatwiające tworzenie tabel. Jeśli pierwszy wiersz tabeli zawiera wyłącznie elementy <th>, to można zauważyc, że takie edytory automatycznie zastosują element <thead>.

Jednym z powodów wprowadzenia odrębnych elementów <thead> i <tfoot> jest ułatwienie przeglądania tabel, które są wyższe od ekranu (lub dłuższe niż jedna strona wydruku). W takich przypadkach przeglądarka może zachować widoczny nagłówek tabeli i przewijać jej zawartość. Dzięki temu użytkownicy mogą się łatwiej zorientować, w której kolumnie znajdują się poszczególne informacje (taka możliwość nie jest domyślnie zaimplementowana w żadnej aktualnie dostępnej przeglądarce).

W zamieszczonym tu przykładzie zostały wykorzystane style CSS, dzięki którym zawartość elementów <thead> oraz <tfoot> jest traktowana inaczej niż pozostałe wiersze. Więcej informacji dotyczących określenia postaci tabel przy użyciu CSS można znaleźć na stronach 308 – 311 oraz 336 – 339.

SZEROKOŚĆ I ODSTĘPY — STARY SPOSÓB

W języku HTML dostępnych jest kilka przestarzałych atrybutów, których nie należy stosować w nowo tworzonych witrynach. Na niektóre z nich jednak można natrafić podczas przeglądania kodu starszych stron WWW, dlatego przedstawimy je tutaj. Wszystkie zostały zastąpione właściwościami CSS. W otwierającym znaczniku `<table>` stosowany był atrybut `width` pozwalający określić szerokość całej tabeli. Ten sam atrybut mógł być stosowany w otwierających znacznikach `<th>` oraz `<td>` — wtedy określał szerokość danej komórki tabeli. Wartość tego atrybutu była liczbą oznaczającą szerokość tabeli lub kolumny, wyrażoną w pikselach.

Kolumny tabeli muszą tworzyć linie proste, dlatego często zdarzało się, że atrybut `width` był stosowany wyłącznie w pierwszym wierszu (w pozostałych wierszach ustawienie to było stosowane automatycznie).

W otwierającym znaczniku `<table>` można było także umieszczać atrybut `cellpadding` określający odstępy pomiędzy zawartością komórki a jej obramowaniem oraz `cellspacing` określający odstępy pomiędzy poszczególnymi komórkami tabeli. Wartości obu tych

rozdział-06/width-and-spacing.html

HTML

```
<table width="400" cellpadding="10" cellspacing="5">
<tr>
  <th width="150"></th>
  <th>Pobrania</th>
  <th>Kredyt</th>
  <th width="150">Stan</th>
</tr>
<tr>
  <th>Styczeń</th>
  <td>250,00</td>
  <td>660,50</td>
  <td>410,50</td>
</tr>
<tr>
  <th>Luty</th>
  <td>135,55</td>
  <td>895,20</td>
  <td>1170,15</td>
</tr>
</table>
```

WYNIK

	Pobrania	Kredyt	Stan
Styczeń	250,00	660,50	410,50
Luty	135,55	895,20	1170,15

atributów były podawane w pikselach.

W przedstawionym tu przykładzie zastosowane zostały style CSS, dzięki którym wyraźnie widać szerokości

komórek. Informacje dotyczące kontrolowania szerokości oraz odstępów w tabelach i komórkach przy użyciu CSS można znaleźć na stronach 302 oraz 336 – 339.

KRAWĘDZIE I KOLOR TŁA — STARY SPOSÓB

HTML

rozdział-06/border-and-background.html

```
<table border="2" bgcolor="#efefef">
<tr>
  <th width="150"></th>
  <th>Pobrania</th>
  <th>Kredyt</th>
  <th width="150" bgcolor="#cccccc">Stan</th>
</tr>
<tr>
  <th>Styczeń</th>
  <td>250,00</td>
  <td>660,50</td>
  <td bgcolor="#cccccc">410,50</td>
</tr>
<tr>
  <th>Luty</th>
  <td>135,55</td>
  <td>895,20</td>
  <td bgcolor="#cccccc">1170,15</td>
</tr>
</table>
```

WYNIK

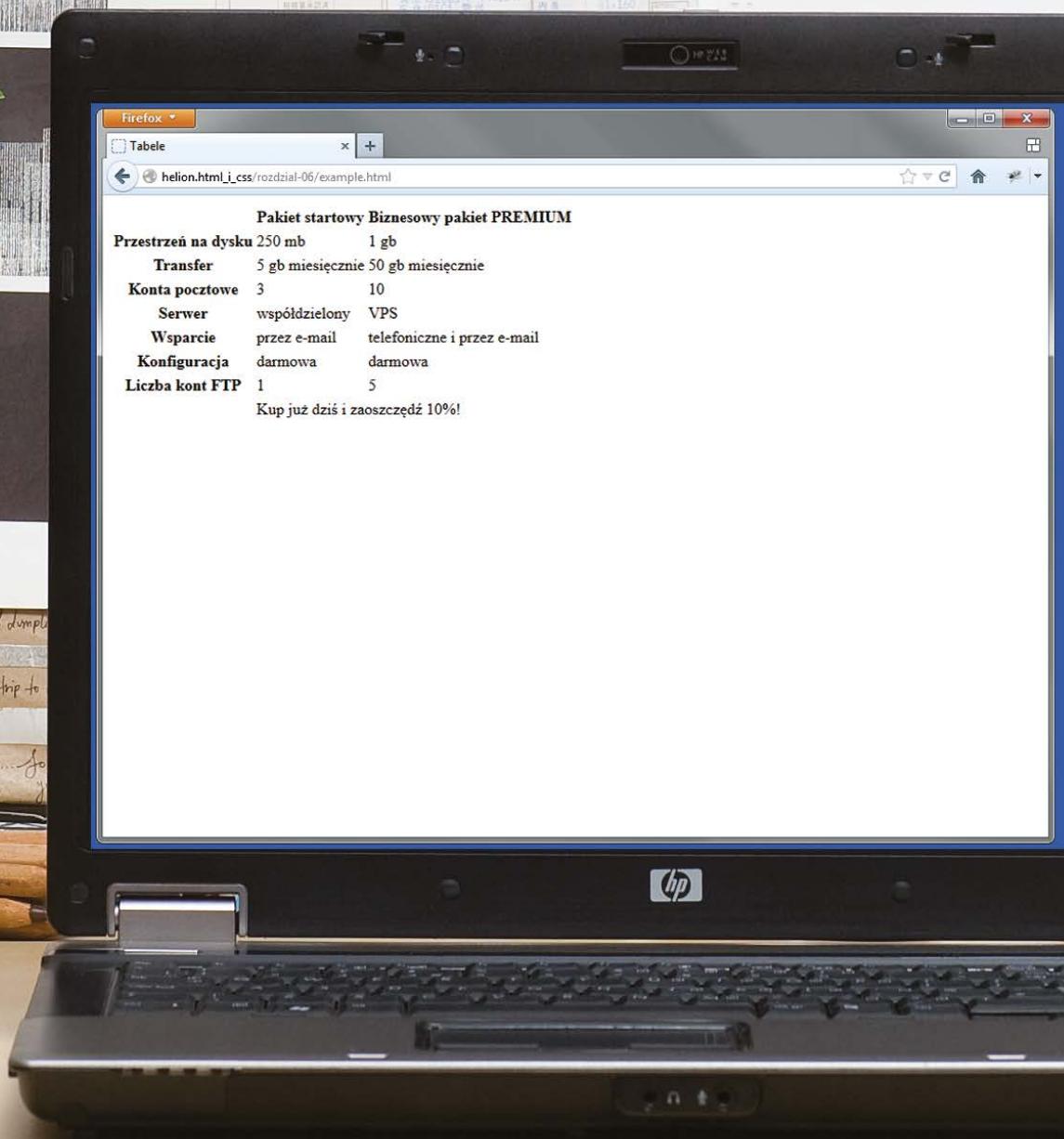
	Pobrania	Kredyt	Stan
Styczeń	250,00	660,50	410,50
Luty	135,55	895,20	1170,15

Atrybut border był używany zarówno w elementach `<table>`, jak i `<td>`, by określić szerokość obramowania (wyrażoną w pikselach).

Atrybut `bgcolor` był używany do określania koloru tła zarówno całej tabeli, jak i jej poszczególnych komórek. Jego wartością był zazwyczaj kod szesnastkowy koloru (sposoby określania kolorów zostały przedstawione na stronach 248 – 251).

W tym przykładzie zastosowano atrybuty HTML: `border` i `bgcolor`, nie zastosowano w nim natomiast żadnych stylów CSS.

Przy określaniu postaci tabel w nowo tworzonych witrynach należy korzystać wyłącznie z CSS, a nie z opisywanych tu atrybutów HTML. Zostały one przedstawione tylko dlatego, że wciąż można je znaleźć w kodzie starszych stron WWW.



Zamieszczony tu przykład przedstawia tabelę umożliwiającą porównanie dwóch wariantów usługi prowadzenia serwera WWW. W pierwszym wierszu oraz w pierwnej kolumnie tabeli

zostały umieszczone nagłówki. Pusta komórka w lewym górnym rogu tabeli także jest reprezentowana przez element `<th>`. Każda komórka tabeli musi być reprezentowana przez element `<th>` lub `<td>`.

W elementach `<th>` został zastosowany atrybut `scope` określający, czy dana komórka jest nagłówkiem kolumn, czy wierszy. W ostatnim wierszu został zastosowany atrybut `colspan`, by wyświetlony w nim tekst zajmował szerokość obu kolumn.

PRZYKŁAD

TABELE



```
<html>
  <head>
    <title>Tabele</title>
  </head>
  <body>
    <table>
      <thead>
        <tr>
          <th></th>
          <th scope="col">Pakiet startowy</th>
          <th scope="col">Biznesowy pakiet PREMIUM</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <th scope="row">Przestrzeń na dysku</th>
          <td>250 mb</td>
          <td>1 gb</td>
        </tr>
        <tr>
          <th scope="row">Transfer</th>
          <td>5 gb miesięcznie</td>
          <td>50 gb miesięcznie</td>
        </tr>
        <!-- kolejne wiersze tabeli -->
      </tbody>
      <tfoot>
        <tr>
          <td></td>
          <td colspan="2">Kup już dziś i zaoszczędź 10%!</td>
        </tr>
      </tfoot>
    </table>
  </body>
</html>
```


PODSUMOWANIE

TABELE

- ▶ Tabele można umieszczać na stronach WWW, używając elementu `<table>`.
- ▶ Tabele są wyświetlane wiersz po wierszu. Każdy wiersz jest tworzony przy użyciu elementu `<tr>`.
- ▶ Wewnątrz każdego wiersza znajduje się pewna liczba komórek reprezentowanych przez elementy `<td>` (lub `<th>` w przypadku nagłówków).
- ▶ Aby komórki tabeli zajmowały więcej niż jeden wiersz lub jedną kolumnę, należy zastosować atrybut `rowspan` lub `colspan`.
- ▶ W przypadku długich tabel ich zawartość można podzielić na trzy sekcje — `<thead>`, `<tbody>` i `<tfoot>`.

7

FORMULARZE

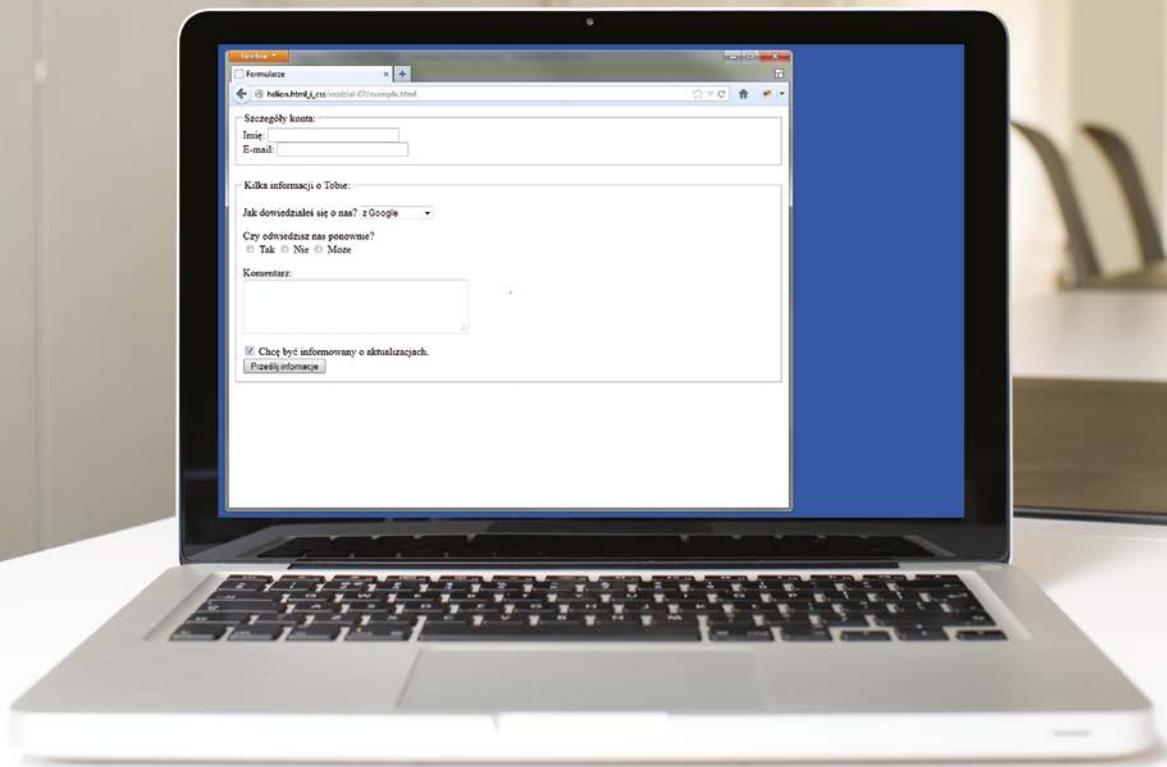
- ▶ Jak zbierać informacje od odwiedzających?
- ▶ Różne rodzaje pól formularzy.
- ▶ Nowe pola formularzy w HTML5.

Tradycyjnie termin „formularz” oznaczał drukowany dokument zawierający puste obszary, w których można było wpisywać informacje.

Język HTML zapożyczył pojęcie formularza i używa go do określania różnego typu elementów pozwalających na pobieranie informacji od osób oglądających witrynę.

Niezależnie od tego, czy dodajemy proste pole wyszukiwania, czy tworzymy złożoną aplikację ubezpieczeniową, formularze HTML udostępniają zbiór elementów umożliwiających pobieranie danych od użytkowników. W tym rozdziale zostaną opisane:

- Sposoby tworzenia formularzy na stronie WWW.
- Różne narzędzia służące do gromadzenia informacji.
- Nowe pola formularzy dostępne w HTML5.

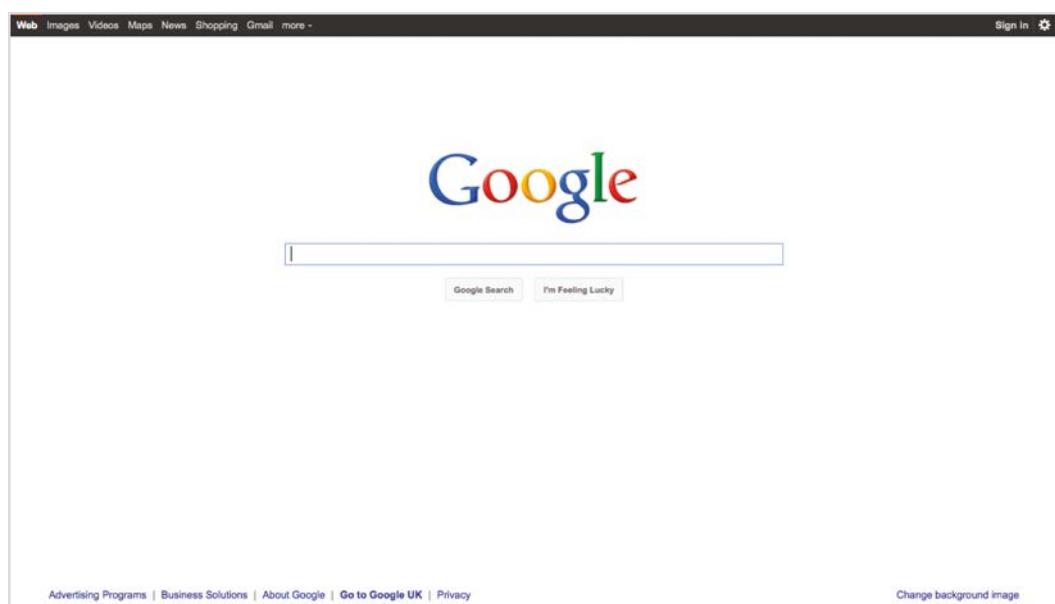


DLACZEGO FORMULARZE?

Prawdopodobnie najbardziej znanym formularzem w sieci jest pole tekstowe służące do wyszukiwania, umieszczone na środku głównej strony wyszukiwarki Google.

Oprócz zapewnienia użytkownikom możliwości wyszukiwania formularze pozwalają na wykonywanie innych czynności. Korzystamy z nich, kiedy rejestrujemy się

w nowej witrynie, kupujemy coś w sklepie internetowym, subskrybujemy jakiś biuletyn lub zapisujemy się do listy dyskusyjnej.



POLA FORMULARZY

Istnieje kilka rodzajów pól formularzy, których można używać do pobierania informacji od osób odwiedzających witryny.

DODAWANIE TEKSTU:

Pola tekstowe (jednowierszowe)

Używane do wprowadzania jednego wiersza tekstu, zawierającego np. adres poczty elektronicznej lub imię.

Pola hasła

Podobne do jednowierszowych pól tekstowych, tyle że maskują wprowadzane znaki.

Wielowierszowe pola tekstowe

Służą do tworzenia większych pól tekstowych używanych do wprowadzania długich wiadomości lub komentarzy.

DOKONYWANIE WYBORÓW:

Przyciski opcji

Stosowane, gdy użytkownik musi wybrać jedną z kilku dostępnych opcji.

 Rock Pop Jazz

Pola wyboru

Stosowane, gdy użytkownik może zaznaczyć lub usunąć zaznaczenie jednej lub kilku opcji.

 iTunes Last.fm Spotify

Listy rozwijane

Stosowane, gdy użytkownik musi wybrać jedną spośród opcji dostępnych na liście.

PRZESYŁANIE FORMULARZY:

Przycisk przesyłający

Służy do przesyłania danych z formularza na serwer.

Przycisk graficzny

Podobny do przycisku przesyłającego, można w nim stosować obrazki.

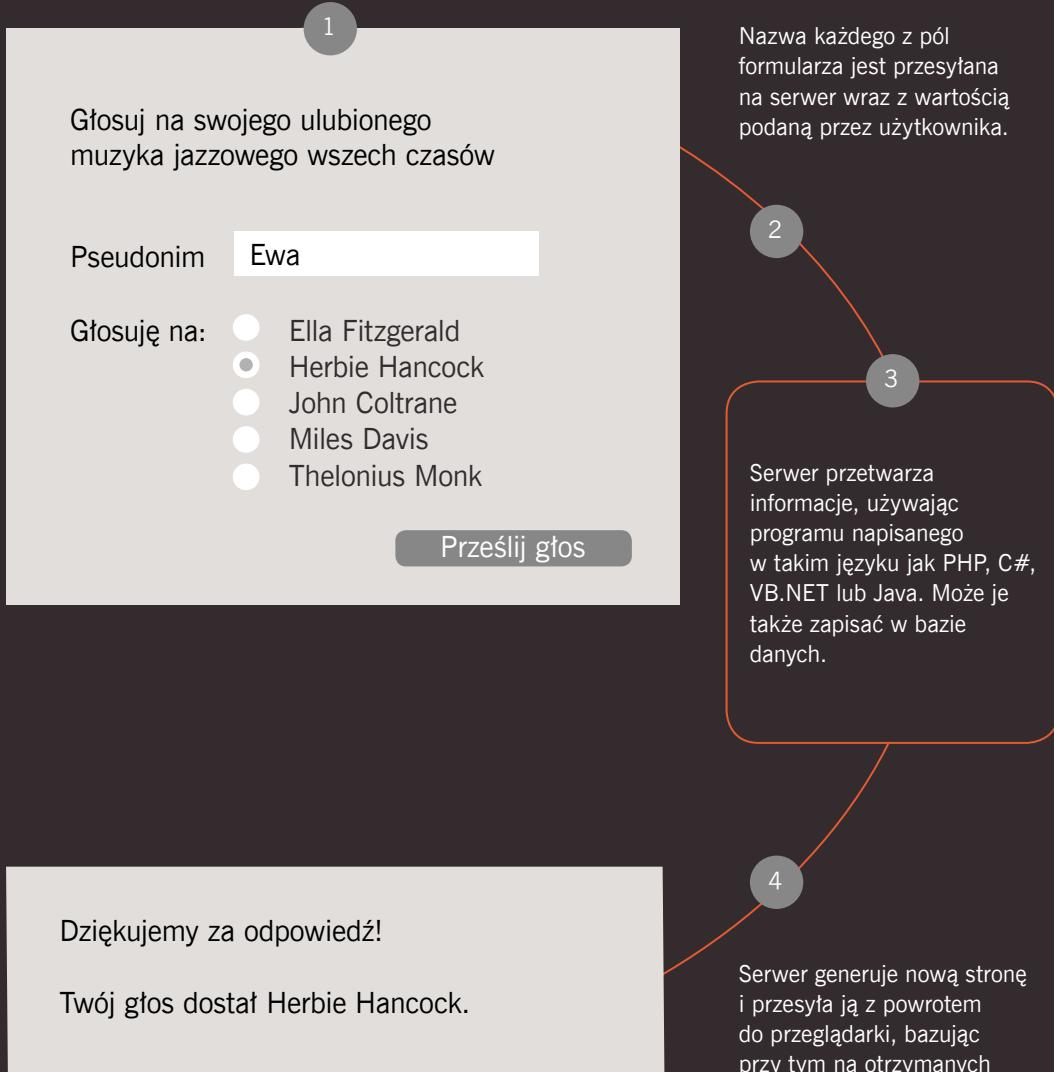
PRZESYŁANIE PLIKÓW:

Przesyłanie plików

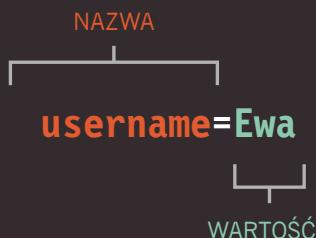
To pole pozwala użytkownikom przesyłać pliki (np. obrazki) na serwer WWW.

JAK DZIAŁAJĄ FORMULARZE?

Użytkownik wypełnia formularz, a następnie naciska przycisk, by przesyłać podane w nim informacje na serwer.



W formularzu można umieścić kilka pól, z których każde będzie zawierać inne informacje. Serwer musi wiedzieć, które z wprowadzonych danych są powiązane z danym polem formularza.



Aby odróżniać różne rodzaje wprowadzanych danych, informacje są przesyłane z przeglądarki na serwer w formie par nazwa – wartość. W przedstawionym przykładzie w formularzu poproszono użytkownika o podanie imienia oraz wybór ulubionego muzyka jazzowego. Poniżej przedstawione zostały pary nazwa – wartość przesyłane na serwer.

username=Ewa

Jeśli pole formularza pozwala na wpisanie informacji, to jego wartością będzie dowolny tekst umieszczony w nim przez użytkownika.

vote=Herbie

Jeśli pole pozwala na wybranie jednej spośród kilku konkretnych wartości (jak jest w przypadku przycisków opcji lub list rozwijanych), to autor strony sam poda kod automatycznie określający wartość danej opcji.

Nigdy nie zmieniamy nazwy pola formularza na stronie, jeśli nie jesteśmy pewni, że kod obsługujący formularz po stronie serwera prawidłowo obsłuży nową nazwę.

STRUKTURA FORMULARZY

<form>

Wszystkie pola formularzy są umieszczane wewnątrz elementu `<form>`. W tym elemencie zawsze należy określić wartość atrybutu `action`; oprócz tego zazwyczaj podawane są wartości atrybutów `method` oraz `id`.

action

Każdy element `<form>` wymaga określenia wartości atrybutu `action`. Jest nią adres URL strony na serwerze, do której w momencie wysłania formularza należy przekazać podane w nim informacje.

method

Formularze można wysyłać, używając jednej z dwóch metod: `get` albo `post`.

W przypadku stosowania metody `get` wartości pól formularza są dodawane na końcu adresu URL podanego w atrybucie `action`. Najlepiej nadaje się ona do:

- przesyłania krótkich formularzy (takich jak formularz wyszukiwania);
- formularzy służących jedynie do pobierania danych z serwera (a nie np. przesyłania informacji, które mają być zapisane lub usunięte z bazy danych).

rozdział-07/form-structure.html

HTML

```
<form action="http://www.jakaswityna.com/subscribe.php"
      method="get">
  <p>Tu będą umieszczone pola formularza.
  </p>
</form>
```

WYNIK

Tu będą umieszczone pola formularza.

W przypadku zastosowania metody `post` wartości są przesyłane w tzw. nagłówkach HTTP. Jako ogólną zasadę należy przyjąć, że metoda `post` powinna być użyta, gdy formularz:

- pozwala na przesyłanie plików;
- jest bardzo duży;
- zawiera wrażliwe dane (takie jak hasła);
- powoduje dodanie informacji do bazy danych lub usunięcie informacji z bazy danych.

Jeśli atrybut `method` nie został podany, przeglądarka prześle formularz, używając metody `get`.

id

Atrybut `id` został dokładniej opisany na stronie 182; jego wartość jest używana do unikalnego identyfikowania elementu formularza i odróżniania go od pozostałych elementów strony (bardzo często stosuje się go w skryptach, np. takich, które sprawdzają, czy w formularzu zostały podane wszystkie niezbędne dane).

POLA TEKSTOWE

<input>

HTML

rozdział-07/text-input.html

```
<form action="http://www.jakaswitryna.com/login.php">
  <p>Nazwa użytkownika:
    <input type="text" name="username" size="15"
           maxlength="30" />
  </p>
</form>
```

WYNIK

Nazwa użytkownika:

size

Atrybutu size nie należy używać w nowych formularzach. Wcześniej był on wykorzystywany w celu określenia długości pola (jego wartość określała liczbę znaków, które miały być widoczne w polu).

Użycie np. wartości 3 spowoduje utworzenie pola tekstowego o takiej szerokości, by były w nim widoczne trzy

znaki (choć w razie potrzeby użytkownik może w nim wpisać więcej znaków).

We wszystkich obecnie tworzonych formularzach szerokość pól należy określać przy użyciu arkuszy stylów CSS. Atrybut size został tu opisany tylko dlatego, że można się na niego natknąć podczas przeglądania kodu starych formularzy.

Element <input> służy do tworzenia kilku różnych rodzajów pól. Rodzaj pola, które zostanie utworzone, zależy od wartości atrybutu type tego elementu.

type="text"

Jeśli atrybut type będzie mieć wartość text, zostanie utworzone pole tekstowe.

name

Kiedy użytkownicy wypełniają formularz, serwer musi wiedzieć, jakie informacje zostały podane w poszczególnych polach (np. w przypadku formularza logowania serwer musi wiedzieć, co zostało podane jako nazwa użytkownika, a co jako hasło). Właśnie z tego powodu w każdym polu formularza należy podać wartość atrybutu name. Identyfikuje ona pole formularza i jest przesyłana wraz z jego zawartością na serwer.

maxlength

Atrybutu maxlength można używać w celu ograniczenia liczby znaków wpisywanych w polu. Jego wartość jest liczbą określającą, ile maksymalnie znaków można umieścić w danym polu. Jeśli np. oczekujemy od użytkownika podania roku, możemy umieścić w polu atrybut maxlength o wartości 4.

POLE HASŁA

<input>

type="password"

Jeśli atrybutowi type zostanie przypisana wartość password, można utworzyć pole działające podobnie do zwyczajnego, jednowierszowego pola tekstowego z tą różnicą, że wpisywane do niego znaki będą maskowane. Dzięki temu nikt, zaglądając nam przez ramię, nie będzie mógł odczytać wpisywanych w polu wrażliwych danych, takich jak hasło.

name

Atrybut name określa nazwę pola hasła, a jego wartość jest przesyłana na serwer wraz z informacjami podanymi przez użytkownika.

size, maxlength

W polach haseł można także używać atrybutów size oraz maxlength, których znaczenie jest takie samo jak w przypadku jednowierszowych pól tekstowych.

rozdział-07/password-input.html

HTML

```
<form action="http://www.jakaswitryna.com/login.php">
  <p>Nazwa użytkownika:
    <input type="text" name="username" size="15"
           maxlength="30" />
  </p>
  <p>Hasło:
    <input type="password" name="password" size="15"
           maxlength="30" />
  </p>
</form>
```

WYNIK

Nazwa użytkownika: Ewa

Hasło: ••••••••

Choć dane wpisywane w polu hasła są maskowane i nie można ich odczytać na ekranie, nie oznacza to wcale, że są one w bezpieczny sposób przesyłane na serwer. Nigdy nie należy używać tych pól do przesyłania wrażliwych informacji, takich jak numery kart kredytowych.

Aby zapewnić maksymalne bezpieczeństwo, należy skonfigurować serwer w taki sposób, by komunikował się z przeglądarką, używając protokołu Secure Socket Layer (SSL). Zagadnienia związane z SSL wykraczają poza zakres tematyczny tej książki, jednak na poświęconej jej stronie można znaleźć odnośniki do witryn zawierających więcej informacji na temat tego protokołu.

WIELOWIERSZOWE POLA TEKSTOWE

HTML

rozdział-07/textarea.html

```
<form action="http://www.jakaswitryna.com/login.php">
<p>Co myślisz o tym koncercie?</p>
<textarea name="comments" cols="30" rows="4">Tutaj
wpisz swój komentarz...</textarea>
</form>
```

WYNIK

Co myślisz o tym koncercie?

Tutaj wpisz swój komentarz...

<textarea>

Element `<textarea>` służy do tworzenia wielowierszowych pól tekstowych. W odróżnieniu od elementu `<input>` nie jest elementem pustym. Oznacza to, że powinien mieć zarówno znacznik otwierający, jak i zamkajający.

Po wczytaniu strony cały tekst umieszczony pomiędzy otwierającym znacznikiem `<textarea>` oraz zamkającym znacznikiem `</textarea>` zostanie wyświetlony w polu.

Jeśli użytkownik nie usunie żadnych znaków umieszczonych początkowo w polu, ten sam komunikat zostanie przesłany na serwer wraz z dowolnym dopisany do niego tekstem. (Na niektórych stronach używane są skrypty, które czyszczą zawartość tego pola, gdy użytkownik w nie kliknie).

W przypadku tworzenia nowych formularzy szerokość oraz wysokość elementów `<textarea>` należy określić przy użyciu arkuszy stylów CSS. Niemniej podczas analizy kodu istniejących formularzy możemy się spotkać z zastosowaniem atrybutów `cols` oraz `rows`.

Atrybut `cols` określa szerokość pola (wyrażoną jako liczba znaków), natomiast atrybut `rows` — jego wysokość (wskazuje ona liczbę wierszy tekstu, które mają być widoczne w polu).

PRZYCISKI OPCJI

<input>

type="radio"

Przyciski opcji pozwalają użytkownikom wybrać tylko jedną opcję z grupy kilku opcji.

name

Atrybut name jest przesyłany na serwer wraz z wartością opcji wybranej przez użytkownika. Jeśli odpowiedzi do pytania zadanego w formularzu są przedstawione w formie przycisków opcji, to wszystkie z nich muszą mieć taką samą wartość atrybutu name.

value

Atrybut value określa wartość, jaka zostanie przesłana na serwer w razie zaznaczenia danego przycisku opcji. Wartości wszystkich przycisków opcji należących do jednej grupy powinny być unikalne (aby serwer wiedział, który przycisk opcji został zaznaczony przez użytkownika).

checked

Atrybutu checked można użyć, by wskazać przycisk opcji, który powinien zostać zaznaczony w momencie wyświetlenia strony. Atrybut ten powinien przyjmować wartość checked. Można go umieścić tylko w jednym przycisku opcji w danej grupie.

rozdział-07/radio-button.html

HTML

```
<form action="http://www.jakaswitryna.com/profile.php">
<p>Wybierz swój ulubiony rodzaj muzyki:
<br />
<input type="radio" name="genre" value="rock"
checked="checked" /> Rock
<input type="radio" name="genre" value="pop" />
Pop
<input type="radio" name="genre" value="jazz" />
Jazz
</p>
</form>
```

WYNIK

Wybierz swój ulubiony rodzaj muzyki:

Rock Pop Jazz

Ważna uwaga: kiedy użytkownik zaznaczy jeden z przycisków opcji, nie może już usunąć tego zaznaczenia. Może co najwyżej zaznaczyć inny przycisk. Jeśli zatem dajemy użytkownikowi do wyboru

tylko jedną opcję i chcemy, by można ją było zaznaczyć lub usunąć jej zaznaczenie (jak jest np. w przypadku pól służących do akceptacji warunków rejestracji), to powinniśmy skorzystać z pól wyboru.

POLA WYBORU

HTML

rozdział-07/checkbox.html

```
<form action="http://www.jakaswitryna.com/profile.php">
<p>Wybierz swoją ulubioną usługę muzyczną:
<br />
<input type="checkbox" name="service"
       value="itunes" checked="checked" /> iTunes
<input type="checkbox" name="service"
       value="lastfm" /> Last.fm
<input type="checkbox" name="service"
       value="spotify" /> Spotify
</p>
</form>
```

WYNIK

Wybierz swoją ulubioną usługę muzyczną:
 iTunes Last.fm Spotify

<input>

type="checkbox"

Pola wyboru umożliwiają użytkownikom zaznaczanie (i usuwanie zaznaczenia) dowolnej liczby odpowiedzi na zadane pytanie.

name

Atrybut name jest przesyłany na serwer wraz z wartościami pól zaznaczonych przez użytkownika. Jeśli odpowiedzi do konkretnego pytania zadanego w formularzu są przedstawione w formie pól wyboru, to wszystkie muszą mieć taką samą wartość atrybutu name.

value

Atrybut value określa wartość, jaka zostanie przesłana na serwer w razie zaznaczenia danego pola wyboru.

checked

Atrybutu checked można użyć, by wskazać pole wyboru, które powinno zostać zaznaczone w momencie wyświetlenia strony. Atrybut ten powinien przyjmować wartość checked.

LISTY ROZWIJANE

<select>

Listy rozwijane pozwalają na wyświetlenie początkowo niewidocznej grupy opcji i wybranie jednej z nich.

Listy tego typu tworzy się przy użyciu elementu <select>. Wewnątrz niego umieszczane są dwa elementy <option> (lub więcej).

name

Atrybut name określa nazwę pola formularza, która jest przesyłana na serwer wraz z wartością opcji wybranej przez użytkownika.

<option>

Element <option> służy do tworzenia opcji listy, z których użytkownik może następnie wybierać. Słowa umieszczone pomiędzy otwierającym znacznikiem <option> a zamykającym znacznikiem </option> zostaną wyświetcone na liście.

value

Element <option> korzysta z atrybutu value, by określić wartość, która, wraz z nazwą pola, zostanie przesłana na serwer po zaznaczeniu konkretnej opcji listy.

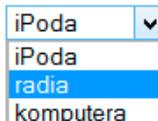
rozdział-07/drop-down-list-box.html

HTML

```
<form action="http://www.jakaswitryna.com/profile.php">
  <p>Jakiego urządzenia używasz do słuchania muzyki?</p>
  <select name="devices">
    <option value="ipod">iPoda</option>
    <option value="radio">radia</option>
    <option value="computer">komputera</option>
  </select>
</form>
```

WYNIK

Jakiego urządzenia używasz do słuchania muzyki?



selected

Atrybutu selected można użyć, by wskazać opcję, która ma być zaznaczona w momencie wyświetlenia strony. Atrybut ten powinien przyjmować wartość selected.

W razie pominięcia tego atrybutu w polu zostanie wyświetlona pierwsza podana opcja. Jeśli użytkownik nie wybierze innej opcji, to w momencie wysłania formularza na serwer zostanie przesłana wartość pierwszej opcji.

Lista rozwijana pełni podobną funkcję jak przyciski opcji (pozwala na wybór tylko jednej spośród kilku opcji). Oto dwa podstawowe czynniki decydujące o tym, które z tych pól należy zastosować:

1. Jeśli wszystkie dostępne opcje muszą być widoczne, to lepszym rozwiązaniem będą przyciski opcji.
2. Jeśli liczba dostępnych opcji jest bardzo duża (np. gdy mamy wybrać jeden z krajów świata), lepszym rozwiązaniem będzie zastosowanie listy rozwijanej.

LISTA WIELOKROTNEGO WYBORU

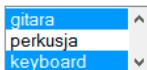
HTML

rozdział-07/multiple-select-box.html

```
<form action="http://www.jakaswitryna.com/profile.php">
<p>Czy grasz na następujących instrumentach?
(Możesz zaznaczyć więcej niż jedną opcję,
naciskając klawisz Ctrl na komputerach PC
lub klawisz Command na komputerach Mac).</p>
<select name="instruments" size="3"
multiple="multiple">
<option value="guitar" selected="selected">
gitara</option>
<option value="drums">perkusja</option>
<option value="keyboard"
selected="selected">keyboard</option>
<option value="bass">gitara basowa</option>
</select>
</form>
```

WYNIK

Czy grasz na następujących instrumentach? (Możesz zaznaczyć więcej niż jedną opcję, naciskając klawisz Ctrl na komputerach PC lub klawisz Command na komputerach Mac).



Dobrym rozwiązaniem jest poinformowanie użytkowników o możliwości zaznaczenia większej liczby opcji na liście. Warto także zaznaczyć, że na komputerach PC trzeba w tym celu wcisnąć klawisz *Ctrl* i kliknąć wybrane opcje, natomiast na komputerach Mac — wcisnąć klawisz *Command*.

<select>

size

Listę rozwijaną można zmienić w listę prezentującą jednocześnie więcej niż jedną opcję — wystarczy w elemencie `<select>` umieścić atrybut `size`. Wartością tego atrybutu jest liczba określająca, ile opcji ma być jednocześnie widocznych na liście. W przedstawionym przykładzie widać, że na liście widoczne są trzy spośród czterech opcji dostępnych na liście.

Niestety, implementacja tego atrybutu w różnych przeglądarkach nie jest idealna, dlatego w razie jego stosowania należy go dokładnie przetestować (zwłaszcza w przeglądarkach Firefox oraz Safari na komputerach Mac).

multiple

Aby umożliwić użytkownikom wybór kilku spośród opcji dostępnych na liście, należy dodać do elementu `<select>` atrybut `multiple` o wartości `multiple`.

POLE PRZESYŁANIA PLIKÓW

<input>

Jeśli chcemy zapewnić użytkownikom możliwość przesyłania na serwer plików (takich jak obrazki, klipy wideo, pliki mp3 lub PDF), to konieczne będzie umieszczenie w formularzu odpowiedniego pola.

type="file"

Zastosowanie tej wartości atrybutu type powoduje utworzenie pola przypominającego zwyczajne jednowierszowe pole tekstowe oraz umieszczonego obok niego przycisku *Przeglądaj*. (Jedynym wyjątkiem jest przeglądarka Safari na komputerach Mac — zazwyczaj wyświetla ona przycisk *Wybierz plik*, za którym umieszczone jest pole tekstowe; początkowo, gdy żaden plik nie został wybrany, jest w nim wyświetlany tekst „Brak zaznaczonych plików”, natomiast gdy użytkownik wybierze plik, w polu zostanie wyświetlona jego nazwa). Kiedy użytkownik kliknie ten przycisk, na ekranie zostanie wyświetlone okno dialogowe pozwalające na wybranie jednego z plików dostępnych na komputerze.

rozdział-07/file-input-box.html

HTML

```
<form action="http://www.jakaswitryna.com/upload.php"
      method="post">
  <p>Prześlij plik muzyczny w formacie MP3:</p>
  <input type="file" name="user-song" /><br />
  <input type="submit" value="Prześlij" />
</form>
```

WYNIK

Prześlij plik muzyczny w formacie MP3:

 Przeglądaj...

Jeśli formularz ma zapewniać możliwość przesyłania plików na serwer, to atrybut method elementu <form> musi mieć wartość post. (Plików nie można przesyłać przy użyciu metody get protokołu HTTP).

Kiedy użytkownik kliknie przycisk w polu przesyłania plików, wyświetlone okno dialogowe pozwalające na wybór pliku będzie miało wygląd dostosowany do używanego systemu operacyjnego. Nie ma możliwości zmiany wyglądu tego okna.

PRZYCISK PRZESYŁAJĄCY

HTML

rozdział-07/submit-button.html

```
<form action="http://www.jakaswitryna.com/profile.php">
<p>Subskrybuj nasz biuletyn informacyjny:</p>
<input type="text" name="email" />
<input type="submit" name="subscribe"
      value="Subskrybuj" />
</form>
```

WYNIK

Subskrybuj nasz biuletyn informacyjny:

<input>

type="submit"

Przyciski tego typu są używane do przesyłania formularzy na serwer.

name

Można określić wartość atrybutu name, jednak nie ma ona żadnego znaczenia.

value

Atrybut value służy do określania tekstu, który będzie wyświetlany na przycisku. Warto podać ten tekst, w przeciwnym razie na przycisku pojawi się komunikat „Wyślij zapytanie”, co w niektórych rodzajach formularzy może nie być właściwą nazwą przycisku.

W różnych przeglądarkach przycisk przesyłający może wyglądać nieco inaczej, jednak zazwyczaj jego wygląd odpowiada sposobowi prezentacji pól formularzy używanemu w danej przeglądarce. Jeśli zależy nam na określaniu wyglądu przycisków, możemy to zrobić za pomocą arkuszy stylów CSS (zgodnie z informacjami podanymi na stronie 342) lub użyć obrazka jako przycisku.

PRZYCISKI Z OBRAZEM

<image>

type="image"

Jeśli chcemy, by przycisk przesyłający formularz miał postać obrazka, to atrybutowi type elementu <input> powinniśmy przypisać wartość image. W takim przypadku można użyć atrybutów: src, width, height oraz alt, działających dokładnie tak samo jak w elemencie (opisany na stronach 98 – 99).

rozdział-07/image-button.html

HTML

```
<form action="http://www.jakaswitryna.com/profile.php">
  <p>Subskrybuj nasz biuletyn informacyjny:</p>
  <input type="text" name="email" />
  <input type="image" src="images/subscribe.jpg"
        width="100" height="20" />
</form>
```

WYNIK

Subskrybuj nasz biuletyn informacyjny:

SUBSKRYBUJ

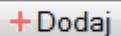
ELEMENT BUTTON ORAZ POLA UKRYTE

HTML

rozdzial-07/button-and-hidden-controls.html

```
<form action="http://www.jakaswitryna.com/profile.php">
  <button> Dodaj</button>
  <input type="hidden" name="bookmark"
    value="liryka" />
</form>
```

WYNIK

 Dodaj

<button>

Element `<button>` został wprowadzony, by zapewnić twórcom stron większą kontrolę nad wyglądem przycisków oraz pozwolić na umieszczanie na nich innych elementów.

Oznacza to, że pomiędzy otwierającym znacznikiem `<button>` a zamykającym znacznikiem `</button>` można umieścić np. tekst i obraz.

<input>

`type="hidden"`

Ten przykład przedstawia zastosowanie ukrytych pól formularzy, tworzonych za pomocą atrybutu `type` o wartości `hidden`. Pola tego typu nie są wyświetlane na stronie (choć można je zobaczyć, wyświetlając jej kod źródłowy przy użyciu opcji *Pokaż źródło strony*). Pozwalają one autorom stron na dodawanie do formularzy wartości niewidocznych dla użytkowników. Autor witryny mógłby zastosować takie pole, by np. przekazać na serwer informację o tym, która strona była wyświetlona w momencie przesyłania formularza.

DODAWANIE ETYKIET DO PÓŁ FORMULARZY

<label>

Podczas prezentowania różnych rodzajów pól formularzy celowo zastosowano uproszczony kod, w którym tekst określający przeznaczenie każdego z pól był umieszczany bezpośrednio przy nim. Jednak każdemu z pól formularzy powinien towarzyszyć element <label>, który ułatwi korzystanie z formularzy osobom mającym problemy ze wzrokiem.

Element <label> może być używany na dwa sposoby:

1. Wewnątrz niego można umieścić zarówno opis tekstowy, jak i samo pole formularza (co widać w pierwszym wierszu przedstawionego przykładu).
2. Można go oddzielić od pola formularza i jednocześnie dodać do niego atrybut for określający, z którym polem skojarzona jest dana etykieta (jak zrobiono w przypadku przycisków opcji).

for

Atrybut for określa, do którego pola formularza należy dana etykieta. Zwróćmy uwagę, że w przedstawionym przykładzie, w przyciskach opcji, zostały podane atrybuty id. Wartość tego atrybutu w jednoznaczny sposób identyfikuje element formularza, odróżniając go od wszystkich innych elementów strony (atybut id został opisany na stronie 182).

rozdział-07/labelling-form-controls.html

HTML

```
<label>Wiek: <input type="text" name="age" /></label>
<br/ >
Płeć:
<input id="female" type="radio" name="gender"
      value="f">
<label for="female">kobieta</label>
<input id="male" type="radio" name="gender"
      value="m">
<label for="male">mężczyzna</label>
```

WYNIK

Wiek:
Płeć: kobieta mężczyzna

Wartość atrybutu for odpowiada wartości atrybutu id pola, do którego odnosi się dana etykieta. Tej techniki, bazującej na wykorzystaniu atrybutów for i id, można używać zawsze, niezależnie do typu pola formularza. W przypadku gdy element <label> jest używany wraz z polami wyboru lub przyciskami opcji, użytkownik może zaznaczyć dane pole, klikając zarówno je, jak i jego etykię. Powiększenie obszaru, który użytkownik może kliknąć, ułatwia korzystanie z formularza. Bardzo duże znaczenie ma także położenie etykiety. Jeśli użytkownicy nie będą wiedzieć, gdzie należy wpisać informacje

lub jakie informacje wpisać w danym polu, zmniejsza się prawdopodobieństwo prawidłowego użycia formularza.

Oto zasady określające optymalne położenie etykiet względem pól formularzy.

POWYŻEJ LUB Z LEWEJ STRONY:

- pola tekstowe,
- wielowierszowe pola tekstowe,
- listy rozwijane,
- pola przesyłania plików.

Z PRAWEJ STRONY:

- poszczególne pola wyboru,
- poszczególne przyciski opcji.

GRUPOWANIE ELEMENTÓW FORMULARZY

HTML

rozdział-07/grouping-form-elements.html

```
<fieldset>
  <legend>Informacje kontaktowe</legend>
  <label>Adres e-mail:<br />
  <input type="text" name="email" /></label><br />
  <label>Telefon komórkowy:<br />
  <input type="text" name="mobile" /></label><br />
  <label>Telefon:<br />
  <input type="text" name="telephone" /></label>
</fieldset>
```

WYNIK

Informacje kontaktowe

Adres e-mail:

Telefon komórkowy:

Telefon:

<fieldset>

Powiązane ze sobą pola formularza można zgrupować, umieszczając je wewnątrz elementu `<fieldset>`.

Rozwiązywanie to jest szczególnie wygodne w przypadku dłuższych formularzy.

W większości przeglądarek zawartość elementów `<fieldset>` jest wyświetlana wewnątrz obramowania, co wizualnie grupuje powiązane ze sobą elementy. Wygląd tego obramowania można określić przy użyciu arkuszy stylów CSS.

<legend>

Element `<legend>` można umieścić bezpośrednio za otwierającym znacznikiem `<fieldset>`. Zawiera on tekst, który ułatwi użytkownikowi zrozumienie przeznaczenia danej grupy pól formularza.

HTML5: WERYFIKACJA PÓŁ FORMULARZY

Prawdopodobnie Czytelnik spotkał się już z formularzami, które prezentowały użytkownikom komunikaty w przypadku, gdy któryś z ich pól nie zostało prawidłowo wypełnione. Takie działanie jest nazywane **weryfikacją pól formularzy**.

Tradycyjnie weryfikacja pól formularzy była realizowana przy wykorzystaniu języka JavaScript (którego przedstawienie wykracza poza zakres tej książki). Jednak język HTML5 udostępnia mechanizmy takiej weryfikacji, przekazując związane z nią operacje przeglądarce.

Weryfikacja pól formularzy zapewnia, że po przesłaniu formularza na serwer informacje wpisane w polach zostaną prawidłowo zrozumiane. Weryfikacja treści przed przesaniem formularza na serwer jest korzystna, gdyż:

- ogranicza czynności, które trzeba będzie wykonać na serwerze;
- sprawia, że użytkownik szybciej dowiaduje się o problemach z podanymi informacjami niż wtedy, gdy weryfikacja jest przeprowadzana na serwerze.

rozdział-07/html5-form-validation.html

HTML

```
<form action="http://www.jakaswitryna.com/login.php"
      method="post">
  <label for="username">Nazwa użytkownika:</label>
  <input type="text" name="username"
         required="required" /><br />
  <label for="password">Hasło:</label>
  <input type="password" name="password"
         required="required" />
  <input type="submit" value="Zaloguj się" />
</form>
```

WYNIK

Nazwa użytkownika: Ewa
Hasło: |

Proszę wypełnić to pole.

Gdy powstawała ta książka, jedynie przeglądarki Chrome i Opera obsługiwały weryfikację pól formularzy w HTML5, choć należy się spodziewać, że inne przeglądarki pójdą w ich ślady. By zapewnić weryfikację formularzy także w starszych przeglądarkach (które nie obsługują języka HTML5), twórcy stron będą zapewne wciąż używać języka JavaScript.

W tym przykładzie weryfikacja bazuje na zastosowaniu atrybutu required — można go umieścić w dowolnym polu formularza, które użytkownik powinien wypełnić. Ten atrybut języka HTML5 nie wymaga określania wartości, co odróżnia go od atrybutów w języku HTML 4, w którym określanie wartości atrybutów było konieczne. Niektórzy twórcy stron przypisują temu atrybutowi wartość required.

HTML5: POLE DATY

HTML

rozdzial-07/html5-date-input.html

```
<form action="http://www.jakaswitryna.com/bookings"
      method="post">
  <label for="username">Data wylotu:</label>
  <input type="date" name="depart" />
  <input type="submit" value="Prześlij" />
</form>
```

WYNIK

Data wylotu:

<input>

Wiele formularzy musi pobierać takie informacje jak daty, adresy e-mail oraz adresy URL. Tradycyjnie do tego celu używane były zwyczajne pola tekstowe.

Język HTML5 jednak wprowadza nowe pola, które mają służyć standaryzacji sposobów wprowadzania niektórych informacji. Starsze przeglądarki, które nie rozpoznają tych typów pól, będą je wyświetlać jako zwyczajne, jednowierszowe pola tekstowe.

type="date"

Jeśli chcemy prosić użytkownika o podanie daty, możemy w tym celu użyć elementu <input>, w którym atrybutowi type przypisana zostanie wartość date. Dzięki temu w przeglądarkach obsługujących HTML5 wyświetcone zostanie specjalne pole służące do wyboru daty.

Ten przykład pokazuje, jak wygląda pole wyboru daty w przeglądarce Opera. W innych przeglądarkach pole to może wyglądać inaczej.

HTML5: POLA ADRESÓW E-MAIL I URL

<input>

HTML5 udostępnia pola pozwalające użytkownikom na podawanie adresów poczty elektronicznej oraz adresów URL. Przeglądarki, które nie obsługują tych typów, będą wyświetlały te pola jako zwyczajne, jednowierszowe pola tekstowe.

type="email"

Jeśli prosimy użytkownika o podanie adresu e-mail, możemy skorzystać z nowego typu pola tekstuowego. Dzięki temu przeglądarki obsługujące weryfikację danych w HTML5 będą sprawdzać, czy podane informacje zostały zapisane w prawidłowym formacie, charakterystycznym dla adresów poczty elektronicznej. Niektóre smartfony optymalizują prezentowaną klawiaturę, wyświetlając na niej przyciski, które najprawdopodobniej będą potrzebne do wpisania adresu e-mail (np. przycisk z symbolem @).

type="url"

Jeśli użytkownik ma podać adres strony WWW, można skorzystać z pola adresu URL. Dzięki temu przeglądarki obsługujące weryfikację danych w HTML5 będą mogły sprawdzić, czy użytkownik wpisał informacje w formacie

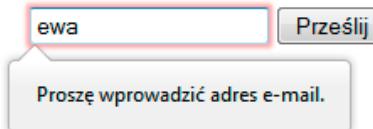
rozdział-07/html5-email-input.html

HTML

```
<form action="http://www.jakaswitryna.com/subscribe.php">
  <p>Proszę podać swój adres e-mail:</p>
  <input type="email" name="email" />
  <input type="submit" value="Prześlij" />
</form>
```

WYNIK

Proszę podać swój adres e-mail:



ewa

Prześlij

Proszę wprowadzić adres e-mail.

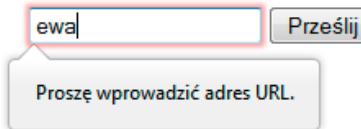
rozdział-07/html5-url-input.html

HTML

```
<form action="http://www.jakaswitryna.com/profile.php">
  <p>Proszę podać adres swojej witryny:</p>
  <input type="url" name="website" />
  <input type="submit" value="Prześlij" />
</form>
```

WYNIK

Proszę podać adres swojej witryny:



ewa

Prześlij

Proszę wprowadzić adres URL.

charakterystycznym dla adresów URL. Niektóre smartfony optymalizują

klawiaturę, wyświetlając na niej przyciski przydatne do wpisywania adresów.

HTML5: POLE WYSZUKIWANIA

HTML

rozdział-07/htm15-search-input.html

```
<form action="http://www.jakaswitryna.com/search.php">
  <p>Szukaj:</p>
  <input type="search" name="search" />
  <input type="submit" value="Szukaj" />
</form>
```

WYNIK

Szukaj:

HTML

rozdział-07/htm15-placeholder.html

```
<form action="http://www.jakaswitryna.com/search.php">
  <p>Szukaj:</p>
  <input type="search" name="search"
    placeholder="Poszukiwany tekst"/>
  <input type="submit" value="Szukaj" />
</form>
```

WYNIK

Szukaj:

placeholder

W dowolnym polu tekstowym można także użyć atrybutu `placeholder`, którego wartość określa tekst, jaki będzie prezentowany w polu do momentu, gdy użytkownik je kliknie. Starsze przeglądarki ignorują ten atrybut.

<input>

Jeśli chcemy umieścić na stronie pojedyncze pole tekstowe służące do wyszukiwania, HTML5 ułatwia nam to, udostępniając specjalny typ pola przeznaczonego właśnie do tego celu.

`type="search"`

Jeśli chcemy umieścić na stronie pojedyncze pole tekstowe służące do wyszukiwania, HTML5 ułatwia nam to, udostępniając specjalne pole wyszukiwania.

Aby utworzyć takie pole, należy do elementu `<input>` dodać atrybut `type` o wartości `search`. Starsze przeglądarki będą je traktować jako zwyczajne, jednowierszowe pole tekstowe.

Najnowsze przeglądarki rozszerzają nieco możliwości pól tego typu, poprawiając ich użyteczność. I tak w przeglądarce Safari na komputery Mac zostało dodane niewielkie kółko z krzyżykiem pozwalające wyczyścić zawartość pola, kiedy chcemy wpisać w nim zupełnie nowe słowo. Dodatkowo przeglądarka ta automatycznie zaokrąglą wierzchołki tego pola.

Firefox

Formularze

helion.html_i_css/rozdzial-07/example.html

Szczegóły konta:

Imię:

E-mail:

Kilka informacji o Tobie:

Jak dowiedziałeś się o nas?

Czy odwiedzisz nas ponownie?

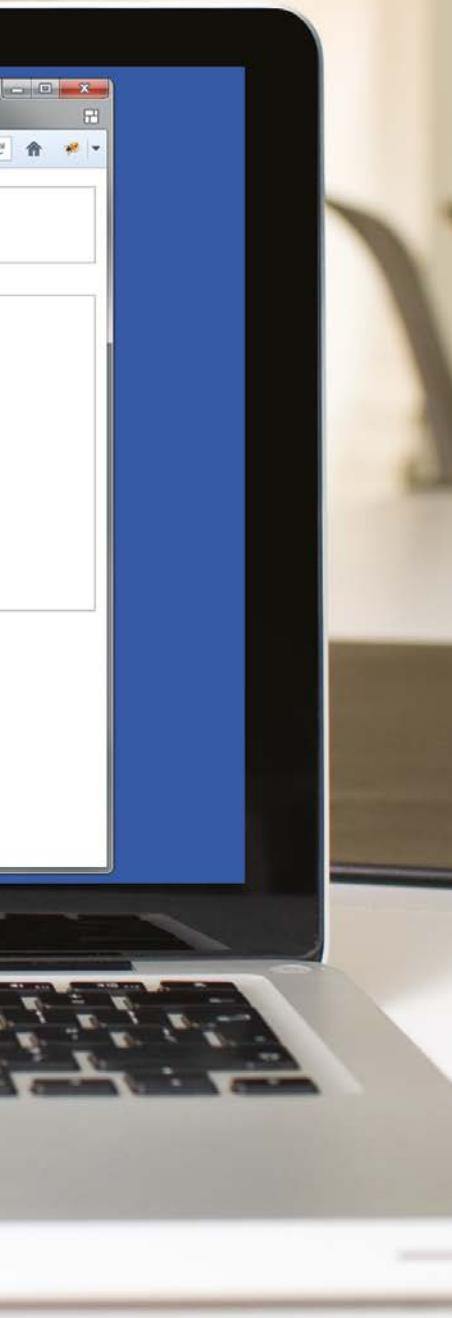
Tak Nie Może

Komentarz:

Chcę być informowany o aktualizacjach.

PRZYKŁAD

FORMULARZE



Ten przykład przedstawia formularze do przesyłania komentarzy oraz subskrypcji biuletynu informacyjnego. Wykorzystano w nich kilka typów pól.

Element `<form>` korzysta z atrybutu `action`, by określić adres strony, do której zostaną przesłane dane. Każdy element reprezentujący pole formularza jest umieszczony wewnątrz elementu `<form>`. Różne typy pól formularzy są dostosowane do pobierania różnych typów danych. Element `<fieldset>` służy do grupowania pól powiązanych ze sobą. Element `<label>` określa przeznaczenie każdego z pól formularza.

PRZYKŁAD

FORMULARZE

```
<html>
  <head>
    <title>Formularze</title>
  </head>
  <body>
    <form action="http://www.jakaswitryna.com/review.php" method="get">
      <fieldset>
        <legend>
          Szczegóły konta:
        </legend>
        <label>
          Imię:
          <input type="text" name="name" size="30" maxlength="100">
        </label>
        <br />
        <label>
          E-mail:
          <input type="email" name="email" size="30" maxlength="100">
        </label>
        <br />
      </fieldset>
      <br />
      <fieldset>
        <legend>
          Kilka informacji o Tobie:
        </legend>
        <p>
          <label for="hear-about">
            Jak się o nas dowiedziałeś?
          </label>
          <select name="referrer" id="hear-about">
            <option value="google">z Google</option>
            <option value="friend">od znajomego</option>
            <option value="advert">z reklamy</option>
            <option value="other">w inny sposób</option>
          </select>
        </p>
        <p>
```

PRZYKŁAD

FORMULARZE

Czy odwiedzisz nas ponownie?

```
<br />
<label>
  <input type="radio" name="rating" value="yes" />
  Tak
</label>
<label>
  <input type="radio" name="rating" value="no" />
  Nie
</label>
<label>
  <input type="radio" name="rating" value="maybe" />
  Może
</label>
</p>
<p>
  <label for="comments">
    Komentarz:
  </label>
  <br />
  <textarea rows="4" cols="40" id="comments">
  </textarea>
</p>
<label>
  <input type="checkbox" name="subscribe" checked="checked" />
  Chcę być informowany o aktualizacjach.
</label>
<br />
<input type="submit" value="Prześlij infomacje" />
</fieldset>
</form>
</body>
</html>
```


PODSUMOWANIE

FORMULARZE

- ▶ Zawsze, gdy chcemy odbierać informacje od użytkowników, musimy skorzystać z formularza umieszczonego wewnątrz znacznika `<form>`.
- ▶ Informacje podawane w formularzach są przesyłane w postaci par nazwa – wartość.
- ▶ Każde pole formularza musi mieć swoją nazwę; wpisywany w nim tekst lub wybierana w nim wartość są przesyłane na serwer.
- ▶ Język HTML5 udostępnia nowe typy pól formularzy, ułatwiające użytkownikom wprowadzanie danych.

8

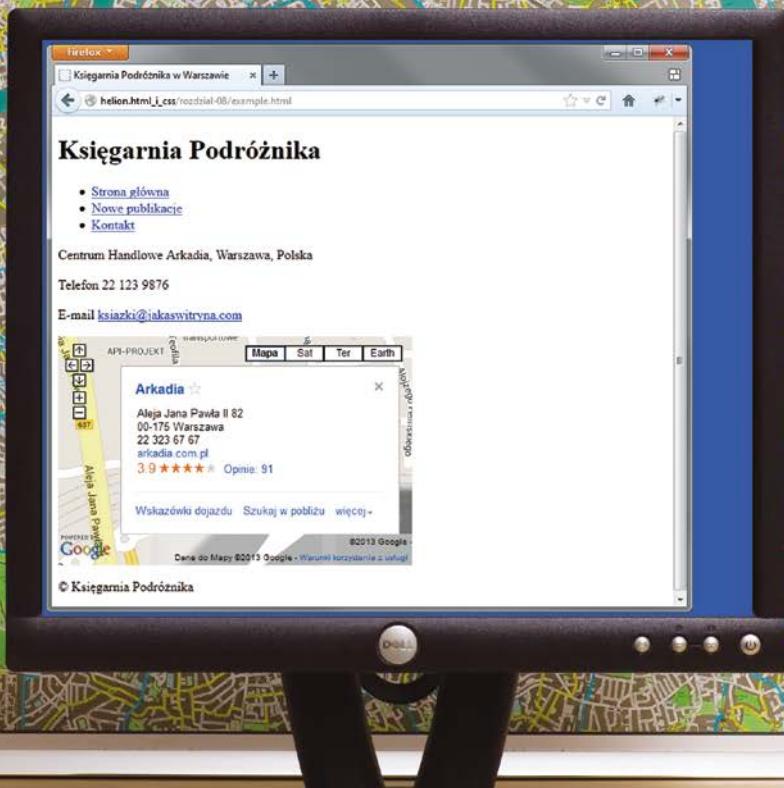
DODATKOWE ELEMENTY HTML

- ▶ Określanie różnych wersji języka HTML.
- ▶ Identyfikacja i grupowanie elementów.
- ▶ Komentarze, metainformacje oraz element `iframe`.

Jako że podstawowe znaczniki zostały już przedstawione, można je teraz podzielić na grupy i kategorie.

W tym rozdziale zostaną opisane przydatne zagadnienia, których da się łatwo przypisać do innych kategorii tematycznych. Są to:

- Różne wersje języka HTML oraz sposoby określania, która wersja języka jest używana.
- Sposoby umieszczania komentarzy na stronach WWW.
- Atrybuty globalne, czyli takie, których można używać we wszystkich elementach; przykładami takich atrybutów są `class` i `id`.
- Elementy służące do grupowania fragmentów stron, gdy żadne inne elementy się do tego nie nadają.
- Umieszczanie stron wewnętrz innych stron przy użyciu elementów `<iframe>`.
- Sposoby dodawania informacji dotyczących strony przy użyciu elementu `<meta>`.
- Dodawanie do treści stron takich znaków jak nawiasy kątowe lub znak praw autorskich.



Księgarnia Podróżnika

- Strona główna
- Nowe publikacje
- Kontakt

Centrum Handlowe Arkadia, Warszawa, Polska

Telefon 22 123 9876

E-mail ksiazki@ukaszvitryna.com



EWOLUCJA JĘZYKA HTML

Od początku istnienia internetu pojawiło się kilka wersji języka HTML.

HTML 4

WPROWADZONY
W 1997 ROKU

Każda nowa wersja była projektowana jako usprawnienie wcześniejszych (zawierała nowe elementy i atrybuty, a stare elementy były usuwane).

Jednocześnie do przeglądania stron było używanych kilka różnych wersji tej samej przeglądarki, a każda kolejna implementowała nowe możliwości. Nie wszyscy użytkownicy jednak mają zainstalowane najnowsze wersje przeglądarek, co oznacza, że nie wszyscy będą mogli się cieszyć najnowszymi możliwościami oraz kodem HTML.

W przypadkach gdy Czytelnik powinien szczególnie uważać na przeglądarki, które nie obsługują pewnych możliwości, wyraźnie zwracając na to uwagę (tak było przy nowych polach formularzy wprowadzonych w języku HTML5, opisanych w poprzednim rozdziale; podobne ostrzeżenia można znaleźć w dalszej części książki, w rozdziałach poświęconych CSS).

Oprócz kilku elementów wprowadzonych w języku HTML5 (które zostały wyróżnione) wszystkie przedstawione do tej pory były dostępne w języku HTML 4.

Choć zawierał on także kilka elementów prezentacyjnych służących do kontroli wyglądu stron WWW, to twórcy witryn internetowych nie byli zachęcani do ich stosowania. (Przykładami takich elementów są: `<center>` służący do wyśrodkowania zawartości na stronie, `` służący do określania wyglądu tekstu, `<strike>` pozwalający na przekreślenie tekstu; efekty zapewniane przez każdy z tych elementów można uzyskać, stosując arkusze stylów).

XHTML 1.0

WPROWADZONY
W 2000 ROKU

W 1998 roku został wprowadzony język XML. Jego celem było zapewnienie możliwości tworzenia nowych języków znaczników. Ponieważ najpopularniejszym językiem tego typu był HTML, podjęto decyzję, że należy go odtworzyć zgodnie z regułami języka XML i zmienić jego nazwę na XHTML. Oznaczało to, że autorzy stron muszą przestrzegać kilku nowych, nieco bardziej rygorystycznych reguł określających sposoby tworzenia kodu. Oto kilka z nich:

- Każdy element musi mieć znacznik zamkający (oprócz elementów pustych, takich jak ``).
- Nazwy atrybutów muszą być zapisywane małymi literami.
- Wszystkie atrybuty powinny mieć określoną wartość, a wartości te muszą być zapisywane w cudzysłowach.
- Elementy wycofane nie powinny być stosowane.
- Każdy element otworzony wewnętrz innego elementu powinien zostać wewnątrz niego zamknięty.

HTML5

WPROWADZONY
W 1997 ROKU

Przykłady zamieszczone w niniejszej książce są zgodne z tymi ścisłymi regułami języka XML.

Jedną z podstawowych korzyści, jakie zapewniało wprowadzenie języka XHTML, była możliwość jego bezproblemowej współpracy ze wszystkimi programami napisanymi z myślą o przetwarzaniu dokumentów XML.

Oprócz tego XHTML-a można używać wraz z danymi zapisanymi w innych formatach, takich jak Scalable Vector Graphics (SVG; język graficzny zapisywany w formie kodu XML), MathML (służący do zapisywania formuł matematycznych) czy CML (służący do zapisywania wzorów chemicznych). Aby ułatwić autorom stron WWW rozpoczęcie korzystania z nowej składni, utworzono dwa podstawowe rodzaje języka XHTML 1.0:

- **Strict XHTML 1.0**, w którym autorzy muszą stosować się do wszystkich reguł.
- **Transitional XHTML 1.0**, w którym autorzy mogli jeszcze korzystać z elementów prezentacyjnych (takich jak `<center>` lub ``).

Wersja Transitional XHTML została utworzona, gdyż pozwalała autorom stron wciąż korzystać ze starych praktyk (dzięki nieco mniej rygorystycznemu przestrzeganiu zasad dotyczących składni kodu) oraz stosować niektóre elementy i atrybuty, które miały zostać usunięte z przyszłych wersji języka HTML.

Dostępna była także trzecia wersja języka XHTML 1.0, tzw. **XHTML 1.0 Frameset**. Pozwalała ona autorom stron dzielić okno przeglądarki na kilka „ramek”, z których każda prezentowała inną stronę WWW. Aktualnie ramki spotyka się bardzo rzadko, są stopniowo wycofywane.

W języku HTML5 autorzy stron nie muszą stosować znaczników zamykających, a w przyszłości zostanie on rozszerzony o nowe elementy i atrybuty. W momencie powstawania tej książki specyfikacja HTML5 nie została jeszcze dokończona, jednak twórcy najważniejszych przeglądarek już implementują wiele spośród nowych możliwości tego języka, a autorzy stron błykawicznie zaczęli z nich korzystać.

Mimo że język HTML5 wciąż się rozwija, bezpiecznie można korzystać z wielu spośród jego nowych możliwości, o ile tylko zagwarantujemy, że użytkownicy posiadający starsze wersje przeglądarek będą mogli przeglądać nasze strony (choć niektóre z dodatkowych możliwości będą dla nich niedostępne).

DEKLARACJE DOCTYPE

Ponieważ powstało kilka wersji języka HTML, każda strona WWW powinna zaczynać się od deklaracji DOCTYPE, informującej przeglądarkę o tym, w której wersji HTML-a dana strona została napisana (choć zazwyczaj przeglądarki wyświetlają strony, nawet jeśli ta deklaracja nie została podana). Dlatego we wszystkich przykładach zamieszczonych w dalszej części książki będziemy umieszczać te deklaracje.

Później, gdy zajmiemy się CSS i ich modelem pudełkowym (na stronie 315), przekonamy się, że deklaracje DOCTYPE ułatwiają także przeglądarkom prawidłowe wyświetlanie strony.

Ponieważ język XHTML został napisany w XML-u, czasami można znaleźć strony zawierające deklarację DOCTYPE wersji Strict XHTML oraz opcjonalną deklarację XML. W razie korzystania z deklaracji XML trzeba ją umieścić na początku dokumentu — przed nią nie może się znaleźć żaden znak, nawet odstęp.

HTML5

```
<!DOCTYPE html>
```

HTML

HTML 4

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

XHTML 1.0 Transitional

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

XHTML 1.0 Strict

```
<!DOCTYPE html PUBLIC  
  "-//W3C//DTD XHTML 1.0 Strict//EN"  
  "http://www.w3.org/TR/xhtml1/DTD/  
  xhtml1-strict.dtd">
```

Deklaracja XML

```
<?xml version="1.0" ?>
```

KOMENTARZE W KODZIE HTML

HTML

rozdział-08/comments-in-html.html

```
<!-- początek wprowadzenia -->
<h1>Aktualne wystawy</h1>
<h2>Olafur Eliasson</h2>
<!-- koniec wprowadzenia -->
<!-- początek głównego tekstu -->
<p>Olafur Eliasson urodził się w 1967 roku w Danii,  
w Kopenhadze. Jego rodzice pochodzą z Islandii.</p>
<p>Jest znany rzeźbiarzem specjalizującym się  
w dużych instalacjach, w których doznania  
widza są wzmacniane poprzez wykorzystanie  
elementów żywiołowych, takich jak światło, woda  
i temperatura powietrza.</p>
<!-- koniec głównego tekstu -->
<!--
<a href="mailto:info@jakaswityna.org">Kontakt</a>
-->
```

<!-- -->

Aby dodać do kodu strony komentarz, który nie będzie widoczny po jej wyświetleniu w przeglądarce, należy zapisać jego treść pomiędzy przedstawionymi poniżej sekwencjami znaków:

<!-- to jest treść
komentarza -->

Dodawanie komentarzy do kodu jest doskonałym pomysłem, gdyż niezależnie od tego, jak dobrze rozumiemy kod strony podczas jej tworzenia, to gdy powrócimy do niego później (lub jeśli ktoś inny będzie go musiał przeanalizować), komentarze znacznie ułatwiają jego zrozumienie.

Choć komentarze nie są widoczne w głównym oknie przeglądarki, bez przeszkód można je zobaczyć po wyświetleniu kodu źródłowego strony.

W przypadku długich stron często można spotkać komentarze wskazujące, gdzie zaczynają się i kończą ich poszczególne sekcje,

jak również przekazujące uwagi dla wszystkich próbujących przeanalizować ich kod.

Wewnątrz komentarzy można także umieszczać bloki kodu, których nie chcemy wyświetlać w przeglądarce. W przedstawionym przykładzie w taki właśnie sposób zostało ukryte łącze przesyłające wiadomość poczty elektronicznej.

WYNIK

Aktualne wystawy

Olafur Eliasson

Olafur Eliasson urodził się w 1967 roku w Danii, w Kopenhadze. Jego rodzice pochodzą z Islandii.

Jest znany rzeźbiarzem specjalizującym się w dużych instalacjach, w których doznania widza są wzmacniane poprzez wykorzystanie elementów żywiołowych, takich jak światło, woda i temperatura powietrza.

ATRYBUT ID

W każdym elemencie HTML można umieścić atrybut id. Jest on używany do jednoznacznego odróżnienia danego elementu od wszystkich innych elementów strony. Wartość tego atrybutu powinna się zaczynać od litery lub znaku podkreślenia (a nie od cyfry czy jakiegoś innego znaku). Bardzo ważne jest, by w żadnych elementach strony wartości atrybutów id się nie powtarzały (w takim przypadku nie będą one unikalne).

W dalszej części książki, gdy zajmiemy się CSS, przekonamy się, że przypisywanie elementom unikalnych identyfikatorów pozwala nadawać im odmienny wygląd niż wszystkim innym elementom tego samego typu umieszczonym na stronie. Możemy np. zdecydować się na przypisanie jednemu akapitowi (który ma pełnić funkcję wyróżnionego cytatu) innego stylu niż wszystkim innym. W przedstawionym obok przykładzie akapit o identyfikatorze `pullquote` został wyróżniony poprzez wyświetlenie jego zawartości wielkimi literami.

rozdział-08/id-attribute.html

HTML

```
<p>Woda i powietrze. Są one tak wszechobecne,  
że w żaden sposób nie przyciągają naszej uwagi,  
choć jednocześnie są nam niezbędne do życia.</p>  
<p id="pullquote">Za każdym razem, gdy spoglądam  
na morze, doznaję uspokajającego poczucia  
bezpieczeństwa,  
jakbym odwiedzał dom przodków; rozpoczynam  
wizualną podróż.</p>  
<p>Tajemnica tajemnic, woda i powietrze są morzem  
widocznym tuż przed nami.</p>
```

WYNIK

Woda i powietrze. Są one tak wszechobecne, że w żaden sposób nie przyciągają naszej uwagi, choć jednocześnie są nam niezbędne do życia.

ZA KAŻDYM RAZEM, GDY SPOGLĄDAM NA MORZE
DOZNAJĘ USPOKAJAJĄCEGO POCZUCIA
BEZPIECZEŃSTWA, JAKBYM ODWIEDZAŁ DOM
PRZODKÓW; ROZPOCZYNAM WIZUALNĄ PODRÓŻ.

Tajemnica tajemnic, woda i powietrze są morzem widocznym tuż przed nami.

Jeśli Czytelnik zdecyduje się na naukę języka JavaScript (stosowanego do zapewniania interaktywności stron WWW), to atrybuty id umożliwiają skryptom operowanie na konkretnych elementach strony.

Atrybut id jest jednym z tzw. **atributów globalnych**, gdyż może być stosowany w dowolnych elementach HTML.

ATRYBUT CLASS

HTML

rozdział-08/class-attribute.html

```
<p class="important">Przez cały rok,  
począwszy od listopada 2010 roku, Muzeum  
Sztuki Współczesnej Marugame Genichiro-Inokuma  
(MIMOCA) będzie prezentować cykl wystaw Hiroshi  
Sugimoto.</p>  
<p>W ramach każdej z nich zostaną zaprezentowane  
prace powiązane z następującymi zagadnieniami  
tematycznymi: "Nauką", "Architekturą", "Historią"  
oraz "Religią"; w ten sposób można poznać  
bogaty dorobek artysty.</p>  
<p class="important admittance">Godziny zwiedzania:  
10:00 – 18:00  
(Po godzinie 17:30 wejście bezpłatne)</p>
```

WYNIK

PRZEZ CAŁY ROK, POCZĄWSZY OD LISTOPADA 2010
ROKU, MUZEUM SZTUKI WSPÓŁCZESNEJ MARUGAME
GENICHIRO-INOKUMA (MIMOCA) BĘDZIE
PREZENTOWAĆ CYKL WYSTAW HIROSHI SUGIMOTO.

W ramach każdej z nich zostaną zaprezentowane prace
powiązane z następującymi zagadnieniami tematycznymi:
"Nauką", "Architekturą", "Historią" oraz "Religią"; w ten
sposób można poznać bogaty dorobek artysty.

GODZINY ZWIEDZANIA: 10:00 – 18:00 (PO GODZINIE
17:30 WEJŚCIE BEZPŁATNE)

Domyślnie stosowanie tych
atributów nie wpływa
na prezentację elementu.
Element zmieni swój wygląd
wyłącznie wtedy, gdy będzie
istniała reguła CSS określająca,
że należy go wyświetlić inaczej.

W przykładzie przedstawionym
na tej stronie zostały
zastosowane reguły, które
sprawiają, że zawartość
elementów posiadających
atribut class o wartości
important zostanie
wyświetlona wielkimi literami,
a zawartość elementów
posiadających atribut class
o wartości admittance zostanie
wyświetlona na czerwono.

W każdym elemencie HTML
można dodać atrybut class.
Może się zdarzyć, że zamiast
identyfikować jeden,
unikalny element na stronie,
będziemy chcieli wskazać
kilka jej elementów, które
mają być traktowane inaczej
od pozostałych; np. gdy
niektóre akapity tekstu będą
ważniejsze od pozostałych
i będziemy chcieli je wyróżnić
lub zechcemy rozróżnić łącza
wskazujące różne strony
witryny od tych prowadzących
do innych witryn.

Właśnie w takich sytuacjach
można skorzystać z atrybutu
class. Jego wartość powinna
określać klasę, do jakiej należy
element. W przedstawionym
obok przykładzie najważniejsze
akapity mają atrybut class
o wartości important.

Atrybuty class różnych
elementów strony mogą mieć
tę samą wartość. A zatem
w przedstawionym przykładzie
wartość important może zostać
użyta zarówno w akapitach,
jak i łączach.

Aby zaznaczyć, że element
należy do kilku klas, wystarczy
podać ich nazwy, oddzielając
je od siebie znakami odstępu
(jak widać na przykładzie
trzeciego z przedstawionych
akapitów).

ELEMENTY BLOKOWE

Niektóre elementy wyświetlane w przeglądarce zawsze będą prezentowane od nowego wiersza. Są to tzw. **elementy blokowe**.



Przykładami takich elementów są: `<h1>`, `<p>`, `` oraz ``.

rozdział-08/block-elements.html

HTML

```
<h1>Hiroshi Sugimoto</h1>
<p>Poniżej zostały podane daty poszczególnych
wystaw z serii POCZĄTKI SZTUKI:</p>
<ul>
  <li>Nauka: 21.11.2010 - 20.02.2011</li>
  <li>Architektura: 6.03.2011 - 15.05.2011</li>
  <li>Historia: 29.05.2011 - 21.09.2011</li>
  <li>Religia: 28.09.2011 - 6.11.2011</li>
</ul>
```

WYNIK

Hiroshi Sugimoto

Poniżej zostały podane daty poszczególnych wystaw z serii POCZĄTKI SZTUKI:

- Nauka: 21.11.2010 - 20.02.2011
- Architektura: 6.03.2011 - 15.05.2011
- Historia: 29.05.2011 - 21.09.2011
- Religia: 28.09.2011 - 6.11.2011

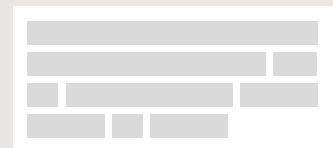
ELEMENTY WEWNĄTRZWIERSZOWE

HTML

rozdział-08/inline-elements.html

Cykl `Początki sztuki`, dopasowany do czasu trwania jednego obrotu Ziemi wokół Słońca, z nachyleniem 23,4 stopnia do płaszczyzny ekliptyki, obejmuje cztery zagadnienia tematyczne: `naukę, architekturę, historię` oraz `religię`.

Niektóre elementy zawsze będą wyświetlane w tym samym wierszu co elementy umieszczone wokół nich. Nazywamy je **elementami wewnętrzwersowymi**.



WYNIK

Cykl *Początki sztuki*, dopasowany do czasu trwania jednego obrotu Ziemi wokół Słońca, z nachyleniem 23,4 stopnia do płaszczyzny ekliptyki, obejmuje cztery zagadnienia tematyczne: **naukę, architekturę, historię** oraz **religię**.

Przykładami elementów wewnętrzwersowych są: `<a>`, ``, `` oraz ``.

GRUPOWANIE TEKSTU I ELEMENTÓW W BLOKACH

<div>

Elementy <div> pozwalają grupować dowolne inne elementy poprzez umieszczenie ich wewnątrz jednego bloku.

Można np. utworzyć element <div> zawierający wszystkie elementy wchodzące w skład nagłówka strony (takie jak logo oraz elementy nawigacyjne) lub komentarze nadsyłane przez użytkowników.

Przeglądarki wyświetlają zawartość elementów <div>, począwszy od nowego wiersza, jednak oprócz tego elementy te nie mają żadnego wpływu na prezentację strony.

Niemniej dzięki atrybutom id oraz class można tworzyć reguły CSS określające wielkość obszaru zajmowanego przez element <div> na stronie oraz modyfikujące wygląd elementów umieszczonych wewnątrz niego.

Oprócz tego umieszczanie poszczególnych sekcji strony wewnątrz odrębnych elementów <div> może ułatwić analizę jej kodu.

rozdział-08/grouping-block-elements.html

HTML

```
<div id="header">
  
  <ul>
    <li><a href="index.html">Strona główna</a></li>
    <li><a href="biography.html">Biografia</a></li>
    <li><a href="works.html">Prace</a></li>
    <li><a href="contact.html">Kontakt</a></li>
  </ul>
</div><!-- koniec nagłówka (id "header") -->
```

WYNIK

ANISH KAPOOR

- [Strona główna](#)
- [Biografia](#)
- [Prace](#)
- [Kontakt](#)

Ponieważ wewnątrz elementu <div> może się znaleźć kilka innych elementów, pomocnym rozwiązaniem będzie umieszczenie komentarza za jego znacznikiem zamykającym.

Dzięki temu bez trudu będzie można się zorientować, z którym znacznikiem otwierającym jest on powiązany (co wyraźnie widać na przedstawionym przykładzie).

GRUPOWANIE TEKSTU ORAZ ELEMENTÓW WEWNĄTRZWIERSZOWYCH

HTML

rozdział-08/grouping-inline-elements.html

```
<p>Anish Kapoor zdobył Nagrodę Turnera w 1991  
roku, a w 2003 roku prezentował swoje prace  
w Londynie, w galerii  
<span class="gallery">Tate Modern</span>.</p>
```

WYNIK

Anish Kapoor zdobył Nagrodę Turnera w 1991 roku, a w 2003 roku prezentował swoje prace w Londynie, w galerii Tate Modern.

Element `` jest wewnętrzwiernym odpowiednikiem elementu `<div>`. Może być stosowany w celu:

1. umieszczenia wewnętrzniego fragmentu tekstu, gdy nie ma żadnych innych elementów nadających się do odróżnienia go od pozostałej treści;
2. grupowania elementów wewnętrzwiernych.

Najczęstszym powodem stosowania elementów `` jest chęć kontrolowania wyglądu ich zawartości przy użyciu arkuszy stylów CSS.

Bardzo często można zauważać, że w elementach `` są umieszczane atrybuty `class` lub `id`; ma to na celu:

- określenie przeznaczenia danego elementu ``;
- umożliwienie tworzenia stylów CSS i stosowanie ich w elementach posiadających konkretne wartości tych atrybutów.

PŁYWAJĄCE RAMKI

<iframe>

Pływająca ramka (ang. *iframe* — *inline frame*) to swoiste okno umieszczone na stronie — okno, przez które można oglądać inne strony.

Popularnym zastosowaniem takich ramek (z którym Czytelnik zapewne się spotkał w wielu witrynach) jest umieszczanie na stronach map Google. Zawartością ramki może być dowolna strona WWW (umieszczona na tym samym albo innym serwerze).

Pływające ramki są tworzone przy użyciu elementu <iframe>. Posiada on kilka atrybutów, które są niezbędne do tworzenia ramek:

src

Atrybut src określa adres URL strony, która ma być wyświetlona w ramce.

height

Atrybut height określa wysokość ramki mierzoną w pikselach.

width

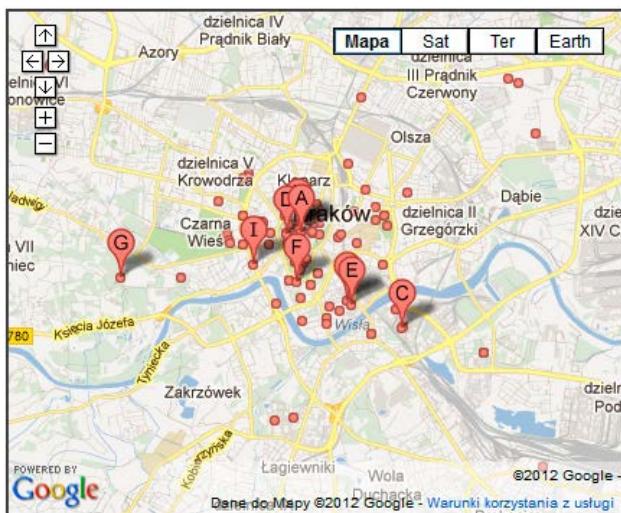
Atrybut width określa szerokość ramki mierzoną w pikselach.

rozdział-08/iframe.html

HTML

```
<iframe  
width="425"  
height="350"  
src="http://maps.google.pl/  
maps?q=krak%C3%B3w+muzeum&output=embed">  
</iframe>
```

WYNIK



scrolling

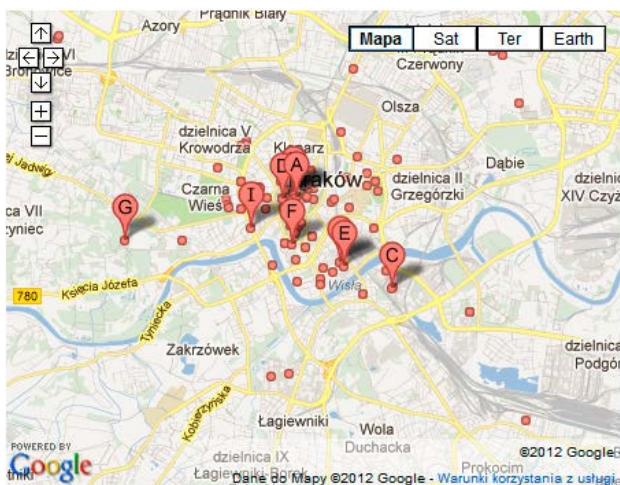
Atrybut scrolling nie będzie obsługiwanego w języku HTML5. W językach HTML 4 oraz XHTML określa on, czy ramka powinna mieć paski przewijania, czy nie. Ma to znaczenie, gdy strona wyświetlana wewnętrz ramki jest większa od wymiarów samej ramki (określonych przy użyciu atrybutów height i width). Paski przewijania pozwalają użytkownikowi przesuwać zawartość wyświetlonej wewnętrz ramki. Atrybut ten może przyjmować jedną z trzech wartości: yes (paski przewijania mają być widoczne), no (paski mają być ukryte) lub auto (by pokazywać je wyłącznie w razie konieczności).

HTML

rozdział-08/iframes-continued.html

```
<iframe  
src="http://maps.google.pl/  
maps?q=krak%C3%B3w+muzeum&output=embed"  
width="450"  
height="350"  
frameborder="0"  
scrolling="no">  
</iframe>
```

WYNIK



frameborder

Atrybut frameborder nie będzie obsługiwanego w języku HTML5. W językach HTML 4 oraz XHTML określa on, czy ramka powinna mieć obramowanie, czy nie. Wartość 1 oznacza, że obramowanie należy wyświetlić, natomiast wartość 0 — że obramowanie ma być niewidoczne.

seamless

W języku HTML5, w ramkach, w których nie należy wyświetlać pasków przewijania, można skorzystać z nowego atrybutu seamless. Atrybut ten (podobnie jak niektóre inne atrybuty języka HTML5) nie wymaga określania żadnej wartości; choć często można znaleźć strony, których autorzy przypisują mu wartość seamless. Starsze przeglądarki nie obsługują tego atrybutu.

INFORMACJE O STRONIE

<meta>

Element <meta> jest umieszczany wewnątrz elementu <head> i zawiera informacje dotyczące danej strony WWW.

Nie jest widoczny dla użytkowników przeglądających stronę, pełni jednak kilka ważnych funkcji, np. przekazuje wyszukiwarkom informacje o stronie, podaje informacje o jej autorze oraz czy jej treść przestanie być aktualna wraz z upływem czasu (jeśli przestanie, to można określić, kiedy straci ważność).

<meta> jest elementem pustym, a zatem nie musi mieć znacznika zamykającego. Wszelkie przekazywane przez niego informacje są podawane w atrybutach.

Najczęściej używanymi atrybutami tego elementu są name oraz content, które zazwyczaj stosuje się razem. Atrybuty te określają właściwości całej strony. Atrybut name podaje nazwę ustawianej właściwości, natomiast content — jej wartość.

W pierwszym wierszu przykładu na następnej stronie przedstawiony został znacznik <meta>, w którym wartość atrybutu name sugeruje chęć określenia opisu (ang. *description*) strony. Sam opis został podany jako wartość atrybutu content.

Wartość atrybutu name może być całkowicie dowolna. Poniżej przedstawione zostały niektóre spośród najczęściej używanych zdefiniowanych wartości tego atrybutu:

description

Używając tej wartości, można określić opis strony. Jest on powszechnie używany przez wyszukiwarki i pozwala im się zorientować, czego dotyczy strona. Długość opisu nie powinna przekraczać 155 znaków. Czasami jest on wyświetlany w wynikach prezentowanych przez wyszukiwarki.

keywords

Korzystając z tej wartości, można podać listę słów (oddzielonych od siebie przecinkami), których użytkownicy mogliby używać w celu odszukania danej strony. W praktyce jednak słowa te nie wywierają żadnego widocznego wpływu na wyniki wyszukiwania.

robots

Ta wartość pozwala określić, czy wyszukiwarki powinny dodawać daną stronę do wyników wyszukiwania, czy nie. Jeśli dana strona nie ma być zapamiętana przez wyszukiwarki, to należy użyć wartości noindex. Z kolei wartośćnofollow oznacza, że dana strona ma zostać zapamiętana, jednak wyszukiwarki powinny zignorować wszelkie strony, do których prowadzą umieszczone na niej łącza.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Informacje o tworzonych stronach</title>
    <meta name="description"
      content="Esej na temat instalacji artystycznych" />
    <meta name="keywords"
      content="instalacje, sztuka, opinie" />
    <meta name="robots"
      content="nofollow" />
    <meta http-equiv="author"
      content="Jan Kowalski" />
    <meta http-equiv="pragma"
      content="no-cache" />
    <meta http-equiv="expires"
      content="Fri, 04 Apr 2014 23:59:59 GMT" />
  </head>
  <body>
  </body>
</html>
```

W elemencie `<meta>` można stosować pary atrybutów `http-equiv` i `content`. W przedstawionym obok przykładzie umieszczone zostały trzy takie elementy. Każdy z nich ma inne przeznaczenie:

author

Ta wartość atrybutu `http-equiv` pozwala określić autora strony.

pragma

Użycie tej wartości zapobiega zapisywaniu strony w pamięci podręcznej przeglądarki (czyli przechowywaniu jej na lokalnym komputerze w celu zaoszczędzenia czasu podczas kolejnych wizyt na tej samej stronie).

expires

Przeglądarki często zapisują zawartość stron w pamięci podręcznej, dlatego dostępna jest wartość `expires` atrybutu `http-equiv`, pozwalająca określić moment wygaśnięcia ważności strony (po jego przekroczeniu strona nie będzie pobierana z pamięci podręcznej). Koniecznie trzeba pamiętać, by ta data była zapisana w formacie przedstawionym w przykładzie.

SYMBOLE SPECJALNE

Istnieją pewne znaki, które stosuje się w kodzie HTML, i z tego względu są zarezerwowane. Przykładem jest lewy i prawy nawias kątowy.

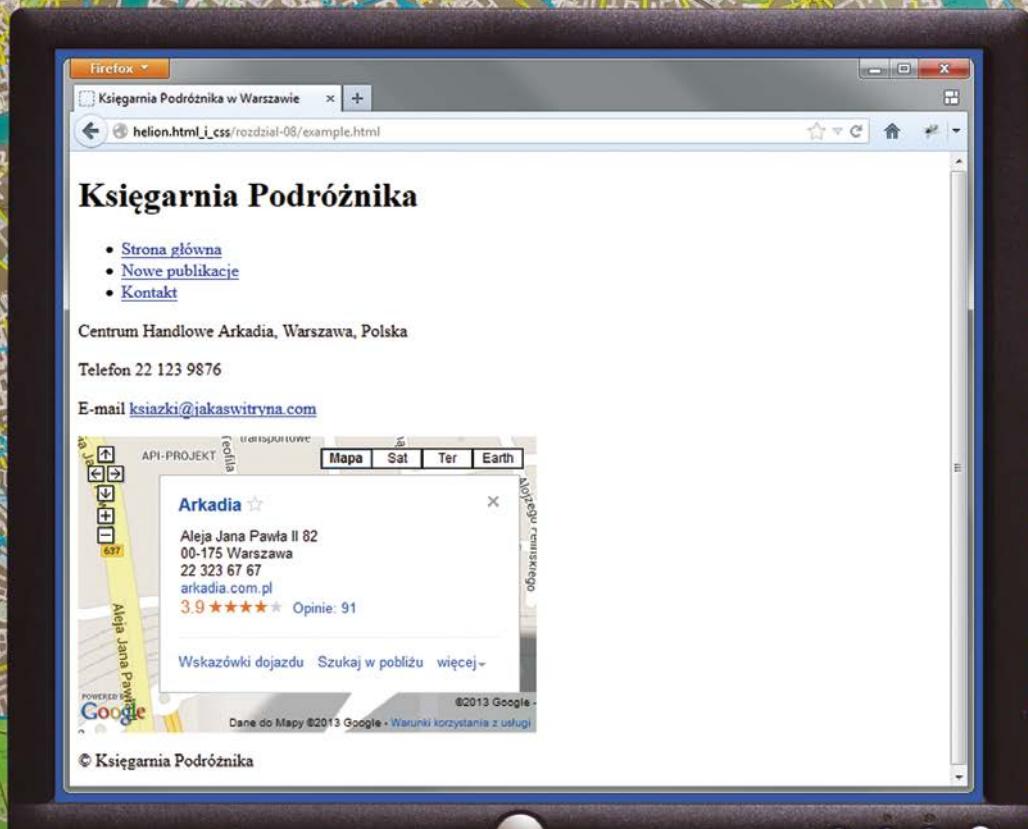
Aby wyświetlić te znaki w treści strony, konieczne jest zastosowanie symboli specjalnych. Aby np. wyświetlić lewy nawias kątowy, należy umieścić w kodzie strony symbol < lub <. Podobnie, aby wyświetlić na stronie znak &, należy użyć symbolu & lub &.

Dostępne są także symbole specjalne umożliwiające wyświetlanie innych znaków: znaku praw autorskich, symboli walut, znaków matematycznych oraz niektórych znaków przestankowych. Aby np. wyświetlić na stronie znak praw autorskich, należy użyć symbolu © lub ©.

W razie stosowania takich symboli specjalnych koniecznie należy sprawdzić stronę w przeglądarce i upewnić się, że są one prawidłowo wyświetlane. Jest to ważne, gdyż nie każdy z tych znaków jest dostępny we wszystkich krojach czcionek, dlatego może się pojawić konieczność wyświetlania ich przy użyciu innej czcionki, określonej przy użyciu stylów CSS.

WIĘCEJ W SIECI
Kompletną listę symboli specjalnych można znaleźć na stronie internetowej poświęconej tej książce.

<	znak mniejszości < <	€	znak centa ¢ ¢	‘	lewy apostrof ‘ ‘
>	znak większości > >	£	znak funta szterlinga £ £	’	prawy apostrof ’ ’
&	znak & & &	¥	znak jena ¥ ¥	“	lewy znak cudzysłowu “ “
“	znak cudzysłowu " "	€	znak euro € €	”	prawy znak cudzysłowu ” ”
		©	znak praw autorskich © ©	×	znak mnożenia × ×
		®	zarejestrowany znak handlowy ® ®	÷	znak dzielenia ÷ ÷
		™	znak handlowy ™ ™		



Przedstawiony tu przykład rozpoczyna się od deklaracji DOCTYPE informującej, że strona została napisana w języku HTML 4. W sekcji nagłówka (head) został umieszczony element <meta> opisujący jej zawartość. W kilku

elementach strony zostały umieszczone atrybuty id oraz class, określające ich przeznaczenie. Na stronie, przy użyciu odpowiedniego symbolu specjalnego, został także umieszczony znak praw autorskich.

Poszczególne fragmenty strony zostały zgrupowane przy użyciu elementów <div>, a w celu łatwej identyfikacji zamykających znaczników </div> zostały przy nich umieszczone stosowne komentarze.

PRZYKŁAD DODATKOWY

KOD HTML



```
<!DOCTYPE html PUBLIC
"-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta name="description" content="Telefon, e-mail
      oraz sposób dojazdu do Księgarni Podróżnika w Centrum
      Handlowym Arkadia w Warszawie, Polska" />
    <title>Księgarnia Podróżnika w Warszawie</title>
  </head>
  <body>
    <div id="header">
      <h1>Księgarnia Podróżnika</h1>
      <ul>
        <li><a href="index.html">Strona główna</a></li>
        <li><a href="index.html">Nowe publikacje</a>
          </li>
        <li class="current-page">
          <a href="index.html">Kontakt</a></li>
        </ul>
    </div><!-- koniec sekcji header -->
    <div id="content">
      <p>Centrum Handlowe Arkadia, Warszawa, Polska</p>
      <p><span class="contact">Telefon</span>
        22 123 9876</p>
      <p><span class="contact">E-mail</span>
        <a href="mailto:ksiazki@jakaswitryna.com">
          ksiazki@jakaswitryna.com</a></p>
      <iframe width="425" height="275" frameborder="0"
        scrolling="no" marginheight="0" marginwidth="0"
        src="http://maps.google.pl/maps?f=q&amp;
        hl=pl&amp;geocode=&amp;q=warszawa+arkadia+polska
        &amp;ie=UTF8&amp;output=embed">
      </iframe>
    </div><!-- koniec sekcji content -->
    <p>&copy; Księgarnia Podróżnika</p>
  </body>
</html>
```


PODSUMOWANIE

DODATKOWE ELEMENTY HTML

- ▶ Deklaracje DOCTYPE informują przeglądarki o tym, w której wersji języka HTML została napisana dana strona.
- ▶ Do kodu HTML można dodawać komentarze, zapisując je pomiędzy sekwencjami znaków `<!-- i -->`.
- ▶ Atrybuty `id` oraz `class` pozwalają na identyfikację konkretnych elementów.
- ▶ Elementy `<div>` oraz `` pozwalają na grupowanie odpowiednio elementów blokowych oraz wewnętrzierszych.
- ▶ Elementy `<iframe>` tworzą na stronie okna, w których można wyświetlać inne strony WWW.
- ▶ Elementy `<meta>` pozwalają na podawanie dodatkowych informacji dotyczących stron, na których się znajdują.
- ▶ Znaki takie jak `<`, `>` i `©` można wyświetlać na stronie dzięki wykorzystaniu symboli specjalnych.

9

FLASH, WIDEO I AUDIO

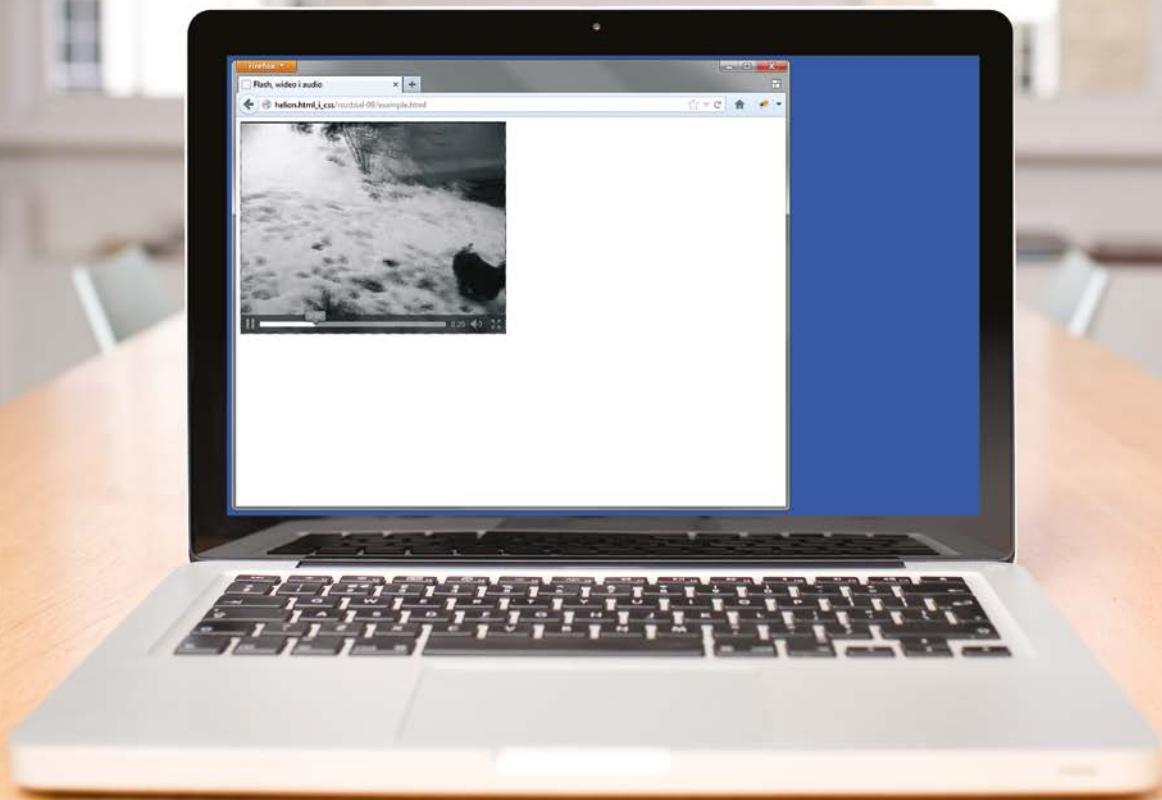
- ▶ Jak umieszczać na stronach filmy tworzone w technologii Flash?
- ▶ Jak umieszczać na stronach klipy wideo i audio?
- ▶ Elementy <video> oraz <audio> języka HTML5.

Flash jest bardzo popularną technologią, używaną w celu dodawania do stron WWW animacji, klipów wideo i audio. Ten rozdział rozpoczniemy od pokazania, w jaki sposób można jej używać na stronach WWW.

W dalszej części rozdziału skoncentrujemy się na dodawaniu do stron klipów wideo i audio zarówno przy użyciu nowych elementów HTML5 — `<video>` oraz `<audio>` — jak i korzystając z dostępnych usług (takich jak YouTube, SoundCloud).

Z tego rozdziału Czytelnik dowie się, jak:

- Używać technologii Flash na tworzonych stronach WWW.
- Korzystać z elementów `<video>` i `<audio>` HTML5.
- Zdecydować, kiedy umieszczać klipy wideo i audio we własnej witrynie, a kiedy korzystać z usług takich jak YouTube.



JAK DZIAŁA TECHNOLOGIA FLASH?

Od późnych lat 90. Flash jest bardzo popularną technologią służącą do tworzenia animacji, teraz także do odtwarzania klipów wideo i audio.

Niezależnie od tego, czy używając technologii Flash, tworzymy animację lub odtwarzacz multimedialny, pliki umieszczane na serwerze określamy jako **filmy Flash**.

Aby można było tworzyć takie filmy, konieczne jest zakupienie odpowiedniego oprogramowania firmy Adobe.

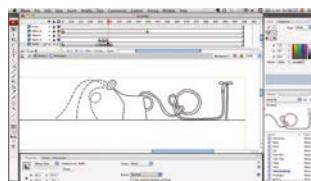
Jest kilka firm udostępniających animacje i pokazy slajdów napisane we Flashu, jak również odtwarzacze audio i wideo, których można używać bez oprogramowania firmy Adobe.

Kiedy utworzymy plik Flash w przeznaczonym do tego środowisku, jest on zapisywany z rozszerzeniem *.fla*. Jednak aby móc używać go na stronach WWW, konieczne jest zapisanie go w innym formacie, nazywanym SWF. (Pliki zapisane w tym formacie mają rozszerzenie *.swf*).

Podczas eksportowania filmu Flash do formatu SWF tworzony jest odpowiedni kod niezbędny do umieszczenia filmu na stronie WWW. Tradycyjnie kod ten korzystał ze znaczników HTML *<object>* i *<embed>*, jednak obecnie coraz częściej używa się w tym celu skryptów JavaScript.

Aby przeglądarka mogła pokazywać filmy Flash, konieczne jest użycie odpowiedniej wtyczki (czyli dodatkowego programu działającego wewnętrznie przeglądarki) o nazwie Flash Player. Statystyki wykazują, że wtyczka ta jest zainstalowana w 98% przeglądarek używanych na komputerach stacjonarnych. (W przypadku komputerów przenośnych oraz tabletów wartość ta jest znacznie niższa).

Niniejsza książka nie jest niestety dostatecznie obszerna, aby opisywać w niej sposoby tworzenia filmów Flash (istnieje wiele publikacji poświęconych wyłącznie temu zagadnieniu), jednak w tym rozdziale pokażemy, jak można umieszczać filmy Flash na tworzonych stronach WWW.



Środowisko służące do tworzenia filmów Flash.



Aby umieścić plik *.fla* na stronie WWW, należy go zapisać w formacie *.swf*.



Utworzony film Flash został wyświetlony na stronie przy użyciu skryptu JavaScript.

ZASTOSOWANIE TECHNOLOGII FLASH

Po 2005 roku pojawiło się kilka czynników, które sprawiły, że coraz mniej witryn jest pisanych całkowicie we Flashu, a wiele w ogóle nie korzysta z jakichkolwiek elementów tworzonych w tej technologii.

Początkowo technologia Flash służyła do tworzenia animacji. Później nastąpił jej szybki rozwój i zaczęto jej używać do tworzenia odtwarzaczy multimedialnych, a nawet całych witryn.

Choć wciąż jest bardzo popularna, obecnie twórcy stron ostrożniej podchodzą do jej stosowania (i rzadko kiedy decydują się na tworzenie w niej całych witryn).

Pomimo to Flash ma przed sobą długą przyszłość, gdyż wciąż są pewne rzeczy, które robi bardzo dobrze, np. animacje.

Można wskazać kilka przyczyn tego, że obecnie coraz mniej witryn jest tworzonych całkowicie we Flashu; oto niektóre z nich.

W latach 2005 – 2006 pojawiło się kilka bibliotek JavaScript (w tym Prototype, script.aculo.us, jQuery), które znacznie ułatwiają tworzenie animowanych efektów wizualnych na stronach WWW.

Kiedy w 2007 roku firma Apple wypuszczała na rynek iPhone'a, a następnie w 2010 roku — tablet iPad, podjęła decyzję, że urządzenia te nie będą obsługiwały technologii Flash.

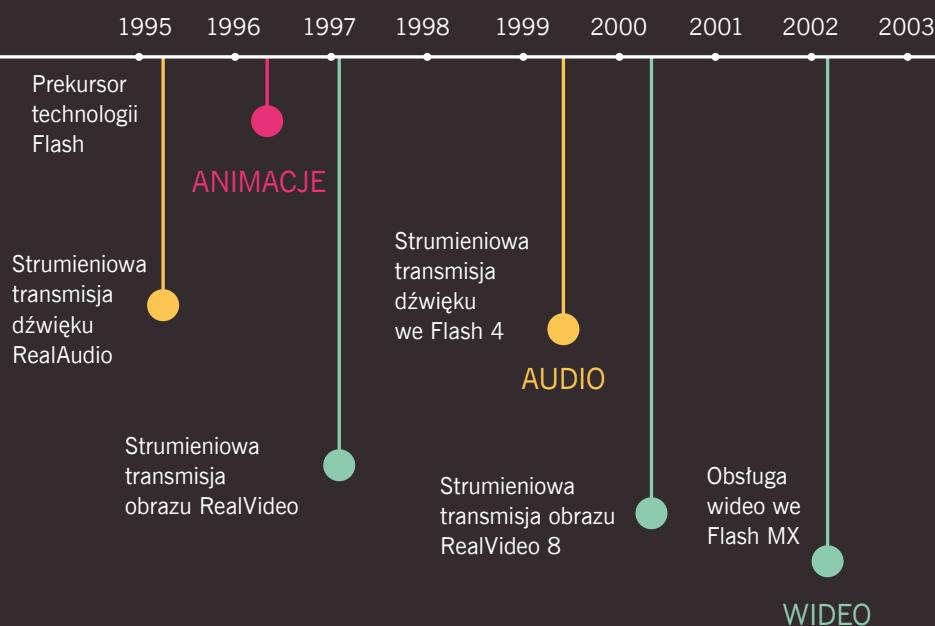
Regulacje prawne dotyczące korzystania z witryn przez osoby z upośledzeniem fizycznym lub mające problemy ze wzrokiem doprowadziły do krytyki technologii Flash, gdyż nie zawsze korzystanie z treści prezentowanej przy jej użyciu jest dostatecznie łatwe.

W 2008 roku przeglądarki zaczęły obsługiwać nowe znaczniki HTML5 — <video> i <audio>. W czasie gdy powstawała ta książka, Flash wciąż był bardzo popularną technologią, powszechnie używaną do odtwarzania materiałów wideo oraz audio, jednak coraz więcej osób decydowało się na wykorzystanie nowych możliwości HTML5.

(Sposoby użycia tych elementów zostały przedstawione w dalszej części rozdziału).

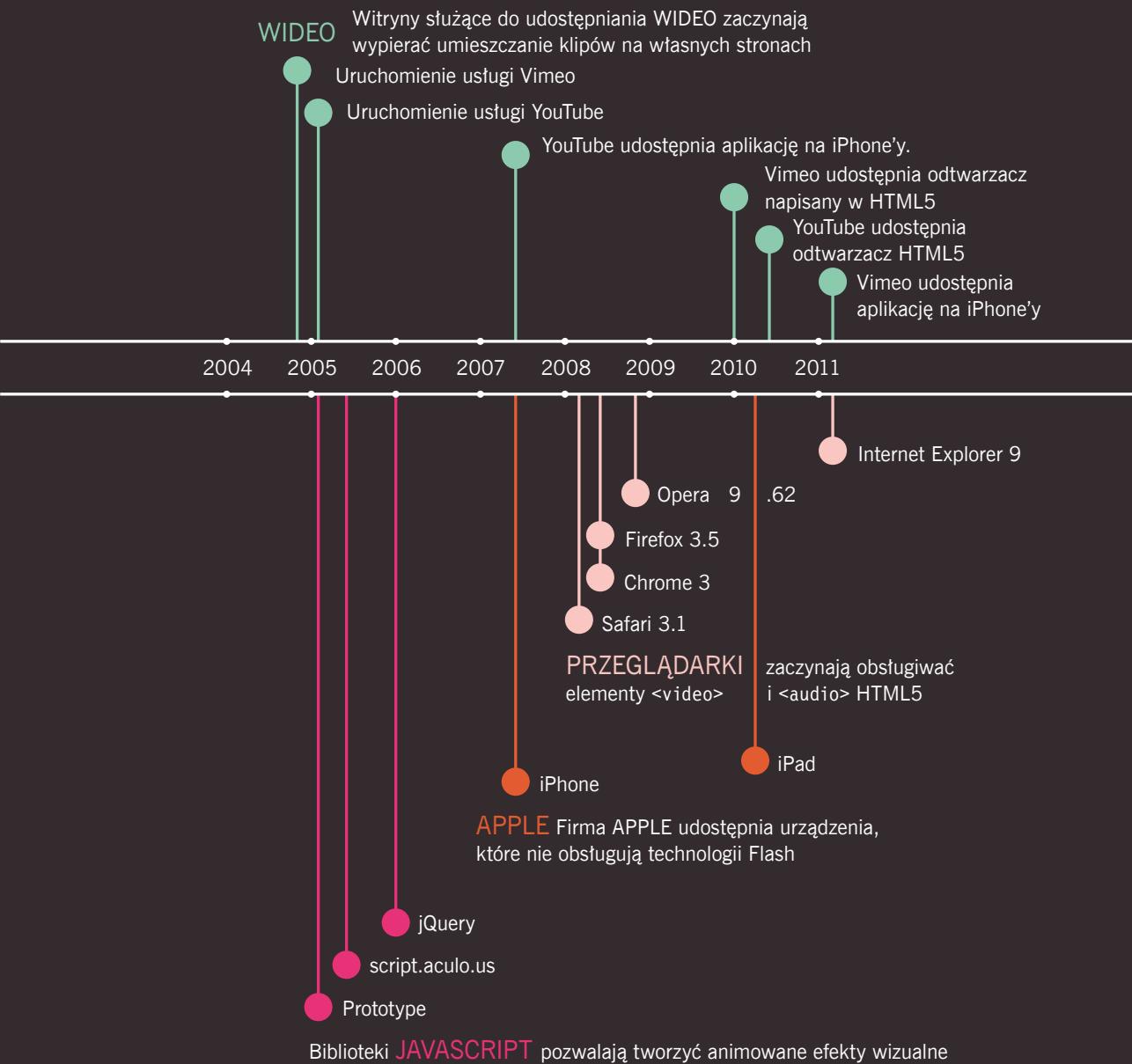
OŚ CZASU: FLASH, WIDEO I AUDIO

Technologie internetowe zmieniają się szybko. Na tej oraz następnej stronie przedstawione zostały niektóre spośród zmian w sposobach tworzenia oraz stosowania animacji, klipów wideo i audio na stronach WWW.



Na tej stronie przedstawione zostały pierwsze najważniejsze rozwiązania związane z udostępnianiem animacji, klipów wideo i audio na stronach WWW.

Na następnej stronie można zobaczyć niektóre zdarzenia i technologie, które w późniejszych latach zastąpiły początkowe rozwiązania.



UMIESZCZANIE FILMÓW FLASH NA STRONACH WWW

Najpopularniejszym sposobem umieszczania filmów Flash na stronach WWW jest użycie skryptów JavaScript. Istnieje kilka takich skryptów, z których można skorzystać bez dogłębnej znajomości języka JavaScript.

Skrypt, który przedstawimy w tej książce, nosi nazwę SWFObject. Można go pobrać bezpłatnie z witryny Google; na następnej stronie pokażemy, jak z niego korzystać.

Jedną z zalet wykorzystania skryptów jest możliwość prezentowania alternatywnych treści tym użytkownikom, których przeglądarki nie obsługują technologii Flash.

Przedstawione tu rozwiązanie bazuje na użyciu elementu `<div>` wyznaczającego obszar strony, w którym będzie prezentowany film Flash. W elemencie tym należy podać wartość atrybutu `id`, która następnie będzie używana przez skrypt SWFObject. W naszym przykładzie atrybut `id` ma wartość `bird`.

Wewnątrz elementu `<div>` można umieścić treść zastępczą, przeznaczoną dla osób, których przeglądarki nie odtwarzają filmów Flash.

rozdział-09/adding-a-flash-movie.html

HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Umieszczanie filmu Flash na stronie</title>
    <script type="text/javascript"
      src="http://ajax.googleapis.com/ajax/libs/
      swfobject/2.2/swfobject.js"></script>
    <script type="text/javascript">
      swfobject.embedSWF("flash/bird.swf",
        "bird", "400", "300", "8.0.0");</script>
  </head>
  <body>
    <div id="bird"><p>Animacja ptaszka biorącego
      prysznic.</p></div>
  </body>
</html>
```

Skrypt SWFObject sprawdzi, czy przeglądarka użytkownika może wyświetlać filmy Flash. Jeśli może, to skrypt zastąpi zawartość elementu `<div>` plikiem `.swf`.

Osoby, których przeglądarki nie obsługują technologii Flash, zamiast filmu zobaczą statyczną treść umieszczoną wewnątrz elementu `<div>`. Można tam także podać tekstowy opis filmu Flash.

Podanie tekstu opisu filmu daje dwie dodatkowe korzyści:

1. Tekst będzie dostępny dla osób mających problemy ze wzrokiem lub cierpiących na inne upośledzenia fizyczne, niemogących korzystać z filmów Flash.
2. Tekst może zostać zindeksowany przez przeglądarki (które nie radzą sobie równie dobrze z analizą zawartości plików `.swf`), co zwiększa prawdopodobieństwo odszukania strony.

WYNIK



W przedstawionym przykładzie skrypt SWFObject jest pobierany z serwerów firmy Google. Skrypt ten zostanie umieszczony na stronie przy użyciu pierwszego z dwóch elementów <script>.

W elementach <script> zostanie umieszczony atrybut type, który informuje, że skrypty napisano w języku JavaScript. Z kolei atrybut src informuje przeglądarkę, skąd należy pobrać kod skryptu.

Drugi z użytych elementów <script> przekazuje informacje dotyczące prezentowanego filmu Flash oraz elementu strony, który powinien zostać zastąpiony filmem. W rzeczywistości element ten przekazuje aż **pięć** różnych informacji zapisanych wewnątrz parów nawiasów, dotyczących:

1. **lokalizacji** pliku .swf: flash/bird.swf;
2. elementu strony, który powinien zostać **zastąpiony** filmem; określonego przez wartość atrybutu id elementu <div>: bird;
3. **szerokości** filmu: 400 px;
4. **wysokości** filmu: 300 px;
5. najniższej **wersji** odtwarzacza Flash, której można użyć do wyświetlenia filmu: Flash Player 8.

PRZEDSTAWIENIE FORMATÓW I ODTWARZACZY KLIPÓW WIDEO

Aby umieścić klip wideo na własnej stronie internetowej, trzeba zrozumieć zagadnienia związane z formatami plików oraz odtwarzaczami (wtyczkami) wideo.

FORMATY

Filmy są dostępne w wielu formatach (Blu-ray, DVD, VHS — to tylko kilka przykładów). W świecie komputerów i internetu formatów wideo jest jeszcze więcej (oto niektóre z nich: AVI, Flash Video, H264, MPEG, Ogg Theora, QuickTime, WebM, Windows Media).

Odtwarzacz DVD nie odtworzy kasety VHS — analogicznie poszczególne przeglądarki różnią się pod względem obsługiwanych formatów wideo.

Aby użytkownicy strony byli w stanie obejrzeć na niej klip wideo, być może będziemy musieli zapisać go w innym formacie.

Taki proces konwertowania klipu wideo z jednego formatu na drugi jest czasami nazywany kodowaniem.

W internecie dostępnych jest kilka aplikacji służących do kodowania wideo (np. www.mirovideoconverter.com).

ODTWARZACZE I WTYCZKI

Pierwsze przeglądarki były projektowane tak, by wyświetlać wyłącznie tekst i obrazy. Z tego powodu przeglądarki tworzone przed 2010 rokiem, by odtwarzać klipy wideo, zazwyczaj wymagały użycia dodatkowych programów nazywanych odtwarzaczami lub wtyczkami.

Te dodatkowe programy odtwarzają jedynie wybrane formaty wideo.

Nowsze przeglądarki zostały rozbudowane i obsługują element `<video>` wprowadzony w HTML5 (który sprawia, że dodatkowe odtwarzacze i wtyczki są zbyteczne).

Niestety, nie można założyć, że wszyscy użytkownicy odwiedzający naszą witrynę będą dysponowali nowoczesnymi przeglądarkami obsługującymi ten nowy element HTML5; przeglądarki, które to potrafią, wymagają, by materiał wideo był kodowany w innych formatach.

ROZWIĄZANIE

Najprostszym sposobem umieszczenia materiału wideo na własnej stronie jest skorzystanie z usługi udostępniającej klipy wideo, takiej jak YouTube lub Vimeo. Zdarzają się jednak sytuacje, w których takie rozwiązanie nie jest odpowiednie (o czym się przekonamy na następnej stronie) i trzeba umieścić klipy wideo we własnej witrynie.

W czasie gdy powstawała ta książka, najlepszym rozwiązaniem, zapewniającym, że jak największa liczba użytkowników obejrzy umieszczone na stronie materiały wideo, było zastosowanie elementu `<video>` HTML5 w przeglądarkach, które go obsługują, a w pozostałych — odtwarzacza napisanego w technologii Flash. A to oznacza, że każdy klip wideo, który planujemy udostępnić, będziemy musieli zapisać w przynajmniej dwóch formatach: WebM i MP4.

KORZYSTANIE Z USŁUG UDOSTĘPNIANIA WIDEO

Najprostszym sposobem prezentowania klipów wideo na własnych stronach jest przesyłanie ich do takiej witryny jak YouTube lub Vimeo i skorzystanie z udostępnianych przez nie mechanizmów przy osadzaniu wideo na własnej stronie.

ZALETY

Witryny umożliwiające prezentowanie klipów wideo (takie jak YouTube) udostępniają odtwarzacze działające w większości przeglądarek.

Korzystając z nich, nie trzeba się przejmować kodowaniem materiału wideo, gdyż pozwalają one na przesyłanie filmów zapisanych w wielu różnych formatach.

Po przesyłaniu materiału wideo usługi te automatycznie go konwertują do różnych formatów wymaganych przez poszczególne przeglądarki.

Firmy prowadzące serwery WWW często pobierają dodatkowe opłaty, jeśli witryna zużywa dużą szerokość pasma, a pliki zawierające filmy są ogromne. Dlatego przechowywanie i udostępnianie klipów wideo na własnej stronie może generować dodatkowe koszty. Jeśli natomiast umieścimy klipy np. na YouTube lub Vimeo, unikniemy dodatkowych kosztów związanych z używaną szerokością pasma.

WADY

Nasze klipy wideo będą umieszczone w witrynie z usługą udostępniającą, a zatem jeśli chcemy, by były one dostępne wyłącznie na naszej stronie (a nie na innych), konieczne będzie umieszczenie materiału wideo na własnym serwerze i dodanie do niej odpowiedniego odtwarzacza.

Niektóre witryny nakładają ograniczenia na dopuszczalną treść klipów wideo. Większość zabrania umieszczania w klipach materiałów reklamowych (co uniemożliwia czerpanie z nich korzyści finansowych).

Część witryn z usługami udostępniania wideo będzie odtwarzać własne reklamy przed rozpoczęciem naszego klipu, a nawet nakładać je na film podczas jego odtwarzania. Oprócz tego jakość klipów wideo oferowana przez takie usługi może być niska.

ALTERNatywy

Jeśli chcemy przechowywać materiały wideo na własnym serwerze — a nie korzystać z usługi udostępniającej — to odpowiednie przygotowanie witryny, takie by mogła je odtwarzać, będzie od nas wymagało znacznie więcej pracy.

W tej książce przyjrzymy się dwóm różnym sposobom udostępniania materiałów wideo przechowywanych na własnej stronie: Flash Video oraz nowemu elementowi HTML5 — <video>.

Aby możliwie jak największa liczba użytkowników mogła korzystać z udostępnianych klipów wideo, konieczne będzie wykorzystanie kombinacji obu tych metod.

PRZYGOTOWYWANIE MATERIAŁÓW ODTWARZANYCH PRZY UŻYCIU FLASH VIDEO

Aby prezentować materiały wideo na własnej stronie przy użyciu Flash Video, należy wykonać trzy operacje.

1

SKONWERTOWAĆ MATERIAŁ WIDEO DO FORMATU FLV

Aby skorzystać z Flash Video, trzeba skonwertować posiadany materiał wideo do formatu FLV. Począwszy od wersji 6, środowisko Flash zawiera narzędzie Flash Video Converter umożliwiające konwersję materiału wideo do formatu FLV.

Niektóre odtwarzacze filmów Flash obsługują także format H264 (a niektóre programy do edycji wideo umożliwiają zapisywanie materiału w tym formacie).

Inne programy umożliwiające konwersję materiału do tych dwóch formatów można znaleźć po wpisaniu w wyszukiwarce Google hasła „FLV or H264 converters”.

W materiałach dołączonych do książki znajduje się przykładowy plik FLV, którego można używać w dostarczonych przykładach (ponieważ pliki wideo są duże, został on umieszczony w osobnym katalogu).

2

ZNALEŹĆ ODTWARZACZ PLIKÓW FLV, KTÓRY UMOŻLIWI PREZENTOWANIE KLIPÓW

Do odtwarzania filmów FLV potrzebny będzie odtwarzacz napisany w technologii Flash. Służy on do odtwarzania filmu oraz udostępnia elementy sterujące, takie jak przyciski do odtwarzania i zatrzymywania filmu. Takie odtwarzacze można znaleźć w wymienionych poniżej witrynach:

www.osflv.com
www.longtailvideo.com

Korzystanie z tych odtwarzaczy nie wymaga kupowania środowiska Flash.

3

UMIEŚCIĆ ODTWARZACZ I MATERIAŁ WIDEO NA STRONIE

Odtwarzacz można umieścić na stronie przy użyciu technik wykorzystujących skrypty JavaScript, takich jak SWFObject (o której wspominaliśmy wcześniej w tym rozdziale).

Do odtwarzacza należy przekazać informację o lokalizacji klipu wideo, który ma być prezentowany. (Niektóre odtwarzacze dysponują zaawansowanymi możliwościami, takimi jak tworzenie list odtwarzania obejmujących wiele klipów lub wyświetlanie przed klipem statycznych obrazów).

W przedstawionym dalej przykładzie używany jest program OS FLV — darmowy odtwarzacz Flash Video udostępniany jako oprogramowanie o otwartym kodzie źródłowym. Można go także znaleźć w kodach dołączonych do książki. Obsługuje jedynie formaty FLV (nie H264).

W przedstawionym dalej przykładzie wykorzystano technikę SWFObject opisaną na stronach 206 – 207.

UMIESZCZANIE FILMÓW FLASH VIDEO NA STRONACH

HTML

rozdział-09/adding-a-flash-video.html

```
<!DOCTYPE html>
<html>
<head>
<title>Dodawanie filmów Flash Video</title>
<script type="text/javascript">
  src="http://ajax.googleapis.com/ajax/libs/
  swfobject/2.2/swfobject.js"></script>
<script type="text/javascript">
  var flashvars = {};
  var params = {movie:"..../video/puppy.flv"};
  swfobject.embedSWF("flash/splayer.swf",
  "snow", "400", "320", "8.0.0",
  flashvars, params);
</script>
</head>
<body>
<div id="snow"><p>Film o szczenięciu bawiącym się
  w śniegu.</p></div>
</body>
</html>
```

WYNIK



Przedstawiony tu przykład używa odtwarzacza OS FLV do wyświetlenie klipu o nazwie *puppy.flv*, który został już skonwertowany do formatu FLV.

Wcześniej pokazaliśmy, jak używać skryptu SWFObject w celu osadzania na stronie prostych animacji, jednak czasami do odtwarzania filmów w formacie Flash konieczne są dodatkowe informacje. W tym przypadku odtwarzacz musi znać ścieżkę dostępu do filmu, dlatego skrypt SWFObject przekazuje te informacje, wykorzystując zmienne JavaScript. Zostały one podane w dwóch wierszach kodu rozpoczynających się od słowa var:

Odtwarzacz zastosowany w tym przykładzie nie oczekuje, że jakiekolwiek informacje będą przekazywane przy użyciu zmiennej *flashvars*, dlatego jest ona pusta.

Ścieżka dostępu do filmu została podana w zmiennej *params*.

```
var params = {move: "../
  videos/puppy.flv"};
```

Wiersz kodu umieszczony poniżej zmiennych informuje skrypt o tym, że należy zastąpić element HTML odtwarzaczem. Jest on bardzo podobny do tego, którego użyliśmy we wcześniejszym przykładzie.

Zazwyczaj różne odtwarzacze wideo wymagają, by takie informacje jak ścieżka dostępu do klipu wideo były podawane w różnych formatach. Na szczęście mają one odpowiednią dokumentację z przykładami, z których można się dowiedzieć, jak należy z nich korzystać.

HTML5: PRZYGOTOWYWANIE KLIPÓW WIDEO DO PREZENTACJI NA STRONACH

Mimo że element `<video>` języka HTML5 pojawił się stosunkowo niedawno, cieszy się dużą popularnością. Oto kilka najważniejszych informacji, o których warto pamiętać, jeśli mamy zamiar z niego korzystać.

WSPARCIE

Nowy element HTML5 `<video>` jest obsługiwany wyłącznie przez najnowsze przeglądarki, a zatem, jeśli zależy nam na tym, by klipy wideo były oglądane przez wszystkich, nie będziemy mogli ograniczyć się do użycia tej jednej techniki (konieczne będzie użycie także Flash Video).

PRAWA WŁASNOŚCI

Gdy powstawała ta książka, element `<video>` nie obsługiwał żadnych systemów DRM (ang. *Digital Rights Management* — system ochrony praw autorskich dla treści cyfrowych). Niemniej zdeterminowany pirat i tak znajdzie sposób obejścia takich zabezpieczeń.

Na stronie 221 pokazano sposób, jak zastosować nowy element `<video>` HTML5 z techniką korzystającą z Flash Video w celu poprawienia dostępności strony.

FORMATY

Nie wszystkie przeglądarki obsługują te same formaty wideo. Dlatego klipy należy udostępniać w kilku różnych formatach.

Aby klipy wideo mogły być oglądane w możliwie jak największej liczbie przeglądarek, należy udostępniać filmy w następujących formatach:

H264: przeglądarki IE oraz Safari.

WebM: przeglądarki w systemie Android, przeglądarki Chrome, Firefox oraz Opera.

Twórcy przeglądarek Chrome, Firefox oraz Opera zaznaczyli, że ich programy będą obsługiwać format WebM. (Niektóre odtwarzacze bazujące na technologii Flash także obsługują formaty H264 i WebM — co pozwoli zmniejszyć liczbę niezbędnych konwersji).

Aby Czytelnik mógł wypróbować przedstawione przykłady, użyty w nich klip wideo został dostarczony w formatach H264 oraz WebM.

ELEMENTY STERUJĄCE

Przeglądarka sama udostępnia elementy sterujące odtwarzaniem, a te w poszczególnych programach mogą być różne. Wygląd elementów sterujących można określić przy użyciu kodu JavaScript (prezentacja tych możliwości wykracza poza zakres tej książki).

W PRZEGŁĄDARCE

Jednym z problemów z odtwarzaczami takimi jak Flash Player jest ich niespójne działanie w przypadku, gdy zostaną przesłonięte przez rozwijane listy lub gdy powiększymy zawartość okna. Jeśli korzystamy z elementu `<video>`, problemy te nie występują.

Jeśli przedstawiona przykładową stronę wyświetlimy w przeglądarce Firefox lub Opera, to dostępne elementy sterujące będą inne.

HTML5: UMIESZCZANIE KLIPÓW WIDEO NA STRONACH WWW

HTML

rozdział-09/adding-html5-video.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Zastosowanie elementu &lt;video&gt; HTML5</title>
  </head>
  <body>
    <video src="video/puppy.mp4"
      poster="images/puppy.jpg"
      width="400" height="300"
      preload
      controls
      loop>
      <p>Film o szczenięciu bawiącym się
      w śniegu.</p>
    </video>
  </body>
</html>
```

W języku HTML5 podawanie wartości wszystkich atrybutów, takich jak `controls`, `autoplay` oraz `loop` elementu `<video>`, nie jest konieczne. Takie atrybuty można porównać do przełączników. Jeśli atrybut został umieszczony w kodzie, włącza on odpowiednią opcję. Jeśli natomiast został pominięty, opcja zostanie wyłączona.

Jeśli przeglądarka nie obsługuje elementu `<video>` lub formatu, w jakim klip został zapisany, to wyświetli treść podaną pomiędzy otwierającym znacznikiem `<video>` i zamykającym znacznikiem `</video>`.

preload

Ten atrybut informuje przeglądarkę, co należy zrobić podczas wczytywania strony. Może on przyjąć jedną z trzech wartości:

none

Przeglądarka nie powinna rozpoczęć wczytywania klipu wideo, zanim użytkownik nie kliknie przycisku rozpoczęjącego odtwarzanie.

auto

Przeglądarka powinna rozpoczęć pobieranie klipu po zakończeniu wczytywania strony.

metadata

Przeglądarka powinna jedynie pobrać informacje o klipie, takie

jak wymiary pierwszej klatki, lista ścieżek, czas trwania.

<video>

Element `<video>` ma kilka atrybutów pozwalających na kontrolę sposobu odtwarzania:

src

Ten atrybut określa ścieżkę dostępu do pliku wideo. (Klip zastosowany w przykładzie został zapisany w formacie H264, a zatem będzie go można odtwarzać wyłącznie w przeglądarkach IE i Safari¹).

poster

Ten atrybut pozwala określić obraz, który będzie wyświetlany w czasie wczytywania klipu lub do momentu naciśnięcia przez użytkownika przycisku rozpoczęjącego odtwarzanie.

width, height

Te atrybuty określają wymiary odtwarzacza i są wyrażone w pikselach.

controls

Jeśli ten atrybut zostanie podany, informuje przeglądarkę, że powinna wyświetlić swoje elementy sterujące odtwarzaniem.

autoplay

Jeśli ten atrybut zostanie podany, informuje przeglądarkę, że powinna automatycznie rozpoczęć odtwarzanie klipu.

loop

W razie użycia tego atrybutu informuje on przeglądarkę, że po zakończeniu klipu powinna go ponownie odtworzyć.

¹ Aktualnie format ten jest obsługiwany także przez przeglądarkę Chrome — przyp. tłum.

HTML5: WIELE ŹRÓDEŁ MATERIAŁU WIDEO

<source>

Do określenia pliku zawierającego klip wideo można użyć elementu `<source>` umieszczonego wewnątrz elementu `<video>`. (Powiniene on zastąpić atrybut `src` elementu `<video>`).

Można także użyć kilku elementów `<source>`, informując tym samym przeglądarkę, że klip jest dostępny w różnych formatach. (Ze względu na błąd występujący na tabletach iPad w pierwszej kolejności należy podać klip zapisany w formacie MP4. W przeciwnym razie nie zostanie odtworzony).

src

Ten atrybut określa ścieżkę dostępu do pliku.

type

Tego atrybutu należy używać, by przekazywać informacje o formacie, w jakim jest zapisany klip wideo. W razie pominięcia atrybutu przeglądarka pobierze któryś z podanych plików, by sprawdzić, czy jest w stanie go odtworzyć (co zabierze czas i obciąży łącze).

codecs

Ten atrybut określa kodek użyty do zakodowania materiału wideo. Warto zwrócić uwagę, że w razie użycia tego atrybutu jego wartość należy zapisywać,

rozdział-09/multiple-video-sources.html

HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Określanie wielu źródeł klipu</title>
  </head>
  <body>
    <video poster="images/puppy.jpg" width="400"
           height="320" preload controls loop>
      <source src="video/puppy.mp4" type='video/
           mp4;codecs="avc1.42E01E, mp4a.40.2"' />
      <source src="video/puppy.webm" type='video/
           webm;codecs="vp8, vorbis"' />
      <p>Film o szczeniaku bawiącym się w śniegu.</p>
    </video>
  </body>
</html>
```

WYNIK



używając zarówno apostrofów, jak i cudzysłowów.

Jeśli przeglądarka nie obsługuje elementu `<video>` lub zastosowanego formatu, to wyświetli treść podaną pomiędzy otwierającym znacznikiem `<video>`

a zamkającym znacznikiem `</video>`.

WIĘCEJ W SIECI

W dziale *Tools* strony towarzyszącej książce można znaleźć łącza do narzędzi pozwalających na kodowanie klipów wideo i audio we właściwych formatach.

HTML5: JEDNOCZESNE WYKORZYSTANIE FLASH VIDEO I ELEMENTU <VIDEO> HTML5

Dzięki udostępnieniu klipów w formatach obsługiwanych przez element **<video>** oraz Flash Video można zagwarantować, że będą one dostępne dla przeważającej większości osób odwiedzających stronę.

Można korzystać z elementu **<video>** języka HTML5 jako podstawowego sposobu prezentacji klipów, natomiast Flash Video wykorzystać jako rozwiązanie alternatywne, przeznaczone dla osób, których przeglądarki nie obsługują HTML5. Można też zrobić na odwrót.

Ponieważ niektóre odtwarzacze napisane w technologii Flash obsługują kodowanie H264, w razie wykorzystania takiego odtwarzacza konieczne będzie przygotowanie klipu wyłącznie w formatach H264 i WebM (z formatu FLV można by zrezygnować).

Jeśli zdecydujemy się w pełni wykorzystać możliwości prezentacji wideo, jakie daje HTML5, będziemy mogli:

- Tworzyć własne elementy sterujące odtwarzaniem.
- Udostępniać różne wersje materiału wideo dostosowane do wielkości ekranu, na którym jest wyświetlana przeglądarka (np. można przygotować klip o mniejszej rozdzielczości, przeznaczony do urządzeń kieszonkowych).
- Zażądać zmiany wybranych fragmentów strony po odtworzeniu określonego miejsca klipu.

UMIESZCZANIE DŹWIĘKÓW NA STRONACH WWW

Bez wątpienia najpopularniejszym formatem używanym przy umieszczaniu dźwięku na stronach WWW jest MP3. Podobnie jak w przypadku wideo, dźwięki można dodawać do stron na trzy różne sposoby.

1

UŻYĆ USŁUGI UDOSTĘPNIJĄcej

W internecie działa kilka witryn pozwalających przestać plik audio i udostępniających odtwarzacz, który można umieścić na własnej stronie, np. *SoundCloud* czy *MySpace*.

2

UŻYĆ TECHNOLOGII FLASH

Dostępnych jest kilka filmów Flash pozwalających na odtwarzanie plików MP3 — od prostych przycisków służących do odtworzenia pojedynczej ścieżki dźwiękowej po złożone odtwarzacze umożliwiające tworzenie list odtwarzania i udające szafy grające.

3

UŻYĆ MOŻLIWOŚCI HTML5

W języku HTML5 wprowadzono nowy element — `<audio>`. Przeglądarki, które go obsługują, udostępniają własne elementy sterujące (podobnie jak w przypadku elementu `<video>`).

Niektórzy pytają, w jaki sposób odtwarzać muzykę w sposób ciągły i nieprzerwany, nawet jeśli odwiedzający przechodzi z jednej strony witryny na drugą.

Taki efekt jest bardzo trudny do osiągnięcia — wymagałby on zastosowania takich technologii jak AJAX, by wczytywać zawartość strony, lub stworzenia całej witryny w technologii Flash.

Właśnie z tego powodu niektóre witryny WWW umożliwiają otwieranie odtwarzaczy multimedialnych w nowym oknie przeglądarki, dzięki czemu odtwarzana melodia nie będzie przerywana podczas przechodzenia z jednej strony witryny na drugą.

STOSOWANIE ODTWARZACZA MP3 NAPISANEGO W TECHNOLOGII FLASH

HTML

rozdział-09/adding-a-flash-mp3-player.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Korzystanie z odtwarzacza MP3 napisanego
    w technologii Flash</title>
    <script type="text/javascript"
      src="http://ajax.googleapis.com/ajax/libs/
      swfobject/2.2/swfobject.js"></script>
    <script type="text/javascript">
      var flashvars = {};
      var params = {mp3: "audio/test-audio.mp3"};
      swfobject.embedSWF(
        "flash/player_mp3_1.0.0.swf",
        "music-player", "200", "20", "8.0.0",
        flashvars, params);</script>
    </head>
    <body>
      <div id="music-player">
        <p>Nie możesz usłyszeć tego utworu muzycznego,
          ponieważ Twój przeglądarka nie obsługuje
          technologii Flash.</p>
      </div>
    </body>
  </html>
```

WYNIK



Dostępnych jest bardzo wiele odtwarzaczy MP3 napisanych w technologii Flash; oto trzy z nich:

flash-mp3-player.net
musicplayer.sourceforge.net
www.wimpyplayer.com

Każdy z tych odtwarzaczy ma inne możliwości funkcjonalne — warto je poznać, zanim wybierzemy ten, którego użyjemy na naszych stronach.

W przedstawionym przykładzie wykorzystaliśmy odtwarzacz ze strony flash-mp3-player.net, osadzając go przy użyciu skryptu SWFObject opisanego na stronach 206 – 207. Do odtwarzacza przekazywana jest informacja o położeniu pliku MP3, która zostaje zapisana w formie parametru o nazwie mp3.

Poniżej drugiego otwierającego znacznika `<script>` widać, że utworzyliśmy dwie zmienne JavaScript. Pierwsza z nich nosi nazwę `flashvars`, a druga — `params`. Choć zmienna `flashvars` nie jest używana, skrypt SWFObject oczekuje, że zostanie podana przed zmienną `params`, dlatego jest nam potrzebna.

```
var flashvars = {};
var params = {
  mp3 = "music/noise.mp3";
```

Obie zmienne zostały następnie umieszczone na końcu wiersza kodu osadzającego odtwarzacz MP3 na stronie (tuż przed drugim zamykającym znacznikiem `</script>`).

HTML5: STOSOWANIE ZNACZNIKA <AUDIO> NA STRONACH WWW

<audio>

Język HTML5 wprowadza element <audio> w celu dodawania plików audio do stron. Podobnie jak w przypadku klipów wideo, różne przeglądarki obsługują różne formaty zapisu dźwięku.

Element <audio> ma kilka różnych atrybutów, które pozwalają na kontrolę odtwarzania pliku:

src

Ten atrybut określa ścieżkę do pliku audio.

controls

Ten atrybut określa, czy należy wyświetlać na stronie elementy sterujące odtwarzaniem. Jeśli nie zostanie podany, domyślnie elementy te będą wyświetlane. Używając skryptu JavaScript, można także określić własne elementy sterujące.

autoplay

Umieszczenie tego atrybutu w elemencie <audio> informuje przeglądarkę, że powinna automatycznie rozpoczęć odtwarzanie pliku audio. (Uważa się, że lepszym rozwiązaniem jest udostępnienie użytkownikowi możliwości uruchomienia odtwarzania).

rozdział-09/adding-html5-audio.html

HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Stosowanie znacznika HTML5 &lt;audio&gt;*</title>
  </head>
  <body>
    <audio src="audio/test-audio.ogg"
           controls autoplay>
      <p>Ta przeglądarka nie obsługuje używanego
         formatu plików audio.</p>
    </audio>
  </body>
</html>
```



preload

Użycie tego atrybutu służy określeniu, co przeglądarka powinna zrobić, jeśli nie zostanie użyty atrybut `autoplay`. Atrybut `preload` może przyjmować te same wartości, które zostały opisane na stronie 213, przy okazji omawiania elementu <video>.

loop

Ten atrybut wskazuje, że po zakończeniu odtwarzania ścieżki dźwiękowej przeglądarka

powinna odtworzyć ją ponownie.

Zamieszczony tu przykład działa w przeglądarkach obsługujących format Ogg Vorbis (Firefox, Chrome, Opera). Aby działał także w przeglądarkach Safari 5 lub IE 9, plik audio powinien zostać zapisany w formacie MP3 (można użyć również opisanego na następnej stronie elementu <source>, by udostępnić plik zapisany w innym formacie).

HTML5: WIELE ŹRÓDEŁ MATERIAŁU AUDIO

HTML

rozdział-09/multiple-audio-sources.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Stosowanie wielu źródeł materiału
    audio</title>
  </head>
  <body>
    <audio controls autoplay>
      <source src="audio/test-audio.ogg" />
      <source src="audio/test-audio.mp3" />
      <p>Ta przeglądarka nie obsługuje używanego
      formatu plików audio.</p>
    </audio>
  </body>
</html>
```

WYNIK



src

Atrybut src elementu <source> służy do określania położenia pliku audio.

type

W czasie gdy powstawała ta książka, atrybut type elementów <source> nie był powszechnie używany w odróżnieniu od analogicznego atrybutu stosowanego w elementach <video>.

<source>

Dzięki zastosowaniu elementów <source>, umieszczanych pomiędzy otwierającym znacznikiem <audio> a zamkającym znacznikiem </audio> (i zastępujących atrybut src umieszczany w znaczniku otwierającym), można określić większą liczbę plików audio, których chcemy użyć.

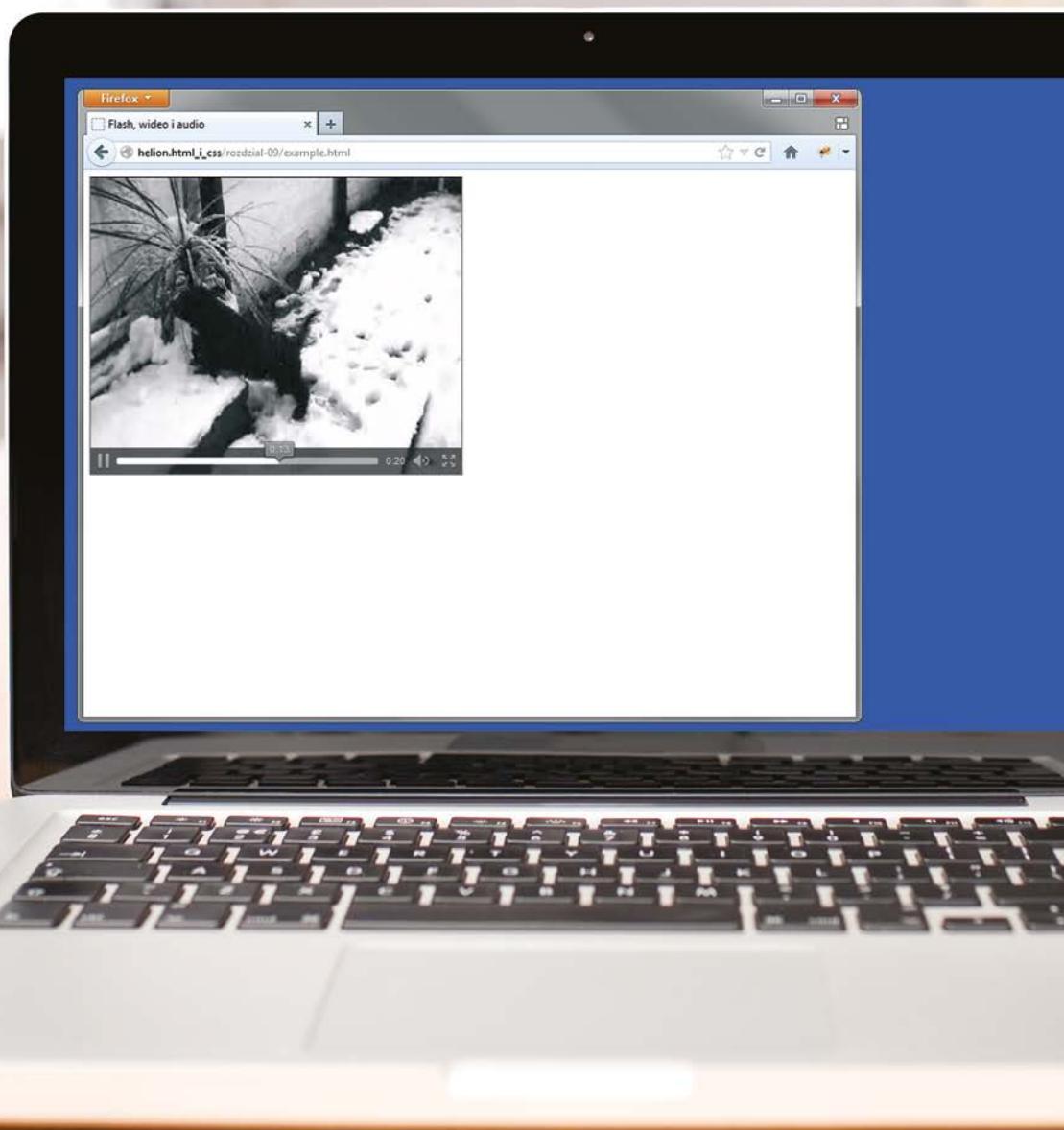
To istotne, gdyż różne przeglądarki obsługują różne formaty plików audio.

MP3: Safari 5+, Chrome 6+, IE 9.

Ogg Vorbis: Firefox 3.6, Chrome 6, Opera 1.5, IE 9.

Aby zatem zapewnić, że plik audio będzie prawidłowo odtwarzany we wszystkich nowoczesnych przeglądarkach obsługujących element <audio>, należy go zapisać w dwóch formatach. Można także zastosować odtwarzacz stworzony przy użyciu technologii Flash, stanowiący rozwiążanie alternatywne, które będzie używane w starszych przeglądarkach nieobsługujących elementu <audio>.

Znacznik <audio> nie zdobył tak dużej popularności jak znacznik <video>, a co gorsza, w pierwszych przeglądarkach, które go implementowały, pojawiały się problemy z jakością odtwarzanego dźwięku.



PRZYKŁAD

FLASH, WIDEO I AUDIO

Przykład prezentuje zastosowanie możliwości HTML5 w celu odtworzenia klipu wideo.

Klip wideo został zakodowany w formatach H264 oraz WebM, by można go było oglądać w możliwie jak największej liczbie przeglądarek. Z myślą o przeglądarkach, które nie obsługują języka HTML5, na stronie został umieszczony także odtwarzacz napisany w technologii Flash; przy czym jest on wyświetlany na stronie przy wykorzystaniu skryptu SWFObject. Jeśli przeglądarka nie obsługuje ani elementu <video> języka HTML5, ani technologii Flash, na stronie zostanie wyświetlony prosty komunikat tekstowy.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Flash, wideo i audio</title>
    <script type="text/javascript"
      src="http://ajax.googleapis.com/ajax/libs/
      swfobject/2.2/swfobject.js"></script>
    <script type="text/javascript">
      var flashvars = {};
      var params = {movie: "../video/puppy.flv"};
      swfobject.embedSWF("flash/osplayer.swf", "snow",
        "400", "320", "8.0.0", flashvars, params);</script>
  </head>
  <body>
    <video poster="images/puppy.jpg" width="400"
      height="320" controls="controls">
      <source src="video/puppy.mp4" type='video/
      mp4; codecs="avc1.42E01E, mp4a.40.2"' />
      <source src="video/puppy.webm" type='video/
      webm; codecs="vp8, vorbis"' />
    <div id="snow">
      <p>Niestety, nie możesz obejrzeć filmu o szczenięciu
        bawiącym się w śniegu, gdyż używana przeglądarka
        nie obsługuje dostępnych formatów wideo.</p>
    </div>
    </video>
  </body>
</html>
```


PODSUMOWANIE

FLASH, WIDEO I AUDIO

- ▶ Technologia Flash pozwala umieszczać na stronach WWW animacje, klipy wideo oraz audio.
- ▶ Technologia ta nie jest obsługiwana w iPhone'ach ani na tabletach iPad.
- ▶ W języku HTML5 zostały wprowadzone elementy `<video>` oraz `<audio>` pozwalające na umieszczanie na stronach treści audio i wideo, jednak są one obsługiwane wyłącznie w najnowszych przeglądarkach.
- ▶ Przeglądarki obsługujące te nowe elementy HTML5 nie obsługują tych samych formatów zapisu plików wideo i audio, a zatem aby zapewnić, że każdy będzie mógł obejrzeć (lub usłyszeć) nasze materiały, trzeba je zapisywać w kilku różnych formatach.

10

PREZENTACJA CSS

- ▶ Do czego służy CSS?
- ▶ Jak działa CSS?
- ▶ Reguły, właściwości i wartości.

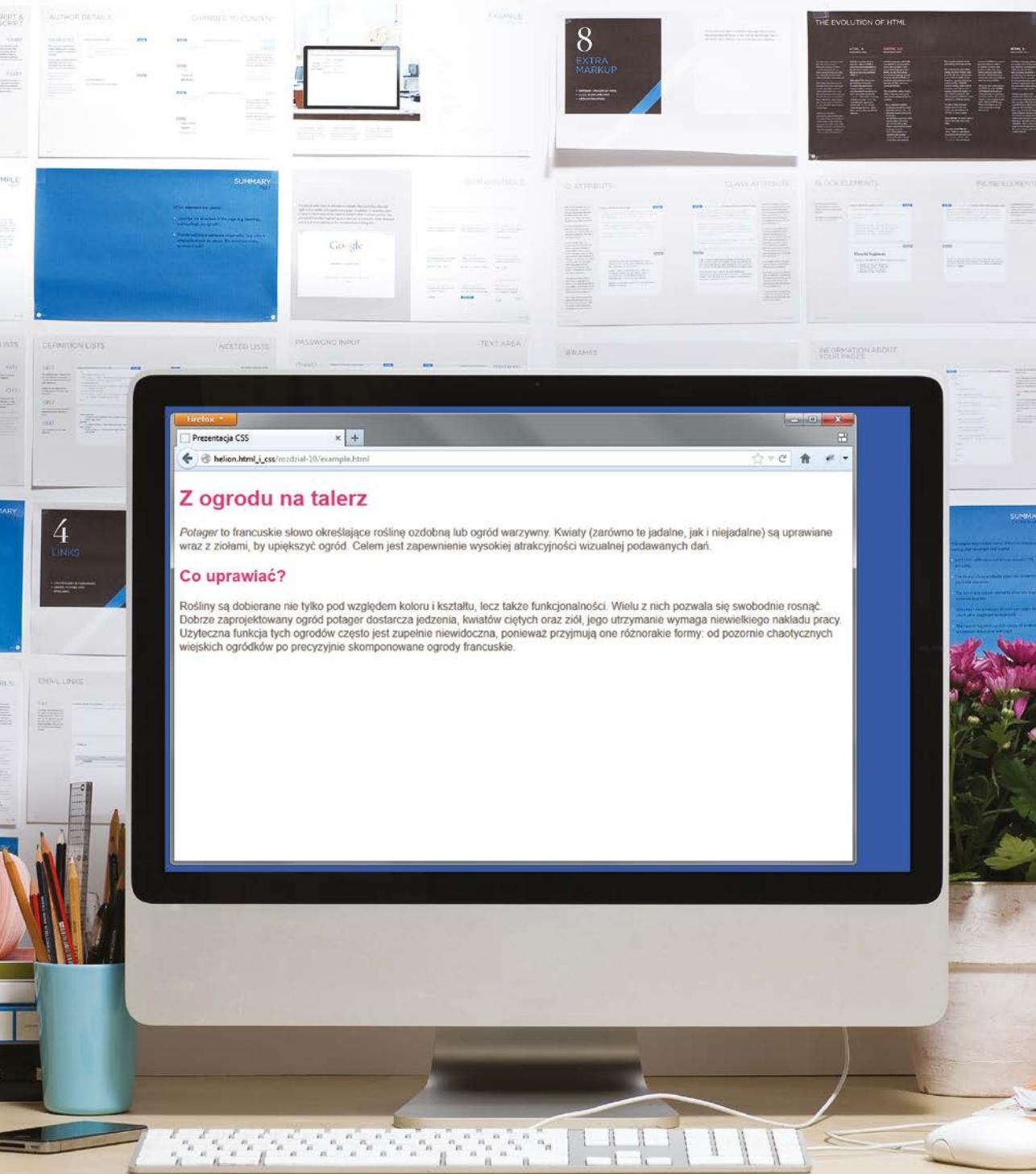
W tej części rozdziału powiemy, jak poprawić atrakcyjność strony WWW, kontrolując jej wygląd przy użyciu CSS.

Kaskadowe arkusze stylów (CSS) pozwalają tworzyć reguły określające wygląd zawartości elementów. Możemy np. podać, że tło strony ma być koloru kremowego, wszystkie akapity powinny być wyświetlane czcionką Arial w kolorze szarym albo że wszystkie nagłówki pierwszego poziomu mają być wyświetlane niebieską, pochyloną czcionką Times.

Kiedy już się dowiemy, jak tworzyć reguły CSS, dalsza nauka posługiwania się tą technologią sprowadzi się do poznania dostępnych właściwości. W tym rozdziale:

- Dowieemy się, jak działają kaskadowe arkusze stylów.
- Nauczmy się pisać reguły CSS.
- Zobaczmy, jak stosować reguły CSS na stronach WWW.

W dalszych rozdziałach książki zostaną przedstawione niemal wszystkie dostępne właściwości CSS.



ZROZUMIEĆ CSS: MYŚL W PUDEŁKU

Aby zrozumieć, jak działa CSS, trzeba sobie wyobrazić, że każdy element HTML jest umieszczony w niewidocznym pudełku.

Obok przedstawiono prostą stronę WWW.

Na następnej stronie można zobaczyć tę samą witrynę, z tym że obszar każdego elementu został zaznaczony ramką. Dzięki temu wyraźnie widzimy te pudełka, wewnątrz których są umieszczone wszystkie elementy strony.

Ogrody przydomowe

Ogród przydomowy (ang. *cottage garden*) to unikalny styl tworzenia ogrodów, bazujący na projekcie nieformalnym, cechującym się gęstym rozmieszczeniem roślin oraz mieszanem roślin ozdobnych i jadalnych.

Ten styl tworzenia ogrodów powstał w Anglii, jego historia sięga wieki wstecz; odrodził się w latach 70. XIX wieku jako reakcja na bardziej strukturalny i rygorystyczny styl angielskich ogrodów pałacowych.

Wczesne ogrody przydomowe miały znacznie bardziej praktyczny charakter niż ich dzisiejsze odpowiedniki - główny nacisk kładziono w nich na uprawę warzyw i ziół, jak również drzew owocowych.

ELEMENTY BLOKOWE I WEWNĄTRZWIERSZOWE

Być może Czytelnik pamięta, że istnieje różnica pomiędzy elementami blokowymi i wewnętrzwiernowymi oraz sposobami ich traktowania przez przeglądarki. Zagadnienie to zostało opisane na stronach 184 – 185.

Elementy blokowe wyglądają tak, jakby rozpoczęły się od nowego wiersza. Przykładami takich elementów są: `<h1>` – `<h6>`, `<p>` i `<div>`.

Elementy wewnętrzwiernowe są rozmieszczone jeden za drugim i nie zaczynają się od nowego wiersza. Przykładami takich elementów są: ``, `<i>`, ``, `` i ``.

CSS pozwalają tworzyć reguły kontrolujące sposób prezentacji pudełek (oraz ich zawartości) poszczególnych elementów.

Ogrody przydomowe

Ogród przydomowy (ang. `cottage garden`) to unikalny styl tworzenia ogrodów, bazujący na projekcie nieformalnym, cechującym się gęstym rozmieszczeniem roślin oraz mieszaniem roślin ozdobnych i jadalnych.

Ten styl tworzenia ogrodów powstał w Anglii, jego historia sięga wieku wstecz; odrodził się w latach 70. XIX wieku jako reakcja na bardziej strukturalny i rygorystyczny styl angielskich ogrodów pałacowych.

Wczesne ogrody przydomowe miały znacznie bardziej praktyczny charakter niż ich dzisiejsze odpowiedniki - główny nacisk kładziono w nich na uprawę warzyw i ziół, jak również drzew owocowych.

W tym przykładzie wszystkie elementy blokowe zostały wyróżnione przy użyciu czerwonego obramowania, natomiast elementy wewnętrzwerszowe — przy użyciu obramowania niebieskiego.

Pierwsze widoczne obramowanie należy do elementu `<body>`, kolejne elementy — `<h1>`, `<h2>`, `<p>`, `<i>` oraz `<a>` — są widoczne wewnętrznie niego.

Używając CSS, możemy dodawać obramowania do pudełek wybranych elementów, określić ich szerokość i wysokość czy dodawać kolor tła. Możemy także określić wygląd tekstu wyświetlanego wewnętrznie pudełka elementu — np. jego kolor, używaną czcionkę.

PRZYKŁADOWE STYLE

PUDEŁKA

Wysokość i szerokość
Obramowanie (kolor, szerokość
oraz styl)
Kolor oraz obraz tła
Położenie wewnętrz okna
przeglądarki

TEKST

Czcionka
Wielkość
Kolor
Kursywa, pogrubienie, wielkie
litery, małe litery, kapitaliki

STYLE SZCZEGÓLÓWE

Dostępne są także bardziej szczegółowe sposoby określania wyglądu konkretnych elementów, takich jak listy, tabele czy formularze.

CSS KOJARZĄ REGUŁY STYLÓW Z ELEMENTAMI HTML

Działanie kaskadowych arkuszy stylów polega na kojarzeniu reguł z elementami HTML. Reguły te określają sposób, w jaki ma być wyświetlana zawartość konkretnych elementów. Reguły CSS składają się z dwóch części: **selekторa** i **deklaracji**.



Ta reguła wskazuje, że wszystkie elementy <p> powinny być wyświetlane przy użyciu czcionki Arial.

Selektory określają, do jakich elementów odnosi się dana reguła. Ta sama reguła może się odnosić do większej liczby elementów, przy czym ich nazwy należy oddzielić od siebie przecinkami.

Deklaracje wskazują, w jaki sposób należy wyświetlać element określony w selektorze. Każda deklaracja składa się z dwóch części (właściwości i wartości), oddzielonych od siebie znakiem dwukropka.

WŁAŚCIWOŚCI CSS MAJĄ WPŁYW NA SPOSÓB WYŚWIETLANIA ELEMENTÓW

Deklaracje CSS są zapisywane pomiędzy parą nawiasów klamrowych, a każda z nich składa się z dwóch części — **właściwości** i **wartości** — oddzielonych od siebie dwukropkiem. W jednej deklaracji można podać większą liczbę właściwości, przy czym poszczególne ich pary muszą być oddzielone od siebie znakiem średnika.

```
h1, h2, h3 {  
    font-family: Arial;  
    color: yellow; }  
  
  ┌─────────┐ ┌─────────┐  
  |         | |         |  
  WŁAŚCIWOŚĆ   WARTOŚĆ
```

Ta reguła określa, że wszystkie elementy `<h1>`, `<h2>` i `<h3>` mają być wyświetlane na żółto, przy użyciu czcionki Arial.

Właściwości wskazują konkretny aspekt wyglądu elementu, który chcemy określić. Może nim być np. kolor, czcionka, szerokość, wysokość albo obramowanie.

Wartości natomiast wskazują konkretne ustawienie, które chcemy nadać wybranej właściwości. Jeśli np. chcemy określić właściwość `color`, to powinniśmy jej przypisać wartość reprezentującą wybrany kolor.

HTML elements are used to:

- Describe the structure of the page (e.g. headings, subheadings, paragraphs).
- Provide additional semantic information (e.g. where emphasis should be placed, the definition of any acronyms used).

The best known form on the web is probably the search box. That's right, in the middle of Google's home page. In addition to making a search box a standard element, the new `input` type `search` also adds a few new features, such as the ability to set a placeholder value for the input, and use terms when registering as a member of a website when signing up for an account or logging in.

Google

The `input` element is the most common form control on the web. It's used for everything from simple text inputs to complex file uploads. The new `input` types add a few new features, such as the ability to set a placeholder value for the input, and use terms when registering as a member of a website when signing up for an account or logging in.

The `input` element is the most common form control on the web. It's used for everything from simple text inputs to complex file uploads. The new `input` types add a few new features, such as the ability to set a placeholder value for the input, and use terms when registering as a member of a website when signing up for an account or logging in.

The `input` element is the most common form control on the web. It's used for everything from simple text inputs to complex file uploads. The new `input` types add a few new features, such as the ability to set a placeholder value for the input, and use terms when registering as a member of a website when signing up for an account or logging in.

The `input` element is the most common form control on the web. It's used for everything from simple text inputs to complex file uploads. The new `input` types add a few new features, such as the ability to set a placeholder value for the input, and use terms when registering as a member of a website when signing up for an account or logging in.

NESTED LISTS

PASSWORD INPUT

TEXT AREA

IFRAMES

Firefox ▾

Prezentacja CSS

helion.html_i_css/rozdzial-10/example.html

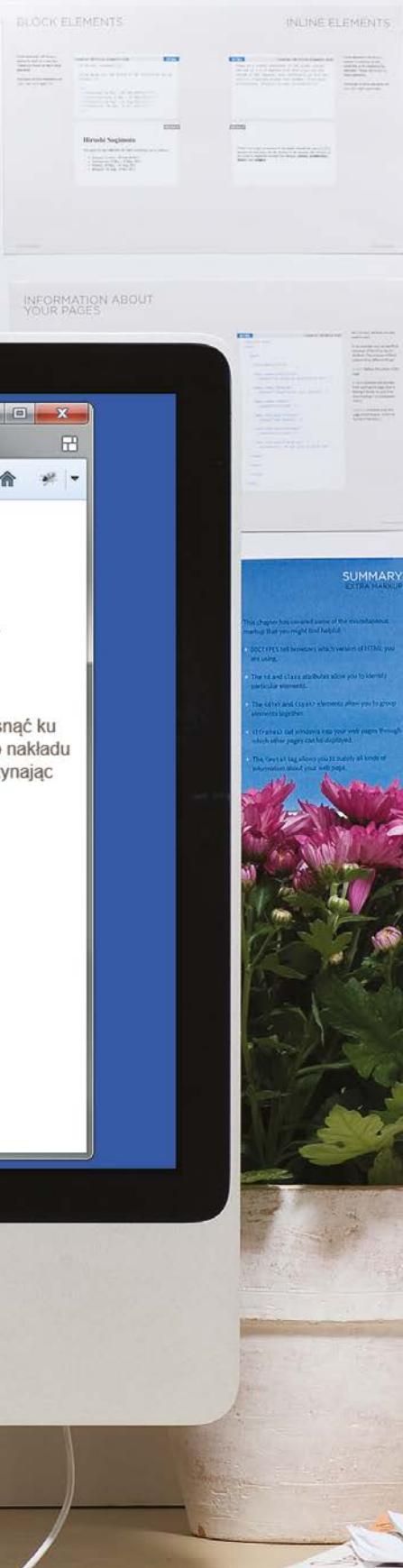


Z ogrodu na talerz

Potager to francuskie wyrażenie określające roślinę ozdobną lub ogród warzywny. Kwiaty (zarówno te jadalne jak i niejadalne) są hodowane wraz z ziołami, by upiększyć ogród. Celem jest zapewnienie wysokiej atrakcyjności wizualnej podawanego pożywienia.

Co uprawiać?

Rośliny są dobierane nie tylko pod względem koloru i kształtu, lecz także funkcjonalności. Wielu z nich pozwala się swobodnie rosnąć w górze. Dobrze zaprojektowany ogród *potager* może dostarczać jedzenia, kwiatów ciętych oraz ziół i wymagać bardzo niewielkiej pracy na utrzymanie. Ogródki te potrafią ukrywać swoją funkcję dostarczania pożywienia przybierając wiele różnych postaci - zaczynając od pozornie хаotycznych ogródków wiejskich a kończąc na formalnych ogrodach francuskich.



PRZYKŁAD

PREZENTACJA CSS

Oto przykład prostej strony WWW, której wygląd został określony przy użyciu CSS.

Ten przykład tworzą dwa pliki: dokument HTML (*example.html*) oraz oddzielnny plik CSS (*example.css*). W piątym wierszu dokumentu HTML został umieszczony element `<link>`, określający położenie używanego pliku CSS.

Na następnej stronie pokazujemy, że reguły CSS mogą być także umieszczane w kodzie stron WWW, oraz piszemy, kiedy można się zdecydować na takie rozwiązanie.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Prezentacja CSS</title>
    <link href="css/example.css"
          type="text/css" rel="stylesheet" />
  </head>
  <body>
    <h1>Z ogrodu na talerz</h1>
    <p><i>Potager</i> to francuskie wyrażenie określające
       roślinę ozdobną lub ogród warzywny. Kwiaty...</p>
    <h2>Co uprawiać?</h2>
    <p>Rośliny są dobierane nie tylko pod względem koloru
       i kształtu, lecz także funkcjonalności. Wielu z nich
       pozwala się...</p>
  </body>
</html>
```

```
body {
  font-family: Arial, Verdana, sans-serif;
}
h1, h2 {
  color: #ee3e80;
}
p {
  color: #665544;
```

STOSOWANIE ZEWNĘTRZNYCH ARKUSZY STYLÓW

<link>

Element <link> może być używany w dokumentach HTML, by informować przeglądarkę, gdzie znajduje się plik CSS, którego należy użyć. Jest to element pusty (nie ma znacznika zamykającego), umieszczany wewnątrz elementu <head>. Należy w nim podać trzy atrybuty:

href

Ten atrybut określa ścieżkę do pliku CSS (który bardzo często jest umieszczany w katalogu o nazwie `css` lub `styles`).

type

Ten atrybut określa typ dokumentu dołączanego do strony. Powinien mieć wartość `text/css`.

rel

Ten atrybut określa związek pomiędzy stroną HTML a dołączanym plikiem. W przypadku dołączania zewnętrznych arkuszy stylów powinien mieć wartość `stylesheet`.

Dokument HTML może używać więcej niż jednego zewnętrznego arkusza stylów CSS. W tym celu należy w jego kodzie umieścić po jednym elemencie <link> dla każdego dołączanego

rozdział-10/using-external-css.html

HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Stosowanie zewnętrznych plików CSS</title>
    <link href="css/styles.css" type="text/css"
          rel="stylesheet" />
  </head>
  <body>
    <h1>Ziemniaki</h1>
    <p>Istnieje wiele różnych rodzajów ziemniaków. Ich odmiany zaliczają się zazwyczaj do jednej z głównych kategorii zależnych od czasu dojrzewania, takich jak: wczesne, średnio wczesne bądź późne.</p>
  </body>
</html>
```

rozdział-10/styles.css

CSS

```
body {
  font-family: arial;
  background-color: rgb(185,179,175);}
h1 {
  color: rgb(255,255,255);}
```

Ziemniaki

Istnieje wiele różnych gatunków ziemniaków. Ich odmiany zaliczają się zazwyczaj do jednej z głównych kategorii zależnych od czasu dojrzewania, takich jak: wczesne, średnio wczesne bądź późne.

WYNIK

pliku CSS. Niektórzy autorzy stron WWW używają jednego arkusza stylów do określania

prezentacji (czyli czcionek i kolorów) oraz innego do określenia układu strony.

STOSOWANIE WEWNĘTRZNYCH ARKUSZY STYLÓW

HTML

CSS

rozdział-10/using-internal-css.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Stosowanie wewnętrznych plików CSS</title>
    <style type="text/css">
      body {
        font-family: arial;
        background-color: rgb(185,179,175);
      }
      h1 {
        color: rgb(255,255,255);
      }
    </style>
  </head>
  <body>
    <h1>Ziemniaki</h1>
    <p>Istnieje wiele różnych gatunków ziemniaków. Ich odmiany zaliczają się zazwyczaj do jednej z głównych kategorii zależnych od czasu dojrzewania, takich jak: wczesne, średnio wczesne bądź późne.</p>
  </body>
</html>
```

<style>

Reguły CSS można także umieszczać w dokumencie HTML, przy czym należy je zapisywać w elemencie `<style>`, który zazwyczaj jest dodawany wewnątrz elementu `<head>`.

Element `<style>` powinien zawierać atrybut `type`, by poinformować przeglądarkę, że jego zawartość to reguły CSS.

Tworząc witrynę składającą się z więcej niż jednej strony, należy korzystać z zewnętrznych arkuszy stylów. Takie rozwiązanie:

- pozwala, by wszystkie strony używały tych samych reguł (czyli nie trzeba ich powtarzać na każdej ze stron);
- zapewnia separację treści od prezentacji;
- pozwala wprowadzać modyfikacje stylów tylko w jednym pliku (a nie na każdej ze stron osobno).

WYNIK

Ziemniaki

Istnieje wiele różnych gatunków ziemniaków. Ich odmiany zaliczają się zazwyczaj do jednej z głównych kategorii zależnych od czasu dojrzewania, takich jak: wczesne, średnio wczesne bądź późne.

W języku HTML 4 oraz XHTML Transitional dostępny był atrybut `style`, który można było umieszczać w większości elementów strony. Reguły CSS umieszczane w wartości

tego atrybutu odnosiły się wyłącznie do danego elementu. W nowych witrynach należy unikać stosowania tego atrybutu, jednak wspominamy tu o nim, gdyż można go

zobaczyć w starszym kodzie. Oto przykład pokazujący, jak zmienić kolor czcionki w akapicie:

```
<p style="color:red;">
```

SELEKTORY CSS

Istnieje wiele różnych typów selektorów CSS pozwalających precyzyjnie określić, do jakich elementów strony odnoszą się poszczególne reguły.

Tabela zamieszczona na kolejnej stronie przedstawia najczęściej stosowane selektory.

Na tej stronie natomiast pokazano przykładowe elementy, do których będą się odnosić te selektory.

W selektorach CSS uwzględniana jest wielkość liter, dlatego muszą one dokładnie odpowiadać nazwom elementów oraz wartościom atrybutów.

Dostępne są także bardziej zaawansowane selektory, pozwalające odwoływać się do konkretnych elementów na podstawie ich atrybutów oraz ich wartości; zostaną one przedstawione na stronie 291.

Internet Explorer 7 był pierwszą wersją, która obsługiwała ostatnie dwa typy selektorów przedstawionych w tabeli na następnej stronie (selektory elementów sąsiednich), dlatego są one stosowane nieco rzadziej niż pozostałe.

rozdział-10/css-selectors.html

HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Selektory CSS</title>
  </head>
  <body>
    <h1><a id="top">Kalendarz ogrodu kuchennego</a></h1>
    <p id="introduction">Poniżej znajdziesz użytkowy poradnik opisujący, co i kiedy należy zrobić.</p>
    <h2>Wiosna</h2>
    <ul>
      <li><a href="mulch.html">Nawożenie ziemi ściółką ogrodniczą.</a></li>
      <li><a href="potato.html">Sadzenie wczesnych ziemniaków.</a></li>
      <li><a href="tomato.html">Wysianie pomidorów.</a></li>
      <li><a href="beet.html">Wysianie buraków ćwikłowych.</a></li>
      <li><a href="zucchini.html">Wysianie cukinii.</a></li>
      <li><a href="rhubarb.html">Wycięcie kwiatów rabarbaru.</a></li>
    </ul>
    <p class="note">Ta strona została napisana przez <a href="mailto:ivy@jakaswitryna.org">ivy@jakaswitryna.org</a> dla <a href="http://www.jakaswitryna.org">Jakiejs Witryny</a>.</p>
    <p><a href="#top">Na początek</a></p>
  </body>
</html>
```

SELEKTOR	ZNACZENIE	PRZYKŁAD
SELEKTOR UNIWERSALNY	Odnosi się do wszystkich elementów strony	* {} Odnosi się do wszystkich elementów na stronie
SELEKTOR TYPU	Odnosi się do elementów o konkretnej nazwie	h1, h2, h3 {} Odnosi się do elementów <h1>, <h2> oraz <h3>
SELEKTOR KLASY	Odnosi się do elementów, których wartość atrybutu class odpowiada wartości podanej za znakiem kropki	.note {} Odnosi się do wszelkich elementów, których atrybut class ma wartość note p.note {} Odnosi się jedynie do elementów <p>, których atrybut class ma wartość note
SELEKTOR IDENTYFIKATORA	Odnosi się do elementów, których wartość atrybutu id odpowiada wartości podanej za znakiem #	#introduction {} Odnosi się do elementu, którego atrybut id ma wartość introduction
SELEKTOR ELEMENTU DZIECKA	Odnosi się do elementów, które są dziećmi innego, określonego elementu	li > a {} Odnosi się do wszystkich elementów <a> umieszczonych bezpośrednio w elementach (lecz nie do innych elementów <a> na stronie)
SELEKTOR ELEMENTU POTOMNEGO	Odnosi się do elementów będących elementami potomnymi innego, określonego elementu	p a {} Odnosi się do dowolnego elementu <a> umieszczonego wewnątrz elementu <p>, nawet jeśli znajduje się on wewnątrz jakichś innych elementów
SELEKTOR ELEMENTÓW SĄSIADUJĄCYCH BEZPOŚREDNIO	Odnosi się do elementu bezpośrednio sąsiadującego z innym, określonym elementem	h1 + p {} Odnosi się do pierwszego elementu <p> umieszczonego za elementem <h1> (lecz nie do kolejnych elementów <p>)
OGÓLNY SELEKTOR ELEMENTÓW SĄSIADUJĄCYCH	Odnosi się do elementu sąsiadującego z innym elementem, choć niekoniecznie bezpośrednio	h1 ~ p {} Jeśli na stronie są dwa elementy <p> sąsiadujące z elementem <h1>, ta reguła odnosi się do obu z nich

JAK POWSTAJE KASKADA REGUŁ CSS?

Jeśli w arkuszu stylów znajdują się dwie (lub więcej) reguły odnoszące się do tego samego elementu, to koniecznie musimy wiedzieć, w jakiej kolejności będą one stosowane.

OSTATNIA REGUŁA

Jeśli dwa selektory są identyczne, to pierwszeństwo będzie mieć drugi z nich. W przedstawionym obok przykładzie drugi z selektorów i będzie miał pierwszeństwo przed pierwszym.

STOPIEŃ SZCZEGÓŁOWOŚCI

Jeśli pewien selektor bardziej precyzyjnie niż inne określa elementy, to taka precyzyjniejsza reguła będzie miała pierwszeństwo przed regułami bardziej ogólnymi. W przedstawionym obok przykładzie:

- selektor h1 jest bardziej szczegółowy od *;
- p b jest bardziej szczegółowy od p;
- p#intro jest bardziej szczegółowy od p.

IMPORTANT

Za wartością każdej właściwości można umieścić wyrażenie !important, by zaznaczyć, że powinna ona być ważniejsza od innych reguł odnoszących się do tego samego elementu.

Zrozumienie zasad powstawania kaskady reguł pozwala tworzyć prostsze arkusze stylów, gdyż można tworzyć bardziej

rozdział-10/cascade.html

HTML

```
<h1>Ziemniaki</h1>
<p id="intro">Istnieje <i>wiele</i> różnych gatunków
<b>ziemniaków</b>.</p>
<p>Ich odmiany zaliczają się zazwyczaj do jednej
z głównych kategorii zależnych
od czasu dojrzewania, takich jak: wczesne,
średnio wczesne bądź późne.</p>
```

CSS

```
* {
  font-family: Arial, Verdana, sans-serif;
}
h1 {
  font-family: "Courier New", monospace;
}
i {
  color: green;
}
i {
  color: red;
}
b {
  color: pink;
}
p b {
  color: blue !important;
}
p b {
  color: violet;
}
p#intro {
  font-size: 100%;
}
p {
  font-size: 75%;}
```

WYNIK

Ziemniaki

Istnieje **wiele** różnych gatunków **ziemniaków**.

Ich odmiany zaliczają się zazwyczaj do jednej z głównych kategorii zależnych od czasu dojrzewania, takich jak: wczesne, średnio wczesne bądź późne.

ogólne reguły odnoszące się do większości elementów, a następnie przesłaniać i zmieniać właściwości

poszczególnych elementów, które muszą być wyświetlane w odmienny sposób.

DZIEDZICZENIE

HTML

rozdział-10/inheritance.html

```
<div class="page">
  <h1>Ziemniaki</h1>
  <p>Istnieje wiele różnych gatunków ziemniaków.
  <p>Ich odmiany zaliczają się zazwyczaj do jednej
    z głównych kategorii zależnych od czasu
    dojrzewania, takich jak: wczesne, średnio
    wczesne bądź późne.</p>
</div>
```

CSS

```
body {
  font-family: Arial, Verdana, sans-serif;
  color: #665544;
  padding: 10px;}
.page {
  border: 1px solid #665544;
  background-color: #efefef;
  padding: inherit;}
```

WYNIK

Ziemniaki

Istnieje wiele różnych gatunków ziemniaków.

Ich odmiany zaliczają się zazwyczaj do jednej z głównych kategorii zależnych od czasu dojrzewania, takich jak: wczesne, średnio wczesne bądź późne.

Jeśli określmy właściwość `font-family` lub `color` w regule odnoszącej się do elementu `<body>`, to zostaną one także zastosowane w większości elementów potomnych. Dzieje się tak dlatego, że wartość właściwości `font-family` jest **dziedziczona** przez elementy potomne. Rozwiążanie to chroni nas przed koniecznością tworzenia reguł odnoszących się do wszystkich elementów, w których tę właściwość chcemy zastosować (a w efekcie sprawia, że arkusze stylów mogą być prostsze).

Dla porównania takie właściwości jak `background-color` lub `border` **nie są dziedziczone**. Gdyby były, na stronie panowałby bałagan.

W przypadku bardzo wielu właściwości można wymusić, by ich wartości były dziedziczone po przodkach. W tym celu wystarczy przypisać właściwości wartość `inherit`. W przedstawionym przykładzie element `<div>` z atrybutem `class` o wartości `page` dziedziczy ustawienia właściwości `padding` po regule odnoszącej się do elementu `<body>`.

DŁACZEGO WARTO UŻYWAĆ ZEWNĘTRZNYCH ARKUSZY STYLÓW?

Umieszczanie reguł CSS w zewnętrznych arkuszach stylów podczas tworzenia witryn WWW ma kilka zalet.

Ten sam arkusz stylów może być używany na wszystkich stronach. Efekt ten można uzyskać, umieszczając na każdej z nich element `<link>` dołączający ten sam plik CSS. Oznacza to, że tego samego kodu nie trzeba będzie umieszczać na każdej ze stron (co je upraszcza i skracia).

Co więcej, kiedy przeglądarka użytkownika wczyta arkusz stylów, to kolejne odwiedzane strony witryny będą wyświetlane szybciej. W razie konieczności zmiany wyglądu witryny modyfikacje trzeba będzie wprowadzić tylko w jednym pliku CSS, a wszystkie strony zostaną

automatycznie zaktualizowane. I tak wygląd wszystkich elementów `<h1>` można zmienić, modyfikując tylko jeden plik CSS, a nie odpowiednie reguły na wszystkich stronach witryny. Powszechnie uważa się, że separacja zawartości witryny od reguł określających jej wygląd jest dobrą praktyką.

Czasami można się zdecydować na umieszczenie reguł CSS w samym dokumencie HTML.

W razie tworzenia jednej strony WWW można umieścić reguły CSS w tym samym pliku, tak by cały kod znajdował się w jednym miejscu. (Wielu twórców stron jednak uznałoby, że umieszczenie kodu CSS w osobnym pliku jest lepszym rozwiązańiem).

Jeśli jedna ze stron witryny wymaga użycia kilku dodatkowych reguł CSS (które nie są potrzebne na żadnej innej stronie), także można się zdecydować na umieszczenie ich w kodzie dokumentu HTML. (Również w tym przypadku większość twórców stron uzna, że lepszym rozwiązańiem będzie umieszczenie ich w odrębnym pliku).

W większości przykładów w tej książce reguły CSS są umieszczane w elemencie `<head>` dokumentu HTML (w elemencie `<style>`), a nie w oddzielnym pliku. Zdecydowaliśmy się na takie rozwiązanie, by oszczędzić Czytelnikowi konieczności otwierania dwóch plików podczas analizy przykładów.

RÓŻNE WERSJE CSS I DZIWACTWA PRZEGŁĄDAREK

Specyfikacja CSS1 została opublikowana w 1996 roku, a CSS2 — dwa lata później. Wciąż trwają prace nad specyfikacją CSS3, jednak wszystkie najpopularniejsze przeglądarki już zaczęły ją implementować.

Podobnie jak w przypadku języka HTML, który miał już kilka wersji, istnieje kilka wersji CSS.

Przeglądarki nie zaimplementowały od razu wszystkich możliwości CSS, a zatem niektóre starsze przeglądarki nie udostępniają wszystkich właściwości CSS.

Wspominamy o tym w miejscach, w których może to mieć znaczenie dla Czytelnika, podając jednocześnie informacje, kiedy właściwości CSS mogą zachowywać się w sposób niezgodny z oczekiwaniami.

Każdy doświadczony użytkownik CSS potwierdzi, że niektóre przeglądarki obsługują pewne właściwości CSS w nieoczekiwany sposób. Jednak ominięcie tych błędów jest proste, o ile wiadomo, jak to zrobić...

Przed udostępnieniem każdej nowej witryny koniecznie należy ją przetestować w kilku przeglądarkach, gdyż w każdej z nich mogą występować nieznaczne różnice w sposobie wyświetlania stron.

Do testowania witryny wcale nie trzeba dysponować wieloma komputerami — w internecie bowiem są dostępne narzędzia pokazujące, jak strona będzie wyglądać w różnych przeglądarkach:

browsercam.com
browserlab.adobe.com
browsershots.org
crossbrowsertesting.com

W razie korzystania z tych narzędzi warto sprawdzić, jak witryna będzie wyglądać na różnych platformach komputerowych (PC, Mac oraz Linux) w starszych oraz najnowszych wersjach najpopularniejszych przeglądarek. Jeśli sprawdzimy witrynę w więcej niż jednej przeglądarce, możemy się przekonać, że niektóre elementy stron nie wyglądają zgodnie z naszymi oczekiwaniami. Kiedy właściwość CSS nie jest prezentowana zgodnie z oczekiwaniami, to taką sytuację nazywamy **dziwactwem przeglądarki** (ang. *browser quirk*) lub **błędem CSS**.

Niektóre najczęściej spotykane dziwactwa przeglądarek zostały

opisane w tej książce, istnieje jednak wiele mniejszych błędów występujących rzadziej lub pojawiających się w starszych przeglądarkach używanych aktualnie przez niewielu użytkowników.

W razie napotkania jakiegoś błędu CSS można skorzystać z ulubionej wyszukiwarki, by spróbować znaleźć rozwiązanie problemu. Można też poszukać go w jednej z poniższych witryn: positioniseverything.net quirksmode.org

PODSUMOWANIE

PREZENTACJA CSS

- ▶ CSS traktują wszystkie elementy HTML tak, jakby były umieszczone wewnętrz pudełek, i używają reguł, by określić, jak powinny one wyglądać.
- ▶ Każda reguła składa się z selektora (określającego elementy, do których reguła się odnosi) oraz deklaracji (określającej, jak te elementy powinien wyglądać).
- ▶ Różne typy selektorów pozwalają tworzyć reguły odnoszące się do różnych elementów.
- ▶ Każda deklaracja składa się z dwóch części: nazwy właściwości elementu, którą chcemy określić, oraz wartości przypisywanej tej właściwości; np. właściwość font-family pozwala na określenie czcionki, natomiast wartość Arial podaje czcionkę, której należy użyć.
- ▶ Reguły CSS są zazwyczaj umieszczane w osobnych plikach, choć można je także podać w kodzie dokumentów HTML.

11

KOLOR

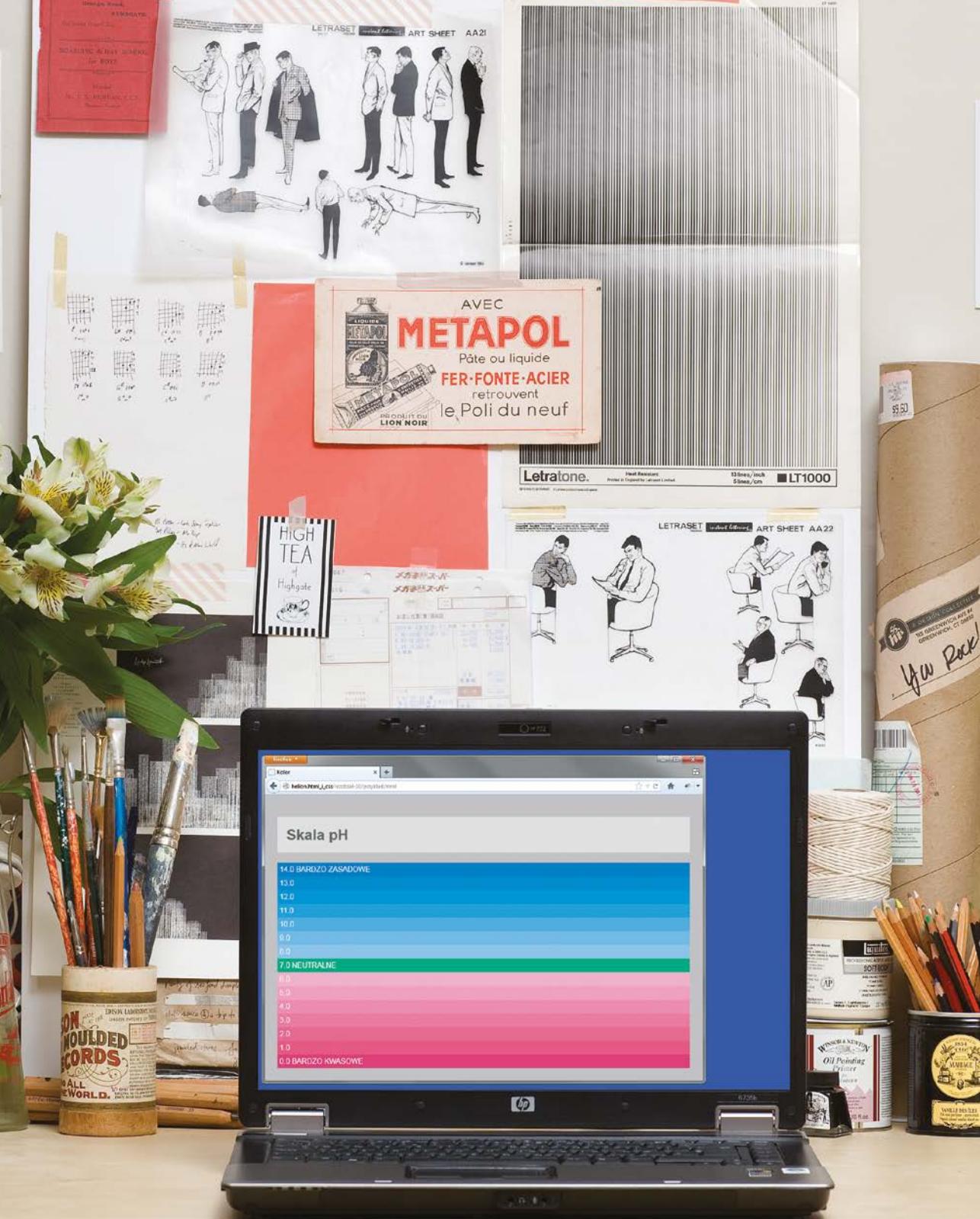
- ▶ Określanie kolorów.
- ▶ Terminologia związana z kolorami oraz kontrast.
- ▶ Kolor tła.

Kolory naprawdę potrafią ożywić tworzone strony WWW.

W tym rozdziale opisane zostaną następujące zagadnienia:

- Określanie kolorów, gdyż istnieją trzy powszechnie stosowane sposoby podawania wybranego koloru (CSS3 udostępnia kolejne).
- Terminologia związana z kolorami, ponieważ istnieje kilka terminów, których zrozumienie może się okazać niezwykle przydatne podczas posługiwania się kolorami.
- Kontrast oraz zapewnianie należytej czytelności tekstów.
- Kolory tła, określane zarówno dla całej strony, jak i jej poszczególnych elementów.

Wszystkie informacje dotyczące kolorów, które Czytelnik zdobędzie w tym rozdziale, będą wykorzystywane w dalszej części książki przy okazji określania kolorów tekstu oraz pudełek elementów.



KOLOR TEKSTU

color

Właściwość `color` pozwala określić kolor tekstu wewnątrz elementu. Kolory w CSS można określić na trzy sposoby:

WARTOŚCI RGB

Wyróżniają kolory jako ilość tworzących go trzech kolorów podstawowych: czerwonego, zielonego i niebieskiego, np. `rgb(100,100,90)`.

KODY SZESNASTKOWE

To sześciocyfrowe kody reprezentujące ilość trzech kolorów podstawowych, poprzedzone znakiem `#`, np. `#ee3e80`.

NAZWY KOLORÓW

Dostępnych jest 147 predefiniowanych nazw kolorów, które rozpoznają przeglądarki, np. `DarkCyan`.

Wszystkie te trzy sposoby określania kolorów zostały dokładniej opisane na stronach 250 – 251.

W CSS3 wprowadzono dodatkowy sposób zapisu kolorów, określany jako SHLA. Zostanie on przedstawiony pod koniec tego rozdziału, na stronach 254 – 255.

rozdział-11/foreground-color.html

CSS

```
/* nazwa koloru */  
h1 {  
  color: DarkCyan;  
/* kod szesnastkowy */  
h2 {  
  color: #ee3e80;  
/* wartość rgb */  
p {  
  color: rgb(100,100,90);
```

Biologia morska

Skład wody morskiej

W wodzie morskiej można znaleźć niemalże wszystko. Dotyczy to materiałów pochodzących zarówno ze skorupy ziemskiej, jak i od organizmów żywych. Najważniejszymi parametrami wody morskiej, mającymi wpływ na morskie formy życia, są: zasolenie, rozpuszczone gazy (głównie tlen i dwutlenek węgla), składniki pokarmowe oraz poziom pH. Parametry te zmieniają się zarówno pod względem ilości, jak i wpływu wywieranego na morskie formy życia.

Powyższy przykład pokazuje, jak można dodawać komentarze do kodu CSS — zostały one zapisane nad każdą z przedstawionych reguł. Dowolny tekst umieszczony pomiędzy symbolami `/* i */` zostanie potraktowany jako komentarz. Na przedstawionym przykładzie komentarze zostały wyróżnione szarym kolorem.

Dodawanie komentarzy może ułatwić zrozumienie pliku CSS (poprawić jego organizację poprzez podzielenie długiego dokumentu na fragmenty). W tym przykładzie użyliśmy komentarzy, by opisać użyte metody określania kolorów.

WYNIK

KOLOR TŁA

background-color

CSS

rozdział-11/background-color.html

```
body {  
    background-color: rgb(200,200,200);}  
h1 {  
    background-color: DarkCyan;}  
h2 {  
    background-color: #ee3e80;}  
p {  
    background-color: white;}
```

WYNIK

Biologia morska

Skład wody morskiej

W wodzie morskiej można znaleźć niemalże wszystko. Dotyczy to materiałów pochodzących zarówno ze skorupy ziemskiej, jak i od organizmów żywych. Najważniejszymi parametrami wody morskiej, mającymi wpływ na morskie formy życia, są: zasolenie, rozpuszczone gazy (głównie tlen i dwutlenek węgla), składniki pokarmowe oraz poziom pH. Parametry te zmieniają się zarówno pod względem ilości, jak i wpływu wywieranego na morskie formy życia.

Wszystkie elementy HTML są traktowane przez CSS w taki sposób, jak gdyby były umieszczone w pudełku, a właściwość `background-color` określa kolor jego tła.

Wybrany kolor tła można określić na trzy sposoby (dokładnie te same jak w przypadku koloru pierwszoplanowego), czyli podając wartości RGB, kod szesnastkowy lub nazwę koloru (metody te zostały omówione na następnej stronie).

Jeśli kolor tła nie zostanie określony, tło elementu będzie przezroczyste.

Domyślnie okno przeglądarki ma białe tło, jednak użytkownik może sam określić kolor tła okien, dlatego aby mieć pewność, że zawartość strony będzie wyświetlana na białym tle, należy określić wartość właściwości `background-color` dla elementu `<body>`.

W tym przykładzie określiliśmy także wartości właściwości `padding`, aby odsunąć nieco tekst od obramowania elementów. Takie rozwiązanie ułatwia czytanie strony. Więcej informacji na temat właściwości `padding` można znaleźć na stronie 312.

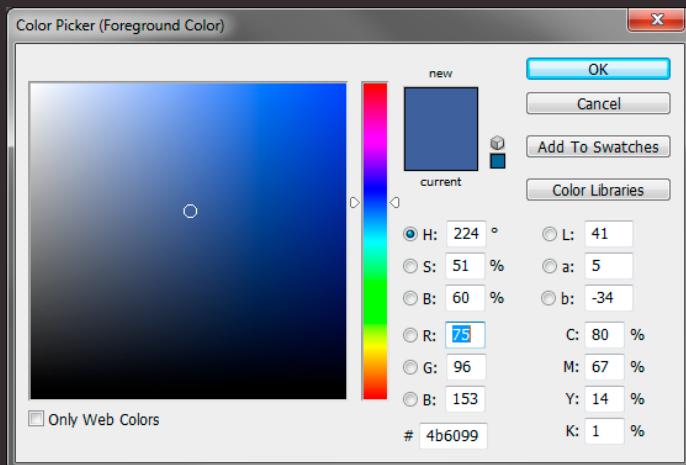
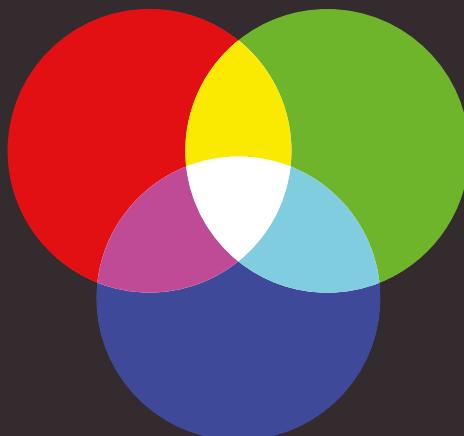
ZROZUMIENIE KOLORÓW

Każdy kolor na ekranie monitora jest tworzony jako połączenie pewnych ilości trzech kolorów podstawowych: czerwonego, zielonego i niebieskiego. Aby znaleźć poszukiwany kolor, można użyć specjalnego narzędzia służącego do ich wybierania.

Monitory komputerowe składają się z tysięcy niewielkich kwadracików nazywanych pikselami (jeśli przyjrzymy się bardzo dokładnie, powinniśmy je zobaczyć).

Kiedy monitor jest wyłączony, jest czarny, gdyż nie emituje żadnego światła. Po włączeniu każdy piksel ekranu może mieć inny kolor, co tworzy obraz.

Kolor każdego piksela na ekranie jest określany jako mieszanka koloru czerwonego, zielonego i niebieskiego — dokładnie tak samo jak w telewizorach.



Narzędzia do wybierania kolorów są dostępne w programach do edycji obrazów, takich jak Photoshop lub GIMP. Wartości RGB są wyświetlane obok przycisków opcji o nazwach *R*, *G* i *B*. Odpowiadająca im wartość szesnastkowa natomiast jest wyświetlana w polu tekstowym poprzedzonym znakiem *#*. Dostępne jest także doskonale narzędzie do wybierania kolorów, które można znaleźć na stronie colorschemedesigner.com.

WARTOŚCI RGB

Wartości kolorów czerwonego, zielonego i niebieskiego są podawane jako liczby z zakresu od 0 do 255.

rgb(102,205,170)

Na ten kolor składają się następujące wartości:
czerwony — 102
zielony — 205
niebieski — 170

ODCIEŃ

Potocznie wyraz „odcień” jest traktowany jak synonim słowa „kolor”. Z technicznego punktu widzenia na kolor składają się także nasycenie oraz jasność.



KODY SZESNASTKOWE

W przypadku kodów szesnastkowych wartości kolorów podstawowych — czerwonego, zielonego i niebieskiego — są zapisywane w postaci liczb szesnastkowych.

#66cdAA

Wartość koloru czerwonego, 102, jest reprezentowana przez liczbę szesnastkową 66. Wartość koloru zielonego, 205, to szesnastkowa liczba cd, natomiast wartość koloru niebieskiego, 170, to szesnastkowa liczba aa.

NASYCENIE

Nasycenie określa ilość szarości w danym kolorze. Przy maksymalnym nasyceniu w kolorze nie ma żadnej szarości. Przy minimalnym nasyceniu kolor będzie niemal szary.



NAZWY KOLORÓW

Kolory są także reprezentowane przez predefiniowane nazwy. Jednak w tym przypadku ich liczba jest ograniczona.

MediumAquaMarine

Przeglądarki obsługują dokładnie 147 kolorów predefiniowanych (widoczny powyżej kolor nosi nazwę MediumAquaMarine). Większość projektantów uważa, że taka paleta jest mocno ograniczona, a poza tym zapamiętanie nazw poszczególnych kolorów (z wyjątkiem białego i czarnego) jest trudne. Dlatego nazwy kolorów są rzadko używane.

JASNOŚĆ

Jasność (ang. *brightness*, nazywana także wartością) określa ilość czerni w danym kolorze. W przypadku maksymalnej jasności w kolorze nie będzie w ogóle czerni. Natomiast przy minimalnej jasności kolor będzie bardzo ciemny.

KONTRAST

Wybierając kolory elementów oraz tła, koniecznie należy zadbać o odpowiednio duży kontrast pomiędzy nimi, tak by tekst był czytelny.

NISKI KONTRAST

Jeśli kontrast pomiędzy kolorem elementu a kolorem tła będzie niski, to odczytanie tekstu umieszczonego w elemencie może przysparzać trudności.

Brak odpowiedniego kontrastu może się okazać szczególnym problemem dla osób cierpiących na upośledzenie wzroku oraz daltonistów.

Może także stanowić utrudnienie dla osób posiadających monitory słabszej jakości, na które pada jasne światło słoneczne (co często się zdarza w przypadku korzystania z komputerów podręcznych na zewnątrz).

WYSOKI KONTRAST

Kiedy kontrast pomiędzy kolorem pierwszoplanowym a kolorem tła jest wysoki, tekst łatwiej jest odczytać.

Jeśli jednak chcemy, by użytkownicy czytali długie teksty zamieszczone na stronie, to także wysoki kontrast może być męczący.

ŚREDNI KONTRAST

W przypadku umieszczenia na stronie długich fragmentów tekstu niewielkie obniżenie kontrastu pomiędzy kolorem pierwszoplanowym a kolorem tła może nieco poprawić jego czytelność.

Kontrast można obniżyć, umieszczając ciemnoszary tekst na białym tle albo jasnoszary tekst na ciemnym tle.

Jeśli kolor tekstu zostanie odwrócony (czyli gdy umieścimy jasny tekst na ciemnym tle), jego czytelność można dodatkowo poprawić, zwiększając odległości pomiędzy poszczególnymi wierszami tekstu oraz grubość czcionki.

Kontrast można sprawdzać przy użyciu internetowego narzędzia dostępnego na stronie www.snook.ca/technical/colour_contrast/colour.html.

CSS3: NIEPRZEZROCZYSTOŚĆ opacity, rgba

CSS

```
p.one {  
background-color: rgb(0,0,0);  
opacity: 0.5;}  
p.two {  
background-color: rgb(0,0,0);  
background-color: rgba(0,0,0,0.5);}
```

rozdział-11/opacity.html

WYNIK



WYNIKI W STARSZYCH PRZEGŁĄDARKACH



W CSS3 wprowadzona została właściwość `opacity`, umożliwiająca określenie nieprzezroczystości elementu oraz dowolnego z jego elementów potomnych. Wartością tej właściwości może być liczba z zakresu od 0.0 do 1.0 (a zatem użycie wartości .5 oznacza, że element będzie w 50% nieprzezroczysty, a użycie wartości .15 — że element będzie nieprzezroczysty w 15%).

Właściwość `rgba` CSS3 pozwala na określanie kolorów w taki sam sposób jak wcześniej przy użyciu wartości RGB, lecz dodatkowo umożliwia podanie czwartej wartości określającej nieprzezroczystość. Wartość ta jest także określana mianem wartości **alfa**, a jej dopuszczalny zakres wynosi od 0.0 do 1.0 (a zatem wartość 0.5 oznacza 50% nieprzezroczystości, a wartość 0.15 — 15% nieprzezroczystości). Zastosowanie tej wartości w regule stylu będzie miało wpływ wyłącznie na wygląd danego elementu (lecz nie jego elementów potomnych).

Ponieważ niektóre przeglądarki nie rozpoznają wartości RGBA, warto stosować rozwiązania alternatywne, dzięki którym na stronie zostanie zastosowany jednolity, nieprzezroczysty kolor. Jeśli w arkuszu stylów zostały podane dwie reguły odnoszące się do tego samego elementu, to druga z nich będzie miała wyższy priorytet. Aby określić

kolor alternatywny, należy umieścić w arkuszu stylów dwie reguły: pierwsza z nich określi kolor, używając nazwy, wartości RGB lub kodu szesnastkowego, a w drugiej zostanie użyta wartość RGBA. Jeśli przeglądarka rozpozna wartość RGBA, skorzysta z niej; w przeciwnym razie użyje podanej wcześniej wartości RGB.

W czasie pisania niniejszej książki ani właściwość `opacity`, ani wartości RGBA nie były jeszcze obsługiwane przez najnowsze przeglądarki¹.

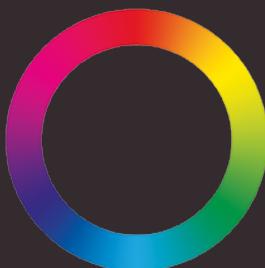
¹ W czasie przygotowywania polskiego tłumaczenia książki zarówno właściwość `opacity`, jak i wartości RGBA były już obsługiwane przez najnowsze wersje wszystkich najpopularniejszych przeglądarek — *przyp. tłum.*

CSS3: KOLORY HSL

W CSS3 został wprowadzony całkowicie nowy, intuicyjny sposób określania kolorów przy użyciu odcienia, nasycenia i jasności.

ODCIEŃ

Potocznie wyraz „odcień” bywa traktowany jak synonim słowa „kolor”. W modelu HSL odcień jest często reprezentowany przy użyciu koła, w którym kąt określa kolor; równie dobrze można go przedstawić w formie suwaka o zakresie wartości od 0 do 360.



NASYCENIE

Nasycenie określa ilość szarości w kolorze. Jest ona wyrażana jako wartość procentowa: 100% to pełne nasycenie koloru, natomiast 0% będzie odpowiadało odcieniowi szarości.



JASNOŚĆ

Jasność (ang. *lightness*) określa ilość bieli (jasność) lub czerni (ciemność). Jest ona wyrażana w formie wartości procentowej. Jasność 0% to kolor czarny, natomiast 100% to kolor biały. Jasność o wartości 50% reprezentuje normalny kolor. Jasność jest także czasami nazywana *świetlistością*.



Trzeba zwrócić uwagę, że jasność (ang. *lightness*) w modelu kolorów HSL jest czymś innym niż jasność (ang. *brightness*) w modelu HSB. Oprogramowanie graficzne (takie jak Photoshop lub GIMP) udostępnia narzędzia do wyboru kolorów, określające je według modelu HSB, w którym zmiana wartości jasności powoduje dodawanie czerni do koloru lub odejmowanie jej z koloru. W przypadku kolorów HSL zmiana jasności powoduje dodanie do koloru zarówno czerni, jak i bieli.

CSS3: HSL ORAZ HSLA hs1, hsla

CSS

rozdział-11/hsla.html

```
body {  
  background-color: #C8C8C8;  
  background-color: hsl(0,0%,78%);}  
p {  
  background-color: #ffffff;  
  background-color: hsla(0,100%,100%,0.5);}
```

WYNIK

Biologia morska

Skład wody morskiej

W wodzie morskiej można znaleźć niemalże wszystko. Dotyczy to materiałów pochodzących zarówno ze skorupy ziemskiej, jak i od organizmów żywych. Najważniejszymi parametrami wody morskiej, mającymi wpływ na morskie formy życia, są: zasolenie, rozpuszczone gazy (głównie tlen i dwutlenek węgla), składniki pokarmowe oraz poziom pH. Parametry te zmieniają się zarówno pod względem ilości, jak i wpływu wywieranego na morskie formy życia.

Ponieważ starsze przeglądarki nie rozpoznają wartości HSL ani HSLA, warto do arkusza stylów dodać regułę określającą kolory przy użyciu wartości RGB, kodów szesnastkowych lub predefiniowanych nazw. Reguła ta powinna być umieszczona w kodzie arkusza przed regułą zawierającą wartość HSL lub HSLA.

W ten sposób tworzymy regułę alternatywną, gdyż jeśli w arkuszu znajdą się dwie reguły odnoszące się do tego samego elementu, to późniejsza z nich będzie miała wyższy priorytet. Oznacza to, że jeśli przeglądarka rozpoznaje wartości HSL i HSLA, użyje reguły, w której zostały one podane. W przeciwnym razie skorzysta z reguły zapisanej wcześniej.

Właściwość hs1 została wprowadzona w CSS3 jako alternatywny sposób określania kolorów. Wartość tej właściwości zaczyna się od liter hs1, po których, w nawiasie, podane są następujące wartości:

ODCIEŃ

Odcień jest wyrażany jako miara kąta (z zakresu od 0 do 360 stopni).

NASYCENIE

Nasyenie jest wyrażane jako wartość procentowa.

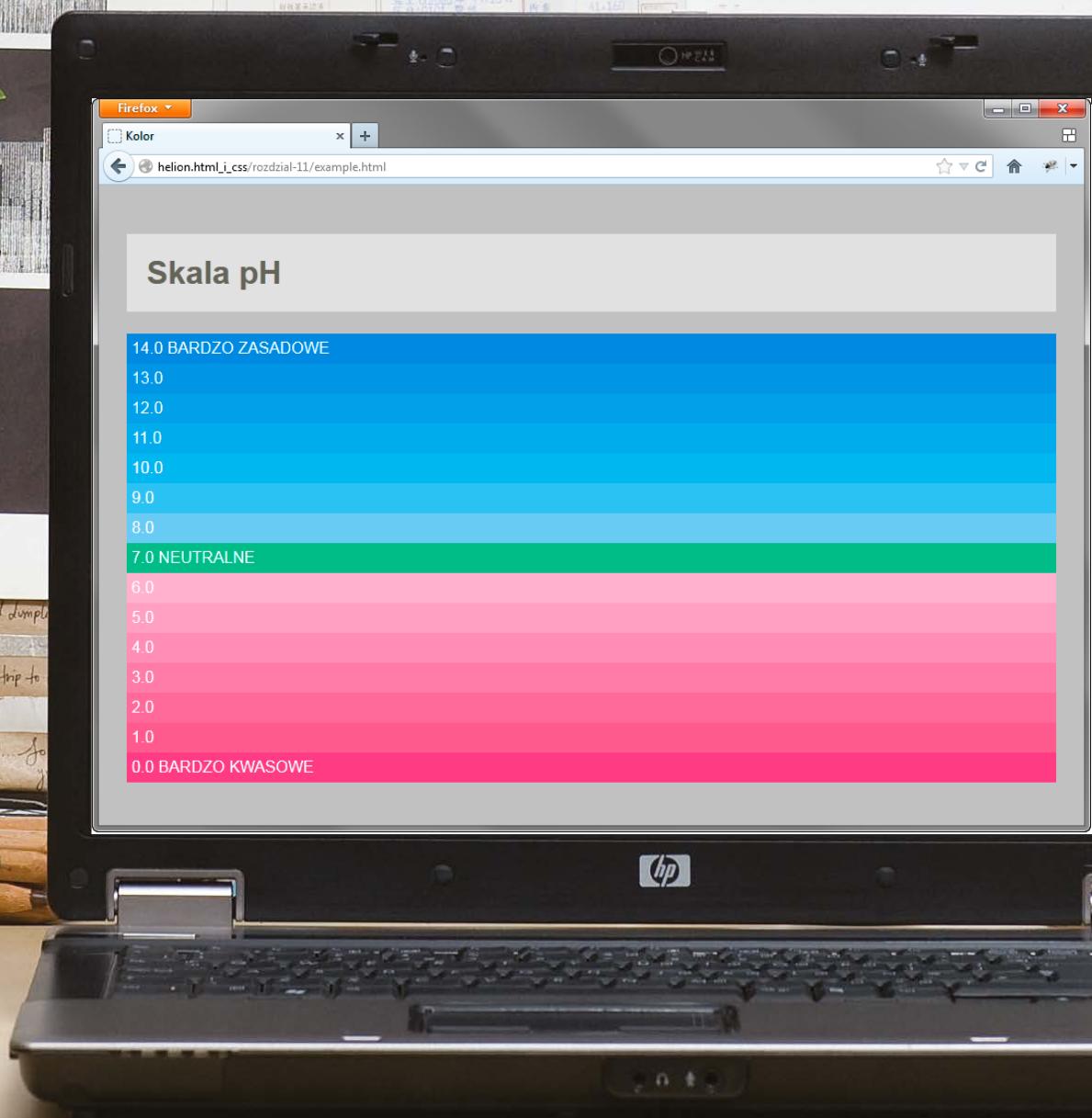
JASNOŚĆ

Jasność jest wyrażana jako wartość procentowa; przy czym 0% oznacza kolor biały, 50% to normalny, wybrany kolor, a 100% oznacza kolor czarny.

Właściwość hsla również umożliwia określenie koloru poprzez podanie jego odcienia, nasycenia oraz jasności (podobnie jak w przypadku właściwości hs1), jednak dodaje do tego czwartą wartość reprezentującą nieprzezroczystość (analogicznie jak przy właściwości rgba).

WARTOŚĆ ALFA

To wartość z zakresu od 0 do 1.0; np. wartość 0.5 oznacza 50% nieprzezroczystości, a 0.75 — 75% nieprzezroczystości.



PRZYKŁAD

KOLOR



Ten przykład wyświetla na stronie skalę pH, używając przy tym różnych sposobów określania kolorów dostępnych w CSS (takich jak nazwy, kody szesnastkowe, wartości RGB oraz HSL).

Reguła odnosząca się do elementu `<body>` określa domyślny kolor całego tekstu na stronie oraz domyślny kolor tła strony. Oba te kolory zostały podane przy użyciu nazw.

Reguła odnosząca się do elementu `<h1>` określa kolor nagłówka, używając kodu szesnastkowego. W tej regule wartość właściwości `background-color` została podana dwukrotnie. Pierwsza jest właściwością alternatywną i używa wartości podanej w formie kodu szesnastkowego, natomiast druga używa wartości HSLA i zostanie zastosowana w przeglądarkach, które takie wartości potrafią zinterpretować.

Wszystkie akapity na stronie mają różne kolory tła, reprezentujące różne poziomy kwasowości i alkaliczności. W ich przypadku kolor tła został określony przy użyciu wartości RGB.

W przykładzie została także użyta właściwość o nazwie `margin`, aby powiększyć odległości pomiędzy pudełkami poszczególnych akapitów, oraz właściwość `padding`, by zwiększyć odległość pomiędzy obramowaniem pudełka elementu a tekstem umieszczonym wewnątrz niego. (Obie te właściwości zostały opisane na stronach 312 – 313).

PRZYKŁAD

KOLOR

```
<!DOCTYPE html>
<html>
<head>
<title>Kolor</title>
<style type="text/css">
body {
    background-color: silver;
    color: white;
    padding: 20px;
    font-family: Arial, Verdana, sans-serif;}
h1 {
    background-color: #ffffff;
    background-color: hsla(0,100%,100%,0.5);
    color: #64645A;
    padding: inherit;}
p {
    padding: 5px;
    margin: 0px;}
p.zero {
    background-color: rgb(238,62,128);}
p.one {
    background-color: rgb(244,90,139);}
p.two {
    background-color: rgb(243,106,152);}
p.three {
    background-color: rgb(244,123,166);}
p.four {
    background-color: rgb(245,140,178);}
p.five {
    background-color: rgb(246,159,192);}
p.six {
    background-color: rgb(245,176,204);}
p.seven {
    background-color: rgb(0,187,136);}
p.eight {
    background-color: rgb(140,202,242);}
p.nine {
    background-color: rgb(114,193,240);}
```

PRZYKŁAD

KOLOR

```
p.ten {  
    background-color: rgb(84,182,237);}  
p.eleven {  
    background-color: rgb(48,170,233);}  
p.twelve {  
    background-color: rgb(0,160,230);}  
p.thirteen {  
    background-color: rgb(0,149,226);}  
p.fourteen {  
    background-color: rgb(0,136,221);}  
</style>  
</head>  
<body>  
    <h1>Skala pH</h1>  
    <p class="fourteen">14.0 BARDZO ZASADOWE</p>  
    <p class="thirteen">13.0</p>  
    <p class="twelve">12.0</p>  
    <p class="eleven">11.0</p>  
    <p class="ten">10.0</p>  
    <p class="nine">9.0</p>  
    <p class="eight">8.0</p>  
    <p class="seven">7.0 NEUTRALNE</p>  
    <p class="six">6.0</p>  
    <p class="five">5.0</p>  
    <p class="four">4.0</p>  
    <p class="three">3.0</p>  
    <p class="two">2.0</p>  
    <p class="one">1.0</p>  
    <p class="zero">0.0 BARDZO KWASOWE</p>  
</body>  
</html>
```


PODSUMOWANIE

KOLOR

- ▶ Kolory nie tylko ożywiają stronę internetową, lecz także ułatwiają tworzenie odpowiedniego nastroju i wzbudzają emocje.
- ▶ W kaskadowych arkuszach stylów kolory można określić na trzy sposoby — używając wartości RGB, kodów szesnastkowych lub nazw.
- ▶ W wyborze odpowiedniego koloru mogą pomóc specjalne, przeznaczone do tego celu narzędzia.
- ▶ Koniecznie należy zadbać o odpowiedni kontrast pomiędzy kolorem tekstu a tłem (w przeciwnym razie użytkownicy nie będą w stanie odczytać treści umieszczonych na stronie).
- ▶ W CSS3 do kolorów RGB została dodana nowa wartość określająca nieprzezroczystość. Te nowe wartości kolorów są określane jako RGBA.
- ▶ CSS3 pozwalają określać kolory także przy użyciu wartości HSL oraz opcjonalnej wartości reprezentującej nieprzezroczystość. W tym drugim przypadku mówimy o wartościach HSLA.

12

TEKST

- ▶ Wielkość i czcionka tekstu.
- ▶ Pogrubienie, kursywa, kapitaliki i podkreślenie.
- ▶ Odstępy pomiędzy wierszami, słowami i literami.

Właściwości pozwalające na kontrolę wyglądu tekstu można podzielić na dwie grupy:

- Właściwości mające bezpośredni wpływ na czcionkę oraz jej wygląd (czyli określające jej krój, to, czy jest ona pogrubiona, pochylona, czy normalna, oraz jej wielkość).
- Właściwości, które wywierałyby ten sam wpływ na postać tekstu niezależnie od użytej czcionki (do nich można zaliczyć kolor tekstu oraz odległość pomiędzy wyróżnionymi literami).

Formatowanie tekstu może znaczco oddziaływać na to, jak czytelne będą tworzone przez nas strony. Przy okazji opisywania właściwości prezentowanych w tym rozdziale podam także kilka wskazówek z zakresu projektowania stron,



TERMINOLOGIA ZWIĄZANA Z KROJEM CZCIONEK

CZCIONKA SZERYFOWA

Czcionki szeryfowe charakteryzują się dodatkowymi kreseczkami na końcach głównych linii tworzących znaki. Elementy te są nazywane szeryfami.

W materiałach drukowanych czcionki szeryfowe były tradycyjnie używane w dłuższych fragmentach tekstu, gdyż powszechnie uważa się, że ułatwiają one czytanie.

CZCIONKA BEZSZERYFOWA

Czcionki bezszeryfowe mają proste zakończenia liter, dlatego ich wygląd jest znacznie bardziej przejrzysty.

Ekrany komputerów mają znacznie mniejszą rozdzielcość od druku. Dlatego jeśli tekst ma niewielkie rozmiary, to zastosowanie czcionki bezszeryfowej sprawi, że będzie on bardziej przejrzysty i łatwiejszy do odczytania.

CZCIONKA O STAŁEJ SZEROKOŚCI ZNAKÓW

W przypadku tych czcionek wszystkie znaki mają taką samą szerokość. (W innych czcionkach poszczególne znaki mogą mieć różne szerokości).

Czcionek o stałej szerokości znaków powszechnie używa się do zapisywania kodu, gdyż jego poszczególne znaki są dzięki temu równo ułożone, co ułatwia analizę.



GRUBOŚĆ

- Cienka
- Średnia
- Pogrubiona**
- Bardzo gruba**

STYL

- Normalny
- Kursywa*
- Pochylone*

ROZCIĄGNIĘCIE

- Zwarty
- Normalny
- Rozszerzony**

Grubość czcionki nie tylko nadaje fragmentom tekstu większe znaczenie, lecz także wypływa na ilość jasnej przestrzeni na stronie oraz na jej kontrast.

Kursywy nadają tekstem charakter pisma odreżnego. Czcionki pochylone używają znaków normalnych czcionek, tyle że są one wyświetlane pod pewnym kątem.

W przypadku zwartej (wąskiej) wersji czcionki jej znaki są węższe, a odległości pomiędzy nimi mniejsze. W przypadku wersji rozszerzonej znaki są grubsze i umieszczone dalej od siebie.

WYBÓR KROJU CZCIONKI UŻYWANEGO NA STRONIE

Wybierając kój pisma z przeznaczeniem na stronę internetową, koniecznie trzeba pamiętać o tym, że zostanie on zastosowany wyłącznie w przypadku, gdy będzie zainstalowany na komputerze użytkownika.

SERIF

Czcionki szeryfowe charakteryzują się dodatkowymi kresekami na końcach głównych linii tworzących znaki.

PRZYKŁADY:

Georgia
Times
Times New Roman

SANS-SERIF

Czcionki bezszeryfowe mają proste zakończenia liter, dlatego wyglądają znacznie bardziej przejrzyste.

PRZYKŁADY:

Arial
Verdana
Helvetica

W efekcie, tworząc strony WWW, zazwyczaj używa się ograniczonego zestawu czcionek instalowanych w większości komputerów (zostały one przedstawione powyżej). Istnieją jednak techniki pozwalające ominąć te ograniczenia (opisano je na stronach 270 – 271).

Można także podać więcej niż jedną czcionkę i określić kolejność, w jakiej będą używane (na wypadek gdyby pierwsze z nich nie były dostępne na komputerze użytkownika). W takich przypadkach mówimy o stosowaniu **stosu czcionek**.

MONOSPACE

To grupa czcionek, w których każdy znak ma taką samą szerokość. (W pozostałych czcionkach ich znaki mogą mieć różne szerokości).

PRZYKŁADY:

Courier

Courier New

CURSIVE

Kursywy mają albo połączenia pomiędzy poszczególnymi znakami, albo cechy pisma pochylonego, takiego jak pismo odręczne.

PRZYKŁADY:

Comic Sans MS

Monotype Corsiva

FANTASY

Do tej grupy należą czcionki dekoracyjne, których często używa się w tytułach. Nie są one projektowane z myślą o użyciu w długich fragmentach tekstu.

PRZYKŁADY:

Impact

Swis921EU

Przeglądarki mają udostępniać przynajmniej jedną czcionkę z każdej z tych kategorii. Dlatego powszechnie stosowanym rozwiązaniem jest umieszczanie za preferowanymi czcionkami odpowiedniej ogólnej nazwy kroju pisma.

Gdy chcemy użyć np. czcionki szeryfowej, możemy umieścić w arkuszu stylów następującą regułę:
`font-family: Georgia, Times, serif;`

TECHNIKI ZAPEWNIJĄCE MOŻLIWOŚĆ SZERSZEGO WYBORU CZCIONEK

Dostępnych jest kilka sposobów pozwalających na stosowanie innych czcionek niż te przedstawione na poprzednich stronach. Trzeba jednak pamiętać, że czcionki są chronione prawami autorskimi, a zatem techniki, których można używać, także są ograniczone licencjami czcionek.

FONT-FAMILY

Czcionka musi być zainstalowana na komputerze użytkownika. Używaną czcionkę określa się przy użyciu CSS

FONT-FACE

CSS określa, skąd można pobrać czcionkę, jeśli nie jest ona dostępna na komputerze użytkownika

FONT-FACE Z WYKORZYSTANIEM USŁUGI

Usługi komercyjne zapewniają użytkownikom dostęp do szerokiej gamy czcionek, z których można korzystać przy użyciu właściwości @font-face

OPISANA NA

Stronach 272 – 273

Stronach 276 – 277

Stronach 276 – 277

PROBLEMY

Liczba czcionek zainstalowanych na komputerach większości użytkowników jest bardzo ograniczona

Użytkownik musi pobrać plik czcionki, co może przyczynić się do wydłużenia czasu wyświetlania strony

Trzeba wnosić opłaty za licencje dla twórców czcionek

LICENCJE

Twórca strony nie rozpowszechnia czcionki, zatem nie musi wnosić żadnych opłat licencyjnych

Licencja, na której czcionka jest udostępniana, musi pozwalać na jej rozpowszechnianie przy użyciu właściwości @font-face

Wszelkie kwestie związane z uzyskaniem licencji na stosowanie czcionki załatwia usługa

WYBÓR CZCIONEK

Liczba dostępnych czcionek jest ograniczona, gdyż muszą one być zainstalowane na komputerze użytkownika

Liczba czcionek jest ograniczona, gdyż rzadko która czcionka może być rozpowszechniana w ten sposób

Każda usługa udostępnia różny zbiór czcionek zależnie od umów podpisanych z ich twórcami

Nadają się do prezentowania tekstów o dowolnej długości

Jeśli strona jest projektowana na komputerze Mac, koniecznie trzeba sprawdzić, jak używane czcionki wyglądają na komputerze PC, gdyż ich krawędzie nie są równie gładkie. Jeśli jednak projektujemy na komputerze PC, to na komputerze Mac strony powinny wyglądać tak samo dobrze.

OBRAZY	SIFR	CUFON
Jeśli chcemy wyświetlić tekst przy użyciu innej czcionki, możemy umieścić na stronie obraz zawierający ten tekst	Czcionka jest osadzana w filmie Flash, a skrypt JavaScript zastępuje wskazany tekst jego odpowiednikiem wyświetlonym przy użyciu technologii Flash	Cufon zapewnia podobne możliwości funkcjonalne do sIFR. Metoda ta korzysta z kodu JavaScript, by zmienić tekst na jego wektorowy odpowiednik prezentowany przy użyciu technologii SVG lub VML
Stronach 98 – 99 oraz 108 – 112	Więcej informacji można znaleźć na stronie poświęconej książce	Więcej informacji można znaleźć na stronie poświęconej książce
Osoby korzystające z czytników ekranowych będą mogły poznać zawartość wyłącznie na podstawie zawartości atrybutu alt	Ta metoda będzie działać wyłącznie, jeśli używane urządzenie obsługuje język JavaScript i technologię Flash	Wymaga włączenia obsługi języka JavaScript. Co więcej, użytkownicy nie mogą zaznaczać tekstu, a sam tekst nie może się zmieniać po wskazaniu go myszą
Można używać dowolnych czcionek, do których posiadamy stosowne licencje (gdyż czcionki nie są rozpowszechniane)	Wielu twórców czcionek zezwala na stosowanie tej techniki, choć czasami konieczny jest zakup specjalnej licencji na zastosowanie w internecie	Podobnie jak w przypadku metody sIFR, niektórzy twórcy czcionek zezwalają na ich stosowanie przy użyciu metody CUFON, niemniej trzeba sprawdzić licencję czcionki
Dostępny jest szeroki zakres czcionek, gdyż można używać wszystkich czcionek, na których używanie mamy licencje	Ta metoda zapewnia możliwość korzystania z wielu czcionek, gdyż wielu czołowych twórców czcionek zezwala na stosowanie ich w taki sposób	W tym przypadku wybór czcionek jest nieco mniejszy niż w razie zastosowania metody sIFR, gdyż twórcy czcionek nie są przychylni tej metodzie

Nie nadają się do prezentowania długich fragmentów tekstu

OKREŚLANIE KROJU CZCIONKI **font-family**

Właściwość `font-family` pozwala określić krój czcionki używanej do prezentowania tekstu umieszczonego wewnątrz elementu (lub elementów), do którego odnosi się reguła. Wartością tej właściwości jest nazwa używanego kroku czcionki.

Osoby odwiedzające stronę muszą mieć daną czcionkę zainstalowaną na swoim komputerze, w przeciwnym razie nie zostanie ona użyta.

We właściwości `font-family` można także podać listę nazw czcionek (oddzielonych od siebie przecinkami), dzięki czemu, jeśli na komputerze użytkownika nie będzie dostępnej pierwszej z podanych czcionek, przeglądarka spróbuje użyć następnej.

Często stosuje się rozwiązanie, w którym na samym końcu takiej listy zostaje dodana ogólna nazwa czcionki (ogólne nazwy czcionek zostały podane na stronach 268 – 269).

Jeśli nazwa czcionki składa się z więcej niż jednego słowa, należy ją zapisać w cudzysłowie.

Projektanci uważają, że strony wyglądają najlepiej, jeśli używa się na nich nie więcej niż trzech różnych krojów pisma.

We wszystkich przykładach zamieszczonych w tym rozdziale będziemy używali nieco rozszerzonej wersji dokumentu HTML przedstawionego na tej stronie.

rozdział-12/font-family.html

HTML + CSS

```
<!DOCTYPE html>
<html>
  <head>
    <title>Właściwość font-family</title>
    <style type="text/css">
      body {
        font-family: Georgia, Times, serif;
      }
      h1, h2 {
        font-family: Arial, Verdana, sans-serif;
      }
      .credits {
        font-family: "Courier New", Courier,
        monospace;
      }
    </style>
  </head>
  <body>
    <h1>Briardy</h1>
    <p class="credits">napisała Ivy Duckett</p>
    <p class="intro"><a class="breed"
      href="http://pl.wikipedia.org/wiki/
      Briard">Briard</a>,
      inaczej owczarek francuski briard, jest dużym
      psem pasterskim tradycyjnie używanym
      do wypasania...</p>
  </body>
</html>
```

WYNIK

Briardy

napisała Ivy Duckett

[Briard](#), inaczej owczarek francuski briard, jest dużym psem pasterskim tradycyjnie używanym do wypasania i pilnowania owiec.

Historia rasy

Uważa się, że psy te zostały wyhodowane we Francji. Od wieków były używane do wypasania i ochrony owiec. W armii francuskiej służyły jako psy wartownicze, do przenoszenia wiadomości oraz, ze względu na doskonały słuch, do poszukiwania rannych. W czasie pierwszej wojny światowej ich umiejętności wykorzystywano w tak dużym stopniu, że niemal doprowadzono do wyginięcia rasy. Aktualnie ich populacja cały czas z wolna rośnie. Charlemagne, Napoleon, Thomas Jefferson oraz Lafayette mieli psy tej rasy.

ROZMIAR CZCIONKI

font-size

CSS

rozdział-12/font-size.html

```
body {  
  font-family: Arial, Verdana, sans-serif;  
  font-size: 12px;}  
h1 {  
  font-size: 200%;}  
h2 {  
  font-size: 1.3em;}
```

WYNIK

Briardy

napisała Ivy Duckett

[Briard](#), inaczej owczarek francuski briard, jest dużym psem pasterskim tradycyjnie używanym do wypasania i pilnowania owiec.

Historia rasy

Uważa się, że psy te zostały wyhodowane we Francji. Od wieków były używane do wypasania i ochrony owiec. W armii francuskiej służyły jako psy wartownicze, do przenoszenia wiadomości oraz, ze względu na doskonały słuch, do poszukiwania rannych. W czasie pierwszej wojny światowej ich umiejętności wykorzystywano w tak dużym stopniu, że niemal doprowadzono do wyginięcia rasy. Aktualnie ich populacja cały czas z wolna rośnie. Charlemagne, Napoleon, Thomas Jefferson oraz Lafayette mieli psy tej rasy.

WARTOŚCI EM

Jednostka em to odpowiednik szerokości małej litery „m”.

Te miary wielkości czcionek zostały dokładniej opisane na następnej stronie.

Właściwość font-size pozwala określić wielkość czcionki. Można to zrobić na kilka sposobów, z których najczęściej używane są:

PIKSELE

Wyrażenie wielkości czcionki w pikselach jest bardzo popularne, gdyż zapewnia projektantom bardzo precyzyjną kontrolę nad obszarem zajmowanym przez tekst. Za liczbą pikseli umieszczane są litery px.

WARTOŚCI PROCENTOWE

Domyślnie tekst wyświetlany przez przeglądarki ma wielkość 16 pikseli. A zatem wartość 75% odpowiada 12 pikselom, a 200% — 32 pikselom.

Jeśli zastosujemy regułę, która określa, że cały tekst umieszczony w elemencie <body> ma mieć 75% domyślnej wielkości (tak by miał on wielkość 12 pikseli), a następnie kolejną regułę ograniczającą wielkość tekstu w wybranym elemencie do 75%, to w efekcie tekst będzie miał wielkość 9 pikseli (bo tyle wynosi 75% wartości 12 px).

SKALE PISMA

Być może Czytelnik zauważyl, że programy takie jak Word, Photoshop i InDesign udostępniają te same wielkości czcionek.

Jest to możliwe dzięki temu, że czcionki są określane według tej samej skali, opracowanej w XVI wieku przez europejskich zecerów.

Uznaje się, że ta skala wielkości czcionek jest miła dla oka i dlatego w ciągu ostatnich 400 lat niewiele się zmieniła.

Korzystanie z tej skali wielkości podczas projektowania stron sprawi, że będą one wizualnie atrakcyjniejsze.

Na następnej stronie jest mowa o tym, w jaki sposób uzyskiwać te same wielkości z wykorzystaniem pikseli, wartości procentowych i jednostek em.

Osoby zajmujące się składaniem materiałów drukowanych często określają wielkość tekstu nie w pikselach, lecz w punktach (dlatego w skali widocznej obok za liczbami umieszczone zostały litery pt). Piksele wielkością odpowiadając mniej więcej punktom, gdyż punkt to dokładnie 1/72 cala, a większość monitorów komputerowych ma rozdzielcość 72 pikseli na cal.

Domyślna wielkość czcionki w przeglądarkach to 16 pikseli. Dlatego w razie stosowania wartości procentowych lub jednostek em konieczne jest przeliczenie wielkości tekstu na podstawie tej wartości domyślnej. Można by np. przeskalać podstawowy tekst do wielkości 12 pikseli, a nagłówki — do wielkości 24 pikseli.

Ostatnio zdarza się, że projektanci pozostawiają podstawowy tekst w domyślnej wielkości 16 pikseli i skalują inne czcionki względem tej wartości.

Kiedy po raz pierwszy spotkamy się z tekstem o wielkości 16 pikseli, możemy uznać, że wydaje się on dosyć duży. Jednak większość osób uważa, że kiedy się już do niego przyzwyczaić, to czytanie takiego dużego tekstu jest łatwiejsze; po ponownym wyświetleniu tekstu o wielkości 12 pikseli okazuje się, że jest on całkiem mały.

8 pt

9 pt

10 pt

11 pt

12 pt

14 pt

18 pt

24 pt

36 pt

48 pt

60 pt

72 pt

JEDNOSTKI WIELKOŚCI CZCIONEK

PIKSELE

Skala 12-pikselowa

h1	24px
h2	18px
h3	14px
body	12px

=

WARTOŚCI PROCENTOWE

h1	200%
h2	150%
h3	117%
body	75%

JEDNOSTKI EM

h1	1.5em
h2	1.3em
h3	1.17em
body	100%
p	0.75em

Skala 16-pikselowa

h1	32px
h2	24px
h3	18px
body	16px

=

h1	200%
h2	150%
h3	133%
body	100%

=

h1	2em
h2	1.5em
h3	1.125em
body	100%
p	1em

Określanie wielkości czcionek w pikselach jest najlepszym sposobem na wymuszenie, by teksty wyświetlane w przeglądarce miały dokładnie taką wielkość, jaką zaplanowaliśmy (gdyż wielkości wyrażone w wartościach procentowych lub jednostkach em mogą się zmieniać, jeśli użytkownik zmodyfikuje domyślne ustawienia przeglądarki).

Wymiary pikseli zależą jedynie od rozdzielczości monitora, dlatego ten sam tekst na ekranie o rozdzielczości 800×600 będzie się wydawał większy niż na ekranie o rozdzielczości 1280×800 .

Można także określić wielkości czcionek w punktach (pt), a nie w pikselach, jednak takie rozwiązanie należy stosować wyłącznie w przypadku tworzenia stylów na stronę przeznaczoną do drukowania.

Tekst wyświetlany w przeglądarkach ma domyślną wielkość 16 pikseli. Korzystając z wartości procentowych, można przeskalać tę wartość, by uzyskać tekst o domyślnej wielkości 12 pikseli, a następnie określić wielkość nagłówków względem tej nowej wartości domyślnej.

Użytkownicy mogą zmieniać domyślną wielkość czcionki w przeglądarce. W takim przypadku czcionki będą miały taką samą skalę, jaką określił twórca strony, lecz ich faktyczne wielkości będą inne.

Jednostki em pozwalają określić wielkość tekstu w zależności od jego wielkości w elemencie rodzica. Ponieważ domyślna wielkość tekstu w przeglądarce to 16 pikseli, w przypadku jednostek em można zastosować podobne reguły jak przy wartościach procentowych.

Ze względu na to, że użytkownicy mogą zmieniać domyślną wielkość czcionki w ustawieniach przeglądarki, faktyczne wielkości czcionek mogą być inne, niż zamierzał projektant strony.

Dodatkowa reguła dotycząca elementu p, przedstawiona w powyższej tabelce, ma zapewnić wyświetlanie tekstów o odpowiedniej wielkości w przeglądarkach Internet Explorer 6 i 7. Bez tej dodatkowej reguły wielkości innych tekstów w tych dwóch przeglądarkach będą zbyt duże.

MOŻLIWOŚCI STOSOWANIA DODATKOWYCH CZCIONEK

@font-face

Reguła @font-face pozwala używać czcionek, nawet jeśli nie są one zainstalowane na komputerze użytkownika przeglądającego stronę. Jest to możliwe dzięki podaniu ścieżki do pliku czcionki, który, w razie konieczności, może zostać pobrany.

Ponieważ technika ta pozwala na skopiowanie czcionki na komputer użytkownika, ważne jest, by jej licencja zezwalała na takie zastosowania.

Czcionki dodaje się do stron WWW przy użyciu reguły @font-face, w sposób przedstawiony na przykładzie zamieszczonym obok.

font-family

Ta właściwość określa nazwę czcionki, której następnie można używać w dalszej części arkuszu stylów, we właściwości font-family (jak widać na przykładzie reguły odnoszących się do elementów <h1> i <h2>).

src

Ta właściwość określa ścieżkę dostępu do czcionki. Aby ta technika działała we wszystkich przeglądarkach, konieczne będzie zapewne podanie ścieżek do kilku różnych wersji tej samej czcionki (zgodnie z informacjami podanymi na następnej stronie).

format

Ta właściwość określa format, w jakim została zapisana czcionka. (Więcej informacji

rozdział-12/font-face.html

CSS

```
@font-face {  
    font-family: 'ChunkFiveRegular';  
    src: url('fonts/chunkfive.eot');}  
h1, h2 {  
    font-family: ChunkFiveRegular, Georgia, serif;}
```

WYNIK

Briardy

napisała Ivy Duckett

[Briard](#), inaczej owczarek francuski briard, jest dużym psem pasterskim tradycyjnie używanym do wypasania i pilnowania owiec.

Historia rasy

Uważa się, że psy te zostały wyhodowane we Francji. Od wieków były używane do wypasania i ochrony owiec. W armii francuskiej służyły jako psy wartownicze, do przenoszenia wiadomości oraz, w względzie na doskonąły słuch, do poszukiwania rannych. W czasie pierwszej wojny światowej ich umiejętności wykorzystywano w tak dużym stopniu, że niemal doprowadzono do wyginięcia rasy. Aktualnie ich populacja cały czas z wolna rośnie. Charlemagne, Napoleon, Thomas Jefferson oraz Lafayette mieli psy tej rasy.

na ten temat można znaleźć na następnej stronie).

Wielu twórców czcionek nie pozwala na stosowanie ich w taki sposób, istnieją jednak ogólnie dostępne czcionki, których można tak używać. Informacje o nich znajdują się na stronach:

www.fontsquirrel.com
www.fontex.org
www.openfontlibrary.org

Przy przeglądaniu dostępnych w tych witrynach czcionek należy sprawdzać ich licencje, gdyż niektóre z nich są dostępne tylko do użytku prywatnego (czyli nie na stronach mających charakter komercyjny).

W sieci istnieje kilka witryn zapewniających dostęp do czcionek komercyjnych. Jest to możliwe dzięki temu, że wynegocjowały dla swoich użytkowników możliwość darmowego korzystania z wybranych czcionek. Do takich witryn należą:

www.typekit.com
www.kernest.com
www.fontspring.com

Także firma Google udostępnia zbiór czcionek, z których można korzystać bezpłatnie. Aby ich użyć, nie trzeba stosować reguły @font-face we własnym arkuszu stylów, lecz skorzystać z pliku CSS na serwerze Google: www.google.com/webfonts

PREZENTACJA FORMATÓW CZCIONEK

CSS

rozdział-12/understanding-font-formats.html

```
@font-face {  
    font-family: 'ChunkFiveRegular';  
    src: url('fonts/chunkfive.eot');  
    src: url('fonts/chunkfive.eot?#iefix')  
        format('embedded-opentype'),  
        url('fonts/chunkfive.woff') format('woff'),  
        url('fonts/chunkfive.ttf')  
        format('truetype'),  
        url('fonts/chunkfive.svg#ChunkFiveRegular')  
        format('svg');}
```

Różne przeglądarki obsługują różne formaty zapisu czcionek (analogicznie jak różne formaty audio i wideo), a zatem by czcionka była dostępna we wszystkich możliwych przeglądarkach, trzeba ją będzie zapisać w kilku różnych formatach.

Jeśli nie dysponujemy wszystkimi niezbędnymi formatami, możemy skopiować czcionkę do witryny FontSquirrel, która dokona odpowiednich konwersji:

www.fontsquirrel.com/fontface/generator

Witryna ta przygotuje także kod CSS zawierający odpowiednią regułę @font-face. To bardzo przydatne rozwiązanie, gdyż w przypadku korzystania z czcionek zapisanych w wielu formatach właściwości src oraz format reguły @font-face mogą być dosyć skomplikowane.

Przykład takiej bardziej skomplikowanej reguły @font-face został przedstawiony obok.

Poszczególne formaty czcionki powinny zostać wymienione w kodzie w następującej kolejności:

1. **eot**,
2. **woff**,
3. **ttf/otf**,
4. **svg**.

PRZEGŁĄDARKA	FORMAT			
	eot	woff	ttf/otf	svg
Chrome (wszystkie wersje)				●
Chrome 6+	●	●	●	●
Firefox 3.5		●		
Firefox 3.6	●	●		
IE 5 – 8	●			
IE 9+	●	●	●	
Opera 10+		●	●	●
Safari 3.1+		●	●	●
iOS < 4.2				●
iOS 4.2+		●	●	

Ponieważ przeglądarka musi pobrać plik czcionki, aby móc jej użyć, użytkownik może zauważyc efekt chwilowej zmiany wyglądu tekstu (w języku angielskim jest on określany jako *FOUC* — *Flash Of Unstyled Content* lub *FOUT* — *Flash Of Unstyled Text*). Istnieją dwa sposoby, których

można użyć, by spróbować ograniczyć ten efekt. Pierwszym jest usunięcie z czcionki niepotrzebnych znaków, a drugim — skorzystanie z sieci udostępniania treści (CDN — ang. *Content Delivery Network*, czyli specjalnej usługi umożliwiającej szybkie pobieranie plików z internetu).

POGRUBIENIE

font-weight

Właściwość `font-weight` pozwala na tworzenie tekstu pogrubionego. Ma ona kilka predefiniowanych wartości, z których najczęściej stosowane są dwie:

normal

Ta wartość sprawia, że tekst ma normalną grubość.

bold

Ta wartość powoduje pogrubienie tekstu.

W przedstawionym obok przykładzie widać, że element posiadający atrybut `class` o wartości `credits` został pogrubiony.

Można się zastanawiać, dlaczego wprowadzono wartość `normal`. Wyobraźmy sobie, że dla elementu `<body>` tworzymy regułę, która sprawia, że cały tekst strony jest pogrubiony. W takim przypadku może się nam przydać możliwość wyświetlenia pewnego fragmentu tekstu czcionką o normalnej grubości. Innymi słowy, wartość ta odgrywa rolę „wyłącznika” pogrubienia.

rozdział-12/font-weight.html

CSS

```
.credits {  
    font-weight: bold;}
```

WYNIK

Briardy

napisała Ivy Duckett

[Briard](#), inaczej owczarek francuski briard, jest dużym psem pasterskim tradycyjnie używanym do wypasania i pilnowania owiec.

Historia rasy

Uważa się, że psy te zostały wyhodowane we Francji. Od wieków były używane do wypasania i ochrony owiec. W armii francuskiej służyły jako psy wartownicze, do przenoszenia wiadomości oraz, ze względu na doskonąłego słuchu, do poszukiwania rannych. W czasie pierwszej wojny światowej ich umiejętności wykorzystywano w tak dużym stopniu, że niemal doprowadzono do wyginięcia rasy. Aktualnie ich populacja cały czas z wolna rośnie. Charlemagne, Napoleon, Thomas Jefferson oraz Lafayette mieli psy tej rasy.

KURSYWA font-style

CSS

rozdział-12/font-style.html

```
.credits {  
    font-style: italic;}
```

WYNIK

Briardy

napisala Ivy Duckett

[Briard](#), inaczej owczarek francuski briard, jest dużym psem pasterskim tradycyjnie używanym do wypasania i pilnowania owiec.

Historia rasy

Uważa się, że psy te zostały wyhodowane we Francji. Od wieków były używane do wypasania i ochrony owiec. W armii francuskiej służyły jako psy wartownicze, do przenoszenia wiadomości oraz, ze względu na doskonały słuch, do poszukiwania rannych. czasie pierwszej wojny światowej ich umiejętności wykorzystywano w tak dużym stopniu, niemal doprowadzono do wyginięcia rasy. Aktualnie ich populacja cały czas z wolna rośnie. Charlemagne, Napoleon, Thomas Jefferson oraz Lafayette mieli psy tej rasy.

Jeśli chcemy wyświetlić tekst kursywą, możemy skorzystać z właściwości `font-style`. Może ona przyjmować trzy wartości:

normal

Zastosowanie tej wartości sprawi, że tekst zostanie wyświetlony normalnym stylem (a nie kursywą lub czcionką pochyloną).

italic

Użycie tej wartości skutkuje zastosowaniem kursywy.

oblique

Użycie tej wartości skutkuje zastosowaniem czcionki pochylonej.

W przykładzie przedstawionym obok widać, że kursywą został wyświetlony element, którego atrybut `class` ma wartość `credits`.

Kursywy są zazwyczaj odrębnymi wersjami czcionek, stylizowanymi na pismo odręczne, natomiast styl pochylony polega na zastosowaniu znaków z normalnej wersji czcionki i wyświetlaniu ich pod pewnym kątem.

Nierazko zdarza się, że przeglądarka nie jest w stanie odszukać konkretnej czcionki w wersji kursywy; w takich przypadkach używa ona odpowiedniego algorytmu, który wyświetla znaki normalnej wersji czcionki pod pewnym kątem; oznacza to, że znacząca część tekstu wyświetlanych na stronach WWW kursywą to w rzeczywistości tekst pochylony.

WIELKIE I MAŁE LITERY

text-transform

Właściwość `text-transform` jest używana do zmiany wielkości liter w tekście i może przyjmować następujące wartości:

uppercase

Ta wartość sprawi, że tekst będzie wyświetlany wielkimi literami.

lowercase

Ta wartość sprawi, że tekst będzie wyświetlany małymi literami.

capitalize

Użycie tej wartości spowoduje, że każde słowo będzie się zaczynać od wielkiej litery.

W tym przykładzie element `<h1>` został wyświetlony wielkimi literami, element `<h2>` — małymi, natomiast w elemencie, którego atrybut `class` ma wartość `credits`, pierwsze litery każdego słowa są zamieniane na wielkie.

W kodzie HTML strony słowo „napisała” zaczyna się od małej litery.

W razie wyświetlania tekstu wielkimi literami przy użyciu tej właściwości warto skorzystać z właściwości `letter-spacing` (opisanej na stronie 283), by powiększyć odstępy pomiędzy znakami. Poprawi to czytelność tekstu.

rozdział-12/text-transform.html

CSS

```
h1 {  
  text-transform: uppercase;}  
h2 {  
  text-transform: lowercase;}  
.credits {  
  text-transform: capitalize;}
```

WYNIK

BRIARDY

Napisała Ivy Duckett

Briard, inaczej owczarek francuski briard, jest dużym psem pasterskim tradycyjnie używanym do wypasania i pilnowania owiec.

Historia rasy

Uważa się, że psy te zostały wyhodowane we Francji. Od wieków były używane do wypasania i ochrony owiec. W armii francuskiej służyły jako psy wartownicze, do przenoszenia wiadomości oraz, ze względu na doskonały słuch, do poszukiwania rannych. W czasie pierwszej wojny światowej ich umiejętności wykorzystywano w tak dużym stopniu, że niemal doprowadzono do wyginięcia rasy. Aktualnie ich populacja cały czas z wolna rośnie. Charlemagne, Napoleon, Thomas Jefferson oraz Lafayette mieli psy tej rasy.

PODKREŚLENIE I PRZEKREŚLENIE

text-decoration

CSS

rozdział-12/text-decoration.html

```
.credits {  
  text-decoration: underline;}  
a {  
  text-decoration: none;}
```

WYNIK

Briardy

napisała Ivy Duckett

Briard, inaczej owczarek francuski briard, jest dużym psem pasterskim tradycyjnie używanym do wypasania i pilnowania owiec.

Historia rasy

Uważa się, że psy te zostały wyhodowane we Francji. Od wieków były używane do wypasania i ochrony owiec. W armii francuskiej służyły jako psy wartownicze, do przenoszenia wiadomości oraz, ze wzgórza na doskonaly słuch, do poszukiwania rannych. Czasie pierwszej wojny światowej ich umiejętności wykorzystywano w tak dużym stopniu, że doprowadzono do wyginięcia rasy. Aktualnie ich populacja cały czas z wolna rośnie. Charlemagne, Napoleon, Thomas Jefferson oraz Lafayette mieli psy tej rasy.

Właściwość ta jest powszechnie używana przez projektantów stron w celu usuwania podkreseń z łączy. Na stronach 289 – 290 pokazano, w jaki sposób dodawać lub usuwać podkreślenie, kiedy użytkownik umieści na łączu wskaźnik myszy.

Właściwość `text-decoration` może przyjmować następujące wartości:

none

Usuwa wszelkie elementy dekoracyjne zastosowane w tekście.

underline

Dodaje linię poniżej znaków.

overline

Dodaje linię powyżej znaków.

line-through

Dodaje linię, która przekreśla znaki.

blink

Dodaje animację, która sprawia, że tekst pojawia się i znika (rozwiążanie to nie jest dobrze przyjmowane, gdyż powszechnie uważa się je za denerwujące).

W przykładzie przedstawionym obok element z atrybutem `class` o wartości `credits` został podkreślony. Dodatkowo nazwa rasy (będąca łączem) nie jest podkreślona, choć domyślnie — jako łączce — powinna być.

INTERLINIA

Line-height

Interlinia jest typograficznym terminem oznaczającym pionową odległość pomiędzy wierszami. W typografii fragmenty znaków umieszczone poniżej linii bazowej są nazywane **wydłużeniem dolnym**, natomiast najwyższe fragmenty znaków — **wydłużeniem górnym**. Interlinia jest mierzona pomiędzy wydłużeniem dolnym górnego wiersza tekstu a wydłużeniem górnym dolnego wiersza.

Hej, ty
interlinia
font-size
Hej, ty line-height

W CSS właściwość line-height określa wysokość całego wiersza tekstu, a zatem interlinię stanowi różnica pomiędzy wartościami właściwości line-height oraz font-size (co pokazano na powyższym rysunku).

Powiększanie wartości właściwości line-height powoduje zwiększenie odstępu pomiędzy kolejnymi wierszami tekstu.

Powiększenie domyślnej wysokości interlinii może poprawić czytelność tekstu. Pionowy odstęp pomiędzy poszczególnymi wierszami tekstu powinien być większy od odstępu pomiędzy wyrazami, gdyż ułatwia to przesuwanie wzroku w poziomie,

rozdział-12/line-height.html

CSS

```
p {  
  line-height: 1.4em;}
```

Briardy

napisała Ivy Duckett

[Briard](#), inaczej owczarek francuski briard, jest dużym psem pasterskim tradycyjnie używanym do wypasania i pilnowania owiec.

Historia rasy

Uważa się, że psy te zostały wyhodowane we Francji. Od wieków były używane do wypasania i ochrony owiec. W armii francuskiej służyły jako psy wartownicze, do przenoszenia wiadomości oraz, ze względu na doskonąłego słuchu, do poszukiwania rannych. W czasie pierwszej wojny światowej ich umiejętności wykorzystywano w tak dużym stopniu, że niemal doprowadzono do wyginięcia rasy. Aktualnie ich populacja cały czas z wolna rośnie. Charlemagne, Napoleon, Thomas Jefferson oraz Lafayette mieli psy tej rasy.

WYNIK

Briardy

napisała Ivy Duckett

[Briard](#), inaczej owczarek francuski briard, jest dużym psem pasterskim tradycyjnie używanym do wypasania i pilnowania owiec.

Historia rasy

Uważa się, że psy te zostały wyhodowane we Francji. Od wieków były używane do wypasania i ochrony owiec. W armii francuskiej służyły jako psy wartownicze, do przenoszenia wiadomości oraz, ze względu na doskonąłego słuchu, do poszukiwania rannych. W czasie pierwszej wojny światowej ich umiejętności wykorzystywano w tak dużym stopniu, że niemal doprowadzono do wyginięcia rasy. Aktualnie ich populacja cały czas z wolna rośnie. Charlemagne, Napoleon, Thomas Jefferson oraz Lafayette mieli psy tej rasy.

WYNIK BEZ UŻYCIA CSS

a nie w pionie. Dobrym ustawieniem wyjściowym są wartości z zakresu od 1.4 em do 1.6 em. Ze względu na to, że użytkownicy mogą modyfikować domyślną wielkość czcionki w ustawieniach przeglądarki,

wartość właściwości line-height najlepiej określić w jednostkach em, a nie w pikselach, gdyż dzięki temu odległość między wierszami będzie ustalona względem wybranej przez użytkownika wielkości czcionki.

ODLEGŁOŚCI POMIĘDZY LITERAMI I SŁOWAMI

letter-spacing, word-spacing

CSS

rozdział-12/letter-and-word-spacing.html

```
h1, h2 {  
    text-transform: uppercase;  
    letter-spacing: 0.2em;}  
.credits {  
    font-weight: bold;  
    word-spacing: 1em;}
```

WYNIK

BRIARDY

napisała Ivy Duckett

[Briard](#), inaczej owczarek francuski briard, jest dużym psem pasterskim tradycyjnie używanym do wypasania i pilnowania owiec.

Historia rasy

Uważa się, że psy te zostały wyhodowane we Francji. Od wieków były używane do wypasania i ochrony owiec. W armii francuskiej służyły jako psy wartownicze, do przenoszenia wiadomości oraz, ze względu na doskonały słuch, do poszukiwania rannych. W czasie pierwszej wojny światowej ich umiejętności wykorzystywano w tak dużym stopniu, że niemal doprowadzono do wyginięcia rasy. Aktualnie ich populacja cały czas z wolna rośnie. Charlemagne, Napoleon, Thomas Jefferson oraz Lafayette mieli psy te rasy.

Kerning jest terminem typograficznym oznaczającym odległości pomiędzy poszczególnymi znakami tekstu. Odległości te można kontrolować przy użyciu właściwości `letter-spacing`.

Powiększenie odległości pomiędzy literami jest szczególnie przydatne wtedy, gdy nagłówek jest zapisany wyłącznie wielkimi literami. W przypadku zwyczajnych zdań (lub użycia małych liter) powiększanie lub zmniejszanie kerningu może utrudnić lekturę tekstu.

Można także kontrolować odległości pomiędzy kolejnymi wyrazami; służy do tego właściwość `word-spacing`.

W razie określania wartości tych właściwości należy się postępować jednostkami em, a podane wartości zostaną dodane do domyślnych odległości ustawionych w używanej czcionce.

Domyślne odległości są określone przez krój pisma (zwykle wynoszą one 0.25 em) i jest raczej mało prawdopodobne, by trzeba było je często zmieniać. W razie stosowania czcionki pogrubionej lub zwiększenia odległości pomiędzy znakami powiększenie odległości między wyrazami może poprawić przejrzystość tekstu.

WYNIK BEZ UŻYCIA CSS

Briardy

napisała Ivy Duckett

[Briard](#), inaczej owczarek francuski briard, jest dużym psem pasterskim tradycyjnie używanym do wypasania i pilnowania owiec.

Historia rasy

Uważa się, że psy te zostały wyhodowane we Francji. Od wieków były używane do wypasania i ochrony owiec. W armii francuskiej służyły jako psy wartownicze, do przenoszenia wiadomości oraz, ze względu na doskonały słuch, do poszukiwania rannych. W czasie pierwszej wojny światowej ich umiejętności wykorzystywano w tak dużym stopniu, że niemal doprowadzono do wyginięcia rasy. Aktualnie ich populacja cały czas z wolna rośnie. Charlemagne, Napoleon, Thomas Jefferson oraz Lafayette mieli psy te rasy.

WYRÓWNANIE

text-align

Właściwość `text-align` pozwala kontrolować wyrównanie tekstu. Może ona przyjmować jedną z czterech wartości:

left

Użycie tej wartości sprawi, że tekst będzie wyrównany do lewej.

right

Użycie tej wartości sprawi, że tekst będzie wyrównany do prawej.

center

Ta wartość umożliwia wyśrodkowanie tekstu.

justify

W przypadku użycia tej wartości wszystkie wiersze akapitu, z wyjątkiem ostatniego, powinny zajmować całą szerokość elementu, wewnętrz którego akapit został umieszczony.

rozdział-12/text-align.html

CSS

```
h1 {  
    text-align: left;}  
p {  
    text-align: justify;}  
.credits {  
    text-align: right;}
```

Briardy

napisała Ivy Ducke

[Briard](#), inaczej owczarek francuski briard, jest dużym psem pasterskim tradycyjnie używanym do wypasania i pilnowania owiec.

Historia rasy

Uważa się, że psy te zostały wyhodowane we Francji. Od wieków były używane do wypasania ochrony owiec. W armii francuskiej służyły jako psy wartownicze, do przenoszenia wiadomości oraz ze względu na doskonały słuch, do poszukiwania rannych. W czasie pierwszej wojny światowej ich umiejętności wykorzystywano w tak dużym stopniu, że niemal doprowadzono do wyginięcia ras. Aktualnie ich populacja cały czas z wolna rośnie. Charlemagne, Napoleon, Thomas Jefferson oraz Lafayette mieli psy tej rasy.

Uważa się, że jeśli na stronie jest umieszczonych kilka akapitów tekstu, łatwiej będzie można je odczytać, jeśli zostaną wyrównane do lewej.

W przypadku wyjustowania każdy wiersz tekstu jest analizowany z osobna, a odległości pomiędzy

poszczególnymi wyrazami zostają wyrównane. Tekst, w którym odległości pomiędzy pewnymi wyrazami będą duże, a pomiędzy innymi małe, może wyglądać dziwnie. Taki efekt często powstaje, kiedy wiersze tekstu nie są zbyt szerokie lub gdy tekst zawiera długie wyrazy.

WYRÓWNANIE W PIONIE

vertical-align

CSS

rozdział-12/vertical-align.html

```
#six-months {  
    vertical-align: text-top;  
}  
#one-year {  
    vertical-align: baseline;  
}  
#two-years {  
    vertical-align: text-bottom;  
}
```

WYNIK

Etapy życia briardów



Sześć miesięcy



Rok



Dwa lata

Właściwość `vertical-align` jest częstym powodem pomyłek. **Nie** służy ona bowiem do wyrównywania tekstu w pionie wewnątrz elementu blokowego, takiego jak `<p>` lub `<div>`, choć zapewnia taki efekt, jeśli zostanie zastosowana w komórce tabeli (elementach `<td>` lub `<th>`).

Znacznie częściej jest używana w elementach wewnętrzierszowych, takich jak ``, `` oraz ``. Wówczas działa w sposób bardzo podobny do atrybutu `align` elementu ``, opisanego na stronach 102 – 105. Poniżej wymieniono wartości, które może przyjmować ta właściwość:

`baseline`
`sub`
`super`
`top`
`text-top`
`middle`
`bottom`
`text-bottom`

Można jej także przypisać wartość określającą **długość** (zazwyczaj wyrażoną w pikselach lub jednostkach `em`) oraz **procentową** wartość wysokości wiersza.

WCIĘCIA

text-indent

Właściwość `text-indent` pozwala na utworzenie wcięcia w pierwszym wierszu tekstu umieszczonego w elemencie. Wielkość tego wcięcia można określić na kilka różnych sposobów, jednak zazwyczaj jest ona podawana w pikselach lub jednostkach `em`.

Wcięcie może też mieć wartość ujemną, co oznacza, że można go użyć w celu wysunięcia tekstu poza okno przeglądarki. Technika ta została zastosowana w przedstawionym obok przykładzie, w którym element `<h1>` używa obrazu, by przedstawić nagłówek w sposób graficzny. Tekst został natomiast przesunięty w lewo tak daleko, że znalazł się poza oknem przeglądarki. (Zagadnienia związane z obrazami tła zostały opisane na stronach 412 – 417).

Sam tekst nagłówka wciąż powinien się znajdować na stronie (ze względu na wyszukiwarki oraz osoby, które nie mogą oglądać obrazów), jednak nie chcemy, by przestonił logo, gdyż całość nagłówka stała się wtedy nieczytelna. Dzięki przesunięciu tekstu nagłówka o 9999 pikseli w lewo znalazł się on daleko poza obszarem strony wyświetlanym w przeglądarce, lecz mimo to wciąż znajduje się w kodzie HTML strony.

rozdział-12/text-indent.html

CSS

```
h1 {  
background-image: url("images/logo.gif");  
background-repeat: no-repeat;  
text-indent: -9999px;}  
.credits {  
text-indent: 20px;}
```

WYNIK

Briardy

napisała Ivy Duckett

[Briard](#), inaczej owczarek francuski briard, jest dużym psem pasterskim tradycyjnie używanym do wypasania i pilnowania owiec.

Historia rasy

Uważa się, że psy te zostały wyhodowane we Francji. Od wieków były używane do wypasania i ochrony owiec. W armii francuskiej służyły jako psy wartownicze, do przenoszenia wiadomości oraz do względu na doskonały słuch, do poszukiwania rannych. W czasie pierwszej wojny światowej ich umiejętności wykorzystywano w tak dużym stopniu, że niemal doprowadzono do wyginięcia rasy. Aktualnie ich populacja cały czas z wolna rośnie. Charlemagne, Napoleon, Thomas Jefferson oraz Lafayette mieli psy tej rasy.

Druga reguła zastosowana w przedstawionym tu przykładzie tworzy wcięcie o szerokości 20 pikseli w elemencie, którego atrybut `class` ma wartość `credits`.

CSS3: CIENIE

text-shadow

CSS

rozdział-12/text-shadow.html

```
p.one {  
background-color: #eeeeee;  
color: #666666;  
text-shadow: 1px 1px 0px #000000;}  
p.two {  
background-color: #dddddd;  
color: #666666;  
text-shadow: 1px 1px 3px #666666;}  
p.three {  
background-color: #cccccc;  
color: #ffffff;  
text-shadow: 2px 2px 7px #111111;}  
p.four {  
background-color: #bbbbbb;  
color: #cccccc;  
text-shadow: -1px -2px #666666;}  
p.five {  
background-color: #aaaaaa;  
color: #ffffff;  
text-shadow: -1px -1px #666666;}
```

WYNIK

Briandy określa się także jako serce otulone futrem.

Właściwość `text-shadow` jest powszechnie używana, choć jeszcze nie wszystkie przeglądarki ją obsługują.

Używa się jej do tworzenia cienia — czyli ciemnej wersji tekstu umieszczonej poniżej niego i nieco przesuniętej. Można jej także użyć do stworzenia efektu wypukłości — w tym przypadku cień musi być nieco jaśniejszy od tekstu.

Wartość tej właściwości jest nieco skomplikowana, gdyż może się składać z aż trzech liczb reprezentujących długości oraz z koloru.

Pierwsza liczba określa wielkość przesunięcia cienia w lewo lub prawo.

Druga liczba określa wielkość przesunięcia cienia w górę lub w dół.

Trzecia wartość jest opcjonalna; określa stopień rozmazania cienia.

Wreszcie czwarta wartość określa kolor cienia.

Właściwość `text-shadow` jest bardzo popularna, jednak gdy powstawała ta książka, nie była obsługiwana przez żadną wersję przeglądarki Internet Explorer (nawet najnowszą — IE9). Inni twórcy przeglądarek zapewnili jej obsługę w wersjach: Firefox 3.1, Safari 3, Chrome 2, Opera 9.5.

PIERWSZA LITERA LUB PIERWSZY WIERSZ

:first-letter, :first-line

Istnieje możliwość określenia dodatkowych reguł stylów dla pierwszego znaku oraz pierwszego wiersza tekstu w elemencie. Służą do tego odpowiednio pseudoelementy :first-letter i :first-line.

Warto zwrócić uwagę, że z technicznego punktu widzenia nie są to właściwości, lecz **pseudoelementy**.

Pseudoelementy są podawane na samym końcu selektora, a za nimi, jak we wszystkich regułach, umieszcza się deklarację stylu.

Warto wypróbować ten przykład w swojej przeglądarce, by się przekonać, że pseudoelement :first-line będzie miał wpływ wyłącznie na pierwszy wiersz tekstu, nawet jeśli zmienimy wielkość okna przeglądarki, a w efekcie zmieni się liczba słów widocznych w wierszu.

rozdział-12/first-letter-and-line.html

CSS

```
p.intro:first-letter {  
  font-size: 200%;}  
p.intro:first-line {  
  font-weight: bold;}
```

WYNIK

Briandy

napisała Ivy Duckett

Briard, inaczej owczarek francuski briard, jest dużym psem pasterskim tradycyjnie używanym do wypasania i pilnowania owiec.

Historia rasy

Uważa się, że psy te zostały wyhodowane we Francji. Od wieków były używane do wypasania i ochrony owiec. W armii francuskiej służyły jako psy wartownicze, do przenoszenia wiadomości oraz, ze względu na doskonały słuch, do poszukiwania rannych. W czasie pierwszej wojny światowej ich umiejętności wykorzystywano w tak dużym stopniu, że niemal doprowadzono do wyginięcia rasy. Aktualnie ich populacja cały czas z wolna rośnie. Charlemagne, Napoleon, Thomas Jefferson oraz Lafayette mieli psy tej rasy.

W CSS dostępne są zarówno pseudoelementy, jak i pseudoklasy. Pseudoelement działa tak, jakby w kodzie strony znajdował się dodatkowy element. Przedstawiane tu pseudoelementy :first-letter i :first-line działają w taki sposób, jak gdyby pierwsza litera lub pierwszy wiersz tekstu w elemencie były umieszczone w dodatkowym elemencie, dla którego możemy określić style.

Pseudoklasy działają, jak gdyby były dodatkowymi wartościami atrybutu class, np. pseudoklasa :visited (przedstawiona na następnej stronie) pozwala na określenie osobnych stylów dla łącz, których strona docelowa została już odwiedzona. Analogicznie pseudoklasa :hover pozwala określić odrębne style, które zostaną zastosowane po wskazaniu elementu myszką.

OKREŚLANIE POSTACI ŁĄCZY :link, :visited

CSS

rozdział-12/link-visited.html

```
a:link {  
    color: deeppink;  
    text-decoration: none;}  
a:visited {  
    color: black;}  
a:hover {  
    color: deeppink;  
    text-decoration: underline;}  
a:active {  
    color: darkcyan;}
```

WYNIK

Rasy psów: B

- Basset
- Beagle
- Bearded collie
- Owczarek francuski beauceron
- Bedlington terier
- Owczarek z Bergamo
- Bloodhound
- Bolończyk
- Border collie
- Border terier
- Borzoy
- Bouvier des Flandres
- Owczarek francuski briard
- Bulterier
- Bulldog angielski

Przeglądarki zazwyczaj wyświetlają łącza w kolorze niebieskim, umieszczając pod tekstem podkreślenie, a dodatkowo odwiedzone łącza są prezentowane w innym kolorze, by pomóc użytkownikom określić, które strony już odwiedzili, a których jeszcze nie. W CSS dostępne są dwie **pseudoklasy** pozwalające na określanie odrębnych reguł dla łączy nieodwiedzonych oraz odwiedzonych.

:link

Ta pseudoklasa pozwala określić style odnoszące się do łączy, które jeszcze nie były odwiedzone.

:visited

Ta pseudoklasa pozwala określić style dla łączy, które zostały już odwiedzone. Obie te pseudoklasy są powszechnie używane, by kontrolować kolor, w jakim są wyświetlane łącza, oraz to, czy są one podkreślone, czy nie.

W przykładzie przedstawionym z lewej strony widać, że odwiedzone łącza są wyświetlane innym kolorem, co pomaga użytkownikom zorientować się, które strony już oglądali.

Pseudoklasy :hover oraz :active (opisana na następnej stronie) są używane do zmiany postaci łączy, które użytkownik wskazuje myszką lub kliką.

RĘAKCJA NA DZIAŁANIA UŻYTKOWNIKA

:hover, :active, :focus

Dostępne są trzy pseudoklasy pozwalające na modyfikację wyglądu elementów w odpowiedzi na poczynania użytkownika.

:hover

Ta pseudoklasa jest stosowana, gdy użytkownik wskaże element przy użyciu jakiegoś urządzenia wskazującego, takiego jak mysz. Powszechnie była używana do zmiany wyglądu łączyc lub przycisków po umieszczeniu na nich wskaźnika myszy. Warto jednak zauważyć, że takie zdarzenia nie występują na urządzeniach wyposażonych w ekran dotykowe (takich jak iPad), gdyż urządzenia te nie są w stanie określić, kiedy użytkownik trzyma palec nad danym elementem.

:active

Ta pseudoklasa jest stosowana, kiedy element zostanie uaktywniony; np. gdy użytkownik naciśnie przycisk lub kliknie łącze. Czasami używa się jej, by stworzyć efekt, w którym przycisk lub łącze będą wyglądać, jak gdyby zostały naciśnięte. W tym celu wystarczy nieznacznie zmienić styl elementu i nieco go przesunąć.

:focus

Ta pseudoklasa jest stosowana, gdy w elemencie znajduje się

rozdział-12/hover-active-focus.html

CSS

```
input {  
  padding: 6px 12px 6px 12px;  
  border: 1px solid #665544;  
  color: #ffffff;}  
input.submit:hover {  
  background-color: #665544;}  
input.submit:active {  
  background-color: chocolate;}  
input.text {  
  color: #cccccc;}  
input.text:focus {  
  color: #665544;}
```

WYNIK

Wpisz adres e-mail...

Wyślij zapytanie

ognisko wprowadzania; dotyczy to wszystkich elementów, z którymi użytkownik może być w interakcji, takich jak pola formularzy.

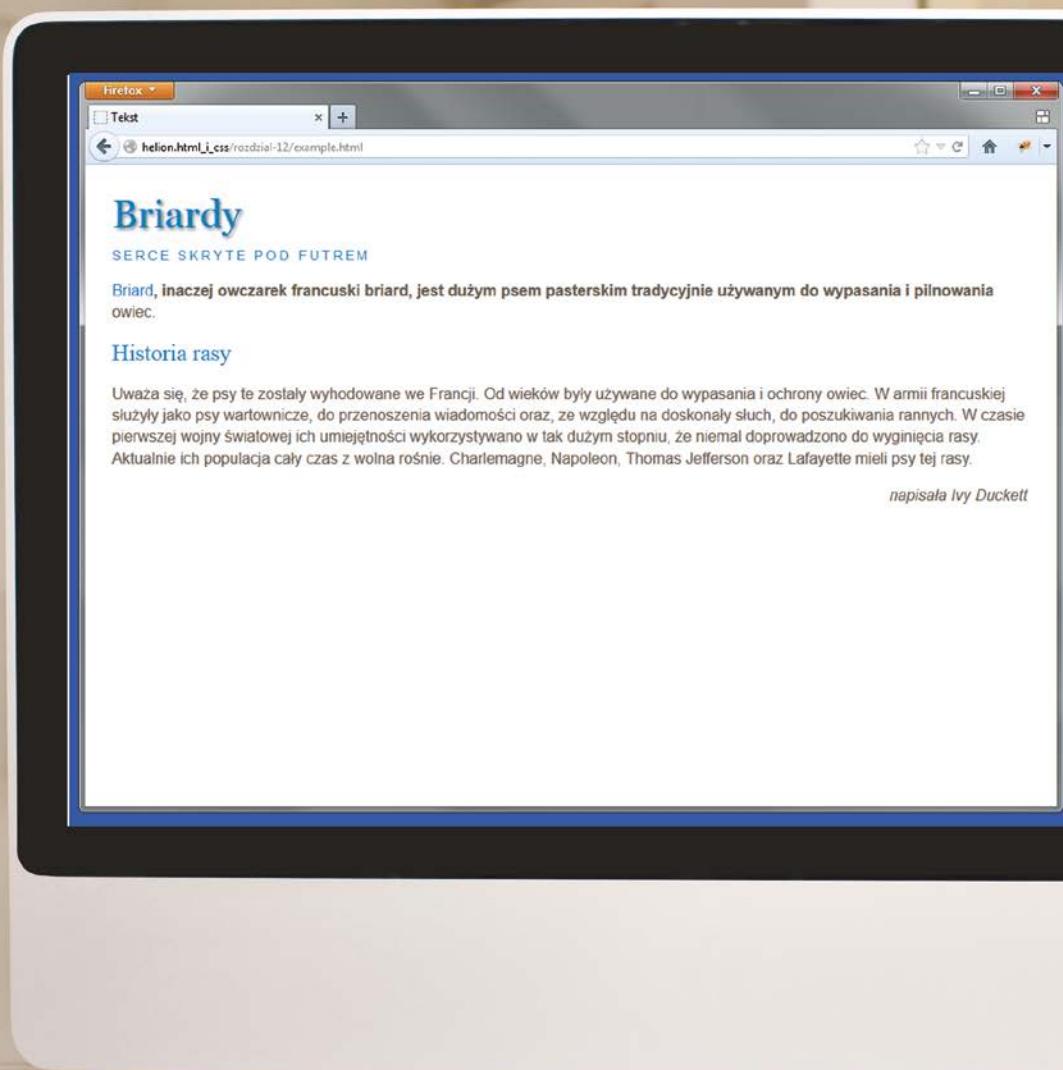
Przeniesienie ogniska wprowadzania do elementu następuje w momencie, gdy przeglądarka odkryje, że użytkownik jest gotowy do rozpoczęcia interakcji z danym elementem. Kiedy np. kursor znajduje się w polu

tekstowym i można w nim wpisywać znaki, mówimy, że w polu znajduje się ognisko wprowadzania. Oprócz tego podczas interakcji ze stroną można przechodzić pomiędzy kolejnymi polami, używając klawisza tabulacji oraz klawiszy strzałek. W razie stosowania tych pseudoklas należy je określić w następującej kolejności: :link, :visited, :hover, :focus, :active.

SELEKTORY ATRYBUTÓW

Najczęściej stosowane selektory CSS3 zostały przedstawione na stronie 237. Istnieje także grupa selektorów atrybutów, pozwalających na tworzenie reguł, które zawierają atrybuty o określonej wartości.

SELEKTOR	ZNACZENIE	PRZYKŁAD
ISTNIENIE	[] Pozwala znaleźć konkretny atrybut (niezależnie od tego, jaką ma wartość)	p[class] Odnosi się do wszelkich elementów <p>, w których został podany atrybut class
RÓWNOŚĆ	[=] Pozwala znaleźć konkretny atrybut o określonej wartości	p[class="dog"] Odnosi się do wszelkich elementów <p>, których atrybut class ma wartość dog
ODSTĘP	[~-] Pozwala znaleźć konkretny atrybut, którego wartość znajduje się na liście słów oddzielonych od siebie przecinkami	p[class~="dog"] Odnosi się do wszelkich elementów <p> zawierających atrybut class, którego wartość znajduje się na liście podanych wartości (w naszym przypadku lista ta zawiera tylko jedną wartość — dog)
POCZĄTKOWY FRAGMENT	[^=] Pozwala znaleźć konkretny atrybut, którego wartość rozpoczyna się podanym łańcuchem znaków	p[attr^="dog"] Odnosi się do wszelkich elementów <p>, w których wartość atrybutu attr zaczyna się od litery „d”.
FRAGMENT ŁAŃCUCHA	[*=] Pozwala znaleźć konkretny atrybut, którego wartość zawiera określony łańcuch znaków	p[attr*="do"] Odnosi się do wszelkich elementów, w których wartość atrybutu attr zawiera ciąg znaków „do”
ZAKOŃCZENIE	[=\$] Pozwala znaleźć konkretny atrybut, którego wartość kończy się podanym łańcuchem znaków	p[attr\$="g"] Odnosi się do wszelkich elementów <p>, w których wartość atrybutu attr kończy się literą „g”



Briardy

SERCÉ SKRYTE POD FUTREM

Briard, inaczej owczarek francuski briard, jest dużym psem pasterskim tradycyjnie używanym do wypasania i pilnowania owiec.

Historia rasy

Uważa się, że psy te zostały wyhodowane we Francji. Od wieków były używane do wypasania i ochrony owiec. W armii francuskiej służyły jako psy wartownicze, do przenoszenia wiadomości oraz, ze względu na doskonały słuch, do poszukiwania rannych. W czasie pierwszej wojny światowej ich umiejętności wykorzystywano w tak dużym stopniu, że niemal doprowadzono do wyginięcia rasy. Aktualnie ich populacja cały czas z wolna rośnie. Charlemagne, Napoleon, Thomas Jefferson oraz Lafayette mieli psy tej rasy.

napisała Ivy Duckett

PRZYKŁAD

TEKST



W przedstawionym tu przykładzie wykorzystano wiele spośród technik zaprezentowanych w tym rozdziale.

Wielkości czcionek są określane z wykorzystaniem właściwości `font-size`. Przy użyciu właściwości `font-style` czcionka w nagłówkach została zmieniona z pogrubionej na normalną. Dodatkowo dzięki użyciu właściwości `font-family` zostały zmienione czcionki używane do prezentacji tekstu w elementach.

W elemencie `<h1>` została zastosowana właściwość CSS3 `text-shadow`, dzięki której poniżej elementu jest wyświetlany cień. Z kolei tekst w elemencie `<h2>` został wyświetlony wielkimi literami przy użyciu właściwości `text-transform`, dzięki czemu łatwiej można go odczytać; dodatkowo, korzystając z właściwości `letter-spacing`, powiększyliśmy odstępy pomiędzy literami.

Jeśli chodzi o główną, tekstową zawartość strony, powiększona została wartość właściwości `line-height`, dzięki czemu odstępy pomiędzy kolejnymi wierszami są większe, a tekst jest łatwiejszy do odczytania. Dzięki użyciu pseudoelementu `first-line` została stworzona reguła, która wyświetla pierwszy wiersz tekstu w akapicie pogrubioną czcionką. I w końcu informacje o autorce tekstu zostały wyświetlane kursywą i wyrównane do prawej.

PRZYKŁAD

TEKST

```
<!DOCTYPE html>
<html>
<head>
<title>Tekst</title>
<style type="text/css">
  body {
    padding: 20px;
  }
  h1, h2, h3, a {
    font-weight: normal;
    color: #0088dd;
    margin: 0px;
  }
  h1 {
    font-family: Georgia, Times, serif;
    font-size: 250%;
    text-shadow: 2px 2px 3px #666666;
    padding-bottom: 10px;
  }
  h2 {
    font-family: "Gill Sans", Arial, sans-serif;
    font-size: 90%;
    text-transform: uppercase;
    letter-spacing: 0.2em;
  }
  h3 {
    font-size: 150%;
  }
  p {
    font-family: Arial, Verdana, sans-serif;
    line-height: 1.4em;
    color: #665544;
  }
  p.intro:first-line {
    font-weight: bold;
  }
  .credits {
    font-style: italic;
    text-align: right;
  }
  a {
    text-decoration: none;
  }
  a:hover {
    text-decoration: underline;
  }
</style>
</head>
```

PRZYKŁAD

TEKST

```
<body>
  <h1>Briardy</h1>
  <h2>Serce skryte pod futrem</h2>
  <p class="intro"><a class="breed" href="http://pl.wikipedia.org/wiki/Briard">
    Briard</a>, inaczej owczarek francuski briard, jest dużym psem pasterskim
    tradycyjnie używanym do wypasania i pilnowania owiec.</p>
  <h3>Historia rasy</h3>
  <p>Uważa się, że psy te zostały wyhodowane we Francji. Od wieków były używane
    do wypasania i ochrony owiec. W armii francuskiej służyły jako psy
    wartownicze, do przenoszenia wiadomości oraz, ze względu na doskonały słuch,
    do poszukiwania rannych. W czasie pierwszej wojny światowej wykorzystywano
    ich umiejętności
    w tak dużym stopniu, że niemal doprowadzono do wyginięcia rasy.
    Aktualnie ich populacja cały czas z wolna rośnie. Charlemagne, Napoleon,
    Thomas Jefferson oraz Lafayette mieli psy tej rasy.</p>
  <p class="credits">napisała Ivy Duckett</p>
  </body>
</html>
```


PODSUMOWANIE

TEKST

- ▶ Język HTML udostępnia właściwości pozwalające na kontrolę używanych czcionek, ich wielkości, wagi, stylu oraz odstępów pomiędzy znakami i wyrazami.
- ▶ Liczba czcionek, co do których można przyjąć, że będą zainstalowane na komputerach użytkowników, jest ograniczona.
- ▶ Jeśli chcemy używać szerszego zakresu krojów pisma, możemy skorzystać z kilku rozwiązań, niemniej konieczne jest posiadanie odpowiednich licencji na czcionki.
- ▶ Można kontrolować odstępy pomiędzy kolejnymi wierszami tekstu, poszczególnymi znakami oraz słowami. Teksty można wyrównywać do lewej, do prawej, wyśrodkowywać oraz justować. Można także określić wcięcie.
- ▶ Korzystając z pseudoelementów, można zmieniać styl elementu, kiedy użytkownik umieści na nim wskaźnik myszy lub go kliknie, albo w odmienny sposób wyświetlać odwiedzone łącza.

13

PUDEŁKA ELEMENTÓW

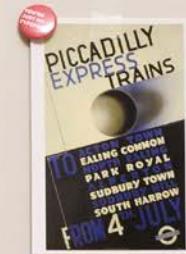
- ▶ Kontrola wielkości pudełka elementu.
- ▶ Model pudełkowy — obramowania, marginesy oraz wypełnienia.
- ▶ Wyświetlanie i ukrywanie pudełka.

Na początku części książki prezentującej CSS była mowa o tym, że każdy element HTML jest traktowany tak, jakby znajdował się w odrębnym pudełku.

Istnieje możliwość określania kilku właściwości, które mają wpływ na wygląd tych pudełek. W tym rozdziale powiemy, w jaki sposób można:

- Kontrolować wymiary pudełek elementów.
- Tworzyć obramowania wokół elementów.
- Określać marginesy i wypełnienia.
- Pokazywać i ukrywać pudełka elementów.

Kiedy już się dowiemy, jak kontrolować wygląd pudełek elementów, w kolejnym rozdziale (15.) dotyczącym układu stron przekonamy się, jak można je rozmieszczać.



WYMIARY PUDEŁKA

width, height

Domyślnie pudełko elementu jest tylko na tyle duże, by pomieścić całą jego zawartość. Aby określić jego wymiary, należy użyć właściwości `width` oraz `height`.

Najczęściej używanym sposobem określania wymiarów elementu jest podawanie wartości wyrażonych w pikselach, wartościach procentowych lub jednostkach em. Tradycyjnie już najpopularniejszym rozwiązaniem jest podawanie wymiarów w pikselach, gdyż zapewnia to projektantom największą precyzję.

W przypadku korzystania z wartości procentowych wielkość elementu jest wyliczana względem wymiarów okna przeglądarki lub, jeśli element znajduje się wewnątrz innego, względem elementu, w którym wymiarowany element się znajduje.

W przypadku korzystania z jednostek em wielkość elementu jest określana na podstawie wielkości umieszczonego w nim tekstu. Obecnie projektanci coraz częściej określają wymiary, korzystając z wartości procentowych lub wielkości wyrażonych w jednostkach em — starają się w ten sposób tworzyć strony o elastycznym układzie pozwalającym wyświetlać je na urządzeniach wyposażonych w ekranы o różnej rozdzielczości.

W przykładzie przedstawionym obok widoczny jest element `<div>` o szerokości i wysokości 300 pikseli. Wewnątrz niego został umieszczony akapit tekstu,

rozdział-13/width-height.html

HTML

```
<div>
  <p>Firma Moog była jedną z pierwszych, które we wczesnych latach 50. rozpoczęły komercyjną sprzedaż analogowych systemów syntezatorowych zasilanych prądem zmiennym.</p>
</div>
```

```
div.box {
  height: 300px;
  width: 300px;
  background-color: #bbbbbaa;}
p {
  height: 75%;
  width: 75%;
  background-color: #0088dd;}
```

CSS

Firma Moog była jedną z pierwszych, które we wczesnych latach 50. rozpoczęły komercyjną sprzedaż analogowych systemów syntezatorowych zasilanych prądem zmiennym.

WYNIK

którego szerokość i wysokość stanowią 75% wymiarów elementu zawierającego. Oznacza to, że szerokość i wysokość akapitu to 255 pikseli.

OGRANICZENIE SZEROKOŚCI

min-width, max-width

HTML

rozdział-13/min-width-max-width.html

```
<tr>
  <td></td>
  <td class="description">Fender Rhodes to
    elektryczno-mechaniczne pianino wymyślone przez
    Harolda Rhodesa w latach 50. XX wieku,
    a następnie produkowane w wielu modelach...</td>
  <td>$1400</td>
</tr>
```

CSS

```
td.description {
  min-width: 450px;
  max-width: 650px;
  text-align: left;
  padding: 5px;
  margin: 0px;}
```

WYNIK

Zdjęcie	Opis	Cena
	Fender Rhodes to elektryczno-mechaniczne pianino wymyślone przez Harolda Rhodesa w latach 50. XX wieku, a następnie produkowane w wielu modelach, najpierw we współpracy z firmą Fender, a po 1965 roku z CBS. Posiada klawisze przypominające pianino oraz młoteczki uderzające niewielkie metalowe preły, których dźwięki są wzmacniane przez przetworniki elektromagnetyczne.	\$1400
	Elektryczne pianino Wurlitzer jest elektromechanicznym instrumentem stworzonym przez firmę Rudolph Wurlitzer Company z Mississipi. W rzeczywistości firma ta nigdy nie nazwała swojego instrumentu pianinem elektrycznym, wymyśliła dla niego nazwę "pianino elektryczne" i używala jej następnie jako znaku handlowego swoich produktów. Instrumenty te posiadają klawiaturę podobną do pianina oraz młoteczki uderzające niewielkie metalowe preły, których dźwięki są wzmacniane przez przetworniki elektromagnetyczne.	\$1600
	Clavinet to klawikord, którego dźwięk jest wzmacniany elektronicznie, produkowany przez firmę Hohner. Każdy klawisz ma gumową końcówkę, która uderza strunę. Charakterystyczne zвуks słaćca tego instrumentu jest często porównywane do dźwięku gitary elektrycznej. Przez lata pojawiło się wiele modeli tego instrumentu, w tym I, II, L, C, D6 oraz E7.	\$1200

Niektóre projekty stron rozszerzają się lub zwężają, dostosowując swoją wielkość do wymiarów ekranu. W takich projektach właściwość `min-width` określa najmniejszą szerokość, jaką może przyjąć pudełko elementu wyświetcone na wąskim ekranie, a właściwość `max-width` — maksymalną szerokość, do jakiej pudełko elementu może się powiększyć, gdy ekran będzie szeroki.

Są to bardzo przydatne właściwości, które sprawią, że zawartość strony będzie należycie czytelna (dotyczy to zwłaszcza małych ekranów urządzeń kieszonkowych). Używając np. właściwości `max-width`, zapewniamy, że wiersze tekstu nie będą zbyt szerokie w dużym oknie przeglądarki, a korzystając z właściwości `min-width` — że na małym ekranie nie będą one zbyt wąskie.

Warto, by Czytelnik wypróbował ten przykład i przekonał się, co się będzie działo, gdy zacznie zmieniać szerokość okna przeglądarki.

Trzeba pamiętać, że właściwości te zaczęły być obsługiwane w przeglądarce IE7 oraz Firefox 2, a zatem w starszych wersjach tych programów nie będą działać.

OGRANICZANIE WYSOKOŚCI

min-height, max-height

Oprócz sytuacji, gdy będziemy chcieli ograniczać szerokość pudełka elementu, może się zdarzyć, że będziemy chcieli ograniczać jego wysokość. Można to zrobić, korzystając z dwóch właściwości: min-height i max-height.

Przykład przedstawiony obok demonstruje efekty zastosowania obu tych właściwości. Oprócz tego pokazuje, co się stanie, kiedy zawartość elementu będzie zbyt duża, by zmieścić się w jego obszarze.

Jeśli pudełko elementu jest zbyt małe, by pomieścić całą jego zawartość, a ta zostaje wyświetlona poza jego obszarem, to w efekcie na stronie może powstać spory bałagan. Do kontroli sposobu postępowania w takich sytuacjach można użyć właściwości overflow opisanej na następnej stronie.

rozdział-13/min-height-max-height.html

HTML

```
<h2>Fender Mustang</h2>
<p>Gitara Fender Mustang została przedstawiona
  w 1964 roku jako podstawa do...</p>
<h2>Fender Stratocaster</h2>
<p>Gitara Fender Stratocaster, nazywana także "Strat",
  jest jedną z najpopularniejszych gitar...</p>
<h2>Gibson Les Paul</h2>
<p>Gibson Les Paul to gitara elektryczna wykonana
  z litego kawałka drewna, która pojawiła się
  w sprzedaży w 1952 roku...</p>
```

```
h2, p {
  width: 400px;
  font-size: 90%;
  line-height: 1.2em;}
h2 {
  color: #0088dd;
  border-bottom: 1px solid #0088dd;}
p {
  min-height: 10px;
  max-height: 30px;}
```

CSS

Fender Mustang

Gitara Fender Mustang została przedstawiona w 1964 roku jako podstawa do znaczącej modyfikacji projektu instrumentów do nauki, którymi w tamtym czasie były Musicmaster oraz

Fender Stratocaster. Gitara ta była popularna w latach 60., a w późniejszych latach 90. zyskała status instrumentu kultowego, głównie za sprawą używania jej przez wiele zespołów rocka alternatywnego. Gitara Fender Stratocaster, nazywana także "Strat", jest jedną z najpopularniejszych gitar elektrycznych w historii.

Gibson Les Paul. Została zaprojektowana przez Leo Fendera, George'a Fullertona oraz Freddiego Tavaresa w 1954 roku. Gibson Les Paul to gitara elektryczna wykonana z litego kawałka drewna, która pojawiła się w sprzedaży w 1952 roku. Została zaprojektowana przez Teda McCarty'ego we współpracy z popularnym gitarzystą Les Paulem, którego Gibson nakoniął do jej reklamowania. Stanowi ona jeden z najbardziej znanych typów gitar elektrycznych na świecie.

WYNIK

OBSŁUGA ZBYT DUŻEJ ZAWARTOŚCI

overflow

HTML

rozdział-13/overflow.html

```
<h2>Fender Stratocaster</h2>
<p class="one">Gitara Fender Stratocaster, nazywana
także "Strat", jest jedną z najpopularniejszych...</p>
<h2>Gibson Les Paul</h2>
<p class="two">Gibson Les Paul to gitara elektryczna
wykonana z litego kawałka drewna, która pojawiła
się w sprzedaży w 1952 roku...</p>
```

CSS

```
p.one {
  overflow: hidden;}
p.two {
  overflow: scroll;}
```

WYNIK

Fender Stratocaster

Gitara Fender Stratocaster, nazywana także "Strat", jest jedną z najpopularniejszych gitar elektrycznych wszech czasów, a jej projekt został

Gibson Les Paul

Gibson Les Paul to gitara elektryczna wykonana z litego kawałka drewna, która pojawiła się w sprzedaży w

Właściwość overflow określa, co przeglądarka ma zrobić, jeśli zawartość elementu nie mieści się w obszarze jego pudełka. Może ona przyjąć jedną z dwóch wartości:

hidden

Użycie tej wartości powoduje ukrycie wszystkich treści, które nie mieszczą się w pudełku elementu.

scroll

W przypadku użycia tej wartości przeglądarka wyświetli wewnątrz elementu paski przewijania pozwalające wyświetlić początkowo niewidoczną zawartość.

W przykładzie przedstawionym z lewej strony można zobaczyć dwa elementy, których zawartość jest zbyt duża i nie mieści się wewnątrz ich pudełek. W pierwszym elemencie właściwość overflow została przypisana wartość hidden, natomiast w drugim — wartość scroll.

Właściwość overflow jest szczególnie przydatna, gdyż niektóre przeglądarki pozwalają użytkownikom dowolnie zmniejszać lub powiększać prezentowane teksty. Jeśli tekst będzie zbyt duży, to na stronie może zapanować bałagan i straci ona czytelność. Na takich stronach ukrycie treści niemieszczących się wewnątrz elementów może zapobiec wzajemnemu nakładaniu się treści.

OBRAMOWANIE, MARGINES ORAZ WYPEŁNIENIE

Każde pudełko elementu ma trzy właściwości, które można modyfikować w celu określenia jego wyglądu.

1

2

3

OBRAMOWANIE

Każde pudełko elementu ma obramowanie (nawet jeśli jest niewidoczne albo ma 0 pikseli szerokości). Obramowanie to oddziela brzeg jednego pudełka od drugiego.

Jeśli określmy wymiary elementu, to grubość jego obramowania oraz szerokość marginesu i wypełnienia zostaną dodane do tych wymiarów.

MARGINES

Margines jest obszarem znajdującym się na zewnątrz obramowania elementu. Można używać marginesu, by zwiększać odstęp pomiędzy obramowaniami dwóch sąsiadujących ze sobą elementów.

WYPEŁNIENIE

Wypełnienie to odstęp pomiędzy obramowaniem elementu a jego zawartością. Powiększanie wypełnienia poprawia czytelność zawartości elementu.



ODSTĘPY I MARGINES PIONOWY

Z MARGINESAMI I WYPEŁNIENIAMI

Moog

Syntezatory Mooga zostały stworzone przez dr. Roberta Mooga, założyciela firmy Moog Music. Do najpopularniejszych modeli tych syntezatorów należą Moog Modular, Minimoog, Micromoog, Moog Rogue oraz Moog Source.

ARP

Firma ARP Instruments Inc. została stworzona przez Alana Pearlmana i w latach 70. była głównym konkurencją Moog Music. Do jej najpopularniejszych produktów należą ARP 2600 oraz ARP Odyssey.

Sequential Circuits

Sequential Circuits Inc. została założona przez Dave'a Smitha i odegrała kluczową rolę w popularizacji MIDI. Najślynniejsze modele instrumentów tej firmy to Prophet 5, Prophet 600 oraz Pro-One.

BEZ MARGINESÓW I WYPEŁNIENI

Moog

Syntezatory Mooga zostały stworzone przez dr. Roberta Mooga, założyciela firmy Moog Music. Do najpopularniejszych modeli tych syntezatorów należą Moog Modular, Minimoog, Micromoog, Moog Rogue oraz Moog Source.

ARP

Firma ARP Instruments Inc. została stworzona przez Alana Pearlmana i w latach 70. była głównym konkurencją Moog Music. Do jej najpopularniejszych produktów należą ARP 2600 oraz ARP Odyssey.

Sequential Circuits

Sequential Circuits Inc. została założona przez Dave'a Smitha i odegrała kluczową rolę w popularizacji MIDI. Najślynniejsze modele instrumentów tej firmy to Prophet 5, Prophet 600 oraz Pro-One.

Właściwości **padding** oraz **margin** są bardzo pomocne, gdyż pozwalają zwiększać odstępy pomiędzy różnymi elementami strony.

Projektanci określają odległości pomiędzy poszczególnymi elementami strony jako **odstępy**. Wyobraźmy sobie, że wokół jakiegoś elementu jest wyświetlone obramowanie. W takim przypadku nie będziemy chcieli, by sąsiadujący z nim tekst dotykał tego obramowania, gdyż to utrudniłoby lekturę treści.

Albo wyobraźmy sobie, że obok siebie na stronie znajdują się dwa elementy (a każdy z nich ma widoczne czarne obramowanie). W takim przypadku też nie chcielibyśmy, by obramowania się stykały, gdyż krawędź styku wydawałaby się dwukrotnie grubszą.

Jeśli dolny margines elementu styka się z górnym marginesem innego elementu, to przeglądarka wyświetli je inaczej, niż można by się tego spodziewać. W takiej sytuacji zostanie wyświetlony jedynie większy z obu marginesów. Jeśli oba marginesy będą miały taką samą wielkość, to zostanie wyświetlony jeden z nich.

SZEROKOŚĆ OBRAMOWANIA border-width

Do określania szerokości obramowania elementu służy właściwość `border-width`. Jej wartość może być wyrażona w pikselach lub podana przy użyciu jednej z predefiniowanych wartości:

`thin`
`medium`
`thick`

(Do określania wartości tej właściwości nie można używać wartości procentowych).

Szerokości poszczególnych krawędzi obramowania można określić przy użyciu czterech właściwości:

`border-top-width`
`border-right-width`
`border-bottom-width`
`border-left-width`

Można także określić odrębne szerokości dla poszczególnych krawędzi obramowania, używając jednej właściwości:

`border-width: 2px 1px 1px 2px;`

Wartości dla poszczególnych krawędzi są podawane w kolejności zgodnej z ruchem wskazówek zegara i odnoszą się do następujących krawędzi: górnej, prawej, dolnej i lewej.

rozdział-13/border-width.html

HTML

```
<p class="one">"Clavinet" Hohnera jest właściwie elektrycznym klawikordem.</p>
<p class="two">"Clavinet" Hohnera jest właściwie elektrycznym klawikordem.</p>
<p class="three">"Clavinet" Hohnera jest właściwie elektrycznym klawikordem.</p>
```

```
p.one {
  border-width: 2px;
}
p.two {
  border-width: thick;
}
p.three {
  border-width: 1px 4px 12px 4px;
```

CSS

WYNIK

"Clavinet" Hohnera jest właściwie elektrycznym klawikordem.

"Clavinet" Hohnera jest właściwie elektrycznym klawikordem.

"Clavinet" Hohnera jest właściwie elektrycznym klawikordem.

STYL OBRAMOWANIA

border-style

HTML

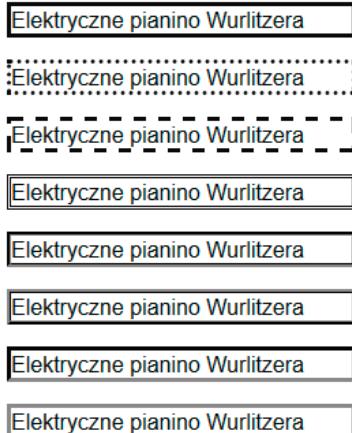
rozdzial-13/border-style.html

```
<p class="one">Elektryczne pianino Wurlitzera</p>
<p class="two">Elektryczne pianino Wurlitzera</p>
<p class="three">Elektryczne pianino Wurlitzera</p>
<p class="four">Elektryczne pianino Wurlitzera</p>
<p class="five">Elektryczne pianino Wurlitzera</p>
<p class="six">Elektryczne pianino Wurlitzera</p>
<p class="seven">Elektryczne pianino Wurlitzera</p>
<p class="eight">Elektryczne pianino Wurlitzera</p>
```

CSS

```
p.one {border-style: solid;}
p.two {border-style: dotted;}
p.three {border-style: dashed;}
p.four {border-style: double;}
p.five {border-style: groove;}
p.six {border-style: ridge;}
p.seven {border-style: inset;}
p.eight {border-style: outset;}
```

WYNIK



Styl obramowania można kontrolować przy użyciu właściwości `border-style`. Może ona przyjmować następujące wartości:

`solid` — pojedyncza linia ciągła;

`dotted` — seria kwadratowych punktów (jeśli obramowanie ma 2 piksele szerokości, to punkty także będą miały wymiary 2×2 piksele i będą oddalone od siebie o 2 piksele);

`dashed` — seria krótkich kresek;

`double` — podwójna linia ciągła (wartość właściwości `border-width` jest sumą grubości obu linii);

`groove` — obramowanie wydaje się wklęste;

`ridge` — obramowanie wydaje się wypukłe;

`inset` — sprawia wrażenie, jak gdyby element był zagębiony w stronie;

`outset` — sprawia wrażenie, jak gdyby element był wypukły;

`hidden/none` — obramowanie nie jest widoczne.

Style poszczególnych krawędzi obramowania można określić przy użyciu właściwości:

`border-top-style`
`border-left-style`
`border-right-style`
`border-bottom-style`

KOLOR OBRAMOWANIA

border-color

Kolor obramowania można określić, używając wartości RGB, kodów szesnastkowych lub predefiniowanych nazw kolorów (zgodnie z informacjami podanymi na stronach 250 – 251).

Istnieje możliwość określania odrębnych kolorów dla poszczególnych krawędzi obramowania; służą do tego następujące właściwości:

`border-top-color`
`border-right-color`
`border-bottom-color`
`border-left-color`

Można także skorzystać z zapisu skróconego, pozwalającego określić kolory wszystkich krawędzi obramowania w jednej właściwości:

`border-color: darkcyan
deeppink darkcyan deeppink;`

W tym przypadku poszczególne krawędzie są określone zgodnie z ruchem wskazówek zegara w następującej kolejności: górna, prawa, dolna i lewa.

Kolor obramowania można także określić przy użyciu wartości HSL (zgodnie z informacjami podanymi na stronach 254 – 255). Trzeba przy tym pamiętać, że zostały one wprowadzone w CSS3 i nie są obsługiwane w starszych przeglądarkach.

rozdział-13/border-color.html

HTML

```
<p class="one">Instrument ARP Odyssey został wprowadzony w 1972 roku.</p>
<p class="two">Instrument ARP Odyssey został wprowadzony w 1972 roku.</p>
```

```
p.one {
  border-color: #0088dd;
}
p.two {
  border-color: #bbbbbaa #111111 #ee3e80 #0088dd;
```

CSS

WYNIK

Instrument ARP Odyssey został wprowadzony w 1972 roku.

Instrument ARP Odyssey został wprowadzony w 1972 roku.

ZAPIS SKRÓTOWY

border

HTML

rozdział-13/border-shorthand.html

```
<p>Oto prosta sekwencja akordów granych na organach  
Hammonda i odtwarzanych przez głośniki Lesliego.</p>
```

Właściwość border pozwala określić szerokość, styl oraz kolor obramowania w jednym miejscu (przy czym wartości powinny być podawane właściwie w takiej kolejności).

CSS

```
p {  
width: 250px;  
border: 3px dotted #0088dd;}
```

WYNIK

Oto prosta sekwencja akordów
granych na organach Hammonda i
odtwarzanych przez głośniki
Lesliego.

WYPEŁNIENIE

padding

Właściwość padding pozwala określić wielkość obszaru pomiędzy zawartością elementu a jego obramowaniem.

Wartość tej właściwości jest zazwyczaj określana w pikselach (można ją także podawać jako wartość procentową lub w jednostkach em). W przypadku użycia wartości procentowej wielkość wypełnienia zostanie wyliczona względem wymiarów okna przeglądarki (lub względem elementu, wewnątrz którego znajduje się interesujący nas element).

Ważna uwaga: jeśli element, którego postać określamy, ma podaną szerokość, to wypełnienie zostanie do niej dodane.

Jak widać na przedstawionym obok przykładzie, drugi z akapitów jest znacznie bardziej czytelny, gdyż pomiędzy jego obramowaniem a tekstem dodaliśmy nieco pustego miejsca. Oprócz tego ze względu na wypełnienie drugi element jest szerszy od pierwszego.

Istnieje możliwość określania wypełnień dla poszczególnych krawędzi elementu:

`padding-top`
`padding-right`
`padding-bottom`
`padding-left`

Można także skorzystać z zapisu skrótowego (w którym wypełnienia są podawane w następującej kolejności: górne, prawe, dolne, lewe):
`padding: 10px 5px 3px 1px;`

rozdział-13/padding.html

HTML

```
<p>Syntezatory analogowe generują falę dźwiękową,  
natomiast dźwięki przechowywane w syntezatorach...</p>  
<p class="example">Syntezatory analogowe generują  
falę dźwiękową, natomiast dźwięki przechowywane...</p>
```

```
p {  
width: 275px;  
border: 2px solid #0088dd;}  
p.example {  
padding: 10px;}
```

CSS

WYNIK

Syntezatory analogowe generują falę dźwiękową, natomiast dźwięki przechowywane w syntezatorach cyfrowych zostały poddane próbkowaniu i przekształcone na liczby.

Syntezatory analogowe generują falę dźwiękową, natomiast dźwięki przechowywane w syntezatorach cyfrowych zostały poddane próbkowaniu i przekształcone na liczby.

Wartość właściwości padding nie jest dziedziczona przez elementy potomne, jak dzieje się np. z wartością właściwości color. Dlatego właściwość padding należy podawać osobno dla wszystkich elementów, które jej wymagają.

Jeszcze w przeglądarce Internet Explorer 6 szerokość elementu zawierała wypełnienie i marginesy. Więcej informacji na ten temat można znaleźć na stronie 315.

MARGINES

margin

HTML

rozdział-13/margin.html

```
<p>Często mawia się, że syntezatory analogowe  
mają "cieplejsze" brzmienie od swoich cyfrowych  
odpowiedników.</p>  
<p class="example">Często mawia się, że  
syntezatory analogowe mają "cieplejsze"  
brzmienie od swoich cyfrowych odpowiedników.</p>
```

CSS

```
p {  
    width: 200px;  
    border: 2px solid #0088dd;  
    padding: 10px;}  
p.example {  
    margin: 20px;}
```

WYNIK

Często mawia się, że
syntezatory analogowe
mają "cieplejsze" brzmienie
od swoich cyfrowych
odpowiedników.

Często mawia się, że
syntezatory analogowe
mają "cieplejsze" brzmienie
od swoich cyfrowych
odpowiedników.

Właściwość margin określa
wielkość odstępu pomiędzy
pułapkami elementów.

Jej wartość jest najczęściej
określana w pikselach, choć
można ją także podawać
w jednostkach em lub jako
wartość procentową.

Jeśli pułapko jednego elementu
znajduje się powyżej pułapka
innego, to oba marginesy
są zastępowane jednym —
większym z nich (mniejszy
margins jest pomijany).

Ważna uwaga: jeśli element,
którego postać określamy, ma
podaną szerokość, to margines
zostanie do niej dodany.

Istnieje możliwość określenia
wielkości marginesu osobno
dla każdej z czterech stron
elementu; służą do tego
następujące właściwości:

`margin-top`
`margin-right`
`margin-bottom`
`margin-left`

Można także skorzystać
z zapisu skrótnego (w którym
marginesy są podawane
w kolejności: górny, prawy,
dolny, lewy):

`margin: 1px 2px 3px 3px;`

Czasami można spotkać
następujący zapis:
`margin: 10px 20px;`

Oznacza on, że margins górnny
i dolny będzie mieć 10 pikseli
wysokości, a lewy i prawy —
20 pikseli szerokości.

(Tego samego skrótnego
zapisu przedstawionego
powyżej można używać
podczas określania wypełnienia).

Wartość właściwości margin nie jest dziedziczona przez elementy
potomne, jak dzieje się np. z wartością właściwości color. Dlatego tę
właściwość należy podawać osobno dla wszystkich elementów, które
jej wymagają.

W Internet Explorerze 6 szerokość elementu jeszcze zawierała
wypełnienie i marginesy. Więcej informacji na ten temat można
znać na stronie 315.

WYŚRODKOWYWANIE ZAWARTOŚCI

Jeśli chcemy wyśrodkować pudełko elementu na stronie (lub wewnątrz innego elementu, w którym jest on umieszczony), możemy to zrobić, przypisując właściwościom margin-left oraz margin-right wartości auto.

Aby wyśrodkować pudełko elementu na stronie, trzeba określić jego szerokość poprzez podanie wartości właściwości width (w przeciwnym razie element zajmie całą dostępną szerokość strony).

Jeśli szerokość elementu została podana, to przypisanie lewemu i prawemu marginowi wartości auto sprawi, że przeglądarka utworzy po obu stronach elementu marginesy o tej samej szerokości; a to spowoduje wyśrodkowanie elementu na stronie (lub wewnątrz innego elementu, w którym on się znajduje).

Aby rozwiązanie to działało w starszych przeglądarkach (a zwłaszcza w IE6), w regule stylów odnoszącej się do elementu, wewnątrz którego znajduje się wyśrodkowywany element, należy podać właściwość text-align o wartości center.

Właściwość text-align jest dziedziczona przez elementy potomne. Dlatego jeśli zawartość umieszczona wewnątrz wyśrodkowywanego elementu nie ma być wyśrodkowana, to także w nim należy odpowiednio określić wartość właściwości text-align.

rozdział-13/centering-content.html

HTML

```
<body>
<p>Często mawia się, że syntezatory analogowe
mają "cieplejsze" brzmienie od swoich cyfrowych
odpowiedników.</p>
<p class="example">Często mawia się, że
syntezatory analogowe mają "cieplejsze" brzmienie
od swoich cyfrowych odpowiedników.</p>
</body>
```

```
body {
  text-align: center;
}
p {
  width: 300px;
  padding: 50px;
  border: 20px solid #0088dd;
}
p.example {
  margin: 10px auto 10px auto;
  text-align: left;}
```

CSS

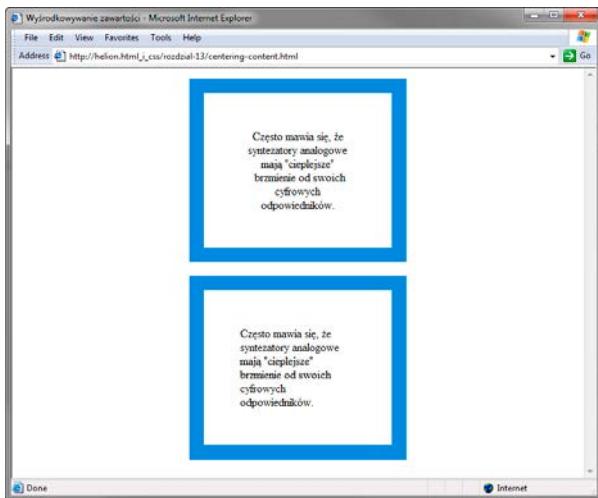
WYNIK

Często mawia się, że syntezatory analogowe
mają "cieplejsze" brzmienie od swoich
cyfrowych odpowiedników.

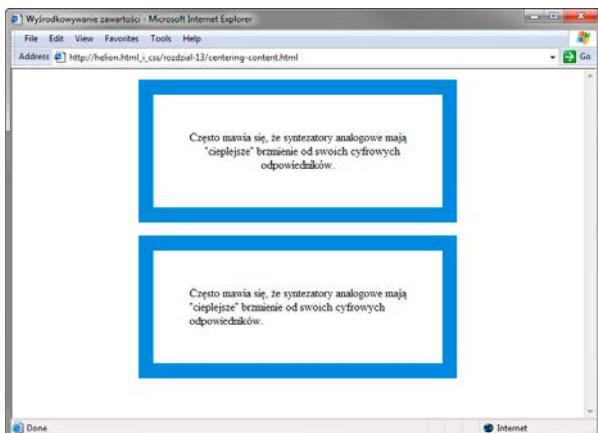
Często mawia się, że syntezatory analogowe
mają "cieplejsze" brzmienie od swoich
cyfrowych odpowiedników.

MODEL PUDEŁKOWY W PRZEGŁĄDARCE IE6

WYNIKI W IE6 BEZ PODAWANIA DEKLARACJI DOCTYPE



WYNIK W IE6 PO DODANIU DEKLARACJI DOCTYPE



W przypadku gdy określmy szerokość pudełka elementu, szerokości marginesów oraz wypełnienia powinny zostać do niej dodane. Jednak przeglądarka Internet Explorer 6 działa w dziwny sposób, który polega na tym, że marginesy oraz wypełnienia są uwzględniane w szerokości elementu.

Rozwiążaniem tego problemu jest umieszczenie w kodzie dokumentu HTML deklaracji DOCTYPE. (Deklaracje DOCTYPE zostały przedstawione na stronie 180). Aby wymusić w przeglądarce stosowanie właściwego modelu pudełkowego, można skorzystać z deklaracji DOCTYPE języka HTML5 albo języka HTML 4 w wersjach STRICT lub TRANSITIONAL.

Aby się przekonać, jak wygląda opisywane tu dziwne działanie przeglądarki IE6, wystarczy w niej wyświetlić przykład przedstawiony na poprzedniej stronie — najpierw bez umieszczania w kodzie strony deklaracji DOCTYPE, a następnie po dodaniu do niej deklaracji dla języka HTML5.

ZMIANY ELEMENTÓW WEWNĄTRZWIERSZOWYCH NA BLOKOWE I NA ODWRÓT

display

Właściwość `display` pozwala zmieniać elementy wewnętrzwerszowe na blokowe i na odwrót. Można jej także używać do ukrywania elementów stron.

Właściwość ta może przyjmować następujące wartości:

inline

Sprawi, że element blokowy zostanie potraktowany przez przeglądarkę jako element wewnętrzwerszowy.

block

Sprawi, że element wewnętrzwerszowy zostanie wyświetlony przez przeglądarkę jako element blokowy.

inline-block

Sprawi, że element blokowy zostanie rozmiieszczony na stronie w sposób charakterystyczny dla elementów wewnętrzwerszowych i zachowa przy tym pozostałe cechy charakterystyczne dla elementów blokowych.

none

Użycie tej wartości powoduje ukrycie elementu na stronie. Konkretnie: przeglądarka potraktuje element tak, jak gdyby w ogóle go nie było na stronie (choć oczywiście użytkownik może go zobaczyć po wyświetleniu kodu źródłowego).

rozdział-13/display.html

HTML

```
<ul>
  <li>Strona główna</li>
  <li>Produkty</li>
  <li class="coming-soon">Usługi</li>
  <li>O nas</li>
  <li>Kontakt</li>
</ul>
```

```
li {
  display: inline;
  margin-right: 10px;
}
li.coming-soon {
  display: none;
```

CSS

Strona główna Produkty O nas Kontakt

WYNIK

W przypadku zastosowania tej właściwości koniecznie trzeba pamiętać, że **nie** powinno się używać elementów wewnętrzwerszowych do tworzenia elementów blokowych.

Powyższy przykład przedstawia listę. Elementy list są zazwyczaj traktowane jako elementy blokowe, niemniej zastosowane reguły stylów oznaczają, że należy je traktować jako wewnętrzwerszowe. Oznacza to, że będą one umieszczone jeden obok drugiego, a nie każdy w nowym wierszu.

Przedstawiona tu technika jest często stosowana do tworzenia elementów nawigacyjnych witryn, a w naszym przykładzie określone zostały marginesy z prawej strony poszczególnych elementów, aby je od siebie oddzielić. Reguła odnosząca się do elementu `` z atrybutem `class` o wartości `coming-soon` sprawi, że będzie on niewidoczny — zupełnie tak, jakby go w ogóle nie było na stronie.

UKRYWANIE PUDEŁEK ELEMENTÓW

visibility

HTML

rozdział-13/visibility.html

```
<ul>
  <li>Strona główna</li>
  <li>Produkty</li>
  <li class="coming-soon">Usługi</li>
  <li>0 nas</li>
  <li>Kontakt</li>
</ul>
```

CSS

```
li {
  display: inline;
  margin-right: 10px;
}
li.coming-soon {
  visibility: hidden;
}
```

WYNIK

Strona główna Produkty

O nas Kontakt

Właściwość **visibility** pozwala ukrywać elementy, pozostawia jednak na stronie obszar, w którym się one znajdują.

Właściwość ta może przyjmować dwie wartości:

hidden

Użycie tej wartości powoduje ukrycie elementu.

visible

Użycie tej wartości sprawia, że element jest widoczny.

Jeśli w regule stylów odnoszącej się do elementu właściwości **visibility** zostanie przypisana wartość **hidden**, to na stronie będzie widoczny jedynie obszar, jaki element ma zajmować.

Jeśli nawet ten obszar ma nie być widoczny na stronie, to należy użyć właściwości **display** i przypisać jej wartość **none** (zgodnie z informacjami podanymi na poprzedniej stronie).

Trzeba pamiętać, że każdy może obejrzeć zawartość elementu, który został ukryty przy użyciu właściwości **visibility** z wartością **hidden** — wystarczy w tym celu wyświetlić w przeglądarce kod źródłowy strony.

CSS3: OBRAZY OBRAMOWAŃ border-image

Właściwość border-image pozwala wyświetlać obraz jako obramowanie pudełka elementu. Wykorzystuje ona wskazany obrazek i dzieli go na dziewięć części.

Oto obraz, który został wykorzystany w tym przykładzie. W książce dodano do niego linie przedstawiające sposób podziału — tworzą one kwadraty o boku długości 18 pikseli, w których znajdują się małe koła; koła te zostaną umieszczone w wierzchołkach elementu. Fragmenty zawierające wierzchołki obrazka zawsze są umieszczane w analogicznych wierzchołkach elementu, natomiast w przypadku fragmentów znajdujących się pośrodku krawędzi obrazu mamy możliwość określenia, czy mają być one rozciągnięte, czy należy je powtarzać.

Właściwość border-image wymaga podania trzech informacji:

1. adresu URL obrazu;
2. sposobu, w jaki należy podzielić obraz;
3. sposobu wyświetlania fragmentów znajdujących się przy prostych krawędziach, w tym przypadku można skorzystać z trzech wartości predefiniowanych: **stretch** (spowoduje rozciągnięcie fragmentu), **repeat** (sprawi, że fragment będzie powtarzany) oraz **round** (sprawi, że fragment będzie powtarzany; jeśli jednak nie

rozdział-13/border-image.html

HTML

```
<p class="one"></p>
<p class="two"></p>
<p class="three"></p>
```

```
p.one {
  -moz-border-image: url("images/dots.gif")
    11 11 11 11 stretch;
  -webkit-border-image: url("images/dots.gif")
    11 11 11 11 stretch;
  border-image: url("images/dots.gif")
    11 11 11 11 stretch;
}
p.two {
  -moz-border-image: url("images/dots.gif")
    11 11 11 11 round;
  -webkit-border-image: url("images/dots.gif")
    11 11 11 11 round;
  border-image: url("images/dots.gif")
    11 11 11 11 round;}
```

CSS



WYNIK

będzie dokładnie pasował do długości krawędzi elementu, to zostanie nieznacznie przeskalowany).

Aby obraz obramowania został wyświetlony, konieczne jest także określenie szerokości obramowania.

Właściwości **-moz-border-image** oraz **-webkit-border-image** nie należą do specyfikacji CSS, umożliwiają jednak wyświetlanie obramowania w starszych wersjach przeglądarek Chrome, Firefox i Safari.

CSS3: CIEŃ PUDEŁKA

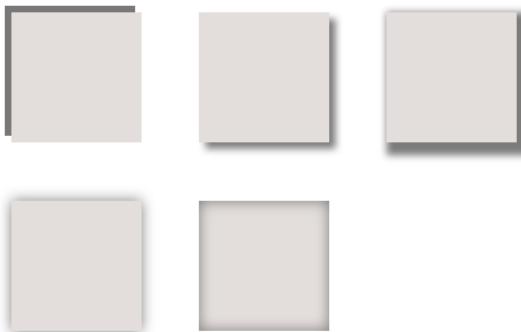
box-shadow

CSS

rozdział-13/box-shadow.html

```
p.one {  
  -moz-box-shadow: -5px -5px #777777;  
  -webkit-box-shadow: -5px -5px #777777;  
  box-shadow: -5px -5px #777777;}  
p.two {  
  -moz-box-shadow: 5px 5px 5px #777777;  
  -webkit-box-shadow: 5px 5px 5px #777777;  
  box-shadow: 5px 5px 5px #777777;}  
p.three {  
  -moz-box-shadow: 5px 5px 5px 5px #777777;  
  -webkit-box-shadow: 5px 5px 5px 5px #777777;  
  box-shadow: 5px 5px 5px 5px #777777;}  
p.four {  
  -moz-box-shadow: 0 0 10px #777777;  
  -webkit-box-shadow: 0 0 10px #777777;  
  box-shadow: 0 0 10px #777777;}  
p.five {  
  -moz-box-shadow: inset 0 0 10px #777777;  
  -webkit-box-shadow: inset 0 0 10px #777777;  
  box-shadow: inset 0 0 10px #777777;}
```

WYNIK



Przeglądarki Chrome, Firefox i Safari bardzo szybko zaczęły obsługiwać tę właściwość, udostępniając ją pod nazwami: -moz-box-shadow oraz -webkit-box-shadow. Choć nie należą one do standardu CSS3, ich stosowanie ułatwia tworzenie cienia elementu w tych przeglądarkach.

Właściwość box-shadow pozwala wyświetlić cień wokół pudełka elementu. Działa ona bardzo podobnie do właściwości text-shadow przedstawionej na stronie 287. Oprócz koloru należy w niej podać przynajmniej jedną z dwóch pierwszych wartości przedstawionych poniżej:

PRZESUNIĘCIE W POZIOMIE

Wartość ujemna sprawi, że cień będzie umieszczony z lewej strony pudełka.

PRZESUNIĘCIE W PIONIE

Wartość ujemna sprawi, że cień będzie umieszczony powyżej pudełka.

WIELKOŚĆ ROZMAZANIA

Jeśli ta wartość zostanie pominięta, cień będzie jednolity, podobnie jak obramowanie elementu.

ROZPIĘTOŚĆ CIENIA

Jeśli ta wartość zostanie podana i będzie dodatnia, cień będzie się rozchodzić we wszystkich kierunkach na zewnątrz elementu; w razie podania wartości ujemnej zostanie on zmniejszony.

Przed tymi wszystkimi wartościami można umieścić słowo `inset`, co sprawi, że cień zostanie wyświetlony wewnątrz elementu.

CSS3: ZAOKRĄGLENE WIERZCHOŁKI

border-radius

CSS3 wprowadza możliwość tworzenia zaokrąglonych wierzchołków pudełek dowolnych elementów. Służy do tego właściwość o nazwie border-radius. Jej wartość określa długość promienia zaokrąglonego wierzchołka i jest podawana w pikselach.

Właściwość ta nie jest obsługiwana przez starsze przeglądarki, w których wierzchołki obramowania elementu zawsze tworzą kąt prosty.

Właściwości -moz-border-radius oraz -webkit-border-radius nie należą do specyfikacji CSS. Niemniej jednak są one używane w niektórych wcześniejszych wersjach przeglądarek: Chrome, Firefox oraz Webkit i umożliwiają w nich tworzenie zaokrąglonych wierzchołków (dlatego można ich używać, by uzyskać ten efekt w większej liczbie przeglądarek).

Odrębne wartości zaokrąglenia dla poszczególnych wierzchołków można podawać, używając następujących właściwości:

border-top-right-radius
border-bottom-right-radius
border-bottom-left-radius
border-top-left-radius

rozdział-13/border-radius.html

HTML

```
<p>Zespół Pet Sounds wykorzystał wiele niekonwencjonalnych instrumentów, takich jak: dzwonki rowerowe, brzęczące organiki, klawesyny, flety, gwizdki na psy, pociągi, puszki po Coca-Cola, instrumenty strunowe o hawajskich dźwiękach, a nawet szczerkające psy.</p>
```

```
p {  
border: 5px solid #cccccc;  
padding: 20px;  
width: 275px;  
text-align: center;  
border-radius: 10px;  
-moz-border-radius: 10px;  
-webkit-border-radius: 10px;}
```

CSS

Zespół Pet Sounds wykorzystał wiele niekonwencjonalnych instrumentów, takich jak: dzwonki rowerowe, brzęczące organiki, klawesyny, flety, gwizdki na psy, pociągi, puszki po Coca-Cola, instrumenty strunowe o hawajskich dźwiękach, a nawet szczerkające psy.

WYNIK

Dostępny jest także sposób skrótonego zapisu powyższych czterech właściwości (przy czym wierzchołki są podawane zgodnie z ruchem wskazówek zegara w kolejności:

górny-prawy, dolny-prawy, dolny-lewy, górny-lewy). Oto przykład:
border-radius: 5px, 10px, 5px, 10px;

CSS3: KSZTAŁTY ELIPTYCZNE

border-radius

HTML

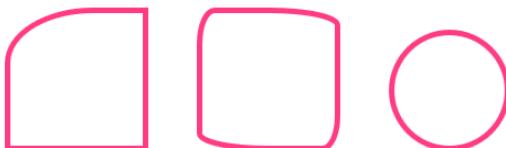
rozdział-13/elliptical-shapes.html

```
<p class="one"></p>
<p class="two"></p>
<p class="three"></p>
```

CSS

```
p.one {
  border-top-left-radius: 80px 50px;
  -moz-border-radius-top-left: 80px 50px;
  -webkit-border-radius-top-left: 80px 50px;}
p.two {
  border-radius: 1em 4em 1em 4em / 2em 1em 2em 1em;
  -moz-border-radius: 1em 4em 1em 4em
    / 2em 1em 2em 1em;
  -webkit-border-radius: 1em 4em 1em 4em
    / 2em 1em 2em 1em;}
p.three {
  padding: 0px;
  border-radius: 100px;
  -moz-border-radius: 100px;
  -webkit-border-radius: 100px;}
```

WYNIK



W powyższy sposób można narysować nawet okrąg — wystarczy utworzyć kwadratowy element i przypisać właściwości border-radius wartość odpowiadającą długości jego krawędzi (patrz trzeci przykład przedstawiony powyżej).

Aby tworzyć bardziej złożone kształty, można określić różne długości dla poziomej oraz pionowej osi zaokrąglonego wierzchołka elementu.

Poniższy przykład pokazuje, jak stworzyć zaokrąglony wierzchołek, którego szerokość jest większa od wysokości. border-radius: 80px 50px;

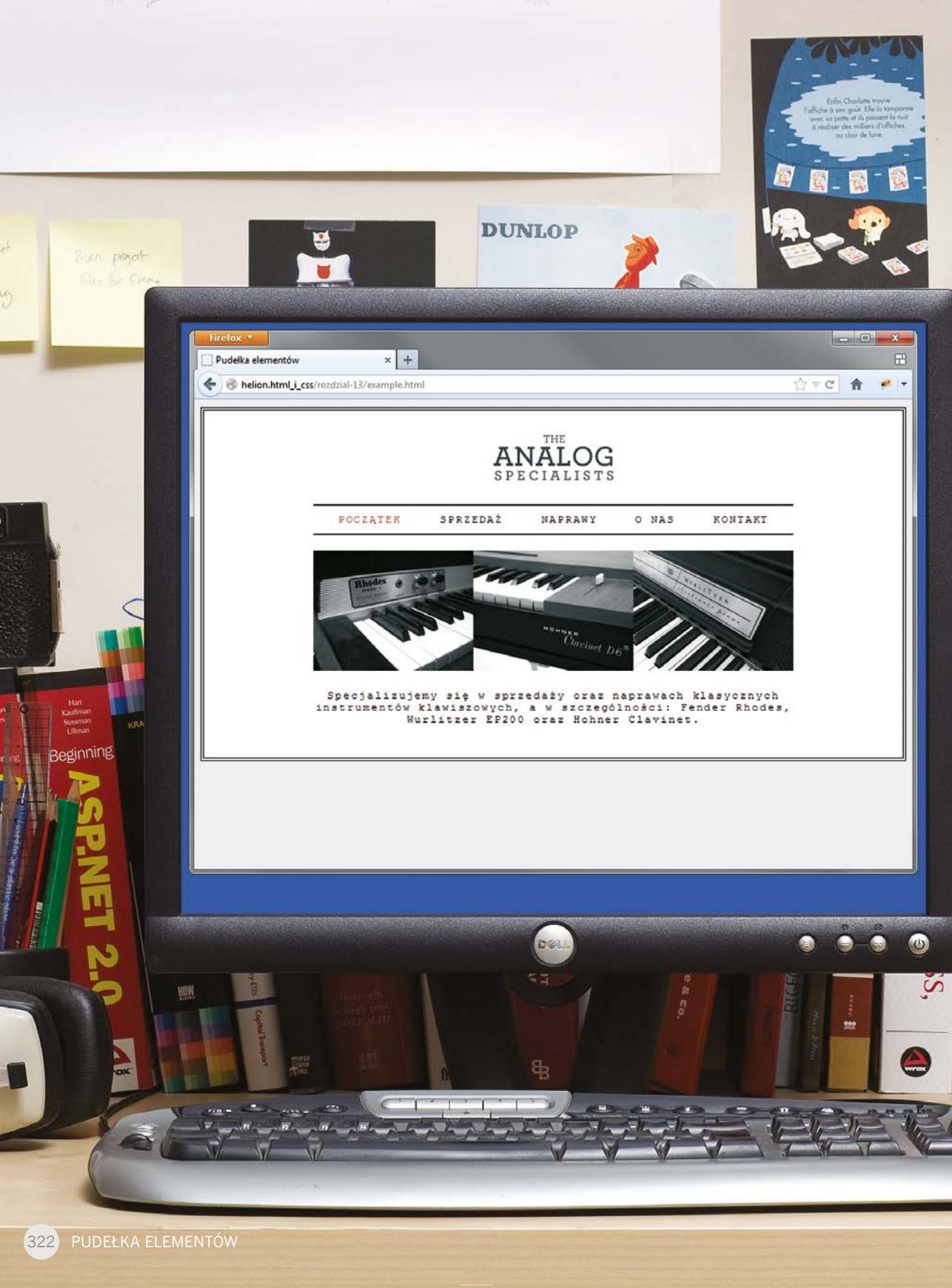


Można także określić parametry jednego, wybranego wierzchołka; oto przykład:

border-top-left-radius: 80px 50px;

Dostępny jest także zapis skrótny, pozwalający określić odrebine parametry dla każdego z czterech wierzchołków; w tym przypadku najpierw podawane są cztery wartości określające szerokość każdego z nich, a następnie cztery wartości określające ich wysokość.

Przykład pokazujący ten sposób określania zaokrąglonych wierzchołków można znaleźć w kodzie zamieszczonym obok.





PRZYKŁAD

PUDEŁKA ELEMENTÓW

Ten przykład przedstawia prostą stronę sklepu muzycznego.

Cała strona jest umieszczona w elemencie `<div>`, w którym atrybutowi `id` przypisana została wartość `page`. Został on wyśrodkowany na stronie przy użyciu właściwości `margin`, która wielkości prawego i lewego marginesu określa jako `auto`. Przy użyciu tej samej techniki wyśrodkowano logo oraz inne treści prezentowane na stronie.

Ten główny element `<div>` ma krawędzie o postaci podwójnej linii ciągłej, a jego szerokość będzie się zmieniać i dostosowywać do wymiarów okna przeglądarki. Aby zapobiec zbytniemu zwężeniu lub rozszerzeniu strony, zastosowane zostały właściwości `min-width` oraz `max-width`.

Elementy nawigacyjne umieszczone na stronie zostały utworzone przy użyciu listy wypunktowanej. Aby lista ta była wyraźnie widoczna na stronie, określiliśmy w niej postać górnej i dolnej krawędzi obramowania. W poszczególnych elementach tej listy podana została właściwość `display`, przez co są one wyświetlane jako elementy wewnętrzwerszowe (a nie blokowe). Dzięki temu poszczególne łącza tworzące elementy nawigacyjne witryny są umieszczone obok siebie w poziomie. Odpowiednie odstępy pomiędzy poszczególnymi elementami listy zostały utworzone przy użyciu właściwości `padding`.

Szerokości elementów na stronie zostały określone przy użyciu właściwości `width` — element `` ma szerokość 570 pikseli, a elementy `<p>` mają szerokość 600 pikseli. W efekcie jednak zarówno lista, jak i akapity mają tę samą szerokość, gdyż w elemencie `` została użyta właściwość `padding` zapewniająca odstęp pomiędzy jego obramowaniem a zawartością; jak wiadomo, wypełnienia, obramowania i marginesy są dodawane do deklarowanej szerokości elementu.

Gdyby na początku dokumentu HTML nie została umieszczona deklaracja `DOCTYPE`, to w przeglądarce IE6 szerokości listy i akapitów byłyby różne ze względu na dziwactwo zaimplementowanego w niej sposobu działania modelu pudełkowego.

PRZYKŁAD

PUDEŁKA

```
<!DOCTYPE html>
<html>
<head>
  <title>Pudełka elementów</title>
  <style type="text/css">
    body {
      font-size: 80%;
      font-family: "Courier New", Courier, monospace;
      letter-spacing: 0.15em;
      background-color: #eefefef;
    }
    #page {
      max-width: 940px;
      min-width: 720px;
      margin: 10px auto 10px auto;
      padding: 20px;
      border: 4px double #000;
      background-color: #ffffff;
    }
    #logo {
      width: 150px;
      margin: 10px auto 25px auto;
    }
    ul {
      width: 580px;
      padding: 10px;
      margin: 0px auto 0px auto;
      border-top: 2px solid #000;
      border-bottom: 1px solid #000;
      text-align: center;
    }
    li {
      display: inline;
      margin: 0px 3px;
    }
    p {
      text-align: center;
      width: 600px;
      margin: 20px auto 20px auto;
      font-weight: normal;
    }
  </style>
</head>
<body>
  <div id="page">
    <div id="logo"></div>
    <ul>
      <li>Element</li>
      <li>Element</li>
      <li>Element</li>
    </ul>
    <p>Witaj w naszym sklepie!</p>
  </div>
</body>
</html>
```

PRZYKŁAD

PUDEŁKA

```
a {  
    color: #000000;  
    text-transform: uppercase;  
    text-decoration: none;  
    padding: 6px 18px 5px 18px;}  
a:hover, a.on {  
    color: #cc3333;  
    background-color: #ffffff;}  
</style>  
</head>  
<body>  
    <div id="page">  
        <div id="logo">  
              
        </div>  
        <ul id="navigation">  
            <li><a href="#" class="on">Początek</a></li>  
            <li><a href="#">Sprzedaż</a></li>  
            <li><a href="#">Naprawy</a></li>  
            <li><a href="#">O nas</a></li>  
            <li><a href="#">Kontakt</a></li>  
        </ul>  
        <p>  
              
        </p>  
        <p>  
            Specjalizujemy się w sprzedaży oraz naprawach klasycznych instrumentów  
            klawiszowych, a w szczególności: Fender Rhodes, Wurlitzer EP200 oraz  
            Hohner Clavinet.  
        </p>  
    </div>  
</body>  
</html>
```


PODSUMOWANIE

PUDEŁKA ELEMENTÓW

- ▶ CSS uznał, że każdy element HTML jest umieszczony w odrębnym pudełku.
- ▶ Wymiary tych pudełek można kontrolować przy użyciu właściwości CSS.
- ▶ Właściwości CSS pozwalają także określać postać obramowania, marginesów oraz wypełnień elementów.
- ▶ Korzystając z właściwości `display` oraz `visibility`, można ukrywać lub wyświetlać elementy.
- ▶ Istnieje możliwość wyświetlania elementów blokowych jako elementów wewnętrznych oraz elementów wewnętrznych jako elementów blokowych.
- ▶ Czytelność i przejrzystość strony można poprawiać, kontrolując wymiary pudełek elementów oraz odstępy pomiędzy wierszami tekstu.
- ▶ W CSS3 wprowadzono możliwość wyświetlania obrazów jako obramowania elementów oraz tworzenia zaokrąglonych wierzchołków.

14

LISTY, TABELE I FORMULARZE

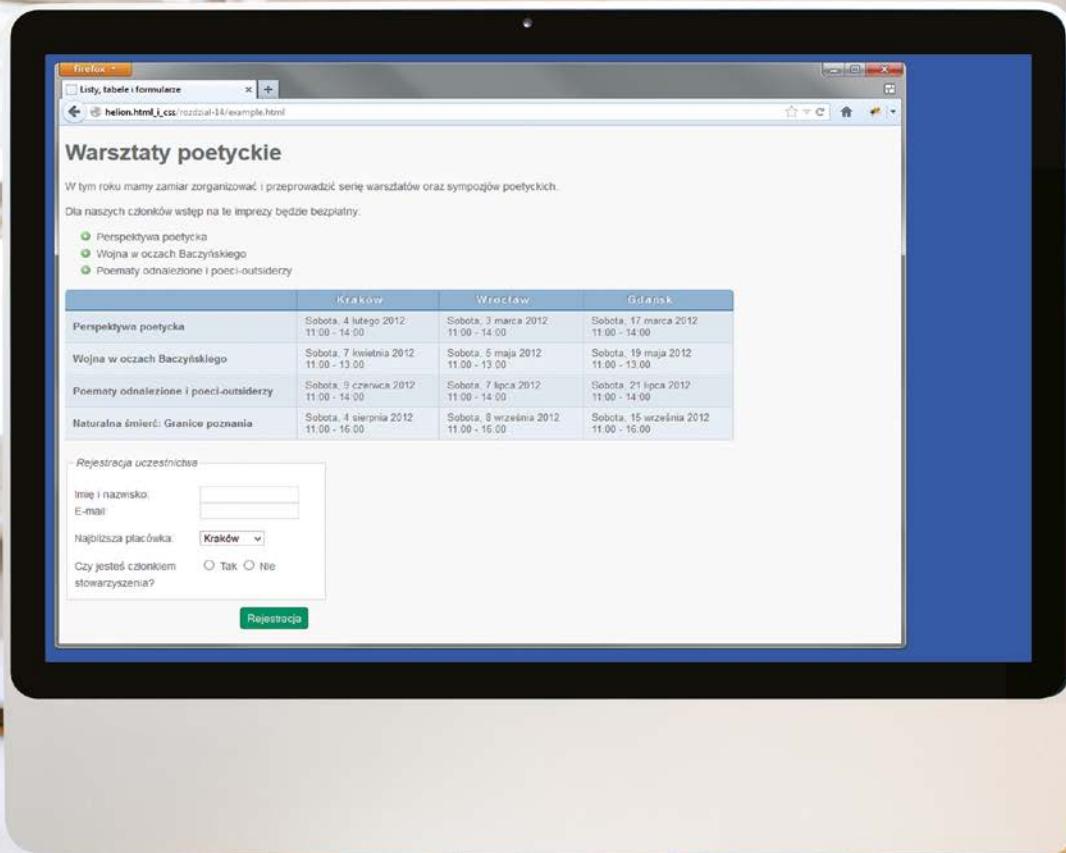
- ▶ Określanie postaci znaczników list.
- ▶ Dodawanie krawędzi oraz tła do tabel.
- ▶ Zmiana wyglądu elementów formularzy.

Kaskadowe arkusze stylów udostępniają kilka właściwości stworzonych z myślą o określaniu postaci konkretnych elementów HTML, takich jak listy, tabele i formularze.

W tym rozdziale nauczymy się:

- Określać typy znaczników lub sposoby oznaczania kolejnych elementów list wypunktowanych.
- Dodawać krawędzie i określać kolory tła w komórkach tabel.
- Określać postać pól formularzy.

Wszystkie te właściwości pozwalają na uzyskanie precyzyjnej kontroli nad wyglądem konkretnych fragmentów strony.



STYLE ZNACZNIKÓW

list-style-type

Właściwość `list-style-type` pozwala kontrolować kształt i styl znaczników (nazywanych także czasem **punktorami**), którymi oznaczane są elementy list.

Tej właściwości można używać w regułach stylów odnoszących się do elementów: ``, `` i ``.

LISTY WYPUNKTOWANE

W listach wypunktowanych tej właściwości można przypisywać wartości:

- none
- disc
- circle
- square

LISTY

UPORZĄDKOWANE

W przypadku list uporządkowanych (numerowanych) właściwość `list-style-type` może przyjmować następujące wartości:

`decimal`

1 2 3

`decimal-leading-zero`

01 02 03

`lower-alpha`

a b c

`upper-alpha`

A B C

rozdział-14/list-style-type.html

HTML

```
<h1>Fraszki</h1>
<h2>Jan Kochanowski</h2>
<ol>
  <li>Na lipę</li>
  <li>O żywocie ludzkim</li>
  <li>Raki</li>
  <li>O fraszkach</li>
  <li>O śmierci</li>
</ol>
```

```
ol {
  list-style-type: lower-roman;}
```

CSS

WYNIK

Fraszki

Jan Kochanowski

- i. Na lipę
- ii. O żywocie ludzkim
- iii. Raki
- iv. O fraszkach
- v. O śmierci

OBRAZY JAKO ZNACZNIKI list-style-image

HTML

rozdział-14/list-style-image.html

```
<h1>Lista komedii</h1>
<h2>William Shakespeare</h2>
<ul>
  <li>Komedia omyłek</li>
  <li>Wiele hałasu o nic</li>
  <li>Sen nocy letniej</li>
  <li>Kupiec wenecki</li>
</ul>
```

CSS

```
ul {
  list-style-image: url("images/star.png");
}
li {
  margin: 10px 0px 0px 0px;}
```

WYNIK

Lista komedii

William Shakespeare

- ★ Komedia omyłek
- ★ Wiele hałasu o nic
- ★ Sen nocy letniej
- ★ Kupiec wenecki

Przy użyciu właściwości `list-style-image` można określić obraz, który będzie używany jako znacznik list wypunktowanych.

Wartość tej właściwości rozpoczyna się od liter `url`, po których jest umieszczana para nawiasów zawierająca zapisany w cudzysłowie adres URL obrazu.

Tej właściwości można używać w regułach odnoszących się do elementów `` oraz ``.

Przykład zamieszczony na tej stronie przedstawia także sposób wykorzystania właściwości `margin` do powiększenia pionowego odstępu pomiędzy kolejnymi elementami listy.

UMIEJSCOWIENIE ZNACZNIKÓW

list-style-position

Domyślnie zawartość list jest nieco wcinana, natomiast właściwość `list-style-position` pozwala określić, czy znaczniki listy powinny się znajdować wewnętrz, czy na zewnątrz pudełka zawierającego główną treść listy.

Właściwość ta może przyjmować dwie wartości:

outside

Znaczniki są umieszczane z lewej strony bloku tekstu. (Jest to domyślny sposób prezentacji list, stosowany, gdy wartość tej właściwości nie zostanie określona).

inside

Znaczniki są umieszczane wewnętrz bloku tekstu (który zostaje wcięty).

W przedstawionym przykładzie szerokość listy została ograniczona do 150 pikseli. W ten sposób zawartość poszczególnych punktów listy jest wyświetlana w co najmniej dwóch wierszach, dzięki czemu wyraźnie widać, że zastosowanie wartości `inside` powoduje umieszczenie znacznika wewnętrz pierwszego wiersza tekstu.

W przykładzie, w poszczególnych elementach listy został zastosowany margines, dzięki czemu powstały pomiędzy nimi wyraźne odstępy.

rozdział-14/list-style-position.html

HTML

```
<ul class="illuminations">
  <li>Nadal jak wróbel ćwierkam...
  <li>Aby się oprzeć o ideały...
  <li>Zakreślany własnym nosem...
</ul>
<ul class="season">
  <li>Uciekałem przez sen w nocy...
  <li>Jeśli nie grzeszysz, jako...
  <li>Folgujmy paniom nie sobie, ma...
</ul>
```

```
ul {
  width: 150px;
}
li {
  margin: 10px;
}
ul.illuminations {
  list-style-position: outside;
}
ul.season {
  list-style-position: inside;
```

WYNIK

- Nadal jak wróbel ćwierkam wesoło,
Choć przeszło po mnie historii koło.
- Aby się oprzeć o ideały,
Trzeba, by one się operaly.
- Zakreślany własnym nosem,
Mikrokosmos jest kosmosem.

WYNIK

- Uciekałem przez sen w nocy,
Mając skrzydła ku pomocy, (...)
- Jeśli nie grzeszysz, jako mi
powiadasz,
Czego się, miła, tak często
spowiadasz?
- Folgujmy paniom nie sobie, ma
rada;
Miłujmy wiernie nie jest w nich
przysada. (...)

ZAPIS SKRÓTOWY

list-style

HTML

rozdział-14/list-style.html

```
<h1>Cytaty - Edgar Allan Poe</h1>
<ul>
  <li>Zawsze wierzyłem w głupców. Moi przyjaciele
  nazywają to wiarą w samego siebie.</li>
  <li>Czas to pieniądz, a pieniądz to więcej
  niż czas.</li>
  <li>Najtrwalsze i najpiękniejsze więzi duchowe
  nawiązywane bywają od pierwszego wejrzenia.</li>
</ul>
```

Podobnie jak w wielu innych sytuacjach w CSS, także podczas określania postaci list można skorzystać z zapisu skróconego, umożliwiającego podanie kilku wartości stylów w jednym miejscu. Możliwość tę daje właściwość `list-style`. Pozwala ona określić styl znacznika listy, jego obraz oraz położenie; przy czym wszystkie te informacje można podawać w dowolnej kolejności.

CSS

```
ul {
  list-style: inside circle;
  width: 300px;
}
li {
  margin: 10px 0px 0px 0px;}
```

WYNIK

Cytaty - Edgar Allan Poe

- Zawsze wierzyłem w głupców. Moi przyjaciele nazywają to wiarą w samego siebie.
- Czas to pieniądz, a pieniądz to więcej niż czas.
- Najtrwalsze i najpiękniejsze więzi duchowe nawiązywane bywają od pierwszego wejrzenia.

WŁAŚCIWOŚCI TABEL

Poznaliśmy już kilka właściwości CSS powszechnie stosowanych do określania wyglądu tabel. Na tej i kolejnej stronie przedstawiony został przykład wykorzystujący wiele z nich, a konkretnie:

width, aby określić szerokość tabeli;

padding, aby określić odległość pomiędzy obramowaniem każdej komórki tabeli a jej zawartością;

text-transform, by zapisać treść nagłówków tabeli wielkimi literami;

letter-spacing, font-size, by w dodatkowy sposób zmodyfikować wygląd nagłówków tabeli;

border-top, border-bottom, by wyświetlić obramowanie powyżej oraz poniżej nagłówka tabeli;

text-align, aby w niektórych komórkach tabeli wyrównać tekst do lewej, a w innych do prawej;

background-color, aby naprzemiennie zmieniać kolor tła w kolejnych wierszach tabeli;

:hover, aby podświetlać wiersz tabeli, w którym aktualnie znajduje się wskaźnik myszy.

rozdział-14/table-properties.html

HTML

```
<h1>Aukcja literatury obcej</h1>
<table>
  <tr>
    <th>Autor</th>
    <th>Tytuł</th>
    <th class="money">Cena minimalna</th>
    <th class="money">Bieżąca oferta</th>
  </tr>
  <tr>
    <td>E.E. Cummings</td>
    <td>Tulips & Chimneys</td>
    <td class="money">6520,00 zł</td>
    <td class="money">7612,95 zł</td>
  </tr>
  <tr class="even">
    <td>Charles d'Orléans</td>
    <td>Poëmes</td>
    <td class="money"></td>
    <td class="money">11 754,55 zł</td>
  </tr>
  <tr>
    <td>T.S. Eliot</td>
    <td>Poems 1909 - 1925</td>
    <td class="money">4075,00 zł</td>
    <td class="money">27 706,00 zł</td>
  </tr>
  <tr class="even">
    <td>Sylvia Plath</td>
    <td>The Colossus</td>
    <td class="money"></td>
    <td class="money">3361,00 zł</td>
  </tr>
</table>
```

Poniżej podaliśmy kilka porad dotyczących określania wyglądu tabel — dzięki nim będą one przejrzyste i czytelne.

CSS

```
body {  
    font-family: Arial, Verdana, sans-serif;  
    color: #111111;}  
table {  
    width: 600px;}  
th, td {  
    padding: 7px 10px 10px 10px;}  
th {  
    text-transform: uppercase;  
    letter-spacing: 0.1em;  
    font-size: 90%;  
    border-bottom: 2px solid #111111;  
    border-top: 1px solid #999;  
    text-align: left;}  
tr.even {  
    background-color: #efefef;}  
tr:hover {  
    background-color: #c3e6e5;}  
.money {  
    text-align: right;}
```

WYNIK

Aukcja literatury obcej

AUTOR	TYTUŁ	CENA MINIMALNA	BIEŻĄCA OFER
E.E. Cummings	Tulips & Chimneys	6520,00 zł	7612,95
Charles d'Orléans	Poèmes		11 754,55
T.S. Eliot	Poems 1909 - 1925	4075,00 zł	27 706,00
Sylvia Plath	The Colossus		3361,00

WIĘCEJ W SIECI

W sekcji *Tools* strony poświęconej książce można znaleźć więcej przykładów wykorzystania CSS do określania wyglądu tabel.

STOSOWANIE WYPEŁNIENIA KOMÓREK

Jeśli tekst umieszczony w komórce tabeli dotyka jej obramowania (lub krawędzi komórki), staje się znacznie mniej czytelny. Dodanie wypełnienia do komórki poprawia czytelność tabeli.

WYRÓŻNIENIE NAGŁÓWKÓW

Wyświetlenie wszystkich nagłówków tabeli pogrubioną czcionką (która jest zresztą domyślnym stylem elementów <th>) ułatwia ich czytanie. Można je także wyświetlać wielkimi literami oraz dodać do nich kolor tła, by wyraźnie odróżniały się od reszty zawartości tabeli.

ZMIANA KOLORU TŁA WIERSZY

Stosowanie dwóch naprzemiennie zmienianych kolorów tła wierszy znacznie ułatwia przeglądanie zawartości tabeli. Aby tabela wyglądała przejrzystie, warto do tego celu użyć dwóch kolorów, nieznacznie się od siebie różniących.

WYRÓWNYWANIE WARTOŚCI LICZBOWYCH

Można skorzystać z właściwości *text-align*, by wyrównać do prawej wszystkie kolumny zawierające liczby, dzięki czemu łatwo będzie można odróżnić bardzo duże wartości od małych.

OBRAMOWANIA PUSTYCH KOMÓREK

empty-cells

Jeśli tworzona tabela zawiera puste komórki, to można skorzystać z właściwości `empty-cells`, by określić, czy należy wyświetlać ich obramowania, czy też nie.

Ponieważ przeglądarki traktują puste komórki w odmienny sposób, to jeśli chcemy wyświetlać ich obramowania, powinniśmy użyć tej właściwości.

Można jej przypisać jedną z następujących wartości:

show

Jej użycie sprawi, że w pustych komórkach będą wyświetlane obramowania.

hide

Jej użycie sprawi, że obramowania w pustych komórkach nie będą wyświetlane.

inherit

Jeśli na stronie jedna tabela jest umieszczona wewnętrznie w innej, to użycie tej wartości sprawi, że komórki tabeli wewnętrznej będą wyświetlane w sposób określony dla tabeli zewnętrznej.

Na przedstawionym obok przykładzie, w lewej tabeli obramowania pustych komórek są wyświetlane, natomiast w tabeli po prawej stronie obramowania są niewidoczne.

rozdział-14/empty-cells.html

HTML

```
<table class="one">
  <tr>
    <td>1</td>
    <td>2</td>
  </tr>
  <tr>
    <td>3</td>
    <td></td>
  </tr>
</table>
```

```
td {
  border: 1px solid #0088dd;
  padding: 15px;}
table.one {
  empty-cells: show;}
table.two {
  empty-cells: hide;}
```

CSS

1	2
3	

1	2
3	

WYNIK

ODSTĘPY POMIĘDZY KOMÓRKAMI border-spacing, border-collapse

HTML

[rozdział-14/gaps-between-cells.html](#)

```
<table class="one">
  <tr>
    <td>1</td>
    <td>2</td>
  </tr>
  <tr>
    <td>3</td>
    <td>4</td>
  </tr>
</table>
```

CSS

```
td {
  background-color: #0088dd;
  padding: 15px;
  border: 2px solid #000000;
}
table.one {
  border-spacing: 5px 15px;
}
table.two {
  border-collapse: collapse;
```

WYNIK

1	2
3	4
3	4
3	4

separate

Obramowania komórek zostaną odsunięte od siebie (a ustawienia właściwości border-spacing oraz empty-cells będą uwzględniane).

Właściwość border-spacing pozwala określić odstępy pomiędzy sąsiadującymi komórkami tabeli. Domyslnie przeglądarki wyświetlają komórki w niewielkiej odległości od siebie, dlatego jeśli chcemy powiększyć lub zmniejszyć te odstępy, możemy to zrobić przy użyciu właściwości border-spacing.

Wartość tej właściwości jest zazwyczaj podawana w pikselach. W razie potrzeby można w niej podać dwie liczby, z których pierwsza będzie określała odstęp pomiędzy komórkami sąsiadującymi w poziomie, a druga — w pionie.

Jeśli komórki tabeli mają obramowania, to tam, gdzie się one stykają, szerokość obramowania będzie dwukrotnie większa od szerokości określonej w stylach. Istnieje jednak możliwość „zwężenia” takich stykających się ze sobą obramowań; służy do tego właściwość border-collapse, która może przyjmować następujące wartości:

collapse

W przypadku użycia tej wartości, jeśli tylko będzie to możliwe, obramowania zostaną zwężone (oznacza to, że właściwość border-spacing zostanie zignorowana, a komórki dosunięte do siebie; podobnie zignorowane zostaną ustawienia właściwości empty-cells).

OKREŚLANIE POSTACI FORMULARZY

Chyba nikt nie lubi wypełniania formularzy, dlatego jeśli postaramy się, by formularze na stronach były bardziej atrakcyjne i łatwiejsze w użyciu, więcej użytkowników je wypełni. Co więcej, jeśli spróbujemy wyświetlić formularz w kilku różnych przeglądarkach (co pokazują rysunki zamieszczone na tej stronie), okaże się, że każda z nich wyświetla go w nieco innym sposobie.

Kaskadowe arkusze stylów są bardzo często używane do określania postaci elementów formularzy. Ma to nie tylko poprawić ich atrakcyjność, lecz także sprawić, by ich wygląd w różnych przeglądarkach był bardziej spójny.

Najczęściej określany jest wygląd:

- pól tekstowych oraz wielowierszowych pól tekstowych,
- przycisków przesyłających formularze,
- etykiet — w celu ich efektownego wyrównania.

Na kolejnych stronach książki będzie mowa o tym, jak realizować te zadania przy wykorzystaniu CSS.

MAC: SAFARI

Pole tekstowe
Pole hasła
Pole wielowierszowe

Lista
Przyciski opcji
Submit

Opcja 1. Przycisk 1. Przycisk 2.
 Pole wyboru

PC: FIREFOX

Pole tekstowe
Pole hasła
Pole wielowierszowe

Lista
Przyciski opcji
Submit

Opcja 1. Przycisk 1. Przycisk 2.
 Pole wyboru

MAC: CHROME

Pole tekstowe
Pole hasła
Pole wielowierszowe

Lista
Przyciski opcji
Wyślij zapytanie

Opcja 1. Przycisk 1. Przycisk 2.
 Pole wyboru

PC: INTERNET EXPLORER

Pole tekstowe
Pole hasła
Pole wielowierszowe

Lista
Przyciski opcji
Przeslij zapytanie

Opcja 1. Przycisk 1. Przycisk 2.
 Pole wyboru

Określanie postaci pól tekstowych oraz przycisków jest stosunkowo proste. Bardziej kłopotliwe jest tworzenie stylów, które zapewnią spójny wygląd list rozwijanych, przycisków opcji oraz pól wyboru w różnych przeglądarkach.

Aby uzyskać taki efekt, warto pobrać arkusz CSS dostępny na stronie <http://formalize.me>.

Autor tej strony włożył bardzo wiele pracy, by zapewnić spójny wygląd formularzy w wielu przeglądarkach. Choć zastosowane rozwiązanie korzysta z języka JavaScript, to aby móc z niego skorzystać, nie trzeba się znać na programowaniu.

OKREŚLANIE POSTACI PÓŁ TEKSTOWYCH

CSS

rozdział-14/styling-text-inputs.html

```
input {  
    font-size: 120%;  
    color: #5a5854;  
    background-color: #f2f2f2;  
    border: 1px solid #bdbdbd;  
    border-radius: 5px;  
    padding: 5px 5px 5px 30px;  
    background-repeat: no-repeat;  
    background-position: 8px 9px;  
    display: block;  
    margin-bottom: 10px;}  
input:focus {  
    background-color: #ffffff;  
    border: 1px solid #b1e1e4;}  
input#email {  
    background-image: url("images/email.png");}  
input#twitter {  
    background-image: url("images/twitter.png");}  
input#web {  
    background-image: url("images/web.png");}
```

WYNIK



Więcej informacji na temat obrazów tła oraz sposobów ich umiejscawiania można znaleźć w rozdziale 16.

Zamieszczony tu przykład przedstawia właściwości CSS powszechnie stosowane do określania postaci pól tekstowych. Większość z nich została już omówiona we wcześniejszych rozdziałach książek.

Właściwość **font-size** określa wielkość czcionki używanej do wpisywania tekstu w polach tekstowych.

Właściwość **color** określa kolor tekstu, a **background-color** — kolor tła pól tekstowych.

Właściwość **border** pozwala wyświetlić obramowanie pola, a **border-radius** umożliwia stworzenie zaokrąglonych wierzchołków (w przeglądarkach, które na to pozwalały).

Pseudoklasa **:focus** służy do zmiany koloru tła pola, w którym aktualnie użytkownik coś wpisuje, a pseudoklasa **:hover** pozwala zastosować te same style w elementach, które użytkownik wskazał myszką.

Właściwość **background-image** określa obraz tła używany w polach. Ponieważ w każdym z pól został zastosowany inny obraz, skorzystaliśmy z selektora identyfikatora, by określić postać konkretnych elementów na podstawie wartości ich atrybutów id.

OKREŚLANIE POSTACI PRZYCISKÓW PRZESYŁAJĄCYCH

Na tej stronie przedstawione zostały niektóre właściwości, których można użyć do określania postaci przycisków przesyłających. Prezentowany przykład bazuje na arkuszu zamieszczonym na poprzedniej stronie, a przyciski przesyłające dziedziczą style określone dla elementu `<input>`.

Właściwość `color` służy do zmiany koloru tekstu wyświetlanego na przyciskach.

Właściwość `text-shadow` pozwoli zapewnić trójwymiarowy wygląd tekstu, o ile tylko używana przeglądarka ją obsługuje.

Właściwość `border-bottom` została użyta do pogrubienia dolnego obramowania przycisku, dzięki czemu wygląda on nieco bardziej przestrzennie.

Właściwość `background-color` pozwala wyróżnić przycisk spośród innych otaczających go elementów. (Stworzenie jednego, spójnego sposobu prezentacji przycisków ułatwi użytkownikom zrozumienie, jak należy korzystać z witryny). Do przycisku dodano także tło o postaci gradientu, które zostanie wykorzystane, jeśli tylko przeglądarka zapewnia taką możliwość. Zagadnienia związane z tworzeniem i stosowaniem gradientów zostały omówione na stronie 418.

rozdział-14/styling-submit-buttons.html

CSS

```
input#submit {  
    color: #444444;  
    text-shadow: 0px 1px 1px #ffffff;  
    border-bottom: 2px solid #b2b2b2;  
    background-color: #b9e4e3;  
    background: -webkit-gradient(linear, left top,  
        left bottom, from(#beeae9), to(#a8cfce));  
    background:  
        -moz-linear-gradient(top, #beeae9, #a8cfce);  
    background:  
        -o-linear-gradient(top, #beeae9, #a8cfce);  
    background:  
        -ms-linear-gradient(top, #beeae9, #a8cfce);}  
input#submit:hover {  
    color: #333333;  
    border: 1px solid #a4a4a4;  
    border-top: 2px solid #b2b2b2;  
    background-color: #a0dbc4;  
    background: -webkit-gradient(linear, left top,  
        left bottom, from(#a8cfce), to(#beeae9));  
    background:  
        -moz-linear-gradient(top, #a8cfce, #beeae9);  
    background:  
        -o-linear-gradient(top, #a8cfce, #beeae9);  
    background:  
        -ms-linear-gradient(top, #a8cfce, #beeae9);}
```

WYNIK

Rejestruj

Do zmiany wyglądu przycisku po wskazaniu go myszą została zastosowana pseudoklasa `:hover`. W tym przypadku umieszczenie wskaźnika myszy na przycisku spowoduje zmianę koloru jego tła oraz zmianę koloru tekstu (na nieco ciemniejszy), a także pogrubienie dolnego obramowania przycisku.

OKREŚLANIE POSTACI GRUP PÓŁ ORAZ ICH OPISU

CSS

[rozdział-14/styling-fieldsets-and-legends.html](#)

```
fieldset {  
    width: 350px;  
    border: 1px solid #dcdcdc;  
    border-radius: 10px;  
    padding: 20px;  
    text-align: right;}  
  
legend {  
    background-color: #efefef;  
    border: 1px solid #dcdcdc;  
    border-radius: 10px;  
    padding: 10px 20px;  
    text-align: left;  
    text-transform: uppercase;}
```

WYNIK

BIULETYN

Imię:

E-mail:

Subskrybuj

Grupy pól — elementy `<fieldset>` — są szczególnie przydatne podczas określania obszaru formularzy.

W przypadku bardzo dużych formularzy można ich używać do grupowania pól, które są ze sobą powiązane.

Z kolei ich opis — element `<legend>` — określa, jakie informacje mają być podane w danej grupie pól.

Podczas określania postaci tych dwóch elementów powszechnie stosowane są następujące właściwości.

Właściwość `width`, której używa się do określania szerokości grupy pól.

W zamieszczonym przykładzie narzucona szerokość wymusza, by pola zostały wyświetcone w dwóch wierszach. (Gdyby szerokość była większa, to pola mogłyby zostać wyświetcone w jednym wierszu).

Właściwość `color` służy do określania koloru tekstu.

Kolor tła grupy pól oraz jej opisu jest określany przy użyciu właściwości `background-color`.

Właściwość `border` określa postać obramowania wyświetlanego wokół grupy pól oraz jej opisu.

Właściwość `border-radius` pozwala zaokrąglić wierzchołki obramowania elementów w przeglądarkach, które ją obsługują.

I wreszcie właściwość `padding` została użyta, by nieco zwiększyć wypełnienie wewnętrz tych elementów.

WYRÓWNYWANIE ELEMENTÓW FORMULARZY: PREZENTACJA PROBLEMU

Zazwyczaj etykiety poszczególnych pól formularzy mają różne długości, co oznacza, że pola nie będą wyświetlane równo jedno pod drugim. Taką sytuację wyraźnie widać na przykładzie przedstawionym u dołu strony (w którym nie są używane żadne style CSS).

W tym przykładowym formularzu każde zagadnienie zostało umieszczone w odrębnym elemencie `<div>`, aby poszczególne pytania i związane z nim pola były prezentowane w osobnym wierszu. Niemniej łatwiej jest wypełniać formularze, jeśli ich pola są ładnie wyrównane jedno pod drugim. Problem ten rozwiązuje arkusz stylów przedstawiony na następnej stronie.

W wierszu, w którym użytkownik jest proszony o określenie płci, zostały umieszczone dwa elementy `<label>` (zawierające odpowiednio słowa: `mężczyzna`, `kobieta`). Sam tytuł tego wiersza został umieszczony w elemencie ``, dzięki czemu łatwiej go będzie wyrównać.

Na poprzedniej stronie przedstawiona została technika wyrównywania elementów formularzy. Jeśli formularz zawiera wyłącznie pola

rozdział-14/aligning-form-controls-problem.html

HTML

```
<form action="example.php" method="post">
  <div>
    <label for="name" class="title">Imię:</label>
    <input type="text" id="name" name="name" />
  </div>
  <div>
    <label for="email" class="title">E-mail:</label>
    <input type="email" id="email" name="email" />
  </div>
  <div>
    <span class="title">Płeć:</span>
    <input type="radio" name="gender" id="male" value="M" />
    <label for="male">M</label>
    <input type="radio" name="gender" id="female" value="F" />
    <label for="female">K</label><br />
  </div>
  <div>
    <input type="submit" value="Rejestruj" id="submit" />
  </div>
</form>
```

WYNIK BEZ UŻYCIA CSS

Imię:

E-mail:

Płeć: M K

tekstowe, wystarczy nadać im taką samą szerokość i wyrównać wszystkie elementy do prawej — dzięki temu pola formularza zostaną rozmiezczone idealnie jedno pod drugim, podobnie jak

etykiety. Jednak w przypadku bardziej złożonych formularzy konieczne będzie zastosowanie rozwiązania przypominającego to przedstawione tutaj.

WYRÓWNYWANIE ELEMENTÓW FORMULARZY: ROZWIĄZANIE PROBLEMU

CSS

[rozdział-14/aligning-form-controls-solution.html](#)

```
div {  
    border-bottom: 1px solid #efefef;  
    margin: 10px;  
    padding-bottom: 10px;  
    width: 260px;}  
.title {  
    float: left;  
    width: 100px;  
    text-align: right;  
    padding-right: 10px;}  
.radio-buttons label {  
    float: none;}  
.submit {  
    text-align: right;}
```

WYNIK

Imię:

E-mail:

Płeć: M K

Odrębna reguła stylów określa postać elementów `<div>`, wewnętrznych których umieszczone zostały poszczególne wiersze formularza (konkretnie precyzuje ona szerokość tych wierszy oraz odstęp pomiędzy nimi). W końcu przycisk przesyłający formularz został wyrównany do prawej.

Każdy wiersz formularza zawiera tytuł informujący użytkowników, co mają w nim wpisywać. W przypadku pól tekstowych takie tytuły są tworzone przy użyciu elementu `<label>`. Natomiast w wierszu zawierającym przyciski opcji został w tym celu zastosowany element ``. W obu przypadkach w tych elementach został umieszczony atrybut `class` o wartości `title`.

Aby przesunąć tytuły pól na lewą stronę, możemy skorzystać z właściwości o nazwie `float`. (Została ona bardziej szczegółowo opisana na stronach 369 – 375).

Dzięki określeniu szerokości elementów (przy użyciu właściwości `width`) wiemy, że tytuły będą miały taką samą szerokość. Dlatego wyświetcone obok nich pola formularza zostaną idealnie wyrównane.

Po użyciu właściwości `text-align` tytuły pól zostały wyrównane do prawej, natomiast użycie właściwości `padding` sprawiło, że pomiędzy tekstem tytułu a polami formularza jest pewien odstęp.

STYLE WSKAŹNIKA MYSZY

cursor

Właściwość cursor pozwala określić postać wskaźnika myszy wyświetlanego przez przeglądarkę.

Możemy się np. zdecydować na to, by wskaźnik wyświetlany w obszarze formularza miał postać dłoni z wyciągniętym palcem wskazującym.

Poniżej przedstawiono najczęściej używane wartości właściwości cursor:

```
auto
corsshair
default
pointer
move
text
wait
help
url("cursor.gif");
```

Powyższych wartości należy używać wyłącznie w celu dodawania przydatnych dla użytkowników informacji w tych miejscach strony, w których mogą się oni spodziewać odpowiedniego wskaźnika myszy (np. wyświetlanie wskaźnika o postaci czterokierunkowej strzałki po wskazaniu łączki może zmylić użytkowników).

rozdział-14/cursor.html

HTML

```
<a href="http://pl.wikipedia.org">Wikipedia</a>
```

```
a {  
    cursor: move;}
```

CSS

WYNIK



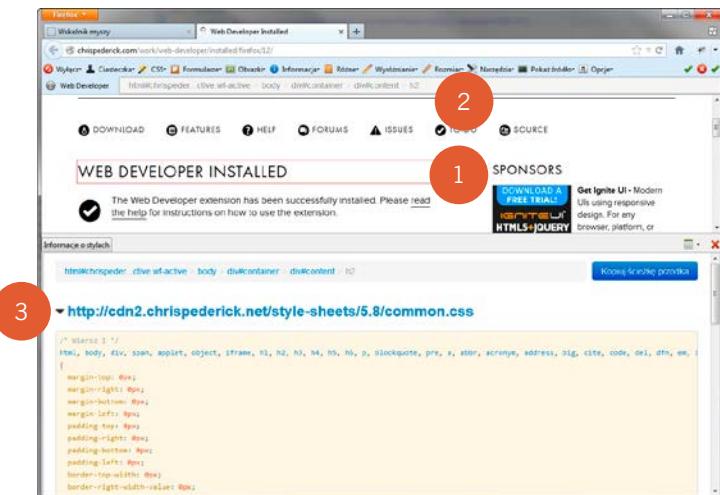
WEB DEVELOPER TOOLBAR

To bardzo przydatne rozszerzenie do przeglądarek Firefox i Chrome udostępnia narzędzia pozwalające wyświetlać style CSS, które odnoszą się do elementu wskazanego myszką, oraz strukturę kodu HTML.

Opisywane tu narzędzie można pobrać ze strony

<http://chrisspiderick.com/work/web-developer/>.

Aby wyświetlić style CSS oraz strukturę kodu HTML strony, należy otworzyć menu CSS narzędzia i wybrać opcję *View Style Information*.



1. ZARYSY ELEMENTÓW

Po wskazaniu elementu myszką wokół niego wyświetlane jest czerwone obramowanie, które doskonale pokazuje, jaki obszar jest zajmowany przez dany element.

2. STRUKTURA

Po wskazaniu elementu myszką u góry okna przeglądarki zostanie wyświetlona struktura kodu HTML. W tym przykładzie widać, że element `` został przypisany do klasy `completed` i znajduje się wewnątrz elementu `` przypisanego do klasy `to-do`. Lista natomiast znajduje się wewnątrz elementu `<div>` o identyfikatorze `page`, umieszczonego w elementach `<body>` oraz `<html>`.

Narzędzie to może być bardzo pomocne podczas tworzenia reguł CSS, gdyż pomaga sprawdzić, czy odnoszą się one do odpowiednich elementów.

Narzędzie to pozwala także na zmianę wielkości ekranu, weryfikację poprawności kodu HTML i CSS oraz wyłączanie obrazków.

3. STYLE CSS

Po umieszczeniu wskaznika myszy na elemencie można go kliknąć, by wyświetlić odnoszące się do niego reguły stylów (wraz z informacjami o numerach wierszy, w których zostały podane). Powyżej reguł wyświetlana jest nazwa pliku arkusza stylów (oraz ścieżka dostępu do niego).

Rozwiążanie to znacznie upraszcza określanie, które reguły stylów odnoszą się do elementu. Można go używać zarówno podczas testowania własnej witryny, jak i do analizy stylów używanych na innych stronach.

Firefox

Lista, tabele i formularze

helion.html_i_css/rozdzial-14/example.html

Warsztaty poetyckie

W tym roku mamy zamiar zorganizować i przeprowadzić serię warsztatów oraz sympozjów poetyckich.

Dla naszych członków wstęp na te imprezy będzie bezpłatny:

- Perspektywa poetycka
- Wojna w oczach Baczyńskiego
- Poematy odnalezione i poeci-outsiderzy

	Kraków	Wrocław	Gdańsk
Perspektywa poetycka	Sobota, 4 lutego 2012 11:00 - 14:00	Sobota, 3 marca 2012 11:00 - 14:00	Sobota, 17 marca 2012 11:00 - 14:00
Wojna w oczach Baczyńskiego	Sobota, 7 kwietnia 2012 11:00 - 13:00	Sobota, 5 maja 2012 11:00 - 13:00	Sobota, 19 maja 2012 11:00 - 13:00
Poematy odnalezione i poeci-outsiderzy	Sobota, 9 czerwca 2012 11:00 - 14:00	Sobota, 7 lipca 2012 11:00 - 14:00	Sobota, 21 lipca 2012 11:00 - 14:00
Naturalna śmierć: Granice poznania	Sobota, 4 sierpnia 2012 11:00 - 16:00	Sobota, 8 września 2012 11:00 - 16:00	Sobota, 15 września 2012 11:00 - 16:00

Rejestracja uczestnictwa

Imię i nazwisko:

E-mail:

Najbliższa placówka:

Czy jesteś członkiem stowarzyszenia?
 Tak Nie

Rejestracja

PRZYKŁAD

LISTY, TABELE I FORMULARZE



Zamieszczony tu przykład przedstawia zastosowanie kilku poznanych w tym rozdziale właściwości służących do określania wyglądu list, tabel oraz formularzy.

Znaczniki listy spotkań poetyckich, umieszczonej na początku strony, zostały wyświetlane przy użyciu obrazów. Odpowiednie odstępy pomiędzy poszczególnymi elementami listy zapewnia właściwość `line-height`.

Jeśli natomiast chodzi o umieszczoną poniżej tabelę, odstępy pomiędzy komórkami usunięto dzięki użyciu właściwości `border-spacing`. Określona została także wielkość czcionki używanej w elementach `<td>` oraz `<th>`, gdyż nie jest ona dziedziczona po przodkach.

Nagłówek tabeli ma ciemniejsze tło, jaśniejszy kolor tekstu i jest oddzielony do pozostałej zawartości tabeli linią o szerokości 2 pikseli. Okrągłe wierzchołki w nagłówku tabeli zostały utworzone przy użyciu pseudoklas: `:first-child` oraz `:last-child` (będą one widoczne w przeglądarkach, które na to pozwalają).

Tło w wierszach tabeli jest naprzemiennie zmieniane, a dodatkowo została zastosowana tekstura z różnymi obramowaniami po obu stronach komórek.

W przedstawionym formularzu powiązane ze sobą pola zostały umieszczone wewnątrz elementu `<fieldset>`. W etykietach wyświetlonych z lewej strony została użyta właściwość `float`, aby zapewnić, że pola formularza będą wyrównane w pionie.

Kiedy użytkownik kliknie pole tekstowe formularza albo umieści na nim wskaźnik myszy, zmieni się kolor tła oraz kolor obramowań pola. Zmodyfikowany zostanie także wygląd przycisku przesyłającego, by użytkownicy nie mieli wątpliwości, w jaki sposób należy przekazać dane na serwer.

PRZYKŁAD

LISTY, TABELE I FORMULARZE

```
<!DOCTYPE html>
<html>
<head>
<title>Listy, tabele i formularze</title>
<style type="text/css">
body {
    font-family: Arial, Verdana, sans-serif;
    font-size: 90%;
    color: #666;
    background-color: #f8f8f8;}
li {
    list-style-image: url("images/icon-plus.png");
    line-height: 1.6em;}
table {
    border-spacing: 0px;}
th, td {
    padding: 5px 30px 5px 10px;
    border-spacing: 0px;
    font-size: 90%;
    margin: 0px;}
th, td {
    text-align: left;
    background-color: #e0e9f0;
    border-top: 1px solid #f1f8fe;
    border-bottom: 1px solid #cbd2d8;
    border-right: 1px solid #cbd2d8;}
tr.head th {
    color: #fff;
    background-color: #90b4d6;
    border-bottom: 2px solid #547ca0;
    border-right: 1px solid #749abe;
    border-top: 1px solid #90b4d6;
    text-align: center;
    text-shadow: -1px -1px 1px #666666;
    letter-spacing: 0.15em;}
td {
    text-shadow: 1px 1px 1px #ffffff;}
```

PRZYKŁAD

LISTY, TABELE I FORMULARZE

```
tr.even td, tr.even th {  
    background-color: #e8eff5;}  
tr.head th:first-child {  
    -webkit-border-top-left-radius: 5px;  
    -moz-border-radius-topleft: 5px;  
    border-top-left-radius: 5px;}  
tr.head th:last-child {  
    -webkit-border-top-right-radius: 5px;  
    -moz-border-radius-topright: 5px;  
    border-top-right-radius: 5px;}  
fieldset {  
    width: 310px;  
    margin-top: 20px;  
    border: 1px solid #d6d6d6;  
    background-color: #ffffff;  
    line-height: 1.6em;}  
legend {  
    font-style: italic;  
    color: #666666;}  
input[type="text"] {  
    width: 120px;  
    border: 1px solid #d6d6d6;  
    padding: 2px;  
    outline: none;}  
input[type="text"]:focus,  
input[type="text"]:hover {  
    background-color: #d0e2f0;  
    border: 1px solid #999999;}  
input[type="submit"] {  
    border: 1px solid #006633;  
    background-color: #009966;  
    color: #ffffff;  
    border-radius: 5px;  
    padding: 5px;  
    margin-top: 10px;}  
input[type="submit"]:hover {  
    border: 1px solid #006633;
```

PRZYKŁAD

LISTY, TABELE I FORMULARZE

```
background-color: #00CC33;
color: #ffffff;
cursor: pointer;}
.title {
  float: left;
  width: 160px;
  clear: left;}
.submit {
  width: 310px;
  text-align: right;}
</style>
</head>
<body>
<h1>Warsztaty poetyckie</h1>
<p>W tym roku mamy zamiar zorganizować i przeprowadzić serię warsztatów
  oraz sympozjów poetyckich.</p>
<p>Dla naszych członków wstęp na te imprezy będzie bezpłatny:</p>
<ul>
  <li>Perspektywa poetycka</li>
  <li>Wojna w oczach Baczyńskiego</li>
  <li>Poematy odnalezione i poeci-outsiderzy</li>
</ul>
<table>
  <tr class="head">
    <th></th>
    <th>Kraków</th>
    <th>Wrocław</th>
    <th>Gdańsk</th>
  </tr>
  <tr>
    <th>Perspektywa poetycka</th>
    <td>Sobota, 4 lutego 2012<br />11:00 - 14:00</td>
    <td>Sobota, 3 marca 2012<br />11:00 - 14:00</td>
    <td>Sobota, 17 marca 2012<br />11:00 - 14:00</td>
  </tr>
  <tr class="even">
    <th>Wojna w oczach Baczyńskiego</th>
```

PRZYKŁAD

LISTY, TABELE I FORMULARZE

```
<td>Sobota, 7 kwietnia 2012<br />11:00 - 13:00</td>
<td>Sobota, 5 maja 2012<br />11:00 - 13:00</td>
<td>Sobota, 19 maja 2012<br />11:00 - 13:00</td>
</tr>
<tr>
<th>Poematy odnalezione i poeci-outsiderzy</th>
<td>Sobota, 9 czerwca 2012<br />11:00 - 14:00</td>
<td>Sobota, 7 lipca 2012<br />11:00 - 14:00</td>
<td>Sobota, 21 lipca 2012<br />11:00 - 14:00</td>
</tr>
<tr class="even">
<th>Naturalna śmierć: Granice poznania</th>
<td>Sobota, 4 sierpnia 2012<br />11:00 - 16:00</td>
<td>Sobota, 8 września 2012<br />11:00 - 16:00</td>
<td>Sobota, 15 września 2012<br />11:00 - 16:00</td>
</tr>
</table>
<form action="http://www.jakaswitryna.com.pl/form.php" method="get">
<fieldset>
<legend>Rejestracja uczestnictwa</legend>
<p><label class="title" for="name">Imię i nazwisko:</label>
<input type="text" name="name" id="name"><br />
<label class="title" for="email">E-mail:</label>
<input type="text" name="email" id="email"></p>
<p><label for="location" class="title">Najbliższa placówka:</label>
<select name="location" id="location">
<option value="kr">Kraków</option>
<option value="wr">Wrocław</option>
<option value="gd">Gdańsk</option>
</select></p>
<span class="title">Czy jesteś członkiem stowarzyszenia?</span>
<label><input type="radio" name="member" value="yes" /> Tak</label>
<label><input type="radio" name="member" value="no" /> Nie</label></p>
</fieldset>
<div class="submit"><input type="submit" value="Rejestracja" /></div>
</form>
</body>
</html>
```


PODSUMOWANIE

LISTY, TABELE I FORMULARZE

- ▶ Oprócz właściwości CSS, które zostały opisane w innych rozdziałach i mogą być stosowane do określania wyglądu zawartości wszystkich elementów, dostępnych jest kilka właściwości przeznaczonych do określania wyglądu list, tabel i formularzy.
- ▶ Postać znaczników list można określić przy użyciu właściwości `list-style-type` oraz `list-style-image`.
- ▶ W różnych przeglądarkach komórki tabel mają różne obramowania oraz odstępy pomiędzy sobą; istnieją właściwości, które pozwalają kontrolować te aspekty wyglądu tabel i sprawić, by wyglądały spójnie.
- ▶ Korzystanie z formularzy będzie prostsze, jeśli ich pola zostaną idealnie wyrównane w pionie przy użyciu stylów.
- ▶ Formularze mogą zdecydowanie zyskać dzięki zastosowaniu stylów, które nadadzą im bardziej interaktywny wygląd.

15

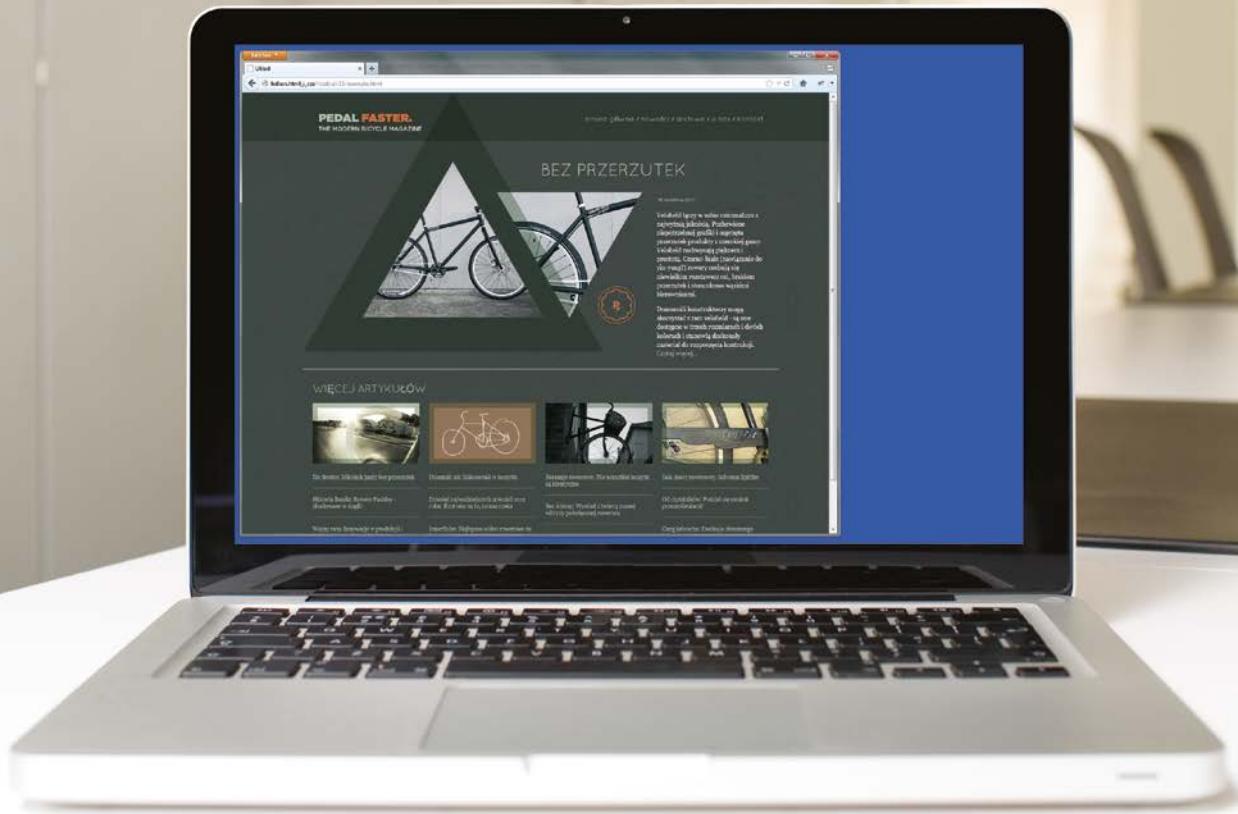
UKŁAD

- ▶ Kontrola położenia elementów.
- ▶ Tworzenie układów stron.
- ▶ Projektowanie pod kątem ekranów o różnej rozdzielczości.

W niniejszym rozdziale będzie mowa o tym, jak kontrolować położenie elementów na stronie oraz jak tworzyć witryny o atrakcyjnym układzie.

Przy tej okazji musimy się dowiedzieć, czym projektowanie stron przeznaczonych do oglądania na ekranie różni się od projektowania materiałów przeznaczonych do prezentacji z użyciem innych mediów (np. druku). W tym rozdziale:

- Poznamy różne sposoby umiejscawiania elementów: przy wykorzystaniu normalnego rozkładu treści, umiejscawiania względnego, bezwzględnego oraz z zastosowaniem elementów płynących.
- Odkryjemy, że różne urządzenia mają ekran o różnych wymiarach oraz rozdzielczości, i dowiemy się, jak te czynniki wpływają na proces projektowania stron.
- Nauczymy się rozróżniać układy o stałej szerokości oraz układy elastyczne, jak również dowiemy się, jak je tworzyć.
- Dowiemy się, jak projektanci wykorzystują siatki, by tworzone przez nich projekty wyglądały bardziej profesjonalnie.



KLUCZOWE POJĘCIA ZWIĄZANE Z UMIEJSCAWIANIEM ELEMENTÓW

ELEMENTY KONSTRUKCYJNE

Kaskadowe arkusze stylów zakładają, że każdy element strony znajduje się w osobnym pudełku. Te pudełka elementów mogą być **blokowe** lub **wewnętrzwerszowe**.

Pudełka blokowe rozpoczynają się w nowym wierszu i pełnią funkcję podstawowych elementów konstrukcyjnych układu strony, natomiast pudełka wewnętrzwerszowe są rozmieszczane wraz z otaczającym je tekstem. Obszar zajmowany przez każde pudełko elementu można kontrolować, określając jego szerokość (a czasami także wysokość). By oddzielić pudełka elementów od siebie, można określić ich obramowania, marginesy, wypełnienia oraz kolory tła.

ELEMENTY BLOKOWE

ROZPOCZYNAJĄ SIĘ
W NOWYM WIERSZU

Oto kilka przykładów:
<h1> <p>

Lorem ipsum

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit.

- *Lorem ipsum dolor sit*
- *Consectetur adipisicing*
- *Elit, sed do eiusmod*

ELEMENTY WEWNĄTRZWERSZOWE

SĄ UMIESZCZANE
POMIĘDZY OTACZAJĄCYM
JE TEKSTEM

Oto kilka przykładów:
 <i>

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

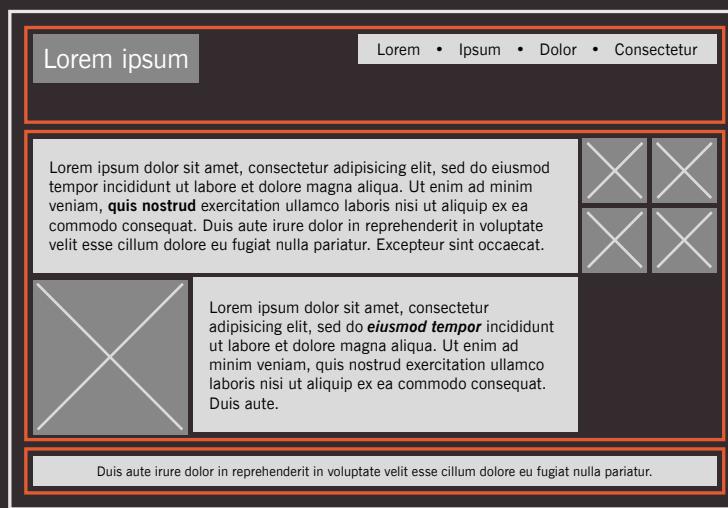


Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

ELEMENTY ZAWIERAJĄCE

Jeśli jeden element blokowy jest umieszczony wewnątrz innego, to zewnętrzny z nich jest nazywany elementem **zawierającym** lub **rodzicem**.

Często spotykanym rozwiązaniem jest grupowanie kilku elementów wewnątrz jednego elementu `<div>` (lub innego elementu blokowego). W taki sposób można np. zgrupować wszystkie elementy tworzące nagłówek witryny (takie jak logo, nazwa czy główne elementy nawigacyjne). Element `<div>`, wewnątrz którego jest umieszczona taka grupa, nazywany jest elementem **zawierającym**.



Pudełko elementu może być umieszczone wewnątrz kilku innych elementów blokowych. Element zawierający zawsze jest bezpośrednim **rodzicem** takiego elementu.

Pomarańczowe linie na powyższym diagramie reprezentują elementy `<div>`. Nagłówek (zawierający logo oraz elementy nawigacyjne) został umieszczony w jednym elemencie `<div>`, główna treść strony znalazła się w kolejnym, a nagłówek — w jeszcze jednym. Te trzy elementy `<div>` są umieszczone bezpośrednio w elemencie `<body>`. Drugi `<div>` jest elementem zawierającym, wewnątrz którego zostały umieszczone dwa akapity tekstu po łacińsku oraz obrazki (reprezentowane przez szare kwadraty z krzyżkami).

KONTROLA POŁOŻENIA ELEMENTÓW

W CSS wykorzystywane są następujące **schematy pozycjonowania**, pozwalające na kontrolowanie układu strony: rozkład normalny, umiejscawianie względne oraz umiejscawianie bezwzględne. Stosowany schemat umiejscawiania określa się przy użyciu właściwości **position**. Elementy pływające tworzy się poprzez zastosowanie właściwości **float**.

ROZKŁAD NORMALNY

Każdy element blokowy jest wyświetlanym w nowym wierszu, przez co każdy kolejny taki element jest wyświetlany poniżej poprzedniego. Jeśli szerokość elementów została określona i jeśli dwa kolejne elementy zmieszcza się obok siebie, to i tak nie zostaną wyświetlone w poziomie. Takie jest domyślne działanie przeglądarki (chyba że każemy jej zrobić coś innego).

UMIEJSCAWIANIE WZGLĘDNE

W tym przypadku elementy są przesuwane z miejsca, które normalnie by zajmowały, ku górze, w prawo, lewo lub ku dołowi. Zmiany te nie wpływają na położenie sąsiednich elementów — te pozostają wyświetlone w tych samych miejscach, w których zostały umieszczone przy wykorzystaniu rozkładu normalnego.

UMIEJSCAWIANIE

BEZWZGLĘDNE

W tym przypadku położenie elementów jest określone względem elementu, w którym są one umieszczone. Elementy umiejscawiane w ten sposób są usuwane z normalnego rozkładu strony, co oznacza, że przestają mieć wpływ na pozostałe otaczające je elementy (po prostu obszar zajmowany przez takie elementy jest ignorowany). Elementy umiejscawiane w sposób bezwzględny przesuwają się, gdy użytkownik korzysta z paska przewijania.

Lorem ipsum

 Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

 Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

 Duis aute irure dolor in reprehenderit in voluptate velit.

Akapity są wyświetlane jeden pod drugim w kierunku dołu strony.

Patrz strona 364.

Lorem ipsum

 Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

 Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

 Duis aute irure dolor in reprehenderit in voluptate velit.

Drugi akapit tekstu zostanie przesunięty w dół i w prawo w stosunku do miejsca, które normalnie by zajmowało.

Patrz strona 365.

Lorem ipsum

 Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

 Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

 Duis aute irure dolor in reprehenderit in voluptate velit.

Nagłówek został umieszczony w prawym górnym rogu strony, a akapit zaczyna się od samej góry ekranu (tak jakby przed nim nie było nagłówka).

Patrz strona 366.

Może się okazać, że w celu wskazania położenia, w jakim powinien się znaleźć element, konieczne będzie **podanie przesunięcia** określającego, jak daleko w góre lub w dół albo w prawo lub w lewo ma zostać przesunięty element. (Więcej na ten temat piszemy na następnych stronach, na których przedstawiamy schematy umiejscawiania).

UMIEJSZCZANIE USTALONE

To pewien rodzaj umiejscawiania bezwzględnego, w którym położenie elementu jest określone względem okna przeglądarki, a nie elementu zawierającego. Elementy umiejscowione w ten sposób nie wpływają na położenie innych, sąsiadujących z nimi elementów i nie przesuwają się, gdy użytkownik przewija zawartość strony.

`Lorem ipsum dolor sit amet,
 consectetur adipisciing elit, sed do
 eiusmod tempor incididunt ut labore et
 dolore magna aliqua.`

`Ut enim ad minim veniam, quis
 nostrud exercitation ullamco laboris
 nisi ut aliquip ex ea commodo
 consequat.`

`Duis aute irure dolor in reprehenderit
 in voluptate velit.`

ELEMENTY PŁYWĄCĄCE

Mechanizm elementów płynących pozwala usunąć wybrany element z normalnego rozkładu strony i umieścić go tuż przy prawej lub lewej krawędzi elementu zawierającego. Taki element płynący staje się elementem blokowym, wokół którego rozmieszczana jest pozostała zawartość strony.

`Lorem ipsum dolor sit
 amet, consectetur
 adipisciing elit, sed do
 eiusmod tempor incididunt ut labore et
 dolore magna aliqua.`

`Ut enim ad minim veniam, quis
 nostrud exercitation ullamco laboris
 nisi ut aliquip ex ea commodo
 consequat.`

`Duis aute irure dolor in reprehenderit
 in voluptate velit.`

Nagłówek został umieszczony pośrodku strony i przesunięty w dół od jej górnej krawędzi o 25% wysokości okna przeglądarki. (Pozostałe elementy strony są rozmieszczane zgodnie z rozkładem normalnym).

Patrz strona 367.

Nagłówek został przekształcony w element płynący umieszczony z lewej strony, dzięki czemu akapit tekstu może go otaczać po prawej.

Patrz strony 369 – 375.

Jeśli którykolwiek element zostanie usunięty z normalnego rozkładu strony, to pudełka elementów mogą na siebie nachodzić. W takich przypadkach właściwość **z-index** pozwala określić, które z nich ma być wyświetlane powyżej drugiego.

ROZKŁAD NORMALNY

position: static

W przypadku rozkładu normalnego każdy element blokowy jest umieszczany poniżej poprzedniego. Ponieważ jest to normalny sposób, w jaki przeglądarki traktują elementy HTML, nie trzeba stosować żadnych właściwości CSS, by zaznaczyć, że elementy mają być tak rozmieszczone; można jednak to zrobić przy użyciu właściwości:

`position: static;`

W zamieszczonym tu przykładzie nie określiliśmy szerokości nagłówka, dzięki czemu widać, że domyślnie będzie on zajmował całą dostępną szerokość okna przeglądarki.

Szerokość akapitów natomiast została ograniczona do 450 pikseli. Na ich przykładzie widać, że elementy blokowe w rozkładzie normalnym są umieszczane w nowym wierszu, nawet jeśli nie zajmują całej szerokości okna przeglądarki.

Wszystkie przykłady użyte do przedstawienia sposobów umiejscawiania będą miały taką samą strukturę kodu HTML.

`rodzial-15/normal-flow.html`

HTML

```
<body>
  <h1>Ewolucja roweru</h1>
  <p>W 1817 roku baron von Drais wymyślił maszynę biegową, która miała mu pomóc w szybszym poruszaniu się po ogrodach królewskich...</p>
</body>
```

```
body {
  width: 750px;
  font-family: Arial, Verdana, sans-serif;
  color: #665544;}
h1 {
  background-color: #efefef;
  padding: 10px;}
p {
  width: 450px;}
```

CSS

WYNIK

Ewolucja roweru

W 1817 roku baron von Drais wymyślił maszynę biegową, która miała mu pomóc w szybszym poruszaniu się po ogrodach królewskich. Składała się ona z dwóch kół o tej samej wielkości, umieszczonych w jednej osi, przednie skręcone koło było zamontowane do ramy, na której się siedziało okrakiem. Urządzenie było napędzane poprzez odpchanie się nogami od ziemi, więc jazda bardzo przypominała spacerowanie.

Od nazwiska konstruktora urządzenie to zostało nazwane draisine (drezyną). Było wykonane w całości z drewna. Nie cieszyło się długą popularnością i stanowiło raczej chwilowy kryzys mody, gdyż w praktyce można było wykorzystywać do poruszania się wyłącznie po dobrze utrzymanych drogach i ścieżkach, takich jak te w parkach lub ogrodach.

Kolejne dwukołowe urządzenie do jazdzenia skonstruowano w 1865 roku, kiedy to pojawił się pomysł umieszczenia pedałów bezpośrednio na przednim kole. Urządzenie nazywane velocypedem (velocipede - szybka stopa) było także znane jako „wytraszacz kości”, gdyż jego drewniana konstrukcja w połączeniu z brukowanymi ulicami ówczesnych miast sprawiała, że jazda nim była wyjątkowo niewygodna. Velocypedy stały się bardzo modne, a w większych miastach powstawały nawet specjalne szkoły uczące jazdy na nich.

UMIEJSCAWIENIE WZGLĘDNE position: relative

HTML

rodzaj-15/position-relative.html

```
<body>
  <h1>Ewolucja roweru</h1>
  <p>W 1817 roku baron von Drais wymyślił maszynę
    biegową, która miała mu pomóc w szybszym
    poruszaniu się po ogrodach królewskich...</p>
</body>
```

CSS

```
p.example {
  position: relative;
  top: 10px;
  left: 100px;}
```

WYNIK

Ewolucja roweru

W 1817 roku baron von Drais wymyślił maszynę biegową, która miała mu pomóc w szybszym poruszaniu się po ogrodach królewskich. Składała się ona z dwóch kół o tej samej wielkości, umieszczonych w jednej osi; przednie skręcone koło było zamontowane do ramy, na której się siedzisko okrakiem. Urządzenie było napędzane poprzez odpychanie się nogami od ziemi, więc jazda bardzo przypominała spacerowanie.

Od nazwiska konstruktora urządzenie to zostało nazwane draisienne (drezyna). Było wykonane w całości z drewna. Nie cieszyło się długą popularnością i stanowiło raczej chwilowy krzyk mody, gdyż w praktyce można było wykorzystywać do poruszania się wyłącznie po dobrze utrzymanych drogach i ścieżkach, takich jak te w parkach lub ogrodach. Kolejne dwukolowe urządzenie do jezdnia skonstruowano w 1865 roku, kiedy to pojawił się pomysł umieszczenia pedałów bezpośrednio na przednim kole. Urządzenie nazywane velocipedem (velocipede - szybka stopa) było także znane jako „wytrząsacz kości”, gdyż jego drewniana konstrukcja w połączeniu z brukowanymi ulicami ówczesnych miast sprawiała, że jazda na nim była wyjątkowo niewygodna. Velocipedy stały się bardzo modne, a w większych miastach powstały nawet specjalne szkoły uczące jazdy na nich.

Umiejscawianie względne umożliwia przesuwanie elementów względem miejsca, w którym zostały umieszczone przy użyciu rozkładu normalnego.

Wybrany element można np. przesunąć o 10 pikseli w dół względem miejsca, w którym zostały on umieszczone, lub o 20% w prawo.

Aby zaznaczyć, że element ma być umiejscawiany względnie, należy użyć właściwości position i przypisać jej wartość relative.

Dodatkowo trzeba użyć jednej z właściwości określających przesunięcie (top, bottom, left lub right), by określić, w którym kierunku i na jaką odległość należy przesunąć element.

Aby przesunąć pudełko elementu w górę lub w dół, należy skorzystać z właściwości top lub bottom.

W przypadku przesuwania pudełka elementu w lewo lub prawo należy użyć właściwości left lub right.

Wartości przypisywane tym właściwościom są zazwyczaj podawane w pikselach, jednostkach em lub są wartościami procentowymi.

UMIEJSCAWIANIE BEZWZGLĘDNE

position: absolute

Kiedy właściwości position przypisana zostanie wartość absolute, element jest usuwany z normalnego rozkładu treści i nie przestaje mieć wpływu na położenie innych, otaczających go elementów strony. (Przeglądarka rozmieszcza je tak, jakby takiego elementu w ogóle nie było w kodzie strony).

Właściwości przesunięcia (top, bottom, left oraz right) określają położenie elementu wyznaczane w odniesieniu do jego elementu zawierającego.

W przedstawionym przykładzie nagłówek został umieszczony na samej górze strony, w odległości 500 pikseli od jej lewej krawędzi. Szerokość nagłówka wynosi 250 pikseli.

Przy wykorzystaniu właściwości width określona została także szerokość elementów <p>, dzięki czemu treść strony nie przesłania nagłówka i jest czytelna.

Domyślnie przeglądarki umieszczają u góry elementów <h1> pewien margines. To właśnie dlatego pomiędzy górną krawędzią strony a tym elementem jest widoczny odstęp. Gdybyśmy chcieli usunąć ten margines, wystarczyłoby dodać do arkusza stylów regułę odnoszącą się do elementów <h1> i umieścić w niej deklarację margin: 0px;.

rodzaj-15/position-absolute.html

HTML

```
<body>
  <h1>Ewolucja roweru</h1>
  <p>W 1817 roku baron von Drais wymyślił maszynę biegową, która miała mu pomóc w szybszym poruszaniu się po ogrodach królewskich...</p>
</body>
```

```
h1 {
  position: absolute;
  top: 0px;
  left: 500px;
  width: 250px;}
p {
  width: 450px;}
```

CSS

WYNIK

Ewolucja roweru

W 1817 roku baron von Drais wymyślił maszynę biegową, która miała mu pomóc w szybszym poruszaniu się po ogrodach królewskich. Składała się ona z dwóch kół o tej samej wielkości, umieszczonych w jednej osi; przednie skręcone kolo było zamontowane do ramy, na której się siedzisko okrakiem. Urządzenie było napędzane poprzez odpychanie się nogami od ziemi, więc jazda bardzo przypominała spacerowanie.

Od nazwiska konstruktora urządzenie to zostało nazwane draisienne (drezyną). Było wykonane w całości z drewna. Nie cieszyło się dużą popularnością i stanowiło raczej chwilowy kryzys mody, gdyż w praktyce można je było wykorzystywać do poruszania się wyłącznie po dobrze utrzymanych drogach i ścieżkach, takich jak te w parkach lub ogrodach.

Kolejne dwukołowe urządzenie do jeżdżenia skonstruowano w 1865 roku, kiedy to pojawił się pomysł umieszczenia pedałów bezpośrednio na przednim kole. Urządzenie nazywane velocypedem (velocipede - szybka stopa) było także znane jako „wytrząsacz kości”, gdyż jego drewniana konstrukcja w połączeniu z brukowanymi ulicami ówczesnych miast sprawiała, że jazda na nim była wyjątkowo niewygodna. Velocipedy stały się bardzo modne, a w większych miastach powstały nawet specjalne szkoły uczące jazdy na nich.

Pierwsze rowery wykonane w całości z metalu pojawiły się w 1870 roku. (Wcześniej metalurgia nie była dostatecznie

UMIEJSCAWIANIE USTALONE

position: fixed

HTML

rodzial-15/position-fixed.html

```
<body>
  <h1>Ewolucja roweru</h1>
  <p class="example">W 1817 roku baron von Drais
    wymyślił maszynę biegową, która miała mu pomóc
    w szybszym poruszaniu się po ogrodach
    królewskich...</p>
</body>
```

CSS

```
h1 {
  position: fixed;
  top: 0px;
  left: 50px;
  padding: 10px;
  margin: 0px;
  width: 100%;
  background-color: #efefef;
}
p.example {
  margin-top: 100px;}
```

WYNIK

Ewolucja roweru

W 1817 roku baron von Drais wymyślił maszynę biegową, która miała mu pomóc w szybszym poruszaniu się po ogrodach królewskich. Składała się ona z dwóch kół o tej samej wielkości, umieszczonych w jednej osi, przednie skrętnie koło było zamontowane do ramy, na której się siedziało okrakiem. Urządzenie było napędzane poprzez odpchanie się nogami od ziemi, więc jazda bardzo przypominała spacerowanie.

Od nazwiska konstruktora urządzenie to zostało nazwane draisine (drezyna). Było wykonane w całości z drewna. Nie cieszyło się dużą popularnością i stanowiło raczej chwilowy krzyk mody, gdyż w praktyce można było wykorzystywać do poruszania się wyłącznie po dobrze utrzymywanych drogach i ścieżkach, takich jak te w parkach lub ogrodach.

Kolejne dwukolowe urządzenie do jazdzenia skonstruowane w 1865 roku, kiedy to pojawił się pomysł umieszczenia pedałów bezpośrednio na przednim kole. Urządzenie nazywane wielocipedem (velocipede - szybka stopa) było także znane jako „wyrzasać kości”, gdyż jego drewniana konstrukcja w połączeniu z brukowanymi ulicami ówczesnych miast sprawiała, że jazda na nim była wyjątkowo niewygodna. Wielocipy stały się bardzo modne, a w większych miastach powstały nawet specjalne szkoły uczące jazdy na nich.

Pierwsze rowery wykonane w całości z metalu pojawiły się w 1870 roku. (Wcześniej metalurgia nie była dostatecznie rozwinięta i nie dawała możliwości wytworzenia metalu na tyle wytrzymalego, by można było go użyć do tworzenia niewielkich, lecz dostatecznie mocnych części). Pedaly były w nich przytwierdzane bezpośrednio do przedniego koła, bez żadnego mechanizmu wolnobiegi. Pełne gumowe opony oraz długie szprychy przednich kół sprawiały, że jazda na tych rowerach była znacznie wygodniejsza niż na ich poprzednikach.

Umiejscawianie ustalone jest jednym z typów umiejscawiania bezwzględnego. Jego zastosowanie wymaga przypisania właściwości position wartości fixed.

W przypadku tego sposobu umiejscawiania położenie elementu jest określone względem okna przeglądarki. Dlatego kiedy użytkownik przewija zawartość strony, element pozostaje w tym samym miejscu. Warto, by Czytelnik wypróbował zamieszczony tu przykład w swojej przeglądarce i zobaczył, jakie efekty da umiejscawianie ustalone.

Do określenia miejsca w oknie przeglądarki, w którym zostanie umieszczony element, używane są właściwości przesunięcia.

W przedstawionym tu przykładzie nagłówek został umieszczony w lewym górnym wierzchołku okna przeglądarki. Kiedy użytkownik zacznie przewijać stronę, kolejne akapity będą znikły poniżej tego nagłówka.

Elementy `<p>` są rozmieszczane zgodnie z rozkładem normalnym i całkowicie ignorują obszar, który zajmowałaby nagłówek. Dlatego w pierwszym elemencie `<p>` została zastosowana właściwość `margin-top`, która sprawi, że akapit ten będzie początkowo wyświetlony poniżej nagłówka `<h1>` widocznego na samej górze strony.

ELEMENTY NACHODZĄCE NA SIEBIE z-index

W przypadku stosowania umiejscawiania względnego, bezwzględnego lub ustalonego pudełka elementów mogą na siebie nachodzić. Wówczas element, który jest umieszczony bliżej końca dokumentu HTML, jest wyświetlany powyżej elementów wcześniejszych.

Jeśli zależy nam na kontroli wzajemnego przesłaniania różnych elementów, możemy to zrobić przy użyciu właściwości `z-index`. Wartościami tej właściwości są liczby, a im większa będzie jej wartość, tym „wyżej” element zostanie umieszczony. I tak element, w którym właściwość `z-index` ma wartość 10, będzie przesłaniał element, w którym właściwość ta ma wartość 5.

Przedstawiony tu przykład jest dosyć podobny do tego z poprzedniej strony, jednak elementy `<p>` są rozmieszczane przy użyciu umiejscawiania względnego. Ponieważ kolejne akapity tekstu są umiejscawiane w sposób względny, w razie przewijania zawartości strony byłyby one domyślnie wyświetlane na nagłówku. Dlatego aby zagwarantować, że element `<h1>` ciągle będzie widoczny, w odnoszącej się do niego regule stylów określiliśmy wartość właściwości `z-index`.

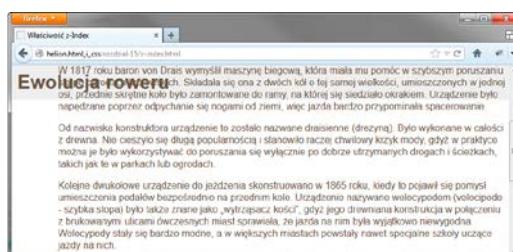
Właściwość `z-index` jest także czasami nazywana **kontekstem stosu** (ang. *stacking context*; gdyż bloki elementów mogą tworzyć stos wzdłuż osi Z). Jeśli Czytelnik miał kontakt z programami do komputerowego składowania tekstu, to może uznać tę właściwość za odpowiednik opcji *Przesuń na wierzch* lub *Przesuń pod spód*.

rodzaj-15/z-index.html

CSS

```
h1 {  
    position: fixed;  
    top: 0px;  
    left: 0px;  
    margin: 0px;  
    padding: 10px;  
    width: 100%;  
    background-color: #efefef;  
    z-index: 10;  
}  
  
p {  
    position: relative;  
    top: 70px;  
    left: 70px;  
}
```

WYNIK BEZ UŻYCIA WŁAŚCIWOŚCI Z-INDEX



WYNIK Z UŻYCIEM WŁAŚCIWOŚCI Z-INDEX



ELEMENTY PŁYWAJĄCE

float

HTML

rodzial-15/float.html

```
<body>
  <h1>Ewolucja roweru</h1>
  <blockquote>Życie jest jak jazda na rowerze.
  Aby zachować równowagę, musisz posuwać się
  do przodu! - Albert Einstein</blockquote>
  <p>W 1817 roku baron von Drais wymyślił
  maszynę biegową, która miała mu pomóc w szybszym
  poruszaniu się po ogrodach królewskich. Składała
  się ona z dwóch kół o tej samej...</p>
</body>
```

CSS

```
blockquote {
  float: right;
  width: 275px;
  font-size: 130%;
  font-style: italic;
  font-family: Georgia, Times, serif;
  margin: 0px 0px 10px 10px;
  padding: 10px;
  border-top: 1px solid #665544;
  border-bottom: 1px solid #665544;}
```

WYNIK

Ewolucja roweru

W 1817 roku baron von Drais wymyślił maszynę biegową, która miała mu pomóc w szybszym poruszaniu się po ogrodach królewskich. Składała się ona z dwóch kół o tej samej wielkości, umieszczonych w jednej osi, przednie skreślone kolo było zamontowane do ramy, na której się siedziło okrakiem. Urządzenie było napędzane poprzez odpchanie się nogami od ziemi, więc jazda bardzo przypominała spacerowanie.

Od nazwiska konstruktora urządzenie to zostało nazwane draisienne (drezyna). Było wykonane w całości z drewna. Nie cieszyło się dużą popularnością i stanowiło raczej chwilowy krzyk mody, gdyż w praktyce można je było wykorzystywać do poruszania się wyłącznie po dobrze utrzymanych drogach i ścieżkach, takich jak te w parkach lub ogrodach.

Kolejne dwukolowe urządzenie do jeżdżenia skonstruowano w 1865 roku, kiedy to pojawił się pomysł umieszczenia pedałów bezpośrednio na przednim kole. Urządzenie nazywane wełocypedem (velocipede - szybka stopa) było także znane jako „wytrząsacz kości”, gdyż jego drewniana konstrukcja w połączeniu z brukowanymi ulicami ówczesnych miast sprawiała, że jazda na nim była wyjątkowo niewygodna. Wełocypedy stały się bardzo modne, a w większych miastach powstawały nawet specjalne szkoły uczące jazdy na nich.

„Życie jest jak jazda na rowerze. Aby zachować równowagę, musisz posuwać się do przodu!” - Albert Einstein

Właściwość float pozwala usunąć element z normalnego rozkładu strony i umieścić go możliwie jak najdalej na lewo lub prawo wewnątrz elementu zawierającego.

Cała pozostała treść umieszczona wewnątrz takiego elementu zawierającego będzie otaczała ten płynący element.

W razie zastosowania właściwości float należy także określić szerokość elementu płynącego — przy użyciu właściwości width. Jeśli tego nie zrobimy, wyniki uzyskiwane w różnych przeglądarkach mogą być niespójne; w efekcie element płynący naprawdopodobnie zajmie całą dostępną szerokość elementu, w którym został umieszczony (czyli tak, jakby był rozmieszczany w normalny sposób).

W przedstawionym obok przykładzie wyświetliśmy na stronie cytat, używając do tego celu elementu <blockquote>. Jego elementem zawierającym jest <body>.

Nasz element <blockquote> jest elementem płynącym, dosuniętym do prawej krawędzi strony; akapity tekstu otaczają go z lewej strony.

STOSOWANIE WŁAŚCIWOŚCI FLOAT DO WYŚWIETLANIA ELEMENTÓW OBOK SIEBIE

W wielu układach stron ich poszczególne elementy są wyświetlane jeden obok drugiego. Bardzo często w celu uzyskania takiego efektu używana jest właściwość float.

W przypadku zastosowania elementów pływających ich wysokość może mieć znaczący wpływ na miejsce, w którym element zostanie umieszczony.

Przykład zamieszczony na tej stronie przedstawia sześć akapitów, w których określono szerokości oraz zastosowano właściwość float.

W odróżnieniu od tego, co można by oczekiwąć, czwarty akapit nie został przeniesiony w dół i wyświetlony przy lewej krawędzi strony — zamiast tego przeglądarka wyświetliła go poniżej trzeciego akapitu.

Stało się tak, ponieważ poniżej trzeciego akapitu jest wolne miejsce pozwalające na rozpoczęcie czwartego, nie można jednak go przesunąć bardziej w lewo, gdyż nie pozwala na to drugi akapit.

Problem ten można by z powodzeniem rozwiązać, nadając wszystkim akapitom tę samą wysokość — równą wysokości najwyższego z nich — jednak metoda ta raczej nie nadaje się do wykorzystania

rodzaj15/using-float.html

HTML

```
<body>
  <h1>Ewolucja roweru</h1>
  <p>W 1817 roku baron von Drais wymyślił maszynę biegową, która miała mu pomóc w szybszym poruszaniu się po ogrodach królewskich...</p>
```

```
body {
  width: 750px;
  font-family: Arial, Verdana, sans-serif;
  color: #665544;
}
p {
  width: 230px;
  float: left;
  margin: 5px;
  padding: 5px;
  background-color: #efefef;}
```

CSS

Ewolucja roweru

WYNIK

W 1817 roku baron von Drais wymyślił maszynę biegową, która miała mu pomóc w szybszym poruszaniu się po ogrodach królewskich.

Od nazwiska konstruktora urządzenie to zostało nazwane draisienne (drezyna). Było wykonane w całości z drewna. Nie cieszyło się długą popularnością i stanowiło raczej chwilowy krzyk mody.

Kolejne dwukołowe urządzenie do jazdy skonstruowano w 1865 roku.

Pierwsze rowery wykonane w całości z metalu pojawiły się w 1870 roku. (Wcześniej metalurgia nie była dostatecznie rozwinięta i nie dawała możliwości wytworzenia metalu na tyle wytrzymalego, by można było go użyć do tworzenia niewielkich, lecz dostatecznie mocnych części).

Przednie koła rowerów zaczęły się robić coraz większe, gdyż uszwadniało sobie, że im większe będą, tym dalej rower przejedzie.

Ponieważ kierowca takiego urządzenia znajdował się wysoko nad jego środkiem ciężkości, to jeśli pojazd uderzył w kamień...

w praktyce, gdyż zazwyczaj wielkość tekstu umieszczanego w akapicie lub kolumnie jest różna. Znacznie częstszym

rozwiązaniem tego problemu jest użycie właściwości clear (przedstawionej na następnej stronie).

POMIJANIE ELEMENTÓW PŁYWAJĄCYCH

clear

HTML

rodzial-15/clear.html

```
<p class="clear">Pierwsze rowery wykonane w całości  
z metalu pojawiły się w roku 1870. (Wcześniej  
metalurgia nie była dostatecznie rozwinięta i nie  
dawała możliwości wytworzenia metalu na tyle  
wytrzymałości, by można go użyć do tworzenia  
niewielkich, lecz dostatecznie mocnych części).</p>
```

CSS

```
body {  
    width: 750px;  
    font-family: Arial, Verdana, sans-serif;  
    color: #665544;}  
p {  
    width: 230px;  
    float: left;  
    margin: 5px;  
    padding: 5px;  
    background-color: #efefef;}  
.clear {  
    clear: left;}
```

WYNIK

Ewolucja roweru

W 1817 roku baron von Drais wymyślił maszynę biegową, która miała mu pomóc w szybszym poruszaniu się po ogrodach królewskich.

Od nazwiska konstruktora urządzenie to zostało nazwane draisienne (dreyna). Było wykonane w całości z drewna. Nie cieszyło się dużą popularnością i stanowiło raczej chwilowy kryzys mody.

Kolejne dwukolowce urządzenie do jeżdżenia skonstruowano w 1865 roku.

Pierwsze rowery wykonane w całości z metalu pojawiły się w 1870 roku. (Wcześniej metalurgia nie była dostatecznie rozwinięta i nie dawała możliwości wytworzenia metalu na tyle wytrzymałości, by można było go użyć do tworzenia niewielkich, lecz dostatecznie mocnych części).

Przednie koła rowerów zaczęły się robić coraz większe, gdyż uświadomiiono sobie, że im większe będą, tym dalej rower przejdzie.

Ponieważ kierowca takiego urządzenia znajdował się wysoko nad jego środkiem ciężkości, to jeśli pojazd uderzył w kamień...

Korzystając z właściwości `clear`, możemy poinformować przeglądarkę, że żaden element (umieszczony wewnątrz tego samego elementu zawierającego) nie powinien stykać się z lewą, albo prawą, krawędzią danego elementu. Właściwość ta może przyjmować następujące wartości:

left

Lewa krawędź pudełka elementu nie powinna stykać się z jakimkolwiek innym elementem znajdującym się wewnątrz tego samego elementu zawierającego.

right

Prawa krawędź pudełka elementu nie powinna stykać się z jakimkolwiek innym elementem znajdującym się wewnątrz tego samego elementu zawierającego.

both

Ani lewa, ani prawa krawędź pudełka elementu nie powinna stykać się z jakimkolwiek innym elementem znajdującym się wewnątrz tego samego elementu zawierającego.

none

Inne elementy mogą dowolnie stykać się z krawędziami pudełka danego elementu. W przedstawionym obok przykładzie właściwość `clear` została zastosowana w czwartym akapicie. Reguła odnosząca się do tego elementu używa właściwości `clear`, by poinformować, że żaden element nie może się znajdować z jego lewej strony — dlatego akapit zostanie przesunięty w dół strony.

RODZICE ELEMENTÓW PŁYWAJĄCYCH: PREZENTACJA PROBLEMU

Jeśli wewnątrz elementu zawierającego znajduje się *jedynie* element pływający, to niektóre przeglądarki uznają, że ma on zerową wysokość.

Jak widać w tym przykładzie, obramowanie o grubości 1 piksela, wyświetlane wokół elementu zawierającego akapity, zostało całkowicie zwinięte i wygląda jak pozioma kreska o grubości 2 pikseli.

rodzaj-15/float-problem.html

HTML

```
<h1>Ewolucja roweru</h1>
<div>
<p>W 1817 roku baron von Drais wymyślił maszynę biegową, która miała mu pomóc w szybszym poruszaniu się po ogrodach królewskich...</p>
</div>
</body>
```

```
div {
  border: 1px solid #665544;}
```

CSS

Ewolucja roweru

W 1817 roku baron von Drais wymyślił maszynę biegową, która miała mu pomóc w szybszym poruszaniu się po ogrodach królewskich.

Pierwsze rowery wykonane w całości z metalu pojawiły się w 1870 roku. (Wczesniej metalurgia nie była dostatecznie rozwinięta i nie dawała możliwości wytworzenia metalu na tyle wytrzymałe, by można było go użyć do tworzenia niewielkich, lecz dostatecznie mocnych części).

Od nazwiska konstruktora urządzenie zostało nazwane draisienne (drewna). Było wykonane w całości z drewna. Nie cieszyło się dużą popularnością i stanowiło raczej chwilowy kryzys mody.

Przednie koła rowerów zaczęły się robić coraz większe, gdyż uświadomiono sobie, że im większe będą, tym dalej rower przejedzie.

Kolejne dwukołowe urządzenie do jezdnienia skonstruowano w 1865 roku.

Ponieważ kierowca takiego urządzenia znajdował się wysoko nad jego środkiem ciężkości, to jeśli pojazd uderzył w kamień...

WYNIK

RODZICE ELEMENTÓW PŁYWAJĄCYCH: ROZWIĄZANIE PROBLEMU

HTML

rodzaj-15/float-solution.html

```
<body>
<h1>Ewolucja roweru</h1>
<div>
<p>W 1817 roku baron von Drais wymyślił maszynę biegową, która miała mu pomóc w szybszym poruszaniu się po ogrodach królewskich...</p>
</div>
</body>
```

CSS

```
div {
  border: 1px solid #665544;
  overflow: auto;
  width: 100%;}
```

WYNIK

Ewolucja roweru

W 1817 roku baron von Drais wymyślił maszynę biegową, która miała mu pomóc w szybszym poruszaniu się po ogrodach królewskich.

Pierwsze rowery wykonane w całości z metalu pojawiły się w 1870 roku. (Wcześniej metalurgia nie była dostatecznie rozwinięta i nie dawała możliwości wytworzenia metalu na tyle wytrzymalego, by można było go użyć do tworzenia niewielkich, lecz dostatecznie mocnych części)

Od nazwiska konstruktora urządzenie to zostało nazwane draisienne (drożyna). Było wykonane w całości z drewna. Nie cieszyło się dużą popularnością i stanowiło raczej chwilowy kryzys mody.

Przedni kola rowerów zaczęły się robić coraz większe, gdyż uświadomiono sobie, że im większe będą, tym dalej rower przejedzie.

Kolejne dwukołowe urządzenie do jeżdżenia skonstruowano w 1865 roku.

Ponieważ kierowca takiego urządzenia znajdował się wysoko nad jego środkiem ciężkości, to jeśli pojazd uderzył w kamień...

Tradycyjnym sposobem na rozwiązywanie tego problemu było dodawanie jeszcze jednego elementu za ostatnim elementem pływającym (lecz wewnątrz elementu zawierającego). Dla tego elementu była tworzona reguła stylów zawierająca właściwość `clear` o wartości `both`. Rozwiązywanie to jednak sprawiało, że w kodzie strony pojawiały się dodatkowy element, którego jedynym przeznaczeniem była korekta wysokości elementu zawierającego.

Nieco później twórcy stron zaczęli stosować rozwiązywanie bazujące wyłącznie na wykorzystaniu stylów — dzięki temu bowiem możliwe jest uniknięcie dodawania kolejnego elementu do kodu strony. Rozwiązywanie to polega na dodaniu do stylów odnoszących się do elementu zawierającego (w naszym przypadku jest to element `<div>`) dwóch reguł:

- właściwości `overflow` z wartością `auto`,
- właściwości `width` z wartością `100%`.

TWORZENIE UKŁADÓW WIELOKOLUMNOWYCH PRZY UŻYCIU ELEMENTÓW PŁYWAJĄCYCH

Wiele stron wykorzystuje projekty składające się z wielu kolumn. Można je utworzyć, definiując każdą z kolumn jako element `<div>`. Trzy przedstawione poniżej właściwości CSS pozwalają umieścić kolumny jedna obok drugiej:

width

Ta właściwość określa szerokości kolumn.

float

Ta właściwość pozwala umieścić kolumny jedna obok drugiej.

margin

Ta właściwość służy do zapewnienia odpowiednich odstępów pomiędzy kolumnami.

Układ dwukolumnowy, taki jak przedstawiony na tej stronie, wymagałby zastosowania dwóch elementów `<div>` — jednego do utworzenia kolumny głównej i drugiego na pasek boczny.

Wewnątrz każdego z nich można umieszczać dowolne elementy — nagłówki, akapity, obrazy, a nawet inne elementy `<div>`.

[rodzaj-15/columns-two.html](#)

HTML

```
<h1>Ewolucja roweru</h1>
<div class="column1of2">
  <h3>Pierwszy rower</h3>
  <p>W 1817 roku baron von Drais wymyślił maszynę biegową, która miała mu pomóc w szybszym poruszaniu się po ogródkach kolonialnych. Składała się ona z dwóch kół o tej samej wielkości, umieszczonych w jednej osi, przednie skrętnie koło było zamontowane do ramy, na której się śledziło okrążenie. Urządzenie było niespełnione poprzez odpadanie się nogami od ziemi, więc zostało zastąpione specjalnymi...
```

```
.column1of2 {
  float: left;
  width: 620px;
  margin: 10px;
}
.column2of2 {
  float: left;
  width: 300px;
  margin: 10px;
}
```

CSS

Ewolucja roweru

Pierwszy rower

W 1817 roku baron von Drais wymyślił maszynę biegową, która miała mu pomóc w szybszym poruszaniu się po ogródkach kolonialnych. Składała się ona z dwóch kół o tej samej wielkości, umieszczonych w jednej osi, przednie skrętnie koło było zamontowane do ramy, na której się śledziło okrążenie. Urządzenie było niespełnione poprzez odpadanie się nogami od ziemi, więc zostało zastąpione specjalnymi...

Od nazwiska konstruktora pochodziło to urządzenie określone mianem "draisine" (drayzyn). Dylemą było to, że nie było możliwości skręcania kierunku jazdy, co było niebezpieczne dla użytkowników. Przykładem jest historia, kiedy na ulicach Paryża pojawiły się pierwsze rowery, a ich użytkownicy zaczęli jeździć po chodnikach, krzykli, głośno i rozbijały kostki. W efekcie pojawiły się pierwsze przepisy zakazujące jazdy po chodnikach, mimo iż rowery były bezpieczne dla użytkowników.

Dalsze usprawnienia

Kolejne dwukolumnowe urządzenie do jazdy zostało stworzone w 1865 roku, kiedy to pojawiły się pierwsze unieszczerzane podkola bezpośrednio na przednim kole. Urządzenie nazwane "velocyped" (velocipede - szybka stopa) było trudne zatrzymać jako "wyskoczące koło", gdyż jego drążeniem koniecznego było przeniesienie z bieżnika do przedniego koła, bez zatrzymania mechanizmu wózka. Przykładem jest rower z mechanizmem biegowym, który pozwalał na jazdę na tych rowerach tylko na miejscu.

Pierwsze rowery wykorzystane w zawodach i mistrzostwach pojawiły się w 1870 roku. Wraz z kolejnymi modyfikacjami i nowymi mechanizmami wykorzystywano mechanizmy na tylnym kołku, który pozwalał na jazdę do przodu i cofanie się, lecz zatrzymywanie i skręcanie było trudne. Pierwsze rowery z mechanizmem biegowym pojawiły się w 1871 roku, kiedy to pojawiły się mechanizmy biegowe do przedniego koła, bez zatrzymania mechanizmu wózka. Przykładem jest rower z mechanizmem biegowym, który pozwalał na jazdę na tych rowerach tylko na miejscu.

Historia rowerów

- 1817: Draisine
- 1865: Velociped
- 1870: Rower z wysokim kołem
- 1871: Rower z biegowym kołem
- 1873: Rower z niskim kołem
- 1885: Kolo popypowane

WYNIK

HTML

rodzial-15/columns-three.html

```
<h1>Ewolucja roweru</h1>
<div class="column1of3">
  <h3>Pierwszy rower</h3>
</div>
<div class="column2of3">
  <h3>Dalsze usprawnienia</h3> ...
</div>
<div class="column3of3">
  <h3>Historia rowerów</h3> ...
</div>
```

CSS

```
.column1of3, .column2of3, .column3of3 {
  width: 300px;
  float: left;
  margin: 10px;}
```

WYNIK

Ewolucja roweru

Pierwszy rower

W 1817 roku baron von Drais wymyślił pierwszy rower, który miał mu służyć w stylu gondoli. Składał się on z dwóch kół o tej samej wielkości, umieszczonych w jednej osi, przednie skrętną koło było zamontowane do ramy, na której się siedzało okrągłe. Urządzenie było napędzane poprzez odpchanie się nogami od ziemi, więc jazda bardziej przypominała spacerowanie.

Od nazwiska konstruktora urządzenie zostało nazwane draisinem (drózyna). Było wykonane w całości z drewna. Nie cieszyło się dużą popularnością i stało się popularne jedynie w niektórych krajach, gdyż w praktyce można je było wykorzystywać do poruszania się wyłącznie po dobre utrzymanych drogach i ścieżkach, takich jak te w parkach lub ogrodach.

Dalsze usprawnienia

Kolejne draisinowe urządzenia do jazdy na kolanach skonstruowano w 1855 roku, kiedy to zastąpiono drewniane pędułówki bezpośrednio na przednim kole. Urządzenie nazywane velocipedem (velocipede - szybka stopa) było także znane jako "wytrząsacz koła", gdyż jego drzewiana konstrukcja w połączeniu z brukowanymi ulicami ówczesnych miast sprawiała, że jazda na nim była wyjątkowo niebezpieczna. Velocipedy stały się bardzo modne, a w niektórych miejscach powstawały nawet specjalne szkoły uczące jazdy na nich.

Pierwsze rowery wykorzystane w pełni z metalu pojawiły się w 1870 roku. (Wczesna metalurgia nie była dostatecznie rozwinięta i nie dawała możliwości wyborowania metalu na tyle wytrzymałe, by można było go użyć do tworzenia niewielkich, lecz dostatecznie mocnych części). Pedaly były w nich przytwierdzone bezpośrednio do przedniego koła, bez zadejścia mechanizmu skrętnego. Palne gumowe opony oraz długie sztypuchy przednich kół sprawiały, że jazda na tych rowerach była znacznie wygodniejsza niż na nich poprzednikach.

Historia rowerów

- 1817 - Draisine
- 1855 - Velociped
- 1870 - Rower z tykrami kołem
- 1878 - Poprawa bezpieczeństwa
- 1885 - Twarde, gumowe koła
- 1888 - Koła pompowane

Analogicznie układ trójkolumnowy można stworzyć, umieszczając obok siebie trzy elementy <div> w sposób przedstawiony na tej stronie.

ROZMIARY EKRANÓW

Osoby odwiedzające nasze witryny będą korzystały z urządzeń wyposażonych w ekrany o różnej wielkości, prezentujące różne ilości informacji. Dlatego ich układ musi zapewniać możliwość oglądania strony na ekranach o różnej wielkości.



iPhone 4

Wielkość: 3,5 cala
Rozdzielcość: 960×640 pikseli



iPad 2

Wielkość: 9,7 cala
Rozdzielcość: 1024×768 pikseli

Projektując materiały do druku, wiemy, jaka będzie wielkość papieru, na którym zostaną one umieszczone. Jednak w przypadku projektowania materiałów z myślą o publikacji w internecie stajemy przed unikalnym problemem związanym z tym, że użytkownicy mogą korzystać z ekranów o różnej wielkości.

Od kiedy zaczęto sprzedawać pierwsze komputery, rozmiary ekranów cały czas się powiększają. Oznacza to, że jedni użytkownicy mogą korzystać z monitorów 13-calowych, a inni z 27-calowych (lub większych).

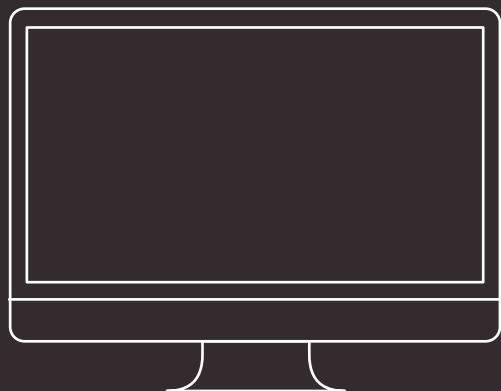
Rozmiar ekranu determinuje wielkość okna, jakie użytkownik może wyświetlić, oraz to, jak duży fragment strony będzie mógł w nim oglądać. Poza tym coraz szybciej rośnie liczba urządzeń kieszonkowych (tabletów i smartfonów) wyposażonych w niewielkie ekranы.

ROZDZIELCZOŚĆ EKRANU

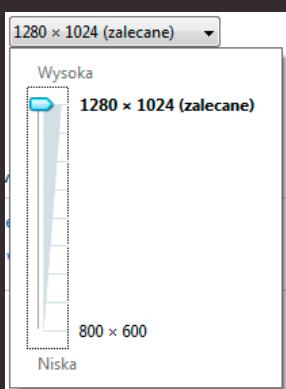
Rozdzielcość określa liczbę punktów wyświetlanych na ekranie na każdy cal jego długości. Niektóre urządzenia mają większą rozdzielcość niż komputery stacjonarne, a większość systemów operacyjnych pozwala użytkownikom zmieniać rozdzielcość monitorów.



13" MacBook
Wielkość: 13,3 cala
Rozdzielcość: 1280×800 pikseli



27" iMac
Wielkość: 27 cali
Rozdzielcość: 2560×1440 pikseli



Wielkość komputerów pozwala użytkownikom zmieniać rozdzielcość monitora lub liczbę wyświetlanych pikseli. Na przykładowym zrzucie obok widać opcje pozwalające na zmianę rozdzielcości ekranu w zakresie od 800×600 do 1280×1024.

Warto zauważyć, że im większa jest rozdzielcość, tym mniejszy będzie wyświetlany tekst. Wiele urządzeń przenośnych ma ekran o wyższej rozdzielcości niż komputery stacjonarne.

ROZMIARY STRON

Ponieważ występują znaczne różnice pomiędzy wielkościami ekranów a ich rozdzielczością, projektanci stron często starają nadać im szerokość z zakresu od 960 do 1000 pikseli (większość użytkowników będzie w stanie oglądać je na ekranach swoich urządzeń).

Znacznie trudniejsze jest odgadnięcie wysokości obszaru strony, jaki użytkownicy będą mogli zobaczyć na ekranach swoich urządzeń. Przez kilka lat projektanci uznawali, że użytkownicy są w stanie bez przewijania oglądać pierwsze 570 – 600 pikseli wysokości strony, i niektórzy starali się zmieścić w tym obszarze wszystkie najważniejsze informacje (obawiając się, że oglądający nie będą przewijać stron w dół).

Jednak ze względu na coraz to większe ekranы oraz zwiększoną popularność urządzeń kieszonkowych obszar, jaki użytkownicy mogą oglądać, może być bardzo różny.

Obszar strony, który użytkownicy oglądają bez konieczności jej przewijania, jest czasami nazywany obszarem przed zagięciem (termin ten został wymyślony

przez wydawców prasy i odnosi się do fragmentu pierwszej strony widocznego, gdy gazeta jest złożona na pół).

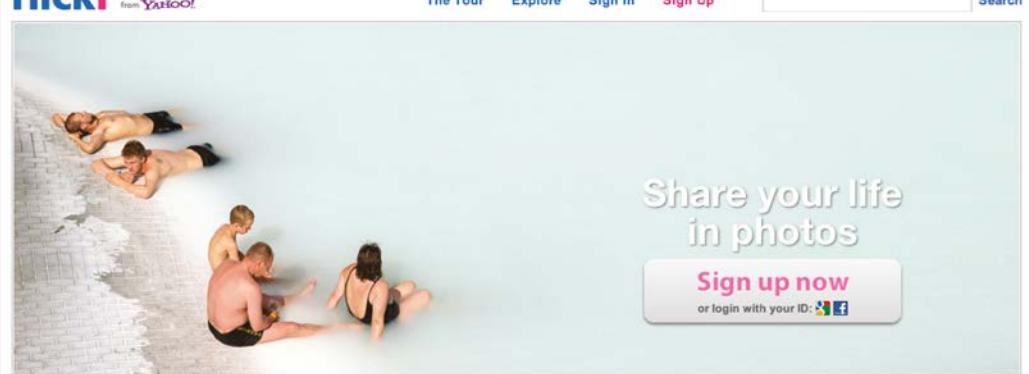
Obecnie uważa się, że jeśli użytkownik jest zainteresowany treścią strony, to najprawdopodobniej przewinie ją w dół. Niemniej jednak badania dotyczące użyteczności wykazały, że użytkownicy są w stanie ocenić stronę w ciągu niecałej sekundy, a zatem i tak wciąż istotne jest, by w tym czasie pokazać użytkownikowi, że witryna może być dla niego ważna i interesująca.

W efekcie wielu projektantów nadal stara się przekazać informacje o zawartości witryny na obszarze o wysokości od 570 do 600 pikseli, umieszczając poniżej niego wskazówki dotyczące dodatkowych treści. Należy jednak pamiętać, by obszar ten nie był zbyt zapchany.

W czasie pisania tej książki coraz większą popularnością wśród projektantów stron cieszyły się projekty dostosowujące się do wielkości ekranu, na którym są prezentowane.

Zaciemniony obszar poniższego zrzutu jest początkowo niewidoczny ze względu na ograniczenia przeglądarki. A zatem by wyświetlić tę dolną część strony, użytkownik musi ją przewinąć.

Przy czym użytkownik widzi informacje o tym, co znajduje się w dalszej części strony, i wie, że znajdzie na niej coś więcej, jeśli tylko ją przewinie.



1000 px

flickr® from YAHOO! The Tour Explore Sign In Sign Up Search

Share your life in photos

Sign up now
or login with your ID:  

© by peterbaker

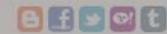
Upload
More ways to get your photos online.
Multiple ways to upload your photos to Flickr—through the web, your mobile device, email or your favorite photo applications.

Discover
See what's going on in your world.
Keep up with your friends and share your stories with comments & notes. Add rich information like tags, locations & people.

Share
Your photos are everywhere you are.
Upload your photos once to Flickr, then easily and safely share them through Facebook, Twitter, email, blogs and more.



Sign up now Free!
or learn more

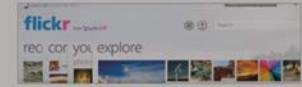
It takes less than a minute to create your free account & start sharing!
 Have a Google or Facebook account? You can use them to sign in!

Community
Flickr is made of people.

Join one of over 10 million active groups to take part in the conversation, learn from our

Privacy
Your photos are safe with us.

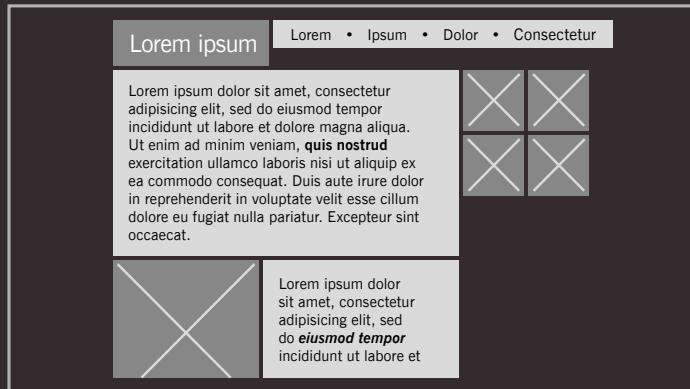
Share photos only with the people you want to with our easy privacy settings. Flickr's multiple-

Flickr on the go
Mobile options to keep you going.

Flickr is always in your back pocket with apps for iPhone, Windows 7, and more. Or use

570 px

UKŁADY O STAŁEJ SZEROKOŚCI

Układy o stałej szerokości nie zmieniają się wraz ze zmianami wymiarów okna przeglądarki. Ich wymiary są zazwyczaj określane w pikselach.



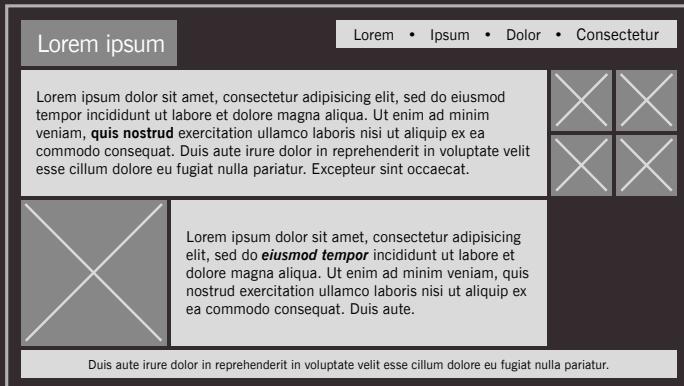
ZALETY

- Wartości wyrażane w pikselach zapewniają precyzyjną kontrolę nad rozmiarami i położeniem elementów.
- Projektanci mają większą kontrolę nad wyglądem i położeniem elementów strony niż w układach elastycznych.
- Istnieje możliwość kontrolowania długości wierszy tekstu niezależnie od wielkości okna przeglądarki.
- Wielkość obrazu zawsze będzie taka sama w stosunku do reszty strony.

WADY

- Wokół zawartości strony mogą się pojawić duże puste przestrzenie.
- Jeśli ekran użytkownika ma znacznie wyższą rozdzielcość od ekranu projektanta, to wynikowa strona może się wydawać mała i być nieczytelna.
- Jeśli użytkownik powiększy czcionkę, tekst może przestać się mieścić w wyznaczonym mu obszarze.
- Te układy zapewniają najlepsze efekty na urządzeniach o wymiarach i rozdzielcości porównywalnej z komputerami stacjonarnymi i laptopami.
- Może się zdarzyć, że strony o tej samej zawartości będą miały większą wysokość niż w przypadku zastosowania układu elastycznego.

UKŁADY ELASTYCZNE



ZALETY

- Strona rozszerza się i zajmuje całą dostępną szerokość okna przeglądarki, dzięki czemu na większych ekranach nie są widoczne wokół niej żadne puste obszary.
- Jeśli okno przeglądarki jest niewielkie, zawartość strony może się zwęzić, dzięki czemu użytkownik nie będzie musiał przewijać jej w poziomie.
- Projekt dostosuje się do wielkości czcionki wybranej przez użytkownika, nawet jeśli będzie większa od tej użytej przez projektanta (strona może się rozszerzyć).

WADY

- W przypadku braku kontroli nad szerokością poszczególnych sekcji strony jej wygląd może się znaczco różnić od planowanego — wokół niektórych elementów mogą się pojawić puste obszary, a inne mogą być stłoczone jeden obok drugiego.
- Jeśli okno przeglądarki będzie bardzo szerokie, to wiersze tekstu mogą się stać długie, co znacznie utrudnia czytanie.
- Jeśli okno przeglądarki będzie bardzo wąskie, słowa mogą zostać ściśnięte, a bardzo krótkie wiersze — zawierać jedynie kilka wyrazów.
- Jeśli element o określonych wymiarach (taki jak obraz) zostanie umieszczony w elemencie nadzależnym

Układy elastyczne rozszerzają się i zwężają, gdy użytkownik zmienia wymiary okna przeglądarki. Zazwyczaj ich wymiary są określane przy użyciu wartości procentowych.

zbyt małym, by go wyświetlić (np. dlatego, że użytkownik zmniejszył okno przeglądarki), to element ten może nachodzić i przestaniać inne treści strony.

Ponieważ układy elastyczne mogą się powiększać i zajmować całą szerokość okna przeglądarki, w efekcie mogą w nich powstawać długie, mało czytelne wiersze tekstu. Z tego względu w niektórych układach tego typu jedynie część strony może się rozszerzać lub zwężyć — pozostałe elementy mają określona minimalną i maksymalną szerokość.

UKŁAD O STAŁEJ SZEROKOŚCI

W celu utworzenia układu o stałej szerokości należy określić (zazwyczaj w pikselach) szerokość głównego elementu strony (czasami określana jest także jego wysokość).

W przedstawionym przykładzie zastosowanych zostało kilka elementów `<div>`, w których, w celu określenia ich przeznaczenia, podane zostały wartości właściwości `id` oraz `class`.

W książce takiej jak ta zarówno układy o stałej szerokości, jak i układy elastyczne wyglądają podobnie. Aby się przekonać, jak one faktycznie działają, trzeba je wyświetlić w przeglądarce i sprawdzić, jak reagują na zmiany wielkości jej okna.

Układ o stałej szerokości zawsze będzie miał tę samą szerokość, niezależnie od wymiarów okna przeglądarki, natomiast układ elastyczny będzie się do nich dostosowywać, rozszerzając się lub kurcząc.

Kod HTML przykładowego układu o stałej szerokości przedstawionego na tej stronie oraz układu elastycznego przedstawionego nieco dalej jest dokładnie taki sam.

rodzial-15/fixed-width-layout.html

HTML

```
<body>
  <div id="header">
    <h1>Logo</h1>
    <div id="nav">
      <ul>
        <li><a href="">Strona główna</a></li>
        <li><a href="">Produkty</a></li>
        <li><a href="">Usługi</a></li>
        <li><a href="">O nas</a></li>
        <li><a href="">Kontakt</a></li>
      </ul>
    </div>
  </div>
  <div id="content">
    <div id="feature">
      <p>Opis</p>
    </div>
    <div class="article column1">
      <p>Kolumna pierwsza</p>
    </div>
    <div class="article column2">
      <p>Kolumna druga</p>
    </div>
    <div class="article column3">
      <p>Kolumna trzecia</p>
    </div>
  </div>
  <div id="footer">
    <p>&copy; Wszelkie prawa zastrzeżone.
      Utworzono w 2011 r.</p>
  </div>
</body>
```

CSS

```
body {  
    width: 960px;  
    margin: 0 auto;}  
#content {  
    overflow: auto;  
    height: 100%;}  
#nav, #feature, #footer {  
    background-color: #efefef;  
    padding: 10px;  
    margin: 10px;}  
.column1, .column2, .column3 {  
    background-color: #efefef;  
    width: 300px;  
    float: left;  
    margin: 10px;}  
li {  
    display: inline;  
    padding: 5px;}
```

Reguła odnosząca się do elementu `<body>` określa jego szerokość na 960 pikseli, a jednocześnie powoduje wyśrodkowanie zawartości strony, przypisując prawemu i lewemu marginesowi wartość `auto`.

Główne elementy strony mają marginesy o wielkości 10 pikseli, tak by były wyświetlane w pewnej odległości od siebie.

Panel nawigacyjny (`#nav`), opis (`#feature`) oraz stopka (`#footer`) rozszerzają się, dostosowując się do wielkości elementu zawierającego (którym w tym przypadku jest element `<body>`), dzięki czemu nie trzeba określać ich szerokości.

Wszystkie kolumny mają szerokość 300 pikseli i korzystają z właściwości `float`, dzięki czemu są wyświetlane jedna przy drugiej.

Czasami do określenia szerokości strony używany jest inny element, a nie sam element `<body>`. Dzięki temu kolor tła okna przeglądarki może być inny od tła zawartości strony.

WYNIK



UKŁAD ELASTYCZNY

W układzie elastycznym szerokości poszczególnych elementów są określane przy użyciu wartości procentowych, dzięki czemu układ może się dostosowywać do zmian wielkości okna przeglądarki.

Podczas testowania tego przykładu w przeglądarce koniecznie należy spróbować zmieniać wielkość okna, a zwłaszcza jego szerokość.

rodzaj-15/liquid-layout.html

HTML

```
<body>
  <div id="header">
    <h1>Logo</h1>
    <div id="nav">
      <ul>
        <li><a href="">Strona główna</a></li>
        <li><a href="">Produkty</a></li>
        <li><a href="">Usługi</a></li>
        <li><a href="">O nas</a></li>
        <li><a href="">Kontakt</a></li>
      </ul>
    </div>
  </div>
  <div id="content">
    <div id="feature">
      <p>Opis</p>
    </div>
    <div class="article column1">
      <p>Kolumna pierwsza</p>
    </div>
    <div class="article column2">
      <p>Kolumna druga</p>
    </div>
    <div class="article column3">
      <p>Kolumna trzecia</p>
    </div>
  </div>
  <div id="footer">
    <p>&copy; Wszelkie prawa zastrzeżone.
      Utworzono w 2011 r.</p>
  </div>
</body>
```

CSS

```
body {  
    width: 90%;  
    margin: 0 auto;}  
#content {overflow: auto;}  
#nav, #feature, #footer {  
    margin: 1%;}  
.column1, .column2, .column3 {  
    width: 31.3%;  
    float: left;  
    margin: 1%;}  
.column3 {margin-right: 0%;}  
li {  
    display: inline;  
    padding: 0.5em;}  
#nav, #footer {  
    background-color: #efefef;  
    padding: 0.5em 0;}  
#feature, .article {  
    height: 10em;  
    margin-bottom: 1em;  
    background-color: #efefef;}
```

WYNIK

Logo

Strona główna Produkty Usługi O nas Kontakt

Opis

Kolumna pierwsza

Kolumna druga

Kolumna trzecia

© Wszelkie prawa zastrzeżone. Utworzono w 2011 r.

Reguła odnosząca się do elementu <body> określa szerokość strony na 90%, dzięki czemu pomiędzy jej zawartością oraz prawą i lewą krawędzią okna przeglądarki będą niewielkie odstępy.

Każda z trzech kolumn ma margines o szerokości 1% oraz szerokość 31.3%. W sumie będą one zajmować 99.9% szerokości elementu <body>, co oznacza, że w niektórych przeglądarkach prawa krawędź trzeciej kolumny może nie być idealnie wyrównana względem pozostałej zawartości strony.

Elementy <div> zawierające elementy nawigacyjne, opis oraz stopkę będą zajmować całą szerokość elementu <body>. Zastosowano w nich także margines o szerokości 1%, by były wyrównane z trzema kolumnami.

Jeśli spróbujemy sobie wyobrazić, że okno przeglądarki jest bardzo szerokie lub bardzo wąskie, przekonamy się, że wiersze tekstu mogą się stać bardzo długie lub bardzo krótkie.

To właśnie w takich sytuacjach mogą nam pomóc właściwości `min-width` oraz `max-width`, umożliwiające określenie granic, w jakich układ może się rozszerzać i zwężać (trzeba jednak pamiętać, że pierwszą wersją przeglądarki Internet Explorer obsługującej te właściwości był IE7).

SIATKI UKŁADÓW

W każdej ze sztuk wizualnych (takich jak projektowanie, malarstwo czy fotografia) kompozycja jest sposobem rozmieszczania lub porządkowania widocznych elementów — sposobem ich organizacji na stronie. Wielu twórców przy rozmieszczaniu poszczególnych elementów strony pomaga sobie, używając struktury o postaci siatki — dokładnie tak samo postępują projektanci stron WWW.

Na następnej stronie widać grube, pionowe paski nałożone na witrynę jednej z gazet, aby pokazać, że została zaprojektowana na podstawie siatki. Nosi ona nazwę **siatki o szerokości 960 pikseli** i jest powszechnie stosowana przez projektantów stron WWW.

Siatki określają spójne proporcje oraz odstępy pomiędzy elementami, co pozwala zapewnić stronom profesjonalny wygląd.

Jeśli dokładniej przyjrzymy się stronom tej książki, to także zauważymy, że została zaprojektowana na podstawie siatki (składającej się z trzech kolumn).

Jak się przekonamy na stronach 388 – 389, na bazie tej elastycznej siatki można stworzyć wiele różnych układów.

Choć mogłoby się wydawać, że siatka stanowi ograniczenie, w rzeczywistości daje ona wiele korzyści:

- Zapewnia spójność pomiędzy wieloma różnymi stronami, które mogą być tworzone na podstawie różnych projektów.
- Ułatwia użytkownikom przewidzenie, w których miejscach strony mogą poszukiwać informacji.
- Ułatwia dodawanie do strony nowych treści w spójny sposób.
- Ułatwia różnym osobom współpracę przy tworzeniu projektu w taki sposób, by sobie wzajemnie nie przeszkadzały, a uzyskiwane efekty miały spójny wygląd.

PRZYKŁADOWA SIATKA

Mobile site Sign in Register A Text larger - smaller About us ▾ Today's paper ▾ Zeitgeist ▾



guardian.co.uk

Your search terms... Culture Search

News | Sport | Comment | Culture | Business | Money | Life & style | Travel | Environment | TV | Blogs | Data | Mobile | Offers | Jobs

Culture | Art and design | Books | Film | Music | Stage | Cuts blog | Kids' books | TV & radio | Games | Blogs | Podcasts | Video

culture

Webfeed

Editors' picks

Cannes 2011



Music



Film Weekly from Cannes

Come to our new band show!

Books



Photography



Stage



TV and radio



Live webchat
An agent talks



On this site

Art and design
Art
Architecture
Design
Photography
Your photographs
Jonathan Jones on art
Books
News
Reviews
Books genres
Books blog
Ellen

Podcast: Jason Solomons meets Cannes festival president Giles Jacob, Palme d'Or juror Mahamat-Saleh Haroun, electro-duo Air and our own Joe Cornish

[Post your comment](#)

[Video: Space milkshake anyone?](#)
[Tilda Swinton: 'It's bloody having a child'](#)
[Diary: Will I ever leave Hotel Cannes-ifornia?](#)
[Full coverage: Cannes](#)

In pictures: Ng Han Guan

In pictures: Ng Han Guan is a Singaporean photographer based in Beijing. In this set of photographs he captures various facets of daily life

Pinter in a cell

Video: Can you make five plays by Harold Pinter into a piece of immersive theatre? Andrew Dickson found out

Last night's TV

Review: Inside the Human Body did more to make death less scary than any theologian

[21 comments](#)

[Your next box set](#)

Sci-fi is no place for dabblers

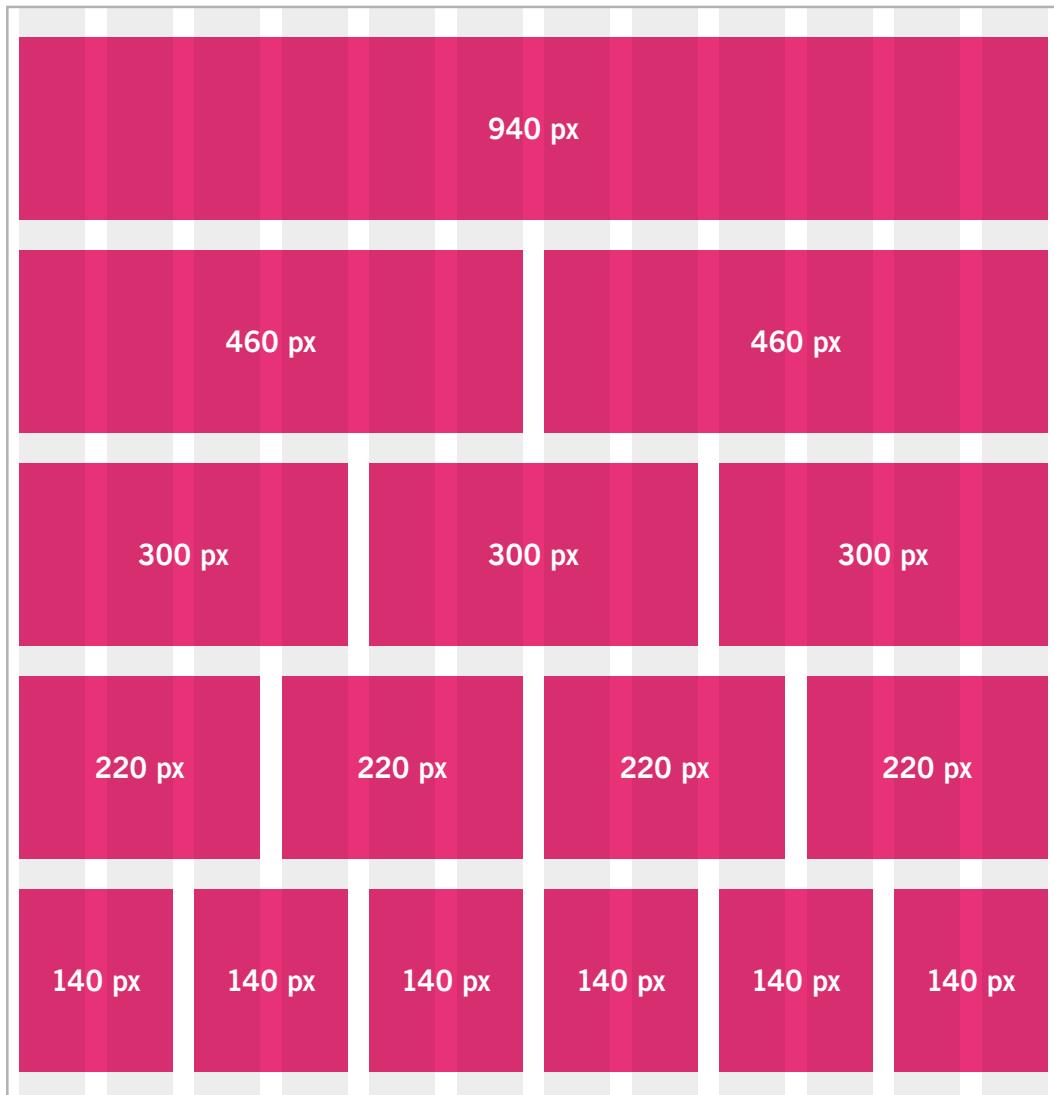
'Write about what you love, rather than what you have contempt for', says Iain M Banks

[105 comments](#)

Children's books
Mini heroes



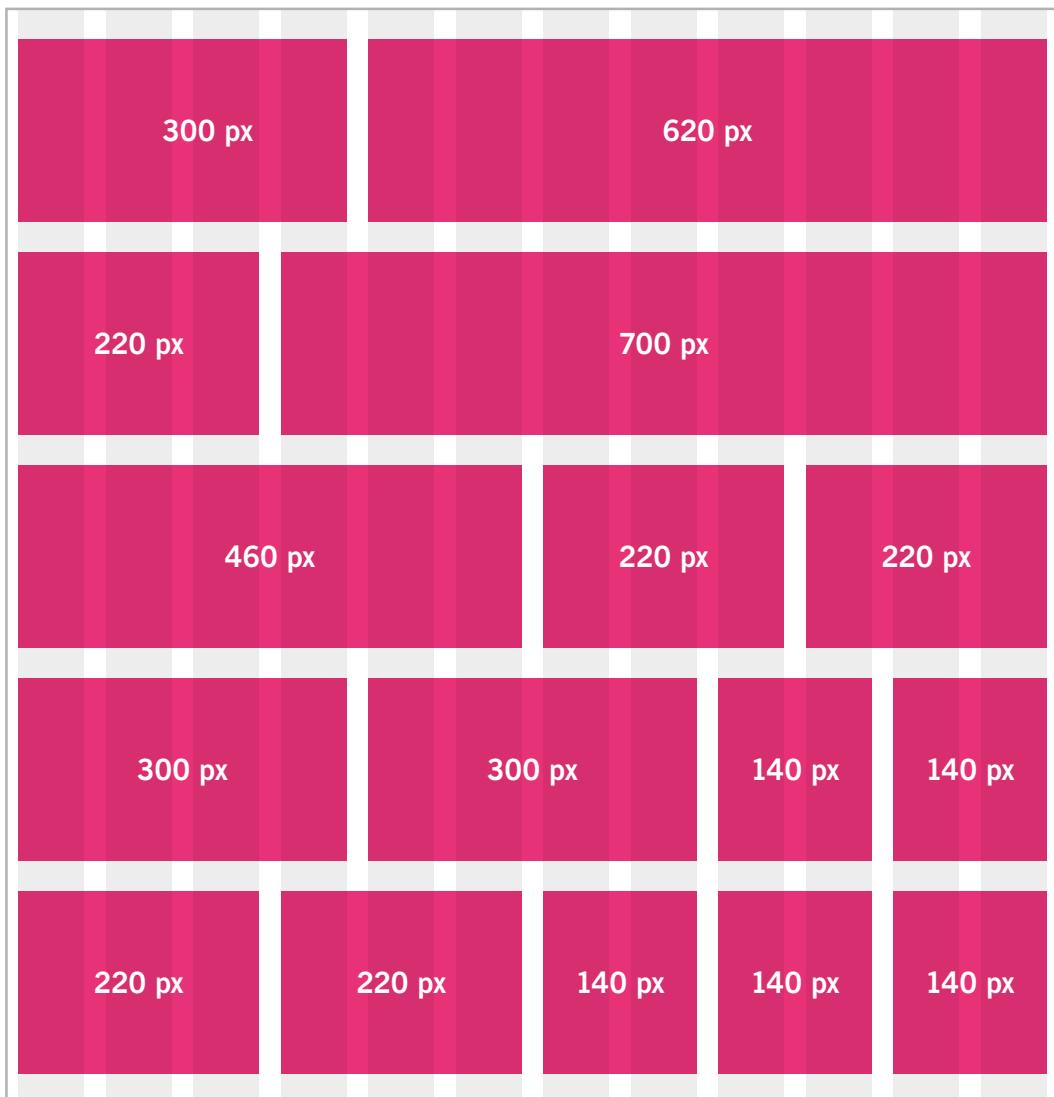
MOŻLIWE UKŁADY: 960 PIKSELI SZEROKOŚCI SIATKA 12-KOLUMNOWA



Na tych dwóch stronach przedstawiono siatkę o szerokości 960 pikseli, składającą się z 12 kolumn. Przykłady te pokazują, jak stworzyć wiele różnych układów na podstawie tej siatki.

Strona ma 960 pikseli szerokości i została podzielona na 12 identycznych kolumn (zaznaczonych szarym kolorem), z których każda ma 60 pikseli szerokości.

Każda kolumna ma marginesy o wielkości 10 pikseli, co sprawia, że pomiędzy sąsiednimi kolumnami jest odstęp o szerokości 20 pikseli, a pomiędzy skrajnymi kolumnami i krawędziami strony — odstęp o szerokości 10 pikseli.



PLATFORMY CSS

Platformy CSS mają za zadanie ułatwiać życie twórcom stron WWW poprzez dostarczanie kodu służącego do wykonywania powtarzających się czynności, takich jak tworzenie układów kolumnowych, określanie postaci formularzy, tworzenie wersji stron nadających się do druku itd. Kod platformy CSS można dołączyć do swojego projektu, zamiast samodzielnie pisać w całości niezbędne arkusze stylów.

ZALETY

- Eliminują konieczność ciągłego pisania kodu realizującego te same cele.
- Są dobrze przetestowane w różnych wersjach przeglądarek (co pozwala omijać błędy występujące w przeglądarkach).

WADY

- Często wymagają umieszczania w kodzie HTML nazw klas związanych wyłącznie ze sposobem prezentacji strony (a nie z opisem jej zawartości).

- W celu zaspokojenia szerokiej gamy potrzeb takie platformy często zawierają więcej kodu, niż potrzeba dla danej witryny (często określa się je jako „napuchnięte”).

PLATFORMA CSS 960.GS

Jednym z najpopularniejszych zastosowań platform CSS jest prezentowanie zawartości stron w formie układów kolumnowych. Dostępnych jest kilka takich platform — w tej książce przedstawiona zostanie platforma o nazwie 960 Grid System (która można znaleźć na stronie <http://www.960.gs>).

Udostępnia ona arkusz stylów, który można dołączać do tworzonych stron WWW. Po jego dołączeniu w kodzie HTML strony należy zastosować odpowiednie nazwy klas, a one stworzą układ kolumnowy. Witryna 960.gs udostępnia także szablony, które można pobrać i wykorzystać

do zaprojektowania stron na podstawie układu składającego się z 12 kolumn. (Dodatkowo dostępna jest inna wersja siatki złożona z 16 kolumn).

Aby stworzyć układ składający się z 12 kolumn, główny element zawierający całą treść strony musi zawierać atrybut `class` o wartości `container_12`. Klasa ta nadaje elementowi szerokość 960 pikseli i informuje, że będzie używana siatka składająca się z 12 kolumn.

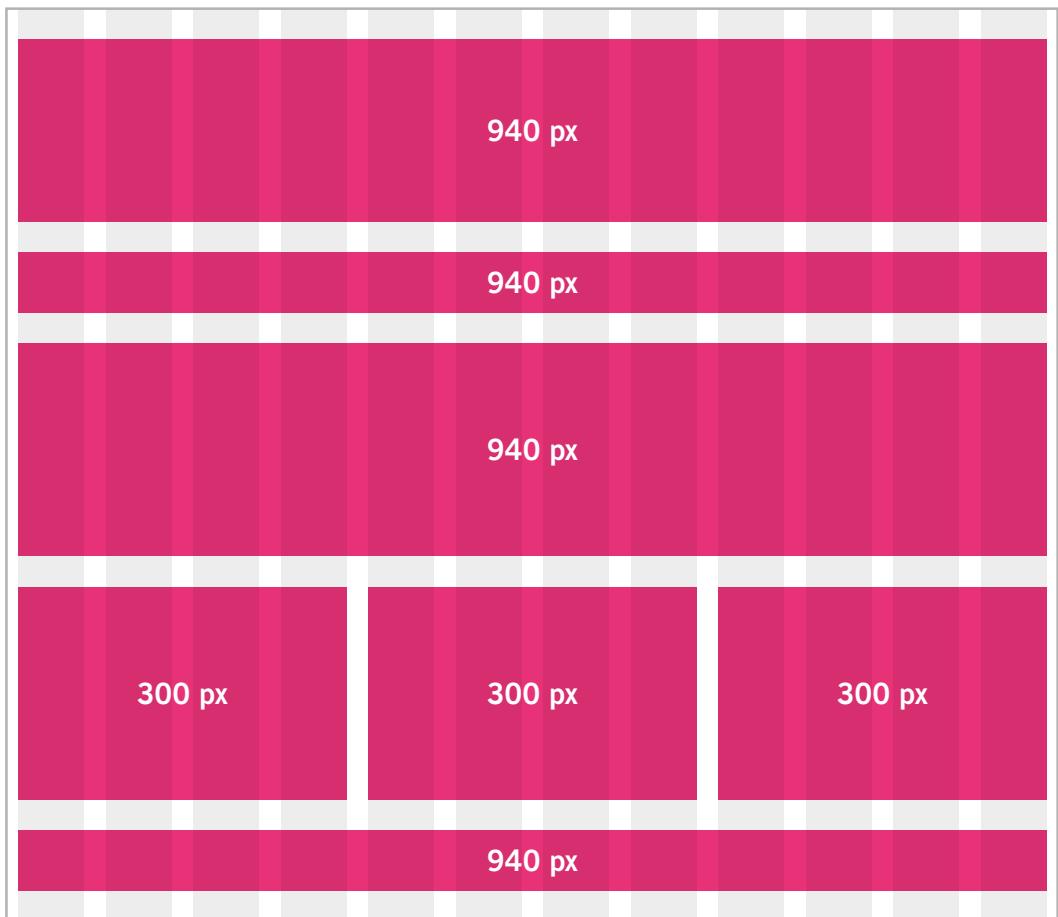
Platforma udostępnia także klasy służące do tworzenia bloków o szerokości od 1 do 12 kolumn.

Każdy z takich bloków tworzy się, używając odpowiedniej nazwy klasy, np. `grid_3` (blok o szerokości 3 kolumn) lub `grid_4` (blok o szerokości 4 kolumn) — aż do `grid_12` (blok zajmujący pełną szerokość strony). Wszystkie kolumny są elementami pływającymi, wyświetlanymi przy lewej krawędzi elementu zawierającego i mają lewy oraz prawy margines o szerokości 10 pikseli.

W internecie można znaleźć inne platformy CSS; oto kilka przykładów: blueprintcss.org, lessframework.com, developer.yahoo.com/yui/grids.

KORZYSTANIE Z PLATFORMY CSS 960.GS

Poniżej przedstawiono przykładowy układ strony, bardzo przypominający układ o stałej szerokości zamieszczony we wcześniejszej części rozdziału. Na następnej stronie utworzymy taki układ, korzystając z arkusza stylów 960.gs. Zamiast pisać własny arkusz stylów, będziemy musieli dodać do kodu HTML nazwy klas określające szerokość poszczególnych elementów strony.



UKŁAD SIATKI KORZYSTAJĄCY Z PLATFORMY CSS 960.GS

Przeanalizujmy teraz kod HTML strony i przekonajmy się, w jaki sposób został on dostosowany do korzystania z platformy CSS 960.GS.

Jak widać, arkusz stylów CSS został dołączony do strony przy użyciu elementu `<link>` umieszczonego wewnątrz elementu `<head>`.

Style, które musieliśmy napisać sami, zostały przedstawione na następnej stronie.

Arkusz stylów `960_12_col.css` zawiera wszystkie reguły niezbędne do kontroli i określania postaci układu. W kodzie HTML zostały użyte następujące nazwy klas:

`container_12`, która pełni funkcję pojemnika zawierającego całą treść strony i jednocześnie informuje, że element zajmuje pełne 12 kolumn.

`clearfix`, która zapewnia, że przeglądarka będzie znać wysokość głównego elementu zawierającego całą treść strony, co jest niezbędne, gdyż wewnątrz niego umieszczone są wyłącznie elementy płynące (rozwiązuje to problem przedstawiony na stronach 370 – 371).

`grid_12`, służąca do tworzenia bloku o szerokości 12 kolumn. `grid_4`, służąca do tworzenia bloku o szerokości 4 kolumn.

rodzaj-15/grid-layout.html

HTML

```
<head>
  <title>Układ siatki</title>
  <link rel="stylesheet" type="text/css"
  href="css/960_12_col.css" />
  <style> patrz następna strona książki </style>
</head>
<body>
  <div class="container_12 clearfix">
    <div id="header" class="grid_12">
      <h1>Logo</h1>
      <div id="nav">
        <ul>
          <li><a href="">Strona główna</a></li>
          <li><a href="">Produkty</a></li>
          <li><a href="">Usługi</a></li>
          <li><a href="">O nas</a></li>
          <li><a href="">Kontakt</a></li>
        </ul>
      </div>
    </div>
    <div id="feature" class="grid_12">
      <p>Opis</p>
    </div>
    <div class="article grid_4">
      <p>Kolumna pierwsza</p>
    </div>
    <div class="article grid_4">
      <p>Kolumna druga</p>
    </div>
    <div class="article grid_4">
      <p>Kolumna trzecia</p>
    </div>
    <div id="footer" class="grid_12">
      <p>© Wszelkie prawa zastrzeżone.
      Utworzono w 2011 r.</p>
    </div>
  </div><!-- .container_12 -->
</body>
```

CSS

```
* {  
    font-family: Arial, Verdana, sans-serif;  
    color: #665544;  
    text-align: center;}  
#nav, #feature, .article, #footer {  
    background-color: #efefef;  
    margin-top: 20px;  
    padding: 10px 0px 5px 0px;}  
#feature, .article {  
    height: 100px;}  
li {  
    display: inline;  
    padding: 5px;}
```

WYNIK



Arkusz stylów 960.gs zadbał o zapewnienie odpowiedniego układu, określenie właściwych szerokości kolumn oraz odstępów pomiędzy nimi. Dlatego reguły przedstawione obok są jedynymi, jakie musimy dodać do kodu strony. Ich zadaniem jest:

- Wybór odpowiednich czcionek oraz rozmieszczenie tekstów wewnątrz pudełek elementów.
- Określenie odpowiedniego koloru tła elementów.
- Określenie wysokości elementu z atrybutem id o wartości feature oraz elementów przypisanych do klasy article.
- Dodanie marginesu u góry oraz u dołu pudełek wybranych elementów.

STOSOWANIE WIELU ARKUSZY STYLÓW

@import

Niektórzy autorzy stron rozdzielają używane reguły i zapisują je w kilku osobnych arkuszach stylów. Jeden z nich może np. zawierać reguły odpowiadające za kontrolę układu, a drugi — reguły odpowiadające za wybór odpowiednich czcionek, kolorów itd.

Są także autorzy, którzy wykorzystują jeszcze bardziej **modularne** podejście i tworzą osobne arkusze stylów określające typografię, układ, postać formularzy, tabele, a nawet osobne style dla poszczególnych podsekcji witryny.

Istnieją dwa sposoby używania na stronie wielu arkuszy stylów:

1. Do strony można dołączyć jeden arkusz stylów, a wewnątrz niego umieścić reguły @import importujące wszystkie pozostałe niezbędne arkusze.
2. W kodzie strony można umieścić elementy <link> dotyczące wszystkie niezbędne arkusze stylów.

Przedstawiony tu przykład używa jednego elementu <link>, by dołączyć do strony arkusz stylów o nazwie *styles.css*. Znajdują się w nim reguły @import importujące dwa kolejne arkusze stylów: *typography.css* i *tables.css*.

rozdział-15/multiple-style-sheets-import.html

HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Stosowanie wielu arkuszy stylów - import</title>
    <link rel="stylesheet" type="text/css" href="css/styles.css" />
  </head>
  <body>
    <!-- Tu jest umieszczona zawartość strony -->
  </body>
</html>
```

rozdział-15/styles.css

CSS

```
@import url("tables.css");
@import url("typography.css");
body {
  color: #666666;
  background-color: #f8f8f8;
  text-align: center;}
#page {
  width: 600px;
  text-align: left;
  margin-left: auto;
  margin-right: auto;
  border: 1px solid #d6d6d6;
  padding: 20px;}
h3 {
  color: #547ca0;}
```

Jeśli w arkuszu używane są reguły @import, to należy je zapisać przed wszelkimi innymi regułami.

STOSOWANIE WIELU ARKUSZY STYLÓW

link

HTML

rodzaj-15/multiple-style-sheets-link.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Stosowanie wielu arkuszy stylów - link</title>
    <link rel="stylesheet" type="text/css"
          href="css/site.css" />
    <link rel="stylesheet" type="text/css"
          href="css/tables.css" />
    <link rel="stylesheet" type="text/css"
          href="css/typography.css" />
  </head>
  <body>
    <!-- Tu jest umieszczona zawartość strony -->
  </body>
</html>
```

WYNIK

Park Kultury i Wypoczynku - Wypożyczalnia rowerów

Pozycz rower, by pojeździć po Parku Kultury i Wypoczynku

	Na godzinę	Na cały dzień
Turystyczny	5 zł	40 zł
21 przerzutek	9 zł	55 zł

GDZIE I KIEDY obok Stadionu Śląskiego

Wypożyczalnia jest czynna od kwietnia do listopada, w godzinach od 9:00 do 19:00

ROZLICZENIA I KAUCJA

Akceptujemy płatność gotówką oraz kartą kredytową

Za rower pobieramy depozyt wysokości 600 zł.

Na tej stronie przedstawiono inną technikę pozwalającą na używanie na stronie kilku arkuszy stylów.

Wewnętrz elementu `<head>` umieszczanych jest kilka elementów `<link>`, po jednym dla każdego używanego arkusza.

Zawartość arkusza `site.css` jest taka sama jak przedstawionego na poprzedniej stronie arkusza `styles.css` z tą różnicą, że nie zawiera reguły `@import`.

Jak zawsze w przypadku CSS, jeśli do danego elementu odnoszą się dwie reguły, ta umieszczona bliżej końca arkusza będzie miała wyższy priorytet.

W tym przykładzie wszystkie reguły umieszczone w arkuszu `typography.css` będą miały wyższy priorytet od reguł zapisanych w arkuszu `site.css` (gdyż zostaną dodane do dokumentu jako ostatnie).

W przykładzie zamieszczonym na poprzedniej stronie wyższy priorytet będą miały reguły umieszczone w arkuszu `styles.css`, a nie `typography.css`. Wynika to z faktu, że w przypadku korzystania z reguły `@import` właśnie miejsce jej występowania jest uznawane przez przeglądarkę za to, w którym są umieszczone reguły CSS.

PEDAL FASTER.
THE MODERN BICYCLE MAGAZINE

BEZ PRZERZUTEK

10 kwietnia 2011

Veloheld łączy w sobie minimalizm z najwyższą jakością. Pozbawione niepotrzebnej grafiki i osprzętu przerzutek produkty z serii gamy Veloheld zachwycają pięknem i prostotą. Czarno-białe (nawiązanie do yin yang?) rowery cedzą się niewielkim rozstawem osi, brakiem przerzutek i stosunkowo wąskimi kierowcami.

Domorodli konstruktorzy mogą skorzystać z tzw. veloheld - są one dostępne w trzech rozmiarach i dwóch kolorach i stanowią doskonały materiał do rozpoczęcia konstrukcji.

[Czytaj więcej...](#)

WIĘCEJ ARTYKUŁÓW

Na drodze: Młodzik jazdy bez przerzutek

Historia Randa: Kowery Paulley - zbiadłości w Anglii

Dzielenie się: Stolikownik w koszyku

Rekreacja rowerowa: Nie wypakuj koszyka na plażę

Ber lejemy: Wywał z twojej mazurki

Od czekoladek: Podziel się swoimi przewodnictwami!

Wojny rąk: Innowacje w produkcji i

YouTube: Najlepsze wideo rowerowe na

Gang lekarski: Dwudziesta skromnego

PRZYKŁAD

UKŁAD



Zamieszczony tu przykład przedstawia nowoczesny układ stosowany często w witrynach czasopism i magazynów, stworzony przy użyciu platformy CSS 960.GS. Wykorzystanie tego arkusza stylów pozwala uniknąć konieczności samodzielnego pisania reguł CSS.

Kilka nazw klas z arkusza 960.gs zostało dodanych do kodu strony, by określić, ile kolumn układu mają zajmować jej poszczególne elementy. Wiemy już, że arkusz ten używa właściwości `float`, by umieszczać kolejne bloki obok siebie.

Nagłówek znajdujący się na samym początku strony korzysta z umiejscawiania ustalonego (co oznacza, że jego położenie nie będzie się zmieniać, gdy użytkownik zacznie przewijać zawartość strony). Dodano do niego także właściwość `z-index`, aby był wyświetlany powyżej pozostały zawartości.

Zarówno nagłówek, jak i stopka zostały umieszczone wewnętrz elementów `<div>` zajmujących całą dostępną szerokość strony. Wewnętrz nich umieszczono inne elementy, które korzystają z klasy arkusza 960.gs, by zapewnić, że będą one prawidłowo wyrównane względem pozostały zawartości strony.

Główny artykuł zajmuje całą szerokość strony. Klasy `push_7` oraz `push_9` są zdefiniowane w arkuszu 960.gs i służą do przesunięcia nagłówka oraz treści artykułu w prawo.

Poniżej głównego artykułu widoczne są cztery bloki, z których każdy ma 3 kolumny szerokości. Każdy z tych bloków zawiera obraz oraz łącze prowadzące do kolejnych artykułów.

W przykładzie wykorzystano także obrazy tła, przy których użyciu zostały utworzone graficzne tła strony i jej nagłówka, jak również obrazy wyświetlane w treści artykułu. Zagadnienia związane ze stosowaniem obrazów zostały opisane w następnym rozdziale.

Uwaga: jeśli strona zostanie wyświetlona w przeglądarce Internet Explorer 6, to używane na niej przezroczyste obrazy PNG mogą mieć szare tło. Aby dowiedzieć się więcej na ten temat, warto zajrzeć na stronę poświęconą tej książce — zamieściliśmy tam prosty skrypt JavaScript stanowiący rozwiązanie tego problemu.

PRZYKŁAD

UKŁAD

```
<!DOCTYPE html>
<html>
<head>
<title>Układ</title>
<link rel="stylesheet" type="text/css" href="css/960_12_col.css" />
<style type="text/css">
@font-face {
    font-family: 'QuicksandBook';
    src: url('fonts/Quicksand_Book-webfont.eot');
    src: url('fonts/Quicksand_Book-webfont.eot?#iefix') format('embedded-opentype'),
        url('fonts/Quicksand_Book-webfont.woff') format('woff'),
        url('fonts/Quicksand_Book-webfont.ttf') format('truetype'),
        url('fonts/Quicksand_Book-webfont.svg#QuicksandBook') format('svg');
    font-weight: normal;
    font-style: normal;}
body {
    color: #ffffff;
    background: #413f3b url("images/bg.jpg");
    font-family: Georgia, "Times New Roman", Times, serif;
    font-size: 90%;
    margin: 0px;
    text-align: center;}
a {
    color: #b5c1ad;
    text-decoration: none;}
a:hover {
    color: #ffffff;}
.header {
    background-image: url("images/bg-header.jpg");
    padding: 0px 0px 0px 0px;
    height: 100px;
    position: fixed;
    top: 0px;
    width: 100%;
    z-index: 50;}
.nav {
    float: right;
    font-family: QuicksandBook, Helvetica, Arial, sans-serif;
```

PRZYKŁAD

UKŁAD

```
padding: 45px 0px 0px 0px;
text-align: right;}
.wrapper {
width: 960px;
margin: 0px auto;
background-image: url("images/bg-triangle.png");
background-repeat: no-repeat;
background-position: 0px 0px;
text-align: left;}
.logo {
margin-bottom: 20px;}
h1, h2 {
font-family: QuicksandBook, Helvetica, Arial, sans-serif;
font-weight: normal;
text-transform: uppercase;}
h1 {
font-size: 240%;
margin-top: 140px;}
.date {
font-family: Arial, Helvetica, sans-serif;
font-size: 75%;
color: #b5c1ad;}
.intro {
clear: left;
font-size: 90%;
line-height: 1.4em;}
.main-story {
background-image: url("images/triangles.png");
background-repeat: no-repeat;
background-position: 122px 142px;
height: 570px;}
.more-articles {
border-top: 1px solid #ffffff;
padding: 10px;}
.more-articles p {
border-bottom: 1px solid #807c72;
padding: 5px 0px 15px 0px;
font-size: 80%;}
```

PRZYKŁAD

UKŁAD

```
.more-articles p:last-child {  
    border-bottom: none;  
}  
.footer {  
    clear: both;  
    background: rgba(0, 0, 0, 0.2);  
    padding: 5px 10px;  
}  
.footer p {  
    font-family: Helvetica, Arial, sans-serif;  
    font-size: 75%;  
    text-align: right;  
}  
.footer a {  
    color: #807c72;  
}  
</style>  
</head>  
<body>  
    <div class="header">  
        <div class="container_12">  
            <div class="grid_5">  
                  
                  
            </div>  
            <div class="nav grid_7">  
                <a href="">strona główna</a> / <a href="">nowości</a> /  
                <a href="">archiwa</a> / <a href="">o nas</a> / <a href="">kontakt</a>  
            </div>  
        </div>  
    </div>  
    <div class="wrapper">  
        <div class="main-story container_12">  
            <div class="grid_6 push_6">  
                <h1><a href="">Bez przerzutek</a></h1>  
            </div>  
            <div class="intro grid_3 push_9">  
                <p class="date">16 kwietnia 2011</p>  
                <p>Veloheld łączy w sobie minimalizm z najwyższą jakością.  
                    Pozbawione niepotrzebnej grafiki oraz osprzętu przerzutek produkty  
                    z szerokiej gamy Veloheld zachwycają pięknem i prostotą...</p>  
            </div>  
        </div>  
    </div>  
</body>
```

PRZYKŁAD

UKŁAD

```
</div>
</div><!-- .main-story -->
<div class="more-articles container_12">
  <h2 class="grid_12"><a href="">Więcej artykułów</a></h2>
  <div class="grid_3">
    
    <p><a href="">Na drodze: Miłośnik jazdy bez przerzutek</a></p>
    <p><a href="">Historia Banda: Rowery Pashley - zbudowane w Anglii</a></p>
    <p><a href="">Wojny ram: Innowacje w produkcji i naprawie rowerów</a></p>
  </div>
  <div class="grid_3">
    
    <p><a href="">Dziennik osi: Szkicownik w koszyku</a></p>
    <p><a href="">Dziesięć najważniejszych nowości 2012 roku: Rzut oka na to, co nas czeka</a></p>
    <p><a href="">InnerTube: Najlepsze wideo rowerowe na świecie</a></p>
  </div>
  <div class="grid_3">
    
    <p><a href="">Recenzje rowerów: Nie wszystkie koszyki są identyczne</a></p>
    <p><a href="">Bez ściemy: Wywiad z twórcą znanej witryny poświęconej rowerom</a></p>
    <p><a href="">Front obrony rowerzystów: Znaj swoje prawa na drodze</a></p>
  </div>
  <div class="grid_3">
    
    <p><a href="">Sala sławy rowerowej: Schwinn Spitfire</a></p>
    <p><a href="">Od czytelników: Podziel się swoimi przemyśleniami!</a></p>
    <p><a href="">Gang łańcucha: Ewolucja skromnego łańcucha rowerowego</a></p>
  </div>
</div><!-- .more-articles -->
</div><!-- .wrapper -->
<div class="footer clearfix">
  <div class="container_12">
    <p class="grid_12"><a href="">Informacje prawne</a> | <a href="">Polityka prywatności</a> | <a href="">Wszelkie prawa zastrzeżone ©; Pedal Faster 2011</a></p>
  </div>
</div>
</body>
</html>
```


PODSUMOWANIE

UKŁAD

- ▶ Elementy <div> często są stosowane jako elementy zawierające, służące do grupowania innych elementów stron.
- ▶ Przeglądarki wyświetlają strony, korzystając z rozkładu normalnego, chyba że ich twórca zażąda użycia umiejscawiania względnego, bezwzględnego lub ustalonego.
- ▶ Właściwość float przenosi zawartość strony do jej prawej lub lewej krawędzi i może być stosowana do tworzenia układów składających się z wielu kolumn. (Takie płynące elementy muszą mieć określoną szerokość).
- ▶ Układy stron mogą być elastyczne lub mieć ustaloną szerokość.
- ▶ Projektanci nadają stronom szerokość z zakresu od 960 do 1000 pikseli i na pierwszych 600 pikselach jej wysokości starają się przekazać informacje o jej zawartości (by pokazać ich wartość bez konieczności przewijania).
- ▶ Siatki pomagają tworzyć profesjonalnie wyglądające i elastyczne układy stron.
- ▶ Platformy CSS zawierają reguły, które realizują najczęściej występujące zadania.
- ▶ Na jednej stronie można używać wielu arkuszy stylów CSS.

16

OBRAZY

- ▶ Kontrola wielkości obrazów przy użyciu CSS.
- ▶ Wyrównywanie obrazów przy użyciu CSS.
- ▶ Dodawanie obrazów tła.

Kontrola rozmiaru oraz wyrównania obrazów przy użyciu CSS sprawia, że wszystkie reguły określające postać strony są umieszczone w arkuszu stylów poza kodem HTML.

Korzystając z obrazów tła, można uzyskać kilka interesujących efektów. W tym rozdziale powiemy, jak:

- Określać rozmiary i wyrównanie obrazów przy użyciu CSS.
- Dodawać do elementów obrazy tła.
- Tworzyć efekt podmieniania obrazów z użyciem CSS.



KONTROLA WIELKOŚCI OBRAZÓW PRZY UŻYCIU CSS

Wymiary obrazów można określić przy użyciu właściwości CSS width oraz height, analogicznie jak wielkość pudełek wszystkich innych elementów.

Określenie wymiarów obrazów umożliwia szybsze i bardziej płynne wyświetlenia stron, gdyż ich kod HTML i CSS jest zazwyczaj wczytywany przed obrazami, a wiedząc, jakie są ich wymiary, przeglądarka może wyświetlić pozostałą zawartość strony bez oczekiwania na pobranie grafiki.

Można sądzić, że nasze witryny będą korzystały z obrazów o przeróżnych wymiarach, jednak w wielu witrynach obrazy o tej samej wielkości są używane na wielu stronach.

Przykładowo zdjęcia produktów prezentowane w witrynach zajmujących się handlem elektronicznym zazwyczaj mają tę samą wielkość. Jeśli nasza witryna została zaprojektowana na bazie siatki, to na jej stronach możemy przygotowywać i prezentować zdjęcia w kilku często stosowanych wymiarach, takich jak:

rodzial-16/image-sizes.html

HTML

```
  
  

```

```
img.large {  
    width: 500px;  
    height: 500px;}  
img.medium {  
    width: 250px;  
    height: 250px;}  
img.small {  
    width: 100px;  
    height: 100px;}
```

CSS

małe zdjęcie w układzie pionowym: 220×360, małe zdjęcie w układzie poziomym: 330×210, duże zdjęcie do artykułu: 620×400.

Jeśli w witrynie konsekwentnie będą stosowane obrazy o pewnych wymiarach, to można je określić w arkuszach stylów CSS, a nie w kodzie HTML.

WYNIK



W pierwszej kolejności należy określić, jakie wymiary obrazów będą najczęściej stosowane w witrynie. Następnie trzeba nadać im nazwy.

Oto przykłady:
small
medium
large

W kodzie strony, w elementach , zamiast atrybutów width oraz height należy użyć atrybutu class, podając jako jego wartość jedną z tych nazw.

W arkuszu stylów CSS dla każdej z tych klas należy podać selektor, uzupełniając regułę o właściwości width i height określające wymiary obrazu.

WYRÓWNYWANIE OBRAZÓW PRZY UŻYCIU CSS

W poprzednim rozdziale była mowa o tym, w jaki sposób, używając właściwości `float`, spowodować, by element został wyświetlony przy lewej lub prawej krawędzi jego elementu zawierającego, a umieszczony obok tekstu go otaczał.

Obecnie, zamiast stosować atrybut `align` elementu ``, twórcy stron coraz częściej wyrównują obrazy, korzystając z właściwości `float`. Powszechnie stosowane są dwa sposoby takiego wyrównywania obrazów:

1. Właściwość `float` zostaje dodana do klasy stworzonej w celu określenia wymiarów obrazu (takiej jak `small` z przedstawionego przykładu).
2. Tworzone są nowe klasy — o takich nazwach jak `align-left` lub `align-right` — których zadaniem jest wyrównanie elementu do lewej lub prawej krawędzi strony.

W przedstawionym obok przykładzie do wyrównania obrazów zostały zastosowane dwie klasy: `align-left` i `align-right`.

Zazwyczaj do obrazów jest dodawany margines, by otaczający je tekst nie stykał się z ich krawędziami.

rodzial-16/aligning-images.html

HTML

```
<p><b><i>Magnoliowe</i></b> to duża rodzina roślin, do której należy ponad 200 gatunków. Jej nazwa pochodzi od nazwiska...</p>
<p>Niektóre magnolie, takie jak <i>Magnolia stellata</i> oraz <i>Magnolia soulangeana</i>, kwitną wczesną wiosną, jeszcze...</p>
```

```
img.align-left {
  float: left;
  margin-right: 10px;}
img.align-right {
  float: right;
  margin-left: 10px;}
img.medium {
  width: 250px;
  height: 250px;}
```



Magnoliowe to duża rodzina roślin, do której należy ponad 200 gatunków. Jej nazwa pochodzi od nazwiska francuskiego botanika Pierre'a Magnola. Ponieważ rośliny te rośły, zanim na Ziemi pojawiły się pączki, ich kwiaty wykształcały się w taki sposób, by były zapylane przez motyle.

Niektóre magnolie, takie jak *Magnolia stellata* oraz *Magnolia soulangeana*, kwitną wczesną wiosną, jeszcze przed wypuszczeniem liści. Z kolei inne, jak *Magnolia grandiflora*, kwitną późną wiosną lub nawet latem.



WYNIK

WYŚRODKOWYWANIE OBRAZÓW PRZY UŻYCIU CSS

HTML

rodzial-16/centering-images.html

```
<p><b><i>Magnoliowate</i></b> to duża rodzina roślin, do której należy ponad 200 gatunków. Jej nazwa pochodzi od nazwiska francuskiego botanika Pierre'a Magnola. Ponieważ rośliny te rosły, zanim na Ziemi pojawiły się pająki, ich kwiaty wykształciły się w taki sposób, by były zapylane przez motyle.</p>
```

CSS

```
img.align-center {  
  display: block;  
  margin: 0px auto;}  
img.medium {  
  width: 250px;  
  height: 250px;}
```

WYNIK



Magnoliowate to duża rodzina roślin, do której należy ponad 200 gatunków. Jej nazwa pochodzi od nazwiska francuskiego botanika Pierre'a Magnola. Ponieważ rośliny te rosły, zanim na Ziemi pojawiły się pająki, ich kwiaty wykształciły się w taki sposób, by były zapylane przez motyle.

Domyślnie obrazy są elementami wewnętrz-wierszowymi. Oznacza to, że są one rozmieszczane wewnątrz otaczającego je tekstu. Aby wyśrodkować obraz, należy go przekształcić w element blokowy, a to można zrobić, używając właściwości `display` z wartością `block`.

Kiedy obraz zostanie już przekształcony na element blokowy, można go wyśrodkować w poziomie na dwa sposoby:

1. W elemencie zawierającym obraz można zastosować właściwość `text-align` z wartością `center`.
2. W samym elemencie obrazu można użyć właściwości `margin`, by przypisać prawemu i lewemu marginom wartości `auto`.

Można także stworzyć klasę umożliwiającą wyśrodkowanie dowolnego elementu i stosować ją w taki sam sposób jak przedstawione wcześniej klasy określające wymiary obrazów.

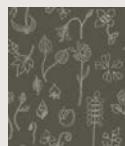
Prezentowane tu techniki określania wymiarów i wyrównania obrazów można stosować w elemencie `<figure>` wprowadzonym w HTML5 i przedstawionym we wcześniejszej części książki, na stronie 119.

OBRAZY TŁA

background-image

Właściwość `background-image` pozwala umieścić wskazany obraz w tle dowolnego elementu HTML. Może to być zarówno cała strona, jak i jej fragment. Domyslnie obraz tła będzie powtarzany, tak by wypełnić cały obszar pudełka elementu.

Ścieżka dostępu do obrazu jest podawana za literami `url` i należy ją zapisywać w nawiasie oraz cudzysłowie.



Oto obraz, którego będziemy używać w przedstawionym tu przykładzie.

W pierwszym przykładzie obraz tła zostanie określony dla całej strony (gdyż podamy go w selektorze odnoszącym się do elementu `<body>`). W drugim — obraz tła zostanie wyświetlony wyłącznie w jednym akapicie.

W internecie można znaleźć wiele witryn udostępniających obrazy tła, których można używać na własnych stronach.

Bardzo często obrazy tła są ostatnimi elementami stron pobieranymi przez przeglądarki (co może sprawiać wrażenie, że cała strona wolno się wczytuje). Trzeba pamiętać, że podobnie jak w przypadku wszystkich innych obrazów używanych na stronach WWW, jeśli pliki obrazów tła będą duże, to ich pobieranie potrwa.

`rozdział-16/background-image-body.html`

CSS

```
body {  
background-image: url("images/pattern.gif");}
```

WYNIK



`rozdział-16/background-image-element.html`

CSS

```
p {  
background-image: url("images/pattern.gif");}
```

WYNIK

Poradnik ogrodnika

Magnolia

Magnolia grandiflora, nazywana także magnolią południową, jest drzewem należącym do rodziny Magnoliaceae, którego naturalnym miejscem występowania są południowe rejony USA - od wybrzeży Wirginii, poprzez Florydę na południu i zachód, po wschodnie stany: Teksas i Oklahoma. Magnolia ta osiąga 27,5 metra wysokości, jest okazałym, wiecznie zielonym drzewem ze sporymi, ciemnozielonymi liśćmi oraz dużymi, białymi, pachnącymi kwiatami. Roślina popularna, powszechnie uprawiana na całym świecie.

POWTARZANIE OBRAZU TŁA

background-repeat background-attachment

CSS

rodzial-16/background-repeat.html

```
body {  
background-image: url("images/header.gif");  
background-repeat: repeat-x;}
```

WYNIK



Magnolia

Magnolia grandiflora, nazywana także magnolią południową, jest drzewem należącym do rodziny Magnoliaceae, którego naturalnym miejscem występowania są południowe rejony USA - od wybrzeży Wirginii, poprzez Florydę na południu i zachód, po wschodnie stany: Teksas i Oklahomę. Magnolia ta osiąga 27,5 metra wysokości, jest okazałym, wiecznie zielonym drzewem ze sporymi, ciemnozielonymi liśćmi oraz dużymi, białymi, pachnącymi kwiatami. Roślina popularna, powszechnie uprawiana na całym świecie; wyhodowano też sporo odmian przeznaczonych do celów przemysłowych. Drewno magnolii jest twardze i ciężkie, używa się go do wytwarzania mebli, talerzy oraz oklein.

CSS

rodzial-16/background-attachment.html

```
body {  
background-image: url("images/tulip.gif");  
background-repeat: no-repeat;  
background-attachment: fixed;}
```

WYNIK



Magnolia

Magnolia grandiflora, nazywana także magnolią południową, jest drzewem należącym do rodziny Magnoliaceae, którego naturalnym miejscem występowania są południowe rejony USA - od wybrzeży Wirginii, poprzez Florydę na południu i zachód, po wschodnie stany: Teksas i Oklahomę. Magnolia ta osiąga 27,5 metra wysokości, jest okazałym, wiecznie zielonym drzewem ze sporymi, ciemnozielonymi liśćmi oraz dużymi, białymi, pachnącymi kwiatami. Roślina popularna, powszechnie uprawiana na całym świecie; wyhodowano też sporo odmian przeznaczonych do celów przemysłowych. Drewno magnolii jest twardze i ciężkie, używa się go do wytwarzania mebli, talerzy oraz oklein.

Właściwość `background-repeat` może przyjmować następujące wartości:

repeat

Obraz tła jest powtarzany zarówno w poziomie, jak i w pionie (to domyślny sposób wyświetlania obrazu tła, stosowany, jeśli nie została określona inna wartość właściwości `background-image`).

repeat-x

Obraz jest powtarzany tylko w poziomie (jak widać na pierwszym przykładzie zamieszczonym obok).

repeat-y

Obraz jest powtarzany tylko w pionie.

no-repeat

Obraz zostaje wyświetlony tylko raz. Właściwość `background-attachment` określa, czy obraz tła powinien pozostać w tym samym miejscu, czy ma się przesuwać, gdy użytkownik będzie przewijają stronę. Właściwość ta może przyjmować dwie wartości:

fixed

Obraz tła pozostaje w tym samym położeniu na stronie.

scroll

Gdy użytkownik będzie przewijają stronę, obraz tła będzie przesuwany w góre i w dół wraz z pozostałą jej zawartością.

POŁOŻENIE OBRAZU TŁA

background-position

Jeśli obraz tła nie jest powtarzany, to można użyć właściwości `background-position`, by określić, w którym miejscu okna przeglądarki należy go umieścić.

Właściwość ta zazwyczaj zawiera parę wartości — pierwsza z nich określa położenie obrazu tła w poziomie, a druga — w pionie.

-  `left top`
-  `left center`
-  `left bottom`
-  `center top`
-  `center center`
-  `center bottom`
-  `right top`
-  `right center`
-  `right bottom`

Jeśli zostanie podana tylko jedna wartość, to druga przyjmie wartość domyślną — `center`.

Używając tej właściwości, można także podać parę wartości wyrażonych w pikselach lub wartości procentowych. W takim przypadku określają one odległość od lewego górnego wierzchołka okna przeglądarki (lub elementu zawierającego). Lewy górny wierzchołek odpowiada wartościom procentowym `0% 0%`.

[rodzial-16/background-position.html](#)

CSS

```
body {  
background-image: url("images/tulip.gif");  
background-repeat: no-repeat;  
background-position: center top;}
```

WYNIK

Poradnik ogrodnika



Magnolia

Magnolia grandiflora, nazywana także magnolia południowa, jest drzewem należącym do rodziny Magnoliaceae, którego naturalnym miejscem występowania są południowe rejony USA - od wybrzeży Wirginii, poprzez Florydę na południu i zachód, po wschodnie stany: Teksas i Oklahoma. Magnolia ta osiąga 27,5 metra wysokości, jest okazałym, wiecznie zielonym drzewem ze sporymi, ciemnozielonymi liśćmi oraz dużymi, białymi, pachnącymi kwiatami. Roślina popularna, powszechnie uprawiana na całym świecie; wyhodowano też sporo odmian przeznaczonych do celów przemysłowych. Drewno magnolii jest twardie i cięzkie, używa się go do wytwarzania mebli, talerzy oraz olein.

[rodzial-16/background-position-percentage.html](#)

CSS

```
body {  
background-image: url("images/tulip.gif");  
background-repeat: no-repeat;  
background-position: 50% 50%;}
```

WYNIK

Poradnik ogrodnika



Magnolia

Magnolia grandiflora, nazywana także magnolia południowa, jest drzewem należącym do rodziny Magnoliaceae, którego naturalnym miejscem występowania są południowe rejony USA - od wybrzeży Wirginii, poprzez Florydę na południu i zachód, po wschodnie stany: Teksas i Oklahoma. Magnolia ta osiąga 27,5 metra wysokości, jest okazałym, wiecznie zielonym drzewem ze sporymi, ciemnozielonymi liśćmi oraz dużymi, białymi, pachnącymi kwiatami. Roślina popularna, powszechnie uprawiana na całym świecie; wyhodowano też sporo odmian przeznaczonych do celów przemysłowych. Drewno magnolii jest twardie i cięzkie, używa się go do wytwarzania mebli, talerzy oraz olein.

Jaskier

Ranunculus asiaticus (jaskier azjatycki) jest rodzajem jaskra (*Ranunculus*) pochodzącego ze wschodnich

W przedstawionym obok przykładzie wykorzystanie wartości `50% 50%` powoduje

wyśrodkowanie obrazu tła zarówno w poziomie, jak i w pionie.

ZAPIS SKRÓTOWY

background

CSS

[rodzial-16/background-shorthand.html](#)

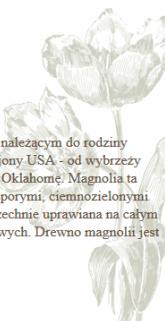
```
body {  
background: #ffffff url("images/tulip.gif")  
no-repeat top right;}
```

WYNIK

Poradnik ogrodnika

Magnolia

Magnolia grandiflora, nazywana także magnolią południową, jest drzewem należącym do rodziny Magnoliaceae, którego naturalnym miejscem występowania są południowe rejony USA - od wybrzeży Wirginii, poprzez Florydę na południu i zachód, po wschodnie stany: Teksas i Oklahoma. Magnolia ta osiąga 27,5 metra wysokości, jest okazałym, wiecznie zielonym drzewem ze sporymi, ciemnozielonymi liśćmi oraz dużymi, białymi, pachnącymi kwiatami. Roślina popularna, powszechnie uprawiana na całym świecie; wyhodowano też sporo odmian przeznaczonych do celów przemysłowych. Drewno magnoli jest twardze i cięzkie, używa się go do wytwarzania mebli, talerzy oraz oklein.



Jaskier

Ranunculus asiaticus (jaskier azjatycki) jest rodzajem jaskra (Ranunculus) pochodzącego ze wschodnich rejonów Morza Śródziemnego - z Azji Mniejszej, Europy Południowej (Krety, Karpatos, Rodos) oraz północno-wschodniej Afryki. Jest to zielna roślina wiełoletnia osiągająca do 45 centymetrów wysokości, o prostych lub rozgałęzionych łodygach. Liście przyziemne są trójdziełne, a im wyżej łodygi, tym podział liści sa głębsze; podobnie jak łodygi, także liście sa gładkie lub włoskowate. Kwiaty mają od 3 do 5 centymetrów średnicy, występują w różnych kolorach: czerwonym, różowym, żółtym, białym; na łodydze może się znajdować od jednego do kilku kwiatów.

Tulipan

Tulipa gesneriana L. to roślina należąca do rodziny liliowatych (Liliaceae). Pochodzenie tego gatunku nie jest pewne; prawdopodobnie pochodzi z Azji i został przeniesiony do południowo-zachodniej Europy. Większość z uprawianych gatunków, podgatunków i hodowlanych gatunków tulipanów pochodzi od *Tulipa gesneriana*. Zarówno roślina, jak i jej bulwa może powodować zapalenie skóry, którego przyczyną jest alergen tuliposide A; choć bulwy nadają się do konsumpcji, powodują niewielkie problemy zdrowotne. Obojnaczne kwiaty o słodkawym zapachu pojawiają się w kwietniu i maju. Bulwy tulipanów są niezwykle odporne na zimno i doskonale znoszą temperatury poniżej punktu zamarcia - pewien okres tak niskich temperatur jest niezbędny do prawidłowego wzrostu i kwitnienia rośliny, wyzwalanych zwiększoną intensywnością hormonu roślinnego - auksynu. Wysuszone i roztarte bulwy dodaje się do płatków zbożowych lub mąki.

Właściwość `background` odgrywa rolę skrótowego zapisu umożliwiającego podawanie wartości wszystkich pozostałych właściwości związanych z tem elementów w jednym miejscu. Można do niej dodać wartość właściwości `background-color`.

Wartości właściwości muszą zostać podane w następującej kolejności (przy czym dowolne z nich można pominąć, jeśli nie są potrzebne):

1. `background-color`
2. `background-image`
3. `background-repeat`
4. `background-attachment`
5. `background-position`

Standard CSS3 umożliwia także stosowanie kilku obrazów tła, określanych poprzez podanie kilku właściwości `background`. Ponieważ w trakcie pisania tej książki niewiele przeglądarek zapewniało tę możliwość, nie była ona często stosowana.

```
div {  
background:  
url(example-1.jpg)  
top left no-repeat,  
url(example-2.jpg)  
bottom left no-repeat,  
url(example-3.jpg)  
centre top repeat-x;}
```

Pierwszy obraz tła jest wyświetlony u góry elementu, a ostatni — u dołu.

PODMIENIANIE OBRAZÓW I SPRAJTY

Korzystając z CSS, można stworzyć łącze lub odnośnik, którego styl będzie się zmieniał po wskazaniu go myszą (efekt ten jest nazywany **podmienianiem**) oraz kliknięciu.

Taki efekt można uzyskać, określając obraz tła łącza lub przycisku, który przedstawia go w trzech różnych stanach (przy czym w danej chwili na stronie może być widoczna tylko jedna z tych postaci). Przyciski, których używamy w tym przykładzie, zostały przedstawione na następnej stronie. W rzeczywistości jest to jeden obraz zawierający trzy postaci dwóch przycisków.

Kiedy użytkownik umieści wskaźnik myszy w obszarze elementu, zmieniane jest położenie obrazu tła, dzięki czemu na stronie zostanie wyświetlony odpowiedni fragment obrazu.

Kiedy jeden obraz jest używany do wyświetlania kilku różnych elementów interfejsu użytkownika, nazywa się go **sprajtem**. W takich obrazach można umieszczać logo oraz wszelkie inne elementy interfejsu, takie jak przyciski.

Zaletą takiego rozwiązania jest to, że przeglądarka musi pobrać tylko jeden obraz, a nie kilka, a to może przyspieszyć wczytywanie i wyświetlanie strony.

rodzial-16/image-rollovers-and-sprites.html

HTML

```
<a class="button" id="add-to-basket">Doda  
do koszyka</a>  
<a class="button" id="framing-options">Opcje  
opakowania</a>
```

```
a.button {  
    height: 36px;  
    background-image: url("images/button-sprite.jpg");  
    text-indent: -9999px;  
    display: inline-block;}  
a#add-to-basket {  
    width: 174px;  
    background-position: 0px 0px;}  
a#framing-options {  
    width: 210px;  
    background-position: -175px 0px;}  
a#add-to-basket:hover {  
    background-position: 0px -40px;}  
a#framing-options:hover {  
    background-position: -175px -40px;}  
a#add-to-basket:active {  
    background-position: 0px -80px;}  
a#framing-options:active {  
    background-position: -175px -80px;}
```

CSS

PEŁNA POSTAĆ SPRAJTA



WYNIK



W tym przykładzie widać dwa łącza wyglądające jak przyciski. Każdy z przycisków występuje w trzech różnych stanach. Wszystkie one są umieszczone w tym samym obrazie.

Ponieważ łącza są elementami wewnętrzowymi, określamy dla nich właściwość `display`, przypisując jej wartość `inline-block`. Dzięki temu możemy określić ich szerokość i wysokość oraz dostosować je do wymiarów graficznych przycisków.

Do przesuwania obrazu tła i wyświetlania przycisku w odpowiednim stanie używana jest właściwość `background-position`.

Kiedy użytkownik umieści wskaźnik myszy w obszarze łącza, zastosowana zostanie reguła z pseudoklasą `:hover`, która zmodyfikuje wartość właściwości `background-position`, wyświetlając fragment obrazu reprezentujący inny stan przycisku.

Na urządzeniach wyposażonych w ekran dotykowy zastosowanie pseudoklasy `:hover` nie da żadnego rezultatu, gdyż urządzenia te jeszcze nie są w stanie wykryć, gdzie znajduje się palec użytkownika. Niemniej kliknięcie łącza spowoduje zmianę jego wyglądu.

Podobnie kiedy użytkownik kliknie łącze, zostanie zastosowana reguła z pseudoklasą `:active`, która spowoduje wyświetlenie trzeciego stanu przycisku.

CSS3: GRADIENTY

background-image

W CSS3 ma zostać wprowadzona możliwość definiowania gradientów wyświetlanego w tle elementu. Gradient taki jest tworzony przy użyciu właściwości `background-image`; gdy powstawała niniejsza książka, używana w tym celu składnia różniła się w zależności od przeglądarki.

Ponieważ tworzenie gradientów nie jest dostępne we wszystkich przeglądarkach, można w pierwszej kolejności określić obraz tła (który będzie przedstawiał identyczny gradient), a później podać alternatywne style CSS, które zostaną użyte w nowszych przeglądarkach.

W przykładzie przedstawionym na tej stronie tworzony jest gradient liniowy. Jak widać, wyświetlenie takiego gradientu wymaga podania dwóch kolorów (gradient tworzą kolory umieszczone pomiędzy nimi).

Niektóre przeglądarki umożliwiają określenie kąta gradientu, a nawet tworzenie gradientów różnych typów (takich jak kołowe), są one jednak obsługiwane znacznie rzadziej niż gradienty liniowe.

rodzial-16/gradient.html

CSS

```
#gradient {  
  /* zastępczy kolor tła */  
  background-color: #66cccc;  
  /* zastępczy obraz tła */  
  background-image: url(images/fallback-image.png);  
  /* Firefox 3.6+ */  
  background-image: -moz-linear-gradient(#336666,  
  #66cccc);  
  /* Safari 4+, Chrome 1+ */  
  background-image: -webkit-gradient(linear, 0% 0%,  
  0% 100%, from(#66cccc), to(#336666));  
  /* Safari 5.1+, Chrome 10+ */  
  background-image: -webkit-linear-gradient(#336666,  
  #66cccc);  
  /* Opera 11.10+ */  
  background-image: -o-linear-gradient(#336666,  
  #66cccc);  
  height: 150px;  
  width: 300px;}
```

WYNIK



KONTRAST W OBRAZACH TŁA

Jeśli chcemy nałożyć tekst na obraz tła, to by tekst był czytelny, obraz musi być mało kontrastowy.

WYSOKI KONTRAST



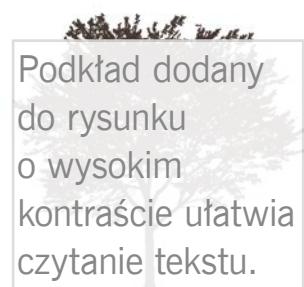
Tło o wysokim kontraście utrudnia czytanie tekstu.

NISKI KONTRAST



Tło o niskim kontraście ułatwia czytanie tekstu.

PODKŁAD

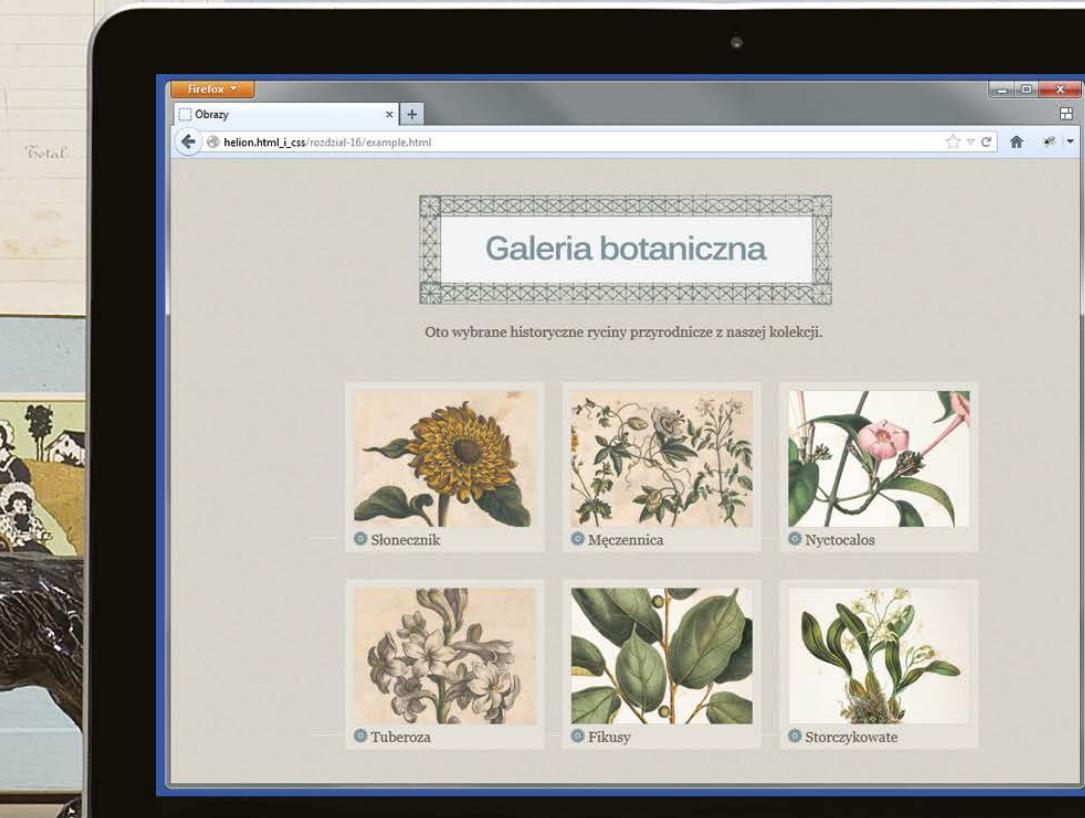


Podkład dodany do rysunku o wysokim kontraście ułatwia czytanie tekstu.

Większość fotografii cechuje się dość wysokim kontrastem, co oznacza, że nie nadają się one do wykorzystania jako obrazy tła.

Programy do edycji zdjęć, takie jak Photoshop i GIMP, udostępniają narzędzia pozwalające ręcznie zmniejszyć kontrast zdjęć.

Aby nałożyć tekst na obraz o wysokim kontraście, można go umieścić na półprzezroczystej warstwie („podkładzie”), co poprawi jego czytelność.



PRZYKŁAD

OBRAZY



Ten przykład pokazuje, w jaki sposób można wykorzystać CSS do utworzenia prostego układu galerii zdjęć.

Dzięki wyświetleniu w tle elementu `<body>` obrazu z tekstonią przykładowa strona uzyskała graficzne tło. Taki wielokrotnie powtarzany obraz tła czasami jest nazywany **tapetą**.

Zawartość strony została umieszczona w elemencie `<div>` przypisanym do klasy `wrapper`. Klasa ta określa szerokość elementu na 720 pikseli. Oprócz tego prawy i lewy margines elementu zostają określone jako `auto`, dzięki czemu jest on wyśrodkowany na ekranie.

Obrazy są umieszczone wewnętrz nowych elementów HTML5 `<figure>`, a ich opisy zostały podane przy użyciu elementów `<figcaption>`. Wymiary oraz kolor tła każdego z elementów `<figure>` zostały określone przy użyciu właściwości CSS. Także wymiary samych obrazów zostały określone w stylach, które dodatkowo wyświetlają wokół nich szare obramowanie o grubości 1 piksela.

W podpisach został zastosowany obraz tła, który jest widoczny po lewej stronie tekstu. Ponieważ nie chcemy, by obraz zajmował całe tło elementu, zaznaczyliśmy, że nie ma on być powtarzany. Dodatkowo określiliśmy wypełnienie z lewej strony tekstu podpisu, dzięki czemu nie będzie zachodził na obraz tła.

Każdy z elementów `<figure>` został umieszczony wewnątrz elementu `<div>`. Rozwiążanie to zastosowaliśmy z dwóch powodów. Po pierwsze ułatwia stworzenie układu trójkolumnowego, który uzyskujemy, określając szerokość elementów, ich margines oraz dodając do nich właściwość `float` przesuwającą elementy w lewo. A po drugie umożliwia dodanie do wszystkich zdjęć delikatnego cienia. W ten sposób nadajemy obrazom jakby trójwymiarowy wygląd, który upodabnia je nieco do kartek pocztowych. Aby zapewnić właściwe rozmieszczenie obrazu tła pod obrazkami, skorzystaliśmy z właściwości `background-position`.

PRZYKŁAD

OBRAZY

```
<!DOCTYPE html>
<html>
<head>
<title>Obrazy</title>
<style type="text/css">
body {
    color: #665544;
    background-color: #d4d0c6;
    background-image: url("images/backdrop.gif");
    font-family: Georgia, "Times New Roman", serif;
    text-align: center;}
.wrapper {
    width: 720px;
    margin: 0px auto;}
.header {
    margin: 40px 0px 20px 0px;}
.entry {
    width: 220px;
    float: left;
    margin: 10px;
    height: 198px;
    background-image: url("images/shadow.png");
    background-repeat: no-repeat;
    background-position: bottom;}
figure {
    display: block;
    width: 202px;
    height: 170px;
    background-color: #e7e3d8;
    padding: 9px;
    text-align: left;}
figure img {
    width: 200px;
    height: 150px;
    border: 1px solid #d6d6d6;}
figcaption {
    background-image: url("images/icon.png");
    padding-left: 20px;
    background-repeat: no-repeat;}
</style>
```

PRZYKŁAD

OBRAZY

```
</head>
<body>
<div class="wrapper">
  <div class="header">
    
    <p>Oto wybrane historyczne rycininy przyrodnicze z naszej kolekcji.</p>
  </div>
  <div class="entry">
    <figure>
      <figcaption>Słonecznik</figcaption>
    </figure>
  </div>
  <div class="entry">
    <figure>
      <figcaption>Męczennica</figcaption>
    </figure>
  </div>
  <div class="entry">
    <figure>
      <figcaption>Nyctocalos</figcaption>
    </figure>
  </div>
  <div class="entry">
    <figure>
      <figcaption>Tuberoza</figcaption>
    </figure>
  </div>
  <div class="entry">
    <figure>
      <figcaption>Fikusy</figcaption>
    </figure>
  </div>
  <div class="entry">
    <figure>
      <figcaption>Storczykowate</figcaption>
    </figure>
  </div>
</div>
</body>
</html>
```


PODSUMOWANIE

OBRAZY

- ▶ Wymiary obrazów można określić przy użyciu CSS. To bardzo wygodne rozwiązanie, zwłaszcza gdy na wielu stronach witryny używane są obrazy o tych samych wymiarach.
- ▶ Korzystając z CSS, można wyrównywać obrazy zarówno w poziomie, jak i w pionie.
- ▶ Istnieje możliwość wyświetlenia obrazu w tle każdego elementu strony.
- ▶ Obrazy tła mogą być wyświetlane tylko jeden raz, mogą się też powtarzać, zajmując całe tło elementu.
- ▶ Zmieniając położenie obrazu tła, można tworzyć efekt podmieniania obrazów.
- ▶ Aby ograniczyć liczbę obrazów pobieranych przez przeglądarkę, można zastosować sprajty.

17

UKŁAD W HTML5

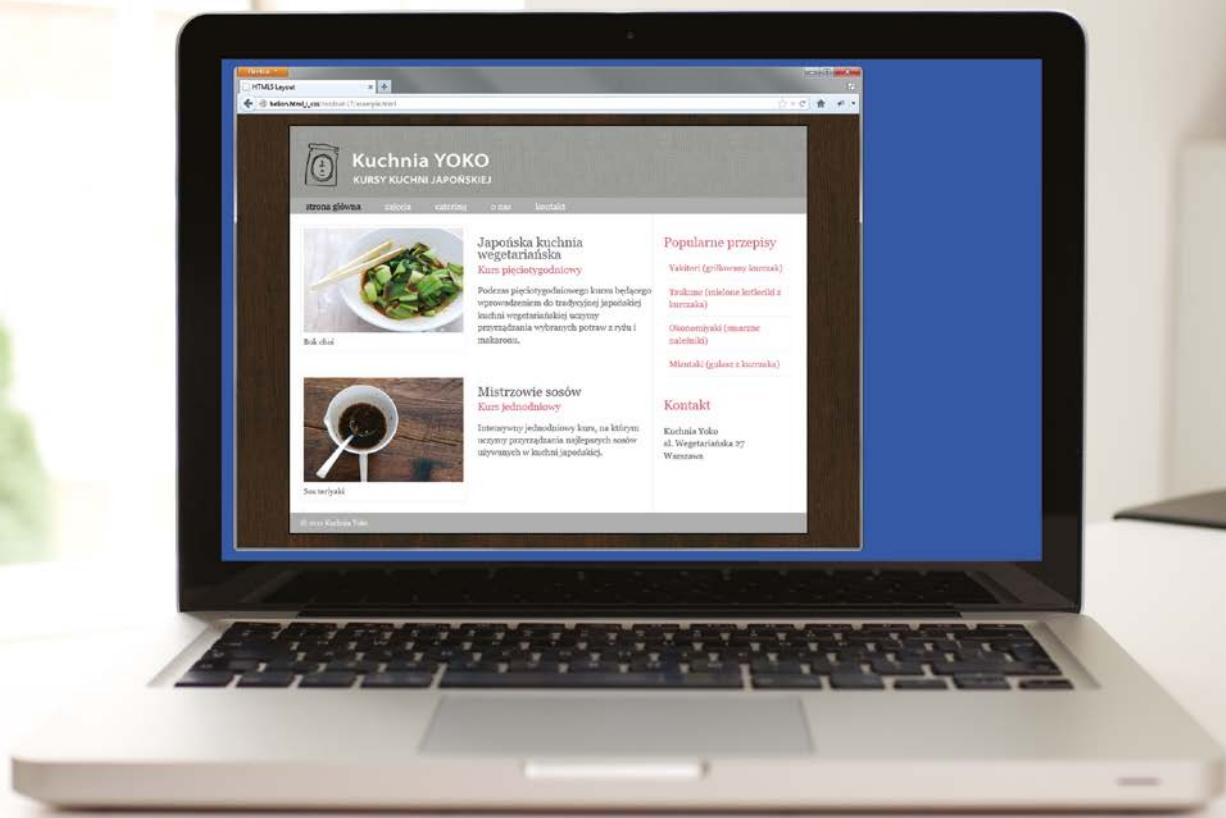
- ▶ Elementy służące do określania układu stron, dostępne w HTML5.
- ▶ Jak stare przeglądarki obsługują nowe elementy?
- ▶ Określanie wyglądu nowych elementów HTML5 przy użyciu stylów.

HTML5 wprowadza zbiór nowych elementów ułatwiających definiowanie struktury strony.

Zostały one opisane w osobnym rozdziale (a nie przy okazji prezentowania innych elementów HTML przedstawionych we wcześniejszej części książki), gdyż teraz, kiedy już wiemy, jak określać układ stron przy użyciu arkuszy stylów, łatwiej będzie nam zrozumieć, jak należy je stosować. Te nowe elementy będą odgrywały znaczącą rolę w tworzeniu nowoczesnych układów stron. W tym rozdziale:

- Poznamy nowe elementy HTML5 służące do określania układu stron oraz ich zastosowania.
- Dowiemy się, dlaczego stanowią użyteczną alternatywę dla elementów `<div>`.
- Dowiemy się, jak sprawić, by starsze przeglądarki rozpoznawały te nowe elementy.

Podobnie jak wszystkie nowości wprowadzane do HTML5 i CSS3, także te nowe elementy mogą się jeszcze zmienić, jednak są one już powszechnie używane przez twórców stron i prawdopodobnie także Czytelnik będzie chciał z nich skorzystać.



TRADYCYJNE UKŁADY HTML

Przez wiele lat twórcy stron WWW używali elementów **<div>** do grupowania powiązanych ze sobą elementów stron (takich jak elementy tworzące nagłówek, artykuł, stopkę lub pasek boczny). Do określania funkcji, jakie te elementy **<div>** pełnią na stronie, używane były atrybuty **class** oraz **id**.

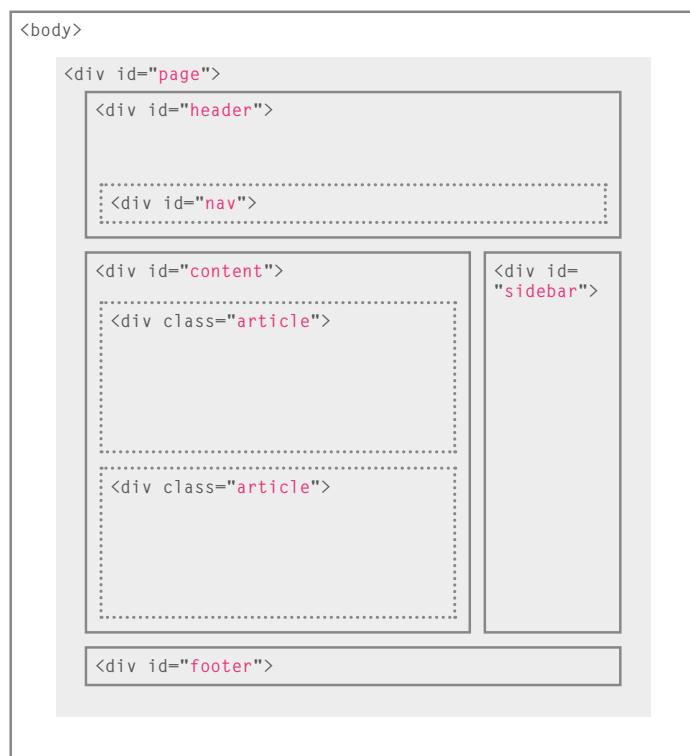
Z prawej strony przedstawiony został przykład dosyć często spotykanego układu strony (stosowany najczęściej na blogach).

W górnej części strony znajduje się nagłówek zawierający logo oraz podstawowe elementy nawigacyjne.

Poniżej nagłówka znajduje się jeden (lub więcej) artykuł lub wpis. Czasami umieszczane są także krótkie podsumowania z łączami prowadzącymi do konkretnych wpisów.

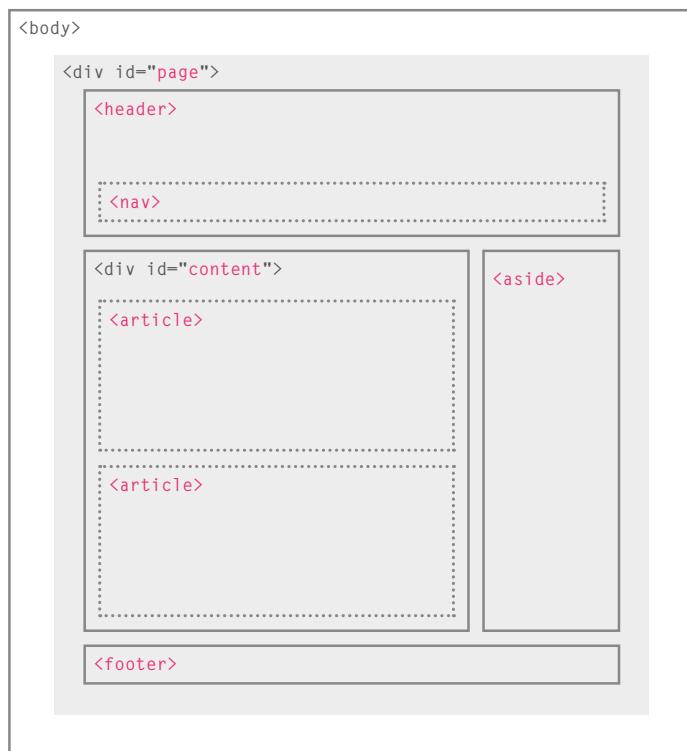
Z prawej strony umieszczony jest pasek boczny (zawierający np. formularz do wyszukiwania, łącza do innych, najnowszych artykułów czy sekcji witryny, a być może nawet do reklamy).

Tworząc taką witrynę, autorzy zazwyczaj umieszczają jej poszczególne główne sekcje wewnętrz elementów **<div>**, określając ich przeznaczenie przy użyciu atrybutów **class** oraz **id**.



NOWE ELEMENTY HTML5 SŁUŻĄCE DO TWORZENIA UKŁADÓW STRON

W języku HTML5 wprowadzony został zbiór nowych elementów służących do oznaczania fragmentów tworzonych stron. Nazwy tych elementów określają rodzaj zawartości, jaką należy w nich umieszczać. Choć mogą się jeszcze zmieniać, nie powstrzymuje to autorów stron przed ich stosowaniem.



Przykład przedstawiony obok ma dokładnie taką samą strukturę jak ten zamieszczony na poprzedniej stronie. Niemniej wiele spośród użytych wcześniej elementów <div> zostało zastąpionych nowymi elementami HTML5. I tak nagłówek został umieszczony wewnętrz elementu <header>, elementy nawigacyjne znalazły się w elemencie <nav>, a artykuły — wewnątrz odrębnych elementów <article>. Elementy te zostały stworzone po to, by autorzy stron mogli za ich pomocą opisywać strukturę tworzonych stron. Czytniki ekranowe np. mogą pozwolić użytkownikom na ignorowanie nagłówków i stopek i prezentować jedynie główną zawartość stron. Podobnie wyszukiwarki mogą nadawać większe znaczenie zawartości elementów <article> niż elementów <header> i <footer>. Sądzę, że Czytelnik także przyzna, że zastosowanie tych elementów ułatwia analizę kodu źródłowego strony.

NAGŁÓWKI I STOPKI

<header> <footer>

Elementy <header>

i <footer> mogą być używane do tworzenia:

- głównego nagłówka i stopki prezentowanych odpowiednio na górze i na dole każdej strony witryny;
- nagłówka i stopki umieszczanych w poszczególnych elementach <article> oraz <section>.

W tym przykładzie element <header> zawiera logo witryny oraz główne elementy nawigacyjne. Z kolei element <footer> zawiera informacje o prawach autorskich oraz odnośniki do stron z polityką prywatności i warunkami korzystania ze strony. Elementy <header> i <footer> można także umieszczać wewnętrz elementów <article> oraz <section>. W takich przypadkach powinny one zawierać odpowiednio nagłówek i stopkę konkretnego artykułu lub sekcji strony.

Przykładowo na stronie zawierającej kilka wpisów blogowych każdy z nich może być umieszczony w osobnej sekcji. W takim przypadku w elemencie <header> można umieścić tytuł wpisu oraz jego datę, natomiast w elemencie <footer> — odnośniki

rodzial-17/html5-layout.html

HTML

```
<header>
  <h1>Kuchnia Yoko</h1>
  <nav>
    <ul>
      <li><a href="" class="current">strona główna
      </a></li>
      <li><a href="">zajęcia</a></li>
      <li><a href="">catering</a></li>
      <li><a href="">o nas</a></li>
      <li><a href="">kontakt</a></li>
    </ul>
  </nav>
</header>
```

rodzial-17/html5-layout.html

HTML

```
<footer>
  &copy; 2011 Kuchnia Yoko
</footer>
```

umożliwiające podzielenie się artykułem w portalach społecznościowych.

Warto zauważyć, że wszystkie przykłady przedstawione w tym rozdziale pochodzą z jednego pliku o nazwie *html5-layout.html*.

NAWIGACJA

<nav>

HTML

rodzaj-17/html5-layout.html

```
<nav>
  <ul>
    <li><a href="" class="current">strona główna</a></li>
    <li><a href="">zajęcia</a></li>
    <li><a href="">catering</a></li>
    <li><a href="">o nas</a></li>
    <li><a href="">kontakt</a></li>
  </ul>
</nav>
```

Element <nav> jest przeznaczony do umieszczania głównych bloków nawigacyjnych witryny, takich jak lista łączy do jej głównych działów.

Wróćmy do przykładowej strony bloga: gdybyśmy chcieli zakończyć artykuł listą łączy do innych powiązanych z nim wpisów, nie można by ich było uznać za główne elementy nawigacyjne, a co za tym idzie, nie należałoby ich umieszczać w elemencie <nav>.

W czasie kiedy powstawała ta książka, projektanci używający języka HTML5 zdecydowali, że element <nav> będzie używany do umieszczania łączy wyświetlanego u dołu każdej ze stron witryny (czyli do stron zawierających politykę prywatności, warunki korzystania, informacje o ułatwieniach dostępu itd.). Wciąż jednak nie wiadomo, czy takie rozwiązanie zyska popularność i będzie powszechnie stosowane.

ARTYKUŁY

<article>

Element `<article>` pełni funkcję pojemnika, w którym są umieszczane dowolne sekcje strony, przy czym powinny one być stosunkowo autonomiczne i nadawać się do zebrania w większą grupę.

Mogą to być poszczególne artykuły lub wpisy do bloga, komentarze lub wpisy na forum czy wszelkie inne niezależne fragmenty treści.

Jeśli strona zawiera kilka artykułów (albo nawet podsumowania kilku artykułów), to każdy z nich powinien zostać umieszczony w osobnym elemencie `<article>`.

Elementy `<article>` można zagnieździć, np. wpis do bloga zostanie umieszczony w jednym elemencie `<article>`, a poszczególne komentarze do tego wpisu znajdą się w kolejnych elementach.

rodzial-17/html5-layout.html

HTML

```
<article>
  <figure>
    
    <figcaption>Bok choi</figcaption>
  </figure>
  <hgroup>
    <h2>Japońska kuchnia wegetariańska</h2>
    <h3>Kurs pięciotygodniowy</h3>
  </hgroup>
  <p>Podczas pięciotygodniowego kursu będącego wprowadzeniem do tradycyjnej japońskiej kuchni wegetariańskiej uczymy przyrządzaania wybranych potraw z ryżu i makaronu.</p>
</article>
<article>
  <figure>
    
    <figcaption>Sos teriyaki</figcaption>
  </figure>
  <hgroup>
    <h2>Mistrzowie sosów</h2>
    <h3>Kurs jednodniowy</h3>
  </hgroup>
  <p>Intensywny jednodniowy kurs, na którym uczymy przyrządzaania najlepszych sosów używanych w kuchni japońskiej.</p>
</article>
```

SEKCJE BOCZNE

<aside>

HTML

rodzaj-17/html5-layout.html

```
<aside>
  <section class="popular-recipes">
    <h2>Popularne przepisy</h2>
    <a href="">Yakitori (grillowany kurczak)</a>
    <a href="">Tsukune (mielone kotleciki z kurczaka)</a>
    <a href="">Okonomiyaki (smaczne naleśniki)</a>
    <a href="">Mizutaki (gulasz z kurczaka)</a>
  </section>
  <section class="contact-details">
    <h2>Kontakt</h2>
    <p>Kuchnia Yoko<br />
      al. Wegetariańska 27<br />
      Warszawa</p>
  </section>
</aside>
```

Element <aside> ma kilka zastosowań, zależnych od tego, czy znajduje się wewnątrz elementu <article>, czy poza nim.

W przypadku umieszczenia elementu <aside> wewnątrz elementu <article> powinien on zawierać informacje powiązane z artykułem, lecz niekoniecznie z jego podstawowym znaczeniem. Za taki element związany z artykułem można by uznać np. wyróżniony cudzysłów lub słownik.

Jeśli natomiast element <aside> jest umieszczony poza elementem <article>, to służy jako pojemnik dla treści związanych z całą stroną. Można by w nim umieścić np. łącza do innych sekcji witryny, listę ostatnich publikacji, pole wyszukiwania albo ostatnie wpisy autora na Twitterze.

SEKCJE

<section>

Element <section> służy do grupowania innych powiązanych ze sobą elementów. Zazwyczaj każda taka sekcja ma własny nagłówek.

Przykładowo na stronie głównej witryny może się znaleźć kilka elementów <section> zawierających różne sekcje strony, takie jak najnowsze publikacje, najpopularniejsze produkty oraz formularze do subskrypcji biuletynu informacyjnego.

Ponieważ element <section> grupuje powiązane ze sobą elementy, można w nim umieścić kilka elementów <article> reprezentujących artykuły poświęcone tej samej tematyce lub mające to samo przeznaczenie.

Alternatywnie, jeśli strona zawiera bardzo długi artykuł, to korzystając z elementu <section>, można podzielić jego treść na kilka fragmentów.

Elementu <section> nie należy używać jako pojemnika zawierającego całą treść strony (chyba że całą jej zawartość stanowi jeden, spójny fragment treści). Jeśli szukamy elementu, który nadalby się do roli pojemnika zawierającego całą stronę, w tym celu wciąż najlepiej zastosować element <div>.

rodzaj-17/html5-layout.html

HTML

```
<section class="popular-recipes">
  <h2>Popularne przepisy</h2>
  <a href="">Yakitori (grillowany kurczak)</a>
  <a href="">Tsukune (mielone kotleciki z kurczaka)</a>
  <a href="">Okonomiyaki (smaczne naleśniki)</a>
  <a href="">Mizutaki (gulasz z kurczaka)</a>
</section>
<section class="contact-details">
  <h2>Kontakt</h2>
  <p>Kuchnia Yoko<br />
    al. Wegetariańska 27<br />
    Warszawa</p>
</section>
```

GRUPY NAGŁÓWKÓW

<hgroup>

HTML

rodzaj-17/html5-layout.html

```
<hgroup>
  <h2>Japońska kuchnia wegetariańska</h2>
  <h3>Kurs pięciotygodniowy</h3>
</hgroup>
```

Element <hgroup> służy do grupowania od jednego do kilku elementów <h1> – <h6>, tak by były one traktowane jako jeden nagłówek.

Elementu <hgroup> można by użyć do podania zarówno tytułu umieszczonego w elemencie <h2>, jak i podtytułu umieszczonego w nagłówku <h3>.

Ten element został przyjęty z mieszanymi uczuciami. Kiedy twórcy języka HTML5 zaproponowali go po raz pierwszy, pojawiły się skargi, które doprowadziły do usunięcia go z propozycji standardu. Nieco później jednak pewne osoby zmieniły zdanie i element <hgroup> ponownie znalazł się w propozycji HTML5. Niektórzy twórcy stron nie lubią go i preferują umieszczanie podtytułów wewnętrz elementów <p> (używają przy tym atrybutu określającego, że dany element jest nagłówkiem niższego stopnia).

Niektórzy sugerują, że element <hgroup> ma niewiele zastosowań oprócz tego, że można go użyć do określania stylów. Niemniej cieszy się popularnością wśród tych projektantów, którzy są zwolennikami grupowania nagłówka głównego oraz nagłówka niższego stopnia (gdyż oba mogą stanowić integralne części nagłówka).

ILUSTRACJE

<figure> <figcaption>

Element <figure> został już przedstawiony w rozdziale 5 poświęconym obrazom. Może on zawierać dowolne treści, do których odwołuje się główna treść artykułu (nie tylko obrazy).

Koniecznie należy zauważyc, że artykuł nie powinien tracić znaczenia w przypadku przesunięcia zawartości elementu <figure> (do innej części strony lub nawet przeniesienia go na inną stronę).

Dlatego należy go używać wyłącznie w sytuacjach, gdy w treści artykułu znajduje się jedynie odwołanie do elementu (a nie gdy stanowi on integralną część tej treści).

Oto kilka przykładów zastosowań elementu <figure>:

- obrazy,
- klipy wideo,
- wykresy,
- diagramy,
- listingi kodu,
- teksty uzupełniające główną treść artykułu.

W elemencie <figure> należy także umieszczać element <figcaption> służący do podania tekstowego opisu zawartości elementu <figure>.

rodzaj-17/html5-layout.html

HTML

```
<figure>
  
  <figcaption>Bok choi</figcaption>
</figure>
```

GRUPOWANIE ELEMENTÓW W SEKCJE

HTML

rodzaj-17/html5-layout.html

```
<div class="wrapper">
  <header>
    <h1>Kuchnia Yoko</h1>
    <nav>
      <!-- zawartość paska nawigacyjnego -->
    </nav>
  </header>
  <section class="courses">
    <!-- zawartość sekcji -->
  </section>
  <aside>
    <!-- zawartość paska bocznego -->
  </aside>
  <footer>
    <!-- zawartość stopki -->
  </footer>
</div><!-- .wrapper -->
```

Może się wydawać dziwne, że po prezentacji nowych elementów HTML5 wracamy do elementu `<div>` (w końcu te nowe elementy mają go zastępować).

Element `<div>` wciąż pozostaje ważnym sposobem grupowania powiązanych ze sobą elementów stron, gdyż przedstawione wcześniej nowe elementy HTML5 należy stosować wyłącznie w ścisłe określonych celach.

Wszędzie tam, gdzie nowe elementy HTML5 nie nadają się do zastosowania, trzeba będzie nadal używać elementu `<div>`. W tym przykładzie został on zastosowany jako pojemnik dla całej zawartości strony.

Pewnie niektórzy zastanawiają się, dlaczego nie ma elementu `<content>`, w którym można by umieścić główną zawartość strony. Wynika to z faktu, że wszystko, co znajduje się poza elementami `<header>`, `<footer>` oraz `<aside>`, można uznać za główną część strony.

UMIESZCZANIE ELEMENTÓW BLOKOWYCH W ŁĄCZACH

HTML5 pozwala umieszczać wewnętrz elementów `<a>` dowolne elementy blokowe wraz z zawartością. Dzięki temu cały blok treści można przekształcić w łącze.

Nie jest to nowy element wprowadzony do HTML5, niemniej we wcześniejszych wersjach języka takie rozwiązanie było uznawane za nieprawidłowe.

Warto zwrócić uwagę na to, że kod tej strony różni się nieco od kodu pozostałych przykładów prezentowanych w tym rozdziale.

rodzaj-17/html5-layout.html

HTML

```
<a href="introduction.html">
  <article>
    <figure>
      
      <figcaption>Bok choi</figcaption>
    </figure>
    <hgroup>
      <h2>Japońska kuchnia wegetariańska</h2>
      <h3>Kurs pięciotygodniowy</h3>
    </hgroup>
    <p>Podczas pięciotygodniowego kursu będącego wprowadzeniem do tradycyjnej japońskiej kuchni wegetariańskiej uczymy przyrządzania wybranych potraw z ryżu i makaronu.</p>
  </article>
</a>
```

UŁATWIENIE STARSZYM PRZEGŁĄDARKOM ZROZUMIENIA NOWEGO KODU

CSS

rodzaj-17/

```
header, section, footer, aside, nav, article, figure
{
    display: block;
```

HTML

rodzaj-17/html5-layout.html

```
<!--[if lt IE 9]>
<script src="http://html5shiv.googlecode.com/svn/
  trunk/html5.js"></script>
<![endif]-->
```

Starsze przeglądarki, które nie rozpoznają nowych elementów HTML5, będą je automatycznie traktować jako elementy wewnętrzwerszowe. Dlatego aby im pomóc, do używanego arkusza stylów należy dodać przedstawioną obok regułę, która określa wszystkie te elementy jako blokowe.

Dodatkowo pierwszą wersją przeglądarki Internet Explorer, która pozwalała na kojarzenie stylów z tymi nowymi elementami HTML5, jest IE9. Aby określić wygląd tych elementów przy użyciu stylów we wcześniejszych wersjach Internet Explorera, należy użyć prostego skryptu JavaScript nazywanego **podkładką HTML5** (ang. *HTML5 shim*).

Niestety, to rozwiązanie wymaga, by użytkownicy korzystający z przeglądarki IE8 lub wersji wcześniejszych mieli włączoną obsługę języka JavaScript. Jeśli obsługa ta będzie wyłączona, nie będą w stanie zobaczyć zawartości nowych elementów HTML5.

Aby korzystać z tego rozwiązania, nie trzeba znać języka JavaScript. Wystarczy dodać do swojej strony skrypt przechowywany na serwerach Google. Można go umieścić wewnątrz **komentarzy warunkowych**, sprawdzających, czy używana wersja przeglądarki jest wcześniejsza od IE9 (stąd zastosowanie wyrażenia `lt IE 9`).

Firefox

HTML5 Layout

helion.html5_css/rozdzial-17/example.html

Kuchnia YOKO

KURSY KUCHNI JAPÓŃSKIEJ

strona główna zajęcia catering o nas kontakt



Bok choi

Japońska kuchnia wegetariańska
Kurs pięciotygodniowy

Podeczas pięciotygodniowego kursu będącego wprowadzeniem do tradycyjnej japońskiej kuchni wegetariańskiej uczymy się przyrządzania wybranych potraw z ryżu i makaronu.



Sos teriyaki

Mistrzowie sosów
Kurs jednodniowy

Intensywny jednodniowy kurs, na którym uczymy się przyrządzania najlepszych sosów używanych w kuchni japońskiej.

Popularne przepisy

- Yakitori (grillowany kurczak)
- Tsukune (mięcone kotleciki z kurczaka)
- Okonomiyaki (smaczne naleśniki)
- Mizutaki (gulasz z kurczaka)

Kontakt

Kuchnia Yoko
al. Wegetariańska 27
Warszawa

© 2011 Kuchnia Yoko

PRZYKŁAD

UKŁAD W HTML5



Ten przykład przedstawia witrynę poświęconą gotowaniu — jej struktura została utworzona przy wykorzystaniu nowych elementów HTML5 (a nie elementów `<div>`).

Nagłówek oraz stopka strony zostały umieszczone odpowiednio w elementach `<header>` oraz `<footer>`. Informacje o zajęciach zostały zgrupowane w elemencie `<section>` z atrybutem `class` o wartości `courses` (który ma odróżniać go od pozostałych elementów `<section>` umieszczonych na stronie). Pasek boczny został umieszczony w elemencie `<aside>`.

Każdy z kursów został przedstawiony w osobnym elemencie `<article>` i uzupełniony elementami `<figure>` oraz `<figcaption>` zawierającymi obrazek. Nagłówek każdego kursu ma dodatkowy podnagłówek, a oba te elementy są umieszczone wewnętrz elementu `<hgroup>`. Pasek boczny zawiera listę przepisów oraz informacje kontaktowe, umieszczone w dwóch osobnych elementach `<section>`.

Wygląd całej strony jest określany przy użyciu CSS. Jedyna różnica polega na tym, że w selektorach zostały użyte nowe elementy HTML5, co pozwala stworzyć reguły odnoszące się do tych elementów. Aby arkusz stylów działał w starszych wersjach przeglądarek Internet Explorer (wcześniejnych od wersji 9), strona zawiera odwołanie do specjalnego skryptu JavaScript (dostępnego na serwerach firmy Google), umieszczone w komentarzach warunkowych.

PRZYKŁAD

UKŁAD W HTML5

```
<!DOCTYPE html>
<html>
<head>
<title>Układ w HTML5</title>
<style type="text/css">
  header, section, footer, aside, nav, article, figure, figcaption {
    display: block;
  }
  body {
    color: #666666;
    background-color: #f9f8f6;
    background-image: url("images/dark-wood.jpg");
    background-position: center;
    font-family: Georgia, Times, serif;
    line-height: 1.4em;
    margin: 0px;
  }
  .wrapper {
    width: 940px;
    margin: 20px auto 20px auto;
    border: 2px solid #000000;
    background-color: #ffffff;
  }
  header {
    height: 160px;
    background-image: url(images/header.jpg);
  }
  h1 {
    text-indent: -9999px;
    width: 940px;
    height: 130px;
    margin: 0px;
  }
  nav, footer {
    clear: both;
    color: #ffffff;
    background-color: #aeaca8;
    height: 30px;
  }
  nav ul {
    margin: 0px;
    padding: 5px 0px 5px 30px;
  }
  nav li {
    display: inline;
    margin-right: 40px;
  }
  nav li a {
```

PRZYKŁAD

UKŁAD W HTML5

```
color: #ffffff;
nav li a:hover, nav li a.current {
  color: #000000;
}
section.courses {
  float: left;
  width: 659px;
  border-right: 1px solid #eeeeee;
}
article {
  clear: both;
  overflow: auto;
  width: 100%;
}
hgroup {
  margin-top: 40px;
}
figure {
  float: left;
  width: 290px;
  height: 220px;
  padding: 5px;
  margin: 20px;
  border: 1px solid #eeeeee;
}
figcaption {
  font-size: 90%;
  text-align: left;
}
aside {
  width: 230px;
  float: left;
  padding: 0px 0px 0px 20px;
}
aside section a {
  display: block;
  padding: 10px;
  border-bottom: 1px solid #eeeeee;
}
aside section a:hover {
  color: #985d6a;
  background-color: #efefef;
}
a {
  color: #de6581;
  text-decoration: none;
}
h1, h2, h3 {
  font-weight: normal;
}
h2 {
```

PRZYKŁAD

UKŁAD W HTML5

```
margin: 10px 0px 5px 0px;
padding: 0px;}
h3 {
  margin: 0px 0px 10px 0px;
  color: #de6581;}
aside h2 {
  padding: 30px 0px 10px 0px;
  color: #de6581;}
footer {
  font-size: 80%;
  padding: 7px 0px 0px 20px;}
</style>
<!--[if lt IE 9]>
<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
</head>
<body>
<div class="wrapper">
  <header>
    <h1>Kuchnia Yoko</h1>
    <nav>
      <ul>
        <li><a href="" class="current">strona główna</a></li>
        <li><a href="">zajęcia</a></li>
        <li><a href="">catering</a></li>
        <li><a href="">o nas</a></li>
        <li><a href="">kontakt</a></li>
      </ul>
    </nav>
  </header>
  <section class="courses">
    <article>
      <figure>
        
        <figcaption>Bok choi</figcaption>
      </figure>
      <hgroup>
        <h2>Japońska kuchnia wegetariańska</h2>
        <h3>Kurs pięciotygodniowy</h3>
      </hgroup>
```

PRZYKŁAD

UKŁAD W HTML5

```
<p>Podczas pięciotygodniowego kursu będącego wprowadzeniem do tradycyjnej
japońskiej kuchni wegetariańskiej uczymy przyrządania
wybranych potraw z ryżu i makaronu.</p>
</article>
<article>
  <figure>
    
    <figcaption>Sos teriyaki</figcaption>
  </figure>
  <hgroup>
    <h2>Mistrzowie sosów</h2>
    <h3>Kurs jednodniowy</h3>
  </hgroup>
  <p>Intensywny jednodniowy kurs, na którym uczymy przyrządania najlepszych
  sosów używanych w kuchni japońskiej.</p>
</article>
</section>
<aside>
  <section class="popular-recipes">
    <h2>Popularne przepisy</h2>
    <a href="">Yakitori (grillowany kurczak)</a>
    <a href="">Tsukune (mielone kotleciki z kurczaka)</a>
    <a href="">Okonomiyaki (smaczne naleśniki)</a>
    <a href="">Mizutaki (gulasz z kurczaka)</a>
  </section>
  <section class="contact-details">
    <h2>Kontakt</h2>
    <p>Kuchnia Yoko<br />
      al. Wegetariańska 27<br />
      Warszawa</p>
  </section>
</aside>
<footer>
  &copy; 2011 Kuchnia Yoko
</footer>
</div><!-- .wrapper -->
</body>
</html>
```


PODSUMOWANIE

UKŁAD W HTML5

- ▶ Nowe elementy HTML5 określają przeznaczenie różnych fragmentów stron WWW i opisują ich strukturę.
- ▶ Te nowe elementy ułatwiają zrozumienie kodu (w porównaniu ze stosowanymi wcześniej elementami <div>).
- ▶ Starsze przeglądarki, które nie rozumieją języka HTML5, należy poinstruować, które z tych nowych elementów są elementami blokowymi.
- ▶ Aby nowe elementy HTML5 działały w przeglądarce IE8 (oraz wersjach wcześniejszych), konieczne jest użycie skryptu JavaScript dostępnego na serwerach firmy Google.

18

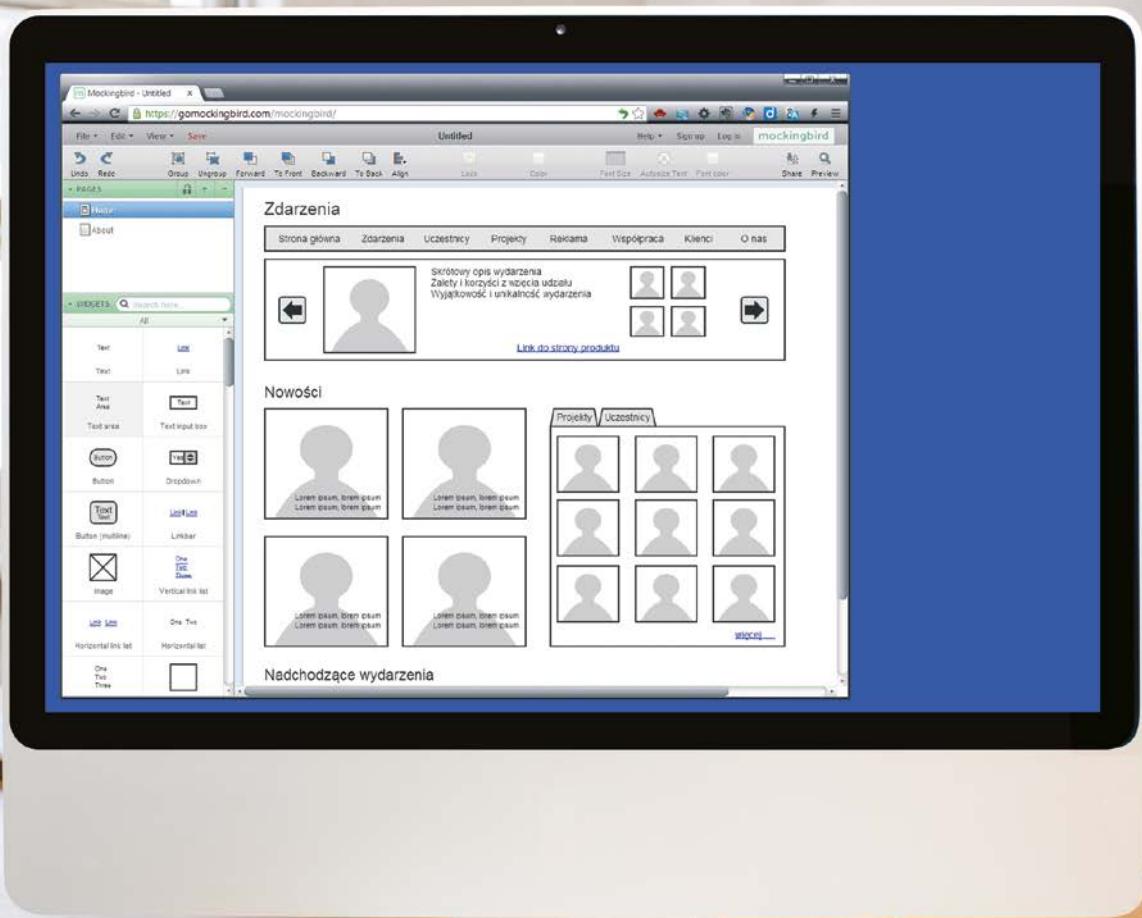
PROCES I PROJEKT

- ▶ Jak podejść do tworzenia witryny WWW?
- ▶ Określenie odbiorców witryny oraz ich potrzeb.
- ▶ Jak przedstawiać informacje, które odbiorcy chcą oglądać?

W tej części rozdziału opiszemy proces, który można zastosować podczas tworzenia nowej witryny.

W tym rozdziale podpowiadamy, jak określić, jakie osoby mogą być zainteresowane odwiedzaniem tworzonej witryny, oraz zapewnić, by na jej stronach znalazły się te informacje, których te osoby potrzebują. Oprócz tego przedstawimy kilka kluczowych pojęć związanych z teorią projektowania — ułatwią nam one tworzenie profesjonalnie wyglądających witryn. W tym rozdziale:

- Powiemy, jak określić docelową grupę użytkowników, którzy mogą być zainteresowani witryną, oraz jakich informacji będą na niej poszukiwać.
- Podpowiemy, jak zorganizować informacje, by użytkownicy mogli odnaleźć to, czego szukają.
- Omówimy wybrane zagadnienia teorii projektowania związane z prezentowaniem informacji w sposób, który pomoże użytkownikom realizować ich cele.
- Przedstawimy porady, które pomogą Czytelnikowi tworzyć bardziej atrakcyjne i profesjonalnie wyglądające witryny.



DO KOGO JEST SKIEROWANA TA WITRYNA?

Każda witryna powinna być projektowana z myślą o pewnej docelowej grupie odbiorców — a nie o jej twórcy lub właścielowi. Dlatego bardzo duże znaczenie ma określenie, kto stanowi tę grupę.

Przydatne może być zadanie pytań o osoby, które, według nas, mogą być zainteresowane tematyką naszej witryny.

Jeśli zapytamy klienta, dla kogo jest przeznaczona witryna, często spotkamy się z odpowiedzią: dla całego świata.

Jeśli jednak realistycznie podejść do sprawy, jest mało prawdopodobne, by każdy był zainteresowany tworzoną witryną. Jeśli służy ona sprzedaży żarówek, to chociaż użytkownicy komputerów zapewne też z nich korzystają, wątpliwe, by zamawiali je w firmie zlokalizowanej w innym kraju.

Nawet jeśli witryna może budzić zainteresowanie u szerokiego grona odbiorców, to i tak warto przeanalizować dane demograficzne próbki odbiorców.

ODBIORCY DOCELOWI: UŻYTKOWNICY INDYWIDUALNI

- Jaki jest zakres wieku docelowej grupy odbiorców?
- Czy witryna będzie się bardziej podobać kobietom, czy mężczyznom?
- Z jakiego kraju będą pochodzić użytkownicy witryny?
- Czy mieszkają oni w rejonach miejskich, czy wiejskich?
- Jaki jest średni przychód osób odwiedzających witrynę?
- Jaki jest ich poziom wykształcenia?
- Jaki jest ich status rodzinny?
- Czym się zajmują?
- Ile godzin w tygodniu pracują?
- Jak często korzystają z internetu?
- Jakich urządzeń używają do przeglądania stron WWW?

ODBIORCY DOCELOWI: FIRMY

- Jak duża jest firma lub jak duży jest jej dział?
- Jaką pozycję w firmie zajmują osoby odwiedzające witrynę?
- Czy użytkownicy będą korzystali z witryny dla siebie, czy dla kogoś innego?
- Jak dużym budżetem zarządzają?

Wymyślmy jakieś fikcyjne osoby będące przedstawicielami docelowej grupy użytkowników. Staną się one naszymi przyjaciółmi. Mogą mieć wpływ na podejmowane decyzje projektowe, począwszy od doboru palety kolorów, a skończywszy na liczbie szczegółów w opisach.

Imię	Grzegorz	Marysia	Janek	Jurek	Ewa
Płeć	M	K	M	M	K
Wiek	28	47	19	32	35
Miejsce zamieszkania	Katowice	Kraków	Warszawa	Wrocław	Gdańsk
Zawód	nauczyciel	prawnik	student	kierownik	dziennikarz
Przychód	39 000	82 000	19 000	55 000	67 000
Korzystanie z internetu	2 – 3 dni i weekendy	codziennie	codziennie	4 – 5 dni i weekendy	codziennie

Jeśli pojawią się pytania o to, w jaki sposób witryna będzie używana albo jakie powinny być obowiązujące w niej priorytety,

będziemy mogli wrócić do tej tabeli i zadać sobie pytanie: czego Grzegorz lub Ewa chcieliby w takiej sytuacji?

DLACZEGO LUDZIE ODWIEDZAJĄ NASZĄ STRONĘ?

Skoro już wiadomo, kim są osoby odwiedzające stronę, należy się zastanowić, **dlaczego** to robią. Choć niektórzy z nich trafiają na stronę przez przypadek, większość znajdzie się na niej z określonego powodu.

Na zawartość witryny oraz jej projekt powinny mieć wpływ cele jej użytkowników.

Aby łatwiej było określić, dlaczego użytkownicy odwiedzają witrynę, należy odpowiedzieć sobie na pytania należące do dwóch prostych kategorii:

1. Pierwszą z nich są próby określenia **motywacji** skłaniającej ludzi od odwiedzania witryny.

2. Druga jest związana z **celami**, do których użytkownicy dążą. To właśnie głównie one skłaniają użytkowników do odwiedzania witryny *w danej chwili*.

KLUCZOWE POWODY

- Czy użytkownicy szukają ogólnie pojętej rozrywki, czy pragną osiągnąć jakieś konkretne cele?
- Jeśli chodzi o konkretny cel, to czy ma on charakter osobisty, czy zawodowy?
- Czy uważają, że poświęcanie czasu na wizytę w witrynie to ważna czynność, czy raczej luksus?

KONKRETNE CELE

- Czy użytkownicy pragną zdobyć ogólne informacje (dotyczące jakiegoś zagadnienia lub firmy), czy szukają czegoś konkretnego (danych na temat jakiegoś faktu albo opisu produktu)?
- Czy już znają udostępniane na stronie usługi lub produkty, czy należy je im przedstawić?
- Czy szukają informacji, które szybko tracą na aktualności, takich jak najnowsze doniesienia, czy raczej chcą uzupełnić wiedzę na konkretny temat?
- Czy chcą uzyskać informacje o konkretnym produkcie lub usłudze, które pomogłyby im zdecydować, czy je kupić, czy też nie?
- Czy potrzebują nawiązania kontaktu z właścicielem witryny? Jeśli tak — czy mogą to zrobić osobiście (co może wymagać podania godzin otwarcia oraz mapy z trasą dojazdu)? A może potrzebują adresu e-mail lub numeru telefonu?

CO STARAJĄ SIĘ UZYSKAĆ UŻYTKOWNICY WITRYNY?

Jest raczej mało prawdopodobne, byśmy byli w stanie stworzyć listę wszystkich powodów, dla których ludzie odwiedzają naszą witrynę, niemniej interesują nas jedynie kluczowe zadania i motywacje. Posłużą nam one jako pomoc podczas projektowania witryny.

Rozważania możemy zacząć od przygotowania listy powodów, dlaczego inne osoby mogłyby odwiedzać naszą witrynę. Następnie możemy przypisać fikcyjnym użytkownikom, utworzonym na poprzedniej stronie, listę zadań.

GRZEGORZ kupił rakietę tenisową kilka lat temu, a teraz chce kupić jeszcze jedną dla swojej dziewczyny.

MARYSIA dowiedziała się z prasy o nowej usłudze opieki nad zwierzętami domowymi i chciałaby się dowiedzieć, czy zaspokoi jej potrzeby.

JANEK miał niemile przygody w hotelu w Warszawie i chciałby złożyć skargę.

JUREK chciałby studiować architekturę i pragnie się czegoś dowiedzieć o oferowanych kursach przygotowawczych.

EWA jest grafikiem komputerowym i chciałaby przyjrzeć się pracom fotografa, zanim zdecyduje się dać mu zlecenie.

JAKICH INFORMACJI POTRZEBUJĄ UŻYTKOWNICY?

Wiemy, kto odwiedza naszą witrynę i dlaczego, teraz musimy ustalić, jakich informacji użytkownicy potrzebują, aby pozwolić im szybko i efektywnie realizować ich cele.

Być może warto udostępnić dodatkowe, pomocne informacje, które, według nas, mogą się przydać użytkownikom.

Przyjrzyjmy się wszystkim wynotowanym powodom, dla których użytkownicy będą odwiedzali naszą witrynę, i określmy, czego im potrzeba, by szybko zrealizować swoje cele.

Możemy określić priorytety informacji, począwszy od zagadnień kluczowych, a skończywszy na informacjach drugorzędnych lub nawet mało ważnych.

Dzięki udostępnieniu użytkownikom informacji, których poszukują, możemy sprawić, że nasza witryna wyda się im bardziej istotna i przydatna.

Dlatego będziemy mieli więcej okazji, by przekazać użytkownikom wszelkie dodatkowe informacje, które — jak sądzimy — mogłyby się im przydać (lub pokazać inne produkty czy usługi, które chcemy promować).

Jeśli użytkownicy uznają, że witryna ma dla nich małe znaczenie, gdyż nie spełnia ich potrzeb, to najprawdopodobniej poszukają jakiejś innej.

Jest kilka pytań, które można sobie zadać, próbując określić, jakie informacje publikować w swojej witrynie.

KLUCZOWE INFORMACJE

- Czy użytkownicy będą znali prezentowaną tematykę (lub markę), czy trzeba będzie ją przedstawić?
- Czy będą znali oferowane produkty, usługi lub informacje, czy potrzebują dodatkowych informacji na ich temat?
- Jakie są najistotniejsze cechy oferowanych usług lub produktów?
- Co szczególnego odróżnia naszą ofertę od innych witryn oferujących podobne produkty lub usługi?
- Czy po zrealizowaniu celów, które sprowadziły użytkowników na naszą witrynę, zgłaszają oni jakieś pytania dotyczące prezentowanych zagadnień; czy któreś z takich pytań często się powtarza?

JAK CZEŚTO UŻYTKOWNICY BĘDĄ ODWIEDZAĆ NASZĄ WITRYNĘ?

Niektóre witryny czerpią korzyści z tego, że ich zawartość jest aktualizowana częściej niż w innych. Niektóre informacje (takie jak doniesienia prasowe) ulegają bezustannym modyfikacjom, natomiast inne są stosunkowo niezmienne.

Witryna poświęcona najnowszym trendom w świecie mody będzie musiała być aktualizowana znacznie częściej niż witryna reklamująca usługi, z których nie korzysta się tak często (takie jak naprawy kanalizacji lub kładzenie płytka).

Kiedy witryna zostanie już utworzona, to jej częsta aktualizacja może zajmować wiele czasu i wymagać użycia wielu zasobów.

Określenie, jak często ludzie będą odwiedzali naszą witrynę, może stanowić sugestię, jak często powinniśmy aktualizować jej zawartość.

Pomocne może być przygotowanie harmonogramu aktualizacji witryny (zamiast przeprowadzania aktualizacji w sposób doraźny).

Zauważymy zapewne, że niektóre działały witryny w większym stopniu niż inne korzystają na częstym aktualizowaniu zawartości.

Oto kilka pytań, które mogą nam pomóc w określeniu, jak często aktualizować zawartość witryny...

DOBRA LUB USŁUGI

- Jak często te same osoby wracają na stronę, by dokonać kolejnego zakupu?
- Jak często zmienia się oferowany asortyment (lub usługa)?

INFORMACJE

- Jak często zmienia się tematyka, której jest poświęcona witryna?
- Jaki procent użytkowników witryny powracałby na nią, by uzyskać zaktualizowane informacje, a jakiej części użytkowników wystarczy jedna wizyta?

MAPY WITRYNY

Teraz, skoro już wiemy, co ma się pojawić w witrynie, możemy zacząć organizować jej działy i strony.

Naszym celem jest stworzenie diagramu stron, który będziemy mogli wykorzystać do określenia struktury witryny. Nosi on nazwę **mapy witryny** i pokazuje, w jaki sposób można pogrupować tworzące ją strony.

Podczas określania zawartości poszczególnych stron można skorzystać z techniki nazywanej **sortowaniem kart**.

Polega ona na umieszczeniu każdej z informacji, których użytkownik może potrzebować, na osobnej karteczce, a następnie na grupowaniu karteczek zawierających informacje powiązane ze sobą.

Każda taka grupa reprezentuje jedną stroną, a w przypadku dużych witryn także strony można podobnie pogrupować, tworząc w ten sposób działy.

Grupy informacji są następnie przekształcane na diagram nazywany mapą witryny.

Czasami pomocne może być poproszenie osób stanowiących docelowych użytkowników witryny o pomoc w pogrupowaniu informacji.

Mapa witryny zazwyczaj rozpoczyna się od strony głównej. Dodatkowo, jeśli witryna jest duża i podzielona na działy, każdy z takich działów może wymagać własnej strony głównej, która będzie zawierać łącza do wszystkich stron danego działu.

Większość sklepów internetowych ma strony główne dla poszczególnych typów oferowanych produktów, a na nich znajdują się łącza do stron poświęconych konkretnym produktom.

Jeśli jakaś informacja musi się pojawić na więcej niż jednej stronie, to konieczne będzie jej powielenie.

Strony (lub grupy stron) będą informowały nas o tym, w jaki sposób użytkownicy poruszają się po naszej witrynie.

Konieczne należy pamiętać o celach, do których dążą użytkownicy witryny.

Warto wspomnieć o tym, że właściciel witryny może zorganizować informacje w sposób odbiegający od oczekiwania użytkowników. Ważne jest jednak, by witryna odpowiadała ogólnemu zrozumieniu prezentowanych zagadnień (a nie sposobowi, w jaki rozumie je jej właściciel).

PRZYKŁADOWA MAPA WITRYNY



SZKIELETY STRON

Szkielet strony to prosty szkic przedstawiający kluczowe informacje, które mają się znaleźć na każdej ze stron witryny. Przedstawia on hierarchię informacji oraz wskazuje, ile miejsca mogą one zajmować.

Wielu projektantów określa elementy, które muszą się znaleźć na poszczególnych stronach witryny, i zaczyna pracę do przygotowania szkieletów stron (ang. *wireframes*). Polega to na narysowaniu lub zacieniowaniu obszarów, w których będą umieszczone poszczególne elementy strony (takie jak logo, główne elementy nawigacyjne, nagłówki, główne fragmenty treści, dane zalogowanego użytkownika itd.).

Dzięki utworzeniu takiego szkieletu strony możemy zagwarantować, że znajdą się na niej wszystkie niezbędne informacje.

Przykład przedstawiony z prawej strony został stworzony w programie Photoshop przy wykorzystaniu szablonów dostępnych w systemie 960.gs.

Podczas przygotowywania szkieletów stron nie należy zajmować się kolorystyką witryny, doborem czcionek ani elementów graficznych. Należy się skoncentrować na tym, jakie informacje muszą się znaleźć na poszczególnych stronach, oraz na stworzeniu hierarchii wizualnej, która pozwoli wyróżnić najważniejsze elementy stron.

Przygotowanie takich szkieletów ułatwia projektowanie witryny, gdyż wiadomo, jakie informacje muszą się znaleźć na poszczególnych stronach, zanim

zaczniemy się zastanawiać, jak one mają wyglądać. Bardzo pomocne może być pokazanie takich szkieletów witryny klientowi, zanim przedstawimy mu jej projekt. Dzięki temu klient upewni się, że witryna udostępnia wszelkie niezbędne funkcje oraz informacje.

Jeśli przedstawimy klientowi jedynie projekt witryny, zazwyczaj skoncentruje się on na jej wyglądzie. A to z kolei oznacza, że aż do momentu stworzenia witryny klient może pominąć wszelkie wątpliwości związane z jej funkcjonalnością.

Szkielety stron można szkicować na papierze lub korzystać z odpowiednich programów komputerowych (takich jak Illustrator lub InDesign).

Dostępne są także internetowe narzędzia do tworzenia szkieletów stron, m.in. <http://gomockingbird.com/>, <http://lovelycharts.com/>.

PRZYKŁADOWY SZKIELET STRONY

LOGO

Strona główna Produkty Usługi O nas Kontakt

Fotografia produktu

Kluczowe informacje o produkcie
Kluczowe informacje o produkcie
Kluczowe informacje o produkcie
Kluczowe informacje o produkcie

Przycisk do kupowania

Doniesienia

Referencje

Sprzedaż

Stopka, informacje prawne, polityka prywatności, kontakt

PRZEKAZYWANIE KOMUNIKATU POPRZEZ PROJEKT

Podstawowym celem każdego rodzaju projektu wizualnego jest dostarczenie jakiegoś przekazu. Odpowiednie zorganizowanie oraz wyznaczenie priorytetów informacji na stronie ułatwia użytkownikom ustalenie ich znaczenia oraz kolejności, w jakiej należy je czytać.

ZAWARTOŚĆ

Strony WWW często mają wiele treści do przekazania, np. witryna gazety może zawierać doniesienia, które nie zostały umieszczone na wszystkich stronach jej papierowego wydania:

- nagłówek lub logo;
- łączka do nawigowania po witrynie;
- łączka do stron o podobnej treści lub innych popularnych artykułów;
- narzędzia do logowania lub rejestracji;
- mechanizmy do komentowania;
- informacje o prawach autorskich;
- łączka do informacji o polityce prywatności, warunkach korzystania, informacje o reklamach, kanał RSS, informacje o subskrypcji.

Przy tak dużej ilości informacji umieszczonych na jednej stronie, aby zakomunikować jej prześanie i pomóc użytkownikom odszukać to, czego potrzebują, projektant musi odpowiednio **zorganizować** informacje i określić ich **priorytety**.

PRIORYTETY

Gdyby wszystkie informacje zamieszczone na stronie były prezentowane w taki sam sposób, to zrozumienie ich byłoby znacznie trudniejsze (trudno byłoby określić jej główny przekaz).

Jednak poprzez odpowiednie **wyróżnienie** fragmentów strony projektanci mogą skierować na nie uwagę użytkowników (albo odciągnąć ją od tych fragmentów).

Projektanci tworzą coś, co jest określane jako **hierarchia wizualna**. Jej celem jest ułatwienie użytkownikom skoncentrowania się na kluczowym przekazie, który przyciągnie ich zainteresowanie i skieruje je na kolejne treści.

Zagadnienia związane z hierarchią wizualną zostały przedstawione na stronach 466 – 467.

ORGANIZACJA

Grupowanie powiązanych ze sobą treści w **bloki** lub **fragmenty** sprawia, że strony wyglądają na prostsze (i łatwiejsze do zrozumienia).

Użytkownicy powinni być w stanie określić przeznaczenie każdego z takich bloków bez wgłębiania się w poszczególne jego elementy.

Dzięki przedstawieniu pewnych typów informacji w **wizualnie** podobny sposób (np. przy użyciu tego samego stylu do wyświetlania wszystkich łączys lub przycisków) użytkownicy nauczą się kojarzyć go z treściami konkretnego typu.

Zagadnienia związane z grupowaniem i podobieństwem zostały opisane na stronach 468 – 469.

Email marketing software for designers and their clients

Send beautiful email campaigns, track the results and manage your subscribers. Earn a tidy profit by letting your clients send at prices you set.

TRY IT FOR FREE

VIEW FEATURES



Create & Send Beautiful Emails

Design great looking emails using your own tools, or create templates and let your clients send their own.



Powerful Analytics

Actionable reports that go beyond opens and clicks. Track your email related conversions and sales.



Manage Lists & Subscribers

We handle signups, unsubscribes and bounces automatically. Easily create targeted segments of subscribers.



Mark-up, Resell and Profit

White-label and rebrandable, you set the price your clients pay and we'll send through your profit each month.

Send unlimited email campaigns for as little as

\$15
PER MONTH

[See Plans & Pricing](#)

Used by leading designers and companies everywhere



Przyjrzyjmy się przykładowi, który demonstruje, w jaki sposób można wykorzystać projekt, by efektywnie pokazać usługi oferowane przez firmę.

HIERARCHIA WIZUALNA

Uwagę osoby odwiedzającej tę stronę momentalnie przyciąga obraz przedstawiający usługi oferowane przez firmę oraz nagłówek, który je wyjaśnia. Wielkość oraz kolorowe tło dodatkowo pokazują, że właśnie te elementy stanowią główny przekaz strony.

Jeśli zaprezentowane usługi zainteresują odwiedzającego, to poniżej znajdzie szczegółowe informacje o zapewnianych możliwościach, cenie oraz aktualnych użytkownikach.

GRUPOWANIE

Na przedstawionej stronie można wyróżnić kilka bloków informacji.

Na samej górze znajdują się logo oraz elementy nawigacyjne. Poniżej zostały umieszczone informacje o usługach oferowanych przez firmę.

Jeszcze niżej zostały umieszczone trzy odrębne grupy przedstawiające możliwości usług, ich ceny oraz użytkowników, którzy już z nich korzystają.

PODOBIEŃSTWA

Na stronie można wskazać kilka przykładów podobieństw.

Cztery punkty (widoczne w lewym dolnym rogu strony) zostały przedstawione w podobny sposób — mają wizualnie spójne nagłówki oraz ikony.

Wszystkie łącza umieszczone w treści strony są niebieskie, więc użytkownicy bez trudu się zorientują, że można je kliknąć.

HIERARCHIA WIZUALNA

Większość internautów nie czyta odwiedzanych stron w całości, raczej je przegląda w poszukiwaniu informacji. Można wykorzystać kontrast, by określić hierarchię wizualną, która ułatwi użytkownikom znalezienie kluczowych komunikatów i tego, czego szukają.

ROZMIAR

Duże elementy będą przyciągać uwagę osób oglądających stronę. Z tego powodu warto zadbać o to, by nagłówki oraz kluczowe punkty strony były stosunkowo duże.

Lorem ipsum
dolor sit amet,
consectetur
adipiscing elit. Lorem
ipsum dolor sit amet,
consectetur
adipiscing elit.

KOLOR

Kolor tekstu oraz kolor tła mogą skierować uwagę czytelnika na kluczowe treści. Zazwyczaj ludzie w pierwszej kolejności patrzą na jaśniejsze fragmenty strony.

Lorem ipsum
dolor sit amet,
consectetur
adipiscing elit. Lorem
ipsum dolor sit amet,
consectetur
adipiscing elit.

STYL

Element może mieć tę samą wielkość oraz kolor jak otaczająca go zawartość strony, lecz inny styl, który sprawi, że będzie się on zdecydowanie wyróżniał.

Lorem ipsum
dolor sit amet,
consectetur
adipiscing elit. Lorem
ipsum dolor sit amet,
consectetur
adipiscing elit.

Hierarchia wizualna odnosi się do kolejności, w jakiej nasze oczy postrzegają to, co widzimy. Tworzy się ją poprzez wprowadzanie **wizualnego kontrastu** pomiędzy wyświetlonymi elementami. Elementy o wyższym kontraście są rozpoznawane i przetwarzane w pierwszej kolejności.



OBRAZY

Obrazy tworzą wysoki kontrast wizualny i często właśnie na nie osoby przeglądające stronę spoglądają w pierwszej kolejności. Można ich używać, by skierować uwagę odwiedzających na konkretny komunikat umieszczony na stronie. W niektórych przypadkach odpowiedni obraz może przekazać więcej niż cała strona tekstu.

Efekty prawidłowo zaprojektowanej hierarchii wizualnej są w znacznej mierze podświadome. Uzyskanie dobrej hierarchii wizualnej wymaga równowagi; jeśli żaden elementy strony się nie wyróżnia, zostanie ona odebrana jako nudna, z kolei jeśli zbyt wiele jej aspektów będzie konkurowało o uwagę, to odwiedzający mogą mieć trudności z określeniem, które z przekazywanych komunikatów są kluczowe. W przedstawionym przykładzie hierarchia wizualna jest oczywista, co ułatwia zaspokojenie potrzeb osób oglądających stronę.

GRUPOWANIE I PODOBIEŃSTWA

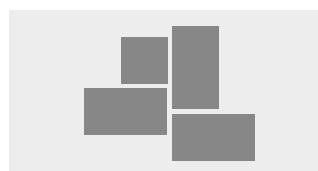
Starając się zrozumieć projekt graficzny, zazwyczaj próbujemy organizować elementy wizualne w grupy. Takie grupowanie informacji może ułatwić zrozumienie projektu. Poniżej przedstawione zostały kluczowe sposoby grupowania.

ZBLIŻENIE



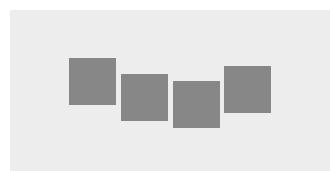
Kiedy kilka elementów zostanie umieszczonych blisko siebie, postrzega się je jako mocniej powiązane ze sobą niż elementy umieszczone w większych odległościach od siebie. (Istnieje także możliwość zagnieźdzania jednych grup informacji wewnętrznych innych).

ZAMKNIĘCIE



Zazwyczaj, kiedy stykamy się ze złożonym układem elementów, poszukujemy w nim pojedynczego lub rozpoznawalnego kształtu albo formy. Ze względu na bliskość lub wyrównanie elementów strony można wokół nich wyznaczyć rzeczywisty lub umowny prostokąt.

KONTYNUACJA



Jeśli elementy zostaną umieszczone w linii lub na krzywej, będą postrzegane jako bardziej powiązane ze sobą niż elementy, które nie są rozmieszczone w tak uporządkowany sposób. Efekt ten można wykorzystać, by skierować czytelnika z jednego miejsca strony w inne.

ODSTĘP



Polega na umieszczaniu powiązanych ze sobą elementów bliżej siebie i pozostawianiu większych odstępów pomiędzy elementami, które nie są ze sobą powiązane.

KOLOR



Umieszczenie kilku elementów na jednolitym tle może podkreślić ich wzajemne powiązanie.

OBRAMOWANIA



Wokół krawędzi grupy elementów albo pomiędzy nimi a ich otoczeniem można narysować linię.

Oczywiście zauważamy podobieństwa występujące w projekcie, a rzeczy podobne są traktowane jako mocniej ze sobą powiązane niż te, które podobne nie są. Powtórzenia podobnego koloru, rozmiaru, kierunku, kształtu czy podobnej tekstury albo czcionki sugerują, że analogiczne do siebie elementy mają zbliżoną wagę.

Recenzje książek

Raw Creation

John Maizels

Raw Creations jest najlepszą książką o sztuce artystycznych outsiderów i stanowi niezastąpiony przewodnik po sztuce samouków tego stulecia oraz fascynujący dowód ludzkiej kreatywności. Rozdział *Wonders of the World* w większości jest poświęcony Rock Garden Neki Chanda i zawiera wiele kolorowych fotografii.

[Kup na Amazon](#)

Nek Chand Outsider Art

Lucienne Peiry, Philippe Lespinasse

Ta publikacja przedstawia historię Neki Chanda, opisuje jego życie i zabiera czytelnika w kolorową podróż przez stworzony przez niego Rock Garden, największy na świecie przykład kreacji zaliczanej do gatunku environmental art.

[Kup na Amazon](#)

Fantasy Worlds

Deidi von Schaewen, John Maizels

Przedstawia najbardziej niezwykłe, kolorowe i poetyckie dzieła stworzone przez człowieka, spośród których część nigdy dotąd nie została zamieszczona w żadnej publikacji drukowanej. Książka ta prezentuje wybrane budowle zaliczane do klasycznej architektury fantastycznej, takie jak Palais Ideal kolo Lyonu oraz, bo jakżeby inaczej, Rock Garden stworzony przez Nekę Chanda w Chandigarh w Indiach.

[Kup na Amazon](#)

The Rock Garden

M.S. Aulakh

Ta niewielka, czarno-biała książka jest komentarzem M.S. Aulakha oraz hohdem na cześć Rock Garden. Publikacja ta nie jest powszechnie dostępna, jednak, od czasu do czasu, można znaleźć jej używane egzemplarze.

[Kup na Amazon](#)

KONSEKWENCJA

W przedstawionym obok przykładzie wszystkie recenzje książek wyglądają podobnie — wykorzystano spójne sposoby prezentacji nagłówków, autorów oraz łączy do sklepu internetowego. Wystarczy przeczytać jeden blok, by móc określić znaczenie innych elementów strony, prezentowanych w taki sam sposób.

NAGŁÓWKI

Poprzedzenie grupy informacji nagłówkiem może w zrozumiały sposób wskazać użytkownikowi, czy zamieszczone poniżej treści będą miały dla niego jakieś znaczenie, czy nie. Jeśli nie będą — użytkownik może je w całości pominąć. Rozwiążanie to może także pomóc użytkownikom posługującym się czytnikami ecranowymi — często mają oni możliwość odczytania wszystkich nagłówków umieszczonych na stronie.

Każdy widoczny fragment może mieć własną hierarchię; co wyraźnie widać na powyższym przykładzie, w którym każda książka jest przedstawiona w osobnej podsekcji i zawiera tytuł, autora oraz łącze do sklepu.

PROJEKTOWANIE ELEMENTÓW NAWIGACYJNYCH

Elementy nawigacyjne witryny nie tylko pomagają użytkownikom dotrzeć tam, gdzie chcą, lecz także ułatwiają im zrozumienie, czego ona dotyczy oraz w jaki sposób jest zorganizowana. Dobre elementy nawigacyjne są zazwyczaj tworzone zgodnie z poniższymi zasadami...

ZWIĘZŁOŚĆ

Elementy nawigacyjne powinny być takie, by użytkownik mógł je odczytać łatwo i szybko. Dobrym pomysłem jest ograniczenie liczby opcji w menu, tak by nie przekraczała ośmiu łącz. Mogą one prowadzić do stron głównych poszczególnych działów witryny, a te — do kolejnych stron.

PRZEJRZYSTOŚĆ

Użytkownicy powinni być w stanie przewidzieć zawartość strony, zanim klikną prowadzące na nią łącze. Jeśli to możliwe, łącza powinny mieć postać jednego, opisowego słowa, a nie kilku wyrazów.

SELEKTYWNOŚĆ

Główne elementy nawigacyjne powinny odpowiadać wyłącznie działom lub zawartości witryny. Funkcjonalności, takie jak logowanie i wyszukiwanie, informacje prawne, np. warunki korzystania, oraz podobne elementy najlepiej umieszczać w innych fragmentach stron.

[Strona główna](#) [Profile twórców](#) [Wystawy i imprezy](#) [Galerie](#) [Książki i czasopisma](#)
[O tej witrynie](#) [Kontakt z nami](#) [Logowanie](#) [Rejestracja](#) [Warunki korzystania](#) [Polityka prywatności](#)



[Strona główna](#) [Profile twórców](#) [Wystawy](#) [Galerie](#) [Publikacje](#) [O nas](#) [Kontakt](#)



Duże witryny mogą posiadać główne elementy nawigacyjne oraz elementy nawigacyjne drugiego, a nawet trzeciego stopnia. Główne elementy nawigacyjne zazwyczaj są umieszczane w górnej części stron, w poziomie, od strony lewej do prawej, lub w pionie — wzdłuż lewej krawędzi strony. Elementy nawigacyjne drugiego stopnia mogą być umieszczane poniżej elementów głównych

albo wzdłuż krawędzi strony. Elementy nawigacyjne trzeciego stopnia są zazwyczaj umieszczane w stopce strony. Menu nie jest jedynym sposobem, w jaki użytkownicy poruszają się po witrynie. Do tego celu wykorzystywane są także łącza umieszczane w treści poszczególnych stron. Niektóre witryny udostępniają również możliwość przeszukiwania ich zawartości.

WIĘCEJ W SIECI

Na stronie poświęconej tej książce można znaleźć informacje o tym, w jaki sposób zaimplementować mechanizm przeszukiwania witryny przy użyciu narzędzia Google Search.

KONTEKST

Dobre elementy nawigacyjne zapewniają kontekst. Pozwalają użytkownikom zorientować się, w jakim miejscu witryny aktualnie przebywają. Do tego celu doskonale nadaje się inny kolor lub jakiś wizualny znacznik.

INTERAKTYWNOŚĆ

Każde łącze powinno być na tyle duże, by można je było kliknąć, a jego wygląd powinien się zmieniać po umieszczeniu na nim wskaźnika myszy i kliknięciu. Łącza powinny się także wizualnie odróżniać od innych elementów stron.

SPÓJNOŚĆ

Im większa liczba stron tworzących witrynę, tym więcej elementów nawigacyjnych. Choć elementy nawigacyjne drugiego stopnia będą się zmieniać na poszczególnych stronach, najlepszym rozwiązaniem jest zachowanie głównych elementów nawigacyjnych w dokładnie takiej samej postaci w całej witrynie.

Strona główna Profile twórców Wystawy Galerie Publikacje O nas Kontakt



Strona główna Profile twórców Wystawy Galerie Publikacje O nas Kontakt



PODSUMOWANIE

PROCES I PROJEKT

- ▶ Bardzo ważne jest określenie, kim są docelowi użytkownicy witryny, dlaczego będą ją odwiedzać, jakie informacje pragną na niej znaleźć oraz jak często będą powracać.
- ▶ Mapy witryny pozwalają zaprojektować jej strukturę.
- ▶ Szkielety stron pozwalają zorganizować informacje, które powinny się na nich znaleźć.
- ▶ Projekt jest ściśle powiązany z komunikacją. Odpowiednia hierarchia wizualna ułatwia użytkownikom zrozumienie komunikatu, który chcemy im przekazać.
- ▶ Różne informacje zamieszczane na stronach można odróżniać do siebie przy użyciu wielkości, kolorów oraz stylu.
- ▶ Poprzez grupowanie informacji oraz stosowanie podobieństw możemy uprościć przekaz.

19

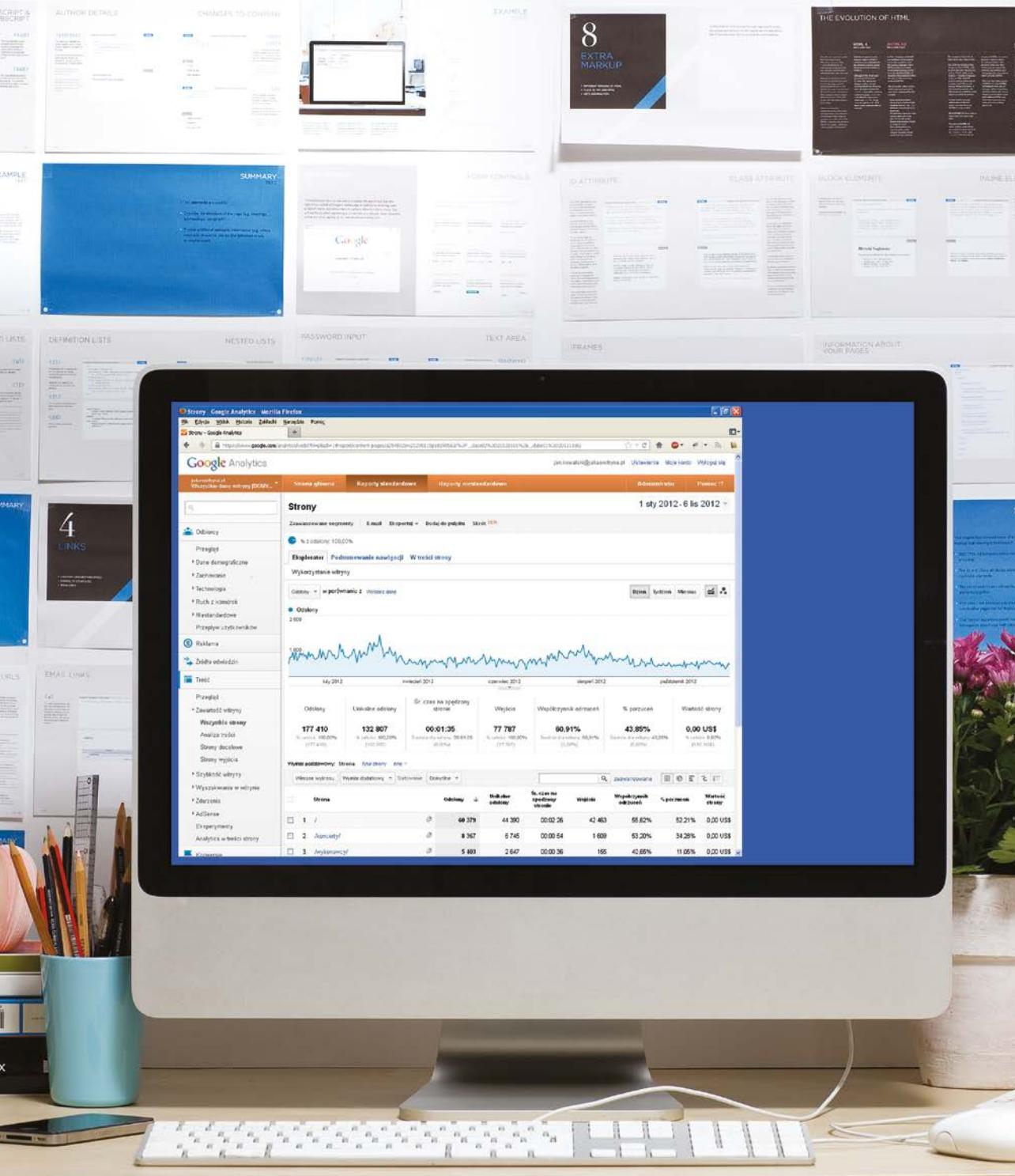
INFORMACJE PRAKTYCZNE

- ▶ Optymalizacja stron pod kątem wyszukiwania.
- ▶ Stosowanie analiz w celu zrozumienia odwiedzających.
- ▶ Umieszczanie witryny w internecie.

W ramach podsumowania niniejszej książki w tym rozdziale zamieszczono praktyczne informacje, które pozwolą Czytelnikowi uruchomić cieszącą się powodzeniem stronę internetową.

Można znaleźć całe książki poświęcone wyłącznie pojedynczym zagadnieniom opisywanym w tym rozdziale. Tutaj jednak zamieszczone zostały jedynie wprowadzenia wyjaśniające, czego te zagadnienia dotyczą, oraz wskazówki sugerujące, co należy wziąć pod uwagę. W tym rozdziale zostaną przedstawione kwestie:

- Podstawy optymalizacji stron pod kątem wyszukiwania.
- Wykorzystanie analiz w celu zrozumienia, jak odwiedzający korzystają z już udostępnionych witryn.
- Umieszczanie witryny w internecie.



OPTYMALIZACJA POD KĄTEM WYSZUKIWANIA (SEO)

Optymalizacja pod kątem wyszukiwania (ang. *Search Engine Optimization* — SEO) jest bardzo rozległym zagadnieniem, któremu poświęcono wiele książek. Informacje zamieszczone na kilku kolejnych stronach pozwolą zrozumieć kluczowe kwestie z nią związane i poprawić pozycję witryny w wyszukiwarkach.

PODSTAWY

Optymalizacja pod kątem wyszukiwania (określana powszechnie jako SEO) to próba zapewnienia, by witryna pojawiała się jak najbliżej początku wyników zwracanych przez wyszukiwarki, wtedy gdy użytkownicy poszukują tematu, którego ona dotyczy.

Kluczowym aspektem SEO jest próba określenia terminów, które użytkownicy mogą podawać w wyszukiwarkach, by odszukać witrynę, a następnie umieszczenie ich w odpowiednich miejscach witryny w celu zwiększenia szans, że wyszukiwarki pokażą łącze do niej w wygenerowanych wynikach.

Aby określić kolejność prezentowanych wyników, wyszukiwarki analizują nie tylko treści zamieszczone na stronie. Uwzględniają także to, jak wiele innych witryn zawiera prowadzące do niej łącza (oraz jak ważne one są). Z tego powodu zagadnienia SEO często dzielą się na dwa obszary: techniki stosowane w witrynie oraz poza nią.

TECHNIKI STOSOWANE W WITRYNIE

Zaliczają się do nich metody, których można używać we własnej witrynie w celu poprawienia jej pozycji w wynikach wyszukiwarek. Bazują one na poszukiwaniu słów kluczowych, które użytkownicy najprawdopodobniej będą wpisywali w wyszukiwarkach, poszukując naszej witryny, a następnie umieszczenia ich w kodzie HTML stron, by poinformować przeglądarki o tematyce witryny. Dla wyszukiwarek duże znaczenie mają teksty znajdujące się na stronach witryny, zatem bardzo istotne jest, by znalazły się w nich terminy, które mogą być używane podczas prób odnalezienia naszej witryny. Istnieje siedem kluczowych miejsc, w których będziemy umieszczać takie słowa kluczowe. Dodatkowo po umieszczeniu odpowiednich tekstów w atrybucie alt obrazów ułatwiamy wyszukiwarkom zrozumienie treści grafik prezentowanych na stronie.

TECHNIKI STOSOWANE POZA WITRYNĄ

Równie duże znaczenie jak techniki stosowane w witrynie ma umieszczenie odwołań do niej w innych witrynach. Wyszukiwarki pomagają sobie w określeniu jej pozycji poprzez określenie liczby innych stron, które się do niej odwołują. Wyszukiwarki są szczególnie zainteresowane witrynami, które mają taką samą zawartość jak nasza. Jeśli np. w naszej witrynie zajmujemy się sprzedawaniem przynęt wędkarskich, to umieszczenie łącza do niej na stronie zakładu fryzjerskiego zapewne nie zostanie potraktowane jako równie istotne jak łącze na forum wędkarstwa spinningowego. Wyszukiwarki analizują także słowa umieszczone w łączach — pomiędzy otwierającym znacznikiem `<a>` i zamkającym znacznikiem ``. Jeśli tekst łącza zawiera słowa kluczowe (zamiast zwykłego „kliknij tutaj” lub nazwy witryny), to zostanie on potraktowany jako bardziej znaczący. Słowa umieszczone w łączach do naszej witryny powinny się także pojawiać w treści stron, do których prowadzą.

TECHNIKI SEO STOSOWANE W WITRYNIE

Na każdej stronie internetowej jest kilka istotnych miejsc, w których można umieszczać słowa kluczowe (podawane przez użytkowników w wyszukiwarkach), tak aby łatwiej było ją odnaleźć.

1. TYTUŁ STRONY

Tytuł strony jest wyświetlany na samej górze okna przeglądarki lub w nazwie karty. Jest on podawany w elemencie `<title>` umieszczanym wewnątrz elementu `<head>`.

2. URL

LUB ADRES WITRYNY

Nazwa pliku stanowi element adresu URL. Zawsze, gdy to tylko możliwe, warto stosować słowa kluczowe jako nazwy plików.

3. NAGŁÓWKI

Jeśli słowa kluczowe zostaną umieszczone w nagłówkach (elementach `<h1>`), to wyszukiwarki będą wiedzieć, że strona jest w całości poświęcona jakiejś tematyce, i przypiszą im większe znaczenie, niż gdyby stanowiły one zwyczajny tekst.

4. TEKST

Gdzie to tylko możliwe, warto powtarzać słowa kluczowe w zwyczajnym tekście strony i to przynajmniej dwa, trzy razy. Nie należy ich jednak używać zbyt często, gdyż tekst musi być zrozumiały dla czytelnika.



5. TEKSTY W ŁĄCZACH

Warto używać słów kluczowych w tekstu łączy prowadzących do innych stron witryny (zamiast standardowych zwrotów typu „kliknij tutaj”).

6. TEKST ALTERNatywny OBRAZÓW

Wyszukiwarki oczekują, że twórcy stron będą podawali precyzyjne opisy obrazów, umieszczając je w atrybutach `alt`. Rozwiążanie to umożliwia także wyświetlanie tych obrazów w wynikach wyszukiwania grafik.

7. OPIS STRONY

Opis strony także jest umieszczany wewnątrz elementu `<head>`, przy czym podaje się, go używając znacznika `<meta>`. Powinien on być zdaniem opisującym zawartość strony. (Opisy te nie są widoczne w oknie przeglądarki, jednak czasami przeglądarki wyświetlają je w wynikach wyszukiwania).

Nie należy starać się przechytrzyć wyszukiwarek! Można zostać za to ukarany. Nigdy np. nie powinno się umieszczać na stronie tekstu w tym samym kolorze co kolor tła — wyszukiwarki potrafią to wykryć.

JAK OKREŚLAĆ SŁOWA I FRAZY KLUCZOWE?

Kiedy myślimy o optymalizacji pod kątem wyszukiwania, jednym z najtrudniejszych problemów może być określenie słów kluczowych, których powinniśmy użyć. Istnieje sześć kroków, które pomagają w określeniu prawidłowych słów i fraz kluczowych.

1. BURZA MÓZGÓW

Warto wypisać listę słów, które ktoś może wpisać w wyszukiwarce Google, by odnaleźć naszą stronę. Trzeba przy tym pamiętać, by uwzględnić różne zagadnienia, produkty i usługi oferowane na stronie.

Często bardzo pomocne jest zapytanie innych o to, jakich słów by użyli, aby odnaleźć naszą witrynę, gdyż osoby, które słabiej znają prezentowane zagadnienia, mogą używać innych terminów niż my (w szczególności nie będą się raczej posługiwać terminologią fachową).

Jeśli zagadnienia prezentowane na stronie są opisywane całymi wyrażeniami, to także one (a nie pojedyncze słowa) mogą się znaleźć na naszej liście.

2. ORGANIZACJA

Słowa kluczowe należy pogrupować w listy odnoszące się do konkretnych działów lub kategorii witryny.

Jeśli np. nasza witryna jest sklepem ze zwierzętami domowymi, kategorie te mogą odpowiadać różnym gatunkom zwierząt (takim jak psy, koty, króliki).

W przypadku dużych witryn można kontynuować ten podział, tworząc podkategorie (np. odrębne grupy różnych rodzajów karmy dla zwierząt).

3. POSZUKIWANIA

Dostępnych jest kilka narzędzi pozwalających podawać słowa kluczowe, a następnie sugerujących dodatkowe słowa, których można by użyć. Oto kilka z nich:

adwords.google.pl/select/
KeywordToolExternal/
(używając tego narzędzia, należy raczej wybierać tryb dopasowywania „Ścisłe”, a nie „Przybliżone”)
www.wordtracker.com
www.keyworddiscovery.com

Po wyświetleniu sugerowanych słów kluczowych te, które będą pasować, można dodać do stworzonej wcześniej listy. (Wymienione narzędzia będą zapewne sugerować terminy niemające większego sensu w kontekście naszej witryny, trzeba zatem pominąć wszystkie, które nie wydają się właściwe).

4. PORÓWNANIE

Jest bardzo mało prawdopodobne, by nasza witryna pojawiła się na samym początku wyników wyszukiwania dla każdego z używanych słów kluczowych. W szczególności dotyczy to tych zagadnień, w których konkurencja jest bardzo duża. Im więcej witryn zostało już zoptymalizowanych pod kątem danego słowa kluczowego, tym trudniej nam będzie umieścić witrynę w czołówce wyników zwracanych podczas wyszukiwania tego terminu.

Niektóre witryny zajmujące się badaniem słów kluczowych potrafią zwracać informacje o liczbie internautów, którzy podczas wyszukiwania użyli konkretnego słowa, co pozwala się zorientować, jaka konkurencja panuje wśród witryn, które je wykorzystują.

Można także wykorzystać zaawansowane mechanizmy wyszukiwarki Google, by przeszukać jedynie tytuły witryn. To pozwala się zorientować, jak wiele witryn używa konkretnego terminu w tytułach swoich stron. (Wyższa liczba stron będzie świadczyć o większej konkurencji o dane słowo kluczowe).

5. UDOSKONALENIE

Teraz musimy zdecydować, na których słowach kluczowych będziemy się koncentrować. Należy wybrać te najistotniejsze dla poszczególnych działań witryny.

Jeśli znajdziemy wyrażenie, które jest bardzo ważne, lecz jest często stosowane w innych witrynach, to mimo wszystko należy go użyć. Aby poprawić szanse odnalezienia witryny, trzeba sprawdzić, czy istnieją inne słowa kluczowe, które można by dodać do tego wyrażenia. Jeśli np. oferowane informacje lub usługi są powiązane z konkretną lokalizacją, może się okazać, że dodanie jej do słów kluczowych ułatwi odnalezienie witryny.

Jeśli nasza witryna oferuje pokrywanie dachów dachówkami łupkowymi w Polsce, to lepsze dla niej będzie przyciągnięcie 100 internautów z Polski niż 10 tysięcy z USA.

6. ZASTOSOWANIE

Kiedy już udoskonalimy listę słów kluczowych i określmy, które z nich są najczęściej używane i które są najistotniejsze dla naszej witryny, trzeba będzie dokonać ostatecznego wyboru i zacząć umieszczać je na stronach.

Warto wybrać od trzech do pięciu słów lub wyrażeń kluczowych, które można dopasować do poszczególnych stron witryny, i zacząć je na nich umieszczać.

Zazwyczaj nie trzeba umieszczać wszystkich słów kluczowych na każdej ze stron. Jest raczej mało prawdopodobne, że przy przechodzeniu ze strony głównej na inne strony witryny używane słowa kluczowe będą się zmieniać, by dokładniej odpowiadać tematyce konkretnych stron.

GOOGLE ANALYTICS: POZNAJ SWOICH UŻYTKOWNIKÓW

Gdy tylko internauci zaczną odwiedzać naszą witrynę, możemy zacząć analizować to, jak ją znaleźli, czego na niej szukają i w jakich miejscach ją opuszczają.

REJESTRACJA

Korzystanie z usługi Google Analytics wymaga zarejestrowania się i utworzenia konta na stronie www.google.com/analytics. Usługa udostępnia fragment kodu śledzącego, który należy umieścić na każdej stronie witryny.

JAK TO DZIAŁA?

Za każdym razem, gdy ktoś wyświetla jakąś stronę witryny, kod śledzący przesyła dane na serwer Google, gdzie są one zapisywane. Usługa Google Analytics udostępnia internetowy interfejs pozwalający analizować, jak odwiedzający korzystają z witryny.

KOD ŚLEDZĄCY

Kod śledzący jest udostępniany przez usługę Google Analytics dla każdej witryny, którą chcemy analizować. Powinien on zostać umieszczony bezpośrednio przed zamkającym znacznikiem `</head>`. Kod ten w żaden sposób nie wpływa na prezentację strony w internecie.

W czasie powstawania niniejszej książki firma Google aktualizowała usługę Analytics do wersji 5. Jeśli zatem Czytelnik zaloguje się do witryny usługi i zobaczy inny interfejs, powinien poszukać umieszczonego w prawym górnym rogu strony łącza *Nowa wersja*.

JAK DUŻO OSÓB ODWIEDZA NASZĄ WITRYNĘ?

Strona przeglądu udostępnia kluczowe informacje, które najprawdopodobniej będziemy chcieli poznać. W szczególności pokazuje, ile osób odwiedza naszą witrynę.

ODWIEDZINY

To liczba odwiedzin naszej witryny. Jeśli użytkownik będzie nieaktywny przez 30 minut, a następnie ponownie rozpocznie przeglądanie stron witryny, zostanie to policzone jako kolejne odwiedziny.

UNIKALNI UŻYTKOWNICY

To sumaryczna liczba osób, które odwiedziły witrynę w określonym przedziale czasu. Jeśli jakieś osoby powróciły na witrynę więcej niż jeden raz w danym przedziale czasu, to liczba unikalnych użytkowników będzie mniejsza od liczby odwiedzin.

ODSŁONY

To całkowita liczba wyświetlonych stron witryny.

STRONY/ODWIEDZINY

Oznacza średnią liczbę stron, które użytkownicy oglądają podczas wizyty w witrynie.

ŚREDNI CZAS TRWANIA ODWIEDZIN

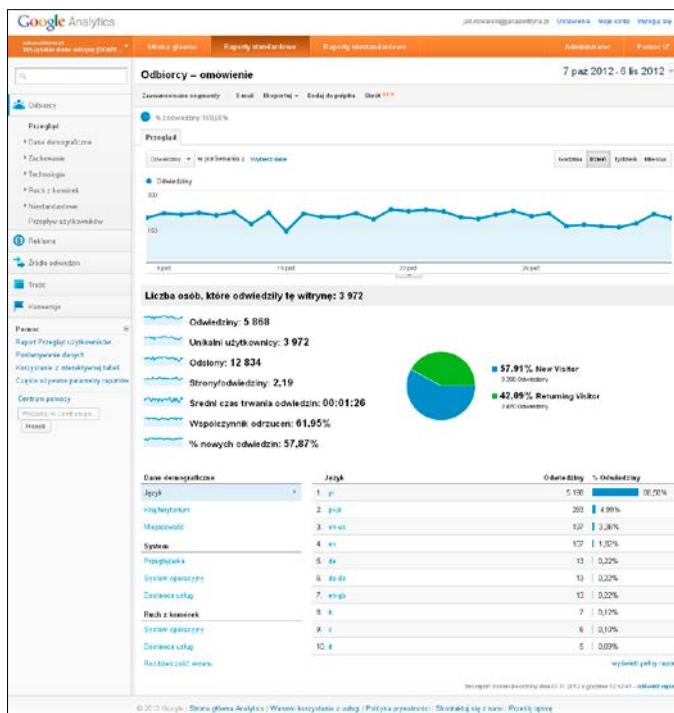
To średni czas, jaki każdy użytkownik spędził, odwiedzając witrynę.

SELEKTOR DATY

Korzystając z selektora daty umieszczonego w prawym górnym rogu strony, można zmienić okres, za jaki jest sporządzany raport. Bezpośrednio po zalogowaniu jest on zazwyczaj ustawiany na jeden miesiąc — można zmienić to ustawienie, by wygenerować raport za dowolny czas.

EKSPORT

Opcja eksportu jest umieszczona na szarym pasku poniżej tytułu strony *Odbiorcy — omówienie*. Umożliwia ona wyeksportowanie statystyk przedstawionych na stronie w kilku różnych formatach, dzięki czemu można te dane przetwarzać w innych aplikacjach, takich jak Excel.



CO OGLĄDAJĄ ODWIEDZAJĄCY?

Opcja **Treść** umieszczona w panelu z lewej strony pozwala uzyskać więcej informacji na temat tego, czego poszukują osoby odwiedzające witrynę.

WSZYSTKIE STRONY

Ta opcja pokazuje, które strony witryny są najczęściej odwiedzane oraz na których odwiedzający spędzają najwięcej czasu.

WSPÓŁCZYNNIK ODRZUCENI

Ta opcja pokazuje, jak wielu użytkowników opuściło witrynę na tej samej stronie, od której rozpoczęto odwiedziny. Wysoki współczynnik odrzuceń sugeruje, że zawartość witryny nie jest tym, czego użytkownicy poszukiwali, albo strona nie zachęcała ich do kontynuowania odwiedzin. Jako odrzucenie jest uwzględniane:

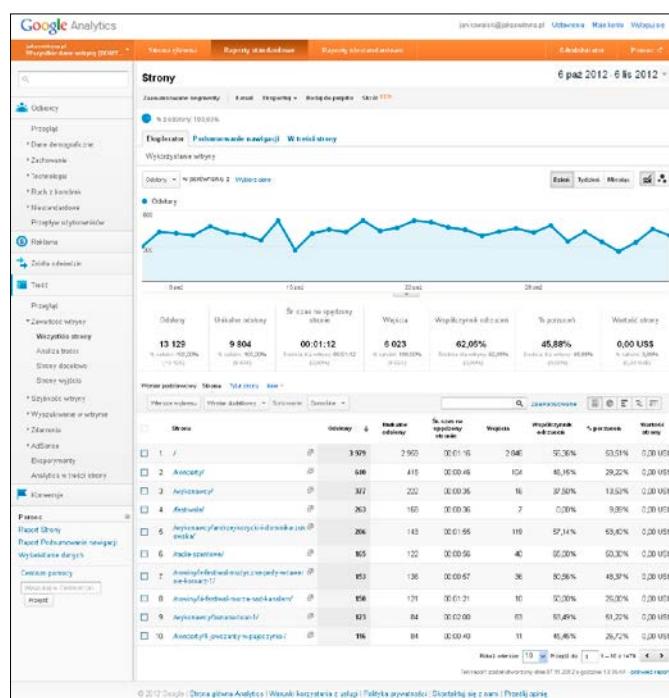
- kliknięcie łącza prowadzącego do innej witryny,
- kliknięcie reklamy,
- wpisanie nowego adresu URL,
- kliknięcie przycisku *Wstecz*,
- zamknięcie przeglądarki.

STRONY DOCELOWE

Strony docelowe to strony, od których użytkownicy zaczynają przeglądanie witryny. Opcja ta jest szczególnie przydatna, gdyż może się okazać, że odwiedziny nie zawsze rozpoczynają się od strony głównej.

STRONY WYJŚCIA

Ta opcja pokazuje, które strony najczęściej są ostatnimi stronami witryny oglanymi przez odwiedzających. Jeśli wiele osób kończy odwiedziny na tej samej stronie, to warto się zastanowić nad jej modyfikacją lub poprawą.



SKĄD PRZYCHODZĄ ODWIEDZAJĄCY?

Opcja **Źródła odwiedzin** umieszczona w panelu po lewej stronie pozwala się dowiedzieć, skąd najczęściej przychodzą osoby odwiedzające witrynę.

CAŁY RUCH

Ta opcja pokazuje wszystkie witryny zawierające łącza prowadzące na naszą stronę oraz liczbę osób, które trafiły na nią dzięki tym łączom. Jeśli z jakiejś witryny trafia do nas wielu użytkowników, to warto się skontaktować z jej właścicielem i wspólnie popracować nad zachowaniem tak wysokiego ruchu. Można także poszukać witryny o podobnej tematyce i poprosić ich właścicieli o umieszczenie na nich łączy do naszej witryny.

BEZPOŚREDNI

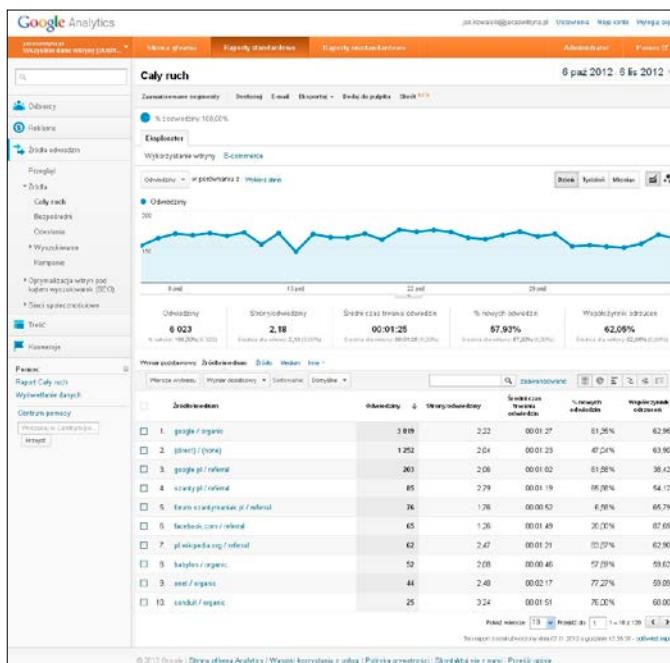
Ta opcja pokazuje, na jakie strony trafiły osoby, które weszły do naszej witryny, korzystając z łącz. Użytkownicy mogą wchodzić do witryny po wpisaniu jej adresu URL w przeglądarce, skorzystaniu z adresu zapisanego w *Ulubionych*, kliknięciu łącza umieszczonego w wiadomości e-mail, w dokumencie PDF lub dokumencie programu Word.

WYSZUKIWANIE

Ta opcja pokazuje terminy użyte przez osoby, które trafiły na naszą stronę za pośrednictwem wyszukiwarki. Dzięki tej opcji możemy się dowiedzieć, jak użytkownicy opisują to, czego poszukują (co często różni się od tego, jak właściciele opisują swoje witryny). Opcja ta może się przydać podczas poprawiania zawartości i słów kluczowych używanych w SEO.

ZAAWANSOWANE MOŻLIWOŚCI

Przedstawione tu możliwości to jedynie drobny ułamek wszystkich informacji na temat witryny, jakie możemy uzyskać, korzystając z Google Analytics. Znacznie więcej informacji na temat zaawansowanych możliwości usługi można znaleźć w jej systemie pomocy. Jeśli prowadzimy sklep internetowy, skorzystajmy z mechanizmów śledzenia e-commerce, udostępniających informacje o kupowanych produktach, średniej liczbie produktów w koszyku itd. Możemy także określić cele definiujące ścieżki, którymi chcemy, by podążali użytkownicy witryny, i sprawdzać, jak daleko po tych ścieżkach docierają (co jest szczególnie przydatne, gdy pobieramy od użytkowników jakieś informacje).



NAZWY DOMEN I SERWER WWW

Do uruchomienia witryny w internecie potrzebne będą nazwa domeny i serwer WWW.

NAZWY DOMEN

Nazwa domeny jest naszym adresem internetowym (jak google.com lub helion.pl). Istnieje wiele witryn pozwalających na rejestrowanie nazw domen. Zazwyczaj posiadanie nazwy domeny wiąże się z koniecznością uiszczenia corocznych opłat. Witryny te zazwyczaj pozwalają sprawdzić, czy preferowana nazwa domeny jest dostępna, a ponieważ zostały już zarejestrowane miliony nazw domen, znalezienie odpowiedniej nazwy może potrwać.

Wiele witryn udostępniających możliwość rejestracji nazw domen świadczy także usługi prowadzenia serwerów WWW.

SERWER WWW

Aby inni użytkownicy internetu mogli oglądać naszą witrynę, musimy ją skopiować na serwer WWW. Serwery WWW to specjalne komputery, które są cały czas podłączone do internetu. Zostały one specjalnie skonfigurowane pod kątem udostępniania stron WWW, kiedy tylko zostaną o to poproszone.

Przeważająca większość witryn, z wyjątkiem tych największych, jest umieszczana na serwerach zarządzanych przez specjalne firmy. Takie rozwiązanie jest zazwyczaj tańsze i bardziej niezawodne niż uruchamianie własnego serwera.

Dostępnych jest wiele typów usług prowadzenia witryn WWW. Przedstawimy kilka kluczowych zagadnień, które pomogą wybrać ofertę odpowiednią dla potrzeb naszej witryny.

PRZESTRZEŃ DYSKOWA

Termin ten odnosi się do całkowitej wielkości wszystkich plików tworzących witrynę (wszystkich plików HTML, CSS, obrazów oraz skryptów).

PRZEPUSTOWOŚĆ

To wielkość danych, które firma zarządzająca serwerem przesyła do osób oglądających naszą witrynę. Jeśli wyobrażymy sobie, że 10 osób chce przejrzeć wszystkie strony witryny, to przepustowość byłaby dziesięciokrotnie większa od zajmowanej przestrzeni dyskowej.

KOPIE ZAPASOWE

Warto sprawdzić, czy firma zarządzająca serwerem WWW tworzy kopie zapasowe witryn (a jeśli tak, to jak często). Niektóre tworzą kopię tylko po to, by móc ją odtworzyć w razie awarii serwera. Inne zapewniają możliwość dostępu do tworzonych kopii (co może się przydać, jeśli omyłkowo uszkodzimy coś w witrynie podczas jej aktualizacji).

UDOSTĘPNIANE USŁUGI

KONTA POCZTOWE

Większość firm prowadzących serwery WWW w ramach pakietu wykupywanych usług udostępnia także możliwość korzystania z serwera poczty elektronicznej. Warto sprawdzić wielkość oferowanych skrynek pocztowych oraz ich liczbę.

SERWEROWE JĘZYKI PROGRAMOWANIA I BAZY DANYCH

Jeśli korzystamy z systemu zarządzania treścią, to najprawdopodobniej będzie on napisany w którymś języku programowania i będzie mu potrzebna jakaś baza danych (np. język PHP jest najczęściej używany wraz z bazą danych MySQL, a ASP.NET — z bazą SQL Server). Koniecznie trzeba sprawdzić, czy firma zarządzająca serwerem udostępnia technologie niezbędne do działania naszego oprogramowania.

Warto także poszukać opinii na temat firmy, z której usług planujemy skorzystać. Niestety, bardzo często jest tak, że o wartości takiej firmy można się przekonać dopiero, gdy coś pójdzie nie tak — wtedy się okazuje, w jakim stopniu jest ona w stanie nam pomóc (dlatego należy się spodziewać, że znajdziemy kilka niepochlebnych opinii o każdej firmie).

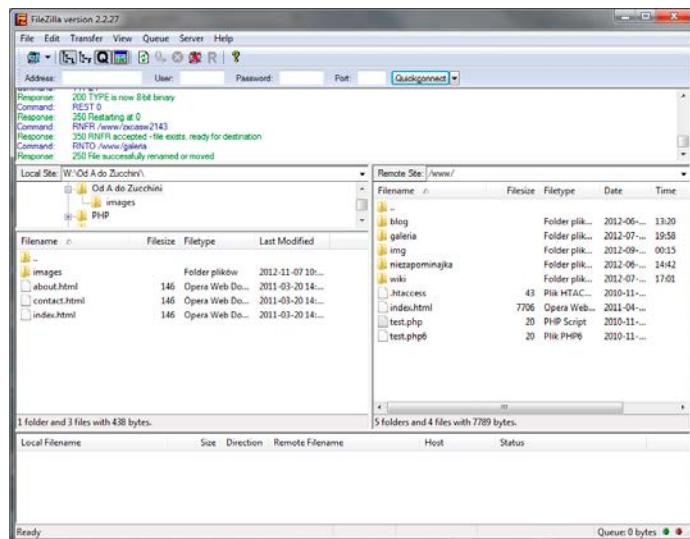
Istnieją usługi internetowe, które pozwalają powiązać nazwę domeny z ich serwerami. Platformy blogowe, takie jak WordPress.com, Tumblr czy Posterous, oraz platformy do handlu elektronicznego, takie jak Big Cartel oraz Shopify, udostępniają serwery, na których można umieszczać własne witryny. Jeśli skorzystamy z takiej platformy, nie będziemy musieli wykupywać serwera WWW, choć trzeba będzie poszukać usługi zapewniającej dostęp do kont poczty elektronicznej. Jeśli stoiemy przed takim problemem, to niektóre firmy prowadzące serwery WWW udostępniają pakiety zapewniające obsługę samych kont pocztowych.

FTP ORAZ INNE PROGRAMY NARZĘDZIOWE

Aby skopiować kody oraz obrazy z naszego komputera na serwer WWW, musimy skorzystać z protokołu FTP (ang. *File Transfer Protocol*, protokół przesyłu plików).

Zgodnie z tym, co sugeruje nazwa, protokół FTP pozwala przesyłać pliki z komputera na serwer firmy, która obsługuje witrynę WWW.

Dostępnych jest wiele programów korzystających z protokołu FTP — dysponują one prostym interfejsem użytkownika, przedstawiającym pliki na naszym komputerze oraz na serwerze WWW. Pozwalają kopiować pliki pomiędzy tymi dwoma komputerami przy użyciu techniki „przeciągnij i upuść”.



Wiele witryn udostępnia usługi powszechnie tworzone przez programistów zajmujących się pisaniem stron WWW (dzięki czemu nie trzeba ich tworzyć samemu).

Niektóre firmy zarządzające serwerami WWW udostępniają narzędzia pozwalające na przesyłanie plików na ich serwery przy użyciu przeglądarek WWW, jednak znacznie częściej korzysta się z programów FTP, gdyż umożliwiają one szybszy transfer plików.

Przy zamówieniu usługi uzyskamy wszystkie informacje, jakie trzeba podać w programie FTP, by nawiązać połączenie z serwerem. Zazwyczaj będzie to adres serwera (np. `ftp://mojadomena.pl`), nazwa użytkownika oraz hasło. Bardzo duże znaczenie ma zachowanie poufności tych informacji, aby nikt niepowołany nie uzyskał dostępu do naszej witryny.

Poniżej przedstawiono listę popularnych programów FTP:

FileZilla

filezilla-project.org
Windows, Mac, Linux

FireFtp

fireftp.mozdev.org
Windows, Mac, Linux

CuteFTP

cuteftp.com
Windows, Mac

SmartFTP

smartftp.com
Windows

Transmit

panic.com/transmit
Mac

A oto lista innych przydatnych narzędzi:

BLOGI

wordpress.com
tumblr.com
posterous.com

HANDEL ELEKTRONICZNY

shopify.com
bigcartel.com
go.magento.com

BIULETYNY ROZSYŁANE POCZĄTKĄ ELEKTRONICZNĄ

campaignmonitor.com
mailchimp.com

PRZYCISKI UDOSTĘPNIANIA W PORTALACH SPOŁECZNOŚCIOWYCH

addthis.com
addtoany.com

PODSUMOWANIE

INFORMACJE PRAKTYCZNE

- ▶ Optymalizacja pod kątem wyszukiwarek pomaga osobom korzystającym z internetowych wyszukiwarek odnajdywać daną witrynę.
- ▶ Narzędzia do analiz, takie jak Google Analytics, pozwalają analizować, w jaki sposób odwiedzający przeglądają witrynę, jak ją odnajdują i co robią, kiedy na nią trafią.
- ▶ Aby udostępnić witrynę w internecie, trzeba wykupić nazwę domeny oraz miejsce na serwerze WWW.
- ▶ Programy FTP pozwalają przenosić pliki z lokalnego komputera na serwer WWW.
- ▶ Wiele firm udostępnia platformy blogowe, biuletyny rozsyłane pocztą elektroniczną oraz inne przydatne narzędzia wykorzystywane w witrynach WWW (dzięki czemu można zaoszczędzić czas, gdyż nie trzeba ich pisać samodzielnie).

SKOROWIDZ

- ▶ A – Z.
- ▶ Rozwiązywanie problemów.
- ▶ Elementy HTML.
- ▶ Atrybuty HTML.
- ▶ Właściwości CSS.

A

Adobe Fireworks,
 Patrz: Fireworks
Adobe Illustrator,
 Patrz: Illustrator
Adobe Photoshop,
 Patrz: Photoshop
adres
 IP, 9
 poczty elektronicznej, 84
 URL, 150, 479
 bezwzględny, 78
 względny, 79, 82, 91
AJAX, 216
akapit, 22, 43
alfa, 253, 255
Android, 212
animacja, 7, 200, 206,
 223, 281
arkusz stylów
 kaskadowy, Patrz: CSS
 zewnętrzny, 235, 240
atribut, 37
 action, 150
 align, 102, 410
 alt, 98, 120, 123, 160,
 478, 479
 autoplay, 213, 218
 bgcolor, 137
 border, 137
 cellpadding, 136
 cellspacing, 136
 checked, 154, 155
 cite, 51
 class, 176, 183, 186, 197,
 288, 430, 443
 codecs, 214
 cols, 153
 colspan, 132, 138, 141
 content, 190
 controls, 213, 218
 format, 162
 frameborder, 189
 globalny, 176, 182
 height, 99, 160, 188, 213
 href, 76, 78, 86, 87, 91, 234
 mailto:, 84
 target, 85

http-equiv, 190
id, 86, 87, 91, 150, 162,
 176, 182, 186, 197,
 206, 237, 430
lang, 24
loop, 213, 218
maxlength, 151, 152
method, 150, 158
multiple, 157
name, 151, 152, 154, 155,
 156, 159, 190
nazwa, 24
placeholder, 167
poster, 213
preload, 213, 218
rel, 234
required, 164
rows, 153
rowspan, 133, 141
scrolling, 189
seamless, 189
selected, 156
size, 151, 152, 157
src, 98, 123, 160, 188, 213,
 214, 218, 219
style, 235
title, 52
title, 98, 120
type, 64, 65, 214, 219
 checkbox, 155
 date, 165
 email, 166
 file, 158
 hidden, 161
 image, 160
 password, 152
 radio, 154
 search, 166
 submit, 159
 text, 151
 text/css, 234
 url, 166
value, 154, 155, 156, 159
wartość, 24
width, 99, 136, 160,
 188, 213
audio, 200
autor, 54, 190

B

bezpieczeństwo, 152
browser quirk,
 Patrz: przeglądarka
dziwactwo

C

CDN, 277
Chrome, 6, 53, 212, 213,
 319, 346
cień, 287, 319, 342
CMS, 7, 32, 48, 487
 szablon, 32, 33
Content Delivery Network,
 Patrz: CDN
CSS, 51, 64, 65, 226, 230,
 238, 243, 327
błąd, Patrz: przeglądarka
 dziwactwo
komentarz, 248
platforma, Patrz: platforma
reguła, Patrz: reguła
schemat pozycjonowania,
 Patrz: schemat
 pozycjonowania
wersja, 241
cytat, 51
czcionka, 266, 267, 268, 269,
 270, 271, 276, 297,
 341, 394
krój, 270, 272
wielkość, 273, 274, 275
czytnik ekranowy, 6, 49, 77,
 134, 431

D

definicja, 53, 62, 66
deklaracja, 230, 243
DOCTYPE, 180, 181, 197,
 315, 323
Digital Rights Management,
 Patrz: DRM
DNS, 8, 9
dokument, 14, 16
domena, 486
dostawca usług internetowych,
 Patrz: ISP

Dreamweaver, 48

DRM, 212

dźwięk, 216

E

edytor

HTML, 48

wizualny, 48

ekran, 6

dotykowy, 290

rozdzielcość, 377

rozmiar, 376

element, *Patrz też: znacznik*

<!-- -->, 181, 197

<a>, 76, 77, 78, 84,
185, 440, 479

<abbr>, 52

<acronym>, 52

<address>, 54

<article>, 431, 432,
434, 436

<aside>, 431, 435, 439

<audio>, 203, 218,
221, 223

, 44, 101, 185,
228, 360

<blockquote>, 49, 51, 369

, 47

<button>, 161

<center>, 178

<cite>, 53

<content>, 439

<dd>, 66

, 55

<dfn>, 53

<div>, 186, 197, 206, 228,
323, 361, 403, 430,
436, 439, 449

<dl>, 66

<dt>, 66

, 49, 50, 101,
185, 228

<fieldset>, 163, 169,
343, 349

<figcaption>, 119, 421,
438, 443

<figure>, 119, 120, 421,
438, 443

, 178

<footer>, 431, 432,
439, 443

<form>, 150, 158, 168, 173

<h1>, 42, 101, 184, 228,
360, 479

<head>, 190, 479

<header>, 431, 432,
439, 443

<hgroup>, 437, 443

<hr />, 47

<i>, 44, 228, 360

<iframe>, 176

<iframe>, 188, 197

, 98, 101, 123, 178,
185, 228, 360, 410

<input>, 151, 152, 154,
161, 165, 166, 167

<ins>, 55

<label>, 162, 169, 344

<legend>, 163

, 64, 65, 184, 332,
333, 360

<link>, 234, 240, 392,
394, 395

<meta>, 176

<meta>, 190, 197

<nav>, 431, 433

, 64, 332

<option>, 156

<p>, 22, 43, 101, 184,
228, 323, 360

<q>, 51

<s>, 55

<script>, 206, 207

<section>, 432, 436, 443

<select>, 156, 157

<source>, 214, 219

, 187, 197, 228, 344

<strike>, 178

, 50

<style>, 235

<sub>, 45

<sup>, 45

<table>, 130, 136, 141

<tbody>, 134

<td>, 130, 132, 133,
141, 349

<textarea>, 153

<tfoot>, 134, 135

<th>, 131, 132, 133, 138,
337, 349

<thead>, 134, 135

<title>, 479

<tr>, 130, 141

<u>, 55

, 65, 184, 323, 332,
333, 360

<video>, 203, 208, 209,
212, 213, 215, 221,
223

blokowy, 101, 184, 228,
316, 327, 360, 362,
440

body, 26, 228, 361

cień, 319

dziecka, 237

grupowanie, 439

head, 26

HTML, 19, 20, 21, 23,
24, 58

konstrukcyjny, 360

margines, 306, 307, 360

nachodzący na inny,
363, 368

nawigacyjny, 186, 316, 323,
361, 383, 385, 430,
431, 432, 433, 462,
470

obramowanie, 306, 307,
308, 309, 311, 318,
327, 341, 360

kolor, 310

styl, 309

szerokość, 308, 318

zaokrąglenia, 320,
321, 327

pływający, 363, 369, 370,
372, 403

poleżenie, 362, 363
bez względne, 362,
367, 368

ustalone, 363, 368

względne, 362, 368

potomny, 237

przesunięcie, 363

pusty, 47

title, 26

ukrywanie, 316, 317

element
 wewnętrzierszowy, 101,
 185, 228, 316, 327,
 360, 417, 441
właściwość,
 `Patrz: właściwość`
wymiary, 302, 303, 304
wypełnienie, 306, 307,
 312, 360
wyśrodkowanie, 314
zawierający, `Patrz: rodzic`
em, 273, 275, 282, 302
etykieta, 340

F

Firefox, 6, 157, 212, 319
Fireworks, 107, 303, 346
Flash, 200, 202, 203, 206,
 216, 217, 221, 223
Flash Of Unstyled Content,
 `Patrz: FOUC`
Flash Of Unstyled Text,
 `Patrz: FOUT`
Flash Player, 202
Flash Video Converter, 210
FontSquirrel, 277
format
 AVI, 208
 CML, 179
 Flash Video,
 `Patrz: format FLV`
 FLV, 201, 210, 215
 GIF, 106, 110, 120
 animowany, 116
 przezroczysty, 117
 H264, 208, 210, 212,
 215, 221
 JPEG, 106, 108, 109, 120
 MathML, 179
 MP3, 216, 219
 MPEG, 208
 Ogg Theora, 208
 Ogg Vorbis, 219
 PNG, 106, 110, 111
 przezroczysty, 117, 397
 QuickTime, 208
 SVG, 115, 179
 SWF, 202
 WebM, 208, 212, 215, 221
 Windows Media, 208

formularz, 144, 146, 148,
 150, 169, 173, 229,
 330, 340, 342, 343,
 349, 355
metoda
 get, 150
 post, 150, 158
pole, `Patrz: pole`
FOUC, 277
FOUT, 277

G

GIMP, 250
Google Analytics, 482, 483,
 484, 485
gradient, 418

H

hasło, 152
HSB, 254
HSL, 254, 255, 261, 310
HSLA, 254, 255, 261
HTML, 4, 5, 7, 19, 20, 27, 37
 edytor, `Patrz: edytor HTML`
 historia, 178
 wersja, 176, 178
HTML4, 52, 53, 178
HTML5, 7, 52, 178, 428, 441,
 443, 449
 podkładka, HTML5
 shim, `Patrz: HTML5`
 podkładka
HyperText Markup Language, 27,
 `Patrz też: HTML`

I

Illustrator, 115, 462
InDesign, 462
inline frame, `Patrz:`
 ramka płynąca
interlinia, 282
Internet Explorer, 6, 51, 212,
 303, 312, 313, 314,
 315, 323, 396, 440,
 443
Internet Service Provider,
 `Patrz: ISP`
ISP, 9

J

JavaScript, 117, 164, 182,
 340, 396, 441,
 443, 449
biblioteka, 203, 205
jednostka em, `Patrz: em`
język
 XML, `Patrz: XML`
 JavaScript, `Patrz: JavaScript`

K

karta, 26
kerning, 283
klip wideo, `Patrz: wideo`
kod
 HTML, 19, 20, 34
 źródłowy, 48
kodowanie, 208, 214
kolor, 246, 261, 341
 alfa, `Patrz: alfa`
 HSB, `Patrz: HSB`
 HSL, `Patrz: HSL`
 HSLA, `Patrz: HSLA`
jasność, 251, 254, 255
jednolity, 111
kod szestnastkowy, 248,
 250, 251, 261, 310
kontrast, 252, 261, 419
nasycenie, 251, 254, 255
nazwa, 248, 251, 261
odcień, 251, 254, 255
RGB, `Patrz: RGB`
komentarz, 176, 181, 186, 197
 CSS, `Patrz: CSS komentarz`
warunkowy, 441

L

lista, 62, 69, 70, 229,
 330, 349
definicji, 62, 66, 70
element, 64
nieuporządkowana, 62, 65,
 66, 69, 70, 323, 332
numerowana, `Patrz:`
 lista uporządkowana
 rozwijana, 147, 156,
 162, 340

uporządkowana, 62, 64, 69, 70, 332
wielokrotnego wyboru, 157
wypunktowana, *Patrz:*
 lista nieuporządkowana
zagnieździona, 67, 70

Ł

łącze, 74, 89, 91, 416, 417, 440, 478
tworzenie, 76
w obrębie witryny, 79

M

metoda
 get, 150
 post, 150
mikroformat hCard, 54
Mozilla Firefox, *Patrz:* Firefox

N

nagłówek, 16, 20, 42, 432, 436, 479
grupa, 437
poziom, 42
nieprzezroczystość, 253

O

obraz, 94, 96, 106, 406, 425, 438, 467
animowany, 116
format, *Patrz:* format
pobieranie, 118
podpis, 119
rastrowy, 115
rozdzielcość, 106, 114
tła, 412, 419
umiejscowienie w kodzie, 100
wektorowy, 115
wymiary, 99, 106, 112, 114, 118, 408, 410
obszar przed zagięciem, 378
odstępu pomijanie, 46
odtwarzacz, 208
 audio, 216
 Flash, 210, 221
Opera, 6, 212

P

Paint.net, 107
PaintShop pro, 107
pasek przewijania, 189, 305, 362
Photoshop, 107, 116, 250, 462
Photoshop Elements, 107
Pixelmator, 107
platforma, 390
 960.GS, 390, 391, 392, 397
 blogowa, 7, 32, 486, 489
 Grid System, *Patrz:*
 platforma 960.GS
 handlu elektronicznego, 7, 32, 487, 489
plik

 audio, 216, 218
 index.html, 81
 przesyłanie, 147, 158
podmienianie, 416
pole
 adresu
 poczyt elektronicznej, 166
 URL, 166
 daty, 165
 etykieta, 162
 grupowanie, 343
 hasta, 147
 przesyłania plików, 147, 158, 162

przycisk, *Patrz:* przycisk
tekstowe, 341
 jednowierszowe, 146, 147, 162, 340
 wielowierszowe, 147, 153, 162, 340
weryfikacja, 164
wyboru, 147, 154, 162
wyrównywanie, 343, 344, 345, 355
wyszukiwania, 167
pomijanie odstępów, 46
Posterous, 487
protokół
 FTP, 488, 489
 SSL, 152

przeglądarka, 6, 241, 340, 440
 Chrome, *Patrz:* Chrome
 dziwactwo, 241, 323

Firefox, *Patrz:* Firefox

Internet Explorer,
 Patrz: Internet Explorer
Mozilla Firefox, *Patrz:* Firefox
okno główne, 27
Opera, *Patrz:* Opera
Safari, *Patrz:* Safari
przezroczystość, 117, 397,
 Patrz też:
 nieprzezroczystość

przycisk, 416
 graficzny, 147, 160
 opcji, 147, 154, 162
przesyłający, 147, 159, 340, 342
pseudoelement, 288
 :first-letter, 288
 :first-line, 288
pseudoklasa, 288, 289
 :active, 289, 290, 417
 :first-child, 349
 :focus, 290, 341
 :hover, 288, 289, 290, 336, 341, 342, 417
 :last-child, 349
 :link, 289
 :visited, 288, 289

R

ramka płynąca, 188
RealAudio, 204
reguła, 226, 230, 238, 243
 @font-face, 270, 276
 @import, 394
RGB, 248, 250, 251, 261
RGBA, 253, 261
rodzic, 361, 369, 371, 372

S

Safari, 6, 53, 157, 158, 167, 212, 319
Scalable Vector Graphics,
 Patrz: format SVG
schemat pozycjonowania, 362, 363, 364, 365
Search Engine Optimization,
 Patrz: SEO

- Secure Socket Layer,
Patrz: protokół SSL
- sekcja, 436
- selektor, 230, 236, 243
- [\$=], 291
- [*=], 291
- [[], 291
- [^=], 291
- [~=], 291
- [=], 291
- elementu
- dziecka, 237
 - potomnego, 237
 - sąsiadującego, 237
- identyfikatora, 237
- klasy, 237
- ogólny elementu, 237
- typu, 237
- uniwersalny, 237
- SEO, 476, 478, 480, 485
- serwer
- poczty elektronicznej, 487
 - systemu nazw domen,
Patrz: DNS
 - WWW, 6, 9, 486
 - kopia zapasowa, 486
 - przepustowość, 486
 - przestrzeń dyskowa, 486
- siatka, 386, 403
- 960 pikseli, 386, 387, 388, 389, 403
- sieć udostępniania treści,
Patrz: CDN
- skrypt
- JavaScript, 7, 202
 - SWFObject, 206, 210, 211, 217, 221
- słowo kluczowe, 478, 479, 480
- sortowanie kart, 460
- sprajt, 416, 425
- stopka, 432
- strona www
- adres, 479
 - folder, Patrz:
strona www katalog
 - hierarchia wizualna, 464, 466
- katalog, 80
- główny, 81
 - nadrzędny, 81, 83
 - podrzędny, 81, 83
- mapa, 460
- optymalizacja pod kątem wyszukiwania,
Patrz: SEO
- projektowanie, 358, 372, 374, 380, 386, 390, 397, 428, 430, 449, 452, 454, 460, 464, 468, 473
- rozmiar, 378
- siatka, Patrz: siatka
- strona
- docelowa, 484
 - wyjścia, 484
- struktura, 80, 358, 372, 374
- szkielet, 462, 473
- śledzenie ruchu, 485
- tytuł, 479
- układ, 362, 363, 364, 365, 366, 367, 394, 395, 430, 449, 452, 464
- elastyczny, 381, 384
- grupowanie informacji, 468
- o stałej szerokości, 380, 382
- wielokolumnowy, 374
- użytkownik, Patrz: użytkownik w nowym oknie
- przeglądarki, 85
- zawartość, 464
- symbol specjalny,
Patrz: znak specjalny
- system zarządzania treścią,
Patrz: CMS
- szablon, 390
- T**
- tabela, 126, 129, 138, 141, 330, 336, 349, 355
- długa, 134, 135, 141
- kolor tła, 137, 337
- komórka, 129, 130, 134, 337, 355
- łączenie, 132, 133
- posta, 338
- nagłówek, 131, 134, 135, 337
- stopka, 134
- szerokość, 136
- obramowania, 137
- wiersz, 130
- tapeta, 421
- tekst, 229, 264
- alternatywny, 98, 479
 - czcionka, Patrz: czcionka
 - dodany, 55
- formatowanie, 264
- interlinia, 282
- kursywa, 44, 48, 49, 50, 53, 54, 278
- linia pisma
- bazowa, 267
 - górna, 267
 - średnia, 267
- łączna, 76
- podkreślony, 55
- pogrubiony, 44, 48, 50, 278
- przekreślony, 55
- usunięty, 55
- uzupełniający, 438
- wcięty, 49, 51, 67, 286
- wydłużenie
- dolne, 267, 282
 - górne, 267, 282
- wysokość x, 267
- wyśrodkowanie, 314
- zdezaktualizowany, 55
- TextEdit, 30
- Tumblr, 487
- U**
- Uniform Resource Locator, 78, Patrz też: adres URL
- użytkownik, 456, 457, 458, 483
- V**
- Vimeo, 204, 208, 209
- W**
- Web Developer Toolbar, 347
- wideo, 200, 208, 223, 438
- wireframe, Patrz:
strona www szkielet

witryna, *Patrz: strona www właściwość*, 231, 243, 320
!important, 238
background, 415
background-attachment, 413
background-color, 239, 249, 336, 341, 343
background-image, 94, 341, 412, 418
background-position, 414, 417
background-repeat, 413
border, 311, 336, 341, 343
border-collapse, 339
border-color, 310
border-image, 318
border-radius, 320, 321, 341, 343
border-separate, 339
border-spacing, 339, 349
border-style, 309
border-width, 308
box-shadow, 319
clear, 371, 373
color, 248, 341, 342, 343
cursor, 346
display, 316, 323, 327
dziedziczenie, 239, 312, 313, 349
empty-cells, 338, 339
float, 362, 369, 370, 373, 397, 403, 410
font-family, 239, 270, 272, 276
font-size, 273, 282, 341
font-style, 279
font-weight, 278
height, 302, 408
letter-spacing, 283, 336
line-height, 282, 349
list-style, 335
list-style-image, 333
list-style-position, 334
list-style-type, 64, 65, 332, 355
margin, 103, 307, 313, 323, 333, 373
margin-left, 314

margin-right, 314
max-height, 304
max-width, 303
min-height, 304
min-width, 303
-moz-border-image, 318
-moz-border-radius, 320
-moz-box-shadow, 319
opacity, 253
overflow, 304, 305, 373
padding, 103, 307, 312, 323, 336, 343
position
 absolute, 366
 fixed, 367
 relative, 365
 static, 364
rgba, 253
text-align, 284, 314, 336, 337
text-decoration, 281
text-indent, 286
text-shadow, 287, 319, 342
text-transform, 280, 336
vertical-align, 285
visibility, 317, 327
-webkit-border-image, 318
-webkit-border-radius, 320
-webkit-box-shadow, 319
width, 302, 323, 336, 343, 369, 373, 373, 408
word-spacing, 283
z-index, 363, 368
WordPress.com, 487
współczynnik odrzuceń, 484
wtyczka, 202, 208
wyszukiwarka, 49, 98, 131, 190, 286, 431, 478, 479, 481, 485

X

XML, 178

Y

YouTube, 204, 208, 209

Z

zdjęcie, 96
znacznik, 37, *Patrz też: element otwierający*, 19, 20, 22
semantyczny, 40, 49
strukturalny, 40
zamykający, 19, 20, 23
znak
 ', *Patrz: znak prawy apostrof*, 192
 ', *Patrz: znak lewy apostrof*, 192
 '', 192
 ¢, *Patrz: znak centa*
 £, *Patrz: znak funta szterlinga*
 ¥, *Patrz: znak jena*
 €, *Patrz: znak euro*
 ×, *Patrz: znak mnożenia*
 ÷, *Patrz: znak dzielenia*
 ©, *Patrz: znak praw autorskich*
 ®, *Patrz: znak handlowy*
 zarejestrowany
apostrof
 lewy, 192
 prawy, 192
centa, 192
cudzysłowu, 192
dzielenia, 192
euro, 192
funta szterlinga, 192
handlowy, 192
 zarejestrowany, 192
jena, 192
mniejszości, 192
mnożenia, 192
praw autorskich, 192
specjalny, 192
™, *Patrz: znak handlowy ukośnika*, 19, 20
większości, 192

Ż

żądanie, 6

ROZWIĄZYWANIE PROBLEMÓW

Poniżej przedstawiono kilkanaście problemów, przed którymi często stają początkujący twórcy stron WWW, wraz z numerami stron, na których można znaleźć informacje o sposobach ich rozwiązania.

NA POCZĄTEK

Przeglądarka wyświetla kod HTML (a nie stronę).

Należy sprawdzić, czy plik ma rozszerzenie `.html`, a nie `.txt`.

[Strony: 29, 31.](#)

Pogrubienie, wyróżnienie, nagłówek lub łączka rozciąga się na większy fragment tekstu, niż powinno.

Należy sprawdzić, czy w kodzie znajduje się odpowiedni znacznik zamykający, np. ``.

[Strony: 20, 21.](#)

Strona docelowa łączka nie została odnaleziona.

Trzeba sprawdzić względny adres URL. [Strony: 82, 83.](#)

OBRAZY

Obrazy nie są wyświetlane.

Trzeba sprawdzić względny adres URL. [Strony: 82, 83.](#)

Obrazy są zamazane.

Trzeba sprawdzić, czy obrazy zostały zapisane we właściwym formacie i czy mają odpowiednie wymiary. [Strony: 108 – 112.](#)

Obrazy tła nie są widoczne.

Działają one wyłącznie w najnowszych wersjach przeglądarek. (Warto także sprawdzić względne adresy URL).

[Strony: 318, 20, 21.](#)

Powiększone obrazy GIF mają wyraźnie widoczne ziarno i postrzępione krawędzie.

W programie graficznym trzeba sprawdzić przestrzeń kolorów. Powinna nią być przestrzeń RGB (a nie kolory indeksowane, ang. *indexed color*).

TEKST

Fragmenty tekstu migają, zanim zostaną wyświetcone odpowiednią czcionką.

To tzw. miganie treści bez określonych stylów. [Strona 277.](#)

Podczas kopiowania tekstu z edytora tekstu w kodzie strony pojawia się bardzo dużo dodatkowych znaczników.

W pierwszej kolejności trzeba skopiować tekst do prostego edytora (aby usunąć formatowanie), a dopiero potem do edytora HTML. [Strona 48.](#)

Na niektórych ekranach tekst wydaje się mniejszy lub większy.

Zazwyczaj wynika to z różnic rozdzielczości. Warto także sprawdzić, czy w znaczniku `<body>` została określona wielkość czcionki.

[Strony: 376, 275.](#)

Kiedy używam jednostek em, wielkość czcionki w IE jest niewłaściwa.

Patrz poprzednie wyjaśnienia. [Strona 275.](#)

Zastosowana czcionka nie jest wyświetlana w niektórych przeglądarkach.

Podana czcionka musi być zainstalowana na komputerze użytkownika. [Strony: 268 – 271.](#)

W razie używania reguły @font-family należy podać czcionki w kilku różnych formatach.

[Strona 276-277.](#)

Na komputerach PC używane czcionki nie są gładkie.

W przypadku niektórych czcionek antialiasing nie daje dobrych efektów. Warto wypróbować inny krój pisma lub użyć czcionki pogrubionej. [Strona 271.](#)

OGÓLNE PROBLEMY CSS

Określony styl nie jest używany.

Trzeba pamiętać, że w selektorach CSS uwzględniana jest wielkość liter. Warto sprawdzić poprawność selektorów. [Strona 237.](#)

Web Developer Toolbar pozwoli określić właściwy selektor.

[Strona 347.](#)

Jeśli selektory są prawidłowe, to czy w dalszej części arkusza CSS znajdują się inne selektory odnoszące się do tego samego elementu? [Strony: 238, 239.](#)

Implementacja CSS w przeglądarkach jest różna. Istnieje sporo różnic i dziwactw, które sprawiają, że przeglądarki wyświetlają strony w nieco odmienny sposób. Warto poszukać informacji o problemie i ustalić, czy jest on znany jako dziwactwo lub błąd przeglądarki. [Strona 241.](#)

HTML5

Przeglądarka nie używa stylów w elementach HTML5.

Jeśli ta przeglądarką jest Internet Explorer, to być może trzeba będzie skorzystać ze specjalnego skryptu JavaScript. [Strona 441](#).

Elementy blokowe są wyświetlane jako elementy wewnętrzwerszowe.

Trzeba skorzystać z właściwości `display: block`, by poinstruować przeglądarkę, które elementy HTML są elementami blokowymi. [Strona 441](#).

UKŁAD

Na niektórych ekranach projekt wygląda na mniejszy lub większy.

Na wielkość elementów wyświetlanych na stronie ma wpływ rozdzielcość ekranu. [Strony: 376, 377](#).

Marginsy nad oraz pod elementem nie są wyświetlane.

Marginsy pionowe są scalane. [Strona 307](#).

Treść nie mieści się w elemencie zawierającym lub oknie przeglądarki.

Z tym problemem można sobie poradzić, używając właściwości `overflow`. [Strony: 305, 315](#).

Pudełka elementów w przeglądarce Internet Explorer mają inne wymiary niż w innych przeglądarkach.

Niektóre wersje IE używają innego modelu pudełkowego niż wszystkie pozostałe przeglądarki. Aby działały tak samo, należy użyć deklaracji DOCTYPE. [Strona 315](#).

Pudełka nie są wyśrodkowywane po przypisaniu lewemu i prawemu marginesowi wartości `auto`.

Być może trzeba będzie użyć właściwości `text-align` w elemencie nadrzędnym.

[Strona 314](#).

Elementy znajdują się na siebie.

Kiedy elementy zostaną usunięte z normalnego rozkładu strony, mogą na siebie nadodzić. Właściwość `z-index` pozwala określić, który z nich ma się znaleźć nad pozostałymi.

[Strona 368](#).

Dlaczego użycie właściwości `vertical-align` nie powoduje wyśrodkowania w pionie elementów blokowych?

Ta właściwość nie służy do tego celu. Służy ona do określania wyrównania w pionie elementów wewnętrzwerszowych.

W internecie można znaleźć kilka różnych sposobów wyśrodkowywania elementów blokowych w pionie (zależnie od ich kontekstu).

W elemencie nie jest wyświetlany obraz tła.

Czy została określona wysokość i szerokość tego elementu?

[Strona 302](#).

Czy w elemencie zawierającym właściwości `overflow` przypisano wartość `auto`? [Strony: 372, 373](#).

Obraz tła elementu nie jest widoczny po wydrukowaniu strony.

Większość przeglądarek, w celu oszczędzania tuszu, domyślnie nie drukuje obrazów tła. Można to zmienić w ustawieniach drukowania.

Pomiędzy oknem przeglądarki a zawartością strony jest widoczny odstęp.

Być może trzeba będzie zlikwidować marginsy i wypełnienia w elemencie `<body>`, przypisując im wartość 0. [Strony: 312, 313](#).

ELEMENTY PŁYWAJĄCE W UKŁADZIE

Pudełko elementu nie zostało umieszczone przy elemencie płynącym.

Trzeba się upewnić, że w elemencie zawierającym jest dostatecznie dużo miejsca, by oba te elementy zostały wyświetlone obok siebie.

[Strona 302](#).

Do szerokości elementu zostały dodane marginesy i wypełnienie (tak się dzieje we wszystkich przeglądarkach oprócz starych wersji IE, w których jest używany inny model pudełkowy). Dlatego jego szerokość może być większa niż ta podana we właściwości `width`. [Strona 315](#).

Czy została określona szerokość elementu płynącego? (Patrz kolejny punkt).

Moje elementy płyniące zajmują pełną szerokość okna przeglądarki (lub elementu zawierającego).

Trzeba sprawdzić, czy została podana szerokość tych elementów płynących.

[Strona 370](#).

Element zawierający moje elementy płyniące ma wysokość 1 piksela.

Element zawierający nie zna wysokości umieszczonej w nim elementów płynących. Można dodać element, który zostanie umieszczony ponizej elementów płynących lub użyć w elemencie zawierającym właściwości `overflow` i przypisać jej wartość `auto`. [Strona 372](#).

Przeglądarka Internet Explorer dodaje do elementów płynących dodatkowy margins.

W elementach tych należy użyć właściwości `display` i przypisać jej wartość `inline`.

ELEMENTY HTML

<a> 76, 78, 84, 440
<abbr> 52
<acronym> 52
<address> 54
<article> 431 – 438
<aside> 435
<audio> 216 – 219
**** 44
<blockquote> 51
<body> 21, 26
**
** 47
<button> 161
<cite> 53
<dd> 66
**** 55
<dfn> 53
<div> 186, 430, 439
<dl> 66
<dt> 66
**** 50
<embed> 202
<fieldset> 163
<figcaption> 118, 421, 438
<figure> 118, 421, 438
<footer> 431, 432
<form> 150

<h1> 21, 42
<h2> 21, 42
<h3> 42
<h4> 42
<h5> 42
<h6> 42
<head> 26
<header> 431, 432
<hgroup> 437
<hr /> 47
<html> 21
<i> 44
<iframe> 188 – 189
**** 98, 99
<input> 151, 154, 158 – 161, 165
<ins> 55
<label> 162
<legend> 163
**** 64, 65
<link> 234, 395
<meta> 190, 191
<nav> 431, 433
<object> 202
**** 64

<option> 156
<p> 21, 43
<q> 51
<s> 55
<script> 207, 217
<section> 432, 436
<select> 156
<source> 214, 219
**** 187
**** 50
<style> 235
<sub> 45
<sup> 45
<table> 130
<tbody> 134
<td> 130
<textarea> 153
<tfoot> 134
<th> 131
<thead> 134
<title> 26, 27
<tr> 130
<u> 55
**** 65
<video> 208, 212 – 215

ATRYBUTY HTML

action 150
align 102 – 104
alt 98
autoplay 213, 218
bgcolor 137
border 137
checked 154, 155
cite 51
class 183, 430
codecs 214
cols 153
colspan 132
content 190, 191
controls 213, 218
description 190
for 162
frameborder 189
atrybuty ogólne 182
height 99, 188, 213

href 76, 78, 87, 234
http-equiv 191
id 86, 87, 150, 162
id (atrybut) 182, 430
loop 213, 218
maxlength 151, 152
method 150
multiple 157
name 151 – 156, 159, 190
placeholder 167
poster 213
preload 213, 218
rel 234
robots 190
rows 153
rowspan 133
scrolling 189
seamless 189
selected 156
size 151, 152, 157
src 98, 188, 213, 214, 218, 219
style 235
target 85
title 52, 98
type 151, 154, 158
type (audio/video) 219, 214
type (CSS) 234 – 235
type="date" 165
type="email" 166
type="hidden" 161
type="image" 160
type="password" 152
type="search" 167
type="submit" 159
type="url" 166
value 154 – 156, 159
width 99, 136, 188, 213

WŁAŚCIWOŚCI CSS

background-attachment 413
background-color 249
background-image 412
background-image (gradienty) 418
background-position 414
background 415
background-repeat 413
border-bottom-color 310
border-bottom-style 309
border-collapse 339
border-color 310
border-image 318
border-left-color 310
border-left-style 309
border property 311
border-radius 320
border-right-color 310
border-right-style 309
border-spacing 339
border-style 309
border-top-color 310
border-top-style 309
bottom 365 – 370
box-shadow 319

clear 371
color 248
cursor 346
display 316
empty-cells 338
float 363, 369 – 374
float (obrazy) 410
@font-face 276
font-face 270
font-family 270, 272
font-size 273, 275
font-style 279
height (obrazy) 408 – 409
hsl / hsla 254 – 255
left 365 – 370
letter-spacing 283
line-height 282
list-style-image 333
list-style-position 334
list-style 335
list-style-type 332
margin-bottom 313
margin-left 313
margin 313
margin-right 313

margin-top 313
opacity 253
padding-bottom 312
padding-left 312
padding 312
padding (tabele) 336
padding-right 312
padding-top 312
position 365, 366
rgba 253
right 365 – 370
text-align 284
text-decoration 281
text-indent 286
text-shadow 287
text-transform 280
top 365 – 370
vertical-align 285
visibility 317
width (obrazy) 408 – 409
width (elementy płynące) 370
width (tabele) 336
word-spacing 283
z-index 363, 368

PSEUDOKLASY, PSEUDOELEMENTY ORAZ REGUŁY

:active 290
:first-letter 288
:first-line 288
:focus 290, 341

:hover 290, 342, 341
podświetlanie wierszy tabel 336
:link 289
:visited 289

@font-face 276
@import reguła 394
!important 238

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**