



Technologia i rozwiązania

Responsive Web Design z jQuery

Twórz strony na miarę XXI wieku!



Gilberto Crespo

[PACKT] open source*
PUBLISHING community experience distilled

Tytuł oryginału: Responsive Web Design with jQuery

Tłumaczenie: Wojciech Moch

ISBN: 978-83-246-9186-9

Copyright © 2013 Packt Publishing.

First published in the English language under the title „Responsive Web Design with jQuery”.

Polish edition copyright © 2014 by Helion S.A.

All rights reserved.

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

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

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

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

Wydawnictwo HELION

ul. Kościuszki 1c, 44-100 GLIWICE

tel. 32 231 22 19, 32 230 98 63

e-mail: helion@helion.pl

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

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

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

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

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

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

Spis treści

O autorze	9
<hr/>	
O recenzentach	11
<hr/>	
Wstęp	15
<hr/>	
Co znajdziesz w tej książce?	15
Czego potrzebujesz?	16
Dla kogo jest ta książka?	17
Konwencje	17
Pobranie przykładów dla książki	18
Errata	18
Piractwo	18
<hr/>	
Rozdział 1. Poznawanie projektów responsywnych	19
<hr/>	
Koncepcja responsywnego projektu stron	20
Porównanie stron responsywnych, płynnych i adaptacyjnych	21
Dopasowywanie strony WWW do ekranu za pomocą zapytań o media	22
Najpierw-mobilne	23
Używanie siatek	25
Ćwiczenie 1. — projektowanie z siatkami w modelu najpierw-mobilne	25
Podsumowanie	26
<hr/>	
Rozdział 2. Projektowanie responsywnych układów i siatek	27
<hr/>	
Dostosowywanie strony za pomocą języka JavaScript	28
Adapt.js	28
Respond.js	30
W jaki sposób wartości procentowe nadają strukturom elastyczność?	30
Konwersja pikseli na procenty	32
Czym jest system responsywnej siatki?	34
Systemy responsywnych siatek	35

Szablony siatek Photoshopa	42
Przygotowanie znacznika meta widoku	43
Ćwiczenie 2a — tworzenie układu siatek	44
Ćwiczenie 2b — wykorzystanie biblioteki Foundation4 do przygotowania struktury strony	45
Podsumowanie	50
Rozdział 3. Tworzenie responsywnego menu	51
Projektowanie menu w celu poprawienia użyteczności	52
Najczęściej używane wzorce responsywnego menu	53
Top nav	53
Footer anchor	55
Toggle	58
Menu Select	60
Footer-only	61
Multi toggle	62
Toggle and slide	65
Menu off-canvas	67
Ćwiczenie 3. — dopasowywanie menu z wykorzystaniem rozwiązania Toggle Menu	69
Podsumowanie	69
Rozdział 4. Projektowanie responsywnych tekstów	71
Konwertowanie tekstu na jednostki względne	72
Jednostka względna — procentowa	72
Jednostka względna — em	73
Jednostka względna — rem	74
Poprawianie wymiarowania elementów za pomocą właściwości box-sizing	74
Dopasowywanie rodziny czcionek do potrzeb nagłówków responsywnych	76
Narzędzie Font Squirrel do generowania czcionek	77
Wtyczka FitText	80
Wtyczka SlabText	81
Lettering	82
Kern.js	85
Responsive Measure	86
Ćwiczenie 4. — dopasowywanie nagłówka strony	87
Podsumowanie	88
Rozdział 5. Przygotowywanie obrazków i filmów	89
Prosta zmiana wymiarów obrazka za pomocą CSS	90
Stosowanie punktów zmiany obrazków	90
Jak działa znacznik picture?	92
Kontrolowanie nastaw responsywnych obrazków	92
Biblioteka CSS — Focal Point	93
Alternatywne rozwiązanie ze znacznikiem picture	95
Foresight — wybieranie właściwego obrazka dla danej wielkości ekranu	95
Picturefill — rozwiązanie najbardziej przypominające znacznik picture	98

Tworzenie responsywnych obrazków tła za pomocą wtyczek jQuery	98
Anystretch — łatwe rozciąganie tła	99
Backstretch — tworzenie responsywnego pokazu slajdów	101
Obsługa ekranów o wysokiej gęstości	103
Tworzenie responsywnych elementów filmowych	104
FitVids — szybkie rozwiązanie problemów z filmami	105
Ćwiczenie 5. — tworzenie różnych wersji obrazka na główną stronę witryny	106
Podsumowanie	108
Rozdział 6. Tworzenie responsywnych galerii obrazów	109
Responsywne galerie obrazów	109
Wtyczka Elastislide	110
FlexSlider2 — w pełni responsywna galeria	115
ResponsiveSlides — najlepsza z prostych galerii	119
Swiper — dotykowa galeria obrazków	123
Slicebox — animacja z użyciem obrazów z galerii	128
Wykorzystanie gestów dotykowych	129
Implementowanie zdarzeń dotykowych we wtyczkach JavaScript	131
QuoJS — prosta biblioteka interakcji dotykowych	131
Hammer — mała biblioteka wielodotyku	132
Ćwiczenie 6. — tworzenie galerii obrazków za pomocą wtyczki Swiper	133
Podsumowanie	134
Rozdział 7. Projektowanie tabel responsywnych	135
Tabele responsywne	135
Rozwijane tabele responsywne	136
Rozszerzenia do wtyczki	139
Stackedtables	141
Jak to zrobić, wykorzystując tabelę z poprzedniego przykładu?	141
Poziome przewijanie	143
Odwroćenie ułożenia nagłówka	145
Oдноśnik do pełnej tabeli	147
Ćwiczenie 7. — tworzenie tabeli responsywnej z cenami za pomocą wtyczki FooTable	149
Podsumowanie	149
Rozdział 8. Implementowanie formularzy responsywnych	151
Typy i atrybuty elementów formularzy	152
Funkcja autouzupelniania z wtyczką Magicsuggest	153
Jak to zaimplementować?	153
Funkcje wyboru daty i czasu	155
Pickadate — responsywne wybieranie daty i czasu	155
Funkcja podpowiedzi	157
Tooltipster — nowoczesna funkcja podpowiedzi	157
Tworzenie formularzy responsywnych za pomocą biblioteki IdealForms	158
Jak to zaimplementować?	159

Ćwiczenie 8. — tworzenie formularza kontaktowego za pomocą biblioteki IdealForms	162
Podsumowanie	163
Rozdział 9. Testowanie responsywnych stron	165
Symulowanie urządzeń za pomocą narzędzi dla przeglądarek	165
Narzędzie sieciowe Viewport Resizer	166
Narzędzie sieciowe Surveyor	166
Narzędzie sieciowe ScreenFly	168
Emulator przeglądarki Opera Mobile	169
Wskazówki dotyczące testowania responsywnych stron WWW	169
Ćwiczenie 9. — przetestujmy naszą stronę na różnych wielkościach ekranu	170
Podsumowanie	170
Rozdział 10. Prawidłowa obsługa przeglądarek	173
Sprawdzanie funkcji obsługiwanych przez przeglądarkę	174
CanIUse.com	174
Mobile HTML5.org	175
QuirksMode.org	175
Definiowanie rozwiązań zastępczych	176
Narzędzia do wykrywania dostępności funkcji	177
CSS Browser Selector +	177
Modernizr	178
YepNope.js	180
Implementacje wypełniaczy	181
MediaElements.js	181
SVG	183
Respond.js	184
Podsumowanie	185
Rozdział 11. Przydatne wtyczki responsywne	187
Wtyczki obsługujące strukturę strony	187
Tworzenie prostych struktur responsywnych za pomocą wtyczki Columns	188
Użycie wtyczki Equalize do dopasowania wymiarów elementów	191
Implementowanie strony z kafelkami za pomocą wtyczki Packery	192
Wtyczki tworzące menu nawigacyjne	194
Tworzenie menu bocznego za pomocą wtyczki Sidr	195
EasyResponsiveTabstoAccordion	197
Większa elastyczność menu dzięki wtyczce FlaxNav	198
Różne rozwiązania	200
SVGeezy	201
Prefix free	201
Magnific Popup	203
Riloadr	204
Calendario	206
Podsumowanie	209

Rozdział 12. Poprawianie wydajności stron	211
Wykorzystywanie sieci dostarczania treści	211
Zmniejszenie liczby żądań HTTP	212
Warunkowe ładowanie komponentów	212
Konsolidacja i minimalizacja zasobów (skryptów JavaScript i stylów CSS)	212
Technika CSS Sprite	213
Zmniejszanie wielkości przesyłanych plików	214
Progresywne obrazki JPEG	214
Optymalizacja obrazków	215
Upraszczenie stron za pomocą HTML5 i CSS3	216
Testowanie wydajności stron WWW	216
PageSpeed Insights	217
YSlow	218
WebPagetest	219
Mobitest	219
Podsumowanie	220
Skorowidz	221

Projektowanie responsywnych tekstów

Gdy mówimy o responsywnych nagłówkach, tak naprawdę mówimy o elastycznych tekstach. A zatem sztywne wielkości czcionek również powinny zostać zastąpione dynamicznymi. Różnica między takimi implementacjami wykonywanymi przed kilkoma laty a dzisiejszymi rozwiązaniami polega na tym, że dawniej rozważaliśmy tylko dwa sposoby prezentowania treści: na ekranie i na papierze. I choć czasy się zmieniają, to problemy z odpowiednim dopasowaniem tekstu pozostają nadal takie same.

Powszechnie uważa się, że typografia jest podstawą naszych projektów i osiłą wszystkich stylów CSS, ponieważ głównym zadaniem stron internetowych jest przekazywanie użytkownikowi informacji będącej odpowiedzią na jego pytania. Dobrze jest zatem zrezygnować z tworzenia dziesięciu rodzajów nagłówków, a to oznacza, że trzeba dobrze zaplanować tych kilka, które znajdują się na naszej stronie.

W tym rozdziale nauczymy się:

- konwertować tekst na jednostki względne,
- zmieniać odstępy w akapitach i elementach blokowych,
- modyfikować rodzinę czcionek w celu uzyskania ładnych i responsywnych nagłówków,
- automatycznie zarządzać rozmiarami czcionek.

Konwertowanie tekstu na jednostki względne

Jedną z największych zalet użycia jednostek względnych jest efekt kaskadowy wywoływany w momencie, gdy użytkownik zmienia podstawową wielkość czcionki w przeglądarce. W takiej sytuacji wszystkie rozmiary czcionek zmieniają się proporcjonalnie.

Dzisiaj niemal wszystkie przeglądarki mają domyślną wielkość elementu bazowego `<html>` ustaloną na 16 pikseli. Wartość ta może zostać jednak zmieniona w ustawieniach przeglądarki, co pozwala użytkownikowi powiększyć tekst, żeby był łatwiejszy do odczytania.

Zanim zaczniemy mówić o najczęściej stosowanych jednostkach miary, muszę opowiedzieć o dwóch szczególnych jednostkach, których popularność cały czas szybko wzrasta. Chodzi mi o jednostki `vw` (*viewport width* — szerokość widoku) i `vh` (*viewport height* — wysokość widoku).

Jednostki odnoszące się do widoku strony nie są obsługiwane we wszystkich przeglądarkach, dlatego dobrze jest zajrzeć na strony <http://www.w3.org/TR/css3-values/> lub <http://caniuse.com/viewport-units>. Warto przyjrzeć się tym jednostkom, ponieważ ułatwiają one skalowanie wielkości czcionek w zależności od wielkości okna przeglądarki.

Najpopularniejsze jednostki stosowane na stronach internetowych przedstawiać będą w kolejnych punktach.

Jednostka względna — procentowa

Wartości procentowe wyrażane są względem elementu nadrzędnego, dlatego najczęściej stosuje się je do definiowania struktury strony, tak jak zrobiliśmy to w rozdziale 2. Nie ma jednak żadnych przeciwwskazań, żeby jednostki tej użyć też do zdefiniowania wielkości czcionki. Proszę przyjrzeć się poniższemu przykładowi:

```
body {
    font-size: 100%; /* bazowe 16px */
}
p {
    font-size: 87.5%; /* 14px ÷ 16 */
}
```

Część twórców stron definiuje wielkość czcionki w elemencie `body` na 62,5 procenta (10 pikseli), co ułatwia późniejsze obliczenia. W takiej sytuacji przypisanie elementom strony wielkości czcionki o wartości 150% spowoduje, że będą one miały wielkość 15 pikseli, natomiast wartość 190% odpowiadać będzie wielkości 19 pikseli itd. Metoda ta pozwala na łatwiejsze określenie wielkości czcionek na stronie, działa ona jednak tylko na pierwszym poziomie kaskady określającej wielkości czcionek.

Jednostka względna — em

Jednostka em powiązana jest z wyliczoną wielkością czcionki elementu nadrzędnego. W poniższym przykładzie chcemy zdefiniować w elemencie potomnym czcionkę o wielkości 20 pikseli:

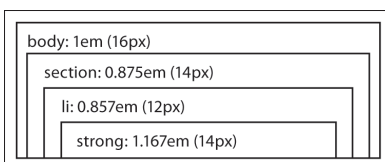
```
body {
  font-size: 1em; /* bazowe 16px */
}
p {
  font-size: 1.25em; /* 20px ÷ 16 */
}
```

W sieci dostępna jest przydatna strona <http://pxtoem.com>, która ułatwia wykonywanie takich obliczeń. Poniższy rysunek wyjaśnia, jak używać tego narzędzia. W lewej kolumnie należy podać wielkość bazową czcionki, wówczas w środkowej kolumnie ukażą się wyniki obliczeń. W prawej kolumnie natomiast można dokonywać podobnych obliczeń dla wielkości czcionek spoza zakresu od 6 do 24 pikseli.

The image shows a web-based calculator titled "Oh la la! Custom conversion" with the subtitle "Here's a calculator for your custom EM needs". It is divided into three main sections:

- 1. Enter a base pixel size:** A text input field contains "16" followed by a "px" unit label.
- 2. Convert:** This section has two columns: "PX to EM" and "EM to PX". The "PX to EM" column has an empty input field followed by "px". The "EM to PX" column has the number "3" followed by "em". Between the columns is the word "or". Below these inputs is a "Convert" button.
- 3. Result:** A large text area displays the result "48px".

Trzeba też pamiętać, żeby podczas wykonywania konwersji jednostek obok ostatecznie obliczonej wartości zawsze dopisywać wartość w pikselach (mówiłem już o tym w rozdziale 2.). Mówię o tym jeszcze raz, ponieważ obsługa wielkości czcionek na stronie związana jest ze stosowaniem wielu kaskadowych stylów, czego przykładem może być poniższy rysunek:



Jednostka względna — rem

Jednostka rem pochodzi ze specyfikacji CSS3 i jest skrótem od **root em** (bazowe em), czyli jest jednostką określaną względem głównego elementu dokumentu HTML. Oznacza to, że zmiana wielkości czcionki w elemencie `<html>` spowoduje przeskalowanie czcionek względem elementu bazowego strony, a nie elementów nadrzędnych. To jedyna różnica w stosunku do jednostek em.

Przyjrzyjmy się teraz, jak można wielkości z poprzednich przykładów przekształcić na jednostki rem:

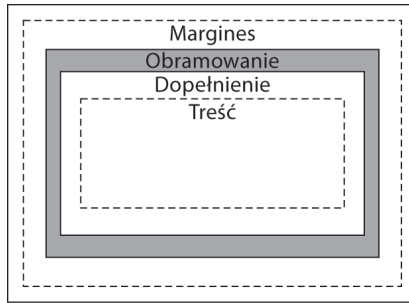
```
body {
    font-size: 1em; /* bazowe 16px */
}
section,
section li strong {
    font-size: 14px; /* definicja dla IE8 i wcześniejszych */
    font-size: 0.875rem; /* 14px ÷ 16 */
}
section li {
    font-size: 12px; /* definicja dla IE8 i wcześniejszych */
    font-size: 0.750rem; /* 12px ÷ 16 */
}
```

Taki sposób działania może zaoszczędzić nam sporo pracy i ułatwić analizę kodu, jeżeli strona wymagać będzie większej ilości pustego miejsca (w zależności od gęstości prezentowanych treści).

Spróbuj zaczynać projekty od przygotowania danych typografii wszystkich głównych elementów struktury, zanim jeszcze zostaną one zaprojektowane. Można to zrobić, tworząc dokument szablonu, w którym zapisane są wszystkie najważniejsze elementy HTML wraz z odpowiednim arkuszem stylów przygotowanym na potrzeby projektu witryny.

Poprawianie wymiarowania elementów za pomocą właściwości box-sizing

Na poniższym rysunku zilustrowany został model znany pod nazwą **box-model** (model polowy), w którym określenie całkowitej szerokości elementu wymaga obliczeń uwzględniających jego obramowanie i dopełnienie. Model ten powoli przestaje być stosowany:



W poniższym przykładzie przedstawiam sposób użycia modelu połowego do podzielenia dostępnej przestrzeni między dwa elementy `div`, z których każdy ma dopełnienie wielkości 5% oraz obramowanie o grubości 2 pikseli. W efekcie obliczenie szerokości tych elementów wcale nie jest proste:

```
div.splitted {
  padding: 0 5%;
  border: 2px solid black;
  float: left;
  width: ?; /* wartość rzeczywista= 50% - 10% - 4px */
}
```

W specyfikacji CSS3 znalazła się właściwość `box-sizing`, której można przypisać wartość `border-box`, co oznacza, że wartość szerokości (właściwość `width`) obejmuje już w sobie wielkości dopełnienia i obramowania. Wprawdzie właściwość ta dobrze funkcjonuje już w ósmej wersji Internet Explorera, nie jest jednak obsługiwana w jego wcześniejszych wersjach. Jeżeli musisz zapewnić stronie obsługę tych dwóch przeglądarek, to możesz wykorzystać dodatek, który załata te braki. Jest on dostępny pod adresem <https://github.com/Schep/box-sizing-polyfill>.

Zobaczmy teraz, jak ta zmiana wpływa na obliczenia wykonywane w naszym przykładzie:

```
div.splitted {
  padding: 0 5%;
  width: 50%;
  border: 2px solid black;
  float: left;
}
```

Jak widać, teraz nie ma już problemów z obliczeniem szerokości elementu i przeanalizowaniem wielkości pustych przestrzeni oddzielających elementy na ekranie. Dodatkowo wiele osób definiuje wielkość dopełnienia za pomocą jednostek `em`, ponieważ pozwala to uniknąć części obliczeń związanych z proporcjami.

```
div.splitted {
  padding: 0 0.5em;
  width: 50%;
}
```

```
border: 2px solid black;
float: left;
-webkit-box-sizing: border-box;
-moz-box-sizing: border-box;
box-sizing: border-box;
}
```

Aktualnie część przeglądarek nadal wymaga stosowania podanych wyżej przedrostków, dlatego nie można ich pominąć. Poza tym w podobny sposób można definiować dopełnienie elementów w pionie, na przykład tak:

```
div.splittd {
padding-top: 1em;
}
```

Dopasowywanie rodziny czcionek do potrzeb nagłówków responsywnych

Odpowiednie czcionki pozwalają na przygotowanie dla użytkowników naprawdę fascynujących stron oraz zachowanie spójnego stylu całej witryny bez konieczności przygotowywania setek obrazków na potrzeby nagłówków.

Czcionki trzeba wybierać naprawdę ostrożnie, ponieważ ich kształt wpływa na odbiór treści przez użytkowników. Źle dobrana czcionka może spowodować, że po przejrzaniu jednej lub dwóch stron użytkownik opuści witrynę.

Reguła CSS `@font-face` pozwala projektantom na wykorzystanie niestandardowych czcionek, które nie są zainstalowane na komputerach użytkowników, dzięki czemu umożliwia szybkie dopasowanie wyglądu nagłówków. Wielką zaletą zastosowania reguły `@font-face` jest to, że nie wymaga ona stosowania obrazków w nagłówkach, a jednocześnie zachowuje elastyczność niezbędną na stronach responsywnych.

Reguła `@font-face` obsługiwana jest przez IE8 oraz wersje wcześniejsze, pod warunkiem że wykorzystywane są czcionki typu EOT. Poza tym Safari iOS 4.1 oraz wersje wcześniejsze obsługują wyłącznie czcionki SVG. W dalszej części tego rozdziału opowiem, jak można rozszerzyć obsługę czcionek w tych przeglądarkach.

Na stronie <http://www.google.com/fonts> można przejrzeć ogromną bibliotekę czcionek wraz z informacją o ich autorach. Zalecam przeczytać informacje licencyjne każdej czcionki przed jej użyciem, aby uzyskać pewność, że można użyć jej komercyjnie.

Podstawowym zadaniem użycia właściwości `@font-face` jest zdefiniowanie reguły stylu. Za pomocą właściwości `font-family` definiowana jest nazwa czcionki, której można użyć w dalszej części dokumentu, natomiast właściwość `src` określa miejsce, w którym czcionka ta jest przechowywana. Z kolei właściwość `font-weight` (nie trzeba jej używać w przypadku normalnych czcionek, ale jest niezbędna, jeżeli chcemy użyć czcionek pogrubionych lub cienkich) definiuje wagę czcionki. Przykład takiej reguły podaję poniżej:

```
@font-face {
  font-family: "Kite One";
  src: url(/fonts/KiteOne-Regular.ttf);
}
```

Teraz można już użyć nowej czcionki, wykorzystując standardową składnię:

```
p {
  font-family: "Kite One", Helvetica, Arial, sans-serif;
}
```

Istnieją jednak lepsze metody. Przyjrzyjmy się teraz sposobom użycia takich narzędzi, jak Font Squirrel, Fit Text, Slabtext, Lettering i Responsive Measure.

Narzędzie Font Squirrel do generowania czcionek

Font Squirrel to świetne narzędzie pozwalające na przekształcenie dowolnej czcionki dla komputera stacjonarnego na wersję sprawdzającą się w sieci. Oprócz tego udostępniana jest usługa generowania prawidłowego kodu oraz plików na podstawie danych czcionki. I to wszystko całkowicie za darmo¹!

Na stronie <http://www.fontsquirrel.com/tools/webfont-generator> znajdziemy dodatkowe informacje o głównych funkcjach tego narzędzia, takie jak:

- Font Squirrel nie wymaga dogłębnej znajomości języka CSS,
- w trybie eksperta udostępnia dodatkowe opcje poprawiania szybkości ładowania strony,
- wygenerowane pliki i kod działają również w starszych przeglądarkach,
- wszystkie udostępniane zasoby są całkowicie darmowe.

Jak to zrobić?

Na początek trzeba otworzyć stronę Font Squirrel i kliknąć przycisk *Add fonts*, a następnie wybrać czcionkę, którą chcemy użyć na własnej stronie i dla której mamy odpowiednią licencję. W kolejnym kroku dobrze jest na początek zaznaczyć opcję *Basic* (wersja prosta) i pobrać wygenerowane pliki.

¹ Darmowa jest jedynie usługa. Pamiętaj, że każda czcionka ma swoją licencję, dlatego tak ważne jest jej przeczytanie przed zastosowaniem czcionki na swojej stronie.

Po rozpakowaniu zawartości pobranego pliku trzeba na początku pliku CSS dopisać poniższy kod, który pozwoli nam skorzystać z plików czcionek i użyć ich zależnie od sytuacji:

```
@font-face{
  font-family: 'kite_oneregular';
  src: url('kiteone-regular-webfont.eot');
  src: url('kiteone-regular-webfont.eot?#iefix') format('embedded-opentype'),
  url('kiteone-regular-webfont.woff') format('woff'),
  url('kiteone-regular-webfont.ttf') format('truetype'),
  url('kiteone-regular-webfont.svg#kite_oneregular') format('svg');
  font-weight: normal;
  font-style: normal;
}
```

Za każdym razem gdy będziemy chcieli wykorzystać nową czcionkę, wystarczy tylko podać jej nazwę zdefiniowaną w regule @font-face, na przykład tak jak poniżej:

```
p {
  font-family: 'kite_oneregular', Helvetica, Arial, sans-serif;
}
```

A wracając do strony pobierania Font Squirrel, mamy też możliwość obrania tylko podzbioru znaków danej czcionki, co bardzo zmniejsza wielkość pliku. W tym celu należy zaznaczyć opcje *Optimal* lub *Expert*. W ramach testów trzy razy przygotowaliśmy tę samą czcionkę Kite One, za każdym razem wybierając inną opcję (*Basic*, *Optimal* i *Expert*). Wyniki tego eksperymentu pokazują, że wielkość pliku jest wprost proporcjonalna do liczby znaków zapisanych w pliku czcionki oraz liczby zawartych w nim dodatkowych zasobów.

Ustawienie *Basic* nie zmienia liczby znaków w czcionce, ale już opcja *Optimal* ogranicza liczbę znaków do mniej więcej 256. W szczególnych przypadkach, w których czcionka Kite One miałaby mniej znaków, nie są wprowadzane żadne optymalizacje.

Największy potencjał do zmniejszania plików uzyskujemy po wybraniu trybu *Expert*, zaznaczeniu opcji *Basic Latin* i ręcznemu podawaniu poszczególnych znaków.

Spróbujmy teraz wygenerować czcionkę, wykorzystując tryb *Expert* narzędzia Font Squirrel:

1. Kliknij przycisk *Add Fonts* i wybierz plik czcionki, z którym chcesz pracować.
2. W sekcji *Rendering* usuń zaznaczenie z opcji *Fix Vertical Metrics*.
3. W sekcji *Font Formats* zaznacz opcję *SVG*, tak jak pokazano na poniższym rysunku.
4. W sekcji *Subsetting* wybierz pozycję *Custom subsetting*.
5. W sekcji *Language* zaznacz opcję *Polish*.
6. W sekcji *Unicode Tables* zaznacz opcję *Basic Latin*.

W tym miejscu zakładam, że w tekstach używane będą wyłącznie znaki alfabetu polskiego. Jeżeli używasz innych języków, wybierz potrzebne w nich znaki.

Add Fonts

You currently have no fonts uploaded.

BASIC
 Straight conversion with minimal processing.

OPTIMAL
 Recommended settings for performance and speed.

EXPERT...
 You decide how best to optimize your fonts.

Font Formats:
 TrueType
 EOT Lite
 SVG
 WOFF
 EOT Compressed

Truetype Hinting:
 Font Squirrel
 Keep Existing

Rendering:
 Fix Vertical Metrics
Normalize across browsers
 Fix GASP Table
Better DirectWrite Rendering
 Remove Kerning
Strip kerning data

Fix Missing Glyphs:
 Spaces
 Hyphens

X-height Matching:
 None
 100 %
 Arial
 Georgia
 Verdana
 Times New Roman
 Trebuchet
 Courier
 Resize to match the selected font's x-height.

Protection:
 WebOnly™
Disable desktop use

Subsetting:
 Basic Subsetting
Western languages
 Custom Subsetting...
Custom language support
 No Subsetting

Czasami w tekstach bardzo ważne są takie znaki, jak cudzysłów lub apostrofy, dlatego dobrze jest skopiować je i wkleić w polu *Single Characters*, tak jak na poniższym rysunku:

Single Characters:

Unicode Ranges:
Comma separated values. Can be single hex values and/or ranges separated with hyphens. Example: "0021-007E,00E7,00EB,00C7,00CB"

Subset Preview:

!	"	#	\$	%	&	'	()	*	+	,	-	.	/	0	1	2	3	4	5	6	7	8	
9	:	;	<	=	>	?	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
R	S	T	U	V	W	X	Y	Z	[\]	^	_	`	a	b	c	d	e	f	g	h	i	j
k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	{		}	~	¢	£	¥	€	™
«	®	'	»	Ó	ó	À	à	Â	â	Ä	ä	È	è	Ł	ł	Ń	ń	Ś	ś	Ž	ž	Ž	ž	°
<	-	—	‘	’	”	”	”	”	”	”	”	”	”	”	”	”	”	”	”	”	”	”	”	”

7. Skontroluj obraz wygenerowany w sekcji *Subset Preview* i w razie potrzeby wprowadź poprawki.
8. Po potwierdzeniu, że przesyłasz czcionkę, do której masz prawa, kliknij po prostu przycisk *DOWNLOAD YOUR KIT*.

Wtyczka FitText

FitText jest popularną wtyczką do biblioteki jQuery, która elastycznie ustala wielkość czcionek. Jest narzędziem o ciągle rosnącej popularności, ponieważ bardzo ułatwia tworzenie elastycznych czcionek. Jego działanie polega na takim skalowaniu tekstu nagłówka, żeby wypełnił on całą szerokość elementu nadrzędnego. Jeżeli chcesz zobaczyć przykład działania tej wtyczki, żeby się przekonać, na jak wielką elastyczność ona pozwala, to zajrzyj na stronę samej wtyczki pod adresem <http://fittextjs.com>.

Jak to zrobić?

Po pobraniu plików wtyczki z serwerów Github (<http://github.com/davatron5000/FitText.js>) do swojej strony HTML wstaw poniższy kod:

```
<h1 id="responsive_headline">Nagłówek używający FitText</h1>
```

Teraz na samym końcu kodu HTML (przed zamykającym znacznikiem `</body>`) trzeba jeszcze dołączyć bibliotekę jQuery i wtyczkę FitText. Na koniec należy jeszcze uruchomić kod wtyczki obsługującej nasz nagłówek:

```
<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
<script src="jquery.fittext.js"></script>
<script>
  $("#responsive_headline").fitText();
</script>
```

Domyślnie wtyczka ustali wielkość czcionki na 1/10 szerokości elementu. Istnieją dwie metody zmiany tego ustawienia: skonfigurowanie mechanizmu kompresji lub zdefiniowanie minimalnego i maksymalnego rozmiaru.

Jeżeli skorzystamy z kompresora, to musimy podać wartość kompresji. Na przykład wartość 1.2 spowoduje delikatniejsze zmiany wielkości tekstu, natomiast wartość 0.8 sprawi, że zmiany wielkości tekstu będą bardziej skokowe:

```
<script>
  $("#responsive_headline").fitText(1.2);
</script>
```

Możemy też zmodyfikować ten domyślny mechanizm kontroli wielkości tekstu, definiując jego minimalną i maksymalną wielkość. W ten sposób zyskujemy większą kontrolę niezbędną w sytuacjach, w których chcemy zachować hierarchię nagłówków, na przykład tak jak poniżej:

```
<script>
  $("#responsive_headline").fitText(1, { minFontSize: '20px',
  maxFontSize: '40px' });
</script>
```

Wtyczka SlabText

SlabText jest wtyczką umożliwiającą tworzenie ślicznych i całkowicie responsywnych nagłówków. Dzięki niej każdy może bardzo łatwo przygotować robiące wrażenie nagłówki. Skrypt dzieli każdy nagłówek na wiersze, a następnie każdy z nich powiększa tak, żeby wypełnił przestrzeń dostępną w poziomie. Idealna liczba znaków w wierszu wyliczana jest automatycznie przez podzielenie dostępnej szerokości przez wielkość czcionki w pikselach.

Poniżej wymieniam najważniejsze funkcje tej wtyczki:

- Jest w pełni responsywna i przygotowana do współpracy z urządzeniami mobilnymi.
- Opcja *Color Control* wybiera dla nagłówka kolor tła, tekstu oraz cienia.
- Opcje z grupy *Extra Options* pozwalają na ustalenie dopełnienia i zdefiniowanie długości cienia. Z kolei opcja *Image Overlay* umożliwia załadowanie obrazka i umieszczenie go w tle nagłówka za pomocą właściwości CSS3.
- Opcja *Font Control* pozwala wybrać czcionkę i ustalić wszystkie jej parametry.
- Opcja *Clonable* tworzy dowolną liczbę klonów danej sekcji tekstu.
- Wtyczka pozwala na ręczne złamanie wiersza.
- Zminimalizowana wersja wtyczki zajmuje tylko 4 KB.
- Nagłówki zajmujące sporo przestrzeni w poziomie lepiej zapełniają okno dowolnej przeglądarki.
- Trzeba się upewnić, że skrypt zostanie wywołany dopiero po załadowaniu wszystkich czcionek z reguły @font-face.

Kod wtyczki można pobrać pod adresem <http://github.com/freqDec/slabText/> i od razu rozpocząć eksperymentowanie.

Jak to zrobić?

Na początek trzeba dopisać do nagłówka header atrybut `id`, który później zostanie wykorzystany w skrypcie, a następnie tuż przed zamykającym znacznikiem `</body>` dodać znacznik `<script>`. Proszę spojrzeć na poniższy przykład:

```
<header>
  <h1 class="page-title">Linux rozkazuje: Nowi użytkownicy, ciągle nowi...
  szybcy i wściekli!</h1>
</header>
```

Poniżej przedstawiam kod, który trzeba dodać do tego samego pliku:

```
<script>
$(".page-title").slabText();
</script>
```

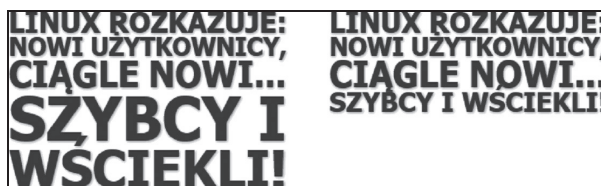
Można też wyłączyć automatyczne łamanie tekstu i samodzielnie wprowadzić miejsca podziału, zmieniając tym samym przekaz dla naszych użytkowników:

```
<header>
  <h1 id="specific-title"></h1>
</header>
```

Taki kod trzeba jeszcze połączyć z poniższym skryptem:

```
<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
<script src="js/jquery.slabtext.min.js"></script>
<script>
  var stS = "<span class='slabtext'>",
      stE = "</span>",
      txt = [
        "Linux rozkazuje:",
        "nowi użytkownicy,",
        "ciągle nowi...",
        "szybcy i wściekli!"];
  $("#specific-title").html(stS + txt.join(stE + stS) + stE).slabText();
</script>
```

Na poniższym rysunku zobaczyć można obie wersje nagłówka. Pierwotną po lewej, a po prawej wersję po modyfikacjach:



Dokładniejszy opis wszystkich opcji można znaleźć pod adresem <http://freqdec.github.io/slabText/>.

Lettering

Wtyczka Lettering pozwala na uzyskanie pełnej kontroli nad każdą literą nagłówka. Opakowuje ona poszczególne znaki, słowa i wiersze, nadając im klasy i tworząc tym samym całkowicie zarządzany kod, który umożliwia szybki dostęp do każdego elementu nagłówka z poziomu pliku CSS. Jest to wtyczka do biblioteki jQuery, co bardzo ułatwia dopasowywanie odstępów między literami, nadawanie im specjalnego wyglądu itp.

Istnieją dwie niesamowicie doskonale zaprojektowane strony, w których wiele pracy włożono w każdą literę nagłówka. Proszę spojrzeć na te przykłady dostępne pod adresami <http://lostworldsfairs.com/moon/> i <http://lostworldsfairs.com/eldorado/>:



Jak to zrobić?

Po pobraniu pliku zip ze strony <https://github.com/davatron5000/Lettering.js> możemy rozpocząć eksperymentowanie od dodania poniższego kodu do nowej strony HTML. Użyty tutaj kod zostanie wykorzystany za chwilę:

```
<h1 class="fancy-title">Tytuł strony</h1>
```

Nie można zapomnieć o dodaniu w znaczniku <head> odwołania do biblioteki jQuery i do wtyczki Lettering:

```
<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
<script src="js/jquery.lettering.js"></script>
```

Na samym końcu pliku HTML (przed zamykającym znacznikiem </body>) trzeba jeszcze wywołać skrypt określający, na której klasie ma działać nasza wtyczka:

```
<script>
  $(document).ready(function() {
    $(".fancy-title").lettering();
  });
</script>
```

Przygotowany w ten sposób kod wygeneruje poniższą strukturę. Wyobraź sobie, że musisz to zapisać bez pomocy wtyczki:

```
<h1 class="fancy-title">
<span class="char1">T</span>
<span class="char2">y</span>
<span class="char3">t</span>
<span class="char4">u</span>
<span class="char5">ř</span>
<span class="char6"></span>
<span class="char7">s</span>
<span class="char8">t</span>
<span class="char9">r</span>
<span class="char10">o</span>
<span class="char11">n</span>
<span class="char12">y</span>
</h1>
```

Tak przygotowanej strukturze można teraz nadawać style. Na przykład takie:

```
<style type="text/css">
  h1 { font-family: 'Impact'; font-size:50px;
    text-transform:uppercase; text-shadow: 1px 1px 2px #666; }
  .char1, .word1, .line1 { color: purple; }
  .char2, .word2, .line2 { color: orange; }
  .char3, .word3, .line3 { color: yellow; }
  .char4, .line4 { color: blue; }
  .char5 { color: green; }
  .char6 { color: indigo; }
  .char7 { color: violet; }
  .char8 { color: gold; }
  .char9 { color: cyan; }
  .char10 { color: lime; }
  .char11 { color: red; }
  .char12 { color: magenta; }
</style>
```

Co więcej, jeżeli nagłówek powinien otrzymać osobne style dla każdego słowa (nie znaku), to można o tym poinformować wtyczkę, podając jej parametr "words", tak jak poniżej:

```
<script>
  $(document).ready(function() {
    $(".fancy-title").lettering("words");
  });
</script>
```

Można też pomyśleć o osobnych stylach dla poszczególnych wierszy nagłówka rozdzielanych znacznikiem `
`:

```
<h1 class="fancy-title">Tytuł strony <br /> z długim tekstem</h1>
```

W stosunku do poprzedniego przykładu jedyna różnica polega na tym, że podajemy parametr "lines":

```
<script>
  $(document).ready(function() {
    $(".fancy-title").lettering("lines");
  });
</script>
```

Teraz zapewne się zastanawiasz, jak wiele pracy trzeba włożyć w przygotowanie stylów, odmierzanie odległości i wyliczanie, o ile trzeba powiększyć czcionkę każdego elementu. Do tych prac doskonale przyda się narzędzie Kern.js.

Kern.js

Kern.js jest narzędziem sieciowym doskonale uzupełniającym funkcje wtyczki Lettering.js, ponieważ udostępnia ono świetny interfejs pozwalający na przeciąganie elementów, dopasowywanie kerningu znaków, ustalanie wysokości wierszy i umiejscowienia liter. A po zakończeniu tych prac wystarczy skopiować wygenerowany kod CSS i użyć go w swoim arkuszu stylów.

Jak tego używać?

Na dole strony <http://www.kernjs.com/> znajdziesz odsyłacz *drag to bookmarks bar to install*. Przeciągnięcie go na pasek zakładek ułatwi później włączanie narzędzia na naszej stronie.

Aby narzędzie to działało prawidłowo, do nagłówka naszej strony należy dołączyć bibliotekę jQuery w wersji 1.7.2 oraz pliki samej biblioteki Kern.js. Jej skrypt można pobrać na stronie <https://github.com/bstro/kern.js>.

```
<script src="http://code.jquery.com/jquery-1.7.2.min.js"></script>
<script src="js/kern.min.js"></script>
```

Oto lista pięciu opcji, które pojawią się na górze strony, gdy tylko po jej otwarciu klikniemy znajdującą się w naszej przeglądarce zakładkę *Kernjs.com*:

- zmiana kerningu (poziomego odstępu między literami),
- zmniejszenie lub zwiększenie rozmiaru czcionki,
- dopasowanie wysokości wiersza (odstęp pionowy),
- dowolne umiejscowienie liter,
- obrót tekstu o pewien kąt.

Wszystkie te opcje reprezentowane są za pomocą poniższych ikon:



Gdy tylko zakończysz prace nad nagłówkiem, kliknij przycisk z „fajką”, aby zapisać wszystkie ustawienia w postaci kodu gotowego do skopiowania na naszą stronę.

Jeżeli na stronie masz kilka dopasowanych w ten sposób nagłówków, to musisz uważać, żeby kod jednego z nich nie wpływał na drugi. Aby ominąć ten problem, wystarczy elementom nagłówka przypisać specjalną klasę, tak jak w poniższym przykładzie:

```
<h1 class="fancy-title">Tytuł strony</h1>
```

Responsive Measure

Responsive Measure to prosty skrypt, któremu można przekazać selektor (najlepiej wybierający element, w którym znajdzie się właściwa treść). Dzięki niemu wygenerowana zostanie idealna definicja wielkości czcionki pozwalająca na dopasowanie tekstu do elementu. Brzmi jak magia? W następnym podpunkcie opiszę sposób użycia tego skryptu i dopasowania go do naszych wymagań.

Jak to zrobić?

Po pobraniu plików tego projektu ze strony <https://github.com/jbrewer/Responsive-Measure/> wstaw poniższy kod do sekcji <head> na samym początku dokumentu DOM:

```
<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
<script src="js/jquery.rm.js"></script>
```

Na końcu strony HTML (przed zamykającym znacznikiem </body>) trzeba jeszcze uruchomić skrypt wywołujący Responsive Measure. Zanim jednak do tego przystąpimy, musimy omówić dwie metody sterowania wielkością tekstu:

- **idealLineLength** — Domyślnie ma wartość 66, ale możemy ją zmienić, określając tym samym liczbę znaków i spacji, które mają się zmieścić w jednym wierszu. Poza tym minimalna wielkość czcionki ustalona jest na 16 pikseli, natomiast maksymalna

na 300 pikseli. Parametr `sampleText` ma pomagać przy obliczaniu liczby znaków o średniej szerokości znaku.

Pamiętaj, że wiersze, w skład których wchodzi od 45 do 75 znaków, powszechnie uznaje się za najlepiej nadające się do czytania.

A oto wspomniany wcześniej kod wywołujący bibliotekę:

```
<script>
$('section').responsiveMeasure({
  idealLineLength: 45,
  minimumFontSize: 16,
  maximumFontSize: 300,
  sampleText: "n"
});
</script>
```

- **minimumFontSize** i **maximumFontSize** — Za pomocą tych parametrów można poprawić obsługę wielkości czcionek, na przykład definiując 13 pikseli jako minimum i 30 jako maksimum. Poza tym domyślna wartość parametru `idealLineLength` może wpływać na wyniki obliczeń. Jeżeli tak się stanie, możesz to poprawić, definiując własną wartość, podobnie jak zrobiliśmy to w poprzednim przykładzie.

A poniżej przykład wykorzystania tych dwóch parametrów:

```
<script>
$('section').responsiveMeasure({
  minimumFontSize: 13,
  maximumFontSize: 30,
  sampleText: "n"
});
</script>
```

Ćwiczenie 4. — dopasowywanie nagłówka strony

To ćwiczenie wykonamy w trzech krokach, przy czym kroki drugi i trzeci będą już tylko uzupełnieniem pierwszego:

1. Najpierw użyj narzędzia Font Squirrel, żeby przygotować sobie specjalną czcionkę. Następnie zastosuj ją w znaczniku `<h1>` umieszczonym na Twojej stronie. Jeżeli nie wiesz, której czcionki użyć, to polecam wykorzystać czcionkę „Kite One” dostępną w serwisie Google Font.

2. Wykorzystaj wtyczkę Lettering, tak aby uzyskać pełną kontrolę nad każdą literą, słowem i wierszem nagłówka strony. Teraz zmień kolor pierwszej litery, zmieniając w pliku CSS wartość właściwości `color` dla klasy `.char1`.
3. Wywołaj narzędzie Kern.js, klikając jego zakładkę w przeglądarce. Następnie kliknij drugi przycisk na pasku narzędzi, wybierz pierwszą literę nagłówka i zwiększ wielkość jej czcionki. Teraz kliknij ostatni przycisk na pasku narzędzi i umieść wygenerowany kod w pliku CSS.

Podsumowanie

W tym rozdziale poznaliśmy metody tworzenia responsywnego tekstu i wielkości czcionek. Poza tym poznaliśmy kilka sposobów zmieniania czcionki na stronie, wykorzystując do tego takie narzędzia, jak Font Squirrel, FitText, SlabText, Lettering i Responsive Measure. Wszystkie te rozwiązania pozwalają na uzyskanie kontroli nad tekstem, która jest niezbędna podczas przygotowywania responsywnej strony WWW.

W następnym rozdziale mówić będziemy o obrazach oraz materiałach filmowych. Dowiemy się, jak można przekształcić je w responsywne media dopasowujące się do różnych urządzeń. Poza tym poznamy sposoby odtwarzania materiałów wideo za pomocą różnych odtwarzaczy i urządzeń.

Skorowidz

1140 Grid, 38, 39

A

accordion, *Patrz:* strona jako harmonijki

Adapt.js, 28, 29

Adaptive.js, 30

Adobe Flash Player, 104

Android, 115

Android 2.1, 22

Android 2.3, 201

animacja, 109, 128

CSS, 203

JavaScript, 203

Anystretch, 99, 101

art direction, *Patrz:* obraz ustalanie nastaw

B

Backstretch, 99, 101, 102

Balsamiq Mockups, 25

biblioteka

1140 Grid, *Patrz:* 1140 Grid

Adapt.js, *Patrz:* Adapt.js

Fluid Baseline Grid, *Patrz:* Fluid Baseline Grid

Focal Point, *Patrz:* Focal Point

Foundation4, *Patrz:*

Foundation4

html5shiv, *Patrz:* html5shiv

jQuery, *Patrz:* jQuery

Modernizr, *Patrz:* Modernizr

Respond.js, *Patrz:* Respond.js

Blip.tv, 105

box-model, *Patrz:* model polowy

breakpoint, *Patrz:* punkt zmiany

Bushella Davida, 145

C

Calendario, 206

CanIUse.com, 174

CDN, *Patrz:* sieć dostarczania treści

Chrome 4, 22, 115

Columns, 187, 188

Content Delivery Network, *Patrz:* sieć dostarczania treści

CSS, 20, 34, 201

image-set, *Patrz:* funkcja

image-set

reguła @font-face, 76, 77, 78

CSS Sprite, 213

CSS2.1, 22

CSS3, 22, 216

czas, 155

czcionka

cienka, 77

elementu

bazowego, 74

nadrzędnego, 73

EOT, 76

pogrubiona, 77

rodzina, 76

Safari, 76

SVG, 76

wielkość

dynamiczna, 71

względna, 72

D

data, 152, 155, 162

DeRouchey Bill, 24

Dimon Garret, 135

Drupal, 115

E

EasyResponsiveTabstoAccordion, 197

efekt

błyskania strony, 202

FOUC, 202

Eis Diego, 21

ekran, 20

dotykowy, 129, 131, 132

gęstość, 103

Retina, 91, 95, 98, 103

rozdzielczość, *Patrz:* ekran wielkość

rozmiar, 22, 38, 55, 168

szerokość, 178

wielkość, 22, 91, 188

Elastislide, 110, 111

element, *Patrz też:* znacznik

dopełnienie, 31

dotykowy, 130

em, 21, 33

email, 152, 162

Equalize, 187, 191

F

fallback, *Patrz:* przeglądarka rozwiązanie zastępcze

film, 89, 104, 203
 Firefox 3.6, 22, 115
 FirefoxOS, 175
 Fit Text, 77, 80
 FitVids, 105
 Flexible Math, 33
 FlexNav, 198
 FlexSlider2, 110, 115
 Fluid Baseline Grid, 35, 37
 Focal Point, 93

Font Squirrel, 77, 78, 87
 FooTable, 136, 137, 149
 rozszerzenia, 139
 Foresight, 95, 104, 215
 format
 JPEG, 214, 215, 216
 PNG, 201, 216
 SVG, 91, 183, 201
 formularz, 20, 151
 autocomplete, 152
 autouzupelnianie, 152, 153
 data, 152, 155, 162
 element, 152
 email, 152, 162
 kontrola danych, 152
 logowania, 51
 multiple, 152
 number, 152
 placeholder, 152, 159
 required, 152
 tel, 152, 162
 time, 152, 155
 tworzenie, 162

Foundation4, 39, 40, 45

Frost Brad, 53

funkcja

3D flow, 126
 image-set, 96
 podmiany obrazków na
 wersję HD, 98
 podpowiedzi, 157
 przejść CSS3, 203
 wypełniania pól formularzy, 20

G

geolokalizacja, 21
 Google Minify, 213
 Google+, 192
 Guideguide, 42, 43

Gustafson Aaron, 21
 gutter, *Patrz:* rymna

H

Hammer, 131, 132
 HTML5, 21, 175
 html5shiv, 37, 54, 177, 180

I

IdealForms, 158, 163
 Imageoptim, 216
 Input Switcher, 40
 Internet Explorer 6, 22, 37
 Internet Explorer 7, 40, 90, 115
 Internet Explorer 8, 22, 28, 37,
 40, 54, 90, 183, 201
 Internet Explorer 9, 22
 Internet Explorer Viewport
 Resizer, 166

J

JavaScript, 28
 jednostka
 em, *Patrz:* em
 PPI, *Patrz:* PPI
 px, *Patrz:* piksel
 vh, 72
 vw, 72
 względna, 21, 27, 72
 em, *Patrz:* em
 procentowa, 72
 rem, *Patrz:* rem
 język JavaScript, 28
 jPanelMenu.js, 67
 JPEGmini, 216
 jQuery, 21, 89, 111, 173
 formularze, 152
 jQuery++, 110, 111
 jQueryUI, 163

K

kalendarz, 206
 karuzela, 116
 ukrywanie, 37
 Kern.js, 85, 88

Kickstarter, 105
 kolumna
 płynna, 37
 wyródkowanie, 40
 kompresja, 215
 komputer stacjonarny, 21, 22,
 24, 25, 37, 91

L

laptop, 22
 Lettering, 77, 82, 84, 85, 88
 lista
 dni miesiąca, 206
 rozwijana, 60
 logo, 51

M

Magicsuggest, 153
 Magnific Popup, 203
 mapa, 203
 Marcotte Ethan, 20
 margines, 31, 34, *Patrz też:* rymna
 MediaElements.js, 181
 menu nawigacyjne, 51, 52, 194,
 195, 198
 Footer anchor, 53, 55, 61
 Footer only, 53, 61
 lokalizacja, 52, 53, 55
 Multi toggle, 53, 62, 65
 Off-canvas, 53, 67
 projektowanie, 53
 przełączane, 53, 58, 62, 65, 69
 Select menu, 53, 60
 Toggle and slide, 53, 65
 Toggle menu, 53, 58, 62, 69
 Top nav, 53
 mikroszablon, 203
 MobileHTML5.org, 175
 Mobitest, 219
 model polowy, 74
 Modernizr, 40, 54, 177, 178,
 179, 183

N

nagłówek, 71, 76, 81, 82
 Normalize, 37, 40

O

obraz, 89
 bitmapowy, 91
 efekty, 115
 galeria, 109, 110, 115, 116,
 118, 119, 126
 animowana, 128
 dotykowa, 123
 HD, 98
 JPEG, *Patrz:* format JPEG
 kadrowanie, 93
 kompresja, *Patrz:* kompresja
 kontekstowy, 204
 ładowanie, 204
 łączenie, 213, 214
 metadane, 215
 okienko wyskakujące, 203
 optymalizacja, 215, 216
 skalowanie, 33, 39, 90, 93
 tła, 98, 99, 100, 183
 dynamiczny, 101
 ukrywanie, 37
 ustalenie nastaw, 92
 wektorowy, 91, 183
 wymiary, 89, 90
 odtwarzacz, 182
 okienko
 modalne, 203
 wyskakujące, 203
 Opera 9.5, 22
 Opera 10, 115
 Opera Mobile, 22
 emulator, 169
 optymalizacja, 20

P

Packery, 187, 192
 PageSpeed Insights, 216, 217
 Pencil, 25
 Photoshop wtyczka, 42
 Pickadate, 155, 156, 163
 Picturefill, 95, 98, 215
 piksel, 30
 konwersja na procenty, 32, 33
 na cal, *Patrz:* PPI
 pixels per inch, *Patrz:* PPI

plik

CSS, 212
 minimalizacja, 213, 214
 PngGauntlet, 216
 pole wyszukiwania, 51
 polyfill, *Patrz:* wypełniacz
 Prefix free, 201
 priorytet, 20
 projekt
 adaptacyjny, 21
 nieresponywny, 21
 płynny, 21
 responywny, 20, 21, 27
 przeglądarka
 Chrome, *Patrz:* Chrome
 emulator, 165, 169
 Firefox, *Patrz:* Firefox
 funkcje, 174, 175, 176, 177
 Internet Explorer, *Patrz:*
 Internet Explorer
 język, 178
 obsługa, 173, 176, 177
 Opera, *Patrz:* Opera
 przedrostek, 201
 rozwiązanie zastępcze, 176, 177
 Safari, *Patrz:* Safari
 testowanie, *Patrz:* testowanie
 ujednolicanie, *Patrz:*
 wypełniacz, wstawka
 ułożenie okna, 178
 wersja, 178
 punkt zmiany, 89, 90, 136, 165

Q

QuirksMode.org, 175
 QuoJS, 131
 QuosJS, 132

R

rem, 74
 RequireJS, 212
 Respond.js, 22, 28, 30, 37, 184
 Responsive Measure, 77, 86
 Responsive Nav, 58
 ResponsiveSlides, 110, 119
 Riloadr, 204, 205
 RIOT, 216

root em, *Patrz:* rem
 rymna, 34

S

Safari, 22, 115
 czcionki, 76
 Samy Silva Maurício, 31
 ScreenFly, 168
 siatka, 20, 25
 linii bazowej, 37
 nieresponywna, 34
 płynnej linii bazowej, 37
 responywna, 34, 35, 188
 Fluid Baseline Grid, *Patrz:*
 Fluid Baseline Grid
 zagnieżdżanie, 40, 41
 sieć dostarczania treści, 211, 212
 skrypt, 212
 Slabtext, 77, 81
 Slicebox, 110, 128
 smartfon, 22, 25, 31, 37, 89, 91,
 130
 SpriteCow, 214
 Stackedtables, 141
 Stitches, 214
 kompresja, 215
 strona
 analiza, 216, 217, 218, 219
 błyskanie, 202
 czas ładowania, 211, 216, 217,
 218, 219
 jako harmonijki, 197
 jako zbiór kart, 191
 widok, 43
 wydajność, 211, 216, 217,
 218, 219
 z kafelkami, 192
 styl CSS, *Patrz:* CSS
 Surveyor, 166
 SVGeezy, 201
 swipe events, *Patrz:* zdarzenie
 przeciągania
 Swiper, 110, 123
 system
 operacyjny, 178
 siatki, 34
 responywny, 34, 35, 188
 szablon, 74
 adresów URI, 96

T

tabela, 135, 141
 filtrowanie, 140
 klucz-wartość, 141
 odsyłacz, 135, 147
 pełna, 147
 przewijana poziomo, 135, 143
 responsywna, 21
 rozwijana, 135, 136
 sortowanie, 139
 sztaplowana, 135
 wiersz nagłówków, 145
 z wielką liczbą kolumn, 143
 tablet, 22, 25, 37, 89, 91, 131
 mały, 22
 telefon, 152, 162
 Ten Alexey, 183
 testowanie, 165, 166, 168, 169,
 173, 174, 175, 176, 216, 217
 TinyNav.js, 60, 61
 Tooltipster, 157
 typografia, 71

U

urządzenie mobilne, 24, 219,
 Patrz też: tablet, smartfon

V

Viddler, 105
 viewport, *Patrz:* strona widok
 Viewport Resizer, 166
 Vimeo, 105

W

Walton Trent, 27
 warunkowe ładowanie
 komponentów, 212
 WebPagetest, 219
 webshim, *Patrz:* wstawka
 wielodotyk, 132
 wireframe.cc, 25
 właściwość
 box-sizing, 74, 75
 font-family, 77
 font-weight, 77
 src, 77
 WordPress, 115
 Wroblewski Luke, 22, 129, 130
 wstawka, 176, 177
 wtyczka
 jQuery, 21
 Photoshop, 42
 wypełniacz, 173, 176, 184
 implementacja, 181

Y

Yahoo!, 218
 YepNope.js, 177, 180, 212
 YouTube, 105
 YSlow, 216, 218
 YUI Compressor, 213

Z

zapytanie o media, 22, 28, 29,
 103, 141, 147
 zdarzenie
 przeciągania, 110
 związane
 z dotykiem, 20
 z myszą, 20
 znacznik
 article, 216
 audio, 181
 div, 31, 216
 footer, 216
 h1, 54
 header, 51, 216
 html, 177
 iframe, 105
 img, 33, 96, 97, 183, 205
 meta, 43
 nav, 54, 216
 ol, 60
 picture, 92, 95, 98
 select, 60
 span, 98, 216
 svg, 183
 ul, 60
 video, 104, 105, 181
 znak
 \$\$, 132
 biały, 213

Ż

żądanie HTTP, 212

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>

Responsive Web Design z jQuery

Jeszcze całkiem niedawno — w świecie bez smartfonów i tabletek — tworzenie stron było zdecydowanie prostsze. Projektanci i deweloperzy nie musieli się martwić o to, jak strona będzie się prezentowała na różnej wielkości ekranach, dopasowywali ją do dwóch, trzech rozdzielczości. Obecnie, w erze urządzeń mobilnych o poziomych, pionowych, małych, średnich i bardzo dużych ekranach, pewnie często zastanawiasz się, jak skutecznie zadbać o responsywność swojej strony. Sięgnij po ten przewodnik i spraw, by użytkownicy zarówno komputerów stacjonarnych, jak i laptopów, tabletek czy smartfonów z przyjemnością korzystali z Twojej witryny.

W trakcie lektury dowiesz się, jak budować responsywne strony WWW. Responsywne, czyli dostosowujące swój wygląd i układ do urządzenia, na którym są przeglądane. Nauczysz się korzystać z dopasowujących się układów i siatek (ang. grids) oraz zbudujesz menu, które wszędzie będzie wyglądało dobrze. Ponadto przekonasz się, że stworzenie galerii, osadzenie mediów oraz tabeli dla urządzeń mobilnych nie musi być karkołomnym zadaniem! Na sam koniec dowiesz się, jak testować takie strony, oraz poznasz przydatne wtyczki dla jQuery. Książka ta jest doskonałym źródłem wiedzy dla wszystkich podążających za trendami w tworzeniu stron WWW!

Zaoszczędź swój czas i twórz uniwersalne strony WWW!

helion.pl
księgarnia internetowa

Nr katalogowy: 22586



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



Zamówienia telefoniczne:
0 801 339900



0 601 339900

[PACKT] open source*
PUBLISHING community experience distilled



Helion

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

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



Sięgnij po książkę i dowiedz się, jak:

- stworzyć atrakcyjne menu i galerię
- wykorzystać możliwości małych ekranów
- zbudować responsywną tabelę
- zastosować dodatkowe wtyczki
- zbudować uniwersalną stronę WWW

sięgnij po **WIĘCEJ**



KOD KORZYSCI

ISBN 978-83-246-9186-9



9 788324 691869

Cena: 49,00 zł

Informatyka w najlepszym wydaniu