



Technologia i rozwiązania

# Sass i Compass

## Praktyczny przewodnik dla projektantów

Projektuj pięknie i wygodnie  
— Sass i Compass do Twoich usług!



Ben Frain



Tytuł oryginału: Sass and Compass for Designers

Tłumaczenie: Łukasz Piwko

ISBN: 978-83-283-1762-8

Copyright © Packt Publishing 2013

First published in the English language under the title 'Sass and Compass for Designers'  
— 9781849694544.

Polish edition copyright © 2016 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](mailto: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/saasco.zip>

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

Printed in Poland.

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

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

# Spis treści

<b>Zespół wydania oryginalnego</b>	<b>9</b>
<b>Przedmowa</b>	<b>11</b>
<b>O autorze</b>	<b>13</b>
<b>O korektorach merytorycznych</b>	<b>15</b>
<b>Wprowadzenie</b>	<b>17</b>
<b>Rozdział 1. Sass i Compass dla początkujących</b>	<b>21</b>
<b>Do czego służą preprocesory CSS</b>	<b>22</b>
Skoro ostatecznym wynikiem jest kod CSS, to lepiej od razu pisać kod CSS	23
<b>Dlaczego należy używać właśnie Sass i Compass</b>	<b>23</b>
Zmienne (aby wartości definiować tylko raz)	23
Automatyczne kolory w formacie RGBA i konwersje	24
Zapomnij o przedrostkach producentów	25
Zagnieżdżanie reguł	26
Prostsze zapytania medialne	27
Automatyczna kompresja kodu CSS i przyspieszenie działania strony	28
<b>Czym jest Sass</b>	<b>29</b>
<b>Czym jest Compass</b>	<b>29</b>
<b>Instalacja Sass i Compass</b>	<b>30</b>
Instalacja pakietu dla systemu OS X	30
<b>Instalacja Sass i Compass i praca z nimi w wierszu poleceń</b>	<b>31</b>
Instalacja interpretera Ruby w systemie Windows	31
Polecenie gem	32
Instalacja z wiersza poleceń w systemie Mac OS X	32
Instalacja z wiersza poleceń w systemie Windows	33
Sprawdzanie wersji Compass i Sass	35
Sprawdzanie, które wersje Sass i Compass są dostępne	35
Instalowanie najnowszej wersji Sass i Compass (włącznie z wydaniem wstępnym)	36
Tworzenie projektu Sass i Compass z poziomu wiersza poleceń	37
Automatyczne kompilowanie kodu na CSS z poziomu wiersza poleceń	38

<b>Graficzne narzędzia do pracy z Sass i Compass</b>	<b>38</b>
Scout	38
CodeKit	39
LiveReload	40
<b>Praca z plikami Sass w edytorach tekstu</b>	<b>41</b>
<b>Podsumowanie</b>	<b>42</b>
<b>Rozdział 2. Tworzenie projektu Sass i Compass</b>	<b>43</b>
<b>Tworzenie projektu Sass i Compass</b>	<b>44</b>
Tworzenie projektów Compass	45
<b>Struktura pliku config.rb</b>	<b>48</b>
Dodawanie niezbędnych wtyczek	49
Ustawianie nazw i ścieżek do zasobów	50
Ustawianie stylu wyjściowego CSS	50
<b>Tworzenie i używanie plików częściowych</b>	<b>54</b>
Sass zapewnia kod gotowy do produkcji i łatwy w obsłudze	55
Importowanie pliku częściowego	56
Składnia definicji zmiennych w Sass	57
<b>Rodzaje komentarzy w języku Sass</b>	<b>58</b>
Standardowe komentarze CSS	58
Jednowierszowe komentarze Sass	58
<b>Podstawowy plik index.html</b>	<b>59</b>
<b>Podstawa dla przyszłych projektów</b>	<b>60</b>
<b>Podsumowanie</b>	<b>60</b>
<b>Rozdział 3. Zagnieżdżanie, rozszerzanie, symbole zastępcze i domieszki</b>	<b>63</b>
<b>Stylizowanie strony za pomocą technologii Sass i Compass</b>	<b>64</b>
<b>Oddzielanie układu od elementów wizualnych</b>	<b>65</b>
<b>Co to jest zagnieżdżanie kodu i jak umożliwia ono tworzenie modułów</b>	<b>68</b>
Składanie zagnieżdżania	68
Dodawanie stylów Modernizr za pomocą selektora rodzica	73
Zagnieżdżanie przestrzeni nazw	79
<b>Rozszerzanie istniejących reguł za pomocą dyrektywy @extend</b>	<b>82</b>
<b>Rozszerzanie stylów, gdy jest taka konieczność, za pomocą selektorów zastępczych</b>	<b>84</b>
<b>Co to są domieszki i jak za ich pomocą definiować często używane fragmenty kodu</b>	<b>85</b>
Podstawowa składnia domieszek	87
Jak pisać domieszki z ustawieniami domyślnymi	88
<b>Ostrzeżenie co do generowanego kodu CSS</b>	<b>91</b>
<b>Podsumowanie</b>	<b>91</b>
<b>Rozdział 4. Praca z kolorami</b>	<b>93</b>
<b>Wystarczy jedna definicja koloru</b>	<b>94</b>
<b>Funkcje rozjaśniania i przyciemniania kolorów</b>	<b>95</b>
Kolory HSL	95
Składnia rozjaśniania i przyciemniania kolorów	96

<b>Modyfikacja znaczników</b>	<b>97</b>
Technika clearfix w Compass	98
<b>Powrót do kolorów</b>	<b>100</b>
Domieszki w domieszkach? Ki diabeł?	102
<b>Funkcje complement i invert</b>	<b>105</b>
<b>Funkcja inwersji</b>	<b>106</b>
<b>Funkcja adjust-hue</b>	<b>107</b>
<b>Funkcje saturacji i desaturacji</b>	<b>108</b>
<b>Funkcje przezroczystości i zanikania</b>	<b>109</b>
<b>Funkcje opacify i fade-in</b>	<b>110</b>
<b>Funkcja grayscale</b>	<b>110</b>
<b>Funkcja rgba</b>	<b>111</b>
<b>Funkcja mix</b>	<b>112</b>
<b>Funkcja adjust-color</b>	<b>113</b>
<b>Funkcja scale-color</b>	<b>114</b>
<b>Funkcje shade i tint</b>	<b>115</b>
<b>Połączenie wszystkiego razem</b>	<b>116</b>
<b>Podsumowanie</b>	<b>117</b>
<b>Rozdział 5. Responsywne i elastyczne siatki</b>	<b>119</b>
<hr/>	
<b>Argumenty przeciwko siatkom</b>	<b>120</b>
<b>Argumenty za używaniem systemów siatkowych</b>	<b>121</b>
<b>Co to jest Susy</b>	<b>121</b>
Co tak naprawdę robi Susy	122
<b>Instalowanie wtyczki do Compass Susy</b>	<b>123</b>
<b>Dołączanie Susy do projektu</b>	<b>124</b>
Zmienne projektowe	124
<b>Tworzenie siatki Susy</b>	<b>125</b>
<b>Definiowanie kontekstu dla siatki</b>	<b>125</b>
Pokazywanie siatki w tle	127
Ustawianie modelu blokowego	128
<b>Tworzenie responsywnej siatki z myślą o urządzeniach przenośnych</b>	<b>128</b>
<b>Tworzenie punktów brzegowych przy użyciu Susy</b>	<b>129</b>
<b>Tworzenie całkiem płynnej siatki</b>	<b>134</b>
<b>Tworzenie statycznej siatki o stałej szerokości</b>	<b>135</b>
<b>Używanie udogodnień dostarczanych przez Susy</b>	<b>136</b>
Domieszki prefix, suffix i pad	136
Domieszka prefix	136
Domieszka suffix	138
Domieszka pad	139
<b>Domieszki pre, post, squish, push i pull</b>	<b>140</b>
Domieszka pre	140
Domieszka post	140
Domieszka squish	141
Domieszki push i pull	141

<b>Siatki w siatkach</b>	<b>141</b>
Domieszka nth-omega	142
Pozycjonowanie w odniesieniu do kontenera	144
Standardowa składnia i wyniki Susy	145
<b>Podsumowanie</b>	<b>147</b>
<b>Rozdział 6. Tworzenie zapytań medialnych za pomocą Sass i domieszek</b>	<b>149</b>
<b>Zapytania medialne w Sass</b>	<b>150</b>
Przenoszenie stylów dotyczących zapytań medialnych do osobnego pliku częściowego	151
<b>Śródliniowe zapytania medialne z Sass</b>	<b>151</b>
<b>Tworzenie domieszki w celu ułatwienia sobie pracy z zapytaniami medialnymi</b>	<b>153</b>
Definiowanie punktów brzegowych jako zmiennych	153
<b>Jak działa domieszka MQ</b>	<b>154</b>
<b>Wariacje motywu</b>	<b>155</b>
<b>Pisanie śródliniowych zapytań medialnych</b>	<b>156</b>
<b>Kompresja GZIP i CSS = zwycięstwo</b>	<b>161</b>
<b>Jaka jest prawdziwa różnica między rozsianymi i zgrupowanymi zapytaniami medialnymi</b>	<b>162</b>
<b>Robienie przeglądu kodu CSS</b>	<b>164</b>
<b>Podsumowanie</b>	<b>166</b>
<b>Rozdział 7. CSS3, duszki i inne cuda</b>	<b>167</b>
<b>Domieszki Compass do CSS3</b>	<b>168</b>
Składnia domieszki własności text-shadow	168
<b>Składnia własności border-radius</b>	<b>170</b>
<b>Kolumny</b>	<b>171</b>
Składnia reguły kolumnowej	172
<b>Domieszka box-shadow</b>	<b>172</b>
Składnia domieszki box-shadow	173
Definiowanie wielu cieni	173
<b>Gradienty w tle</b>	<b>175</b>
Składnia gradientu liniowego	175
Składnia gradientu promienistego	177
Kombinacje obrazów i gradientów w tle	177
<b>Dodawanie obrazów do tła za pomocą funkcji pomocniczej Compass image-url</b>	<b>178</b>
<b>Określanie szerokości i wysokości obrazów za pomocą funkcji pomocniczych</b>	<b>179</b>
Pogromca bufora Compass	179
<b>Duszki graficzne</b>	<b>180</b>
Dodatkowe opcje konfiguracyjne duszków	183
Opcje układu	184
<b>Domieszki do zamiany tekstu</b>	<b>184</b>
Domieszka hide-text	184
Domieszka squish-text	185
Zamienianie tekstu na obraz	186
<b>Tworzenie adresów URI z danymi z obrazów</b>	<b>187</b>
Składnia wstawiania obrazów śródliniowych	188

<b>Rozwiązania awaryjne na wypadek braku obsługi SVG</b>	<b>190</b>
<b>Przekształcenia CSS</b>	<b>190</b>
<b>Filtry CSS</b>	<b>193</b>
<b>Przejścia</b>	<b>195</b>
<b>Podsumowanie</b>	<b>195</b>
<b>Rozdział 8. Logika programistyczna w Sass</b>	<b>197</b>
<b>Obliczenia matematyczne w Sass</b>	<b>198</b>
Dodawanie	198
Odejmowanie	198
Mnożenie	198
Dzielenie	199
Wykonywanie obliczeń przy użyciu zmiennych	200
<b>Dyrektywy sterujące</b>	<b>201</b>
Dyrektywy sterujące @if i @else if	201
Pętla @for	203
Zmienna licznikowa	204
Instrukcje from to i from through	205
Pętla @each	206
<b>Usuwanie i dodawanie jednostek</b>	<b>209</b>
Usuwanie jednostki z wartości zmiennej	209
Dodawanie jednostki do wartości zmiennej	210
<b>Pisanie funkcji w Sass</b>	<b>210</b>
Operatory równości i nierówności	211
Operatory relacyjne	212
Dyrektywa @return	212
Sposób użycia wyniku działania funkcji	212
<b>Dyrektywa @debug</b>	<b>213</b>
<b>Dyrektywa @warn</b>	<b>215</b>
<b>Podsumowanie</b>	<b>215</b>
<b>Rozdział 9. Sass i Compass dla zaawansowanych</b>	<b>217</b>
<b>Wyłączenie obsługi wybranych przeglądarek w Compass</b>	<b>218</b>
Zmienne konfiguracyjne	219
Opera przeszła na silnik WebKit	220
<b>Dodawanie eksperymentalnej obsługi dla najnowszych własności CSS</b>	<b>221</b>
<b>Definiowanie wartości eksperymentalnych</b>	<b>223</b>
<b>Interaktywna powłoka Sass</b>	<b>223</b>
Interaktywny Compass	224
<b>Dodawanie wtyczki Sass do importowania plików partiami</b>	<b>224</b>
<b>Tworzenie wielu osobnych plików CSS</b>	<b>226</b>
<b>Konwertowanie plików częściowych na samodzielne arkusze stylów</b>	<b>226</b>
<b>Statystyki Compass</b>	<b>226</b>
<b>Kasowanie zawartości bufora Sass</b>	<b>228</b>
<b>Wybieranie metody kompilacji kodu Sass na jeden raz</b>	<b>229</b>

<b>Sprawozdanie z misji</b>	<b>229</b>
Naprawianie błędów ludzkich	229
Wychwytywanie typowych problemów za pomocą narzędzia lint	231
<b>Fakty, nie przypuszczenia</b>	<b>232</b>
Wszyscy kochają Narzędzia dla programistów przeglądarki Chrome	232
Wyszukiwanie nieużywanych stylów	235
<b>Uwagi końcowe</b>	<b>236</b>
<b>Podsumowanie</b>	<b>236</b>
<b>Skorowidz</b>	<b>237</b>

---



# Sass i Compass dla początkujących

Arkusze napisane przy użyciu Sass i Compass są elastyczniejsze i łatwiejsze w obsłudze serwisowej, tworzy się je szybciej niż tradycyjne arkusze stylów. Dlatego właśnie takie firmy jak eBay, bet365.com, BBS, Instagram, LinkedIn, Square i Groupon korzystają z tych rozwiązań w swojej działalności.

Przez długi czas nie mogłem przekonać się do Sass i Compass, ponieważ bałem się początkowych trudności. Na samą myśl, że musiałbym użyć wiersza poleceń, aby wszystko przygotować, dostawałem gęziej skórki. Gdy tylko napotykałem instrukcje w rodzaju „zainstaluj *gemę* Ruby” albo „wykonaj polecenie watch”, było po mnie. Cytując Doktora Zła: „Niech mi ktoś, do diabła, pomoże!”.

Jeśli jesteś projektantem, to mogłeś mieć podobne odczucia do moich. Celem tej książki jest wyjaśnienie, jak działają Sass i Compass, oraz ułatwienie rozpoczęcia pracy z nimi. Dzięki temu będziesz mógł wykorzystać wszystkie możliwości tych technologii w swojej pracy. Jednocześnie zapewniam Cię, że jeśli znasz HTML i CSS, to bez trudu opanujesz też Sass i Compass.

Sass to według oficjalnego opisu metajęzyk, choć częściej określa się go preprocesorem CSS. Jak zwał, tak zwał. Ważne jest to, że Sass jest łatwy w użyciu. Kod pisze się w jednym pliku (pliku Sass z rozszerzeniem *.scss*), który w chwili zapisywania jest konwertowany na dobrze Ci znany kod CSS (a jeśli nie znasz CSS, odłóż tę książkę i sięgnij na inną półkę).

Pliki Sass tworzy się bardzo łatwo. Wystarczy wziąć dowolny plik CSS i zmienić jego rozszerzenie *.css* na *.scss*. W ten sposób powstanie kompletny plik Sass, do którego można dodawać elementy składni języka Sass. Traktuj go po prostu jak CSS z turbodoładowaniem.

Jeśli wciąż martwisz się wierszem poleceń, to przestań już się lękać. Powstało kilka łatwych w obsłudze narzędzi z graficznym interfejsem użytkownika do pracy z Sass i Compass. Ich opis znajduje się w dalszej części tego rozdziału (w której opisuję też techniki pracy z wierszem poleceń dla tych, którzy są w bojowym nastroju).

Jako że ta książka jest przeznaczona dla projektantów, oprócz wychwalania wszelkich zalet Sass i Compass w rozdziale tym znajdziesz też wyjaśnienie, co w ogóle oznaczają te dwie nazwy, co wiąże te dwie technologie oraz jak je zainstalować. Potem będziesz gotowy do rozpoczęcia pracy nad pierwszym projektem.

Podsumowując, w rozdziale tym dowiesz się:

- dlaczego preprocesory CSS są potrzebne;
- dlaczego należy używać właśnie Sass i Compass;
- czym jest Sass;
- czym jest Compass i jaki ma związek z Sass;
- jak zainstalować Sass i Compass w systemach OS X i Windows;
- jakie są dostępne graficzne narzędzia, którymi można zastąpić wiersz poleceń;
- jakie są różne składnie Sass.

## Do czego służą preprocesory CSS

CSS to język deklaracji, a nie programowania. Oznacza to, że własności i wartości stylistyczne wpisywane w arkuszach stylów są wykorzystywane bezpośrednio przez przeglądarkę do rysowania na ekranie. Język programowania natomiast służy do definiowania pewnej logiki. Mówiąc prościej, instrukcja logiczna może mieć następującą treść: jeśli elementy `h1` znajdują się w elemencie `nav`, pokoloruj je na niebiesko; jeśli znajdują się w elemencie `header`, nadaj im kolor czerwony. Ponadto języki programowania zawierają **zmienne**. Są one czymś w rodzaju magazynów do przechowywania różnych rzeczy, które mogą być przydatne w późniejszym czasie (na przykład w zmiennej można zapisać wartość koloru). Co więcej, w językach programowania występują **funkcje** służące do wykonywania działań na wartościach (na przykład rozjaśnij ten kolor o 20 procent). Sass i Compass pozwalają na korzystanie z tych i wielu innych mechanizmów.

Nie martw się, jeśli nie zrozumiałeś wszystkich słów w poprzednim akapicie. Wszystko się wyjaśni w odpowiednim czasie. Jednak najpierw rozprawimy się z kilkoma mitami na temat preprocesorów CSS.

## Skoro ostatecznym wynikiem jest kod CSS, to lepiej od razu pisać kod CSS

Gdy pierwszy raz natknąłem się na preprocesor CSS Sass, pomyślałem: „Skoro ostatecznym wynikiem jest kod CSS, to czy nie lepiej od razu pisać kod CSS?”. Podobnie zareagowało wiele innych osób. W końcu używamy CSS na co dzień. Za pomocą arkuszy stylów jesteśmy w stanie naprawić praktycznie każdy problem z układem strony, zbudować responsywną stronę internetową, która będzie pięknie wyglądała na wszystkich urządzeniach, i ogólnie, korzystając z arkuszy stylów przez większość czasu, mamy odczucie, że dobrze wiemy, co robimy.

Wyjaśnijmy sobie pewne rzeczy już teraz. Sass nie sprawi automatycznie, że Twój kod CSS stanie się lepszy. Na przykład jeśli jeszcze dobrze nie rozumiesz CSS, to Sass i Compass nie pomogą Ci uzupełnić braków wiedzy. Jednak dzięki Sass z pewnością będziesz szybciej pisać arkusze stylów i łatwiej Ci będzie nad nimi zapanować.

### Sass, LESS czy Stylus?

Skoro czytasz tę książkę, to możliwe, że już się trochę porozglądałeś i wybrałeś Sass zamiast LESS lub Stylus. LESS i Stylus to także preprocesory CSS o podobnym zastosowaniu jak Sass. Moim skromnym zdaniem Sass jest najlepszy i ma największe możliwości, ale muszę przyznać, że dokumentacja LESS jest dla początkującego bardziej przystępna. Jeśli chciałbyś zobaczyć porównanie technologii Sass i LESS z wykazem mocnych i słabych stron obu, zajrzyj na stronę *CSS Tricks*, na której mistrz Chris Coyier zamieścił artykuł na ten temat (<http://css-tricks.com/sass-vs-less/>).

## Dlaczego należy używać właśnie Sass i Compass

Jak napisałem wcześniej, lista firm korzystających z Sass i Compass do pisania i obsługi kaskadowych arkuszy stylów ciągle się wydłuża i zawiera już takie pozycje jak BBC, eBay czy LinkedIn. Nietrudno się domyślić, że jeśli tak duże jednostki przechodzą z CSS na Sass, to musi im się to naprawdę opłacać. Istotnie tak jest! A zatem teraz przyjrzymy się kilku flagowym elementom języka Sass. Nie będzie to wyczerpujący opis możliwości tej technologii, a jedynie przedsmak tego, jak bardzo Sass i Compass mogą ułatwić Ci pracę z kaskadowymi arkuszami stylów.

### Zmienne (aby wartości definiować tylko raz)

Ile razy podczas pracy nad stroną internetową deklarujesz wartości różnych kolorów w CSS? Najczęściej używa się do tego notacji szesnastkowej typu #bfbfbf. Robisz to dziesięć razy?

Dwadzieścia? Ile by nie było, często staram się zapamiętać te szesnastkowe wartości, zwłaszcza jeśli w serwisie używam dwóch, trzech kolorów. Jednak w Sass można zdefiniować kolory jako zmienne. Zmienna to tylko odnośnik do wartości. Spójrz na poniższe trzy definicje zmiennych:

```
$red: #ff0b13;
$blue: #091fff;
$green: #11c909;
```

## Składnia zmiennych

W Sass znak dolara sygnalizuje początek definicji zmiennej. Za nim powinna znajdować się nazwa (między znakiem dolara a nazwą nie może być spacji). Później wpisuje się dwukropek sygnalizujący, że zaraz pojawi się wartość, po której będzie znajdować się średnik oznaczający zamknięcie definicji tej zmiennej. W przedstawionym przykładzie w zmiennej \$green zapisana została szesnastkowa wartość koloru zielonego. Tak zdefiniowanych zmiennych można używać w arkuszu Sass w następujący sposób:

```
.i-want-to-be-green {
  color: $green;
}
```

A to jest kod CSS, który powstanie w wyniku kompilacji powyższej konstrukcji:

```
.i-want-to-be-green {
  color: #11c909;
}
```

W odniesieniu do Sass słowo „kompilacja” oznacza konwersję z Sass na CSS.

Określenie to jest często używane w różnych poradnikach i dyskusjach dotyczących Sass, więc warto pamiętać, że oznacza one jedynie zamianę kodu Sass (znajdującego się w pliku `.scss` lub `.sass` — o czym za chwilę) na CSS. Proste, prawda?

O wiele łatwiej jest zapamiętać nazwy samodzielnie zdefiniowanych zmiennych niż nienaturalne wartości szesnastkowe. Ponadto jeśli kiedyś zechcesz zmienić te kolory, wystarczy przypisać nową wartość zmiennej, a reszta zmieni się automatycznie. Możesz pożegnać w swoim edytorze starą funkcję znajdowania i zamiany tekstu. Uff!

## Automatyczne kolory w formacie RGBA i konwersje

Coraz więcej przeglądarek obsługuje formaty zapisu kolorów **RGBA** (*Red, Green, Blue, Alpha*) i **HSLA** (*Hue, Saturation, Lightness, Alpha*). Często stosowaną sztuczką asekuracyjną, na wypadek gdyby ktoś jeszcze do przeglądania naszych stron użył przestarzałej przeglądarki, jest wpisanie najpierw wartości w formacie szesnastkowym, a następnie jej ekwiwalentu w formacie RGBA lub HSLA rozpoznawanego przez nowe przeglądarki (dzięki temu nowe aplikacje

wykorzystują formaty RGBA i HSLA, a stare — format szesnastkowy). Na przykład jeśli chcielibyśmy zdefiniować kolor o pewnym stopniu przezroczystości, moglibyśmy napisać taką regułę CSS:

```
.color-me-bad {
  color: #11c909;
  color: rgba(17, 201, 9, 0.9);
}
```

Czasami pobieranie kolorów w formatach szesnastkowym i RGBA ze złożonych obrazów (w aplikacjach typu Photoshop, Fireworks itp.) jest utrudnione. Kiedyś używałem małego programu służącego tylko do sprawdzania wartości kolorów. Jednak od kiedy używam Sass, wspomniana aplikacja stała się niepotrzebna, ponieważ mogę pisać takie deklaracje:

```
.color-me-good {
  color: $green;
  color: rgba($green, 0.9);
}
```

W kodzie tym użyłem łatwej do zapamiętania nazwy zmiennej reprezentującej kolor i funkcji kolorowej Sass w celu przekonwertowania tego koloru na format RGBA. W trzeciej linii tego kodu proszę Sass o przekazanie wartości koloru (zdefiniowanego w postaci zmiennej \$green) jako wartości RGBA z kanałem alfa o wartości 0.9. W wyniku kompilacji powstanie następujący kod CSS:

```
.color-me-good {
  color: #11c909;
  color: rgba(17, 201, 9, 0.9);
}
```

Tym, którzy do tej pory jeszcze się odpowiednio nie skoncentrowali, wyjaśniam, że Sass automatycznie dostarczył kolor w formacie RGBA. Kanał alfa został ustawiony na 90 procent. To oznacza, że w nowoczesnych przeglądarkach przez kolorową powierzchnię będzie widać 10 procent tego, co się pod nią znajduje.

## Zapomnij o przedrostkach producentów

Jestem wielbicielem technologii CSS3, ponieważ umożliwia pozbycie się wielu obrazów i ogólnie stwarza bardzo duże możliwości. Jednak korzystanie z nowych funkcji (gradientów tła, cieni elementów, przekształceń itd.), które wciąż są w fazie eksperymentalnej, wymaga stosowania przedrostków producentów i czasami zróżnicowanej składni. Wiesz, o co chodzi. W ramach przykładu spojrz na poniższą, już przestarzałą, regułę CSS zaokrąglającą rogi:

```
.rounded {
  -webkit-border-radius: 4px;
  -moz-border-radius: 4px;
  -ms-border-radius: 4px;
  -o-border-radius: 4px;
}
```

```
border-radius: 4px;
}
```

Wraz z towarzyszącym Sass systemem Compass otrzymujemy do dyspozycji mnóstwo darmowych **domieszek** (na razie nie musisz wiedzieć, czym dokładnie one są, wkrótce to wyjaśnię). Dzięki temu zamiast zapamiętywać wszystkie przedrostki i odmiany składni, wystarczy napisać taki prosty kod:

```
.rounded {
  @include border-radius(4px);
}
```

Po kompilacji tego kodu powstałby dokładnie taki sam arkusz CSS jak przedstawiony powyżej. Wszystkie przedrostki producentów zostałyby dodane automatycznie. Jest to wielka oszczędność czasu.

## Zagnieżdżanie reguł

W Sass można zagnieżdżać reguły. Jeśli na przykład chcesz utworzyć zestaw odnośników w elemencie `nav` i dodać alternatywne pseudoklasy `:hover` i `:active`, możesz napisać taki arkusz Sass:

```
nav {
  a {
    color: $red;
    &:hover {
      color: $green;
    }
    &:visited {
      color: $blue;
    }
  }
}
```

W regule elementu `nav` zagnieżdżyliśmy regułę przeznaczoną dla elementów odnośników `a`, w której z kolei zagnieżdżyliśmy reguły dotyczące formatowania stanów `:hover` i `:visited` tych odnośników. To wygląda na bardziej skomplikowane, niż jest w rzeczywistości. Po kompilacji kod ten przybierze następującą postać:

```
nav a {
  color: #ff0b13;
}
nav a:hover {
  color: #11c909;
}
nav a:visited {
  color: #091fff;
}
```

Lubię wykonywać taki zabieg na niewielkich samodzielnych modułach CSS, ponieważ pozwala mi to zachować wszystkie powiązane ze sobą style razem, zwłaszcza gdy potrzebne są pseudoklasy (na przykład `:hover` czy `:active`).

Pamiętaj, że tworzenie bardzo konkretnych selektorów CSS mało kiedy jest dobrym pomysłem. Na przykład poniżej znajduje się reguła CSS z rodzaju tych, które potrafią uprzykrzyć człowiekowi życie:

```
#container .callout-area ul#callout-one li.callout-list a.callout-link {
  color: #bfbfbf;
}
```

Selektor sięga bardzo głęboko w strukturę dokumentu, przez co jest bardzo konkretny. Z punktu widzenia łatwości obsługi kodu o wiele lepszym rozwiązaniem jest taki selektor:

```
callout-link {
  color: #bfbfbf;
}
```

Pamiętaj więc, że istnienie możliwości zagnieżdżania reguł wcale nie oznacza, że należy z niej ciągle korzystać.

## Prostsze zapytania medialne

Jeśli nie ma jakiegoś ważnego powodu, aby zrobić inaczej, moim zdaniem wszystkie strony internetowe powinny być responsywne (ekhm, możesz kupić moją książkę *Responsive Web Design. Projektowanie elastycznych witryn w HTML5 i CSS3*). W odniesieniu do CSS oznacza to konieczność napisania wielu zapytań medialnych dla różnych **punktów granicznych** w projekcie. Na przykład aby zmienić typografię, jeśli strona jest przeglądana na urządzeniu o określonej szerokości ekranu, można napisać taki kod CSS:

```
@media only screen and (min-width: 280px) and (max-width: 479px) {
  .h1 {
    font-size: 1.1em;
  }
}
@media only screen and (min-width: 480px) and (max-width: 599px) {
  .h1 {
    font-size: 1em;
  }
}
@media only screen and (min-width: 600px) and (max-width: 767px) {
  .h1 {
    font-size: 0.9em;
  }
}
```

W arkuszu tym ustawiono różne rozmiary tekstu w elementach h1 w zależności od szerokości ekranu w pikselach. Osobiście uważam, że kod ten jest zbyt obszerny i zmusza do zapamiętania wielu rzeczy.

Natomiast w Sass wystarczy napisać tylko tyle:

```
h1 {
  @include MQ(XS) {
    font-size: 1.1em;
  }
  @include MQ(S) {
    font-size: 1em;
  }
  @include MQ(M) {
    font-size: 0.9em;
  }
}
```

Zmienne dla XS, S i M są zdefiniowane gdzieś indziej (każda reprezentuje pewną szerokość). W domieszce MQ nazwy zmiennych są zastępowane ich wartościami. Za pomocą domieszki MQ można wstawić zapytania medialne, gdzie się chce. W ten sposób kod powinien stać się bardziej czytelny.

## Automatyczna kompresja kodu CSS i przyspieszenie działania strony

Opisałem już kilka elementów Sass i Compass, które powinny Cię przekonać do zainteresowania się tymi technologiami, a przecież jeszcze nawet nie wspomniałem o @extend, stylach zastępczych, plikach częściowych czy duszkach graficznych. Liczę, że przeczytasz dalsze rozdziały, w których objaśniam także te składniki. Tymczasem, jak to zwykli mawiać Columbo i Steve Jobs: „I jeszcze jedno...”.

Jak kompresujesz kod CSS przed wprowadzeniem go do użytku? Kompresja znacznie zmniejsza pliki CSS, dzięki czemu działają one szybciej we wszystkich pobierających je urządzeniach. Łatwo zredukować rozmiar plików o połowę. Oczywiście zawsze można skopiować cały arkusz stylów do jakiegoś internetowego kompresora albo skorzystać z odpowiedniej opcji w edytorze, ale Sass i tak robi to lepiej. Po prostu to robi.

W Sass można wybrać kilka formatów kompilacji kodu do postaci CSS, z których jeden to format skompresowany. Gdy tylko zapiszesz plik Sass, zostaje on automatycznie skompilowany na skompresowaną postać CSS, gotową do produkcji najmniejszą możliwą wersję. Jest to ogromna oszczędność czasu i wielka korzyść dla wszystkich użytkowników Twojej strony internetowej, nawet jeśli nie zdają sobie z tego sprawy.

To tyle. Przedstawiłem kilka zalet i możliwości stwarzanych przez tandem Sass i Compass. Teraz wyjaśnię, czym dokładnie są Sass i Compass, a potem pokażę Ci, jak zacząć z nimi pracować.



## Czym jest Sass

Na stronie internetowej Sass (<http://sass-lang.com/>) znajdujemy następujący opis tego języka:

*Sass to strukturalny metajęzyk odnoszący się do CSS, służący do tworzenia klarownych opisów stylu dokumentów oraz posiadający szersze możliwości niż czysty CSS. Sass ma prostszą i elegantszą składnię niż CSS oraz zawiera wiele funkcji przydatnych przy tworzeniu i obsłudze arkuszy stylów.*

Ponadto na stronie internetowej Sass można znaleźć informacje na temat pochodzenia tego języka. Jego twórcą jest Hampton Catlin (ten sam, który stworzył też HAML). Od momentu powstania w 2006 roku Sass był promowany, uwielbiany i obsługiwany przez wiele innych osób. Jednak największe zasługi dla jego rozwoju mają Nathan Weizenbaum (projektant i programista, który współpracował z Hamptonem Catlinem do wersji 2) i Christopher Eppstein (który dołączył do zespołu w 2008 roku i był w nim obecny od wersji 2.2; ponadto Eppstein jest twórcą systemu Compass). Przy rozwijaniu projektu pomagało także wielu wolontariuszy. Strona projektu w serwisie GitHub znajduje się pod adresem <http://github.com/nex3/sass>.

Jako że początki Sass ściśle się wiążą ze społecznością skupioną wokół języka Ruby (Ruby jest językiem programowania), jego dokumentacja została napisana językiem zrozumiałym dla programistów. Z tego powodu była ona trudna do zrozumienia dla osób niezwiązanych z programowaniem jako takim. To wielka szkoda, ponieważ projektanci samodzielnie piszący kod dla swoich produktów skorzystaliby z udogodnień technologii Sass tak samo jak wszyscy inni.

Język Sass ma dwa rodzaje składni. Pierwotna (zwana Sass i wykorzystująca pliki z rozszerzeniem `.sass`) jest zwięzła i bazuje na wcięciach. Brak w niej wszechobecnych w CSS klamer. Jej szczegółowy opis można znaleźć na stronie [http://sass-lang.com/docs/yardoc/file.INDENTED\\_SYNTAX.html](http://sass-lang.com/docs/yardoc/file.INDENTED_SYNTAX.html).

Natomiast w tej książce opisuję składnię SCSS, której kod zapisuje się w plikach z rozszerzeniem `.scss`. Składnia ta jest bardziej obszerna niż jej poprzedniczka, ale też bardziej przypomina normalne arkusze stylów.

Wynn Netherland, Nathan Weizenbaum i Christopher Eppstein napisali własną książkę o Sass i Compass — *Sass and Compass in Action*. Jako że Nathan Weizenbaum i Christopher Eppstein nadal są zarządcami projektu Sass, to książka ta, mimo że nie jest przeznaczona wprost dla projektantów, może być godna uwagi. Zajrzyj na stronę <http://manning.com/netherland/>.

Języka Sass można używać za darmo, bez posiadania jakiegokolwiek licencji.

## Czym jest Compass

Na stronie Compass pod adresem <http://compass-style.org> znajduje się następująca informacja:

*Compass to otwarte narzędzie do tworzenia kaskadowych arkuszy stylów.*

W istocie Compass był pierwszym narzędziem obsługującym Sass. Instalując go wraz z Sass, zaopatrujemy się w mnóstwo przydatnych wzorców i dodatków ułatwiających tworzenie kodu CSS. Widziałeś telewizyjny program „Odpicuj mi brykę”? Gdyby Xzibit zabrał Sass do West Coast Customs, to Compass byłby pierwszym zamontowanym dodatkiem!

Innymi słowy, Compass umożliwia korzystanie z nowych właściwości CSS3, takich jak cienie elementów, gradienty, kolumny i przekształcenia, za pomocą pojedynczych poleceń, które następnie magicznie zamienia na działający bezproblemowo, zgodny ze wszystkimi przeglądarkami kod CSS. Ponadto pozwala na stosowanie dodatkowych wtyczek umożliwiających tworzenie niesamowitych lekkich systemów siatkowych, o których będzie jeszcze mowa w dalszej części książki.

Strona projektu Compass w serwisie GitHub znajduje się pod adresem <https://github.com/chrisepstein/compass>.

Compass to program typu *charityware*, co oznacza, że można z niego korzystać do woli, ale jego właściciele zachęcają do wpłacania darowizn na rzecz fundacji UMDF pomagającej w poszukiwaniu lekarstwa na choroby mitochondrialne. Gdy zatem zainstalujesz Compass i zorientujesz się, ile czasu dzięki niemu oszczędzasz, rozważ możliwość wpłaty dowolnej darowizny (potrzebne informacje znajdziesz pod adresem <http://umdf.org/compass>).

## Instalacja Sass i Compass

W czasach słusznie minionych Sass i Compass można było zainstalować tylko przy użyciu wiersza poleceń. Na szczęście jednak czasy się zmieniły. Jeśli nie jesteś fanem wiersza poleceń, to nie musisz z niego korzystać. Istnieje kilka graficznych narzędzi dla systemów OS X, Linux i Windows, które także zawierają wszystkie pliki potrzebne do kompilowania plików Sass na CSS podczas ich zapisywania. Opiszę je nieco dalej, ale najpierw pokażę, jak zainstalować Sass i Compass za pomocą wiersza poleceń. Jesteś gotów? Do dzieła.

Rozumiem, jeśli Ciebie to nie bawi. W takim razie możesz przejść od razu do sekcji poświęconej narzędziom graficznym. Zawsze możesz tu wrócić kiedy indziej. Nikogo nie oceniam!

## Instalacja pakietu dla systemu OS X

Użytkownicy systemu Mac OS X mają ułatwione zadanie, ponieważ twórca programu Compass Chris Eppstein stworzył graficzny pakiet instalacyjny. Wystarczy pobrać ten pakiet ze strony <https://github.com/chrisepstein/compass/downloads> i uruchomić go na swoim komputerze.

Jednak warto też wiedzieć co nieco o sposobie działania wiersza poleceń, więc możesz też spróbować podnieść trochę swoje kwalifikacje i skorzystać z tego narzędzia.

**Praca w wierszu poleceń**

Aby pracować z Sass i Compass w wierszu poleceń, należy znać tylko kilka poleceń. Oto lista tych, które prawie na pewno Ci się przydadzą:

Wyświetlanie listy plików w bieżącym folderze:

Windows:

```
dir
```

Mac:

```
ls
```

Zmiana folderu — przejście do innego folderu znajdującego się w bieżącym:

Windows i Mac:

```
cd nazwa-folderu
```

W miejsce napisu nazwa-folderu należy wpisać właściwą nazwę folderu.

Aby przejść do folderu nadrzędnego:

Windows:

```
cd..
```

Mac

```
cd ..
```

## Instalacja Sass i Compass i praca z nimi w wierszu poleceń

Do instalacji Sass i Compass potrzebny jest interpreter języka Ruby. Jeśli używasz systemu OS X, to interpreter ten jest już gotowy do użycia.

### Instalacja interpretera Ruby w systemie Windows

Jeśli korzystasz z systemu Windows, wejdź na stronę <http://rubyinstaller.org/downloads/> oraz pobierz najnowszy plik instalacyjny interpretera języka Ruby (odnośnik do tego pliku będzie wyglądał mniej więcej tak: <http://rubyforge.org/frs/download.php/76054/rubyinstaller-1.9.3-p194.exe>, choć może zawierać inny numer wersji). Podczas instalacji pozostaw ustawienia domyślne (nie musisz zaznaczać żadnych pól wyboru w opcjach). W systemie Linux interpreter Ruby powinno dać się zainstalować bezpośrednio za pomocą menedżera pakietów.

Zakładam, że jeśli ktoś używa systemu Linux, to dość sprawnie posługuje się wierszem poleceń. Jeśli nie, to zalecam przesiąść się na Maca.

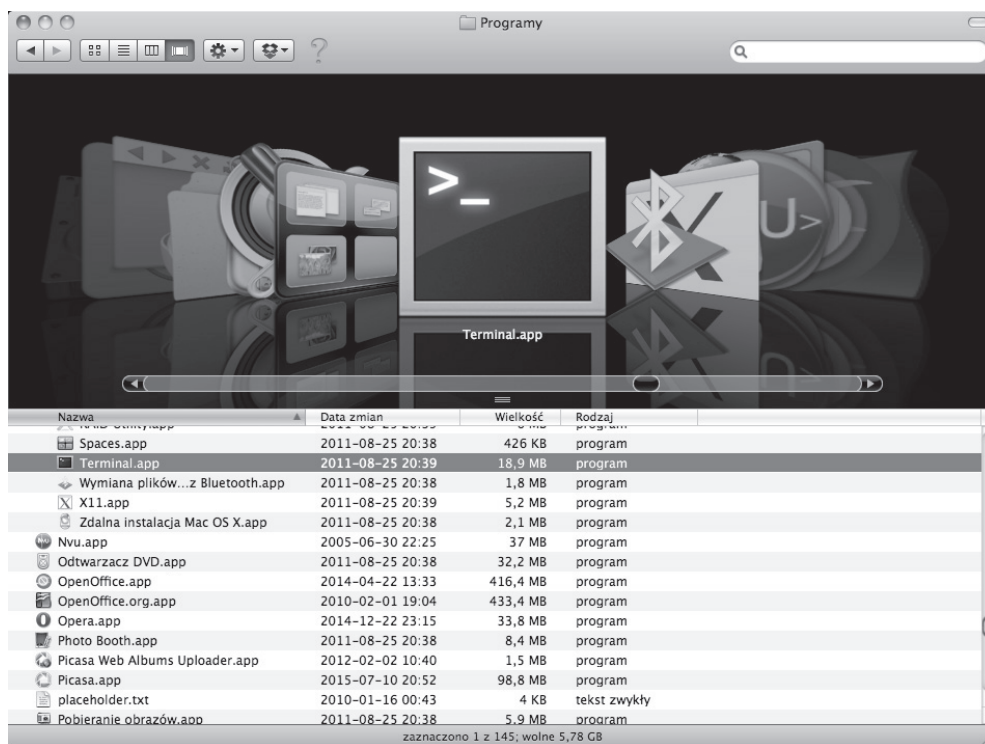
## Polecenie gem

Niezależnie od tego, jakiego systemu operacyjnego używasz (Windows, Linux lub Mac), masz już w nim przygotowany do działania interpreter języka Ruby. Teraz skorzystamy z polecenia **gem**, którego wykonanie będzie równoznaczne z wydaniem komputerowi następującego polecenia: „Ruby, zainstaluj mi *gem* o nazwie *compass*”. Jako że Compass wymaga Sass, instalacja tego pierwszego pociąga za sobą automatyczną instalację drugiego.

Traktuj polecenie gem jak niewielki program albo wtyczkę. Jest to po prostu rozszerzenie funkcjonalności narzędzia korzystającego z interpretera Ruby. Jako że Sass i Compass wykorzystują Ruby, gdy zaczniesz na dobre korzystać z Sass, często będziesz pobierać nowe „klejnoty” (*gem* oznacza „klejnot”). Istnieją na przykład *gemy* dla systemów siatkowych, jak Susy, *gemy* dla stylów przycisków, jak Sassy Buttons, i wiele więcej.

## Instalacja z wiersza poleceń w systemie Mac OS X

W systemie OS X należy uruchomić aplikację o nazwie Terminal, którą zazwyczaj można znaleźć w folderze *Aplikacje/Narzędzia*. Może to wyglądać tak:



Kliknij dwukrotnie ten plik, aby wyświetlić okno widoczne poniżej:



Jeśli nigdy nie korzystałeś z tego narzędzia, to podpowiem, że jest to właśnie wiersz poleceń! Teraz tylko wpisz poniższe polecenie i naciśnij klawisz *Enter*:

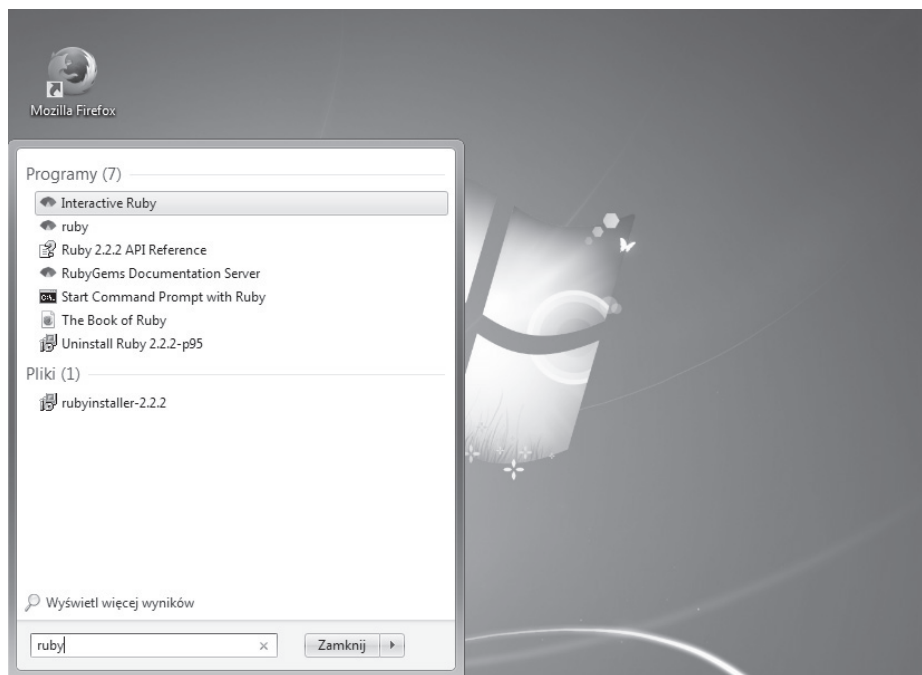
```
sudo gem install compass
```

Zostaniesz poproszony o podanie hasła (należy wpisać to samo, którego używasz do logowania do pulpitu). Wpisz je i naciśnij klawisz *Enter*. Podczas wpisywania hasła nie przestrasz się, że nic się nie dzieje. Terminal nie reaguje w widoczny sposób, gdy wpisuje się hasło. Jednak po naciśnięciu klawisza *Enter* Sass i Compass zostaną zainstalowane.

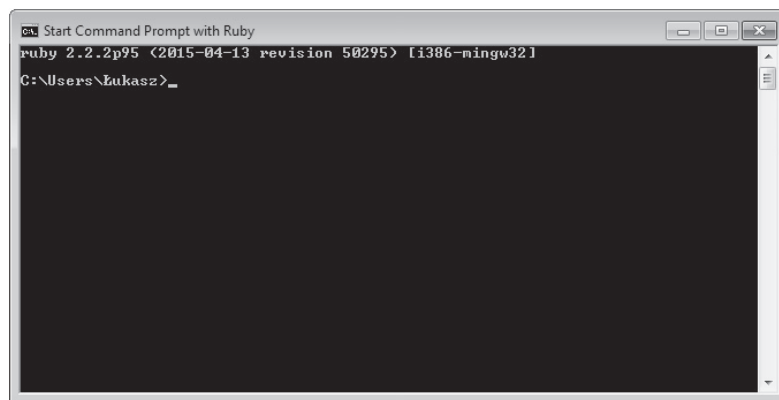
I jak się czujesz? Właśnie zainstalowałeś *gem* Ruby. Czujesz się jak maniak komputerowy? Nie przejmuj się. To będzie nasza mała tajemnica.

## Instalacja z wiersza poleceń w systemie Windows

W systemie Windows Vista i 7 należy kliknąć przycisk z logo Microsoftu, wpisać słowo *ruby* w polu wyszukiwania i kliknąć na liście pozycję *Start Command Prompt with Ruby* (uruchom wiersz poleceń z obsługą języka Ruby). W systemie Windows 8 należy kliknąć prawym przyciskiem myszy na ekranie startowym, kliknąć opcję *Wszystkie aplikacje*, a następnie kliknąć opcję *Start Command Prompt with Ruby*.



Zostanie wyświetlone okno dialogowe wiersza poleceń podobne do pokazanego poniżej:



Wpisz poniższe polecenie i naciśnij klawisz *Enter*:

```
gem install compass
```

Spowoduje to zainstalowanie Compass i Sass. Po instalacji pojawi się nowy wiersz poleceń z migającym kursorem.

To wszystko. Sass i Compass są zainstalowane i możesz zacząć tworzyć pliki.

## Sprawdzanie wersji Compass i Sass

Jako że od czasu do czasu Sass i Compass są wzbogacane o nowe funkcje, dobrze jest wiedzieć, jak się sprawdza bieżącą i instaluje najnowszą wersję tych narzędzi. Do sprawdzania wersji w wierszu poleceń służy poniższe polecenie:

```
sass -v
```

W odpowiedzi otrzymasz napis w stylu Sass 3.4.15 (Selective Steve).

Aby sprawdzić wersję narzędzia Compass, należy wykonać poniższe polecenie:

```
compass -v
```

W odpowiedzi otrzymasz napis w stylu Compass 1.0.3 (Polaris).

Jeśli Cię to interesuje, Polaris to Gwiazda Polarna (tak, sprawdziłem to w Google).

## Sprawdzanie, które wersje Sass i Compass są dostępne

Oczywiście zawsze można wejść na strony tych narzędzi i sprawdzić na nich, jakie wersje są dostępne, ale łatwiej i szybciej jest skorzystać w tym celu z wiersza poleceń. Wystarczy wykonać poniższe polecenie:

```
gem list sass -a -r
```

W ten sposób prosimy interpreter języka Ruby o podanie listy wersji wszystkich *gemów* ze słowem sass w nazwie. Część `-r` oznacza, że prosimy o zasięgnięcie informacji zdalnych (na przykład z internetu, a nie z systemu operacyjnego), natomiast parametr `-a` nakazuje wyświetlić listę wszystkich wersji. Poniżej znajduje się to samo polecenie w wersji dla Compass:

```
gem list compass -a -r
```

Każde z tych poleceń spowoduje zwrócenie listy *gemów* zawierających w nazwie słowo sass lub compass z podaniem numeru wersji w nawiasie. Na przykład:

```
sass (3.4.15, 3.4.14, 3.4.13, 3.4.12, 3.4.11, 3.4.10, 3.4.9, 3.4.8, 3.4.7, 3.4.6,
3.4.5, 3.4.4, 3.4.3, 3.4.2, 3.4.1, 3.4.0, 3.3.14, 3.3.13, 3.3.12, 3.3.11, 3.3.10,
3.3.9, 3.3.8, 3.3.7, 3.3.6, 3.3.5, 3.3.4, 3.3.3, 3.3.2, 3.3.1, 3.3.0, 3.2.19,
3.2.18, 3.2.17, 3.2.16, 3.2.15, 3.2.14, 3.2.13, 3.2.12, 3.2.11, 3.2.10, 3.2.9,
3.2.8, 3.2.7, 3.2.6, 3.2.5, 3.2.4, 3.2.3, 3.2.2, 3.2.1, 3.2.0, 3.1.21, 3.1.20,
3.1.19, 3.1.18, 3.1.17, 3.1.16, 3.1.15, 3.1.14, 3.1.13, 3.1.12, 3.1.11, 3.1.10,
3.1.9, 3.1.8, 3.1.7, 3.1.6, 3.1.5, 3.1.4, 3.1.3, 3.1.2, 3.1.1, 3.1.0)
```

Wersje są wymienione w odwrotnej kolejności chronologicznej (od najnowszej).

Aby sprawdzić, jakie wersje Sass są przygotowywane do opublikowania, należy użyć poniższego polecenia:

```
gem list sass --pre -r
```

Aby sprawdzić, jakie wersje narzędzia Compass są przygotowywane do opublikowania, należy użyć poniższego polecenia:

```
gem list compass --pre -r
```

## Instalowanie najnowszej wersji Sass i Compass (włącznie z wydaniem wstępnym)

Aby zainstalować najnowszą stabilną wersję Sass, wystarczy wykonać w wierszu poleceń poniższe polecenie (słowo `sudo` w niektórych systemach jest niepotrzebne):

```
sudo gem install sass
```

Jeśli przyszła wersja ma jakieś niezwykle przydatne składniki, które koniecznie chcesz wypróbować, ale jeszcze oficjalnie nie została wydana, za pomocą poniższego polecenia możesz zainstalować najnowszą wersję wstępną:

```
sudo gem install sass --pre
```

Miej jednak świadomość, że w takiej wersji mogą znajdować się błędy i różne usterki. Dlatego jeżeli nie masz bardzo ważnego powodu, by zrobić inaczej, zalecam używanie wyłącznie stabilnych wersji oprogramowania.

Dla Compass polecenia wyglądają bardzo podobnie:

```
sudo gem install compass
```

I polecenie dla wersji wstępnej:

```
sudo gem install compass -pre
```

### Jak odinstalować wybraną wersję Sass

Jeśli coś zacznie iść nie po Twojej myśli, zawsze możesz cofnąć się do starszej wersji Sass za pomocą poniższego polecenia:

```
gem uninstall sass --version numerwersji
```

W miejsce członu `numerwersji` należy wpisać żądany numer wersji (na przykład `3.2.0.alpha.103`).

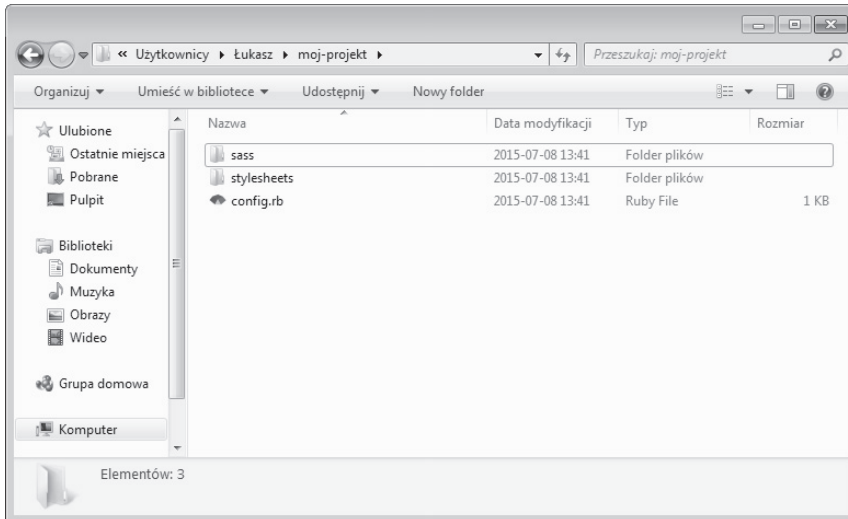


## Tworzenie projektu Sass i Compass z poziomu wiersza poleceń

Chcesz utworzyć projekt Sass i Compass z poziomu wiersza poleceń? W tym celu musisz przejść do folderu witryny (ja na przykład swoje witryny przechowuję w folderze o nazwie *Sites*) i wykonać poniższe polecenie:

```
compass create moj-projekt
```

Użyliśmy wbudowanego polecenia narzędzia Compass o nazwie `create` do utworzenia projektu w folderze o nazwie *moj-projekt*. Gdy naciśniesz klawisz `Enter`, Compass utworzy kilka plików i folderów. Poniżej pokazuję przykładową strukturę projektu utworzonego w systemie Windows:



### Do czego służą wygenerowane pliki w projekcie Compass

Szczegółowy opis generowanych plików i folderów znajduje się w rozdziale 2. Na razie przedstawiam tylko krótki przegląd ich funkcji.

Folder *.sass-cache* (utworzony o jeden poziom wyżej): folder zawierający pliki podręczne wykorzystywane przez Sass w celu przyspieszenia budowy plików CSS. Nie trzeba z nim nic robić.

Folder *sass*: katalog na pliki Sass, w których będziemy pracować. Nazwa *sass* jest domyślna i można ją zmienić na dowolną inną.

Folder *stylesheets*: katalog na skompilowane pliki CSS wygenerowane przez Sass. Nazwa *stylesheets* jest domyślna w projektach Compass i można ją zmienić na dowolną inną.

Plik *config.rb*: plik zawierający domyślne ustawienia konfiguracyjne projektu, takie jak domyślne nazwy i lokalizacje folderów. Ponadto w pliku tym ustawia się metodę kompresji generowanych plików CSS.

## Automatyczne kompilowanie kodu na CSS z poziomu wiersza poleceń

Aby Compass „obserwował” zmiany w plikach Sass (znajdujących się w folderze *sass* projektu) i automatycznie kompilował je na CSS po każdym zapisaniu pliku, najpierw należy przejść do folderu zawierającego interesujące nas pliki Sass (aby odświeżyć sobie pamięć na temat potrzebnych poleceń, możesz jeszcze raz przeczytać ramkę „Praca w wierszu poleceń” w podrozdziale „Instalacja Sass i Compass”). Będąc w katalogu głównym projektu (lista zawartości powinna zawierać plik *config.rb*), wykonaj poniższe polecenie:

```
compass watch
```

Dla narzędzia Compass oznacza ono: „Obserwuj ten projekt i jeśli zmieni się coś w którymkolwiek pliku Sass, skompiluj go do postaci CSS”. Od tej pory, jeśli zmienisz cokolwiek w plikach Sass i zapiszesz tę zmianę, Compass automatycznie wykryje to zdarzenie i dokona ponownej kompilacji odpowiednich plików CSS (kompilowane pliki CSS domyślnie są zapisywane w folderze *stylesheets* projektu).

## Graficzne narzędzia do pracy z Sass i Compass

Istnieje kilka graficznych narzędzi, które uwalniają nas od konieczności zajmowania się całym tym gemowym bałaganem języka Ruby (wszystkie zawierają własną wersję interpretera) i kompilują pliki Sass na CSS. W podrozdziale tym przedstawiam trzy z nich: LiveReload (OS X i Windows), CodeKit (OS X) i Scout (OS X i Windows).

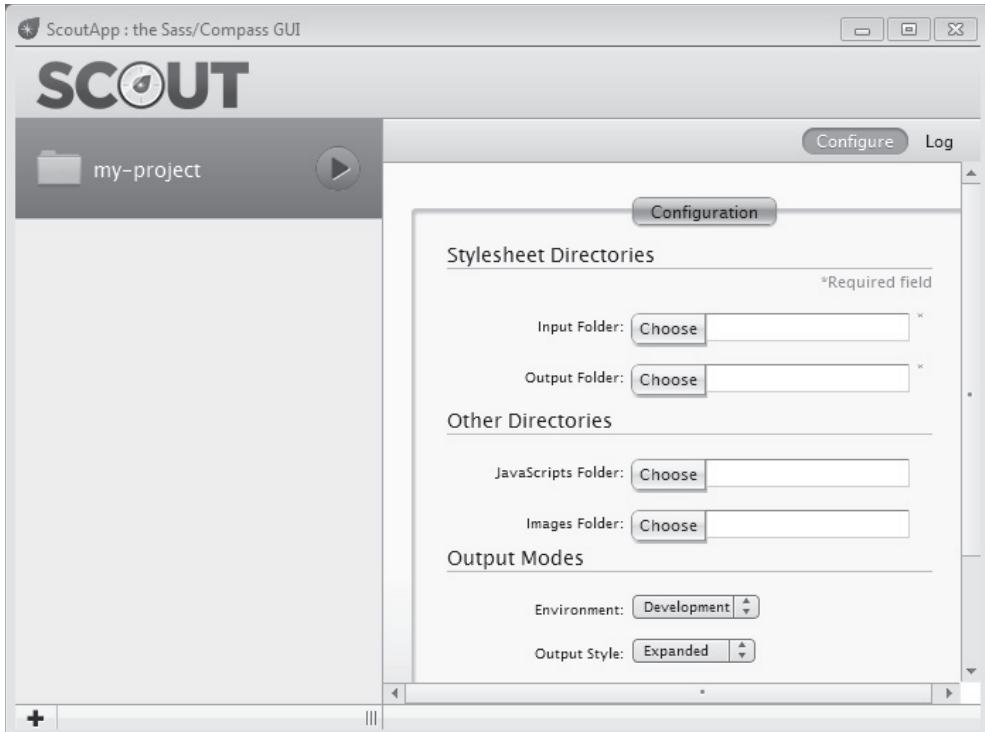
Początkowo z Sass i Compass pracowałem przy użyciu wiersza poleceń, ale teraz, jeśli nie muszę wykonać jakiegoś konkretnego polecenia, na co dzień korzystam z LiveReload lub CodeKit. Podkreślam jednak, że nie ma jednego słusznego narzędzia. Wybierz po prostu to narzędzie, które najbardziej Ci się podoba.

Bez względu na to, co wybierzesz, wystarczy kilka kliknięć, aby rozpocząć automatyczną obserwację wybranego katalogu. Pamiętaj tylko, aby uruchomić program, i możesz już pracować z plikami Sass.

### Scout

**Scout** to darmowe narzędzie zbudowane przy użyciu technologii Adobe Air. Jest dostępne do pobrania na stronie <http://mhs.github.com/scout-app>. Po zainstalowaniu uruchom program i kliknij znajdujący się w lewym dolnym rogu okna przycisk ze znakiem plus. Potem musisz skon-

figurować ustawienia, ale wystarczy, że zrobisz to raz i później pliki będą kompilowane automatycznie. Ten program jest darmowy, ale osobiście uważam, że warto wydać parę złotych na jedno z pozostałych rozwiązań, które zapewniają dodatkowe funkcje.

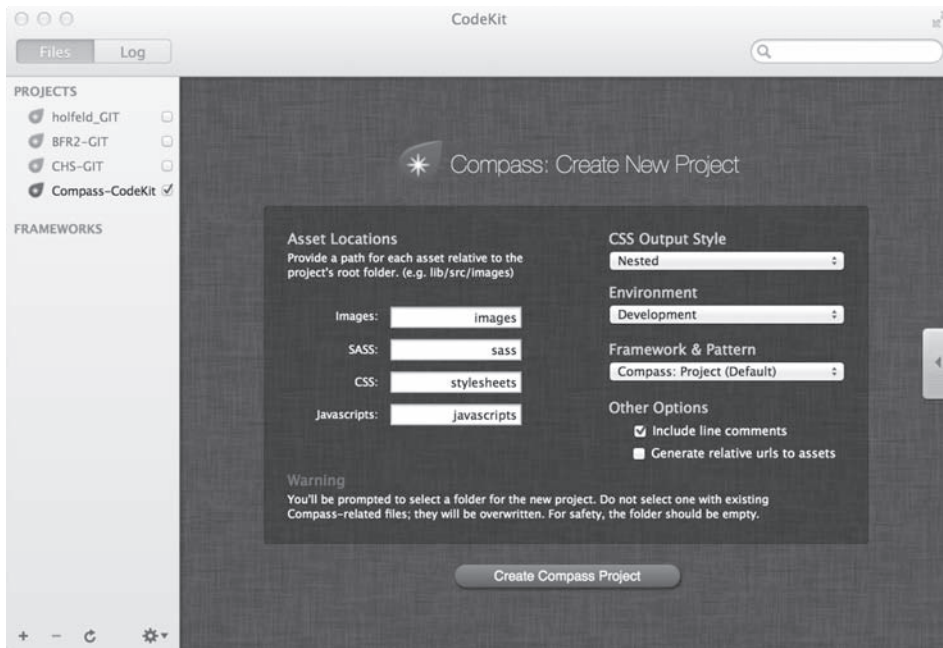


## CodeKit

Program **CodeKit** jest dostępny tylko dla użytkowników systemu Mac OS X (przykro mi, wielbiciele Windowsa i Linuksa). Ma świetny prosty interfejs użytkownika i nie tylko kompiluje pliki Sass (czy też pliki w kilku innych tego typu językach), lecz także dodatkowo odświeża okno przeglądarki. To pozwala zaoszczędzić naprawdę dużo czasu w pracy. Ponadto aplikacja ta umożliwia łączenie plików i ma wiele innych funkcji.

Jedną z największych zalet programu CodeKit jest to, że można w nim tworzyć projekty Compass wprost z interfejsu. Wystarczy określić ustawienia, wybrać folder i to wystarczy, aby narzędzie automatycznie obserwowało zmiany.

Pełna wersja aplikacji kosztuje 25 dolarów, ale jest też dostępna darmowa wersja próbna z ograniczeniem czasowym, więc możesz najpierw sprawdzić, czy aplikacja Ci się podoba (<http://incident57.com/codekit/>).



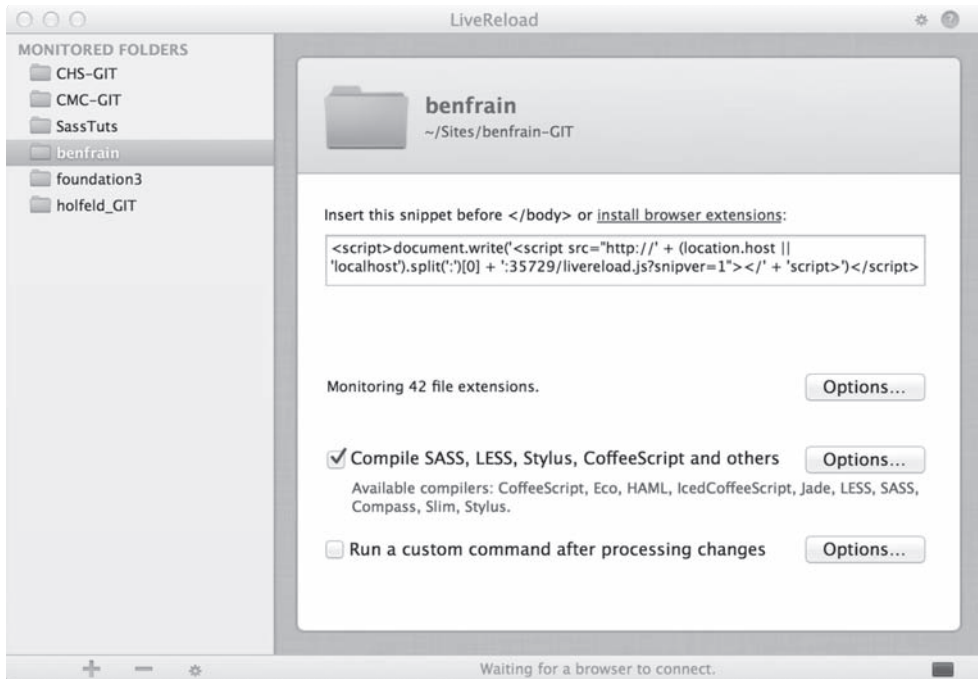
## LiveReload

Program **LiveReload** podobnie jak CodeKit nie tylko kompiluje kod w językach typu Sass, lecz także odświeża przeglądarkę, aby ukazać dokonane zmiany. Wystarczy nacisnąć znajdującą się na dole ikonę plusa, znaleźć potrzebny folderu z projektem i zaznaczyć opcję *Compile SASS, LESS, Stylus, CoffeeScript and others* (kompiluj kod SASS, LESS, Stylus, CoffeeScript i inne).

Program kosztuje 9,99 dolara (dla użytkowników systemu OS X dostępny jest w App Store). Oprócz wersji dla systemu OS X dostępna jest też wersja dla systemu Windows (<http://livereload.com>).

### Utwórz pierwszy plik Sass w mniej niż dziesięć sekund

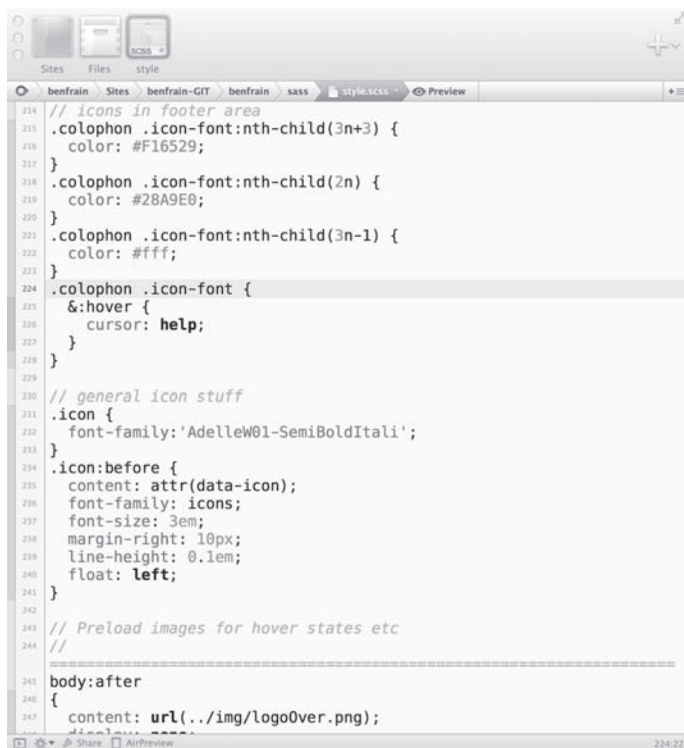
Pamiętaj, że każdy plik CSS można bez problemu zamienić w plik Sass. W tym celu wystarczy zmienić rozszerzenie `.css` na `.scss`. Po takiej zmianie otrzymasz w pełni funkcjonalny i całkowicie prawidłowy plik Sass.



## Praca z plikami Sass w edytorach tekstu

Edytory tekstu są jak drużyny sportowe — każdy ma swój ulubiony. Nie istnieje nic takiego jak edytor tekstu, który jest odpowiedni lub nieodpowiedni do pracy z Sass. Choć oczywiście obsługa składni tego języka bardzo pomaga, ponieważ kolorowanie różnych konstrukcji ułatwia orientowanie się w kodzie źródłowym.

Na szczęście większość wysokiej jakości edytorów tekstu standardowo obsługuje składnię Sass lub ma zewnętrzne wtyczki zapewniające taką obsługę. Aktualnie moim ulubionym edytorem tekstu jest Sublime Text (wieloplatformowy i dostępny pod adresem <http://sublimetext.com>), chociaż nie gorsze są Coda 2 (tylko dla systemu OS X — <http://panic.com/coda>), Espresso 2 (tylko dla systemu OS X — <http://macrabbbit.com>) i Aptana (wieloplatformowy — <http://aptana.com>).



```

214 // icons in footer area
215 .colophon .icon-font:nth-child(3n+3) {
216   color: #F16529;
217 }
218 .colophon .icon-font:nth-child(2n) {
219   color: #28A9E0;
220 }
221 .colophon .icon-font:nth-child(3n-1) {
222   color: #fff;
223 }
224 .colophon .icon-font {
225   &:hover {
226     cursor: help;
227   }
228 }
229
230 // general icon stuff
231 .icon {
232   font-family: 'AdelleW01-SemiBoldItali';
233 }
234 .icon:before {
235   content: attr(data-icon);
236   font-family: icons;
237   font-size: 3em;
238   margin-right: 10px;
239   line-height: 0.1em;
240   float: left;
241 }
242
243 // Preload images for hover states etc
244 //
245 =====
246 body:after
247 {
248   content: url(../img/logoOver.png);

```

## Podsumowanie

W rozdziale tym przedstawiłem kilka powodów, dla których warto korzystać z narzędzi Sass i Compass. Ponadto dokładnie opisałem, czym każde z nich jest, do czego służy i jaki ma związek z drugim.

Później pokazałem, jak zainstalować Sass i Compass z poziomu wiersza poleceń (przy okazji wyjaśniając, do czego służą *gemy* Ruby) oraz przedstawiłem kilka graficznych narzędzi obsługujących Sass i Compass. Niezależnie od tego, które rozwiązanie wybrałeś, teraz masz już w swoim komputerze odpowiednie oprogramowanie do obserwowania zmian w plikach Sass i automatycznego ich kompilowania na CSS po każdym zapisaniu.

Uporaliśmy się więc z największym problemem utrudniającym rozpoczęcie pracy z Sass i Compass. Skoro tak, to w następnym rozdziale możemy utworzyć pierwszy projekt. Dowiesz się, jak zmieniać ustawienia w pliku konfiguracyjnym, jak tworzyć komentarze w plikach Sass, jak posługiwać się plikami częściowymi oraz jak upraszczać budowę arkuszy stylów przez zastosowanie w nich zmiennych.

# Skorowidz

## A

adres URI, 187  
aplikacja Terminal, 32  
automatyczna kompresja kodu, 28  
automatyczne  
  kolory, 24  
  kompilowanie kodu, 38

## B

biblioteka Modernizr, 73  
błędy, 229

## C

Chrome  
  Narzędzia dla programistów,  
  232, 235  
ciągle przerysowywanie strony,  
  233  
cienie, 173  
cień tekstu, 169  
clearfix, 98  
Compass, 23, 29  
CSS Lint, 231  
CSS3, 25, 167  
  domieszki Compass, 168  
CSScomb, 232

## D

definiowanie  
  koloru, 94  
  kontekstu, 125  
  punktów brzegowych, 153

  wartości eksperymentalnych,  
  223  
  wielu cieni, 173  
  zmiennych, 57  
desaturacja, 108  
dodawanie, 198  
  jednostek, 209, 210  
  obrazów do tła, 178  
  stylów Modernizr, 73  
  wtyczek, 49  
domieszka, 26, 63, 85  
  box-shadow, 172  
  hide-text, 184  
  isolate-grid, 146  
  MQ, 154  
  nth-omega, 142  
  pad, 139  
  post, 140  
  pre, 140  
  prefix, 136  
  pull, 141  
  push, 141  
  squish, 141  
  squish-text, 185  
  suffix, 138  
domieszki  
  do zamiany tekstu, 184  
  składnia podstawowa, 87  
  ustawienia domyślne, 88  
  w domieszkach, 102  
dopełnienie wokół obrazów, 183  
duszki graficzne, 180  
dyrektywa  
  @debug, 213  
  @else if, 201  
  @extend, 65, 82, 158

  @if, 201  
  @import, 125  
  @return, 212  
  @warn, 215  
dyrektywy sterujące, 201  
dzielenie, 199

## E

edytor tekstu, 41  
elastyczne siatki, 119  
element testimonial, 80  
elementy wizualne, 65

## F

filtry CSS, 193  
folder  
  .sass-cache, 46, 49  
  images, 46  
  partials, 55  
  sass, 47  
font, 75  
format  
  HSLA, 24  
  PNG, 190  
  RGBA, 24  
  SVG, 190  
Frain Ben, 13  
funkcja, 22  
  adjust-color, 113  
  adjust-hue, 107  
  complement, 105  
  darken, 95  
  desaturate, 108  
  fade-in, 110  
  fade-out, 110

- funkcja  
 grayscale, 110  
 image-url, 177, 178  
 inline-image, 188  
 invert, 106  
 lighten, 95  
 mix, 112  
 opacity, 110  
 rgba, 111  
 saturate, 108  
 scale-color, 114  
 shade, 115  
 tint, 115  
 transparentize, 109
- G**
- generowanie kodu CSS, 91  
 gradient  
 liniowy, 175  
 promienisty, 177  
 w tle, 175
- H**
- HSL, 95  
 HSLA, 24  
 HTTP2, 54
- I**
- importowanie  
 plików partiami, 224  
 pliku częściowego, 56  
 instalowanie  
 gemu oily\_png, 181  
 Sass i Compass, 11  
 w systemie Mac OS X,  
 30, 32  
 w systemie Windows, 33  
 z wiersza poleceń, 31  
 Susy, 123  
 instrukcja  
 from through, 205  
 from to, 205  
 if-else, 155  
 interaktywna powłoka Sass, 223  
 interaktywny Compass, 224  
 interfejs wiersza poleceń, 223
- interpolacja, 205  
 interpreter Ruby, 31  
 inwersja, 106
- J**
- jednostka, 209  
 em, 98  
 jQuery, 133
- K**
- klasa  
 .blockquote--small, 81  
 .no-fontface, 76  
 class-within, 78  
 nesting, 78  
 kod CSS, 23  
 kolory, 93  
 automatyczne, 24  
 definiowanie wartości, 94  
 HSL, 95  
 HSLA, 24  
 przyciemnianie, 95  
 RGBA, 24  
 rozjaśnianie, 95  
 składnia rozjaśniania, 96  
 kolumny, 171  
 kombinacje  
 funkcji, 117  
 obrazów i gradientów, 177  
 komentarze, 52  
 głośne, 53  
 jednowierszowe Sass, 58  
 standardowe CSS, 58  
 kompilacja kodu Sass, 229  
 kompresja  
 arkuszy stylów, 52, 53  
 GZIP, 161  
 kodu CSS, 28  
 konfiguracja duszków, 183  
 kontekst dla siatki, 125  
 kontener, 144  
 konwertowanie plików  
 częściowych, 226
- L**
- LESS, 23
- Ł**
- łańcuchy, 72  
 łączenie selektorów w łańcuchy,  
 72
- M**
- mnożenie, 198  
 model blokowy, 128  
 moduł układów kolumnowych,  
 171  
 modyfikacja znaczników, 97
- N**
- nagłówek, header, 97  
 narzędzia  
 automatyzacyjne, 44  
 dla programistów, 232, 235  
 graficzne, 38  
 narzędzie lint, 231  
 nawiasy, 103  
 nazwy  
 folderów, 50  
 zmiennych, 94  
 normalizacja, 55
- O**
- obraz śródliniowy, 188  
 obrazy, 177, 179  
 obsługa  
 przeglądark, 218  
 SVG, 190  
 własności CSS, 221  
 obszar widoku, 128  
 odejmowanie, 198  
 opcja  
 kompaktowa, 52  
 kompresji, 52  
 zagnieżdżania, 51  
 opcje  
 konfiguracyjne duszków, 183  
 układu, 184  
 operatory  
 nierówności, 211  
 relacyjne, 212  
 równości, 211



**P**

parametr --bare, 47  
 pętla  
   @each, 206  
   @for, 203  
 pisanie funkcji, 210  
 plik  
   \_base.scss, 66  
   \_fonts.scss, 75  
   \_footer.scss, 55  
   \_header.scss, 55  
   \_layout.scss, 67, 97, 125, 136  
   \_mixins.scss, 86, 100  
   \_modules.scss, 67, 94, 100, 102  
   \_normalize.scss, 55, 57, 65  
   \_placeholders.scss, 99, 168  
   \_variables.scss, 57, 69, 94, 155  
 config.rb, 38, 47–50, 53, 56  
 footer.css, 54  
 header.css, 54  
 ie.css, 46  
 index.html, 44, 47, 59  
 main.js, 44  
 plugins.js, 44  
 print.css, 46  
 styles.css, 44, 46  
 styles.scss, 56, 67, 99  
 pliki  
   .css, 21  
   .scss, 21  
   częściowe, 54  
     importowanie, 56  
     dla fontów, 75  
 podział kodu źródłowego, 66  
 pogromca bufora Compass, 179  
 polecenie  
   Compass create, 46  
   gem, 32  
 pomoc diagnostyczna, 67  
 pozycjonowanie względem kontenera, 144  
 preprocesory CSS, 22  
 program  
   Aptana, 41  
   Coda 2, 41  
   CodeKit, 39, 45  
   Espresso 2, 41

LiveReload, 40  
 Scout, 38  
 Sublime Text, 41  
 przeglądarki internetowe, 218  
 przejścia, 195  
 przekształcenia CSS, 190  
 przerysowywanie strony, 233  
 przestrzenie nazw, 79  
 przezroczystość, 109  
 przyciemnianie kolorów, 96  
 pseudoklasa last-child, 71  
 pseudoselektory, 72  
 punkt  
   brzegowy, breakpoint, 128, 149, 153  
   graniczny, 27

**R**

reguła  
   @font-face, 76  
   kolumnowa, 172  
 reset, 55  
 responsywne siatki, 119, 128  
 RGBA, 24  
 rozjaśnianie kolorów, 96  
 rozmiar  
   czcionki, 160  
   pliku CSS, 91  
 rozszerzanie, 63  
 stylów, 84

**S**

samodzielne arkusze stylów, 226  
 Sass, 21, 23, 29  
   interaktywna powłoka, 223  
   kasowanie zawartości bufora, 228  
   logika programistyczna, 197  
   metody kompilacji, 229  
   obliczenia matematyczne, 198  
   pisanie funkcji, 210  
   zapytania medialne, 149, 150  
 saturacja, 108  
 selektor  
   nth-child, 106  
   rodzica, 72, 73  
   zastępczy, 84

selektory identyfikatorów, 78  
 siatka, 119, 120  
   elastyczna, 134  
   responsywna, 128  
   statyczna, 135  
   w siatce, 141  
   wyświetlanie, 127  
 silnik WebKit, 220  
 składnia  
   definicji zmiennych, 24, 57  
   domieszek, 87  
   domieszki box-shadow, 173  
   gradientu liniowego, 175  
   gradientu promienistego, 177  
   kombinacji funkcji, 117  
   reguły kolumnowej, 172  
   rozjaśniania koloru, 96  
   własności border-radius, 170  
   własności text-shadow, 169  
   zagnieżdżania, 68  
 SPDY, 54  
 sprawdzanie wersji, 35  
 statystyki Compass, 226  
 stopka, footer, 97  
 stosowanie  
   plików częściowych, 55  
   przedrostków producentów, 25  
   systemów siatkowych, 121  
 struktura pliku config.rb, 48  
 styl  
   .headline, 160  
   wyjściowy CSS, 50  
 stylizowanie strony, 64  
 Stylus, 23  
 Susy, 121, 122  
   dołączanie do projektu, 124  
   instalowanie, 123  
   tworzenie nawigacji, 133  
   tworzenie punktów  
     brzegowych, 129  
   tworzenie siatki, 122, 125  
 SVG, 190  
 symbole zastępcze, 63  
 szkielec  
   projektu, 60  
   siatkowy CSS, 117

## Ś

ścieżki  
do zasobów, 50  
względne, 53  
śródliniowe zapytania medialne,  
156

## T

technika  
clearfix, 98  
off-canvas, 133  
tworzenie  
adresów URI, 187  
cieni tekstu, 168  
modułów, 68  
nawigacji, 133  
pliku częściowego, 75  
projektu, 37, 43  
projektu Compass, 45, 46  
punktów brzegowych, 129  
siatki  
responsywnej, 128  
płynnej, 134  
statycznej, 135  
Susy, 122, 125

wielu plików CSS, 226  
zapytań medialnych, 149

## U

układ, 65  
URI danych, 188  
urządzenia przenośne, 128  
usuwanie jednostek, 209

## W

wariacje motywu, 155  
wersje Sass i Compass, 35  
wiersz poleceń, 31  
instalacja Sass i Compass, 31  
kompilowanie kodu na CSS, 38  
tworzenie projektu, 37  
własność  
border-radius, 170  
text-shadow, 168  
włączanie względnych zasobów,  
53  
wtyczka, 49  
FitText.js, 168  
wynik działania funkcji, 212  
wyświetlanie siatki, 127

## Z

zagnieżdżanie, 63  
kodu, 68  
przestrzeni nazw, 79  
reguł CSS, 26, 51  
wielopoziomowe, 77  
zamienianie  
obrazów, 187  
tekstu na obraz, 186  
zanikanie, 109  
zaokrąglenie rogów, 25  
zapytania medialne, 27, 149  
rozszane, 162  
śródliniowe, 151, 156  
w Sass, 150  
zgrupowane, 162  
zmienianie fontu, 75  
zmiennne, 22, 23  
dla punktów brzegowych, 153  
konfiguracyjne, 219  
licznikowe, 204  
projektowe, 124  
składnia definicji, 57  
wykonywanie obliczeń, 200

# 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>

# Sass i Compass

## Praktyczny przewodnik dla projektantów

Kaskadowe arkusze stylów są dziś wykorzystywane przez wszystkich projektantów stron internetowych na całym świecie. Jest to bardzo wygodne i elastyczne narzędzie, jednak prawdziwie spektakularne efekty można osiągnąć dzięki zastosowaniu CSS w połączeniu z preprocesorem Sass oraz technologią Compass. Możliwe, że nie bardzo wiesz, co znaczą te nazwy, i nie masz pojęcia, do czego mogłyby Ci się przydać te dodatki. Jeśli jednak jesteś projektantem, powinieneś czym prędzej zapoznać się z tymi narzędziami, ponieważ niebawem ułatwią Ci pracę. I nie musisz być geniuszem ani informatykiem, żeby skorzystać z ich mocy. Wystarczy, że znasz HTML i CSS.

Ta książka nie została napisana z myślą o programistach. Jej celem jest pokazanie spektrum działania technologii Sass i Compass wszystkim tym, którzy często używają CSS. Znajdziesz tu informacje na temat instalowania i konfigurowania obu dodatków, sprawnego posługiwania się kolorami i tworzenia responsywnych układów stron. Po tych podstawach przyjdzie czas na naukę pisania zapytań medialnych i eksperymentalnych deklaracji CSS dla różnych przeglądark, a także ćwiczenia ze sporządzania statystyk. Następnie dowiesz się, jak utworzyć idealnego duszka w rekordowo krótkim czasie i zautomatyzować swoją pracę za pomocą inteligentnych pętli. Przeczytaj i zastosuj, a szybko zakończysz się w dodatkach Sass i Compass!

**Sass i Compass skrócą Ci drogę do pięknych stron!**

**[PACKT]** open source  
PUBLISHING community experience distilled

<b>Helion</b>	
38256	numer katalogowy
księgarnia internetowa	
<a href="http://helion.pl">http://helion.pl</a>	
zamówienia telefoniczne	
	<b>0 801 339900</b>
	<b>0 601 339900</b>
Informatyka w najlepszym wydaniu	

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

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

siegnij po WIĘCEJ



KOD KORZYSCI

ISBN 978-83-283-1762-8



9 788328 317628

cena: 49,00 zł

**Dzięki tej książce:**

- stworzysz swój projekt z technologiami Sass i Compass
- nauczysz się tworzyć responsywne i elastyczne siatki
- poznasz metody pracy z kolorami
- opanujesz zapytania medialne
- zastosujesz zaawansowane narzędzia Sass i Compass

**Ben Frain** — projektant i programista stron WWW. Autor artykułów technicznych poświęconych platformie Mac, technologiom konsumenckim, projektowaniu stron internetowych oraz lotnictwu. Można go znaleźć na stronie internetowej <http://www.benfrain.com> i na Twitterze pod adresem [twitter.com/benfrain](http://twitter.com/benfrain).

