

Benjamin Jakobus
Jason Marah

Bootstrap 4

dla zaawansowanych

Jak pisać znakomite
aplikacje internetowe

Wydanie II

Helion 

Packt 

Tytuł oryginału: Mastering Bootstrap 4 – Second Edition

Tłumaczenie: Katarzyna Wojtkowiak

ISBN: 978-83-283-4874-5

Copyright © Packt Publishing 2018. First published in the English language under the title 'Mastering Bootstrap 4 - Second Edition – (9781788834902)'.
Copyright © 2019 by Helion SA

Polish edition copyright © 2019 by Helion SA
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 Helion SA 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 Helion SA nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Helion SA

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 231 22 19, 32 230 98 63

e-mail: helion@helion.pl

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

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:

<ftp://ftp.helion.pl/przyklady/boo4z2.zip>

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/boo4z2>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

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

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

Spis treści

O autorach	7
O recenzencie	8
Wstęp	9
Rozdział 1. Podkręcanie Bootstrapa	13
Wprowadzenie do projektu demo	14
Co oferuje Bootstrap 4	16
Układ	16
Nadawanie treściom stylu	17
Komponenty	18
Wsparcie urządzeń mobilnych	21
Klasy pomocnicze	21
Wspierane przeglądarki	21
Sass zamiast Less	22
Od pikseli do root emów	22
Mieszanie prefiksów	23
Ikony	23
Szablony	24
Kosmetyka	25
Przygotowanie projektu	25
Podsumowanie	31
Rozdział 2. Definiowanie stylu	33
System siatkowy	33
Kontenery	34
Rzędy	40
Kolumny	40

Obrazy	52
Responsywne obrazy	54
Modyfikatory obrazów	56
Responsywne narzędzia	58
Podsumowanie	60
Rozdział 3. Tworzenie układu	61
Tworzenie sekcji	61
Dodawanie komponentów Bootstrapa	64
Jumbotron	65
Zakładki	66
Karuzela	72
Karty	75
Pasek nawigacyjny	77
Okna modalne	84
Podsumowanie	89
Rozdział 4. O nawigacji, stopkach, ostrzeżeniach i treści	91
Przyklejanie paska nawigacji	92
Więcej o stylu paska nawigacji	93
Poprawa nawigacji za pomocą Scrollspy	93
Dopasowanie prędkości przewijania	96
Okruszki	99
Ikony	99
Wykorzystanie i dostosowanie ostrzeżeń	102
Tworzymy stopkę	107
Tworzymy dopasowane formularze	109
Walidacja formularza	116
Wskaźniki postępu	118
Używanie obiektów medialnych	119
Rysunki	120
Podsumowanie	122
Rozdział 5. Projektowanie usprawnione dzięki wtyczkom	123
Wykrywanie przeglądarki	124
Rozszerzona paginacja dzięki wtyczce bootpag	126
Wyświetlanie obrazów za pomocą Lightboku	135
Poprawiamy listę cen za pomocą DataTables	139
Typeahead	142
Podpowiedzi	144
Wyskakujące okienka	144
Podsumowanie	145
Rozdział 6. Wtyczki dopasowane do potrzeb	147
Anatomia wtyczek	148
JavaScript	148
Sass	153

Dopasowanie wtyczek do potrzeb	154
Konfiguracja wtyczki ostrzegawczej	154
Dostosowywanie karuzeli jQuery	161
Pisanie własnej wtyczki jQuery	166
Idea — wtyczka A11yHCM	166
Plik a11yhcm.js	169
Kod	171
Dodawanie stylu	174
Podsumowanie	176
Rozdział 7. Zaawansowane wtyczki	179
<hr/>	
Budowanie komponentu referencje za pomocą Salvatore'a	179
Poznaj Salvatore'a	182
Dodawanie Animate.css do Mojej Fotki	187
Skaczące powiadomienia	188
Animowanie siatki Salvatore'a	191
Hover	192
Dodawanie Hovera do Mojej Fotki	193
Rozbudowywanie paska nawigacyjnego	193
Ikony Awesome Hover	194
Salvatore Hover	196
Podsumowanie	199
Rozdział 8. Narzędzia	201
<hr/>	
Flexbox	201
Właściwości display i alignment	202
Pomoc dla właściwości display	203
Pomoc dla właściwości alignment	203
Obramowanie	204
Kolory kontekstowe	206
Marginesy i wypełnienie	207
Opływanie (floating)	211
Przytwierdzanie	211
Przełączanie widoczności	212
Wyrównanie tekstu i przekształcenia	213
Podświetlanie tekstu	214
Zmniejszanie rozmiaru tekstu	215
Listy	215
Listy bez stylu	215
Listy inline	216
Szerokość i wysokość	217
Kod	218
Sub i Sup	219
Formatowanie danych wprowadzanych przez użytkownika	219
Formatowanie danych wyjściowych użytkownika	220
Clearfix	220
Podsumowanie	220

Rozdział 9. Listy i akordeony	221
Grupy list	221
Wchodzenie w interakcję z elementami listy	222
Stosowanie plakietek	223
Stosowanie klas kontekstowych	224
Ukrywanie treści	225
Akordeony	226
Podsumowanie	228
Rozdział 10. Optymalizowanie strony	229
Optymalizacja CSS-u	230
Style inline	230
Długie identyfikatory i nazwy klas	231
Zasady skracania	234
Selektory grupujące	234
Czas renderowania	235
Minifikacja CSS-u i JavaScriptu	236
Przedstawiamy Grunta	236
Minifikacja i konkatencja z Gruntem	238
Automatyczne uruchamianie zadań	240
Wyrzucanie nieużywanego CSS-u ze strony	242
Przetwarzanie HTML-a	244
Ustawianie zasobów	247
Pozbywanie się komentarzy CSS-u	248
Konkatencja plików JavaScriptu	249
Podsumowanie	250
Rozdział 11. Integrowanie z AngularJS i Reactem	251
Poznajemy AngularJS	252
Przygotowanie AngularJS	252
Poprawianie komponentu referencji	253
Uczynienie referencji dynamicznymi	254
Tworzenie Promise z \$q	256
Tworzenie dyrektywy AngularJS	257
Pisanie szablonu referencji	259
Testowanie dyrektywy testimonial	260
Importowanie biblioteki Salvatore	261
Poznajemy Reacta	262
Ustawianie Reacta	262
Galeria w bibliotece React	264
Wykorzystanie karuzeli	267
Podsumowanie	271
Skorowidz	273

Podkreścanie Bootstrapa

Bootstrap ułatwia tworzenie interfejsów webowych. Jest frameworkiem służącym do rozwijania stron i aplikacji internetowych. Został utworzony przez Marka Otta i Jacoba Thorntona w 2011 r., gdy byli oni zatrudnieni w Twitterze. Dzisiaj jest frameworkiem open source i stał się jednym z najbardziej popularnych frameworków przeznaczonych do pracy nad stronami i aplikacjami internetowymi. Można z niego korzystać za darmo — zarówno do celów prywatnych, edukacyjnych, jak i komercyjnych. Z tego powodu szybko zyskał popularność. Dzisiaj tysiące organizacji polegają na Bootstrapie, włączając w to NASA, Walmarta i Bloomberga. Zgodnie z informacjami dostępnymi na <http://BuiltWith.com> ponad 10% największych stron internetowych na świecie zostało wykonanych z użyciem Bootstrapa (<http://trends.builtwith.com/docinfo/Twitter-Bootstrap>). Zatem znajomość Bootstrapa jest ważną umiejętnością i będzie stanowić solidny dodatek do zestawu umiejętności każdego web dewelopera.

Główne składowe frameworku opierają się na JavaScriptcie i CSS-ie. Dostarcza on deweloperom wszystkie niezbędne komponenty do tworzenia w pełni funkcjonalnego interfejsu webowego. W trakcie lektury tej książki poznasz najważniejsze elementy Bootstrapa i nauczysz się, jak używać frameworku w celu wykonania od zera kompletnej strony. Ponieważ dostępnych jest wiele książek poświęconych CSS-owi, HTML-owi i JavaScriptowi, zakładamy, że masz podstawową wiedzę o tych językach.

Niniejszy rozdział rozpoczniemy od wprowadzenia do projektu demo: *Moja Fotka*. Ten projekt będzie nam towarzyszyć przez całą książkę i służyć jako praktyczny punkt odniesienia.

Następnie omówimy framework Bootstrap, wymienimy jego elementy i porównamy aktualną wersję z poprzednią (Bootstrap 3).

Jakby tego było mało, ten rozdział pomoże Ci także w przygotowaniu środowiska programistycznego. Aby pracować na tej samej bazie, przeprowadzimy Cię przez proces instalowania odpowiednich narzędzi i szczegółowo omówimy sposoby na zintegrowanie Bootstrapa z projektem. Bardziej zaawansowani czytelnicy mogą ominąć tę część i przejść do rozdziału 2. „Deklarowanie stylu”.

Podsumowując, w tym rozdziale:

- zapoznamy się z projektem demo — stroną dla fotografów;
- dowiemy się, co nowego pojawiło się w ostatniej wersji Bootstrapa i czym różni się ona od poprzedniej;
- nauczymy się, w jaki sposób włączyć Bootstrapa do swojego projektu.

Wprowadzenie do projektu demo

Z tej książki nauczymy się tworzyć kompletne strony w Bootstrapie, od zera. Rozpoczniemy od prostego układu w rozdziałach 2. „Deklarowanie stylu” i 3. „Tworzenie layoutu”. Następnie poprawimy go i rozbudujemy w kolejnych rozdziałach. Koncepcja naszej strony jest prosta: chcemy przygotować witrynę dla fotografów. Dzięki naszemu projektowi (hipotetyczni) użytkownicy będą mogli zaferować swoje towary i usługi. W trakcie tworzenia strony będziemy używać tych samych narzędzi i bibliotek, z których korzystają profesjonalni programiści. Wybraliśmy konkretne wtyczki i narzędzia głównie ze względu na ich powszechne stosowanie. Nauka sposobu ich zastosowania i zintegrowania zaoszczędzi nam sporo pracy przy rozwijaniu stron internetowych w przyszłości. Narzędziami, które będą nam towarzyszyć podczas procesu projektowania *Mojej Fotki*, są: **node package manager (npm)** oraz **grunt**.

Dzięki procesowi tworzenia *Mojej Fotki* nauczymy się, jak zastosować wszystkie podstawowe koncepcje dotyczące interfejsów użytkownika oraz komponenty konieczne do zbudowania w pełni funkcjonalnej strony. Nauczymy się m.in.:

- Używania systemu siatkowego Bootstrapa, aby zorganizować informacje na swojej stronie.
- Budowania przypiętego, posiadającego efekt przewijania, nawigacyjnego paska z oznaczeniem marki.
- Korzystania z karuzeli, aby wyświetlić różne zdjęcia zaimplementowane za pomocą karuzeli i jumbotrona (jumbotron służy do wyświetlania istotnych treści). Trzeba zaznaczyć, że karuzele cieszą się coraz mniejszą sympatią. Jednakże nadal są powszechnie używane i stanowią ważny element Bootstrapa. Nie jesteśmy ani zwolennikami, ani przeciwnikami karuzeli, ponieważ jej efektywność zależy raczej od tego, w jaki sposób jej użyto, niż od tego, czy użyto jej w ogóle.
- Tworzenia własnych kart, które pozwolą użytkownikom na nawigację poprzez kolejne treści.
- Zastosowania okienek modalnych Bootstrapa.
- Wykorzystania przypiętej stopki.
- Wykonywania formularzy z możliwością wprowadzania danych dzięki kontrolkom wejścia (pole tekstowe, obszar tekstowy, przyciski) i stosowania wejściowych stylów walidacyjnych Bootstrapa.
- Najbardziej efektywnego wykorzystania klas kontekstowych.
- Tworzenia indywidualnie stylizowanych wiadomości ostrzegawczych.

- Ekspresowego budowania tabel interaktywnych do wyświetlania informacji o produktach.
- Korzystania z rozwijanych menu, własnych czcionek i ikon.
- Tworzenia okruszków, aby wspomóc nawigację.
- Używania podpowiedzi dla pól tekstowych.
- Efektywnego użycia klas pomocniczych, aby przyspieszyć proces zaprojektowania strony internetowej.
- Kontroli wyrównania i layoutu za pomocą flexboksa.
- Tworzenia interfejsów, które wspierają czytelniki ekranowe.

Oprócz nauki korzystania z Bootstrapa 4 proces tworzenia *Mojej Fotki* wprowadzi nas do kilku zewnętrznych bibliotek, takich jak Scrollspy (dla przewijanych animacji), SalvatoreJS (biblioteka uzupełniająca naszą bootstrapową siatkę), Animate.css (dla pięknych animacji CSS-owych, takich jak efekt wygaszania) oraz DataTables (pozwala na ekspresowe wyświetlenie danych w formie tabelarycznej).

Sama strona będzie się składała z różnych części:

- *Powrót*,
- *O stronie*,
- *Usługi*,
- *Galeria*,
- *Kontakt*.

Budowanie każdej z tych sekcji ma na celu naukę, jak korzystać z różnych zestawów elementów zewnętrznych bibliotek. Przykładowo: podczas pracy nad częścią powitalną nauczymy się, jak korzystać z jumbotrona, okienek dialogowych oraz z różnych czcionek i stylów, a sekcja *O stronie* pomoże w zrozumieniu, jak używać kart. Część *Usługi* stanowi wprowadzenie do niestandardowych kart — dowiemy się tutaj, jak dzięki kartom w Bootstrapie wyświetlić wiele różnych usług oferowanych przez naszą stronę.

Aby kontynuować rozwijanie sekcji *Usługi*, będziesz potrzebować bogatego materiału wizualnego, żeby przykładowe usługi dostępne na stronie naprawdę się wyróżniały. Osiągniesz to, gdy opanujesz responsywny rdzeń Bootstrapa oraz karuzelę i wtyczki *jQuery*. W sekcji *Kontakt* zademonstrujemy, jak używać formularzy i funkcji pomocniczych — dowiesz się, jak za pomocą Bootstrapa utworzyć stylowe formularze HTML-a, jak korzystać z pól formularzy i grup wejściowych i jak poradzić sobie z wyświetleniem rezultatu sprawdzenia poprawności danych.

Z końcowej części książki dowiesz się, jak zoptymalizować swoją stronę i zintegrować ją z popularnym frameworkiem JavaScriptu, AngularJS (<https://angularjs.org/>) i Reactem (<https://reactjs.org/>). Na rynku dostępne są całe książki poświęcone AngularJS, więc omówimy jedynie podstawy niezbędne do samej integracji.

Skoro już zapoznałeś się z krótkim przeglądem *Mojej Fotki*, przyjrzyjmy się Bootstrapowi 4 bardziej szczegółowo i wyjaśnimy, czym różni się od poprzedniej wersji (zob. rysunek 1.1).



Rysunek 1.1. Przedsmak tego, co zostanie zrobione: strona Moja Fotka

Co oferuje Bootstrap 4

Od pierwszego twitterowego wydania Bootstrapa, z 19 sierpnia 2011 r., wiele się zmieniło. W skrócie: pierwsza wersja Bootstrapa była zbiorem reguł CSS-u, które dawały programistom możliwość opracowania układu strony, utworzenia formularzy, przycisków oraz poprawiały ogólny wygląd i nawigację strony. Jeśli chodzi o te fundamentalne cechy, Bootstrap 4 jest ciągle podobny do swoich poprzedników — tak samo używa się frameworku do wykonania szkieletu i interfejsu posiadającego spójny styl. Jeśli chodzi o to ostatnie, osiągamy ten efekt poprzez zastosowanie stylów Bootstrapa dla przycisków, formularzy i innych elementów interfejsu użytkownika.

Utrzymały się podstawowe elementy Bootstrapa, jednakże w pozostałej części kompletnie się on zmienił. Framework wprowadza nowe klasy pomocnicze i nowe komponenty, a jednocześnie przestaje wspierać inne. Również wygląd pojedynczych komponentów został zmodyfikowany — tak samo, jak zmienił się sposób definiowania i tworzenia zewnętrznych wtyczek. Deweloperzy Bootstrapa 4 wycofali też wsparcie dla niektórych ze starszych przeglądarek, a dodali dla innych — ściślej mówiąc: dla przeglądarki systemu Android 5.0 Lollipop i WebView. Podczas omawiania głównych elementów Bootstrapa w kolejnych podrozdziałach powiemy sobie także o tych zmianach.

Układ

Prawdopodobnie najważniejszą i najczęściej wykorzystywaną możliwością frameworku jest jego zdolność do rozplanowania i zorganizowania elementów na stronie. U podstawy tej funkcjonalności znajduje się system **siatkowy** Bootstrapa: szereg klas CSS i *media queries*, które

pozwalają na proste zdefiniowanie pozycji poziomej i pionowej elementów na stronie, jednocześnie z uwzględnieniem różnic w ich rozmiarach. Korzystanie z systemu siatkowego jest tak proste jak przypisanie elementom kilku klas i określenie wielkości (rozmiaru ekranu, przy którym element zostanie przeskalowany). Nie ma potrzeby dodawania magicznego JavaScriptu ani pisania własnych reguł CSS-u. W Bootstrapie 4 system siatkowy został całkowicie odświeżony i rozbudowano go o jedną nową rozdzielczość. To oznacza, że w przeciwieństwie do Bootstrapa 3 system siatkowy oferuje rozdzielczości dla supermałych, małych, średnich, dużych i bardzo dużych wyświetlaczy. W rozdziale 2. „Deklarowanie stylu” obszernie omówimy wielkości, responsywne wartości graniczne i system siatkowy Bootstrapa.

Razem z systemem siatkowym Bootstrap oferuje także:

- responsywne kontenery;
- responsywne wartości graniczne (z ang. *breakpointy*) dla poprawy widoczności layoutu na różnych wielkościach ekranu;
- obiekty multimedialne, które pozwalają na wykonanie własnych komponentów strukturalnych;
- klasy pomocnicze pozwalające na manipulowanie elementami w responsywny sposób. Przykładowo: możesz skorzystać z klas pomocniczych layoutu, żeby ukrywać elementy w zależności od rozmiaru ekranu.

Bootstrap 4 otrzymał także wsparcie dla trybu CSS3, który nazywa się **flexbox** — pozwala on na łatwiejsze pozycjonowanie elementów przy uwzględnieniu różnych rozmiarów ekranu. Jak poradzić sobie z wyrównaniem elementów za pomocą flexboka, powiemy w rozdziale 8. „Narzędzia”.

Nadawanie treściom stylu

Jak już wspomniano, jednym z głównych celów Bootstrapa jest utrzymanie spójnego interfejsu użytkownika. To znaczy, że za pomocą Bootstrapa da się zaprojektować strony, które będą się podobnie prezentować niezależnie od rodzaju przeglądarki internetowej. Inaczej mówiąc, jak zaznaczono w oficjalnej dokumentacji Bootstrapa:

część zadania postawionego przed Bootstrapem sprowadza się do dostarczenia eleganckiej, spójnej i prostej bazy, na której można by dalej budować.

Różne przeglądarki mają różne domyślne style elementów, dlatego Bootstrap nadpisuje ustawienia przeglądarek, dając gwarancję, że różne elementy w różnych przeglądarkach będą wyświetlane w sposób jednolity. Nadpisywanie jest możliwe poprzez zastosowanie zestawu reguł CSS-u znanych zbiorczo pod nazwą **Reboot**. Te reguły są modyfikacją **Normalize** (biblioteki zewnętrznej używanej przez Bootstrapa 3, wykonanej samodzielnie przez twórców Bootstrapa).

Opierając się na tej bibliotece, Bootstrap 4 dostarcza zestaw podstawowych stylów, które pozwalają efektywnie poprawić ogólny wygląd elementów powszechnie używanych do tworzenia interfejsu użytkownika. Przykładem elementów, które można wystylizować, są przyciski (rysunek 1.2), pola wejściowe, nagłówki (rysunek 1.3), akapity, tekst, rysunki, tabele i kontrolery nawigacji.

Przycisk Primary

Przycisk Secondary

Przycisk Success

Przycisk Info

Przycisk Warning

Przycisk Danger

Rysunek 1.2. Sześć stylów przycisków, które dostarcza Bootstrap, to: `btn-primary`, `btn-secondary`, `btn-danger`, `btn-link`, `btn-info` i `btn-warning`. Istnieją również dodatkowe style, niepokazane tutaj (`btn-light`, `btn-dark` oraz przyciski z obrysem: `btn-outline`)

Dodatkowo Bootstrap oferuje style stanu, które można stosować do wszelkich kontrolerów wejścia, np. do wyłączonych przycisków lub do włączonych.

Wyświetl nagłówek 1
 Wyświetl nagłówek 2
 Wyświetl nagłówek 3
 Wyświetl nagłówek 4

Rysunek 1.3. Style typograficzne Bootstrapa. Na rysunku możemy zobaczyć stylizację różnych nagłówków

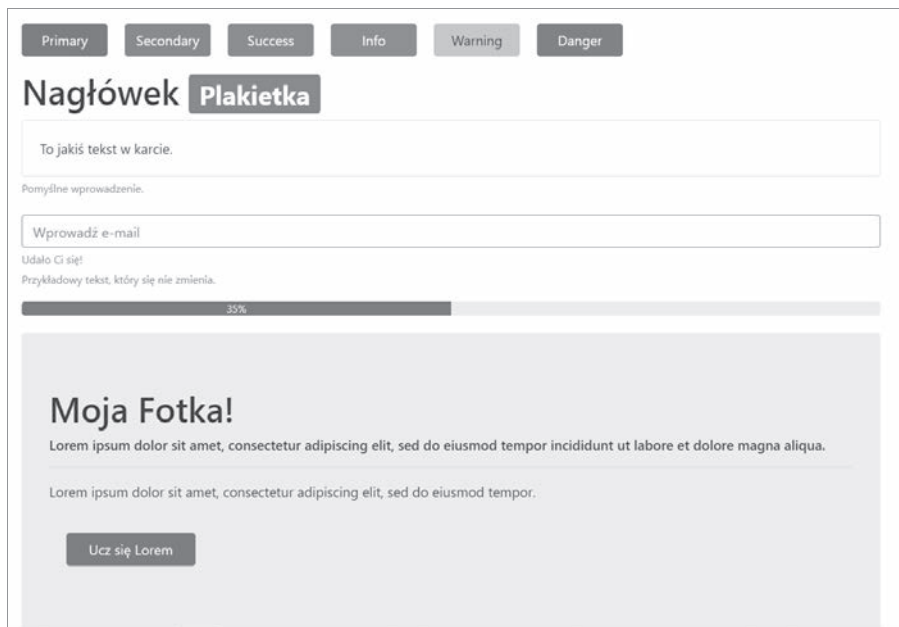
Trzeba nadmienić, że domyślny rozmiar czcionki w Bootstrapie 4 jest o 2 px większy niż w poprzedniej wersji — powiększył się z 14 px do 16 px. W konsekwencji tekst (taki jak treść elementu `body` oraz nagłówki) będzie nieco większy, jednakże zmiana rozmiaru nie będzie się rzucała w oczy. Wyświetlanemu tekstowi można teraz przypisać czcionkę z zestawu rodzimych (znanych także pod nazwą **systemowych**, co odnosi się do domyślnej kolekcji czcionek dostępnej w systemie operacyjnym) zamiast czcionek webowych. Używanie czcionek rodzimych ma następujące zalety:

- poprawiona zostaje prędkość renderowania tekstu, ponieważ przeglądarka nie musi już ściągać plików czcionki;
- zapewnia to większą spójność wyglądu i jednolite wrażenia wizualne, ponieważ treść tekstowa odpowiada teraz tej dostępnej w systemie operacyjnym, pod którym pracuje przeglądarka.

Komponenty

Bootstrap oferuje olbrzymią kolekcję podstawowych komponentów dla **interfejsu użytkownika (UI)**, które pozwalają na utworzenie każdego typu interfejsu w bardzo krótkim czasie. Ścisłe mówiąc, Bootstrap oferuje okienka modalne, pasek postępu, paski nawigacyjne, podpowiedzi, wyskakujące okna, karuzele, ostrzeżenia, rozwijane menu, grupy wejściowe, karty, paginacje, zakładki, formularze, grupy list, okruszki i jumbotrona (zob. rysunek 1.4). Wszystkie z tych komponentów, jak również rozwijane elementy — np. akordeony — są od razu gotowe do użycia.

Dodatkowo, żeby spełniały określoną funkcję, można je stylizować na jeden z czterech kolorów kontekstowych omówionych w poprzednim podrozdziale.



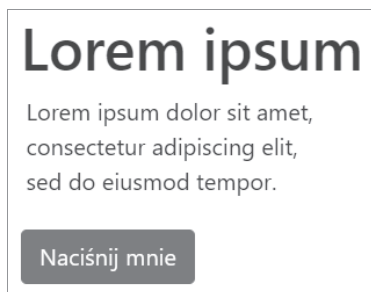
Rysunek 1.4. Różne komponenty Bootstrapa 4 w działaniu. Na rysunku widzimy jumbotrona, kartę, przykładowy tekst, pasek postępu i kilka przycisków

Czytelnicy zaznajomieni z Bootstrpem 3 zauważą, że na rysunku 1.4 nie ma paneli, studni i miniatur. To dlatego, że te komponenty zostały zastąpione nowymi — **zakładkami**. Jeżeli obca Ci jest koncepcja studni: studnie są komponentem UI, który pozwala deweloperom podkreślić tekst poprzez dodanie do niego zacienionej ramki. Z drugiej strony panel również służy uwydatnianiu informacji, ale poprzez określenie odstępu tekstu od ramki (padding) i zastosowanie zaokrąglonych obramowań. Jednakże ani panele, ani studnie nie dostosowują się do różnego rodzaju treści.

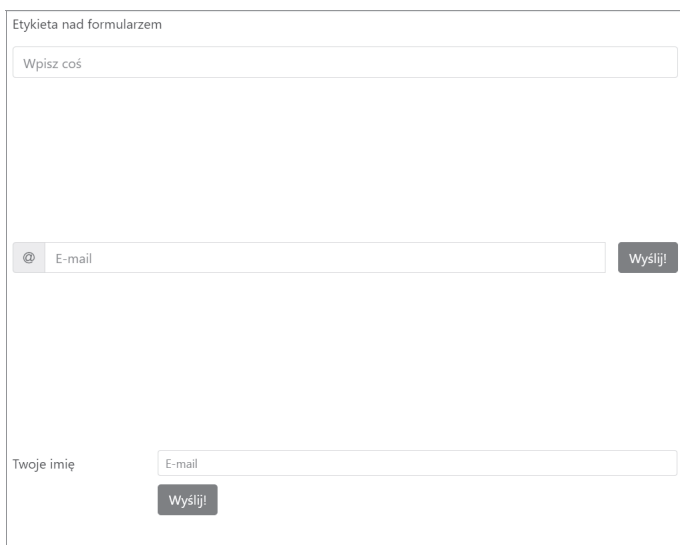
Lukę tę wypełniają karty. Są mniej restrykcyjne i zapewniają większą elastyczność w doborze typu treści, np. obrazków, list czy tekstu. Pozwalają także na dodanie stoppek i nagłówków (zob. rysunek 1.5).

Również formularze zostały całkowicie przebudowane i teraz umożliwiają konfigurację różnych opcji układu (zob. rysunek 1.6) oraz mają walidacyjne klasy wejścia (rysunek 1.7).

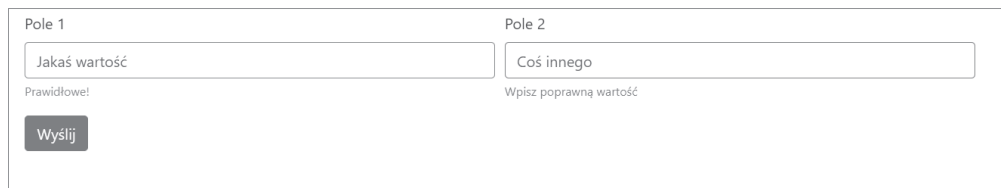
Dzięki zastosowaniu tych klas programista może zostać szybko poinformowany o błędnych danych wejściowych albo potwierdzić użytkownikowi pomyślne przesłanie formularza. W przeszłości wymagało to korzystania z zewnętrznych bibliotek lub ręcznej implementacji. Rozmiar można dostosować również pojedynczym kontrolerom formularzy — jak np. przyciskom — opcja, która wydaje się trywialna, ale często jest pożądana.



Rysunek 1.5. Komponent karta zastąpił studnie, miniatury i panele



Rysunek 1.6. Formularze można dostosować za pomocą różnych opcji układu — na poprzednim rysunku można zobaczyć trzy formularze: jeden z etykietą nad polem wejściowym, jeden z elementami ułożonymi w linii i jeden z etykietą po lewej stronie pola wejściowego



Rysunek 1.7. Nowe style walidacyjne danych wejściowych wskazują pomyślne przetworzenie danych

Wygląd kontrolerów formularzy można dopasować, utrzymując jednocześnie spójność danych wejściowych na różnych przeglądarkach — użyjemy do tego klasy custom-* (zob. rysunek 1.7). Mówiąc w skrócie: te klasy pozwalają zastąpić domyślne pola wyboru (*checkbox*), przyciski opcji (*radio*), listy rozwijane (*select*) i pola przesyłania plików (*file upload*) przy zachowaniu spójnego wyglądu — zob. rysunek 1.8.

Zaznacz ten checkbox Przełącz przycisk radio

Rysunek 1.8. Dopasowane do potrzeb kontrolki, które zastępują domyślne kontrolery przeglądarek, aby zapewnić konsekwencję w UI między przeglądarkami

Wsparcie urządzeń mobilnych

Podobnie jak jego poprzednik, Bootstrap 4 pozwala na tworzenie stron przyjaznych urządzeniom mobilnym bez dużych nakładów dodatkowej pracy. Bootstrap domyślnie jest zaprojektowany tak, aby pracował z różnymi rozdzielczościami i rozmiarami ekranu — od telefonu, poprzez tablet, do komputera. Właściwie Bootstrapowi przyświeca filozofia **najpierw mobilne**, co oznacza, że jego komponenty powinny się wyświetlać i funkcjonować bez zastrzeżeń na najmniejszym dostępnym rozmiarze ekranu. Stożąc za tym idea jest prosta. Wyobraźmy sobie tworzenie strony internetowej bez pamiętania o małych ekranach mobilnych. W takim przypadku „napakujemy” swoją stronę dużą liczbą przycisków, etykiet i tabel. Prawdopodobnie odkryjemy problemy z użytecznością takiej strony dopiero, gdy jakiś użytkownik wejdzie na nią z urządzenia mobilnego i jedyne, co będzie w stanie zobaczyć, to małą stronę pełną przycisków i formularzy. Na tym etapie będziemy musieli poprawić cały interfejs, żeby mógł się wyświetlać na małych ekranach. Właśnie z tego powodu Bootstrap promuje podejście oddolne, wymuszając na deweloperach pracę nad prawidłowym wyświetlaniem interfejsu na najmniejszym dostępnym ekranie, zanim rozpoczną pracę na większych rozmiarach ekranów.

Klasy pomocnicze

Pomijając gotowe elementy, Bootstrap oferuje duży wybór klas pomocniczych, które dostarczają najczęściej spotykane funkcjonalności w celu zredukowania liczby powtarzających się reguł stylizacji. Przykłady tych narzędzi zawierają m.in. klasy szybkiej stylizacji obramowań, usuwanie opływania, tworzenie typowej ikony zamknięcia, definiowanie kolorów, zmiana opcji wyświetlania i widoczności elementów oraz dopasowanie rozmiaru i odstępów. Niektóre z tych klas pomocniczych istniały już w wersji 3, inne są całkiem nowe lub zostały zaadaptowane, zmodyfikowane i zmieniono im nazwę. Będziemy korzystać z tych klas podczas studiowania materiału zawartego w książce, a całą naszą niepodzielną uwagę zyskają w rozdziale 8. „Narzędzia”, np. reguły wyrównywania tekstu, ukrywania elementu lub ustawiania kontekstowego koloru dla ostrzeżeń.

Wspierane przeglądarki

Framework wspiera wszystkie najnowsze, stabilne wersje współczesnych przeglądarek zarówno na desktopach jak i urządzeniach mobilnych: Firefoksa, Chrome’a, Opery, Safari, Internet Explorera 10-11 i Microsoft Edge’a. Co więcej:

Alternatywne przeglądarki, które używają ostatniej wersji WebKity, Blinka lub Gecko, czy to bezpośrednio, czy też API przeglądarek udostępnianych przez system operacyjny, nie są wspierane wprost. Jednakże Bootstrap powinien (w większości przypadków) działać prawidłowo również i w tych przeglądarkach. Bardziej szczegółowe informacje znajdują się poniżej.

(źródło: oficjalna dokumentacja Bootstrapa 4)

Należy podkreślić, że w przeciwieństwie do swojego poprzednika Bootstrap 4 nie wspiera już Internet Explorera w wersji 9 i wcześniejszych. Decyzja o wsparciu jedynie nowszych wersji Internet Explorera była rozsądna, ponieważ Microsoft nie oferuje już pomocy technicznej ani aktualizacji dla wersji 8 i wcześniejszych (stan na styczeń 2016 r.). Co więcej, jak przekonamy się w dalszej części książki, Bootstrap 4 opiera się na flexboksie, który również nie jest wspierany przez starsze przeglądarki.

Ponadto Internet Explorer 8 (i wcześniejsze) nie wspiera jednostek `rem`, co oznacza, że Bootstrap 4 musiałby zapewnić rozwiązanie problemu. To implikowałoby wiele dodatkowej pracy dla programistów i potencjalnych nieścisłości. Poza tym tworzenie responsywnych stron dla Internet Explorera 8 (i wcześniejszych) jest trudne, bo przeglądarka nie obsługuje media queries w CSS-ie. Biorąc pod uwagę te trzy rzeczy, porzucenie wsparcia dla Internet Explorera było najbardziej rozsądną decyzją.

Sass zamiast Less

Less i Sass (*Syntactically Awesome Stylesheets*) są rozszerzeniami języka CSS — tzn. są językami, które rozszerzają słownictwo CSS-u, żeby uczynić prostszym tworzenie wielu olbrzymich i kompleksowych arkuszy stylów. Mimo że Less i Sass są diametralnie różnymi językami, to sposób, w jaki rozszerzają CSS, jest taki sam — oba opierają się na preprocesorze. Podczas gdy tworzymy swoją wersję, preprocesor jest uruchomiony i parsuje skrypt Less/Sass, zamieniając instrukcje Less lub Sass na czysty CSS.

Less jest oficjalnym językiem Bootstrapa 3, a Bootstrap 4 został napisany od nowa, całkowicie w Sassie. Less i Sass są skompilowane w CSS do pliku `bootstrap.css`. Jednocześnie została utworzona wersja zminiaturyzowana — `bootstrap.min.css` — oraz pliki map. To ten plik CSS-u, do którego będziemy się odwoływać przez całą książkę (z wyjątkiem rozdziału 3. „Tworzenie layoutu”). W konsekwencji nie musisz znać Sassa, żeby nadążyć za książką. Jednakże zachęcamy, abyś przerobił 20-minutowy kurs wprowadzający do Sassa, jeżeli jest to dla Ciebie kompletna nowość. Możesz być pewny, że jeśli znasz CSS, nie będziesz potrzebować niczego więcej. Składnia języka jest bardzo zbliżona do normalnego CSS-u, a jego podstawowe koncepcje są podobne do zawartych w dowolnym języku programowania.

Od pikseli do root emów

W przeciwieństwie do swojego poprzednika Bootstrap 4 nie używa już **pikseli (px)** jako jednostki miary typograficznej. Korzysta za to z **root emów (rem)**. Rozumowanie stojące za wyborem `rem` jest oparte na powszechnym problemie z pikselami: strony używające pikseli mogą nie

renderować się poprawnie lub nie tak, jak zakładano, ponieważ użytkownicy zmieniają wielkość czcionek przeglądarki. Wykorzystanie jednostki miary, która jest powiązana z korzeniem strony, pomaga w rozwiązaniu tego problemu, gdyż elementowi bazowemu zostaną nadane wymiary relatywnie do bazowej czcionki przeglądarki internetowej. W efekcie strona zostanie przeskalowana odpowiednio do korzenia.

Typograficzne jednostki miary

Ujmując to prosto: typograficzne jednostki miary determinują rozmiar elementów i czcionki. Najczęściej używanymi jednostkami są px i em. Pierwsze jest skrótem od piksel i jako punkt odniesienia, konieczny do podania dokładnego rozmiaru czcionki, wykorzystuje piksel. To znaczy, że dla wyświetlacza o rozdzielczości 96 punktów na cal (**dots per inch — dpi**) 1 px będzie równy faktycznemu pikselowi na ekranie. W przypadku wyższych rozdzielczości piksel referencyjny spowoduje przeskalowanie piksela tak, aby pokrywał się z rozdzielczością wyświetlacza. Przykładowo: czcionka o rozmiarze 100 px oznacza, że czcionka ma dokładnie rozmiar 100 pikseli (na wyświetlaczu 96 dpi) bez względu na inne elementy na stronie.

Em jest jednostką miary powiązaną z *rodzicem* elementu, do którego jest przypisywana. Przykładowo: jeśli mielibyśmy dwa zagnieżdżone elementy *div*: zewnętrzny element z czcionką o wielkości 100 px i wewnętrzny z czcionką o wielkości 2 em, to czcionka wewnętrznego elementu zostałaby przetransponowana do wielkości 200 px (w tym przypadku 1 em = 100 px). Problem z używaniem jednostki, która jest powiązana z elementem nadrzędnym, polega na tym, że wzrasta stopień skomplikowania całego kodu — zagnieżdżanie elementów utrudnia obliczenia wielkości.

Ostatnio wprowadzona jednostka rem ma na celu zlikwidowanie wad em i px oraz połączenie ich mocnych stron. Rem nie jest już relatywny do elementu nadrzędnego, ale do korzenia strony.

Mieszanie prefiksów

Ponieważ przy automatycznym dodawaniu prefiksów do reguł CSS-u Bootstrap 4 polega na narzędziu zwanym *Autoprefixer*, domieszki z prefiksami dla różnych przeglądarek okazały się niepożądane w wersji 3.2 i zostały teraz całkowicie usunięte. Dodatkowo również poniższe domieszki zostały usunięte (źródło: *dokumentacja Bootstrapa 4* — <https://getbootstrap.com/docs/4.0/migration/>):

- animation, animation-delay, animation-direction, animation-duration,
- ↳ animation-fill-mode, animation-iteration-count, animation-name,
- ↳ animation-timing-function, backface-visibility, box-sizing, content-columns,
- ↳ hyphens, opacity, perspective, perspective-origin, rotate, rotateX, rotateY,
- ↳ scale, scaleX, scaleY, skew, transform-origin, transition-delay,
- ↳ transition-duration, transition-property, transition-timing-function,
- ↳ transition-transform, translate, translate3d, user-select.

Ikony

Do Bootstrapa 3 dołączono *Clifikony* — ładną kolekcję ponad 250 czcionek gotowych do użycia. W teorii są wspaniałe, jednak praktyka przynosi kilka problemów:

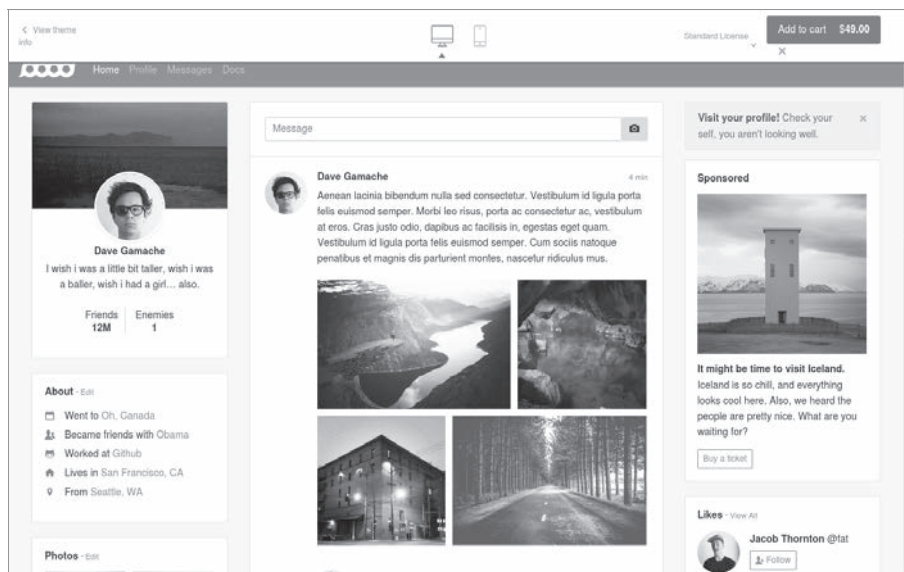
- ikony mogą spowodować awarię czytników z synteźatorami mowy, ponieważ będą się one starały przeczytać ikonę;
- w przypadku czcionek wspierających czytanie ikony się nie wyświetlają;
- jeżeli ikony się nie wyświetlą, przeglądarka zastąpi je domyślną czcionką; kontrolki korzystające z tych ikon będą w najlepszym wypadku wyglądać zabawnie.

Te trzy problemy połączone z chęcią uczynienia Bootstrapa 4 prostszym w użyciu doprowadziły programistów do decyzji o usunięciu glifikon (aczkolwiek nie przeszkadza to w samodzielnym dołączeniu ikon przez użytkowników).

Szablony

Bootstrap 4 dostarcza kolekcję szablonów premium. W czasie pisania książki dostępne były trzy różne szablony:

- *Dashboard* — zawiera komponenty premium dla stron administracyjnych i analitycznych;
- *Marketing* — dla generycznych stron marketingowych i e-commerce;
- *Application* — zawiera komponenty i narzędzia do budowy generycznych aplikacji webowych (zob. rysunek 1.9).



Rysunek 1.9. Szablon premium, Application, ze strony <https://themes.getbootstrap.com/>

Każdy z tych tematów bezproblemowo współpracuje z Bootstrapem 4 i prezentuje wszystkie jego zalety. Licencja do wielokrotnego wykorzystania tych szablonów kosztuje 49 dolarów, rozszerzona licencja natomiast ma cenę 499 dolarów i pozwala na redystrybucję oraz odsprzedanie.

Kosmetyka

Programiści położyli szczególny nacisk na dopasowanie Bootstrapa 4 do indywidualnych potrzeb użytkownika. Zostało wprowadzonych wiele nowych zmiennych, aby w prosty sposób można było dokonać pożądanych zmian. Za pomocą zmiennych Sass `$enabled-*` (do znalezienia w `node_modules/bootstrap/scss/_variables.scss`) można teraz włączyć lub wyłączyć dane ustawienia globalne CSS-u.

Przygotowanie projektu

Teraz, kiedy wiemy, czego się spodziewać po Bootstrapie, przygotujmy nasz projekt:

1. Utwórzmy nowy katalog projektu i nazwijmy go *MojaFotka*. To będzie główny katalog naszego projektu.
2. Utwórzmy pusty plik *index.html* i wstawmy do niego poniższy kod:

```
<!DOCTYPE html>
<html lang="pl">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1,
  ↳shrink-to-fit=no">
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <title>MojaFotka</title>
</head>
<body>
  <div class="alert alert-success"> Witaj, świecie!
  </div>
</body>
</html>
```

Zwróćmy uwagę na trzy znaczniki meta. Pierwszy z nich informuje przeglądarkę, że dokument jest kodowany w utf-8. Jako że Bootstrap optymalizuje swoje treści dla urządzeń mobilnych, konieczny jest kolejny znacznik meta, aby wesprzeć skalowanie okna roboczego. Ostatni znacznik meta wymusza na przeglądarce renderowanie dokumentu za pomocą trybu renderowania ostatniego dokumentu dostępnego podczas korzystania z Internet Explorera.

3. Otwórzmy w przeglądarce plik *index.html*. Powinna się ukazać pusta strona z napisem *Witaj, świecie*.

Nadszedł czas, aby załączyć Bootstrapa. Jego trzon stanowi sławny arkusz stylów CSS-u. Bootstrap demonstruje potężne cechy CSS-u przy zachowaniu prostej w użyciu składni. Ponieważ jest to arkusz stylów, dołączamy go do projektu tak samo jak każdy inny wykonany samodzielnie arkusz stylów — tzn. otwieramy *index.html* i linkujemy bezpośrednio do arkusza.

Skalowanie okna roboczego

Termin okno robocze (*viewport*) odnosi się do rozmiaru wyświetlacza dostępnego dla renderowania treści strony. Metaznacznik `viewport` pozwala zdefiniować ten dostępny rozmiar. Skalowanie *viewportu* za pomocą metaznacznika zostało po raz pierwszy wprowadzone przez Apple, a w trakcie pisania książki jest wspierane przez wszystkie główne przeglądarki.

Za pomocą parametru `width` możemy zdefiniować dokładną szerokość okna roboczego (*viewportu*) użytkownika. Przykładowo `<meta name="viewport" content="width=320">` wyda instrukcję przeglądarce, aby ustawiła szerokość okna na 320 px. Umiejętność kontrolowania szerokości okna jest przydatna w trakcie tworzenia stron przyjaznych urządzeniom mobilnym. Domyślnie przeglądarki mobilne będą próbowały zmieścić całą stronę w swoim oknie, pomniejszając wszystko tak dalece, jak to możliwe. To pozwala użytkownikom oglądać strony, które nie były przeznaczone do użytku mobilnego, i wchodzić w interakcję z nimi. Jednakże Bootstrapowi przyświeca filozofia „najpierw urządzenia mobilne”, więc pomniejszanie będzie stanowiło niepożądany skutek uboczny. Przykładowo: punkty graniczne (które omówimy w rozdziale 2. „Deklarowanie stylu”), nie będą już działały jak trzeba, ponieważ będą musiały sobie radzić z pomniejszonym odpowiednikiem na stronie. Dlatego jednoznaczne zdefiniowanie szerokości okna jest tak ważne. Gdy wpisujemy `content="width=device-width, initial-scale=1, shrink-to-fit=no"`, informujemy przeglądarkę, że:

- chcemy, aby szerokość okna była równa faktycznej szerokości okna na urządzeniu;
- nie chcemy żadnego początkowego powiększenia;
- nie chcemy, aby treść została zmniejszona w celu zmieszczenia jej w oknie.

Na tym etapie będziemy używać instalacji Bootstrapa hostowanej na oficjalnym *Content Delivery Network (CDN)*¹. Włączymy w tym celu następujący znacznik HTML do sekcji `head` dokumentu HTML (sekcja *head* w dokumencie odnosi się do treści zawartej pomiędzy `<head>` znacznikiem otwierającym i `</head>` zamykającym):

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
```

Bootstrap polega na jQuery, frameworku JavaScriptu, który stanowi abstrakcję pozwalającą na uproszczenie najczęstszych operacji JavaScriptu (takich jak selekcja elementów i obsługa zdarzeń). Jeżeli chciałbyś wykorzystać Bootstrapa jedynie do nadania stronie szkieletu, nie musisz dołączać plików JS. Te pliki potrzebne są jedynie, gdy wykorzystuje się któryś z następujących komponentów:

- ostrzeżenia,
- przyciski,
- karuzele,
- zwijana treść,
- rozwijane listy,

¹ Polecamy tę metodę również, gdyby opisywany przez autora proces instalowania poszczególnych komponentów z różnych powodów okazał się niemożliwy na komputerze Czytelnika lub nie przyniósł oczekiwanych rezultatów. Dotyczy to nie tylko linku do samego Bootstrapa, ale także np. do zbioru *Font Awesome* (<https://maxcdn.bootstrapcdn.com/font-awesome/4.5.0/css/font-awesome.min.css>), o którym będzie mowa w dalszej części książki. — *przyp. tłum.*

- paski nawigacyjne,
- odpowiedzi,
- Scrollspy.

Ponieważ będziemy korzystać ze wszystkich wymienionych powyżej elementów, musimy dołączyć do Bootstrapa plik JavaScriptu, a najpierw musimy dołączyć jQuery. Oba włączenia powinny zostać wprowadzone przed znacznikiem zamykającym `</body>`.

Dodajmy do dokumentu HTML następujące linijki:

```
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/
↳1.12.9/umd/popper.min.js">
</script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js">
</script>
```

Te skrypty oczywiście mogłyby zostać wgrane na górze strony. Jednakże wgrywanie ich na dole dokumentu jest uważane za najlepszą praktykę i przyspiesza ładowanie strony oraz pozwala ominąć problemy z JavaScriptem, które nie pozwoliłyby stronie się wczytać. Dzieje się tak dlatego, że przeglądarki nie ściągają równolegle wszystkich zależności (aczkolwiek pewna liczba zapytań tworzona jest asynchronicznie, w zależności od przeglądarki i domeny). Konsekwentnie: wymuszenie na przeglądarce wczesnego ściągnięcia zależności zablokuje renderowanie strony do czasu ściągnięcia tych elementów. Co więcej: upewnienie się, że skrypty zostały załadowane, zagwarantuje, że w momencie odwołania się do operacji **Document Object Model (DOM)** w skrypcie mamy pewność, że elementy strony zostały już zrenderowane. W rezultacie możemy ominąć sprawdzanie, czy dane elementy istnieją.

Czym jest Content Delivery Network?

Celem **Content Delivery Network (CDN)** jest udostępnienie użytkownikom treści o najwyższym poziomie dostępności. To znaczy, że CDN stara się udostępnić treści tak, żeby nigdy (lub bardzo rzadko) nie były one niedostępne. W tym celu treści są zazwyczaj przechowywane na dużych, zdecentralizowanych serwerach. CDN w zasadzie pozwala podłączyć się do arkusza stylów Bootstrapa, zwalniając użytkownika z konieczności jego samodzielnego hostowania.

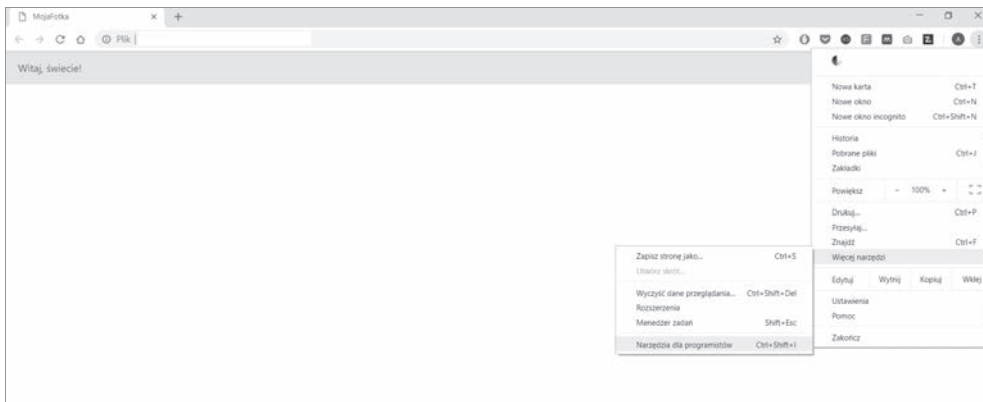
Teraz należy zapisać zmiany i ponownie uruchomić `index.html` w przeglądarce. Łącuch `Witaj, świecie` powinien teraz mieć zielone tło (rysunek 1.10).

Witaj, świecie!

Rysunek 1.10. Witaj, świecie z użyciem Bootstrapa 4

Teraz, kiedy mamy już Bootstrapa w naszym projekcie, otworzymy konsolę przeglądarki (na Chromie pod Windowsem to skrót `Ctrl+Shift+I`; na MacOS X możesz nacisnąć `Cmd+Alt+I`).

Do wyświetlenia wyskakujących okienek i podpowiedzi Bootstrap wymaga zewnętrznej biblioteki Popper, więc konsola poinformuje o błędzie (rysunek 1.6). Ilustruje to rysunek 1.11.



Rysunek 1.11. Narzędzia deweloperskie Chrome'a można otworzyć, wybierając kolejno z menu: Więcej narzędzi, a następnie klikając Narzędzia dla programistów. Po prawej stronie pojawi się nowe okno. Na karcie Konsola wyświetli się błąd z powodu braku elementu zależnego

Popper dostępny jest poprzez *CloudFlare CDN* i składa się z pliku CSS-u oraz JavaScriptu. Zależność jest biblioteką, która pozwala na proste pozycjonowanie podpowiedzi oraz wyskakujących okien i zajmuje ok. 6 KB. Trzeba zaznaczyć, że dla Bootstrapa wymagana jest wersja 1.12.9, aczkolwiek framework wciąż będzie działał prawidłowo (z wyjątkiem podpowiedzi oraz wyskakujących okienek), jeżeli biblioteka zostanie dołączona w nieprawidłowej wersji:

```
<!DOCTYPE html>
<html lang="pl">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <title>MojaFotka</title>
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
</head>

<body>
  <div class="alert alert-success">
    Witaj, świecie!
  </div>
  <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js">
</script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js">
</script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js">
</script>
</body>

</html>
```

CDN jest wartościowym udogodnieniem, jednakże istnieje kilka powodów, dla których czasami korzystanie z zewnętrznego CDN może nie być pożądane:

- CDN przynosi dodatkową przyczynę możliwej awarii, ponieważ polegasz na zewnętrznych serwerach;
- prywatność i bezpieczeństwo użytkowników mogą być zagrożone, gdyż nie ma gwarancji, że dostawca CDN nie przemyca złośliwego kodu do hostowanych bibliotek, jak również nie ma pewności, że CDN nie będzie śledzić użytkowników;
- niektóre CDN mogą być blokowane przez dostawców w różnych regionach geograficznych;
- jeśli będziesz polegać na zdalnym CDN, prace offline nie będą możliwe;
- nie będziesz mógł optymalizować plików hostowanych na CDN; ta utrata kontroli może mieć wpływ na zachowanie przeglądarki (aczkolwiek częściej niż rzadziej dostajesz zoptymalizowaną wersję biblioteki poprzez CDN).

Zamiast polegać na CDN, można samodzielnie ściągnąć jQuery, Poppera i pliki projektu Bootstrap. Następnie można skopiować te biblioteki do głównego katalogu projektu i podlinkować je do plików dystrybucyjnych. Przewagą tego sposobu jest to, że ręczne utrzymywanie kolekcji zależności może szybko stać się kłopotliwe i prawie niemożliwe, gdy zwiększą się rozmiar i stopień skomplikowania strony.

Dlatego nie będziemy ściągać Bootstrapa samodzielnie. Zamiast tego polecimy NPM, by to zrobił. NPM jest systemem zarządzania pakietami — narzędziem, którego można użyć do zarządzania zależnościami swojej strony. Automatycznie ściąga, organizuje i (na komendę) uaktualnia zależności strony. Aby zainstalować NPM, należy przejść do <https://www.npmjs.com/get-npm>.

Kiedy NPM zostanie zainstalowany, trzeba otworzyć terminal, przejść do głównego katalogu swojego projektu (tego, który został przygotowany wcześniej) i przejąć kontrolę nad instalacją Bootstrapa:

```
npm init
npm install bootstrap
```

Zostanie utworzony nowy folder w strukturze projektu:

```
| node_modules
| | bootstrap
| | | LICENSE
| | | README.md
| | | package.json
| | | dist
| | | js
| | | scss
```

Wszystkie te pliki i katalogi zostaną wyjaśnione później. Na razie możemy spokojnie zignorować wszystko oprócz katalogu `dist` znajdującego się w `node_modules/bootstrap/`. Należy otworzyć katalog `dist`. Powinniśmy zobaczyć dwa podfoldery:

- `css`,
- `js`.

Nazwa `dist` oznacza dystrybucję. Zazwyczaj katalog dystrybucji zawiera gotowy kod, który użytkownicy mogą wykorzystać. Jak sama nazwa wskazuje, katalog `css` wewnątrz `dist` zawiera gotowe do użycia arkusze stylów. Podobnie katalog `js` zawiera pliki JavaScriptu, które komponują Bootstrapa.

Aby się odwołać do lokalnego pliku CSS-u w naszym `index.html`, zmodyfikujmy atrybut `href` znacznika linku, który się odwołuje do `bootstrap.min.css`:

```
<link rel="stylesheet" href="node_modules/bootstrap/dist/css/bootstrap.min.css">
```

To samo zróbmy z plikiem JavaScriptu:

```
<script src="node_modules/bootstrap/dist/js/bootstrap.min.js"></script>
```

Powtórzmy ten proces dla jQuery i Poppera. Aby zainstalować jQuery za pomocą NPM, użyjmy poniższej komendy:

```
npm install jquery@>=3.0.0
```

Tak jak poprzednio nowy folder zostanie utworzony wewnątrz folderu `node_modules`:

```
| node_modules
| | jquery
| | | AUTHORS.txt
| | | LICENSE.txt
| | | README.md
| | | bower.json
| | | dist
| | | external
| | | package.json
| | | src
```

Tak jak poprzednio interesuje nas tylko zawartość folderu `dist`, który oprócz innych plików będzie zawierał skompresowaną wersję biblioteki `jquery.min.js`.

Odwołajmy się do tego pliku, modyfikując atrybut `src` skryptu, który w chwili obecnej wskazuje na `jquery.min.js` z CDN, zamieniając URL na ścieżkę do naszej lokalnej kopii jQuery:

```
<script src="node_modules/jquery/dist/jquery.min.js"></script>
```

Na koniec powtórzmy te kroki dla Poppera:

```
npm install popper.js@^1.12.9
```

Gdy instalacja się zakończy, powstanie podobna struktura folderów jak poprzednio. Zweryfikujmy zawartość `node_modules/popper.js/dist` i zamieńmy odniesienie do CDN Poppera w pliku `index.html` na jego lokalny odpowiednik. Końcowy plik `index.html` powinien wyglądać następująco:

```
<!DOCTYPE html>
<html lang="pl">

<head>
  <meta charset="utf-8">
```



```

<meta name="viewport" content="width=device-width, initial-scale=1,
↳shrink-to-fit=no">
<meta http-equiv="x-ua-compatible" content="ie=edge">
<title>MojaFotka</title>
<link rel="stylesheet" href="node_modules/bootstrap/dist/css/bootstrap.min.css">
</head>

<body>
  <div class="alert alert-success">
    Witaj, świecie!
  </div>
  <script src="node_modules/jquery/dist/jquery.min.js"></script>
  <script src="node_modules/popper.js/dist/umd/popper.min.js">
</script>
  <script src="node_modules/bootstrap/dist/js/bootstrap.min.js">
</script>
</body>

</html>

```

Teraz trzeba odświeżyć *index.html* w przeglądarce, aby się upewnić, że wszystko działa.

Z jakiego IDE i przeglądarki powinienem korzystać, gdy przerabiam przykłady z tej książki?

Chociaż polecamy IDE firmy JetBrains czy Sublime Text oraz Google Chrome, możesz wybrać dowolne narzędzie. Nasz gust co do IDE i przeglądarki jest subiektywny. Jednakże trzeba pamiętać, że Bootstrap 4 nie wspiera Internet Explorera 9 i wcześniejszych. Jeżeli zatem korzystasz z IE9, powinieneś zaktualizować go do najnowszej wersji.

Podsumowanie

Celem tego rozdziału było wprowadzenie do Bootstrapa 4 i wyjaśnienie:

- co nam oferuje;
- dlaczego powinniśmy z niego korzystać;
- jak rozpocząć pracę.

Teraz powinieneś rozumieć, że Bootstrap pomaga deweloperom w tworzeniu strony, nadawaniu jej struktury i stylizowaniu treści. Powinieneś także wiedzieć, jak zainicjować prostą stronę HTML.

Podczas szerszej dyskusji nakreśliśmy i wyjaśniliśmy ważne koncepcje w tworzeniu stron internetowych, takie jak: **typograficzne jednostki miary** czy definicję, przeznaczenie i uzasadnienie użycia sieci **Content Delivery Network**.

Na koniec zaznajomiliśmy Cię z przykładowym projektem — *Moja Fotka* — który będzie obecny z nami przez resztę książki i będziemy z niego korzystać, by zilustrować sposób korzystania z Bootstrapa, żeby zaprojektować współczesną stronę.

Przejdźmy teraz do rozdziału 2. „Deklarowanie stylu”, żebyśmy mogli wprowadzić system siatkowy Bootstrapa, tworząc ogólny układ pierwszej strony naszego projektu.

Skorowidz

A

- akordeony, 226
- AngularJS, 252
- animacja, 187
- animowanie siatki Salvatore'a, 191
- arkusze stylów
 - karuzeli, 164
 - ostrzeżenia, 157
- automatyczne
 - rozmieszczanie kolumn, 46
 - uruchamianie zadań, 240

B

- biblioteka
 - Hover, 192
 - Lightbox, 135
 - React, 262
 - Salvatore, 179, 261
 - Typeahead, 142
- bootpag, 126
- Bootstrap 4, 13, 16
- breakpoint, 35
- budowa pudełkowa, 39

C

- CDN, Content Delivery Network, 27
- clearfix, 220
- cytaty, 121
- czas renderowania, 235

D

- DataTables, 139
- definicja
 - klasy, 149
 - siatki, 42
 - stylu, 33
- dodawanie
 - animacji, 187
 - komponentów, 64
- DOM, Document Object Model, 27
- dostosowanie ostrzeżeń, 102
- dynamika referencji, 254
- dyrektywa testimonial, 260
- dyrektywy, 257

E

- element
 - code, 218
 - sub, 219
 - sup, 219
- elementy
 - flex, 50
 - listy, 222

F

- Flexbox, 17, 201
- floating, 211
- formatowanie danych, 220

formularze, 109
 walidacja, 116
 framework
 AngularJS, 252
 Bootstrap, 4, 13, 16

G

galeria, 264
 Grunt, 236
 grupy list, 221

H

Hover, 192
 ikony, 194
 rozbudowywanie paska nawigacyjnego, 193
 HTML, 244

I

identyfikatory, 231
 ikony, 23, 99
 Awesome Hover, 194
 implementacja API danych, 152
 importowanie biblioteki Salvattore, 261
 interfejs użytkownika, UI, 18

J

JavaScript, 148
 konkatenacja plików, 249
 minifikacja, 236
 rozbudowywanie funkcjonalności, 158
 rozszerzanie funkcjonalności, 163
 jednostki miary typograficznej, 22
 jQuery, 152

K

karty, 75
 karuzela, 72, 267
 jQuery, 161
 funkcjonalności, 163
 kod, 162
 klasa, 149
 clearfix, 220
 code, 218

container, 35
 pagination, 127
 klasy
 kontekstowe, 224
 pomocnicze, 21
 kolory kontekstowe, 206
 kolumny, 40
 komentarze CSS-u, 248
 komponent, 18
 Jumbotron, 65
 karuzela, 72
 navbar, 61
 referencje, 179, 253
 konfiguracja wtyczki ostrzegawczej, 154
 konkatenacja, 238
 plików JavaScriptu, 249
 kontener, 34
 Container-fluid, 38
 kosmetyka, 25

L

Less, 22
 Lightbox, 135
 listy, 221
 bez stylu, 216
 inline, 216

M

marginesy, 207
 minifikacja, 238
 CSS-u, 236
 modyfikatory obrazów, 56
 MVC, Model-Widok-Kontroler, 251

N

nadawanie treściom stylu, 17
 narzędzia, 201
 konkatenacja, 238
 minifikacja, 238
 responsywne, 58
 narzędzie Grunt, 236
 nawigacja, 93
 nazwy klas, 231

O

obiekt
 jQuery, 148
 medialny, 119
 typu singleton, 254
 obietnice, 256
 obramowanie, 204
 obrazy, 52
 modyfikatory, 56
 responsywne, 54
 offsetowanie, 51
 okna modalne, 84
 okno robocze, 34
 okruszki, 99
 opływanie, 211
 optymalizacja
 CSS-u, 230
 strony, 229
 ostrzeżenia, 102

P

paginacja, 126
 pasek nawigacji, 77, 92, 193
 piksel, 22
 plakietyki, 223
 plik a11yhem.js, 169
 pliki Sass, 153
 pływające treści, 220
 podpowiedzi, 144
 podświetlanie tekstu, 214
 porządek elementów flex, 50
 powiadomienia, 188
 prefiksy, 23
 prędkość przewijania, 96
 projekt demo, 14
 przeglądarka, 21
 przekształcenia, 213
 przełączanie widoczności, 212
 przetwarzanie HTML-a, 244
 przewijanie
 dopasowanie prędkości, 96
 przytwierdzanie, 211

R

React, 262
 galeria, 264
 karuzela, 267

referencje, 253
 szablony, 259
 rem, 22
 renderowanie strony, 235
 responsywne
 narzędzia, 58
 obrazy, 54
 rysunki, 120
 rzędy, 40

S

Salvattore, 179, 182, 261
 animate.css, 187
 animowanie siatki, 191
 powiadomienia, 188
 Salvattore Hover, 196
 Sass, 22, 153
 Scrollspy, 93
 sekcje, 61
 selektory grupujące, 234
 skaczące powiadomienia, 188
 skalowanie okna roboczego, 26
 skróty, 122
 stopka, 107
 stosowanie plaketek, 223
 styl
 navbar-dark, 86
 navbar-light, 86
 style, 33
 inline, 230
 paska nawigacji, 93
 system siatkowy, 16, 33
 szablon referencji, 259
 szablony, 24
 szerokość elementu, 217

T

tekst
 podświetlanie, 214
 wyrównanie, 213
 zmniejszanie rozmiaru, 215
 testowanie dyrektywy, 260
 tworzenie
 dyrektywy AngularJS, 257
 formularzy, 109
 Promise z \$q, 256
 projektu, 25

tworzenie
 sekcji, 61
 stopki, 107
 układu, 61
 Typeahead, 142

U

układ, 16, 61
 ukrywanie treści, 225
 UnCss, 246
 urządzenia mobilne, 21
 ustawianie zasobów, 247
 usuwanie
 CSS-u, 242
 komentarzy, 248

V

viewport, 34

W

walidacja formularza, 116
 własna wtyczka jQuery, 166
 właściwość
 alignment, 203
 display, 203
 wrappery, 50
 wskaźniki postępu, 118
 wsparcie urządzeń mobilnych, 21
 wspierane przeglądarki, 21

wtyczka, 123, 147
 All HCM, 166
 dodawanie stylu, 174
 kod, 171
 plik allhcm.js, 169
 bootpag, 123, 126
 DataTables, 139
 ostrzegawcza, 154
 funkcjonalności ostrzeżenia, 158
 kod, 155
 style ostrzeżenia, 157
 Scrollspy, 93
 wtyczki zaawansowane, 179
 wykrywanie przeglądarki, 124
 wypełnienie, 207
 wyrównanie tekstu, 213
 wyskakujące okienka, 144
 wysokość elementu, 217
 wyświetlanie obrazów, 135

Z

zagnieżdżanie, 44
 zakładka, 19, 66
 zapytania mediów, 37
 zasady skracania, 234
 zasoby, 247
 zmniejszanie rozmiaru tekstu, 215

PROGRAM PARTNERSKI

— GRUPY HELION —

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

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

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

GRUPA
Helion 

Bootstrap 4 dla projektanta: responsywność, jakość, styl!

Bootstrap jest coraz popularniejszym frameworkiem CSS i JavaScriptu, służącym do szybko i bezproblemowego tworzenia responsywnych interfejsów WWW. W sierpniu 2011 roku został zaprezentowany światu w jako projekt open source, a dzisiaj jest marką powszechnie znaną wśród projektantów front-endów aplikacji. Bootstrap jest prosty w użyciu, kompatybilny z różnymi przeglądarkami i umożliwia tworzenie interfejsów mobilnych. Zalety te są doceniane przez rosnące grono użytkowników na całym świecie. Mimo tej popularności dość trudno znaleźć dobry materiał do nauki zaawansowanych opcji dostępnych w najnowszej, czwartej wersji Bootstrapa.

To książka przeznaczona dla developerów, którzy chcą pisać kompletne, przyjemne w użytkowaniu i doskonale jakościowo strony za pomocą Bootstrapa 4. Opisano tu różnorakie podejścia do tworzenia stron od podstaw. Dowiesz się, jak budować właściwy układ strony, stosować różne komponenty nawigacyjne Bootstrapa, korzystać z formularzy i stylizować różnego typu treści. Zrozumiesz budowę wtyczki w Bootstrapie, nauczysz się pisać własne komponenty i rozszerzać Bootstrapa za pomocą jQuery. W książce przedstawiono także klasy pomocnicze Bootstrapa, dzięki którym proces tworzenia strony przebiega sprawniej. Nie zabrakło również szczegółowych instrukcji dotyczących optymalizacji strony i jej integrowania z zewnętrznymi frameworkami.

W tej książce między innymi:

- wprowadzenie do Bootstrapa 4 i jego najużyteczniejszych elementów
- tworzenie od podstaw układu strony
- korzystanie z wtyczek i bibliotek oraz tworzenie własnych wtyczek
- klasy pomocnicze Bootstrapa 4, ich implementacja i możliwości wykorzystania
- tworzenie zaawansowanych interfejsów za pomocą akordeonów, rozwijanych list i grup list

Benjamin Jakobus jest informatykiem, specjalizował się w zaawansowanych systemach obliczeniowych. Jako inżynier oprogramowania pracował nad wieloma projektami w Europie i Brazylii.

Jason Marah jest inżynierem VP w Teckro. Pracuje jako full-stack engineer, specjalizuje się w JavaScriptcie i realizuje różne projekty w Europie.

 Helion	<i>Sprawdź nasze szkolenia!</i>	KOD KORZYŚCI Sięgnij po więcej! ▶ 
 helion.pl	SZKOLENIA 	ISBN 978-83-283-4874-5
 HELION SA ul. Kościuszki 1c 44-100 Gliwice tel.: 32 230 98 63 helion@helion.pl	AKADEMIA IT & BUSINESS WWW.SZKOLENIA.HELION.PL	 9 788328 348745
INFORMATYKA W NAJLEPSZYM WYDANIU		Cena: 49,00 zł

Packt