

W PROSTOCIE TKWI SIĘ



# JavaScript dla dzieci

dla  
**bystrzaków**



Zbuduj  
animowanego robota

—  
Stwórz fajne gry

—  
Zrób stronę internetową

—  
Naucz się programować

**Chris Minnick**

**Eva Holland**

założyciele WatzThis?

Tytuł oryginału: JavaScript For Kids For Dummies

Tłumaczenie: Jakub Kwaśniewski

ISBN: 978-83-283-6031-0

Original English language edition Copyright © 2015 by John Wiley & Sons, Inc., Hoboken, New Jersey.  
All rights reserved including the right of reproduction in whole or in part in any form.  
This translation published by arrangement with John Wiley & Sons, Inc.

Oryginalne angielskie wydanie © 2015 by John Wiley & Sons, Inc., Hoboken, New Jersey.  
Wszelkie prawa, włączając prawo do reprodukcji całości lub części w jakiegokolwiek formie, zarezerwowane.  
Tłumaczenie opublikowane na mocy porozumienia z John Wiley & Sons, Inc.

Translation copyright © 2020 by Helion S.A.

Wiley, the Wiley Publishing logo, For Dummies, Dla Bystrzaków, the Dummies Man logo, Making Everything Easier and related trade dress are trademarks or registered trademarks of John Wiley and Sons, Inc. and/or its affiliates in the United States and/or other countries. Used by permission.

JavaScript is a registered trademark of Oracle, Inc.  
All other trademarks are the property of their respective owners.

Wiley, the Wiley Publishing logo, For Dummies, Dla Bystrzaków, the Dummies Man logo, Making Everything Easier i związana z tym szata graficzna są markami handlowymi John Wiley and Sons, Inc. i/lub firm stowarzyszonych w Stanach Zjednoczonych i/lub innych krajach. Wykorzystywane na podstawie licencji. Wszystkie pozostałe znaki handlowe są własnością ich właścicieli.

JavaScript jest zastrzeżonym znakiem towarowym Oracle, Inc.  
Wszystkie pozostałe znaki handlowe są własnością ich właścicieli.

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.

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.

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://dlabystrzakow.pl/user/opinie/jsdzy>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Helion SA

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 231 22 19, 32 230 98 63

e-mail: [dlabystrzakow@dlabystrzakow.pl](mailto:dlabystrzakow@dlabystrzakow.pl)

WWW: <http://dlabystrzakow.pl>

Printed in Poland.

- Kup książkę
- Poleć książkę
- Oceń książkę

- Księgarnia internetowa
- Lubię to! » Nasza społeczność

# Spis treści

---

|                                       |           |
|---------------------------------------|-----------|
| <b>O autorach .....</b>               | <b>15</b> |
| <b>Podziękowania od autorów .....</b> | <b>15</b> |
| <b>Wprowadzenie .....</b>             | <b>17</b> |

|  |           |
|--|-----------|
| <b>CZĘŚĆ I: CZYM JEST JAVASCRIPT? ALERT!<br/>JAVASCRIPT JEST CZYMŚ ŚWIETNYM! .....</b> | <b>21</b> |
|--|-----------|

|  |           |
|--|-----------|
| <b>ROZDZIAŁ 1: Programowanie internetu .....</b> | <b>23</b> |
| Czym jest programowanie? .....                   | 24        |
| Rozmowa z komputerem .....                       | 24        |
| Wybór języka .....                               | 26        |
| Czym jest JavaScript? .....                      | 27        |
| Przygotuj swoją przeglądarkę .....               | 29        |
| Otwieranie narzędzi dla deweloperów .....        | 30        |
| Poznajemy konsolę JavaScriptu .....              | 32        |
| Uruchamianie pierwszych poleceń .....            | 32        |
| Zabawa z matematyką .....                        | 34        |

|  |           |
|--|-----------|
| <b>ROZDZIAŁ 2: Zrozumienie składni .....</b> | <b>35</b> |
| Dokładne precyzowanie intencji .....         | 36        |
| Wydawanie instrukcji .....                   | 37        |
| Przestrzeganie zasad .....                   | 38        |
| Użycie tekstu w napisach .....               | 38        |
| Użycie tekstu w kodzie .....                 | 41        |
| Zwracanie uwagi na znaki niedrukowane .....  | 41        |
| Komentowanie .....                           | 42        |

|                    |  |           |
|--------------------|--|-----------|
| <b>ROZDZIAŁ 3:</b> | <b>Odbieranie i zwracanie danych .....</b> | <b>45</b> |
|                    | Panowanie nad zmiennymi .....              | 46        |
|                    | Tworzenie zmiennych .....                  | 46        |
|                    | Przechowywanie danych w zmiennych .....    | 47        |
|                    | Typy danych .....                          | 49        |
|                    | Napisy .....                               | 49        |
|                    | Liczby .....                               | 51        |
|                    | Typ logiczny .....                         | 51        |
|                    | Wprowadzanie danych wejściowych .....      | 52        |
|                    | Przechowywanie wprowadzonych danych .....  | 53        |
|                    | Odpowiadanie na dane wejściowe .....       | 53        |
|                    | Użycie polecenia alert() .....             | 53        |
|                    | Użycie polecenia document.write() .....    | 55        |
|                    | Łączenie wejścia i wyjścia .....           | 56        |

|                    |   |           |
|--------------------|---|-----------|
| <b>ROZDZIAŁ 4:</b> | <b>Majsterkowanie z aplikacjami internetowymi .....</b> | <b>59</b> |
|                    | Wprowadzenie do JSFiddle .....                          | 60        |
|                    | Wyświetlanie naszych projektów .....                    | 61        |
|                    | Zabawa z projektami .....                               | 62        |
|                    | Majsterkowanie z CSS .....                              | 62        |
|                    | Majsterkowanie z HTML-em .....                          | 67        |
|                    | Majsterkowanie z JavaScriptem .....                     | 68        |
|                    | Tworzenie konta JSFiddle .....                          | 69        |
|                    | Dzielenie się swoim projektem .....                     | 70        |
|                    | Zapisywanie swojej aplikacji .....                      | 71        |

## **CZĘŚĆ II: ANIMOWANIE INTERNETU .....** **73**

|                    |   |           |
|--------------------|---|-----------|
| <b>ROZDZIAŁ 5:</b> | <b>JavaScript i HTML .....</b>                          | <b>75</b> |
|                    | Pisanie w HTML-u .....                                  | 76        |
|                    | Tekst bez HTML-a .....                                  | 76        |
|                    | Użycie HTML-a — znaczniki ponad wszystko .....          | 76        |
|                    | Podstawowa struktura strony internetowej .....          | 78        |
|                    | Tworzenie pierwszej strony internetowej .....           | 79        |
|                    | Poznaj elementy HTML-a .....                            | 80        |
|                    | Dodawanie atrybutów do elementów .....                  | 83        |
|                    | Zmienianie HTML-a przy użyciu JavaScriptu .....         | 84        |
|                    | Wybieranie elementów przy użyciu getElementById .....   | 85        |
|                    | Pobieranie wnętrza elementu przy użyciu innerHTML ..... | 85        |
|                    | Sprawdźmy to: zmiana listy .....                        | 86        |

|                    |   |            |
|--------------------|---|------------|
| <b>ROZDZIAŁ 6:</b> | <b>JavaScript i CSS .....</b>                     | <b>89</b>  |
|                    | Poznaj Douglasa — javascriptowego robota .....    | 90         |
|                    | Podstawy CSS .....                                | 90         |
|                    | Selektory CSS .....                               | 91         |
|                    | Deklaracje CSS .....                              | 92         |
|                    | Właściwości CSS nadają styl .....                 | 92         |
|                    | Użycie kolorów w CSS .....                        | 94         |
|                    | Zmiana rozmiaru elementów w CSS .....             | 95         |
|                    | Zrozumienie kaskadowości .....                    | 98         |
|                    | Umiejscowienie elementów w CSS .....              | 98         |
|                    | Skonfiguruj swojego javascriptowego robota! ..... | 99         |
| <b>ROZDZIAŁ 7:</b> | <b>Budowa animowanego robota .....</b>            | <b>101</b> |
|                    | Zmiana CSS przy użyciu JavaScriptu .....          | 102        |
|                    | Zmieniamy Douglasa przy użyciu JavaScriptu .....  | 103        |
|                    | Eksperymenty na Douglasie .....                   | 104        |
|                    | Roztańcz Douglasa! .....                          | 105        |
|                    | Obsługa zdarzeń .....                             | 106        |
|                    | Tworzenie słuchacza zdarzeń .....                 | 108        |
|                    | Tworzenie animacji w JavaScriptcie .....          | 110        |
|                    | Animowanie kolejnego elementu .....               | 111        |
|                    | Dodanie drugiej funkcji animacji .....            | 112        |

## **CZĘŚĆ III: POZNAWANIE OPERACJI .....** 115

|                    |   |            |
|--------------------|---|------------|
| <b>ROZDZIAŁ 8:</b> | <b>Budujemy auto marzeń, używając operandów .....</b>               | <b>117</b> |
|                    | Poznajemy operandy .....  | 118        |
|                    | Pracujemy z obiektami .....   | 121        |
|                    | Konfigurujemy auto marzeń .....                                     | 122        |
| <b>ROZDZIAŁ 9:</b> | <b>Składanie wszystkiego razem<br/>przy użyciu operatorów .....</b> | <b>127</b> |
|                    | Przedstawiamy Superkalkulator .....                                 | 128        |
|                    | Forkujemy Superkalkulator .....                                     | 128        |
|                    | Używamy Superkalkulatora .....                                      | 129        |
|                    | Superkalkulacje z napisami i operatorami arytmetycznymi .....       | 131        |
|                    | Superkalkulacje z operatorami porównania .....                      | 133        |
|                    | Sztuczki w Superkalkulatorze .....                                  | 137        |

|                     |   |            |
|---------------------|---|------------|
| <b>ROZDZIAŁ 10:</b> | <b>Tworzenie javascriptowej gry słownej .....</b> | <b>139</b> |
|                     | Tworzenie zmiennej historii .....                 | 140        |
|                     | Tworzenie gry słownej .....                       | 140        |
|                     | Pisanie kodu HTML-a .....                         | 142        |
|                     | Dodanie grze słownej stylu .....                  | 144        |
|                     | Pisanie kodu JavaScriptu .....                    | 146        |
|                     | Wykańczanie programu .....                        | 148        |

## **CZĘŚĆ IV: TABLICE I FUNKCJE .....** 153

|                     |  |            |
|---------------------|--|------------|
| <b>ROZDZIAŁ 11:</b> | <b>Tworzenie i zmienianie tablic .....</b> | <b>155</b> |
|                     | Czym są tablice? .....                     | 156        |
|                     | Tworzenie i dostęp do tablic .....         | 157        |
|                     | Przechowywanie różnych typów danych .....  | 157        |
|                     | Pobieranie wartości z tablicy .....        | 157        |
|                     | Używanie zmiennych w tablicy .....         | 157        |
|                     | Zmiana wartości elementów tablicy .....    | 158        |
|                     | Praca z metodami tablicy .....             | 159        |
|                     | Praca z tablicami .....                    | 159        |
|                     | toString() i valueOf() .....               | 160        |
|                     | concat() .....                             | 161        |
|                     | indexOf() .....                            | 162        |
|                     | join() .....                               | 162        |
|                     | lastIndexOf() .....                        | 163        |
|                     | pop() .....                                | 164        |
|                     | push() .....                               | 164        |
|                     | reverse() .....                            | 165        |
|                     | shift() i unshift() .....                  | 166        |
|                     | slice() .....                              | 166        |
|                     | sort() .....                               | 167        |
|                     | splice() .....                             | 167        |
| <b>ROZDZIAŁ 12:</b> | <b>Dodawanie funkcji .....</b>             | <b>169</b> |
|                     | Zrozumienie funkcji .....                  | 170        |
|                     | Funkcje wbudowane .....                    | 170        |
|                     | Własne funkcje .....                       | 170        |
|                     | Poznanie składników funkcji .....          | 171        |
|                     | Definiowanie funkcji .....                 | 171        |
|                     | Nagłówek funkcji .....                     | 172        |
|                     | Wypełnianie ciała funkcji .....            | 172        |
|                     | Wywołanie funkcji .....                    | 172        |

|   |            |
|---|------------|
| Definiowanie parametrów .....                             | 172        |
| Przekazywanie argumentów .....                            | 173        |
| Zwracanie wartości .....                                  | 173        |
| <b>Budujemy „Pociąg do funkcji” .....</b>                 | <b>174</b> |
| Zwiedzanie HTML-a .....                                   | 175        |
| Przechadzka po CSS .....                                  | 175        |
| Pisanie kodu JavaScriptu „Pociągu do funkcji” .....       | 176        |
| Twoja kolej: przedłużanie torów .....                     | 180        |
| <br>  |            |
| <b>ROZDZIAŁ 13: Tworzenie programu listy życzeń .....</b> | <b>181</b> |
| Przedstawienie programu „Lista życzeń” .....              | 182        |
| Rzut oka na kompletny program .....                       | 182        |
| Forkujemy kod .....                                       | 183        |
| Pisanie kodu HTML-a .....                                 | 184        |
| Pisanie kodu JavaScriptu .....                            | 186        |
| Tworzenie procedur obsługi zdarzeń .....                  | 186        |
| Deklaracje zmiennych globalnych .....                     | 187        |
| Pisanie funkcji .....                                     | 188        |
| Otwieranie okienka dialogowego do druku .....             | 196        |
| Ulepszanie listy życzeń .....                             | 197        |

## **CZĘŚĆ V: WOLNOŚĆ WYBORU .....** 199

|   |            |
|---|------------|
| <b>ROZDZIAŁ 14: Podejmowanie decyzji</b>                  |            |
| <b>przy użyciu instrukcji if...else .....</b>             | <b>201</b> |
| Logika boolowska .....                                    | 202        |
| Równość .....   | 202        |
| Różność .....   | 202        |
| Większość i mniejszość .....                              | 202        |
| Więszy niż lub równy i mniejszy niż lub równy .....       | 202        |
| Nie większy niż i nie mniejszy niż .....                  | 203        |
| Poznajemy instrukcję if...else .....                      | 204        |
| Zmienne bez operatorów .....                              | 205        |
| Łączenie porównań przy użyciu operatorów logicznych ..... | 206        |
| Odświeżamy javascriptową pizzerię .....                   | 207        |
| Uruchomienie aplikacji .....                              | 207        |
| Forkowanie kodu .....                                     | 207        |
| Planowanie udoskonaleń .....                              | 208        |
| Dodanie nowej opcji do menu .....                         | 208        |
| Dostawa do innych miast .....                             | 210        |
| Wyświetlenie opłaty za dowóz .....                        | 211        |
| Zaprogramowanie oferty urodzinowej .....                  | 212        |

|                     |  |            |
|---------------------|--|------------|
| <b>ROZDZIAŁ 15:</b> | <b>Wykorzystanie instrukcji switch .....</b>       | <b>217</b> |
|                     | Pisanie instrukcji switch .....                    | 218        |
|                     | Budowanie „Kalendarza aktywności” .....            | 219        |
|                     | Używanie „Kalendarza aktywności” .....             | 220        |
|                     | Forkowanie „Kalendarza aktywności” .....           | 221        |
|                     | Używanie obiektu Date .....                        | 221        |
|                     | Budowanie „Kalendarza aktywności” .....            | 224        |
| <b>ROZDZIAŁ 16:</b> | <b>Napisz swoją własną historię .....</b>          | <b>229</b> |
|                     | Planowanie historii .....                          | 230        |
|                     | Tworzenie schematu blokowego .....                 | 230        |
|                     | Pisanie historii .....                             | 230        |
|                     | Granie w grę .....                                 | 231        |
|                     | Forkowanie kodu .....                              | 232        |
|                     | Przejście przez kod HTML-a i CSS .....             | 232        |
|                     | Chowanie elementów przy użyciu display: none ..... | 234        |
|                     | Patrzmy (lub nie) na części historii .....         | 235        |
|                     | Pisanie kodu JavaScriptu „Misji ratunkowej!” ..... | 236        |
|                     | Tworzenie skrótów do elementów .....               | 237        |
|                     | Tworzenie pustej tablicy .....                     | 237        |
|                     | Tworzenie procedury obsługi zdarzeń .....          | 238        |
|                     | Wywołanie funkcji zadajPytanie() .....             | 238        |
|                     | Pisanie funkcji .....                              | 239        |
|                     | Pisanie funkcji kontynuujHistorie() .....          | 242        |
|                     | Pisanie funkcji koniec() .....                     | 246        |

## **CZĘŚĆ VI: PĘTLE .....** **249**

|                     |                                    |            |
|---------------------|------------------------------------|------------|
| <b>ROZDZIAŁ 17:</b> | <b>Pętla for .....</b>             | <b>251</b> |
|                     | Przedstawiamy pętlę for .....      | 252        |
|                     | Trzy części pętli for .....        | 252        |
|                     | Pisanie i używanie pętli for ..... | 253        |
|                     | Losowa prognoza pogody .....       | 254        |
|                     | Użycie Math.random() .....         | 254        |
|                     | Pisanie aplikacji .....            | 256        |
|                     | Inspekcja wyników .....            | 259        |
|                     | Dodanie stylów do aplikacji .....  | 260        |



|                     |   |            |
|---------------------|---|------------|
| <b>ROZDZIAŁ 18:</b> | <b>Użycie pętli while .....</b>                           | <b>263</b> |
|                     | Pisanie pętli while .....                                 | 264        |
|                     | Wykonanie pętli konkretną liczbę razy .....               | 264        |
|                     | Odliczanie przy użyciu pętli while .....                  | 265        |
|                     | Przechodzenie przez tablicę przy użyciu pętli while ..... | 265        |
|                     | Piszemy „Grę w lunch” .....                               | 266        |
|                     | Forkowanie kodu .....                                     | 266        |
|                     | Pisanie funkcji kupLunche() .....                         | 267        |
|                     | Sprawdzamy działanie .....                                | 269        |
|                     | Przenosimy na Twoją własną stronę .....                   | 269        |
|                     | Zrozumienie hostingu .....                                | 270        |
|                     | Zaczynamy zabawę z cba.pl .....                           | 270        |
| <b>ROZDZIAŁ 19:</b> | <b>Budowanie stoiska z lemoniadą .....</b>                | <b>275</b> |
|                     | Najpierw gramy .....                                      | 276        |
|                     | Lekcja biznesu .....                                      | 277        |
|                     | Wypracowanie zysku .....                                  | 278        |
|                     | Zrozumienie potrzeb klienta .....                         | 278        |
|                     | Zrozumienie matematyki .....                              | 278        |
|                     | Powiązanie sprzedaży, temperatury i ceny .....            | 279        |
|                     | Budujemy grę .....  | 281        |
|                     | Forkujemy kod .....                                       | 282        |
|                     | Piszemy JavaScript .....                                  | 282        |
|                     | Tworzymy zmienne globalne .....                           | 282        |
|                     | Generujemy pogodę .....                                   | 284        |
|                     | Otwieramy stoisko .....                                   | 287        |
|                     | Resetowanie programu .....                                | 289        |
|                     | Wyświetlanie raportu .....                                | 290        |
|                     | Kończenie i testowanie programu .....                     | 291        |
|                     | Ulepszanie stoiska z lemoniadą .....                      | 295        |
|                     | <b>Skorowidz .....</b>                                    | <b>297</b> |



## Rozdział 4

# Majsterkowanie z aplikacjami internetowymi

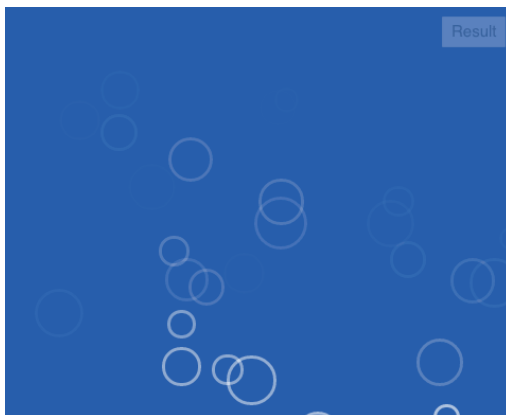
**W** rozdziale 1. tłumaczymy, jak używać konsoli JavaScriptu. W rozdziałach 2. i 3. pokazujemy, jak połączyć ze sobą kilka instrukcji, żeby utworzyć program. W tym rozdziale nasza zabawa nabierze rumieńców, przedstawimy Ci bowiem **JSFiddle** — nasz ulubiony plac zabaw. Zamiast huśtawek i zjeżdżalni będziesz się na nim bawił instrukcjami języka JavaScript, znacznikami HTML-a i stylami CSS.

JSFiddle pozwala na pisanie kodu JavaScriptu i eksperymentowanie z nim bezpośrednio w przeglądarce internetowej. Umożliwia on sprawdzenie działania dowolnego kodu, dzielenie się swoim kodem z innymi i wyrażanie opinii na jego temat, a nawet współpracę nad programem ze znajomymi! Dowiesz się również, jak używać JSFiddle, aby wyświetlać, zmieniać, zapisywać i udostępniać całe aplikacje internetowe.

Pewnie się zastanawiasz, co mamy na myśli, pisząc o **aplikacji internetowej**. Aplikacja internetowa (ang. *web application* lub *web app*) to oprogramowanie, które jest uruchamiane w przeglądarce internetowej i zazwyczaj używa języka JavaScript. Jedną z takich aplikacji, które możesz znać, jest np. Google Earth. Umożliwia ona znalezienie i zobaczenie wysokiej jakości zdjęć niemal każdego miejsca na Ziemi. Google Earth jest również *stroną internetową*, bo można się do niej dostać za pomocą adresu URL. Jak myślisz, czy JSFiddle jest aplikacją, czy

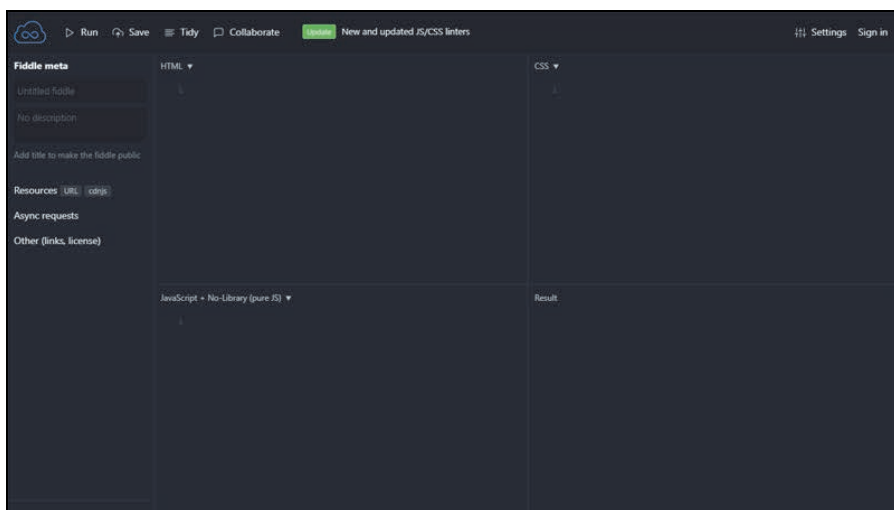
stroną internetową? Tak naprawdę i jednym, i drugim. Właściwie każda aplikacja internetowa jest stroną internetową, ale nie każda strona internetowa to aplikacja internetowa.

W tym rozdziale użyjesz JSFiddle, aby poeksperymentować z animacjami. Dzięki temu otrzymasz w pełni konfigurowalną maszynę do puszczenia baniek! Nazwa strony to JSFiddle, bo pozwala ona na „majsterkowanie” (ang. *to fiddle*) z językiem JavaScript. A więc zaczynamy!



## Wprowadzenie do JSFiddle

Aby zacząć zabawę z JSFiddle, otwórz swoją przeglądarkę internetową i wpisz adres <http://jsfiddle.net>. Zobaczysz stronę JSFiddle, taką jak na rysunku 4.1.



**RYSUNEK 4.1.**

Przejrzysty i ustrukturyzowany interfejs JSFiddle



Interfejs użytkownika strony JSFiddle składa się z trzech paneli, w których można pisać kod w językach HTML, CSS i JavaScript, a także z panelu *Result* (wynik), w którym można zobaczyć wynik działania instrukcji wprowadzonych w pozostałych panelach. Na pasku narzędzi po lewej stronie można skonfigurować dodatkowe opcje, kilka ciekawych możliwości ukrytych jest też pod przyciskiem *Settings* (ustawienia) na górnym pasku narzędzi — ten zawiera również przyciski do uruchamiania, zapisywania i porządkowania kodu.

Możesz zmienić rozmiar każdego z paneli w JSFiddle, łapiąc za oddzielające je linie i przeciągając je.

W tej chwili jesteśmy najbardziej zainteresowani panelem JavaScriptu. Działa on w bardzo podobny sposób jak konsola JavaScriptu, w JSFiddle jednak o momencie uruchomienia kodu, który piszesz, decydujesz sam.

Aby uruchomić swój pierwszy program w JSFiddle:

1. Kliknij wewnątrz panelu JavaScriptu.
2. Wpisz następującą instrukcję JavaScriptu:

```
alert("Cześć wszystkim!");
```

3. Kliknij przycisk *Run* (uruchom) na górnym pasku narzędzi. Pojawi się wyskakujące okienko z tekstem „Cześć wszystkim!”.
4. Zamknij okienko, klikając *OK*.

Nie ma nic zaskakującego w działaniu tego prostego programu. Jeśli przeczytałeś trzy pierwsze rozdziały tej książki, wiesz już, jak działa instrukcja `alert`.

Uruchamianie kodu JavaScriptu to nie jedyna świetna funkcjonalność JSFiddle. W panelach HTML-a i CSS można tworzyć kod współpracujący z tym napisanym w JavaScriptcie. W kilku następnych sekcjach zajmiemy się każdym z tych paneli osobno i pokażemy, jak ich używać. Ale wcześniej zobacz, jakie jeszcze możliwości ma JSFiddle.

## Wyświetlanie naszych projektów

Zdradzimy Ci pewien sekret. Możesz zobaczyć, uruchomić, skopiować i zmienić każdy program z tej książki, wchodząc na stronę [https://jsfiddle.net/user/Helion\\_JavaScript/fiddles](https://jsfiddle.net/user/Helion_JavaScript/fiddles). Naprawdę: cały kod gotowy do użycia, ładnie sformatowany i przetestowany!

To nasza własna, publiczna tablica JSFiddle. Publiczna tablica to miejsce, za pomocą którego każdy użytkownik JSFiddle może dzielić się swoimi programami (w nomenklaturze JSFiddle określanymi jako „fiddle”) ze światem.



Chociaż wszystkie nasze projekty są już gotowe i dostępne w JSFiddle, ważne, żebyś sam przeszedł przez każdy kolejny krok podczas ich tworzenia, by je dobrze zrozumieć. Jeśli chcesz jak najlepiej wykorzystać tę książkę, nie wahaj się kopiować, modyfikować, całkowicie zmieniać czy przepisywać nasz kod, aby zobaczyć, jak wpłynie to na wyniki. Majsterkuj do woli!

## Zabawa z projektami

Zanim zaczniesz przeglądać wszystkie projekty z innych części i rozdziałów tej książki, rzuć okiem na niektóre programy, które nie są jej częścią. JSFiddle pozwala każdemu na założenie konta i dzielenie się swoimi programami na publicznej tablicy — i wielu świetnych, doświadczonych programistów JavaScriptu to robi!



Kiedy programiści decydują się podzielić swoimi projektami na JSFiddle, godzą się na to, że każdy może skopiować ich kod, zmienić go i opublikować ponownie. Jednak zawsze mile widziane będzie powiadomienie, kto jest autorem oryginalnego rozwiązania. My zrobiliśmy kopie wszystkich naszych projektów, aby mieć pewność, że będą one zawsze spójne z zawartością książki. Jeśli chcesz się dowiedzieć, kto jest oryginalnym autorem programu, kliknij zakładkę *Author* na pasku narzędzi po lewej stronie.

Postępuj zgodnie z poniższymi krokami, aby obejrzeć i uruchomić niesamowite programy z naszej publicznej tablicy JSFiddle:

1. Przejdź do naszej tablicy na stronie [https://jsfiddle.net/user/Helion\\_JavaScript/fiddles](https://jsfiddle.net/user/Helion_JavaScript/fiddles).

Zobaczysz listę przykładów i projektów z całej książki.

Być może będziesz musiał użyć stronicowania poniżej listy, aby zobaczyć więcej wyników.

2. Znajdź projekt, który wydaje Ci się ciekawy, i otwórz go.

Kiedy program się otworzy, automatycznie się uruchomi.

Jeśli znajdziesz program, który Ci się podoba, spróbuj odkryć, jak działa. Zmień niektóre wartości i zobacz, co się stanie.



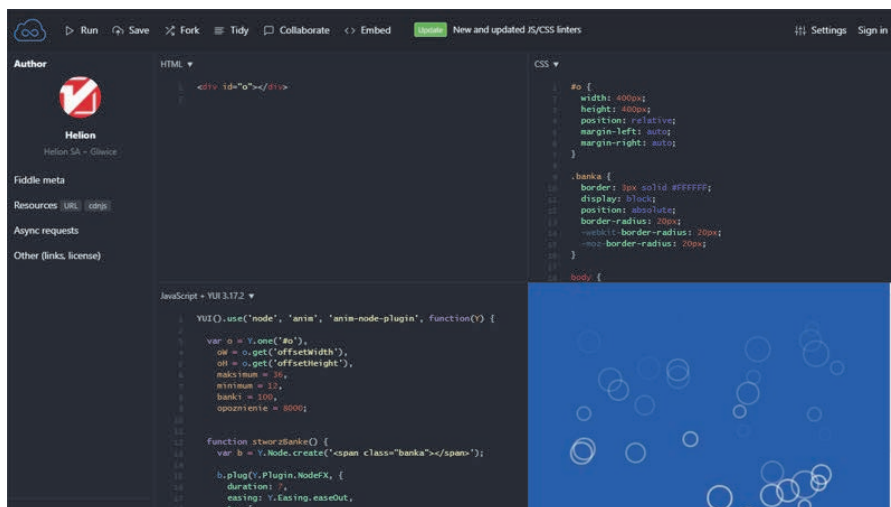
Twoje zmiany w projekcie JSFiddle nie nadpiszą oryginału! Możesz zmienić wszystko, a inni czytelnicy będą wciąż widzieli oryginalną wersję kodu. Najgorsze, co może się wydarzyć, to że po Twoich zmianach program się nie uruchomi.

## Majsterkowanie z CSS

Panel CSS w JSFiddle jest umieszczony w prawym górnym rogu. Poza pracą z JavaScriptem, w JSFiddle możemy również bawić się **CSS** (ang. *Cascading Style Sheets* — kaskadowe arkusze stylów). CSS pozwala na zmianę wyglądu takich elementów strony jak tekst czy grafika. Jeśli chcesz zmienić kolor napisu na stronie, używasz do tego właśnie CSS.

Bardziej szczegółowo zajmiemy się CSS w rozdziale 6. Teraz wprowadzimy tylko kilka zmian w jednym z naszych programów. Wykonaj następujące kroki:

1. Przejdź do strony [https://jsfiddle.net/Helion\\_JavaScript/L8br6uos](https://jsfiddle.net/Helion_JavaScript/L8br6uos). Zobaczysz projekt „Bańki”, jak na rysunku 4.2.



**RYSUNEK 4.2.**  
Projekt „Bańki”

2. Przyjrzyj się dokładnie czterem panelom na ekranie. Trzy z nich zawierają kod, a w czwartym wyświetlana jest animacja z bańkami. Czy domyślasz się, czytając kod, za co on odpowiada?
3. Spójrz na panel CSS (w prawym górnym rogu). Widzisz trzy linie kodu.
4. Znajdź kod o treści `border: 3px solid #FFFFFF;` i zmień go na `border: 8px solid #FFFFFF;`.
5. Kliknij przycisk *Run* na górnym pasku narzędzi, aby rozpocząć animację ponownie. Zauważysz, że krawędzie baniek są teraz znacznie grubsze, jak na rysunku 4.3.



**RYSUNEK 4.3.**  
Krawędzie baniek zrobiły się grubsze

Biorąc pod uwagę Twoją zmianę w kodzie i jej efekt, jak myślisz, co w tej samej instrukcji opisuje słowo `solid`? Aby się przekonać, wykonaj następujące kroki:

1. W panelu CSS zmień pierwszą wartość po `border:` na mniejszą liczbę (2 lub 3) i kliknij *Run*.  
Krawędzie baniek są znów cienkie.
2. Zmień drugą wartość po `border:` na jedną z poniższych:

```
dotted  
dashed  
double  
groove  
ridge  
inset  
outset
```

3. Kliknij *Run*, aby zobaczyć, co to zmienia.  
Ta wartość mówi przeglądarce o stylu krawędzi. Rysunek 4.4 przedstawia bańki, których styl krawędzi został ustawiony na `dotted`.



**RYSUNEK 4.4.**  
Bańki  
z wykropkowanymi  
krawędziami


Spójrz teraz na trzecią wartość po `border:`, która jest obecnie ustawiona na `#FFFFFF`. Ten zestaw znaków oznacza kolor krawędzi baniek.

Kolory w CSS zapisywane są zazwyczaj w tzw. **notacji szesnastkowej**, w której używamy trzech wartości z przedziału od 00 do FF na określenie ilości kolorów czerwonego, zielonego i niebieskiego.


Więcej szczegółów na temat kolorów w CSS zawiera rozdział 6. Możesz również używać wielu standardowych, angielskich nazw kolorów. Tabela 4.1 zawiera zestawienie najczęściej występujących nazwanych kolorów rozumianych przez przeglądarki internetowe.



**TABELA 4.1.** Standardowe nazwy kolorów

| Nazwa koloru        | Wartość w notacji szesnastkowej   |
|---------------------|---|
| Aqua (cyjan)        | #00FFFF    |
| Black (czarny)      | #000000    |
| Blue (niebieski)    | #0000FF    |
| Fuchsia (fukcja)    | #FF00FF    |
| Gray (szary)        | #808080    |
| Green (zielony)     | #008000   |
| Lime (limonkowy)    | #00FF00  |
| Maroon (kasztanowy) | #800000  |
| Navy (granatowy)    | #000080  |

**TABELA 4.1.** Standardowe nazwy kolorów — ciąg dalszy

| Nazwa koloru          | Wartość w notacji szesnastkowej   |
|-----------------------|---|
| Olive (oliwkowy)      | #808000    |
| Orange (pomarańczowy) | #FFA500    |
| Purple (purpurowy)    | #800080    |
| Red (czerwony)        | #FF0000    |
| Silver (srebrny)      | #C0C0C0    |
| Teal (morski)         | #008080   |
| White (biały)         | #FFFFFF  |
| Yellow (żółty)        | #FFFF00  |

Aby zmienić kolor baniek, wykonaj następujące kroki:

1. Wybierz nazwę koloru lub jego kod w notacji szesnastkowej z tabeli 4.1.
2. W panelu CSS zamień dotychczasową wartość #FFFFFF na tę wybraną przez siebie.
3. Kliknij *Run*.

W panelu *Result* zobaczysz teraz bańki w wybranym przez siebie kolorze.

## Majsterkowanie z HTML-em

Teraz rzuć okiem na panel HTML-a, znajdujący się w lewym górnym rogu. W porównaniu z panelami CSS i JavaScriptu dzieje się tutaj naprawdę niewiele!

W języku HTML, który omawiamy szerzej w rozdziale 5., tworzymy strukturę strony internetowej. W przypadku projektu „Bańki” za pomocą HTML-a po prostu stworzyliśmy miejsce na stronie, w którym mają być puszczane bańki.

Ale z HTML-em możesz zrobić dużo więcej! Zmień kod HTML-a projektu „Bańki”, wykonując następujące kroki:

1. Umieść kursor zaraz za znacznikiem `</div>` i wpisz:

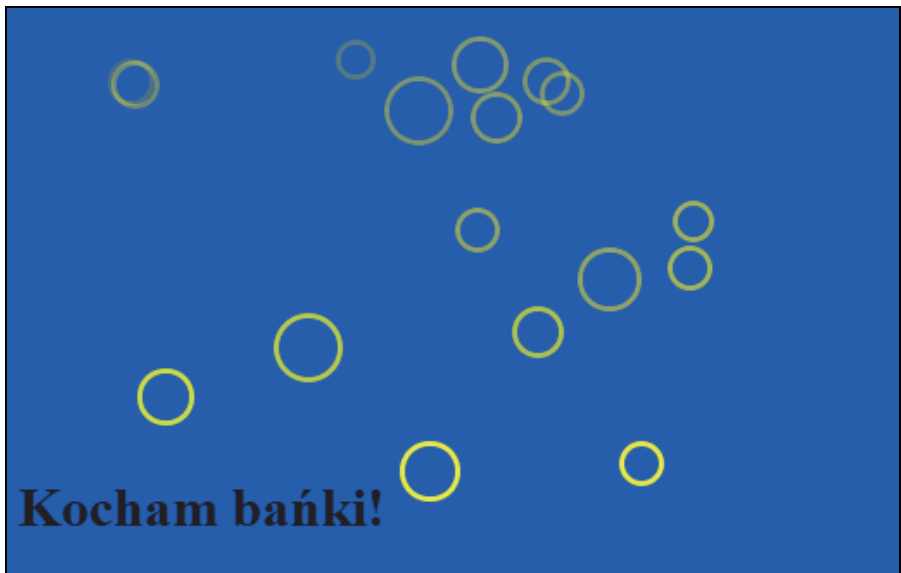
```
<h1>Kocham bańki!</h1>
```

Twój panel HTML-a powinien teraz zawierać następujący kod:

```
<div id="o"></div><h1>Kocham bańki!</h1>
```

2. Kliknij *Run* i zobacz zmiany w panelu *Result*.

Jak widać na rysunku 4.5, pod naszymi bańkami pojawił się teraz specjalny napis.



**RYСУNEK 4.5.**  
Kolorowe bańki ze specjalnym napisem

`<h1>` i `</h1>` to tzw. **znaczniki HTML-a**. Znaczniki okalające tekst informują przeglądarkę, że tekst zawiera coś szczególnego. W tym wypadku znacznik `<h1>` reprezentuje nagłówek najwyższego poziomu — największy wizualnie i najważniejszy nagłówek na stronie.

Kolejnym użytecznym znacznikiem HTML-a jest `<p>`, którym oznaczamy akapity. Aby wstawić znacznik `<p>`, wykonaj następujące kroki:

1. Za znacznikiem `</h1>` naciśnij *Enter*, aby przejść do nowej linii.
2. Rozpocznij akapit znacznikiem otwierającym `<p>`, następnie wpisz, co tylko chcesz, w jego treści i zakończ go znacznikiem zamykającym `</p>`.
3. Kliknij *Run* i zobacz zmiany w panelu *Result*.

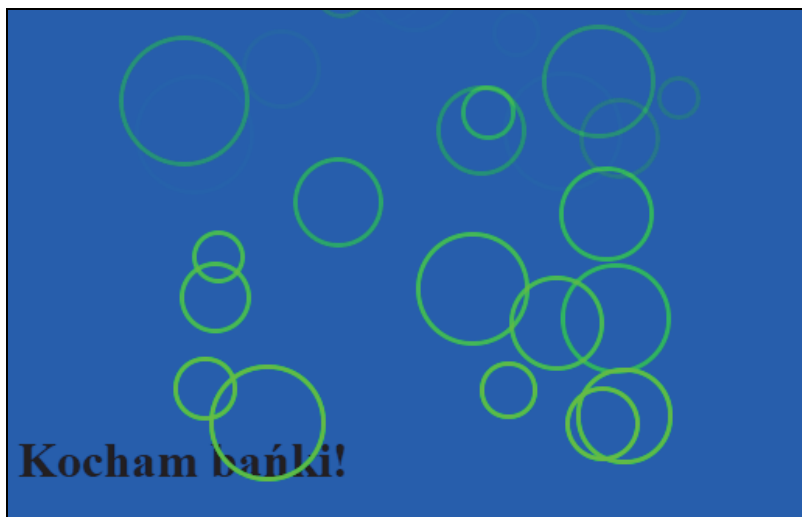
## Majsterkowanie z JavaScriptem

Panel JavaScriptu, umieszczony w lewym dolnym rogu interfejsu JSFiddle, to właśnie to miejsce, w którym dzieją się naprawdę ciekawe rzeczy.

1. Znajdź w panelu JavaScriptu linię o treści `maksimum = 36` i zmień ją na `maksimum = 80`.
2. Kliknij *Run*.  
Sporo baniek, choć nie wszystkie, jest teraz większych niż przedtem.

Czy bazując na wyniku poprzedniej zmiany, jesteś w stanie zgadnąć, co się stanie, gdy zmienisz linię kodu o treści `minimum = 12`? Spróbuj i zobacz, czy masz rację!

Jeśli myślałeś, że wartość `maksimum` oznacza maksymalną wielkość bańki, a `minimum` jej wielkość minimalną, miałeś absolutną rację. Rysunek 4.6 przedstawia zawartość panelu *Result* przy wartości `maksimum` ustawionej na 80, a `minimum` na 20.



**RYSUNEK 4.6.**  
Zmiany wartości minimum i maksimum wpływają na rozmiar baniek

Dwie następne linie kodu w panelu JavaScriptu to `banki = 100` i `opoznienie = 8000`. Możesz zmienić te wartości, wykonując te same kroki co zazwyczaj, czyli:

1. Zmień wartość.
2. Kliknij *Run* i zobacz rezultat wprowadzenia zmiany w panelu *Result*.

Spróbuj zmienić obie wartości i zobacz, co się stanie. Wypróbowując różne wartości (albo po prostu zgadując), odkryjesz, że wartość banki oznacza liczbę tworzonych baniek, a opóźnienie ma wpływ na ich prędkość.

Czy eksperymentując, jesteś w stanie poznać dokładne znaczenie wartości banki i opóźnienie? Mała odpowiedź: wartość opóźnienie jest podana w milisekundach (czyli tysięcznych częściach sekundy). 8000 milisekund to 8 sekund. Zmień wartość na 10000, kliknij *Run* i zwróć uwagę na chronologię poszczególnych akcji na ekranie. Potem zmień ją na 1000 i uruchom program ponownie, klikając *Run*.

Jeśli podejrzewasz już, że banki to liczba baniek, które zostaną utworzone, a opóźnienie dotyczy tego, jak szybko się one pojawiają, masz rację!

## Tworzenie konta JSFiddle

---

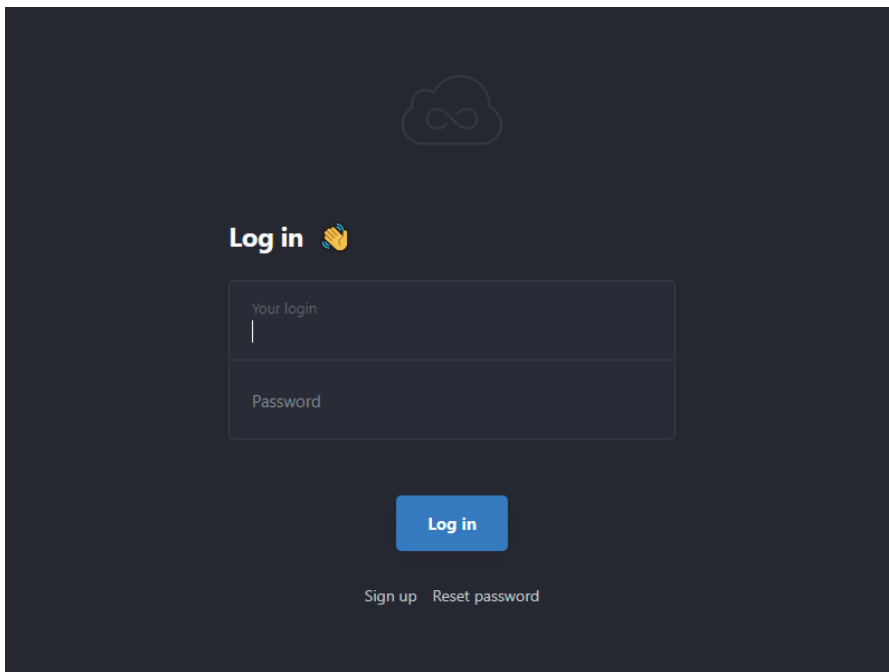
Tworzenie konta JSFiddle nie jest wymagane do dalszej lektury tej książki, jednak posiadanie go ułatwi wyświetlanie przykładów i dzielenie się swoją pracą.

Aby utworzyć konto JSFiddle, postępuj zgodnie z następującymi krokami:

1. Kliknij przycisk *Fork* na górnym pasku narzędzi i w wyskakującym okienku kliknij przycisk *Fork fiddle*.  
Kiedy tworzysz swoją wersję programu na bazie cudzego kodu, operację taką nazywa się **forkowaniem**<sup>1</sup> kodu.
2. Zaznacz adres URL swojego programu na pasku adresu przeglądarki internetowej i skopiuj go lub zapisz, abyś mógł go użyć po utworzeniu konta.
3. Kliknij przycisk *Sign in* (zaloguj się) znajdujący się w prawym górnym rogu strony. Pojawi się strona logowania, jak na rysunku 4.7.
4. Kliknij przycisk *Sign up* (zarejestruj się) znajdujący się pod formularzem logowania. Pojawi się strona rejestracji.
5. Wypełnij formularz tworzenia nowego konta, podając swoją nazwę użytkownika (*username*), adres e-mail (*e-mail address*) i hasło (*password*), a następnie kliknij przycisk *Create an account* (utwórz nowe konto).  
Pojawi się strona główna JSFiddle.
6. Wklej lub przepisz adres URL z kroku 2. do paska adresu przeglądarki internetowej i naciśnij *Enter*.  
Wrócisz do swojej wersji projektu „Bańki”.

---

<sup>1</sup> W również spotykanym tłumaczeniu dosłownym: *rozwidlaniem* — *przyp. tłum.*



**RYSUNEK 4.7.**  
Strona logowania  
JSFiddle

7. Kliknij ponownie *Fork*, aby dodać swoją wersję projektu do swojego konta JSFiddle. Zauważ, że teraz adres URL na pasku adresu przeglądarki zawiera Twoją nazwę użytkownika JSFiddle!

## Dzielenie się swoim projektem

Kiedy stworzyłeś już swoją własną, spersonalizowaną wersję projektu „Bańki”, pora pokazać ją znajomym!

Najprostszym sposobem na zademonstrowanie innym swojego projektu jest skopiowanie adresu URL z paska adresu przeglądarki — możesz go wrzucić na Facebook lub Twitter. Każdy będzie mógł zobaczyć Twój kod CSS, HTML-a i JavaScriptu oraz wynik jego działania, a także na jego bazie zacząć tworzyć swoją wersję projektu (nie zmieniając jednak Twojej — podobnie jak Ty, tworząc swoją wersję, nie zmieniłeś naszej).



Jeśli zamierzasz umieścić adres do swojej wersji projektu na Facebooku lub Twitterze, pamiętaj, żeby oznaczyć nas w poście (@watzthisco na Twitterze lub [www.facebook.com/watzthisco](http://www.facebook.com/watzthisco) na Facebooku), a zobaczymy Twoje dzieło!

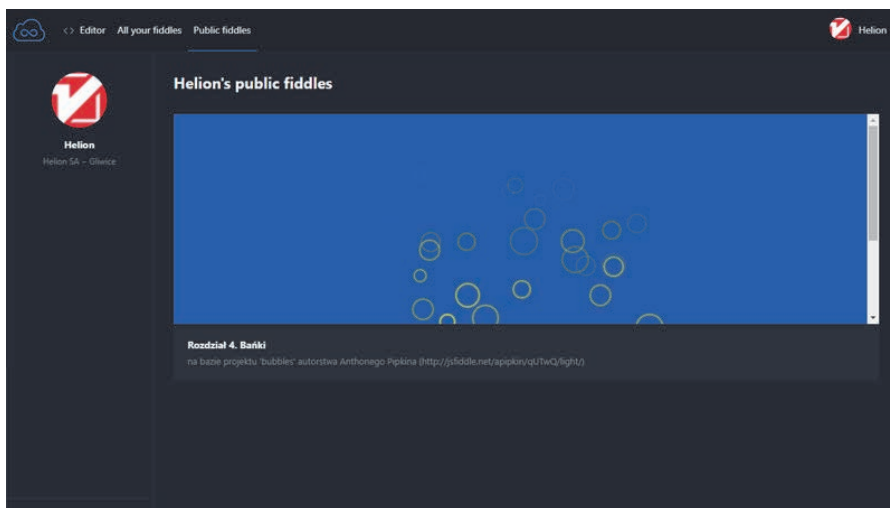
Możesz też udostępnić sam panel z wynikiem swojego projektu, bez kodu, dodając `/show` do jego adresu. Na przykład sam panel z wynikiem naszego projektu „Bańki” jest dostępny pod adresem [https://jsfiddle.net/Helion\\_JavaScript/L8br6uos/show](https://jsfiddle.net/Helion_JavaScript/L8br6uos/show).

Jeśli chcesz wrócić do oryginalnej wersji projektu „Bańki”, możesz to zrobić, korzystając z naszej publicznej tablicy pod adresem [https://jsfiddle.net/user/Helion/\\_JavaScript/fiddles](https://jsfiddle.net/user/Helion/_JavaScript/fiddles).

A co, jeśli chciałbyś mieć własną publiczną tablicę JSFiddle, zawierającą kolekcję Twoich projektów? Czytaj dalej!

## Zapisywanie swojej aplikacji

Teraz, kiedy masz już swoje konto JSFiddle, możesz utworzyć swoją publiczną tablicę. A właściwie nawet już ją masz! Możesz się o tym przekonać, klikając swój awatar w prawym górnym rogu ekranu i wybierając opcję *Your public fiddles* (twoje publiczne projekty). Wyświetli się Twoja publiczna tablica, z Twoją wersją projektu „Bańki”. Powinna wyglądać podobnie do tej na rysunku 4.8.



**RYСУNEK 4.8.**  
Twoja własna publiczna tablica na JSFiddle

Kiedy dodałeś własną wersję projektu do swojego konta JSFiddle (forkując naszą), trafiła ona od razu na Twoją publiczną tablicę. Stało się tak dlatego, że w JSFiddle za publiczne uznawane są wszystkie projekty, które posiadają tytuł. Jako że nasz projekt go posiadał, Twój po jego skopiowaniu również takowy zyskał!

Możesz zmienić tytuł projektu na inny lub go usunąć, tym samym usuwając projekt ze swojej publicznej tablicy. Aby to zrobić, postępuj zgodnie z następującymi krokami:

1. Kliknij projekt „Bańki” na swojej publicznej tablicy.
2. Przejdź do zakładki *Fiddle meta* (metadane projektu) na lewym pasku narzędzi.
3. Zmień lub usuń tekst z pierwszego pola tekstowego — to właśnie tytuł projektu.
4. Kliknij przycisk *Save* (zapisz) na górnym pasku narzędzi.
5. Kliknij przycisk *Set as base* (ustaw jako wersję bazową) na górnym pasku narzędzi.

Za każdym razem kiedy zapisujesz projekt, JSFiddle tworzy jego nową wersję (możesz to zaobserwować po zmieniającym się adresie URL). Przycisk *Set as base* pozwala na ustawienie obecnie wyświetlanej w JSFiddle wersji projektu jako bazowej. To właśnie do wersji bazowej prowadzić będą adresy URL na Twojej publicznej tablicy, jak również na liście wszystkich Twoich projektów. Twój projekt znajdzie się na Twojej publicznej tablicy, jeśli jego wersja bazowa (niekoniecznie ostatnia) będzie miała nadany tytuł.



# Skorowidz

## A

akapit, 77, 81  
anchor, *Patrz:* kotwica  
animacja, 60, 105, 110, 111  
    klatka, *Patrz:* klatka  
aplikacja internetowa, 59  
atrybut, 83  
    alt, 83  
    class, 91, 98, 257, 259, 260, 261  
    id, 83, 84, 91, 98, 142, 257, 259, 260  
    src, 83  
    type, 142

## B

bajt, 25  
bit, 25  
błąd, 189, 190  
    składni, 33  
Boole George, 51  
branching, *Patrz:* rozgałęzianie

## C

case, 217, 218  
central processing unit, *Patrz:* procesor  
check box, *Patrz:* przycisk wyboru  
Chrome, 28  
    narzędzia dla deweloperów, 30  
    Console, *Patrz:* Console  
    zakładki, 31  
Console, 31, 32  
CPU, *Patrz:* procesor

CSS, 61, 62, 89, 123, 233  
    deklaracja, 92  
    kolor, 64, 66, 94, 95  
    organizacja kodu, 144  
    reguła, 90  
        blok deklaracji, 90  
        kaskadowość, 98  
        selektor, *Patrz:* selektor  
    właściwość, 92, 102  
czas, 222, 223

## D

dane, 47  
    typ, *Patrz:* typ  
    wejściowe, 45, 56  
        wprowadzanie, 52  
    wyjściowe, 45, 56  
data, 222, 223  
debugowanie, 25  
dochód, 278  
dokument, 55, *Patrz też:* strona

## E

Edge, 29  
element, *Patrz też:* znacznik  
    a, 81  
    body, 78, 79, 91  
    button, 86, 184  
    div, 81, 142, 184  
    em, 81  
    h1, 79  
    head, 78, 79

- hr, 81
- html, 79
- img, 81
- input, 142
- li, 79, 81, 142
  - dodawanie, 192
- ol, 79, 81
- p, 81
- pozycjonowanie, 98, 103
- rozmiar, 95
- span, 147
- strong, 81
- ul, 79, 81, 184, 192
- wybieranie, 85

em, 93

event, *Patrz:* zdarzenie

## F

- Facebook, 28
- Firefox, 29
- formularz, 107, 182, 184, 193
- funkcja, 108, 188, *Patrz też:* metoda
  - addEventListener, 170
  - alert, 171
  - argument, 173
  - ciało, 108, 172
  - definiowanie, 171, 172
  - getElementById, 170
  - indexOf, 170
  - Math.floor, 255
  - Math.random, 254
  - nagłówek, 172
  - parametr, 172, 173
    - argument, 173
  - pusta, 190
  - toString, 170
  - tworzenie, 170, 171, 172
  - wartość zwracana, 173
  - wbudowana, 170
  - wywołanie, 172
- funkcji, 169

## G

- getter, 222
- gigabajt, 25, 26
- Google Chrome, *Patrz:* Chrome
- Google Earth, 59
- gra słowna, 139, 140, 141

## H

- hipertekst, 76
- Hopper Grace, 25
- hosting, 270
- HTML, 61, 67, 75, 76, 233
  - element, 77
  - formularz, *Patrz:* formularz
  - znacznik, *Patrz:* znacznik

## I

- indeksowanie od zera, 50
- inkrementacja, 253
- input, *Patrz:* dane wejściowe
- instrukcja, 37
  - alert, 53
  - break, 218, 226, 240
  - document.write, 55
  - for, 42
  - if, 204
  - if...else, 201, 204, 242, 268
  - operand, *Patrz:* operand
  - operator, *Patrz:* operator
  - prompt, 52, 53
  - switch, 218, 240, 242
    - case, *Patrz:* case
- internet, 27
- Internet Explorer, 29

## J

- JavaScript, 27, 61, 68, 75
  - konsola, *Patrz:* Console
  - składnia, 35
    - wcięcia, 42
    - wielkość liter, 41

właściwość, *Patrz:* właściwość  
jednostka centralna, *Patrz:* procesor  
język

naturalny, 35, 37, 117  
programowania, 26  
CSS, *Patrz:* CSS  
HTML, *Patrz:* HTML  
instrukcja, *Patrz:* instrukcja  
składnia, 35  
znaczników, 76  
JSFiddle, 59, 60, 269  
interfejs użytkownika, 61  
konto, 62  
tworzenie, 69  
ograniczenia, 270  
panel  
CSS, 61, 62, 63  
HTML, 61, 67  
JavaScript, 61, 68  
Result, 61, 96  
standaryzowanie formatowania, 185  
tablica publiczna, 61, 62, 71, 103

## K

kalendarz, 220  
kilobajt, 25, 26  
klatka, 110  
klient FTP, 272  
kod  
binarny, 24, 26  
szesnastkowy, 95  
kodowanie, *Patrz:* programowanie  
komentarz, 42  
jednowierszowy, 43  
todo, 177  
wielowierszowy, 43  
kompilator, 25, 26  
komunikat błędu  
NaN, 34  
SyntaxError, 33  
konkatenacja, 33  
koszty, 278  
kotwica, 81

## L

liczba, 49, 51, 118  
losowa, 254  
postać wykładnicza, 133  
lista  
element, 79  
nieuporządkowana, 79, 81, 142  
uporządkowana, 78, 79, 81  
listener, *Patrz:* zdarzenie słuchacz  
Lovelace Ada, 25

## M

megabajt, 25, 26  
metoda, 50, 121, *Patrz też:* funkcja  
addEventListener, 106, 107, 146, 187,  
238  
appendChild, 192  
concat, 159, 161, 162  
getDate, 222  
getDay, 222  
getElementById, 84, 85, 86, 102  
getFullYear, 222  
getHours, 222  
getMilliseconds, 222  
getMonth, 222  
getSeconds, 222  
getTime, 222  
indexOf, 50, 159, 162  
join, 159, 162  
lastIndexOf, 159, 163  
pop, 159, 164, 166  
print, 196  
push, 159, 164  
reverse, 159, 165  
setDate, 223  
setDay, 223  
setFullYear, 223  
setHours, 223  
setMilliseconds, 223  
setMonth, 223  
setSeconds, 223  
setTime, 223  
shift, 159, 166

slice, 159, 166  
sort, 159  
splice, 159, 167  
toString, 159, 160, 161  
unshift, 159  
valueOf, 161  
write, 55

## N

nagłówek h1, 79  
napis, 38, 40, 49, 118, 137  
  długość, 49  
  pusty, 49  
  sekwencja zmodyfikowana, 40  
notacja  
  camelCase, 46, 102  
  naukowa, 133  
  szesnastkowa, 64, 66, 94, 95

## O

obiekt, 55, 121  
  Date, 221, 222  
  metoda, 222, 223  
  metoda, 121, 170  
  tworzenie, 121, 124  
  window, 196  
  właściwość, 121  
obrazek, 81  
Opera, 29  
operacja modulo, 128, 132  
operand, 117, 118, 130  
  typ, 119  
operator, 117, 118, 127  
  „i”, 206  
  „lub”, 206  
  arytmetyczny, 130, 132  
  identyczności, 134, 202  
  konkatenacji, 128, 131, 139  
  logiczny, 206  
  mniejszości, 202  
  modulo, 132  
  napisowy, 130  
  nieidentyczności, 135, 202

porównania, 128, 130, 133, 136, 204  
przypisania, 53  
równości, 134, 202  
różności, 135, 202  
większości, 202  
oprogramowanie, 24  
output, *Patrz:* dane wyjściowe

## P

petabajt, 26  
pętla  
  for, 38, 252, 253, 254, 264  
  while, 263, 264, 265  
piksel, 93, 96  
plik dźwiękowy, 77  
pole wejściowe, 142, 144  
polecenie, *Patrz:* instrukcja  
procedura obsługi zdarzeń, 106  
procesor, 24  
program, 24  
programista, 24, 36  
programowanie, 24  
property, *Patrz:* właściwość  
prymityw, 48, *Patrz też:* typ  
przeglądarka, 29  
  Chrome, *Patrz:* Chrome  
  Firefox, *Patrz:* Firefox  
przychód, 278  
przycisk  
  opcji, 129  
  wyboru, 129

## R

radio button, *Patrz:* przycisk opcji  
rozgałęzianie, 229

## S

schemat blokowy, 230  
selektor, 90, 91  
  elementu, 91  
  identyfikatora, 91, 92, 176  
  klasy, 91

setter, 223  
ShinyText, 28  
słowo kluczowe, 37  
    for, 41  
    function, 172  
    new, 221  
    switch, 218  
    var, 46, 48, 121  
    zastrzeżone, 47  
software, *Patrz: oprogramowanie*  
stopka, 77  
string, 38  
strona, *Patrz też: dokument*  
    interaktywna, 27  
    nagłówek, 77  
    responsywna, 96  
    szkielet, 76  
    tworzenie, 78

## T

tablica, 155, 156, 253  
    długość, 192, 254  
    element, 156  
        dodawanie, 159, 167  
        indeks, 159, 163  
        modyfikacja, 158  
        pobieranie, 157, 158  
        sortowanie, 159  
        usuwanie, 159, 164, 167  
    konwersja na napis, 159, 160  
    łączenie, 159, 161  
    pusta, 237  
    tworzenie, 156, 157  
    wycinek, 159  
target, *Patrz: zdarzenie obiekt*  
tekst, 37, 49  
    akapit, *Patrz: akapit*  
    rozmiar, 93  
    w cudzysłowie, *Patrz: napis*  
terabajt, 26  
 tranzystor, 24  
typ, 48, 49  
    boolowski, *Patrz: typ logiczny*  
    konwersja, 134, 202

liczbowy, 49, 51, 118, *Patrz też: liczba*  
logiczny, 49, 51, 118  
napis, *Patrz: napis*

## W

wartość zwracana, 33  
wejście, *Patrz: dane wejściowe*  
właściwość, 49, 55, 102, 104, 104, 121  
    CSS, 102  
    display none, 235, 236  
    innerHTML, 85, 192  
    length, 254  
    line-height, 145  
    list-style-type, 145  
    margin, 145  
    margin-top, 145  
    nazwa, 103  
    padding, 145  
    position, 98  
    style, 102  
    text-transform, 145  
WolframAlpha, 279, 281  
wyjście, *Patrz: dane wyjściowe*  
wyliczanie do wartości, 118  
wyrażenie, 118  
    logiczne, 252, 265

## Z

zasady ekonomii, 277  
zdarzenie, 106  
    click, 107, 291  
    copy, 107  
    drag, 107  
    drop, 107  
    load, 107  
    mouseover, 107  
    nasłuchiwanie, 106, 107, 291  
    obiekt, 107  
    onclick, 86  
    paste, 107  
    procedura obsługi, *Patrz: procedura*  
        obsługi zdarzeń  
    słuchacz, 107, 108

submit, 107  
 zmienna, 46, 119  
   globalna, 187, 237, 282  
   licznikowa, 252, 254  
   lokalna, 187  
   nazwa, 46, 47  
   tworzenie, 46  
   wartość, 48, 53  
 znacznik, 67, 76, *Patrz też:* element  
   audio, 77  
   body, 78  
   br, 142  
   footer, 77  
   h1, 67  
   header, 77  
   html, 78  
   img, 77  
   ol, 78  
   otwierający, 77  
   p, 67, 77  
   video, 77  
   zamykający, 77  
 znak  
   ', 40  
   !=, 135, 202  
   !=", 135, 202, 203  
   ", 33, 40  
   #, 91, 95  
   \$, 47  
   &&, 206  
   \*, 33  
   \*/, 43  
   ., 91  
   /, 33, 77  
   /\*, 43  
   //, 43  
   :, 92  
   ;, 37, 40, 42, 48  
   \\, 40  
   \_, 47  
   { }, 42  
   ||, 206  
   <, 202, 203  
   < >, 77  
   <!, 203  
   <=, 203  
   =, 53  
   ==, 134, 135, 202  
   ===, 135, 202, 203  
   >, 136, 202, 203  
   >!, 203  
   >=, 136, 203  
   niedrukowany, 41  
   specjalny, 37  
   tabulacji, 42  
 zysk, 277, 278

# 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** 

# Wymyśl i zaprogramuj!

JavaScript sprawia, że strony i aplikacje internetowe działają efektywnie. Ta książka pokaże Ci, jak pisać kod w tym języku, poprzez tworzenie strony internetowej, gier słownych i przygodowych, aplikacji kalkulatora czy stoiska lemoniady — i dużo więcej. Naucz się JavaScriptu i zaskocz swoich przyjaciół!



## W książce:

- Szybki start w świat kodu
- Tworzenie użytecznych gier i aplikacji
- Przepis na kreatywnego programistę

## Chris Minnick i Eva Holland

to doświadczeni programiści rozwiązań internetowych, prowadzący szkolenia z zakresu technologii, a także autorzy książki *Podstawy programowania dla młodych bystrzaków*. Wspólnie stworzyli WatzThis?, firmę skupioną na szkoleniach i kursach programowania.

dla  
**bystrzaków**

Zamówienia telefoniczne:



0 801 339900



0 601 339900

septem  
septem.pl

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

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

Cena 39,90 zł

ISBN 978-83-283-6031-0



9 788328 360310