

W PROSTOCIE TKWI SIĘŁA



Podstawy programowania

dla
**młodych
bystrzaków**



Stwórz
swojego robota

Nadaj mu styl

Wpraw robota
w ruch

**Chris Minnick
i Eva Holland**

doświadczeni programiści

Tytuł oryginału: Writing Computer Code: Learn the Language of Computers!

Tłumaczenie: Wojciech Usarzewicz

ISBN: 978-83-283-5298-8

Original English language edition Copyright © 2016 by John Wiley & Sons, Inc., Hoboken, New Jersey

All rights reserved including the right of reproduction in whole or in part any form.

This translation published by arrangement with Wiley Publishing, Inc.

Oryginalne angielskie wydanie © 2016 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 Wiley Publishing, Inc.

Translation copyright © 2019 by Helion SA

Wiley, the Wiley Publishing logo, For Dummies, Dla Bystrzaków, Dummies Man, Dummies.com, 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.

Wiley, the Wiley Publishing logo, For Dummies, Dla Bystrzaków, Dummies Man, Dummies.com 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.

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)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

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

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

- Kup książkę
- Poleć książkę
- Oceń książkę

- Księgarnia internetowa
- Lubię to! » Nasza społeczność

SPIS TREŚCI

WPROWADZENIE **5**

O książce	5
Wymagania technologiczne	7
O Tobie	7
Ikony użyte w tej książce	8
Pierwszy krok	8

PROJEKT 1: ZOSTAJEMY PROGRAMISTĄ **9**

Co to takiego programowanie?	9
Jak rozmawiać z komputerem?	14
Jakich języków się nauczysz?	18
Narzędzia, których potrzebujesz	22
Przygotowanie przeglądarki	22
Poznajemy JSFiddle	23
Dzielenie się z innymi	31
Podsumowanie	32

PROJEKT 2: GROMADZIMY CZĘŚCI DO BUDOWY ROBOTA **33**

Mów to, co myślisz	34
Piszemy instrukcje	35
Trzymaj się zasad	37
Przekazywanie i odbieranie danych	38
Argumenty i operatory	43
Przygoda z Superkalkulatorem	44
Rozwidlamy kalkulator	45
Używanie Superkalkulatora	46
Superkalkulacje z tekstem	48
Podsumowanie	51

PROJEKT 3: BUDUJEMY CIAŁO ROBOTA **52**

Piszemy kod HTML	52
Jak wygląda tekst bez HTML?	53
Praca z HTML — wszystko sprowadza się do znaczników	54

Zagnieżdżanie znaczników HTML	56
Twój pierwszy dokument HTML	59
Znaj swoje elementy HTML	60
Poznaj robota Douglasa	62
Zmieniamy HTML z użyciem JavaScript	68
Podsumowanie	70

PROJEKT 4: NADAJEMY ROBOTOWI STYL 71

Nadajemy Douglasowi styl	72
Poznajemy podstawy CSS	73
Selektory CSS	74
Deklaracje CSS	78
Właściwości CSS	80
Kolorowanie Douglasa	82
Używanie kolorów CSS	83
Zmieniamy kolory	85
Modyfikowanie robota	89
Podsumowanie	90

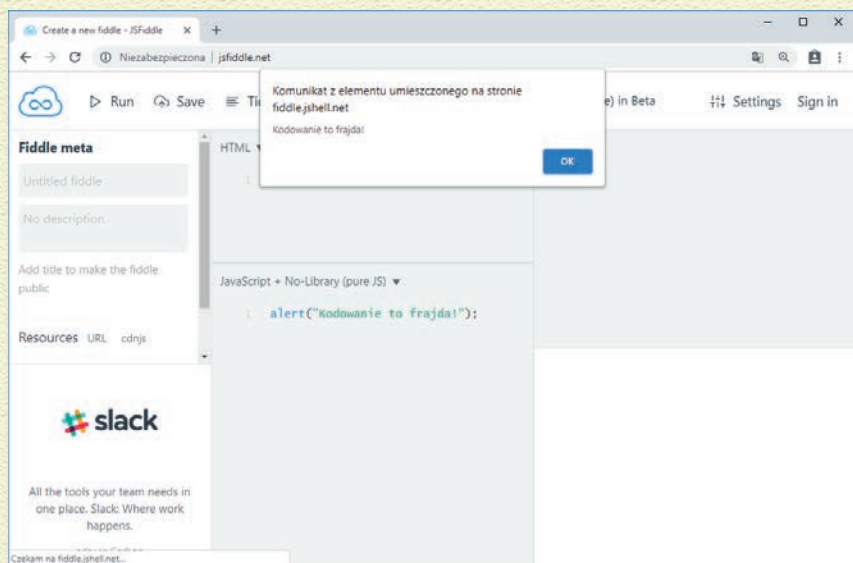
PROJEKT 5: UCZYMY SIĘ ANIMOWANIA ROBOTA 91

Zmiana CSS za pomocą JavaScript	91
Modyfikowanie Douglasa za pomocą JavaScript	93
Eksperymentujemy z Douglasem	96
Tańczący Douglas	98
Tworzenie animacji za pomocą JavaScript	101
Animowanie innego elementu	104
Używanie drugiej funkcji animacji	104
Podsumowanie	108

PROJEKT 6: TWORZYMY GRĘ W SŁÓWKA W JĘZYKU JAVASCRIPT 109

Tworzenie zmiennej historyjki	109
Tworzenie gry	110
Piszemy HTML	112
Stylowanie gry	115
Piszemy kod JavaScript	118
Kończenie programu	121

PROJEKT 2 GROMADZIMY CZĘŚCI DO BUDOWY ROBOTA

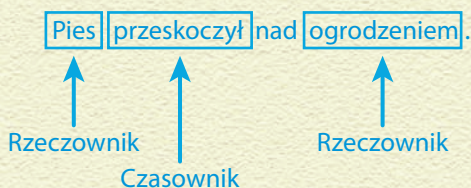


TAK JAK JĘZYK MÓWIONY RZĄDZI SIĘ PEWNYMI ZASADAMI (KTÓRE NAZYWAMY GRAMATYKĄ), TAK I JĘZYK KOMPUTEROWY MA SVOJE ZASADY (ZWANE SKŁADNIĄ).

Kiedy zrozumiesz podstawowe zasady mówienia w języku JavaScript, w dużej mierze zacznie on przypominać zwykły język.

Jeśli uważasz, że to Twój nauczyciel jest ostry, poprawiając Cię, gdy mówisz „poszłem”, to czekaj, aż zobaczysz, jak ostre są zasady JavaScript! Język ten w ogóle nie zareaguje, jeśli popełnisz pewne błędy związane ze składnią.

Gdy piszesz coś w naszym ludzkim języku, Twoje zdania składają się z różnych części, choćby z rzeczowników i czasowników (zobacz rysunek 2.1).



Rysunek 2.1. Zdania składają się z różnych części

W języku JavaScript zdania nazywane są **instrukcjami**. Instrukcje składają się z różnych części, takich jak **argumenty** (inaczej **operandy**, które można porównać do rzeczowników) czy **operatory** (przypominające czasowniki).

W tym rozdziale nauczymy się podstaw składni języka JavaScript i odkryjemy, jak działają w nim argumenty i operatory.

MÓW TO, CO MYŚLISZ

Jeśli jesteś programistą, Twoją pracą jest myślenie o tym, co ma robić program komputerowy, a potem podzielenie tego na mniejsze części i kroki, które komputer może wykonać bez popełniania błędów. Jeśli chcesz, by robot zszedł po schodach na dół i przygotował Ci kanapkę, możesz zacząć od takich instrukcji:

- 1 Obróć głowę w kierunku schodów.
- 2 Użyj czujników wzrokowych, by wypatrzyć przeszkodę.
- 3 Jeśli znajdziesz przeszkodę, określ, czym ona jest.
- 4 Jeśli przeszkoda to kot, spróbuj go skłonić do odblokowania schodów:
 - » Rzucając piłkę.
 - » Mówiąc do kota.
 - » Delikatnie przesuwając kota ręką, póki ten się nie przesunie.
- 5 Jeśli nie ma przeszkód, obróć lewą stopę w kierunku schodów.

- 6** Umieść lewą stopę przed prawą stopą.
- 7** Poszukaj ewentualnych przeszkód.
- 8** Określ, czy stoisz już nad schodami.
- 9** Jeśli nie stoisz jeszcze nad schodami, obróć prawą stopę w kierunku schodów.
- 10** Umieść prawą stopę przed lewą.
- 11** Powtarzaj kroki od 1. do 10., póki nie znajdziesz się nad schodami.

Napisałeś już 11 instrukcji, a robot nawet nie zaczął schodzić po schodach! Prawdziwy program komputerowy instruujący robota do zejścia po schodach i przygotowania kanapki musi być znacznie bardziej rozbudowany niż to, co opisaliśmy w punktach powyżej. Przy każdym wykonanym kroku każdy silnik musiałby być instruowany, jak długo ma działać, a każda możliwa do napotkania przeszkoda musiałaby zostać szczegółowo opisana.

Wszystkie te instrukcje muszą być napisane jako osobne polecenia w języku JavaScript.

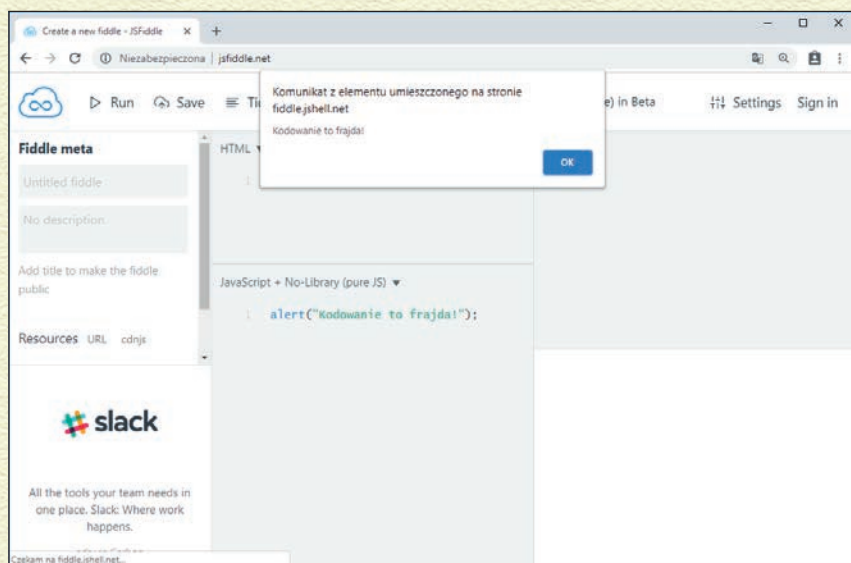
PISZEMY INSTRUKCJE

W języku polskim mówimy zdaniami. W kodzie komputerowym pojedyncze polecenie dla komputera nazywamy *instrukcją*. Podobnie jak zdania, instrukcje składają się z różnych części i trzeba przestrzegać pewnych zasad, by instrukcje te były dla komputera zrozumiałe. Program może składać się z wielu instrukcji albo tylko z jednej.

Następująca instrukcja sprawia, że przeglądarka wyświetla okienko ze zdaniem „Kodowanie to frajda!”:

```
alert("Kodowanie to frajda!");
```

Jeśli wpiszesz to polecenie do panelu JavaScript w narzędziu JSFiddle i klikniesz przycisk *Run*, zobaczysz coś takiego, jak na rysunku 2.2.



Rysunek 2.2. Efekt wykonania instrukcji JavaScript alert

Zauważ, że instrukcja ta zawiera słowo kluczowe (`alert`), kilka symboli (nawiasy i cudzysłowy) oraz trochę tekstu (Kodowanie to frajda!), a kończy się średnikiem (`;`).

W języku polskim można stworzyć nieskończoną liczbę zdań. Tak samo nieskończoną liczbę instrukcji można napisać w języku JavaScript.

Słowo `alert`, którego nie tłumaczymy na język polski, jest przykładem słowa kluczowego JavaScript. Wiele instrukcji zaczyna się od takiego słowa kluczowego, ale nie wszystkie.



Średnik oddziela jedną instrukcję od drugiej, tak samo, jak zdania oddzielane są od siebie kropką. Każda instrukcja powinna zakończyć się średnikiem.

TRZYMAJ SIĘ ZASAD

W języku JavaScript jest kilka zasad, których trzeba przestrzegać, jeśli chcesz, by komputer Cię rozumiał. Oto najważniejsze zasady:

- » Pisownia jest istotna.
- » Duże litery są istotne.
- » Spacje i pusta przestrzeń nie mają znaczenia.

Języki programowania są *naprawdę wybredne*, jeśli chodzi o pisownię. Są wręcz tak wybredne, że nawet jeśli użyjesz dużej litery gdzieś, gdzie nie powinieś, na przykład wpisując **Alert** zamiast **alert**, program nie zadziała.



Kiedy wielkość liter jest istotna, mówimy o danym języku programowania, że jest on „case sensitive”, czyli rozróżnia wielkość liter. Innymi słowy, dla takiego języka ważne jest, czy w instrukcjach stosujemy duże, czy małe litery.

Jedyną sytuacją, kiedy zasada ta nie obowiązuje, jest ta, kiedy piszemy coś, używając cudzysłowu (") lub między apostrofami ('). W takim przypadku błędy w pisowni nie będą problemem. Możesz wtedy pisać, cokolwiek zechcesz!

W naszym przykładzie „Kodowanie to frajda” możesz podmienić słowa w cudzysłowach na inne! Przykładowo, możesz wpisać swoje imię: „Nazywam się Ewa i myślę, że kodowanie to frajda!”.



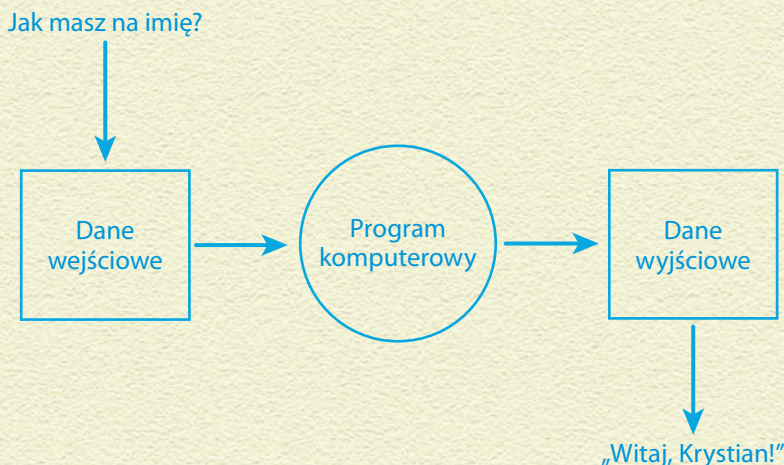
Jeśli piszesz coś, nie używając cudzysłowu, musisz przestrzegać wspomnianych zasad; w przeciwnym wypadku program w ogóle się nie uruchomi.

PRZEKAZYWANIE I ODBIERANIE DANYCH

Programy mają różne rozmiary i wiele różnych celów. Oto kilka rzeczy, które są wspólne dla wszystkich programów komputerowych:

- » Wszystkie programy komputerowe w jakiś sposób odbierają dane od użytkownika.
- » Wszystkie programy komputerowe potrafią przekazywać dane do użytkownika.
- » Wszystkie programy komputerowe posiadają sposób na przechowywanie informacji i na pracę z nimi.

Informacje, inaczej dane, które program odbiera od użytkownika, nazywane są **danymi wejściowymi**. Natomiast to, co program daje użytkownikowi, nazywamy **danymi wyjściowymi** (zobacz rysunek 2.3).



Rysunek 2.3. Komputer używa danych wejściowych, by stworzyć dane wyjściowe

W prawdziwym świecie, kiedy chcesz coś przechować, coś podarować (na przykład prezent), coś przesunąć albo uporządkować, często umieszczasz to coś w pudełku.

Komputer jednak nie interesuje się pudełkiem czekoladek w kształcie serduszka albo pudełkiem z modnymi trampkami w środku. Komputer za to kocha dane. By zaś przechowywać dane i je przemieszczać, komputer wykorzystuje specjalne pudełka, nazywane **zmiennymi**. Zmienna to pudełko, któremu możesz nadać własną nazwę. Nazwa ta będzie reprezentować wszystkie informacje przechowywane w tym pudełku, czyli w zmiennej.

Zmienne umożliwiają jednemu programowi pracę z różnego rodzaju danymi wejściowymi i produkowanie różnych danych wyjściowych.

Wykonaj następujące kroki, aby stworzyć i wykorzystać zmienne:

- 1** Kliknij ikonę JSFiddle w lewym górnym rogu witryny, a potem kliknij *Empty*, by otworzyć nowy projekt.
- 2** Wpisz następujący kod do panelu JavaScript, by utworzyć zmienną o nazwie `mojaKsiazka`:

```
var mojaKsiazka = "Piszemy program komputerowy";
```

- 3** Kliknij przycisk *Run*.

Jeśli wszystko zrobisz poprawnie, na razie na ekranie nie zdarzy się nic nowego.

- 4** Wpisz następujący kod w kolejnym wierszu w panelu JavaScript:

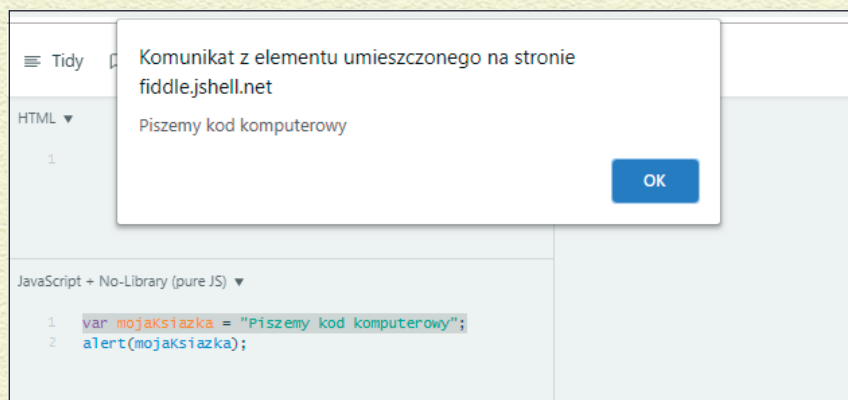
```
alert(mojaKsiazka);
```

- 5** Kliknij przycisk *Run*.

Teraz pojawi się okienko, w którym przeczytasz komunikat `Piszemy program komputerowy` (zobacz rysunek 2.4).

Jak widzisz, tworzenie zmiennej w języku JavaScript jest bardzo proste! Aby stworzyć zmienną, musisz użyć słowa kluczowego `var`, po którym podajesz nazwę zmiennej, na przykład:

```
var mojaKsiazka
```



Rysunek 2.4. Tworzenie zmiennej i komunikatu



Zauważ, że oddzielamy kolejne słowa w naszej zmiennej, używając dużej litery rozpoczynającej drugie słowo. Nazwy zmiennych nie mogą zawierać spacji, więc programiści stworzyli różne sposoby na oddzielanie od siebie kolejnych słów. Ten konkretny sposób zapisu zmiennych nazywamy camelCase, a nazwa ta wychodzi się z angielskiego słowa camel, czyli wielbłąd. Czy domyślasz się, skąd taka nazwa?

Jako programista, masz sporo swobody w sposobie nazywania zmiennych. Możesz być dość kreatywny w tworzeniu tych nazw, ale bez przesady. Najważniejsze jest to, by nazwy zmiennych określały, co jest w nich przechowywane.

Oto przykłady dobrze nazwanych zmiennych:

```
var mojeImie  
var ulubioneJedzenie  
var dataUrodzin  
var aktualnaGodzina
```

Samo popatrzenie na nazwę zmiennej pozwala Ci się domyślić, jakiego rodzaju informacja jest w tej zmiennej przechowywana.

Czy po przyjrzeniu się tym przykładom potrafisz wymyślić dobre nazwy zmiennych do przechowania następujących informacji?

- » imię Twojego zwierzaka,
- » Twój ulubiony przedmiot w szkole,
- » imię najlepszego przyjaciela,
- » Twój adres.

Oto nasze sugestie:

- » imieMojegoZwierzaka,
- » mojUlubionyPrzedmiotSzkolny,
- » imieNajlepszegoPrzyjaciela,
- » mojAdres.

Oprócz reguły mówiącej, że nazwa zmiennej nie może zawierać spacji, jest jeszcze kilka zasad, których trzeba przestrzegać:

- » Nazwa zmiennej musi zaczynać się od litery, podkreślenia (`_`) lub symbolu dolara (`$`).
- » Nazwy zmiennych mogą zawierać tylko litery, cyfry, podkreślenia i symbole dolara.
- » Nazwy zmiennych rozróżniają wielkość liter!
- » Pewne słowa nie mogą być użyte jako nazwy zmiennych, ponieważ mają inne znaczenie w języku JavaScript. Takie zarezerwowane słowa to:

<code>break</code>	<code>case</code>	<code>class</code>	<code>catch</code>
<code>const</code>	<code>continue</code>	<code>debugger</code>	<code>default</code>
<code>delete</code>	<code>do</code>	<code>else</code>	<code>export</code>
<code>extends</code>	<code>finally</code>	<code>for</code>	<code>function</code>
<code>if</code>	<code>import</code>	<code>in</code>	<code>instanceof</code>
<code>let</code>	<code>new</code>	<code>return</code>	<code>super</code>
<code>switch</code>	<code>this</code>	<code>throw</code>	<code>try</code>
<code>typeof</code>	<code>var</code>	<code>void</code>	<code>while</code>
<code>with</code>	<code>yield</code>		

Tworząc zmienne, słowo `var` musisz wpisać tylko, kiedy pierwszy raz nazywasz te zmienne. Kiedy chcesz użyć zmiennej albo nadać jej nową wartość, wykorzystujesz tylko nazwę tej zmiennej.

Kiedy w zmiennej przechowujesz dane, możesz je odczytać w dowolnym momencie, wpisując nazwę zmiennej. Kiedy więc używasz w programie nazwy zmiennej `mojeUlubioneJedzenie`, to tak samo, jakbyś po prostu mówił „pizza”.

Spróbujmy!

1 Po ostatniej instrukcji, którą wpisałeś w panelu JavaScript, dopisz:

```
mojaKsiazka = "JavaScript dla dzieci";
```

2 W kolejnym wierszu napisz:

```
alert(mojaKsiazka);
```

3 Kliknij przycisk *Run*.

Najpierw pojawi się pierwsze okienko, w którym przeczytasz *Piszemy program komputerowy*.

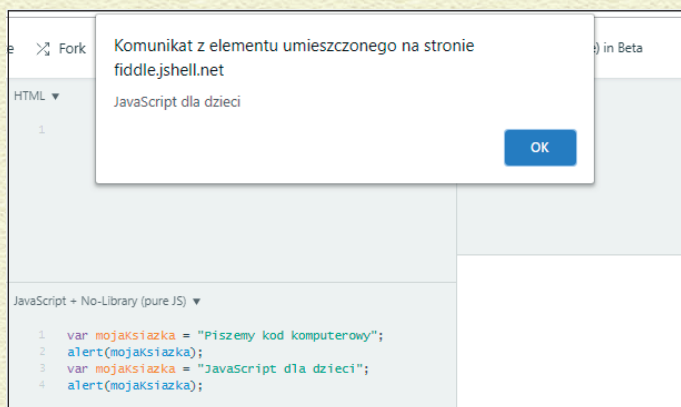
4 Zamknij to okienko, klikając *OK*.

Teraz pojawi się drugie okienko, w którym przeczytasz *JavaScript dla dzieci* (rysunek 2.5).

Zauważ, że użyliśmy tej samej instrukcji do wyświetlania okienek z dwoma różnymi komunikatami. To zaledwie jedna z wielu niesamowitych rzeczy, które można zrobić za pomocą zmiennych.



Dane w zmiennej nazywamy również wartością zmiennej.



Rysunek 2.5. Otwarcie drugiego okna z innym komunikatem

ARGUMENTY I OPERATORY

W języku JavaScript, kiedy fragment kodu coś robi, na przykład dodaje dwie liczby do siebie, by uzyskać inną liczbę, fragment ten nazywamy **wyrażeniem**. Oto przykład:

$1+1 = 2$

Wyrażenia składają się z argumentów (inaczej operandów, na przykład liczba 1) i operatorów (na przykład znaków = albo +).

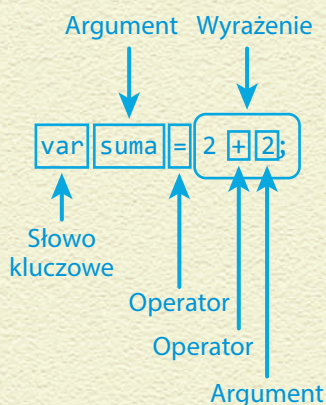
Argumentami mogą być słowa, liczby, a nawet zmienne.



Kiedy zmienna zawiera wartość, można ją wykorzystać w dowolnym miejscu, w którym normalnie użyłbyś tej wartości.

Operator to symbol wykonujący jakieś zadanie (czyli operację) z wykorzystaniem argumentów.

Rysunek 2.6 pokazuje różne fragmenty instrukcji JavaScript.



Rysunek 2.6. Różne fragmenty instrukcji

JavaScript ma wiele różnych operatorów, które wykonują różne zadania. Zamiast pokazać Ci listę wszystkich operatorów, napisaliśmy program, który pozwala *Tobie* właśnie wybrać operatory i argumenty, a potem uzyskać wyniki operacji!

Oto Superkalkulator! Obsługuje zarówno słowa, jak i litery oraz liczby.

PRZYGODA Z SUPERKALKULATOREM

Nigdy wcześniej nie widziałeś takiego kalkulatora. Może on dodawać, odejmować, mnożyć, a nawet dzielić!

To nie wszystko! Ten Superkalkulator może też wziąć słowa i skleić je razem! Powiedzmy, że masz słowo *Java* oraz drugie słowo, *Script*, i chcesz je jakoś skleić razem. Niemożliwe? Nie z naszym kalkulatorem!

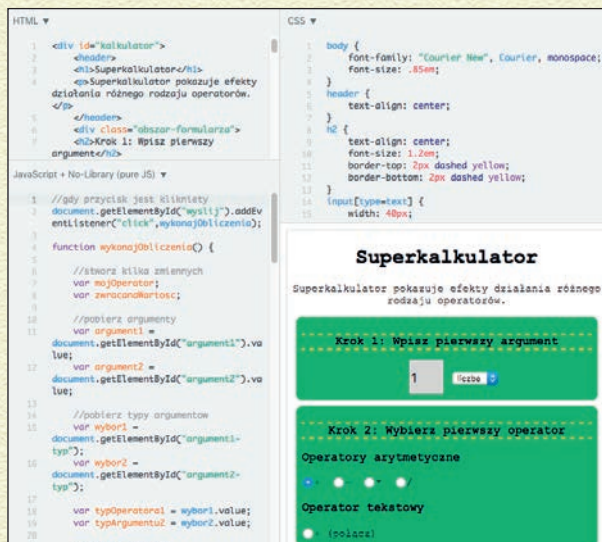
W jaki sposób działa ten kalkulator? Jak możesz go zdobyć? Zaraz Ci wszystko powiemy i Twój świat już nigdy nie będzie taki sam!

ROZWIIDLAMY KALKULATOR

Stworzyliśmy ten wspaniały kalkulator w JSFiddle. Po prostu wykonaj następujące kroki, by stworzyć własną wersję naszego programu:

- 1** Otwórz przeglądarkę internetową i zaloguj się na <http://jsfiddle.net>.
- 2** Następnie przejdź do programu naszego kalkulatora, https://jsfiddle.net/Helion_przyklady/4zcbh62a.

Dzięki temu zobaczysz kod naszego programu (jak na rysunku 2.7).



Rysunek 2.7. Superkalkulator w języku JavaScript

- 3** Kliknij przycisk *Fork* (rozwidlenie) na górnym pasku, by stworzyć kopię naszego programu pod własnym kontem w JSFiddle.
- 4** Użyj opcji po lewej stronie, by zmienić nazwę Superkalkulatora na „Superkalkulator *Twoje imię*”.
- 5** Kliknij przycisk *Update* (zaktualizuj) w górnym menu, a następnie kliknij tam przycisk *Set as base* (ustaw jako projekt bazowy).

Teraz masz już swoją własną wersję Superkalkulatora! Jesteś gotów nauczyć się, jak program ten działa, a potem wykorzystać go, by nauczyć się pracy z argumentami i operatorami.

UŻYWANIE SUPERKALKULATORA

Superkalkulator pozwala Ci widzieć rezultaty operacji na różnych operatorach. Posiada dwa miejsca, w których możesz umieścić argumenty; przy każdym z nich znajduje się rozwijane menu, w którym możesz określić rodzaj wpisywanych danych (liczby lub tekst), oraz przyciski opcji, dzięki którym określisz operator.



Przyciski opcji (ang. radio buttons) to nazwa w HTML określająca okrągłe przyciski, które mogą być grupowane ze sobą. Różnią się one od przycisków wyboru tym, że można wybrać tylko jeden przycisk opcji z całej grupy. Tymczasem w przyciskach wyboru możesz jednocześnie zaznaczyć wiele przycisków. Wydaje nam się, że nazwano te przyciski słowem radio (co tłumaczymy dosłownie jako radio) dlatego, że przypominają one działaniem przyciski wyboru stacji radiowej. Wciśnięcie jednego przycisku ustawia stację, niejako ją zaznaczając, a odznaczając wszystkie pozostałe.

Aby zacząć pracę z Superkalkulatorem, popatrz teraz na ustawienia widoczne w panelu *Result* w prawym dolnym regionie ekranu.



*Możesz zmienić wielkość paneli w JSFiddle tak, by panel *Result* był tak duży, jak to tylko możliwe.*

W pierwszej części kalkulatora, **Krok 1.**, pole tekstowe na rysunku 2.8 zawiera pojedynczą wartość, cyfrę 1. Rozwijane menu obok ustawione jest na typ danych *liczba*.

Poniżej tych opcji znajduje się region **Krok 2.**, w którym możesz wybrać rodzaj operatora. Operatory podzielone są na dwie grupy: operatory arytmetyczne oraz operator tekstowy. Zauważ, że przy operatorach arytmetycznych zaznaczony jest już pierwszy operator, +, jak pokazuje rysunek 2.9. To operator dodawania.

Superkalkulator

Superkalkulator pokazuje efekty działania różnego rodzaju operatorów.

Krok 1.: Wpisz pierwszy argument

1

liczba ↕

Rysunek 2.8. Domyślną wartością pierwszego argumentu jest cyfra 1

Krok 2.: Wybierz pierwszy operator

Operatory arytmetyczne

☒ + ☐ - ☐ * ☐ /

Operator tekstowy

☐ + (połącz)

Rysunek 2.9. Domyślnie wybrany jest operator dodawania

Pod obszarem wyboru operatorów znajduje się obszar wyboru drugiego argumentu, **Krok 3.** Domyślnie również i tutaj podaliśmy cyfrę 1 (zobacz rysunek 2.10).

Krok 3.: Wpisz drugi argument

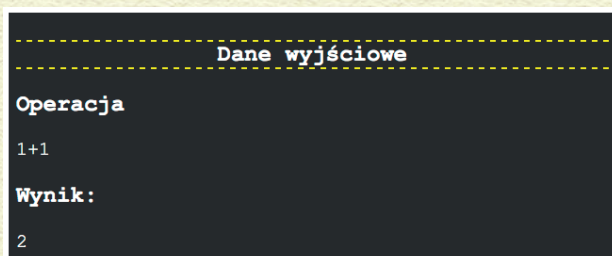
1

liczba ↕

Rysunek 2.10. Dla drugiego argumentu podajemy wartość 1

Kiedy wszystko to zbierzemy razem, domyślną operacją naszego kalkulatora okazuje się najprostszy problem matematyczny świata: $1+1$.

Kliknij przycisk *Wykonaj operację!* na dole kalkulatora. Wynik zostanie wyświetlony na samym dole, w obszarze *Dane wyjściowe*, jak pokazuje rysunek 2.11.



Rysunek 2.11. Dodawanie liczb w Superkalkulatorze

Oczywiście, znałeś już wynik tej operacji! Możesz sobie teraz myśleć, że nasz kalkulator wcale nie jest taki super. Nie tak szybko! Spróbujmy czegoś trudniejszego.

- 1 Dla pierwszego argumentu wpisz liczbę 435.**
- 2 Jako operator wybierz operator mnożenia, *.**
- 3 Dla drugiego argumentu wpisz liczbę 888.**
- 4 Wciśnij przycisk *Wykonaj operację!*.**

Wynik, 386280, pojawia się od razu, jak pokazuje rysunek 2.12.

To już jest super, nie?

Każdy kalkulator wykonuje tego typu operacje. Jednak nasz Superkalkulator, w przeciwieństwie do zwykłych, może również pracować ze słowami! W kolejnym podrozdziale dowiesz się jak.

SUPERKALKULACJE Z TEKSTEM

Spróbujmy teraz wykonać operację, która choć wciąż prosta (bo uwzględnia tylko dwa argumenty), pokaże Ci coś interesującego na temat JavaScript.

- 1 Pozostaw wartości argumentów niezmienione (1 oraz 1), ale wybierz operator tekstowy, +, czyli połączenia, i zmień typ danych każdego argumentu z liczby na tekst, używając do tego rozwijanego menu.**

Teraz, gdy klikniesz przycisk *Wykonaj operację!*, zobaczysz wynik 11 (jak na rysunku 2.13).

Superkalkulator

Superkalkulator pokazuje efekty działania różnego rodzaju operatorów.

Krok 1.: Wpisz pierwszy argument

liczba ▾

Krok 2.: Wybierz pierwszy operator

Operatory arytmetyczne

+
-
*
/

Operator tekstowy

+ (połącz)

Krok 3.: Wpisz drugi argument

liczba ▾

Krok 4.: Kliknij przycisk

Wykonaj operację!

Dane wyjściowe

Operacja

435*888

Wynik:

386280

Rysunek 2.12. Mnożenie za pomocą Superkalkulatora

- 2** Pozostaw operator ustawiony jako operator tekstowy. Tak samo niech typ danych wciąż będzie ustawiony na tekst. Zmień natomiast wartość pierwszego argumentu z 1 na Java oraz drugiego argumentu z 1 na Script.
- 3** Kliknij przycisk *Wykonaj operację!*.
 Efekt połączenia słów „Java” oraz „Script” pokazano na rysunku 2.14.
- 4** Nie zmieniaj rodzaju operatora ani typu danych. Zmień natomiast pierwszy argument z Java na swoje imię, po którym naciśnij również spację, a potem zmień drugi argument ze Script na Twoje nazwisko.

```
----- Dane wyjściowe -----  
  
Operacja  
"1"+"1"  
  
Wynik:  
11
```

Rysunek 2.13. Wynik połączenia 1 oraz 1

```
----- Dane wyjściowe -----  
  
Operacja  
"Java"+"Script"  
  
Wynik:  
JavaScript
```

Rysunek 2.14. Połączenie słów Java oraz Script

5 Kliknij przycisk *Wykonaj operację!*.

Efektem operacji będzie Twoje imię i nazwisko, wraz z przerwą (spacją) między nimi. Przykład ten nauczył Cię ważnej rzeczy: kiedy piszesz w języku JavaScript, język ten uwzględni wszystkie spacje, które wstawisz do argumentów.

6 Zmień operator z tekstowego na dowolny operator arytmetyczny inny niż dodawanie. Nie zmieniaj jednak typów danych, niech wciąż będą ustawione na tekst.

7 Kliknij przycisk *Wykonaj operację!*.

Rezultat, widoczny na rysunku 2.15, to NaN, co oznacza „Not a Number” („To nie liczba”). JavaScript nie potrafi wykonać operacji arytmetycznej, używając w tym celu liter, a komunikat NaN właśnie o tym Cię informuje.

```
-----
Dane wyjściowe
-----
Operacja
"Eva"-"Holland"

Wynik:

NaN
```

Rysunek 2.15. Próba wykonania obliczenia matematycznego (innego niż dodawanie) za pomocą imienia i nazwiska skutkuje wynikiem NaN

PODSUMOWANIE

W tym rozdziale nauczyłeś się ważnych rzeczy na temat programowania. Dowiedziałeś się, że zmienne to pudełka, w których możesz przechowywać wartości. Nauczyłeś się również, że „zdania” w programowaniu to inaczej **instrukcje**. W końcu dowiedziałeś się, jak wykorzystywać operatory oraz argumenty, by budować instrukcje, które mogą zmienić dane wejściowe w dane wyjściowe. W kolejnym rozdziale zaczniemy budować robota, którego będziesz mógł modyfikować i animować za pomocą języków HTML, CSS oraz JavaScript.



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 

Poznaj język komputerów

Nabywanie umiejętności programowania przypomina uczenie się języka obcego. Ta książka to świetny poradnik, który sprawi, że zaczniesz pisać w dziwnie wyglądających językach tworzących globalną sieć. Dzięki prostym wskazówkom nauczysz się pracować z prawdziwym kodem i zbudujesz własnego przeglądarkowego robota.



W książce:

- Kod, który tworzy ciało i strukturę robota
- Zmiana kolorystyki i kształtu maszyny
- Uczymy robota tańczyć!

Chris Minnick i **Eva Holland** to doświadczeni programiści stron internetowych i założyciele firmy WatzThis?, w której wspólnie szkolą w zakresie 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>

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

Cena 29,90 zł

ISBN 978-83-283-5298-8



9 788328 352988