

LEKCJE Z JAVASCRIPT

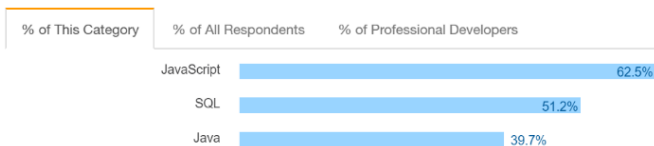
Agnieszka Borowiecka, Wanda Jochemczyk, Katarzyna Olędzka
Ośrodek Edukacji Informatycznej i Zastosowań Komputerów
02-026 Warszawa, ul. Raszyńska 8/10
{agnieszka.borowiecka, wanda.jochemczyk,
katarzyna.oledzka}@oeiizk.waw.pl

Abstract. JavaScript is a great tool for learning programming and algorithmic thinking. Writing a first program is surprisingly simple. Starting from simple examples, one can lead students step by step to more difficult problems. Moreover, practical applications are at your fingertips.

1. Wprowadzenie

Niektórym JavaScript kojarzy się z wyskakującymi okienkami na stronach internetowych lub innymi sztucznymi „upiększaczami” stron. W rzeczywistości JavaScript to język sieci (*Language of the Web*), który niekoniecznie musi służyć do irytowania użytkownika. Dzięki jego stosowaniu możemy dodać naprawdę pożyteczne funkcjonalności na stronie. Postaramy się pokazać, że JavaScript stanowi świetne narzędzie do nauki programowania i myślenia algorytmicznego, gdyż łączy w sobie wiele elementów m.in. powszechną dostępność i wszechstronne zastosowania. Trzeba przyznać, że w wielu rankingach popularności języków programowania JavaScript jest na czołowym, jeżeli nie na pierwszym miejscu. Jednak, należy zwrócić uwagę, że mało kto poleca go jako pierwszy język programowania. Zwykle zaczynamy się uczyć programować w C/C++, programujemy w Pythonie, do niedawna także w Pascalu, ale nie w JS. Postawmy retoryczne pytanie, dlaczego?

Programming Languages



Rysunek 1 Ranking języków programowania¹

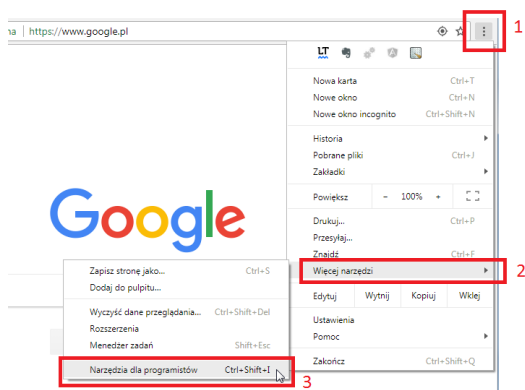
¹ <https://insights.stackoverflow.com/survey/2017#overview>

JavaScript został stworzony przez amerykańskiego programistę (i hackera!) Brendan'a Eich, współtwórcę projektu Mozilla, Mozilla Foundation i Mozilla Corporation. JavaScript – jak sama nazwa wskazuje – jest skryptowym językiem programowania, którego kod może być interpretowany przez przeglądarkę. Z jednej strony można go stosować na stronach internetowych w połączeniu z HTML i CSS, z drugiej – po stronie serwerów na platformie Node.js. Uruchamiany przez przeglądarkę, może działać na obiektach DOM – drzewa struktury dokumentu oraz zarządzać arkuszami stylów. Umożliwia także tworzenie ciasteczek, pobieranie informacji o przeglądarce, wyświetlanie okien dialogowych. Skrypty JS mogą także reagować na różne zdarzenia. Istnieją tysiące bibliotek i frameworków opartych na JavaScript o różnych zastosowaniach, które ułatwiają tworzenie ciekawych projektów. Chociaż nie wszyscy zdają sobie z tego sprawę, niemal na każdym urządzeniu mamy jego interpreter wbudowany w przeglądarkę.

Z dydaktycznego punktu widzenia warto zwrócić uwagę na kilka aspektów. Napisanie pierwszego programu jest zaskakująco proste, a zaczynając od prostych przykładów możemy prowadzić ucznia krok po kroku do coraz trudniejszych zagadnień. Należy przy tym pamiętać, że zastosowania praktyczne są na wyciągnięcie ręki, a nic nie motywuje uczniów do wysiłku bardziej, niż świat gier. Uczniowie, gdy tylko złapią bakcyła, są w stanie poświęcić naprawdę wiele czasu, by stworzyć coś atrakcyjnego i wartościowego zarazem.

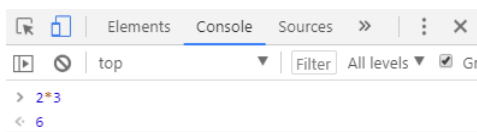
2. Zupełnie pierwszy program

Jak zacząć? Można uruchomić przeglądarkę Chrome i przejść do **Narzędzia dla programistów (Ctrl+Shift+I)**.



Rysunek 2 Narzędzia dla programistów w przeglądarce Google Chrome

Następnie w konsoli napisać program: **2*3**. Otrzymamy wynik: **6**.



Rysunek 3 Pierwszy program w konsoli

JavaScript można też „zatrudnić” do wypisania tekstu na stronie korzystając z polecenia **document.write(tekst)**.

```
01. <!DOCTYPE html>
02. <html>
03.   <head>
04.     <title>Page Title</title>
05.   </head>
06.   <body>
07.     <script> document.write("Zaczynam programować w JS.")
08.   </script>
09. </body>
10. </html>
```

Rysunek 4 Wypisywanie tekstu na stronie

Wypisywany tekst może być sformatowany za pomocą znaczników HTML.

```
01. document.write("Zaczynam programować w <strong>JS</strong>.")
```

Warto zaznaczyć, że zwykle skrypty umieszcza się w plikach zewnętrznych.

```
01. <script src="animacja.js"></script>
```

Przykłady te pokazują, że początki programowania w JavaScript są bardzo proste, a dostępne narzędzia mamy na wyciągnięcie ręki.

3. Od prostych do bardziej skomplikowanych przykładów

W JavaScript można nie tylko liczyć i wypisywać teksty, ale animować obiekty na ekranie. Stworzymy prostą stronę HTML z jasnozielonym animowanym kwadratem poruszającym się wewnątrz ciemnozielonej planszy.

```
01. <!DOCTYPE html>
02. <html>
03.   <head>
04.     <title>Animacja</title>
05.   </head>
06.   <body>
07.     <div id="plansza">
08.       <div id="kwadrat"> </div>
09.     </div>
10.   </body>
11. </html>
```

Rysunek 5 Kod HTML

```
01. #plansza {
02.     width: 400px;
03.     height: 400px;
04.     background: #aae82e;
05.     position: relative;
06. }
07. #kwadrat {
08.     width: 10px;
09.     height: 10px;
10.     background: #006600;
11.     position: absolute;
12. }
```

Rysunek 6 Kod CSS

Przyjrzyjmy się funkcji animacja zmieniającej położenie kwadratu. Zaczynamy od najprostszej wersji, w której kwadrat przesuwa się poziomo od lewej do prawej. Gdy dojdzie do prawej krawędzi planszy, animacja zaczyna się od początku.

```
01. var pozycja = 0;
02. var kwadrat = document.getElementById('kwadrat');
03. function animacja() {
04.     pozycja = (pozycja + 1) % 400;
05.     kwadrat.style.left = pozycja + 'px';
06. }
07. window.onload = function() {
08.     var t = setInterval(animacja, 10);
09. };
```

Rysunek 7 Kod JavaScript

Następnie zadajemy uczniom pytania: Jak zmodyfikować skrypt, żeby kwadrat odbijał się od brzegu? Co trzeba zmienić, by poruszał się w pionie? Czy może poruszać się po skosie? Wprowadzamy kolejne poprawki, tworząc coraz bardziej skomplikowany program. Możemy także dodać reakcję na przyciski i sterować kwadratem w czterech kierunkach.

Warto zwrócić uwagę, że jednym z częstych błędów przy nauce programowania jest pokazywanie uczniom długiego kodu, który wymaga wyjaśnień niemal przy każdej linii. Zadaniem uczniów jest jedynie przepisanie gotowego rozwiązania zaprezentowanego przez nauczyciela. Zamiast tego można tak dobrać przykłady, by uczeń samodzielnie konstruował wiedzę rozwiązując krok po kroku coraz to trudniejsze zadania. Zasada stopniowania trudności sprawdza się znakomicie.

4. Strona www z elementami dynamicznymi

Spróbujmy teraz stworzyć stronę www ze zmieniającą się zawartością. Można na niej umieścić przewijaną galerię zdjęć, losowo pojawiające się cytaty czy quiz z wieloma pytaniami. Wszystkie te elementy wymagają znajomości HTML, CSS i oczywiście JS.



Rysunek 8 Fragment strony o sławnych informatykach

```
01. var obrazki = ["ada_lovelace", "alan_turing", "niklaus_wirth", "dennis_ritchie"];
02. var nr = 0;
03. function nast() {
04.     var galeria = document.getElementById("galeria");
05.     nr = (nr + 1) % obrazki.length;
06.     galeria.src = "grafika/"+obrazki[nr]+".png";
07. }
08. window.onload = function() {
09.     var t = setInterval(nast, 1000);
10. };
```

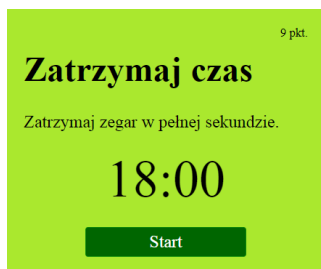
Rysunek 9 Kod JavaScript

Funkcja zmieniająca zawartość strony tylko nieznacznie różni się od animacji kwadratu. Wprowadzamy w niej tablicę przechowującą nazwy obrazków. Pełną ścieżkę z adresem obrazka budujemy z trzech składowych: nazwy folderu, nazwy obrazka i rozszerzenia pliku. Następnie uczniowie rozszerzają skrypt o zmianę podpisów pod obrazkiem. Potrzebne treści również będą przechowywane w tablicy.

Przykład ze stroną [www](#) jest o tyle cenny, że pokazuje konkretne zastosowanie poznanej wiedzy. Uczeń nie tylko wie teoretycznie, jak napisać skrypt, ale może go wykorzystać na stworzonej przez siebie stronie [www](#).

5. JS i tworzenie gier

Gra polega na kliknięciu w pole Start/Stop w odpowiednim momencie. Dokładniej, jeżeli użytkownik kliknie Start, na ekranie pojawi się czas wyświetlany z dokładnością do setnych sekund. Ponowne kliknięcie w pole powoduje zatrzymanie stopera. Jeżeli zatrzymaliśmy stoper o pełnej sekundzie to otrzymujemy punkt, w przeciwnym przypadku go tracimy.



Rysunek 10 Gra w zatrzymywanie zegara

Gra wymaga zaprogramowania kilku elementów: zmieniającego się pola Start/Stop, wyświetlania prawidłowo czasu, gdy stoper jest włączony, zatrzymania stopera i zliczania punktów. Każdy z tych elementów można przeanalizować osobno tak, aby potem łącząc je stworzyć funkcjonalny skrypt.

```

01.   var czy_biegnie = false;
02.   var punkty = 10;
03.
04.   // wyświetlanie czasu
05.   function teraz(){
06.     var x = document.getElementById("zegar");
07.     var czas = new Date();
08.
09.     var sekundy = czas.getSeconds();
10.     if (sekundy < 10) {sekundy = "0"+sekundy};
11.
12.     var mili = czas.getMilliseconds();
13.     var mili2 = Math.round(mili/100 - 0.5);
14.     if (mili2==0) mili2 = "00";
15.     else mili2 = 10*mili2
16.
17.     if (czy_biegnie) x.innerHTML = sekundy + ":" + mili2;
18.     setTimeout(teraz,1);
19.   }

```

Rysunek 11 Wyświetlanie czasu

```

01.   // zatrzymanie i uruchamianie stopera
02.   function zatrzymaj(){
03.     if (czy_biegnie){
04.       var mili= new Date().getMilliseconds();
05.       var x = Math.round(mili/100 - 0.5);
06.       if (x == 0) punkty++;
07.       else punkty--;
08.       document.getElementById("wyswietl_punkty").innerHTML = punkty;
09.     }
10.
11.     czy_biegnie = !czy_biegnie;
12.     if (czy_biegnie)
13.       document.getElementById("przycisk_stop").innerHTML ="Stop";
14.     else
15.       document.getElementById("przycisk_stop").innerHTML ="Start";
16.   }

```

Rysunek 12 Zatrzymanie/uruchamianie stopera oraz zmiana napisu w polu

Dlaczego tworzenie gier jest dobrym sposobem na naukę programowania? Ponieważ gry są mocno związane z pozytywnymi emocjami. Wiadomo, że gdy nauka wiąże się z przyjemnością, jest bardziej efektywna. Łatwiej zabrać się do pracy, gdy tworzenie czegoś nowego sprawia ogromną frajdę. Co prawda nie każdy zostanie programistą gier, ale stworzenie jednej czy dwóch może przyczynić się do rozwoju wielu umiejętności.

6. Trzeba się uczyć

JavaScript jest dobrym narzędziem wspomagającym naukę programowania. Jego zaletą jest powszechna dostępność. Uczeń ma dostęp do interpretera języka, dodatkowych, często darmowych bibliotek, a ponadto może uzyskiwać wiedzę na temat programowania w JavaScript w wielu miejscach. Można poznać podstawy języka korzystając z bogatych materiałów w Akademii Khana. Jest tam przygotowany kurs krok po kroku z materiałami i ćwiczeniami. Szkoda tylko, że opiera się jedynie na grafice i przez to brakuje przykładów praktycznych zastosowań. Alternatywą jest platforma SoloLearn.com, w której mamy w języku angielskim kurs wprowadzający w JS bardzo systematycznie. Jest on dostępny zarówno na komputery, jak i na urządzenia mobilne. Zawiera sporo praktycznych przykładów, jest jednak trudniejszy niż ten w KhanAcademy. Nie można też zapomnieć o stale rozwijanych tutorialach na w3schools.com². Są tam przygotowane uporządkowane materiały poprzeplatane ćwiczeniami. Warto po nie sięgać. Ogólnie, materiałów i kursów w Internecie jest sporo. Każdy może wyszukać coś dla siebie, jak również dzielić się wynikami swojej pracy.

Uczeń może się nimi pochwalić wśród kolegów i wymienić doświadczeniami. Warto pamiętać, że ponieważ JavaScript jest językiem interpretowanym, można „podglądać” rozwiązania innych analizując kod na różnych stronach.

7. Na zakończenie

Zamiast podsumowania, wypowiedź z blogu...

„Jeśli chodzi o JavaScript to niesamowicie dużo dało mi wyzwanie #JavaScript30. Na początku patrzyłam na kod JS jakby był z innej planety (bo w sumie trochę tak jest). Dzisiaj, potrafię go przeczytać. Nie rozumiem jeszcze wszystkiego, ale rozpoznaję tablice, pętle i funkcje. Staram się rozumieć, co do czego się odwołuje, jak coś działa. Jeszcze ogrom pracy przede mną, ale czuję, że zmierzam w dobrym kierunku. Nie wzdyga mnie już też na samą myśl o nauce JavaScript.”

² <https://www.w3schools.com/jS>

Powstaje tylko pytanie na koniec: Czy można podjąć wyzwanie i razem z uczniami zacząć przygodę z JS?

Literatura

1. Crozkford D., *JavaScript: The Good Parts*, O'REILLY, 2008.
2. Akademia Khana, <https://pl.khanacademy.org>, ostatni dostęp 10.06.2018 roku.
3. Platforma elearningowa SoloLearn, <https://www.sololearn.com>, ostatni dostęp 10.06.2018 roku.
4. Platforma edukacyjna OEIiZK, <http://programowanie.oeiizk.edu.pl>, ostatni dostęp 10.06.2018 roku.