

---

# Koduj w listopadzie

## Materiały z wyzwania

[www.wakeupandcode.pl](http://www.wakeupandcode.pl)

---

Cześć!

Jestem Joanna i prowadzę bloga Wake up and Code. To właśnie w ramach newslettera, którego prowadzę razem z blogiem postanowiłam zorganizować pięciodniowe wyzwanie z podstaw JavaScriptu. Poniżej znajdziesz zadania z pięciu dni. Mam nadzieję, że dzięki nim poznasz podstawy JS i zostaniesz z nim na dłużej :)

---

# Instrukcje

Celem wyzwania jest spędzenie z kodem około 15 minut dziennie. Zadania obejmują zupełne podstawy JavaScript. Ma to być dla Ciebie wstęp do dalszej nauki. Jeśli już znasz podstawy JS, wyzwanie będzie dla Ciebie powtórką. Mam nadzieję, że Ci się spodoba!

Będziemy działać lokalnie na Twoim komputerze, nie na stronie internetowej, więc potrzebujesz edytora, w którym będziesz mogła/mógł zapisywać kod. Polecam Brackets. Ściągnij edytor według instrukcji na oficjalnej stronie. Jeśli korzystasz z innego edytora, to świetnie! Nie musisz specjalnie instalować Brackets.

Klikając w poniższy link, możesz pobrać .zip z plikami startowymi.

<https://drive.google.com/file/d/1v3E7REm5kr0ooA0yB79Q8U1MFINzbQog/view?usp=sharing>

Wypakuj pliki do wybranej lokalizacji, a następnie otwórz je w edytorze. Znajdziesz tam plik HTML, CSS oraz JavaScript, które będą służyć Ci do rozwiązywania zadań. Kliknij dwukrotnie na plik index.html, aby otworzyć go w przeglądarce. W konsoli przeglądarki zobaczysz efekty Twoich działań. Instrukcje otwarcia konsoli, znajdziesz w pliku HTML w plikach źródłowych (wystarczy, że otworzysz go w przeglądarce).

Jeśli umiesz budować podstawową strukturę, tj. stworzyć plik HTML i podłączyć do niego plik JS, możesz zrobić to sama/sam. Nie musisz korzystać z plików startowych.

Jeśli nie masz możliwości pracować lokalnie, możesz alternatywnie założyć konto na Codepen i tam rozwiązywać zadania. Jednak lepiej byłoby pracować na plikach na

---

Twoim komputerze, żeby zasymulować faktyczną pracę z kodem.

Zadanie ma nauczyć Cię samodzielnego wyszukiwania rozwiązań. Nie dostaniesz tutaj wszystkiego podanego na tacy. Przyjdzie do Ciebie zadanie i będziesz musiała/musiał poszukać rozwiązania.

Jeśli zupełnie nie znasz JavaScriptu, pomocne mogą być następujące strony:

- <https://developer.mozilla.org/pl/docs/Web/JavaScript>
- [https://www.w3schools.com/js/js\\_intro.asp](https://www.w3schools.com/js/js_intro.asp)

Ale pamiętaj: nie chodzi o to, by teraz rzucić się na czytanie dokumentacji. Wyszukuj tego, czego będziesz potrzebować do rozwiązania danego zadania. Nie zarzucaj się od razu ogromem teorii, spróbuj w praktyce zacząć od małych zagadnień. Zadania z wyzwania właśnie to mają na celu :)

Oczywiście powinnaś/powinieneś się też zaprzyjaźnić z wyszukiwarką. Polecam mocno również filmiki na Youtube, które bardzo często fajnie tłumaczą zagadnienia. Szukaj, próbuj, czytaj, koduj. O to właśnie chodzi w programowaniu!

Rozwiązanie wszystkich zadań znajdziesz pod poniższym adresem:

<https://codepen.io/joannabochynska/pen/dZjJyJ>

Pamiętaj, zajrzyj tam dopiero, gdy sama/sam zrobisz zadania albo zupełnie utkniesz z tematem :)

Powodzenia!

---

## Zadanie 1 - Zmienne

Zaczynamy klasycznie, czyli od zmiennych w JS. Zmienne pozwalają nam przechowywać fragmenty pamięci. Zmienne w JS definiujemy przez użycie słowa kluczowego *var* (od angielskiego *variable*).

Przykładowa zmienna może wyglądać tak:

```
var name = "Wake up and Code";  
console.log(name);
```

Druga linijka wyświetli nam zadeklarowaną zmienną w konsoli.

A teraz zadanie dla Ciebie: zmienne mogą w JS mieć różne typy. Może to być tekst (czyli typ string), mogą to być liczby, albo wartości logiczne i inne. Poszukaj, jakie typy mogą mieć zmienne w JavaScript, a następnie stwórz przykładową zmienną z każdego typu i wypisz ją w konsoli.

Następnie sprawdź (tak dla pewności), jakiego typu są Twoje zmienne używając [operatora typeof](#) i wypisując wynik w konsoli.

---

## Zadanie 2 - Tablice

Tablice pozwalają na przechowywanie wielu danych w jednym miejscu. To właśnie nimi dzisiaj się zajmiemy. Elementy tablicy oddzielamy przecinkami i umieszczamy je w nawiasie kwadratowym. Do poszczególnych elementów tablicy możemy się dostać podając nazwę tablicy oraz indeks elementu w nawiasie kwadratowym. Ważne: indeks liczymy od 0, czyli pierwszy element tablicy ma indeks zero.

Przykładowa tablica:

```
var languages = [ "Polish", "French", "German", "Dutch" ];  
console.log(languages[0]);
```

Druga linijka wypisze w konsoli pierwszy element tablicy, czyli *Polish*.

Zadanie dla Ciebie: stwórz tablicę z imionami pięciu swoich znajomych. Następnie wypisz w konsoli ich imiona odwołując się do odpowiednich indeksów w tablicy.

Poszukaj, w jaki sposób można sprawdzić długość tablicy. Kiedy znajdziesz odpowiedni atrybut, wypisz długość swojej tablicy w konsoli.

Stwórz jeszcze jedną tablicę, tym razem z liczbami. Dodaj w konsoli wartość pierwszego i ostatniego elementu. Chodzi o to, by w konsoli zobaczyć tylko wynik działania. Nie wpisuj jednak liczb ręcznie do *console.log*, ale odwołaj się do nich wskazując odpowiednie indeksy w Twojej tablicy.

---

## Zadanie 3 - Operatory porównania

Czas na operatory porównania. Pewnie je znasz, bo używa się ich choćby podczas lekcji matematyki w podstawówce ;) Jednak w JS jest ich troszkę więcej. Jak sama nazwa wskazuje, operatory porównania służą nam do porównywania wartości. Będziemy mogli wyniki tych porównań wykorzystać później do pisania instrukcji warunkowych.

Przykładowe użycie:

```
var a = 5;  
var b = 10;  
console.log(a > b);
```

W konsoli zobaczymy *false*, ponieważ a nie jest większe od b.

Zadanie dla Ciebie: spójrz na tabelkę wszystkich operatorów w JS [na tej stronie](#). Następnie stwórz kilka zmiennych i przypisz do nich liczby. Wypisz w konsoli wynik porównania różnych zmiennych korzystając po kolei ze wszystkich operatorów wypisanych w podanej wyżej tabeli.

Następnie stwórz następujące zmienne:

```
var x = 2;  
var y = "2";
```

Wypisz w konsoli wynik ich porównania wykorzystując najpierw operator `==`, a potem operator `===`. Czym różnią się wyniki? Dlaczego tak się dzieje?

---

## Zadanie 4 - Instrukcje warunkowe

Wiemy już, jak tworzyć zmienne, tablice oraz jak je porównywać. Dzisiaj pójdziemy dalej, tzn. zajmiemy się instrukcjami warunkowymi. Tworzymy je przez użycie `if`, a następnie w nawiasie okrągłym podajemy warunek. Potem w nawiasie klamrowym wskazujemy, co ma się wykonać, gdy warunek będzie spełniony.

Dla przykładu:

```
var a = 5;
var b = 10;

if (a < b) {
  console.log('a jest mniejsze od b');
} else {
  console.log('a nie jest mniejsze od b');
}
```

Powyższy kod sprawi, że gdy `a` będzie mniejsze od `b`, zobaczymy w konsoli napis *a jest mniejsze od b*. Gdy będzie odwrotnie, zobaczymy *a nie jest mniejsze od b*. *Else* pozwala nam wskazać, co ma się stać, gdy warunek nie będzie spełniony.

Zadanie dla Ciebie: stwórz dwie zmienne i do każdej z nich przypisz liczbę. Następnie napisz instrukcję warunkową: gdy pierwsza zmienna będzie równa drugiej, w konsoli ma się wypisać *Liczby są równe*, w przeciwnym wypadku w konsoli ma się pojawić napis *Liczby są różne*. Zmień teraz wartości przypisane do zmiennych i zobacz, jak to wpłynęło do napis w konsoli.

Sprawdź, w jaki sposób można umieścić w instrukcji warunkowej więcej niż jeden warunek. Chodzi o to, jak sprawić, by dany kod się wykonał, gdy oba warunki będą spełnione albo gdy tylko jeden z nich będzie spełniony.

---

Podpowiem, że musisz poszukać czegoś o operatorach AND i OR. Napisz swoje instrukcje warunkowe wykorzystujące te operatory.



---

## Zadanie 5 - Pętla for

Żeby to wyzwanie pokryło zupełne podstawy JS, nie sposób nie wspomnieć o pętlach. Pętle pozwalają nam wykonać kod pewną ilość razy. Dzięki temu zaoszczędzamy czas i nasz kod staje się optymalny. Pamiętaj, jak tworzyliśmy tablicę, a następnie wypisywaliśmy ich elementy w konsoli po kolei odnosząc się do indeksu każdego z nich? Wyobraź sobie teraz, że mamy tablicę zawierającą 100 elementów. Napisanie `console.log` 100 razy byłoby ogromnie czasochłonne. I tu z pomocą przychodzi pętla.

Najpierw przeczytaj o pętli for [na tej stronie](#), a potem spójrz na kod poniżej i wykonaj zadanie.

Przykład pętli:

```
var friends = [ "Ala", "Marek", "Ela", "Tomek", "Iza" ];
```

```
for (var i = 0; i <= friends.length - 1; i++) {  
    console.log(friends[i]);  
}
```

Powyższy kod wypisze w konsoli po kolei wszystkie elementy tablicy, ponieważ zmienna `i` będzie wzrastać o 1 za każdym razem (to właśnie oznacza `++`, to to samo co `+1`). Już kilka dni zgłębiasz podstawy JS, także tutaj Cię zostawię. Spróbuj poczytać więcej o pętli for, jeśli nie rozumiesz mechanizmu. Na pewno uda Ci się znaleźć świetne przykłady w internecie!

A teraz sama/sam napisz swoją pierwszą pętlę. Pamiętaj tablicę z imionami znajomych z drugiego dnia wyzwania? Patrząc na przykład wyżej, napisz pętlę, która wypisze po kolei imiona w konsoli.

Zrobione? Stwórz teraz pętlę, która 10 razy wypisze w konsoli "Pętle są super!". Powodzenia! :)

---

Wszystkie rozwiązania znajdziesz tutaj:

<https://codepen.io/joannabochynska/pen/dZjJyJ>