

**Kurs** Front-End **Developer**  
JavaScript

# WPROWADZENIE DO JAVASCRIPT

JavaScript (w skrócie JS) jest skryptowym językiem programowania, którego kod wykonywany jest przez silniki implementowane w przeglądarkach internetowych. Pozwala dodać do strony dynamiczne elementy, takie jak animacje, galerie obrazów, efekty pojawiające się po kliknięciu wybranego elementu i wiele innych. Pozwala także w sposób dynamiczny modyfikować strukturę dokumentu HTML, poprzez dodawanie, usuwanie lub modyfikowanie znaczników i atrybutów HTML.

# OSADZANIE KODU JAVASCRIPT

Istnieją dwa sposoby na dodawanie kodu JS do projekt naszej strony internetowej.

Pierwszy z nich, to osadzania kodu bezpośrednio w dokumencie HTML. W takim przypadku nasz kod osadzamy w znaczniku

```
<script></script>
```

```
<script>  
    alert('Alarm!');  
</script>
```

Powyższy kod wyświetli okno alarmu w momencie, gdy przeglądarka wczyta ten fragment kodu podczas renderowania strony internetowej.

# OSADZANIE KODU JAVASCRIPT

Drugim sposobem osadzania kodu JS jest dodanie linku do zewnętrznego pliku JavaScript. Do tego również wykorzystujemy znacznik `<script></script>`, który nie będzie zawierał żadnej treści, będzie miał natomiast atrybut “src” w którym podamy ścieżkę do pliku .js z którego ma korzystać nasza strona internetowa.

```
<script src="js/main.js"></script>
```

Powyższy kod dodany w dokumencie HTML poinformuje przeglądarkę, że ma odładować plik main.js zlokalizowany w folderze “js” i wykonać zawarty w nim skryp.

# OSADZANIE KODU JAVASCRIPT

## Który sposób osadzania kodu JS jest właściwy?

W większości przypadków lepszym rozwiązaniem jest linkowanie do kodu zawartego w zewnętrznych plikach .js. Dzięki temu piszemy nasz kod JS raz i możemy go wykorzystywać na wszystkich podstronach naszego projektu bez potrzeby jego kopiowania. Również jakiekolwiek zmiany w kodzie JS nie będą wymagały modyfikacji kilku dokumentów HTML, a jedynie samego pliku .js.

# PIERWSZY KOD JAVASCRIPT

JavaScript działa tylko w połączeniu z dokumentem HTML. Dlatego stwórz folder o nazwie projekt-js, a w nim plik index.html z poniższym kodem:

```
<!DOCTYPE html>
<html lang="pl">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>
  <h1></h1>
  <script src="js/main.js"></script>
</body>
```

# PIERWSZY KOD JAVASCRIPT

W folderze projekt-js stwórz folder “js”. W folderze “js” stwórz nowy plik tekstowy i zapisz go z nazwą “main” i rozszerzeniem “.js” (main.js).

W pliku main.js dodaj następujący kod i zapisz plik:

```
alert("Alarm!");
```

Otwórz plik index.html w przeglądarce. Po załadowaniu pliku w przeglądarce wyświetli się dodatkowe okno z komunikatem “Alarm!” dodanym w naszym pliku main.js.

# LOGOWANIE W KONSOLI

W części dotyczącej CSS pisaliśmy o narzędziach developerkich wbudowanych w przeglądarkę (<https://developers.google.com/web/tools/chrome-devtools/>).

Do pracy z kodem JS wykorzystujemy wbudowaną w nie konsolę. Po kliknięci prawym przyciskiem myszy na dowolny element na stronie wybieramy opcję “Zbadaj”. Następnie wybieramy zakładkę “Console”. Spróbujmy stworzyć loga, który wyświetli się w konsoli.

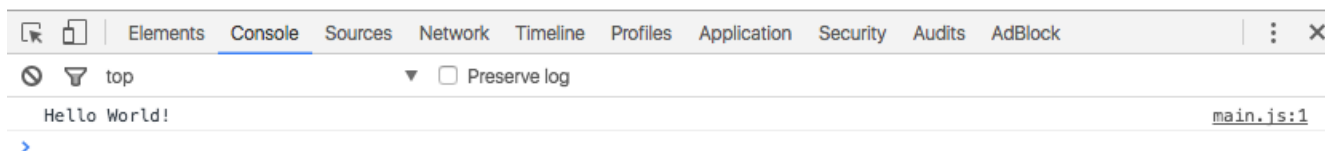


# LOGOWANIE W KONSOLI

Dodajemy kod:

```
console.log("Hello World!");
```

Zapisujemy plik i odświeżamy przeglądarkę. W narzędziach developerskich, w zakładce “Console” powinniśmy zobaczyć tekst “Hello World!”.



# MODYFIKACJA STRUKTURY DOKUMENTU

JS pozwala dynamicznie modyfikować strukturę oraz treść dokumentu HTML. Usuń dodany wcześniej alert oraz console.log. Następnie dodaj kod, który zmodyfikuje treść w znaczniku h1, który dodaliśmy w kodzie naszego dokumentu.

```
var myHeading = document.querySelector('h1');  
myHeading.textContent = 'Hello world!';
```

Następnie zapisz plik i odśwież stronę. W znacznikach h1 pojawiła się treść “Hello World!”.

Pierwsza linia tego kodu tworzy zmienną, do której przypisujemy element HTML jakim jest znacznik nagłówka h1. Pobiera zmienną z naszym nagłów i z wykorzystaniem wbudowanej w JS właściwości “textContent” dodaje do niego zdefiniowaną przez nas treść.

# ZMIENNE W JAVASCRIPT

**Zmienne** - są to elementy przechowujące zdefiniowaną dla nich wartość

Stwórzmy zmienną:

```
var name = "Jan Nowak";
```

składa się ona ze słowa kluczowego “var” (variable), które definiuje zmienną, z nazwy zmiennej (“name”), operatora przypisania ( znak = ) oraz wartości (ciąg znaków “Jan Nowak”). Wykorzystując nazwę zmiennej możemy wywołać jej wartość w kodzie.

Stwórzmy następujący kod:

```
console.log(name);
```

W konsoli naszej przeglądarki zobaczymy zalogowana wartość zmiennej “Jan Nowak”.

# ZMIENNE W JAVASCRIPT

Dane przypisane do zmiennych mogą mieć różny typ.

W JS wyróżniamy pięć typów zmiennych

Tekstowe (string):

`var text = "Jan Nowak";` - wartość zmiennej zapisana jest w cudzysłowie  
"""

Numeryczne (number):

`var number = 12;` - wartość bez cudzysłowu

# ZMIENNE W JAVASCRIPT

Logiczne (boolean)

`var boolean = false;` - przyjmują wartość prawda (true) lub fałsz (false)

Tablicowe (array):

`var array = [1, 2, 3];` - tworzy trzelementową tablicę

Obiektowe (object):

`var object = document.querySelector('h1');` - do zmiennej przypisany jest obiekt, jakim jest selektor h1

# OPERATORY W JAVASCRIPT

**Operator** - symbole matematyczne pozwalające wykonywać operacje na zmiennych

Podstawowe operatory, które powinieneś znać, to:

= - operator przypisania - przypisuje zmiennej wartość

```
var name = "Jan Nowak";
```

+, -, \*, / - operatory matematyczne dodawania, odejmowanie, mnożenie, dzielenie zmiennych typu numerycznego

```
var add = 2 + 5;
```

zmienna “add” przyjmie wartość 7

# OPERATORY W JAVASCRIPT

**==, ===** - operator tożsamości (jest równe) - porównuje, czy zmienna jest równa innej zmiennej

```
var number1 = 1;
```

```
var number2 = 2;
```

**number1 === number2** - takie porównanie zwróci fałsz (false) ponieważ 1 nie jest równe 2

# OPERATORY W JAVASCRIPT

**!=, !==** - operator nierówności (nie jest równe) - porównuje, czy zmienne są różne

```
var number1 = 1;  
var number2 = 2;
```

**number1 !== number2** - takie porównanie zwróci prawdę (true) ponieważ 1 nie jest równe 2



# WARUNKI W JAVASCRIPT

**Warunki** - struktura kodu, która testuje określone wartości i wykonuje kod uzależniony od wyniku testu

```
var number = 2;
```

```
if (number === 2) {  
    console.log("warunek spełniony");  
} else {  
    console.log("warunek niespełniony");  
}
```

Warunek powyższy wykonuje następujący algorytm. Sprawdza, czy zmienna “number” ma wartość równą 2. Jeśli tak (if), to zaloguj w konsoli “warunek spełniony”. Jeśli zmienna “number” nie spełnia tego warunku (else), to zaloguj w konsoli “warunek niespełniony”.

# FUNKCJE W JAVASCRIPT

**Funkcje** - wydzielone bloki kodu przeznaczone do wykonywania konkretnych zadań, które można wielokrotnie wykorzystywać w kodzie.

```
var num1 = 2;  
var num2 = 3;
```

```
function multiply(num1, num2) {  
    var result = num1 * num2;  
    return result;  
}
```

Wywołanie funkcji następuje poprzez wpisanie jej nazwy oraz nawiasów ()

multiply() - Taka funkcja zwróci wynik mnożenia równy 6.

# FUNKCJE W JAVASCRIPT

**Zdarzenia (events)** - struktury kodu, które wychwytyją zdarzenia w przeglądarce i wykonują w odpowiedzi na nie zdefiniowaną akcję

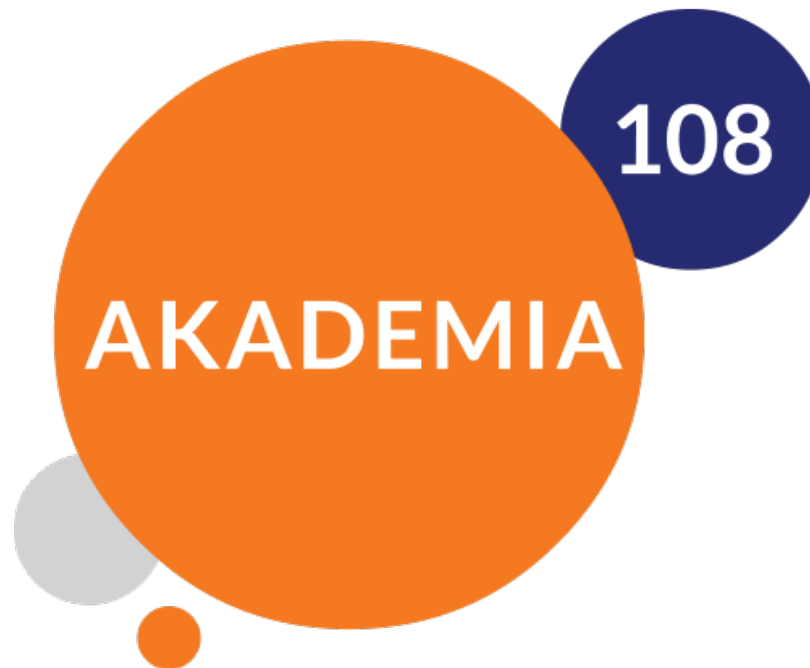
```
document.querySelector('h1').onclick = function() {  
    alert('Właśnie mnie kliknąłeś');  
}
```

Ten kod odpowiada za wykonanie działania po kliknięciu w element h1. Po wychwyceniu takiego zdarzenia wyświetli komunikat alarmu 'Właśnie mnie kliknąłeś'.

# DOKUMENTACJA JAVASCRIPT

Pełną dokumentację języka JavaScript znajdziesz pod linkiem

<http://www.w3schools.com/jsref/default.asp>



Akademia 108

<https://akademia108.pl>