


Projektowanie aplikacji webowych

SEBASTIAN SŁOMIAN
JAVA DEVELOPER AT WSZIB

The background is a dark navy blue. On the left side, there are several parallel teal lines that form a corner-like shape, extending from the top left towards the center. On the bottom right, there are several parallel teal lines that form a diagonal shape, extending from the bottom right towards the center.

JavaScript

PODSTAWY



HTML – warstwa zawartości, szkielet strony internetowej

CSS – warstwa prezentacji strony internetowej

JavaScript – warstwa zachowania strony internetowej

Dołączanie kodu JavaScript

- wewnętrzne head

.html

```
<!doctype html>
<html>
<head>
  <title>Wewnętrzny JS w head</title>
  <script>
    console.log('JavaScript');
  </script>
</head>
<body>
</body>
</html>
```

Dołączanie kodu JavaScript

- wewnętrzne body



```
<!doctype html>
<html>
<head>
    <title>Wewnętrzny JS w body</title>
</head>
<body>
    <input type="button" onclick="console.log('JavaScript');">
</body>
</html>
```

Dołączanie kodu JavaScript

- zewnętrzne



```
<!doctype html>
<html>
<head>
  <title>Wewnętrzny JS w body</title>
  <script src="script.js"></script>
</head>
<body>
</body>
</html>
```

```
console.log('JavaScript');
```

Deklaracja zmiennej

słowo kluczowe
zmiennej



var nazwaZmiennej;



nazwa zmiennej

Przypisanie zmiennej

znak przypisania



zmienna liczbowa

```
var nazwaZmiennej = -5.0;
```

zmienna tekstowa

```
var nazwaZmiennej = 'Tekst';
```

zmienna boolowska

```
var nazwaZmiennej = true;
```


Tablice - deklaracja

tablica liczb

```
var nazwaZmiennej = [1, 2, 3];
```

tablica tekstowa

```
var nazwaZmiennej = ['jeden', 'dwa'];
```

tablice boolowska

```
var nazwaZmiennej = [true, false];
```

Tablice – dostęp do n-go elementu tablicy

0 1 2

```
var nazwaZmiennej = ['jeden', 'dwa', 'trzy'];
```

`nazwaZmiennej[2]` element 'trzy'

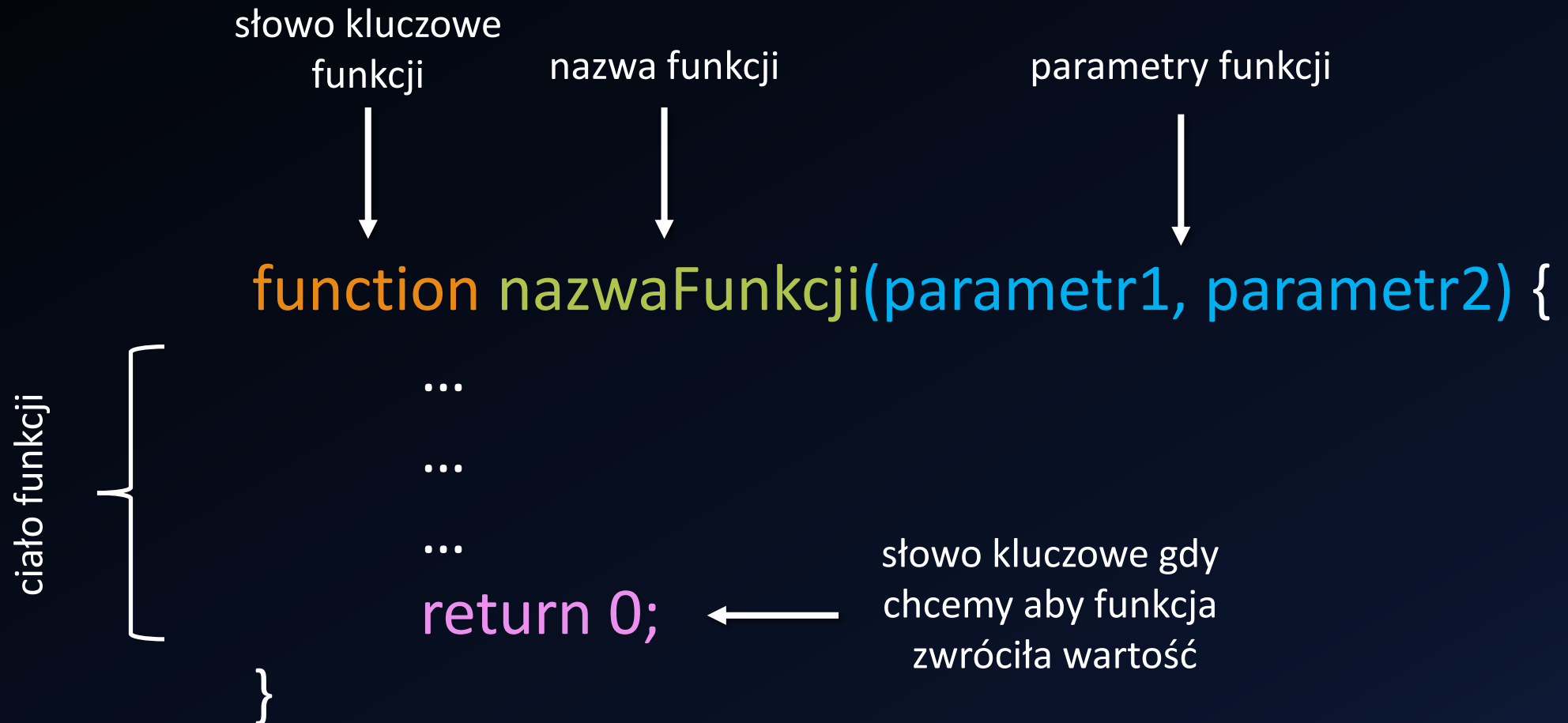


wyraz tablicy
od 0 do n-1

Ilość elementów
w tablicy

`nazwaZmiennej.length`

Funkcje - deklaracja



Funkcje - wywołanie

```
nazwaFunkcji(parametr1, parametr2);
```

Obiekty

zmienna --> właściwość obiektu

funkcja --> metoda obiektu

Tworzenie obiektów

- poprzez notacje literałów

```
var osoba = {  
  imie: 'Jan',  
  nazwisko: 'Nowak',
```

}

właściwości

metoda

{

```
  idz: function() {  
  }
```

```
}
```

Tworzenie obiektów

- poprzez notacje konstruktora

```
function Osoba(imie, nazwisko) {  
    this.imie = imie;  
    this.nazwisko = nazwisko;  
}
```

właściwości

metoda {
 this.idz = function() {
 }
}

słowo kluczowe
dla nowych obiektów

wywołanie konstruktora

```
var osoba = new Osoba('Jan', 'Kowalski');
```

Dostęp do właściwości i metod obiektu

```
var imie = osoba.imie; | osoba.imie = 'Jan';
```

```
osoba.idz();
```


Operatory

- przypisania

zmienna = 'Przykładowy tekst';

- porównania

==

!=

===

!==

>

<

>=

<=

- arytmetyczne

+

-

/

*

++

--

%

Operatory

- logiczne

&&

||

!

- ciągów tekstowych

```
zmiennaTekstowa = 'Jeden, ' + 'dwa';
```



JavaScript

DECYZJE I PĘTLE



IF

warunek



```
if (i == 0) {
```

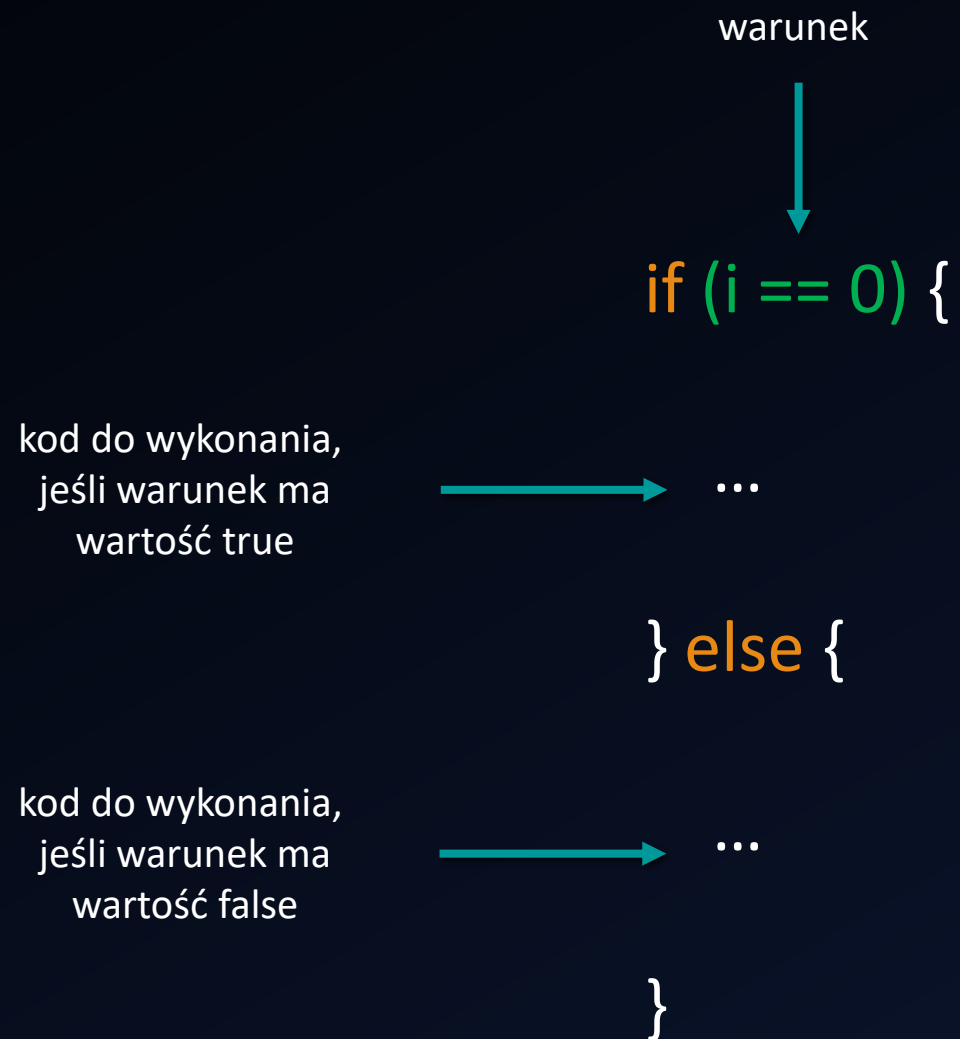
kod do wykonania,
jeśli warunek ma
wartość true



...

```
}
```

IF-ELSE



SWITCH - CASE

kod do wykonania,
jeśli zmienna ma
wartość jeden



```
switch (zmienna) {
```

```
  case 'jeden':
```

```
    ...
```

```
    break;
```

kod do wykonania,
jeśli zmienna ma
wartość dwa



```
  case 'dwa':
```

```
    ...
```

```
    break;
```

kod do wykonania,
jeśli zmienna nie
ma żadnej z powyższych
wartości



```
  default:
```

```
    ...
```

```
    break;
```

```
}
```

FOR

warunek (licznik)



```
for (var i = 0; i < 10; i++) {
```

kod do wykonania
w trakcie pętli



...

```
}
```

WHILE

warunek



```
while (true) {
```

kod do wykonania
dopóki warunek
jest spełniany



```
...
```

```
}
```


DO - WHILE

kod wykonany przynajmniej
jeden raz i dopóki warunek
jest spełniony



do {

...

} while (true);



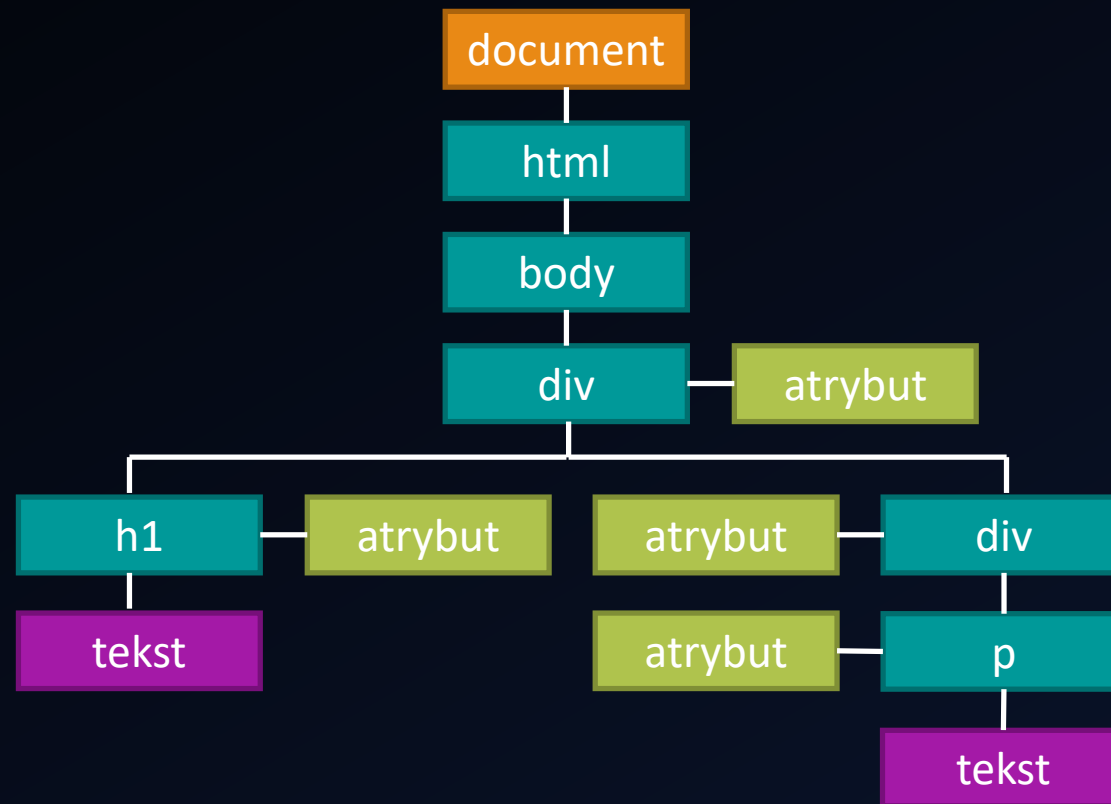
warunek



JavaScript

DOM – OBIEKTOWY MODEL
DOKUMENTU

```
<html>
<body>
  <div>
    <h1>Nagłówek</h1>
    <div>
      <p>Paragraf</p>
    </div>
  </div>
</body>
</html>
```



document	węzeł document
html	węzeł elementów
atribut	węzeł atrybutów
tekst	węzły tekstowe

Dostęp do DOM

- wybór konkretnego węzła elementu

wybór elementu za pomocą id

```
document.getElementById('idElementu');
```

wybór elementu za pomocą selektora css

```
document.querySelector('h1');
```

zwracany jest pierwszy element

Dostęp do DOM

- wybór wielu elementów

wybór elementów za pomocą klasy

```
document.getElementsByClassName('klasaElementu');
```

wybór elementów za pomocą nazwy znacznika

```
document.getElementsByTagName('nazwaZnacznika');
```

wybór wszystkich elementów za pomocą selektora css

```
document.querySelectorAll('h1');
```

Dostęp do DOM

- poruszanie się między elementami

wybór rodzica elementu

`.parentNode`

wybór poprzedniego/następnego elementu na tym samym poziomie

`.previousSibling`

`.nextSibling`

wybór pierwszego/ostatniego dziecka elementu

`.firstChild`

`.lastChild`

Edycja zawartości węzła

- `nodeValue`

```
document.getElementById('id').nodeValue = 'Nowa wartość';
```

- `textContent`

```
document.getElementById('id').textContent = 'Nowa wartość';
```

- `innerHTML`

```
document.getElementById('id').innerHTML = '<b>Wartość</b>';
```

Dodawanie elementów do DOM

1)

```
var nowyElement = document.createElement('div');
```

2)

```
var nowyWezelTekstowy = document.createTextNode('tekst');
```

3)

```
nowyElement.appendChild(nowyWezelTekstowy);
```

4)

```
document.getElementById('id').appendChild(nowyElement);
```


Usuwanie elementów z DOM

1)

```
var elementDoUsuniecia = document.getElementById('idDziecka');
```

2)

```
var elementNadrzędny = document.getElementById('idRodzic');
```

3)

```
elementNadrzędny.removeChild(elementDoUsuniecia);
```

Atrybuty elementów

- dodanie atrybutu

document.getElementById('id').setAttribute('hidden');

nazwa atrybutu



- usunięcie atrybutu

document.getElementById('id').removeAttribute('hidden');

nazwa atrybutu



- sprawdzenie czy element posiada atrybut

document.getElementById('id').hasAttribute('hidden');

nazwa atrybutu



The background is a dark navy blue. On the left side, there are several parallel teal lines that start from the top and extend downwards, with some lines curving slightly. On the bottom right, there are several parallel teal lines that start from the bottom and extend diagonally upwards towards the right.

JavaScript

ZDARZENIA

Obsługa zdarzeń

- w atrybutach HTML



.html

...

```
<button onclick='clickMe()>Kliknij</button>
```

...



.js

```
function clickMe() {  
    console.log('Przycisk został kliknięty');  
}
```

Obsługa zdarzeń

- w modelu DOM

.html

...

```
<button id='myButton'>Kliknij</button>
```

...

.js

```
function clickMe() {  
    console.log('Przycisk został kliknięty');  
}
```

```
var element = document.getElementById('myButton');  
element.onclick = clickMe;
```



zdarzenie

Obsługa zdarzeń

- obserwator zdarzeń

element.addEventListener('click', clickMe);

zdarzenie

kod wywoływany
po zdarzeniu

The diagram illustrates the parameters of the `addEventListener` method. A teal arrow points upwards from the label 'zdarzenie' (event) to the string 'click' in the code. Another teal arrow points downwards from the label 'kod wywoływany po zdarzeniu' (code called after the event) to the function name 'clickMe' in the code.

Obsługa zdarzeń

- obserwator zdarzeń

.html

...

```
<button id='myButton'>Kliknij</button>
```

...

.js

```
function clickMe() {  
    console.log('Przycisk został kliknięty');  
}
```

```
var element = document.getElementById('myButton');  
element.addEventListener('click', clickMe);
```

The image features a dark blue background with abstract teal-colored lines. In the top-left corner, there are several parallel lines that form a corner-like shape. In the bottom-right corner, there are several parallel lines that form a diagonal shape.

jQuery

jQuery dołączanie biblioteki

```
<script  
  src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">  
</script>
```

jQuery

selektor

`$('.klasa1, .klasa2')`

obiekt jQuery