

Hello World

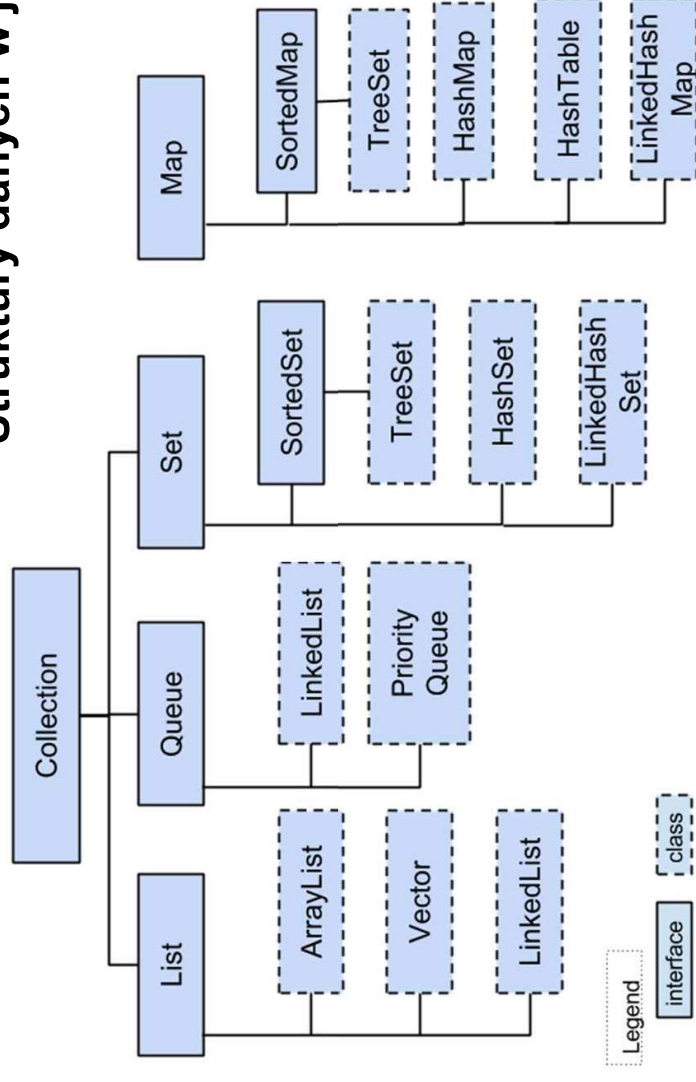


Prowadzący
Tomasz Nastały

| Struktura danyeh

Struktury danych

Struktury danych w języku JAVA



Struktury danych

Struktury danych w języku JavaScript

- Objects
- Arrays

Nowe w ES6

- Set
- Map
- WeakSet
- WeakMap

Jest możliwość ręcznego implementowania innych struktur

<https://github.com/benoitvallon/computer-science-in-javascript/tree/master/data-structures-in-javascript>

Obiekt

Proporcje obiektu (property)

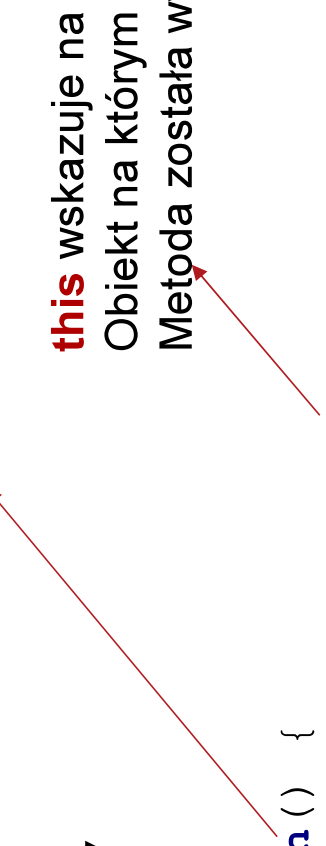
```
var person = {  
  firstName: 'John',  
  lastName: 'Doe',  
  age: 50,  
  eyeColor: 'blue'  
};
```

Obiekt

Funkcję w obiekcie nazywamy
Metodą obiektu

```
var person = {  
  firstName: 'John',  
  lastName: 'Doe',  
  age: 50,  
  eyeColor: 'blue',  
  sayHello: function() {  
    return 'Hello, my name is ' + this.firstName;  
  }  
};
```

this wskazuje na
Obiekt na którym
Metoda została wywołana

A red arrow originates from the text 'this wskazuje na Obiekt na którym Metoda została wywołana' and points to the 'this' keyword in the code snippet, specifically within the 'sayHello' function's return statement.

This w obiekcie

```
var person = {  
  firstName: 'John',  
  sayHello: function () {  
    return 'Hello, my name is ' + this.firstName;  
  }  
};
```

*This obiektem na lewo od kropki
przed .sayHello() czyli this.firstName
Został zastąpiony person.firstName
Czyli John!*

```
person.sayHello(); // 'Hello, my name is John'  
person.firstName // 'John'
```

Metoda woła metodę

```
var person = {  
  firstName: 'John',  
  sayHello: function() {  
    return 'Hello, my name is ' + this.firstName;  
  },  
  sayLovelyHello: function(who) {  
    return this.sayHello() + ' i love you!!'  
  }  
};
```


Metoda woła metodę

```
var person = {  
  firstName: 'John',  
  sayHello: function() {  
    return 'Hello, my name is ' + this.firstName;  
  },  
  sayLovelyHello: function() {  
    return this.sayHello() + ' i love you!!'  
  }  
};  
  
var funnyPerson = {  
  firstName: 'Klaun',  
  sayFunnyHello: person.sayHello  
};  
  
funnyPerson.sayFunnyHello(); // Hello, my name is Klaun'
```

Usuwanie kluczy

```
var person = {  
  firstName: 'John'  
};
```

```
delete person.firstName;
```

Klucz firstName zostanie usunięty!

ZADANIE

1. CREATING OBJECTS.JS

info **Share**
<academy/>



Metoda Object.keys(objekt)

```
var superman = {  
  origin: 'Krypton',  
  alterEgo: 'Klark Kent',  
  speed: 300,  
  strength: 200  
};
```

```
Object.keys(superman)  
// -> ['origin', 'alterEgo', 'speed', 'strength']
```

Referencja

```
var superman = {  
  origin: 'Krypton',  
  alterEgo: 'Klark Kent',  
  speed: 300,  
  strength: 200  
};
```

```
var supermanV2 = superman;  
supermanV2.alterEgo = 'Joker';  
superman.alterEgo // Joker a nie Klark Kent!
```

Pozybicie się referencji Object.assign({}, obiekt)

```
var superman = {  
  origin: 'Krypton',  
  alterEgo: 'Klark Kent',  
  speed: 300,  
  strength: 200  
};
```

```
var supermanV2 = Object.assign({}, superman);  
supermanV2.alterEgo = 'Joker';  
superman.alterEgo // Uff, Klark Kent
```

Złączenie dwóch obiektów

```
var superman = {  
  origin: 'Krypton',  
  alterEgo: 'Klark Kent',  
};  
  
var powers = {  
  speed: 300,  
  strength: 200  
};
```

```
Object.assign(superman, powers); // superman zmodyfikowany
```

Pętla po kluczach obiektu

```
for (var key in object) {  
    console.log(key);  
    console.log(object[key]) // value  
}
```


Porównanie dwóch obiektów po wartościach

```
{ a: 3 } === { a: 3 } // false
```

Brak w JS wbudowanych mechanizmów do porównania dwóch obiektów (porównywane po referencji).

W takim przypadku najlepiej skorzystać z biblioteki np. Lodash

<https://lodash.com/docs#isEqual>