

JavaScript

Laboratorium 3

Cechy języka

- Dynamiczne typy danych
 - typ związany z wartością a nie zmienną
- Obiektość
 - obiekty jako tablice asocjacyjne
 - funkcje jako obiekty pierwszego rzędu
 - dziedziczenie przez prototypy, brak klas
- Wykonywanie
 - zewnętrzne środowisko
 - automatyczne dopisywanie średników

Zmienne i funkcje

- Zmienne

```
let x = 5;
```

- Funkcje

```
//Deklaracja  
function a(p) {  
    ...  
}  
//lub  
const a = function(p) {  
    ...  
}  
  
//Wywołanie  
a(5); //OK  
a("asdf"); //OK  
a(); //OK
```

Typy

- Obiekt
 - Kolekcja par <pole, wartość>
- Boolean
 - Wartości: **true/false**
- Undefined
 - Wartości: **undefined**
- Null
 - Wartości: **null**
- Number
 - Wartości: 64 bit, float
- String
 - Wartości: **'tekst'** lub **"tekst"**

Typy – na co uważać

- Type coercion

- `4 == "4"` → `true`

- `4 === "4"` → `false`

- Boolean

- `false` = `false`, `null`, `undefined`, `""`, `0`, `NaN`

- `true` = `true`, `"asdf"`, `1`, `-1`, `"false"`, ogólnie –
wszystko co nie jest `false`.

Tablice

```
let t = new Array();  
t[0] = 23;  
t[1] = "asdf";  
t[2] = function(x) { return x; };  
t[100] = { imię: "Maciej" };  
  
console.log(t[2](5));  
console.log(t[100].imię);
```

Petle

```
const t = [5, 10, 15];

for (let i = 0; i < t.length; i++) {
  console.log(t[i]);
}

for (let i in t) {
  console.log(t[i]);
}

t.forEach((e) => { console.log(e); });

for (let e of t) {
  console.log(e);
}

for (let [i, e] of t.entries()) {
  console.log(`t[${i}] = ${e}`);
}
```

DOM

window

- Kontekst globalny

document

- Drzewo dokumentu
- Przydatne metody

```
getElementById("id")
```

```
getElementsByTagName("div")
```

```
querySelector(".klasa div")
```


Przydatne materiały

- <http://speakingjs.com/es5/>
- <http://exploringjs.com/es6/index.html>
- <http://exploringjs.com/es2016-es2017/>
- <https://developers.google.com/web/fundamentals/primers/promises>
- <https://medium.com/javascript-scene/master-the-javascript-interview-what-is-a-promise-27fc71e77261>