

# 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

```
var x = 5;
```

- Funkcje

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

# Scoping

- Dwa rodzaje zasięgu:
  - Globalny
  - Funkcji
- Łańcuch zasięgu

```
var x = 1;
f1();

function f1() {
  var x = 2;
  f2();
}

function f2() {
  console.log(x);
}
```

# 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`.

# Obiekty

- Deklaracja

```
var mójObiekt = new Object();  
mójObiekt.pole = 5;  
mójObiekt.metoda = function(x) {  
    return Math.pow(x, 2);  
};  
//lub  
var mójObiekt = {  
    pole: 5,  
    metoda: function(x) {  
        return Math.pow(x, 2);  
    }  
};
```

- Wykorzystanie

```
console.log(mójObiekt.pole);  
console.log(mójObiekt.metoda(2));  
//lub  
console.log(mójObiekt["pole"]);  
console.log(mójObiekt["metoda"](2));
```

# Obiekty – konstruktor

- Deklaracja

```
function Koło(promień) {  
    this.promień = promień;  
    this.getPole = function() {  
        return Math.PI * Math.pow(this.promień, 2);  
    };  
}
```

- Wykorzystanie

```
var noweKoło = new Koło(10);  
console.log(noweKoło.getPole());
```

- Prototyp

```
function Koło(promień) {  
    this.promień = promień;  
}  
Koło.prototype.getPole = function() {  
    return Math.PI * Math.pow(this.promień, 2);  
};
```

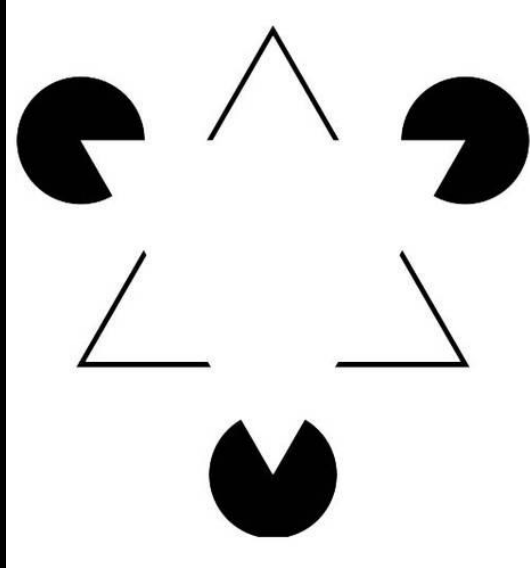


# Tablice

```
var 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);
```

```
var t = ["Maciej", "Mieciu", "Heniu"];  
  
for (var i = 0; i < t.length; i++) {  
    console.log(t[i]);  
}  
  
for (var i in t) {  
    console.log(t[i]);  
}
```

# Domknięcie



```
function pomnozPrzez(mnoznik) {  
  return function(x) {  
    return x * mnoznik;  
  };  
}  
  
var podwojenie = pomnozPrzez(2);  
  
console.log(podwojenie(5));
```

# DOM

## window

- Kontekst globalny

## document

- Drzewo dokumentu
- Przydatne metody

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

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

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