

# Programowanie w JavaScript

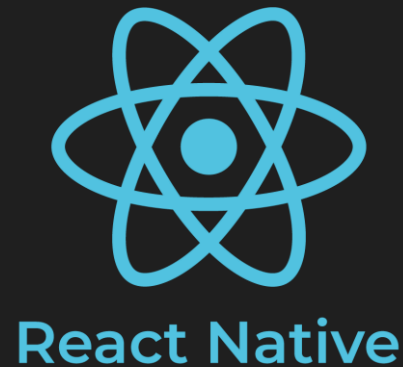
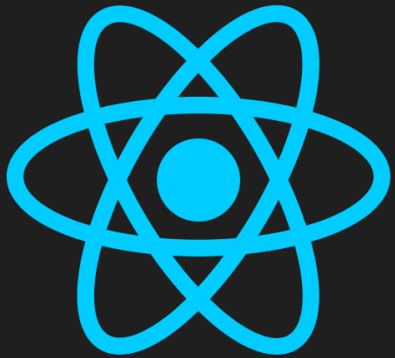


**WSEI**  
#szkoła programowania





- Powstał w 1995 roku
- JavaScript to nie JAVA!
- Na początku służył jedynie do interakcji z użytkownikiem
- Jest najbardziej wszechstronnym językiem programowania





DEVTOOLS

W JavaScript możemy dodawać dwa rodzaje komentarzy:

- liniowy
- blokowy

```
// to jest komentarz liniowy
```

```
/**  
 * To jest komentarz blokowy  
 * Jeśli chcemy dodać więcej niż jedną linijkę kodu  
 */
```

W normalnym języku każde zdanie oddzielamy kropką. W JavaScript mamy analogicznie z tym że każde zdanie to instrukcja a zamiast kropki mamy średnik. **Instrukcją** nazywamy taki kawałek kodu który spełnia jakieś określone zadanie.

```
console.log('Hello World!') // Prawie dobrze
```

```
console.log('Hello World!'); // Idealnie
```

```
console.log(  
  'Hello World!'  
); // też może być ale brzydko wygląda
```

```
console.log('Hello World!') console.log('Hello World!') // Baaaardzo źle!!!
```



**Język programowania** – zbiór zasad opisujących instrukcje budujące program. Jest jednoznaczny oraz precyzyjny,

**Zmienna** - symbol oznaczający wielkość która przyjmuje różne wartości

**Stała** – symbol oznaczający wielkość która jest nie zmienna

**Tablica** – typ danych który pozwala na przechowywanie wielu różnych wartości

**Pętla** – fragment kodu który odpowiada za powtarzanie pewnej czynności do momentu wystąpienia warunku wyjścia

**Algorytm** – skończony ciąg jasno zdefiniowanych czynności koniecznych do wykonania pewnego rodzaju zadań,

**Pseudokod** – luźna notacja kodu oraz zwykłego języka mająca na celu szybki zapis konceptu,

**Kod źródłowy** – algorytm rozwiązujący konkretny problem zapisany w konkretnym języku programowania

**Funkcja** – jest zbiorem wyrażeń które wykonują określone zadanie. Aby użyć funkcji należy ją najpierw zdefiniować.



- Każda zmienna poprzedzona jest słówkiem kluczowym **var** lub **let**
- Każda stała poprzedzona jest słówkiem kluczowym **const**
- Nazwa musi być **krótka, samoopisująca się, po angielsku**
- Nazwa musi zaczynać się od litery
- Wielkość liter w nazwie zmiennej ma znaczenie!

```
var lorem = 10;
```

```
let ipsum = 'Ala ma kota';
```

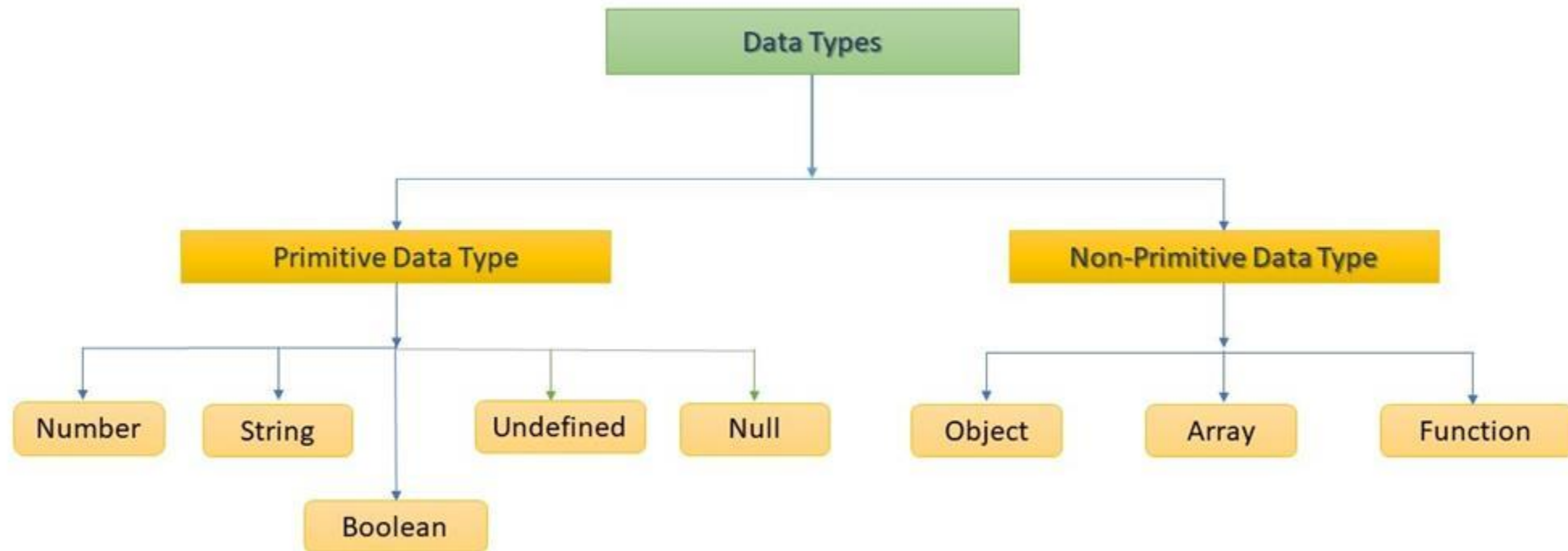
```
const array = ['ala', 'ma', 'kota'];
```

- Każda zmienna poprzedzona jest słówkiem kluczowym **var** lub **let**
- Każda stała poprzedzona jest słówkiem kluczowym **const**
- Nazwa musi być **krótka, samoopisująca się, po angielsku**
- Nazwa musi zaczynać się od litery
- Wielkość liter w nazwie zmiennej ma znaczenie!

```
var lorem = 10;
```

```
let ipsum = 'Ala ma kota';
```

```
const array = ['ala', 'ma', 'kota'];
```



Non-zero value



null



0



undefined



Equality(==)	null	undefined	false	""	0	NaN
null	=	=	≠	≠	≠	≠
undefined	=	=	≠	≠	≠	≠
false	≠	≠	=	=	=	≠
""	≠	≠	=	=	=	≠
0	≠	≠	=	=	=	≠
NaN	≠	≠	≠	≠	≠	≠

- **Null** -> reprezentuje wartość pustą ale nie oznacza zera
- **Undefined** -> wartość niezdefiniowana.

**Null** używamy jeśli chcemy przypisać wartość pustą, **undefined** jest informacją że zmienna nie ma żadnej wartości.

- Wartości liczbowe są 64-bitowe, zmiennoprzecinkowe
- Obliczenia w JS nie są zbyt dokładne
- Na liczbach możemy wykonywać działania matematyczne

```
const foo = 30;
```

```
const bar = 30.5;
```

```
30 === 30.0; // true
```

```
0.1 + 0.2 = 0.30000000000000004;
```

```
(0.1 + 0.2).toFixed(2) === (0.3).toFixed(2);
```

+ (dodawanie)

- (odejmowanie)

/ (dzielenie)

\* (mnożenie)

% (modulo - reszta z dzielenia)

- Stringi to ciągi znaków
- Dodawanie stringów nazywamy konkatencją – czyli łączenie łańcuchów

```
const text = 'Ala ma kota';  
const buttonText = 'Click me';  
const stringNumber = '20';  
const number = 20;
```

```
text + 'Puszka' // Ala ma kota Puszka  
stringNumber + 20 // 2020  
stringNumber + '20' // 2020  
number + 20 // 40
```

- Wartości logiczne (**boolean**) to wartości reprezentujące prawdę (**true**) oraz fałsz (**false**)
- Falsy values czyli wartości które przy konwersji na wartość logiczną dają zawsze fałsz

```
const checked = true;  
const clicked = false;
```

```
!clicked // true
```

```
/** FALSY VALUES */
```

```
0 // false
```

```
' ' // false
```

```
null // false
```

```
undefined // false
```

```
NaN // false
```

- Aby dowiedzieć się jakiego typu jest zmienna wystarczy użyć **operatora typu typeof()**

```
let number = 10;  
let text = "u mnie działa";  
let logic = true;
```

```
typeof(number); // number  
typeof(text); // string  
typeof(logic); // boolean
```





- **Operatory** – to znaki służące do operowania na zmiennych
- Wyróżniamy:
  - Operatory arytmetyczne
  - Operatory logiczne
  - Operatory przypisywania
  - Operatory porównania
  - Operatory działań na napisach

- Operatory arytmetyczne służą nam do operacji matematycznych na zmiennych

Operator	Opis
+	dodawanie
-	odejmowanie
*	mnożenie
/	dzielenie
%	reszta z dzielenia (modulo)

- Zwracają zawsze typ boolean. Służą do porównywania zmiennych ze sobą

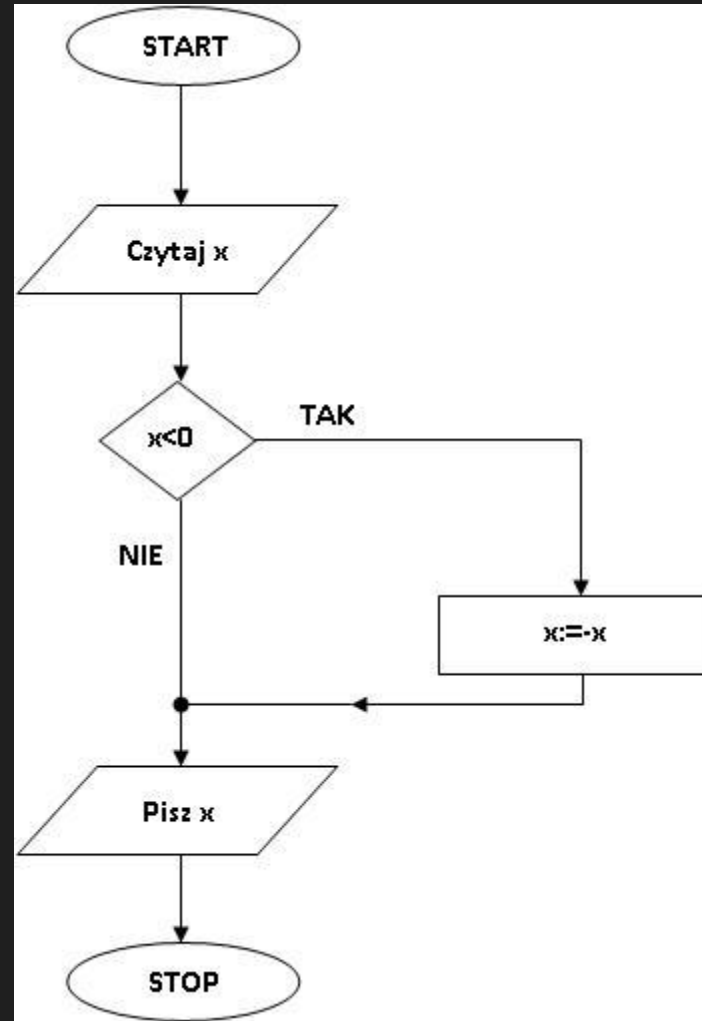
Operator	Opis	Przykład
<code>==</code>	równe sobie (ta sama wartość, ale mogą mieć różne typy)	<code>2 == "2" // true</code>
<code>===</code>	równe sobie i mające ten sam typ	<code>4 === "4" // false</code>
<code>!=</code>	nierówne	<code>6 != 12 // true</code>
<code>!==</code>	nierówne i/lub różnego typu	<code>4 !== "4" // true</code>
<code>&gt; &gt;=</code>	większe niż (i większe równe niż)	<code>12 &gt; 10 // true</code> <code>12 &gt;= 12 // true</code>
<code>&lt; &lt;=</code>	mniejsze niż (i mniejsze równe niż)	<code>10 &lt; 12 // true</code> <code>10 &lt;= 10 // true</code>

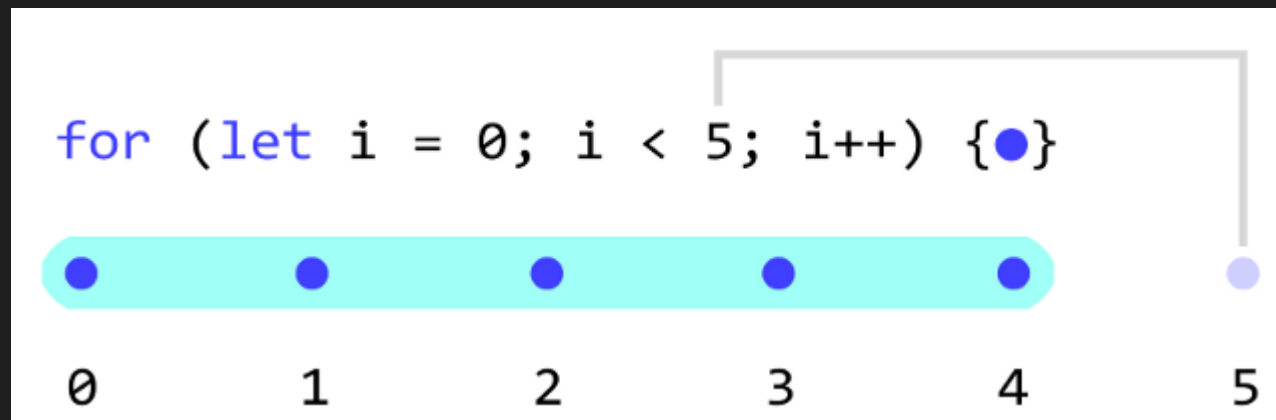
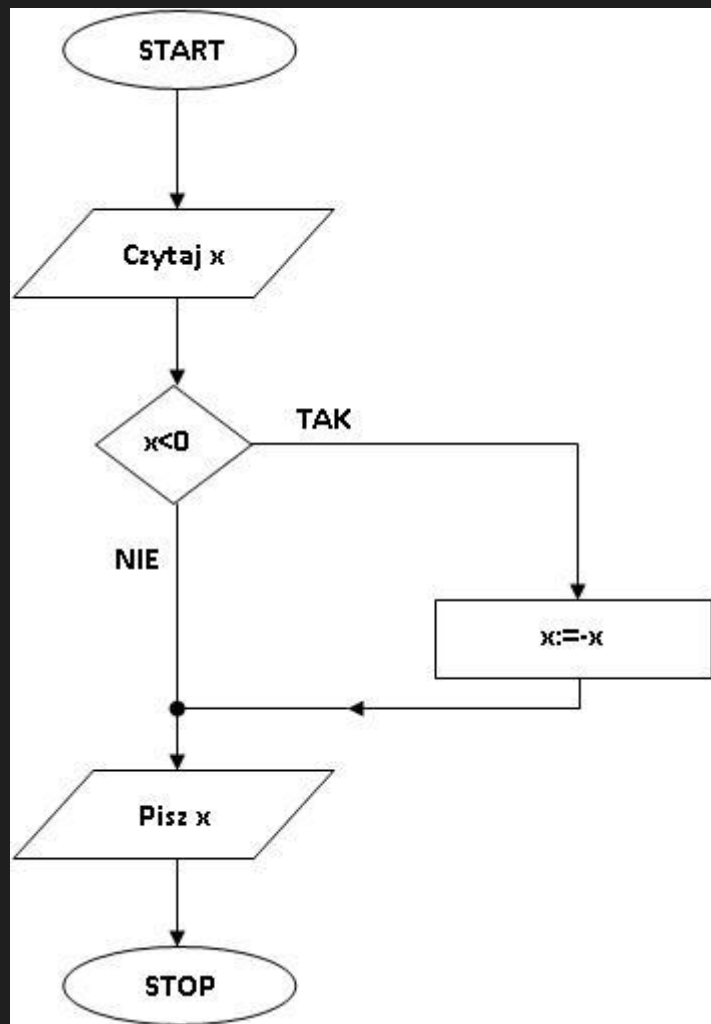
- Operator = przypisuje wartość do zmiennej lub stałej.
- Operator = może również zmieniać wartość zmiennej poprzez ponowne przypisanie
- Inkrementacja to zwiększanie zmiennej o jeden
- Dekrementacja to zmniejszanie zmiennej o jeden

```
const text = 'Ala ma kota';  
const buttonText = 'Click me';  
let numberOne = 1;
```

```
text = 'Janek ma kota';  
++numberOne; // inkrementacja  
--numberOne; // dekrementacja
```









```
const someData = 5;  
if (typeof someData === 'number') {  
    console.log('zmienna jest typu number')  
} else if (typeof someData === 'string') {  
    console.log('zmienna jest typu string')  
} else {  
    console.log('zmienna jest jakaś inna')  
}
```

```
const someData = 3;  
switch(someData) {  
    case 1: {  
        console.log('liczba 1');  
        break;  
    }  
    case 2: {  
        console.log('liczba 1');  
        break;  
    }  
    case 3: {  
        console.log('liczba 1');  
        break;  
    }  
    case 4: {  
        console.log('liczba 1');  
        break;  
    }  
    default: {  
        console.log('jakaś inna liczba')  
    }  
}
```

```
for (var i = 0; i <= 10; i = i + 1) {  
    console.log(i);  
}
```

```
// 0 1 2 3 4 5 6 7 8 9 10
```

```
for (var i = 10; i >= 0; i = i - 1) {  
    console.log(i);  
}
```

```
// 10 9 8 7 6 5 4 3 2 1 0
```

```
for (var i = 0; i < 3; i++) {  
    for (var j = 0; j < 4; j++) {  
        console.log("i=" + i + ", j=" + j);  
    }  
}
```

```
// niezależna
```

```
for (var i = 0; i < 4; i++) {  
    for (var j = i; j < 4; j++) {  
        console.log("i=" + i + ", j=" + j);  
    }  
}
```

```
// zależna
```

# BREAK

```
for (var i = 0; i < 3; i++) {  
    var result = Math.floor(Math.random() * 10);  
    if (result === 5) {  
        break;  
    }  
}
```

