

JS: liczby

Liczby zapisujemy
bez cudzysłowu / apostrofu.

`typeof 5 === "number"`

W JS-ie wszystkie liczby są
tego samego typu: **number**.

$$0.1 + 0.2 \neq 0.3$$

drobna uwaga na temat liczb zmiennoprzecinkowych

Nieskończoność

Infinity lub -Infinity
np. kiedy dzielimy przez 0

NaN

- Not A Number
- liczba, która nie jest liczbą
- może powstać na wiele sposobów
- każda operacja arytmetyczna z NaN da NaN
- NaN !== NaN
- Number.isNaN()

Podstawowa matematyka

JS daje nam wiele sposobów na manipulowanie liczbami

Zaokrąglanie

- **Math.round(liczba)** - do najbliższej liczby całkowitej
- **Math.ceil(liczba)** - w górę
- **Math.floor(liczba)** - w dół

Konkretna liczba miejsc po przecinku

- `5.12345.toFixed(2)` // "5.12"
- zwraca łańcuch znaków

Pierwiastek kwadratowy

- `Math.sqrt(16) // 4`
- `Math.sqrt(-5) // NaN`

Najmniejsza i największa wartość

- `Math.max(1, 2, 3) // 3`
- `Math.min(1, 2, 3) // 1`
- Dowolna liczba argumentów

Losowa liczba

- **Math.random()**
- pseudo-losowa liczba od zero do “prawie 1”
 - np. **0.3796176827826454**
- **Math.floor(Math.random() * (5))**
 - losowa liczba całkowita od **0** do **4**

Konwertowanie na liczbę

"10" + 0 === "100"

Number()



```
1 Number( "5" )           // 5
2 Number( "2 jabłka" )    // NaN
```

+



```
1  + "5"           // 5
2  + "2 jabłka"    // NaN
```


działania arytmetyczne inne niż dodawanie

```
1 "5" * 1 // 5
2 " 5 " - 3 // 2
3 "2 jabłko" / 2 // NaN
```

parseInt(), parseFloat()

```
1 parseInt("5")           // 5
2 parseInt("5 6")         // 5
3 parseInt("2 jabłka")    // 2
4 parseInt("jabłka")      // NaN
5 parseInt("kurki 3")     // NaN
```

parseFloat - analogicznie dla liczb z cyframi po przecinku

operatory arytmetyczne

- + - * /
- % - reszta z dzielenia (modulo)
- ** - potęgowanie (IE)

kolejność operatorów

- priorytety operatorów są ściśle zdefiniowane
- mnożenie i dzielenie ważniejsze niż dodawanie i odejmowanie
- kolejność działań możemy zmienić, dodając nawias

Inkrementacja i dekrementacja

- ++, --
- dodaje lub odejmuje 1
- może wystąpić **przed** lub **po** liczbą
- działa tylko ze zmiennymi (stałymi)

Operatory przypisania

- =
- +=, -=, *=, /=, %=, **=

Operatory porównania

- `===` - identyczne, ta sama wartość i typ
- `!==` - nie identyczne
- `<, >` - wiadomo
- `<=, >=` - mniejsze bądź równe, większe bądź równe
- `==, !=` - ta sama wartość, typ bez znaczenia

Linki

Specyfikacja

- <https://tc39.es/ecma262/#sec-ecmascript-language-expressions>
- <https://tc39.es/ecma262/#sec-math-object>
- <https://tc39.es/ecma262/#sec-number-objects>

MDN

- <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators>
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Number