# Funkcje wbudowane

## parseInt()

```
parseInt(str [, radix])
```

**parseInt** analizuje swój pierwszy argument, łańcuch znaków str i próbuje zwrócić liczbę całkowitą o podstawie wskazanej przez drugi, opcjonalny argument **radix**.

Na przykład,

- podstawa o wartości 10 wskazuje konwersję do liczby dziesiętnej,
- podstawa równa 8 do liczby ósemkowej,
- 16 do heksadecymalnej itd.

Dla podstawy większej od 10 litery alfabetu wskazują liczby większe od 9. Na przykład, dla liczb heksadecymalnych (podstawa 16), używane są litery od A do F.

### Przykład

```
parseInt("F", 16); Wyświetli nam piętnaście (F w szesnastkowym to 15 w dziesiątkowym)
```

| Funkcja | Wynik |
|---|---|
| parseInt("123"); | 123 |
| parseInt("123AB",16); | 74667 |
| parseInt("123",8); | 83 |
| parseInt("123AB",8); | NaN |
| parseInt("0377"); | 255 |
| parseInt("0x373"); | 883 |

## parseFloat()

```
parseFloat(str)
```

gdzie **parseFloat** analizuje swój argument, łańcuch znaków str i próbuje zwrócić liczbę zmiennoprzecinkową.

W razie napotkania symbolu innego niż znak (+ lub -), liczby (0-9), znaku dziesiętnego lub wykładnika, funkcja zwraca wartość do momentu jego wystąpienia ignorując sam symbol i wszystkie inne po nim następujące.

**Jeśli pierwszy znak nie może być przekonwertowany do liczby, zwrócona zostaje wartość “NaN” (nie liczba).**

## Przykład

```
parseFloat("314e-2"); wyświetli nam 3.14
```

## isNaN()

```
isNaN(testowanaWartość)
```

**isNaN** jest funkcją najwyższego rzędu i nie jest przypisana do żadnego obiektu.

Funkcje parseFloat i parseInt zwracają NaN, kiedy wyliczą wartość, która nie jest liczbą. isNaN zwraca true, jeśli przekazano jej NaN, a false w przeciwnym wypadku.

Funkcja ta jest o tyle przydatna, że wartości NaN nie można skutecznie sprawdzać przy użyciu operatorów równości.

**x == NaN i x === NaN mają zawsze wartość false, bez względu na to, jaką wartość ma x, nawet jeśli x to NaN.**

Na przykład, zarówno 1 == NaN, jak i NaN == NaN zwracają false.

| Funkcja | Wynik |
|---|---|
| `isNaN(NaN)` | true |
| `isNaN(567)` | false |
| `isNaN(37.2)` | false |
| `isNaN(parseInt("zx23"))` | true |

## isFinite()

**isFinite** jest funkcją najwyższego poziomu, niepowiązaną z żadnym obiektem.

**Można użyć tej metody do określenia czy dana liczba jest skończona.**

Metoda isFinite sprawdza liczbę podaną jako jej argument. Jeśli argument ma wartość NaN (nie jest liczbą), jest dodatnią lub ujemną nieskończonością, metoda ta zwraca false, w przeciwnym wypadku zwraca wartość true.

```
isFinite(56)
Zwraca true
```

```
isFinite(Number.POSITIVE_INFINITY)
Zwraca false
```

| Funkcja | Wynik |
|---|---|
| `isFinite(Infinity)` | `false` |
| `isFinite(-Infinity)` | `false` |
| `isFinite(67)` | `true` |
| `isFinite(2E12)` | `true` |

## alert

Funkcji **alert()** nie ma w specyfikacji języka, ale można jej używać w środowisku przeglądarki. Służy do wyświetlania komunikatów w oknie dialogowym.

**Zadanie 1.**

Z wykorzystaniem funkcji **parseInt**() utwórz skrypt przeliczający liczby między systemem dziesiętnym, dwójkowym, ósemkowym i szesnastkowym.