

# Pole v JavaScriptu

## Vytváření polí

Chceme-li použít pole, musíme jej předtím deklarovat. Při deklaraci můžeme vytvořit pole prázdné, nebo můžeme vytvořit pole s konkrétními hodnotami. Nové pole vytvoříme pomocí konstruktoru `Array()`.

```
var pole = new Array();           vytvoření prázdného pole
var pole = new Array(10);        vytvoření pole s deseti
                                  prázdnými prvky
var jmena = new Array("Petr", "Ivan", "Radek"); vytvoření pole se třemi
                                  prvky
```

## Čtení a zápis prvků

K jednotlivým prvkům pole přistupujeme pomocí jejich indexů (pořadových čísel prvků v poli). Indexy se zapisují za název pole do hranatých závorek `[]`. Budeme-li chtít zjistit hodnotu prvního prvku pole *jmena* z předchozího příkladu a tuto hodnotu načíst do proměnné *jmeno*, použijeme následující kód:

```
jmeno = jmena[0];
```

Všimněme si jedné důležité věci: indexování prvků v poli **začíná číslem 0** nikoliv číslem 1!!!

Podobně, pokud se budeme chtít odkázat na třetí prvek pole, použijeme následující:

```
jmeno = jmena[2];
```

Zápis prvků do pole probíhá podobně. Chceme-li do pole se jmény dopsat další prvek napíšeme následující kód:

```
jmena[3] = "Filip";
```

Tento způsob má ale malou nevýhodu. Musíme si totiž pamatovat, kolik prvků pole obsahuje. Pokud použijeme již existující index, prvek pole přepíšeme, což bude ve většině případů nežádoucí. Naštěstí existuje několik možností, jak tomu předejít.

První způsob je ten, že si zjistíme počet prvků pole a poté zapíšeme na konec pole nový prvek. Velikost pole zjistíme pomocí vlastnosti pole `length`. Zápis na konec pole by potom mohl vypadat následovně:

```
velikost = jmena.length;
jmena[velikost] = "Josef";
```

Mějme na paměti, že z důvodu indexování od nuly je velikost pole o jedno větší, než poslední použitý index.

Nejrychlejší způsob, jak zapsat na konec pole je pomocí jedné z metod objektu *Array*, jejichž seznam následuje.

## Metody objektu Array

Pro příklady použijeme opět pole *jmena*, které jsme vytvořili na začátku stránky.

- o *join* – metoda slouží k převodu pole na řetězec s tím, že je oddělí zadaným znakem

```
text = jmena.join(", ");
```

proměnná text obsahuje řetězec: Petr, Ivan, Radek

- reverse – obrací pořadí prvků v poli

```
jmena.reverse();  
text = jmena.join(", ");
```

proměnná text obsahuje řetězec: Radek, Ivan, Petr

- sort – slouží k abecednímu řazení prvků v poli

```
jmena.sort();  
text = jmena.join(", ");
```

proměnná text obsahuje řetězec: Ivan, Petr, Radek

- slice – vrací část pole, Metoda má dva argumenty. První určuje počátek a druhý konec výběru.

```
pole = jmena.slice(0,2);
```

proměnná pole obsahuje dva prvky: Petr, Ivan

- pop – odstraní poslední prvek pole

```
jmena.pop();
```

- push – přidá prvek na konec pole

```
jmena.push("Karel");
```

pole jmena obsahuje čtyři prvky: Petr, Ivan, Radek, Karel

- unshift – přidá prvek nebo prvky na začátek pole

```
jmena.unshift("Karel", "Josef");
```

pole jmena obsahuje pět prvků: Karel , Josef, Petr, Ivan, Radek

- shift – odstraní první prvek pole

```
jmena.shift();
```