



Osnove programiranja

Osnove programiranja

Nizovi

Povezana literatura

JavaScript: The Definitive Guide

- Part I. Core JavaScript, 7. Arrays

https://www.w3schools.com/js/js_arrays.asp

<https://thecodebarbarian.com/for-vs-for-each-vs-for-in-vs-for-of-in-javascript.html>

<https://lodash.com/docs/4.17.15>

Niz (array)

Niz je sortirana zbirka vrijednosti.

Svaka pojedina vrijednost u nizu se naziva elementom niza. Može biti bilo kojeg JS tipa podatka, a sam niz je tipa objekt. Niz prepoznamo po uglatim zagradama ([...]).

```
var name1 = 'John';  
var name2 = 'Jane';  
var name3 = 'Mike';
```

```
var names = ['John', 'Jane', 'Mike'];  
  
console.log(names);  
console.log(typeof(names));
```

Niz

Kreiranje niza

literal

```
var emptyArray = [];  
var weirdArray = [1,,3];  
var names = ['John', 'Jane', 'Mike'];
```

new Array()

```
var emptyArray = new Array();  
var arrayLength5 = new Array(5);  
var names = new Array('John', 'Jane', 'Mike');
```

Elementi niza

Postavljanje i dohvaćanje

Za razliku od klasičnih objekata, elementi u nizu se dohvaćaju brojkom (ne-negativnim intergerom), tj. **indexom** elementa. Indexi elemenata počinju od **nule**.

Nizovi se mogu proširivati i smanjivati po volji.

```
var names = ['John', 'Jane', 'Mike'];
```

```
var name1 = names[0];
```

```
names[3] = 'Holly';
```

Metode nad nizom

| | |
|---------------|---|
| push | Dodaje element na kraj niza |
| pop | Miće zadnji element niza |
| join | Pretvara sve elemente niza u konkatenirani string |
| reverse | Obrne redosljed elemenata u nizu |
| sort | Sortira elemente niza (default: po abecedi) |
| forEach (ES5) | Iterira kroz niz, pozivajući funkciju koju odredite za svaki element |
| map (ES5) | Prosljeđuje svaki element niza funkciji i vraća niz koji sadrži vrijednosti vraćene tom funkcijom |
| ... | Pročitajte o svim metodama u Javascript: The Definite Guide |

Multidimenzionalni nizovi

Za stvaranje dvo-dimenzionalnih matrica, u JavaScriptu jednostavno koristimo niz nizova.

```
var matrix = [[1, 2, 3], ['John', 'Jane', 'Mike']];
```

```
matrix[0][1];  
matrix[2][2];  
matrix[0][4] = 5;  
matrix[1] = [];
```

Koje su vrijednosti ovih izjava?
- Koja je duljina matrix varijable
nakon zadnje izjave?

Iteracije nad nizom

for, forEach

```
var names = ['John', 'Jane',,, 'Mike'];

for (var i = 0; i < names.length; ++i) {
    console.log(names[i]);
}

names.forEach(function(value){
    console.log(value);
});
```

Zadaća: Proučite i for/of i for/in metode iteracija nad nizom, te njihove prednosti/nedostatke.

Nizovi - vježba

Pratite upute u Upute.txt.

Trajanje vježbe: 25min

Osnove programiranja

Funkcije

Povezana literatura

JavaScript: The Definitive Guide

- Part I. Core JavaScript, 8. Functions

JavaScript Patterns

- 4. Functions
- https://www.w3schools.com/js/js_functions.asp

Funkcija

Funkcija je blok JavaScripta koda koji izvršava neki zadatak.

U drugim jezicima funkcija se naziva procedura ili subrutina.

Funkcije koristimo jer njezin kôd možete **ponovo upotrijebiti**: definirajte ga jednom i koristite ga više puta. Možete koristiti isti kôd **više puta s različitim argumentima** za dobivanje različitih rezultata.

Definicija i pozivanje funkcije

```
function imeFunkcije() {  
    // nesto napravi  
}  
  
// neki jos kod između  
  
imeFunkcije();
```

Definicija i pozivanje funkcije

DEFINICIJA

```
function imeFunkcije() {  
    // nesto napravi  
}
```

POZIVANJE

```
imeFunkcije();
```

“Return” ključna riječ

Svaka funkcija vraća vrijednost undefined, osim ako upotrijebimo ključnu riječ return. Vraća izračunatu vrijednost iz funkcije.

U slučaju izjave return prije kraja funkcije, ostatak funkcije je ignoriran.

```
function imeFunkcije() {  
    var value = 1;  
    return value;  
}  
console.log(imeFunkcije());
```

```
function imeFunkcije() {  
    var value = 1;  
    return value;  
    value += 1;  
}  
console.log(imeFunkcije());
```

Što se ispisuje u konzoli u ova dva slučaja?

Parametri i argumenti

Definicija funkcije može (ali ne mora) sadržavati popis identifikatora, poznatih kao parametri, koji djeluju kao lokalne varijable za tijelo funkcije.

PARAMETRI

```
function udaljenostKoordinata(x1, y1, x2, y2) {  
    var dx = x2 - x1;  
    var dy = y2 - y1;  
    console.log(Math.sqrt(dx*dx + dy*dy));  
}
```

ARGUMENTI

```
udaljenostKoordinata(1,1,3,4);
```

Parametri

JavaScript ne zahtijeva tip podataka u parametrima koje dobiva u funkciji, kao ni točan broj parametara. U funkciji možemo računati na te opcionalne parametre.

```
function myFunc(a, b) {  
    if (!b) {  
        b = 1;  
    }  
    return a + b;  
}  
  
myFunc(1);
```

```
function myFunc(a, b) {  
    b = b || 1;  
    return a + b;  
}  
  
myFunc(1,1);
```

```
function myFunc(a, b = 1) {  
    return a + b;  
}  
  
myFunc(1,1);
```

samo ES6

Funkcija kao vrijednost

Funkcija se u JavaScriptu može koristiti kao bilo koja druga varijabla.

```
var addOne = function(value) {  
    return value + 1;  
};
```

```
var myVar = addOne(1);
```

Koja je vrijednost myVar
varijable u ova dva slučaja?

Funkcije - vježba

1. Riješite zadatke sa slideova sa ispisom u konzoli. Trajanje vježbi: 10min
2. Pratite upute u Upute.txt. Trajanje vježbe: 15min