

Nizovi i objekti u JavaScript-u

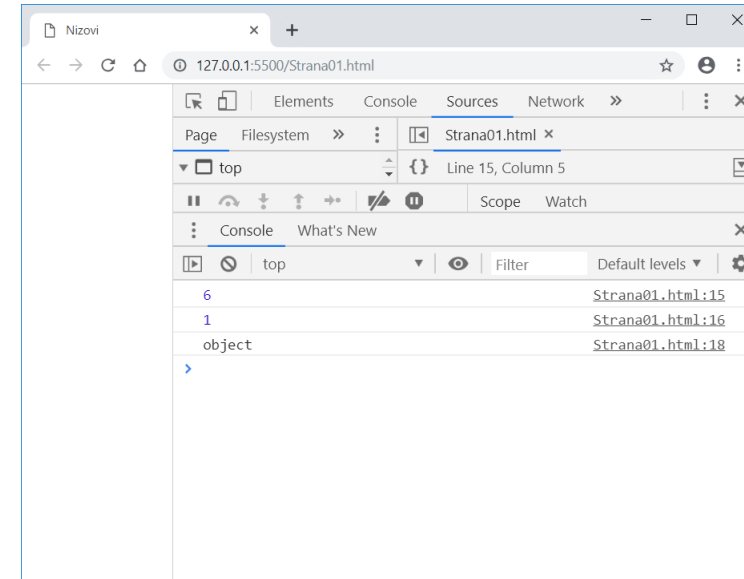
Nizovi

- JavaScript nizovi se koriste za čuvanje više vrednosti u jednoj promenljivoj
- Članovima niza pristupa se na osnovu indeksa koji je baziran na 0
- Nizovi su specijalni tipovi objekata

```
<script>
  const x = [1, 3, 5, 7, 4, 8];
  const brojClanova = x.length;
  const x0 = x[0];

  console.log(brojClanova);
  console.log(x0);

  console.log(typeof x);
</script>
```



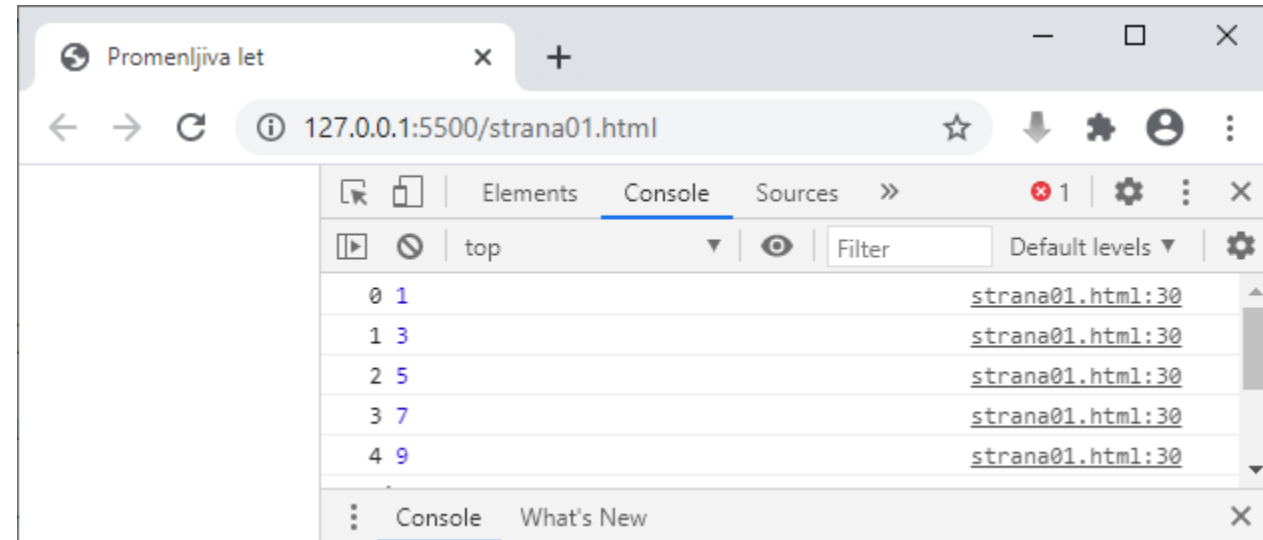
Kreiranje inicijalizovanog niza i prolazak for petljom

```
<script>
  const x = [1, 3, 5, 7, 9];
  for (let i = 0; i < x.length; i++) {
    console.log(x[i]);
  }
</script>
```

Prolazak kroz niz for...in petljom

- forin prolazi kroz ključeve "0", "1", "2"
- i se ponaša kao da je const unutar jedne iteracije
- koristi se se za objekte

```
<script>
  const x = [1, 3, 5, 7, 9];
  //fin
  for (const i in x) {
    console.log(i, x[i]);
  }
</script>
```



fin code snippet
Javascript (ES6) code snippets

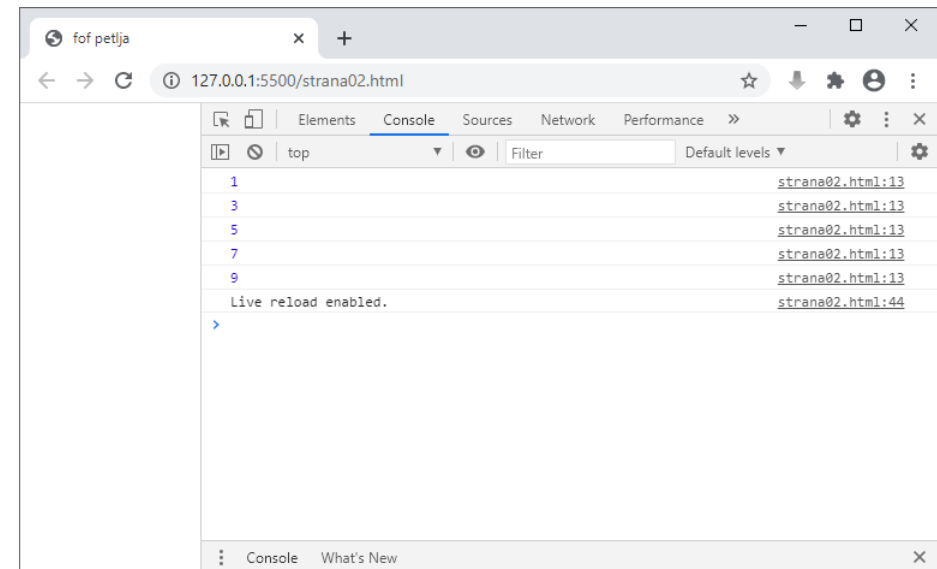
Prolazak kroz niz for...of petljom

- forof prolazi kroz vrednosti
- koristi se za nizove i stringove
- for...of sigurniji i preporučeni način za prolazak kroz elemente niza

```
<script>
  const x = [1, 3, 5, 7, 9];

  for (const i of x) {
    console.log(i);
  }
</script>
```

fof code snippet
Javascript (ES6) code snippets



Upotreba let i const za deklarisanje niza

const znači da se promenljiva ne menja ali se njen sadržaj može menjati

```
const b = [];  
b = [1, 2, 3]; //nije dozvoljeno  
b[1] = 5; //dozvoljeno
```

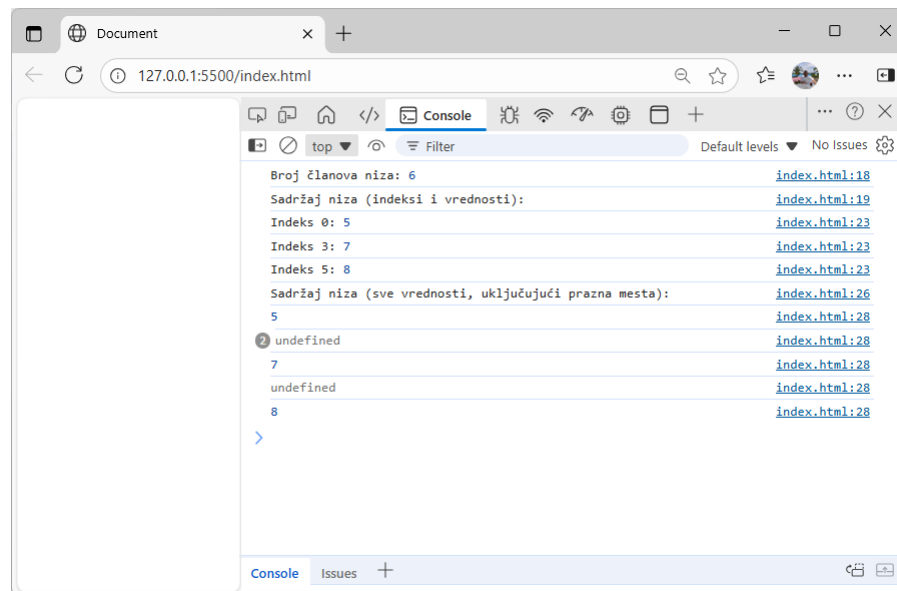
```
let a = [];  
a = [5, 7, 8]; //dozvoljeno
```

Kreiranje neinicijalizovanog niza

```
<script>
  // Kreiranje praznog niza
  //const a = new Array();
  const a = [];

  // Dodela vrednosti na pojedine indekse
  a[0] = 5;
  a[3] = 7;
  a[5] = 8;

  console.log("Broj članova niza:", a.length);
  console.log("Sadržaj niza (indeksi i vrednosti:");
  // Ispis postojećih elemenata (indeksi sa vrednostima)
  for (const i in a) {
    console.log(`Indeks ${i}:`, a[i]);
  }
  console.log("Sadržaj niza (sve vrednosti, uključujući prazna mesta:");
  for (const vrednost of a) {
    console.log(vrednost);
  }
</script>
```



- **for...in** prolazi kroz **postojeće indekse** (0, 3, 5)
- **for...of** prolazi kroz sve pozicije u nizu i za prazna (nepostojeća) mesta ispisuje **undefined**

Metoda push

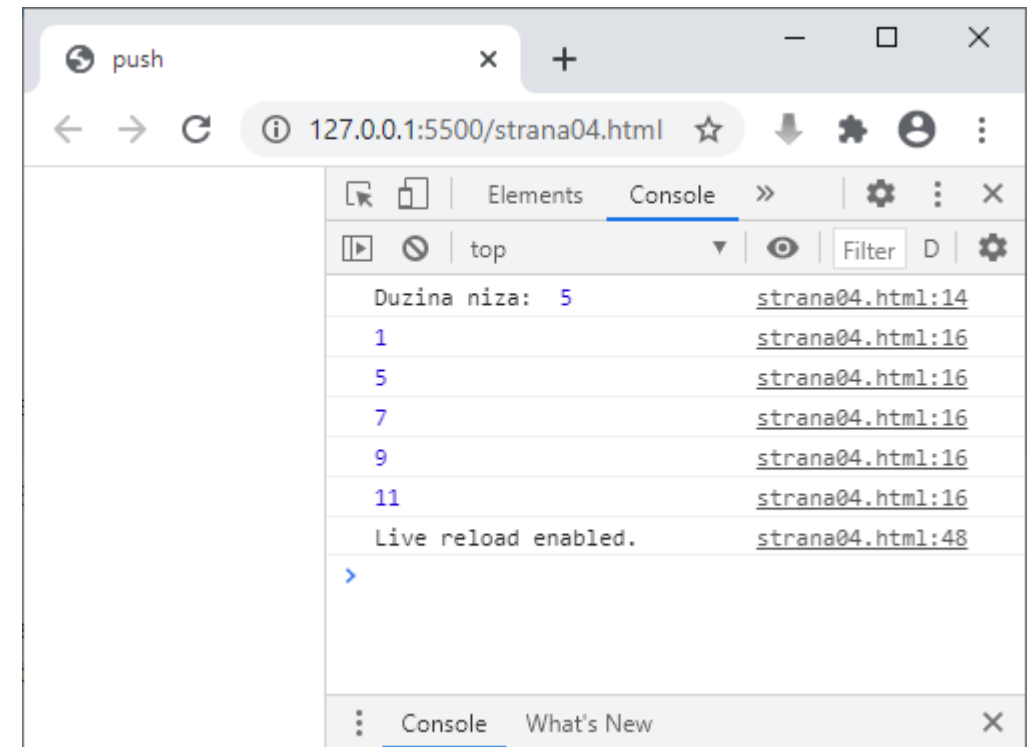
Metoda push() dodaje element na kraj niza i vraća broj članova novog niza.

```
<script>
  const a = [1, 5, 7, 9];

  // push vraća novu dužinu niza
  const duzina = a.push(11);

  console.log("Dužina niza:", duzina);

  for (const broj of a) {
    console.log(broj);
  }
</script>
```



Metoda push() sa više argumenata

```
<script>
  const brojevi = [1, 3, 5];

  // push može primiti više argumenata odjednom
  const duzina = brojevi.push(7, 9, 11);

  console.log("Dužina niza:", duzina);
  console.log("Novi niz:", brojevi);

  for (const broj of brojevi) {
    console.log(broj);
  }
</script>
```

Metoda pop()

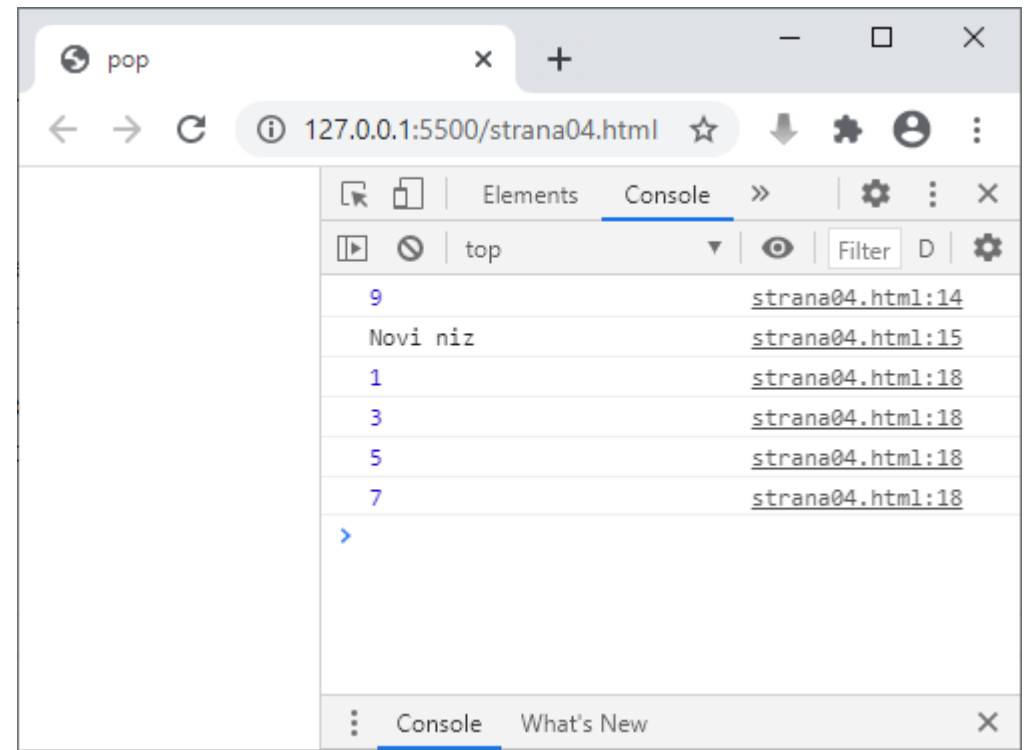
Uklanja element sa kraja niza i vraća taj element.

```
<script>
  // Kreiranje niza
  const x = [1, 3, 5, 7, 9];

  // Uklanja poslednji element i vraća ga
  const poslednji = x.pop();

  console.log("Uklonjeni poslednji element:", poslednji);
  console.log("Novi niz:");

  // Ispis svih preostalih elemenata niza
  for (const broj of x) {
    console.log(broj);
  }
</script>
```



Metoda shift()

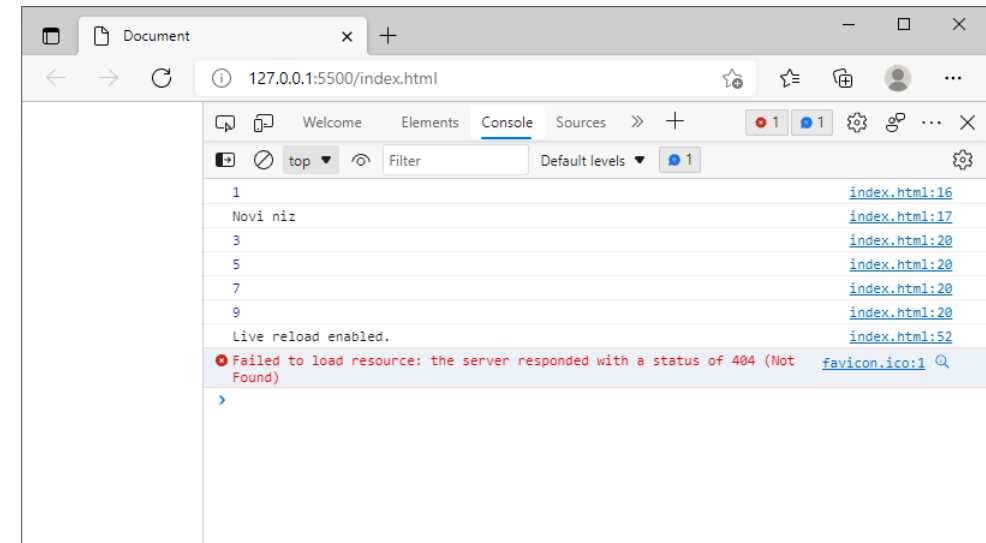
Uklanja element sa početka niza i vraća taj element.

```
<script>
  // Kreiranje niza
  const x = [1, 3, 5, 7, 9];

  // Uklanja prvi element niza i vraća ga
  const prvi = x.shift();

  console.log("Uklonjeni prvi element:", prvi);
  console.log("Novi niz:");

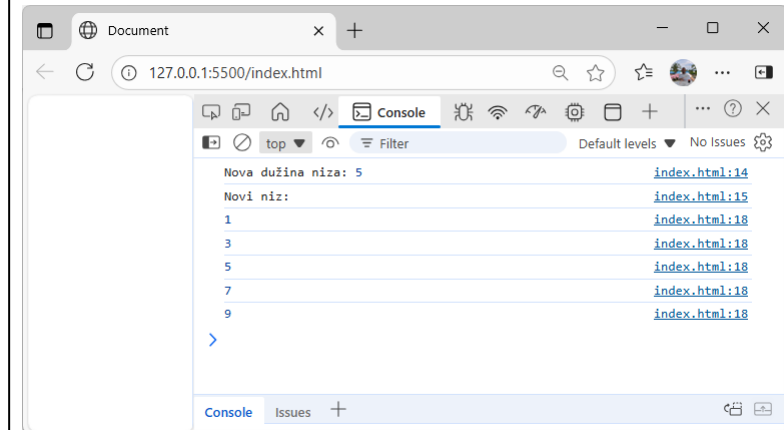
  // Ispis svih elemenata preostalog niza
  for (const broj of x) {
    console.log(broj);
  }
</script>
```



Metoda unshift

```
<script>
  // Kreiranje niza
  const a = [3, 5, 7, 9];
  // Dodaje novi element na početak niza i vraća novu dužinu
  const duzina = a.unshift(1);

  console.log("Nova dužina niza:", duzina);
  console.log("Novi niz:");
  // Ispis svih elemenata niza
  for (const broj of a) {
    console.log(broj);
  }
</script>
```



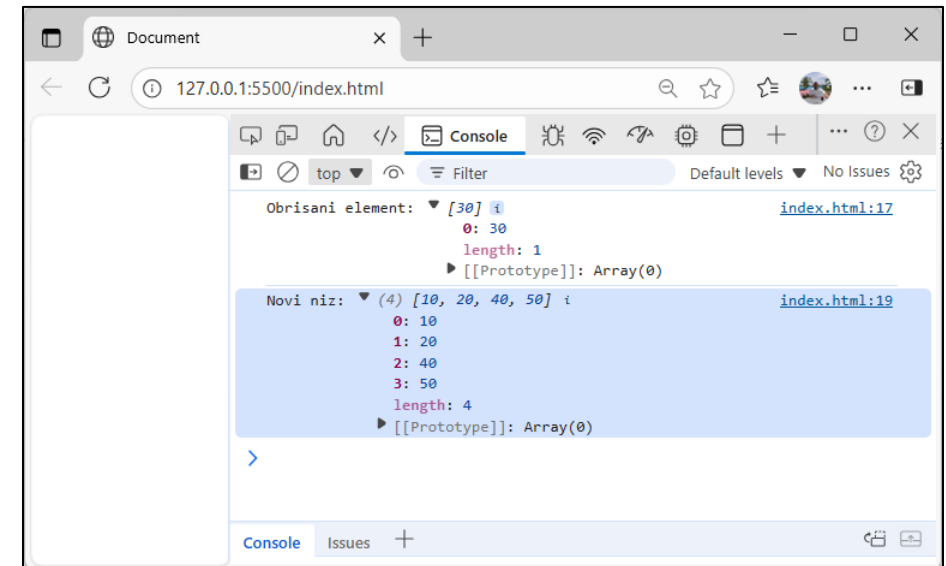
Metoda splice() – uklanjanje elementa sa pozicije

```
array.splice(pocetniIndeks, brojElemenataZaBrisanje, [noviElement1, noviElement2, ...])
```

```
<script>
  const brojevi = [10, 20, 30, 40, 50];

  // Uklanja jedan element sa pozicije (indeksa) 2
  // splice vraca niz uklonjenih elementa niza
  const obrisani = brojevi.splice(2, 1);

  console.log("Obrisani element:", obrisani);
  console.log("Novi niz:", brojevi);
</script>
```

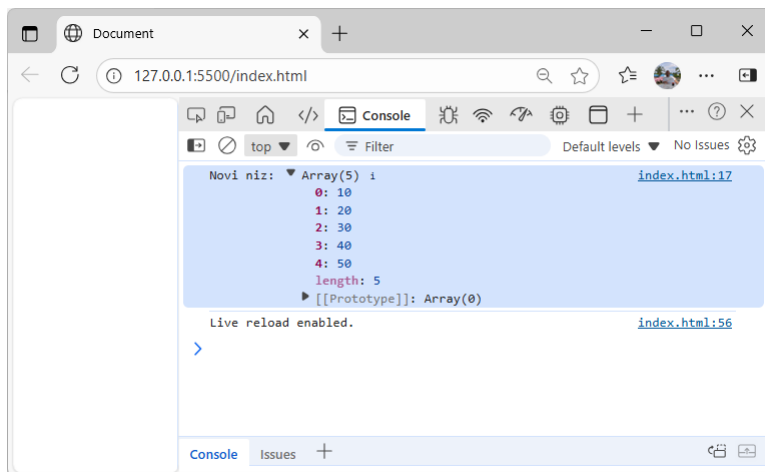


Metoda splice() – dodavanje elementa na poziciju

```
<script>
  const brojevi = [10, 20, 40, 50];

  // Dodajemo broj 30 na poziciju (indeks) 2
  // Prvi parametar: indeks gde ubacujemo
  // Drugi parametar: koliko elemenata brišemo (0 znači nijedan)
  // Treći parametar: novi element koji dodajemo
  brojevi.splice(2, 0, 30);

  console.log("Novi niz:", brojevi);
</script>
```



Rad sa delovima niza

```
const a = [1, 2, 3, 4];  
const b = [5, 6, 7];
```

Metoda	Objašnjenje	Primer	Rezultat
slice()	vraća <i>deo niza</i> od start do end - 1 (ne menja original)	<code>a.slice(1, 3)</code>	[2, 3]
concat()	spaja više nizova u jedan (ne menja originalne)	<code>a.concat(b)</code>	[1, 2, 3, 4, 5, 6, 7]
join()	spaja sve elemente niza u string , razdvojene zadatim znakom	<code>a.join('-')</code>	"1-2-3-4"

Iteracija i transformacija

```
const a = [1, 2, 3, 4, 5];
```

Metoda	Šta radi	Primer	Rezultat
forEach()	prolazi kroz svaki element (ne vraća novi niz)	a.forEach(x => console.log(x))	ispisuje: 1 2 3 4 5
map()	pravi novi niz sa rezultatima funkcije	a.map(x => x * 2)	[2, 4, 6, 8, 10]
filter()	pravi novi niz sa elementima koji zadovoljavaju uslov	a.filter(x => x > 3)	[4, 5]
reduce()	sabija niz u jednu vrednost (npr. zbir)	a.reduce((s, x) => s + x, 0)	15
some()	proverava da li barem jedan element ispunjava uslov	a.some(x => x > 4)	true
every()	proverava da li svi elementi ispunjavaju uslov	a.every(x => x > 0)	true

Pronalaženje i pretraga

```
const a = [10, 20, 30, 40, 50];  
const b = ["Ana", "Marko", "Jelena", "Ivana"];
```

Metoda	Šta radi	Primer	Rezultat
indexOf()	vraća indeks prve pojave zadate vrednosti	a.indexOf(30)	2
lastIndexOf()	vraća indeks poslednje pojave zadate vrednosti	[10,20,10,40].lastIndexOf(10)	2
includes()	proverava da li niz sadrži određenu vrednost	a.includes(20)	true
find()	vraća prvi element koji zadovoljava uslov	a.find(x => x > 25)	30
findIndex()	vraća indeks prvog elementa koji zadovoljava uslov	a.findIndex(x => x > 25)	2
some()	vraća true ako barem jedan element ispunjava uslov	a.some(x => x > 45)	true
every()	vraća true ako svi elementi ispunjavaju uslov	a.every(x => x > 5)	true

JavaScript objekti

- Strukture podataka koje čuvaju parove **ključ–vrednost** (key–value pairs)
- Svaki ključ (key) je string, a vrednost (value) može biti bilo koji tip (broj, string, niz, funkcija, drugi objekat)
- Svaki par se **razdvaja dvotačkom (:)**
- Više parova se u objektu razdvaja **zarezom (,)**
- Objekti su dinamični — svojstva se mogu dodavati ili brisati u toku izvršavanja programa
- Mogu sadržati **metode** (funkcije koje pripadaju objektu)

Pristup svojstvima objekta

```
<script>
  const osoba = { ime: "Pera", prezime: "Perić", starost: 24 };

  //tačkasta notacija – kad znamo ime svojstva unapred
  console.log(osoba.ime); // ispisuje: Pera

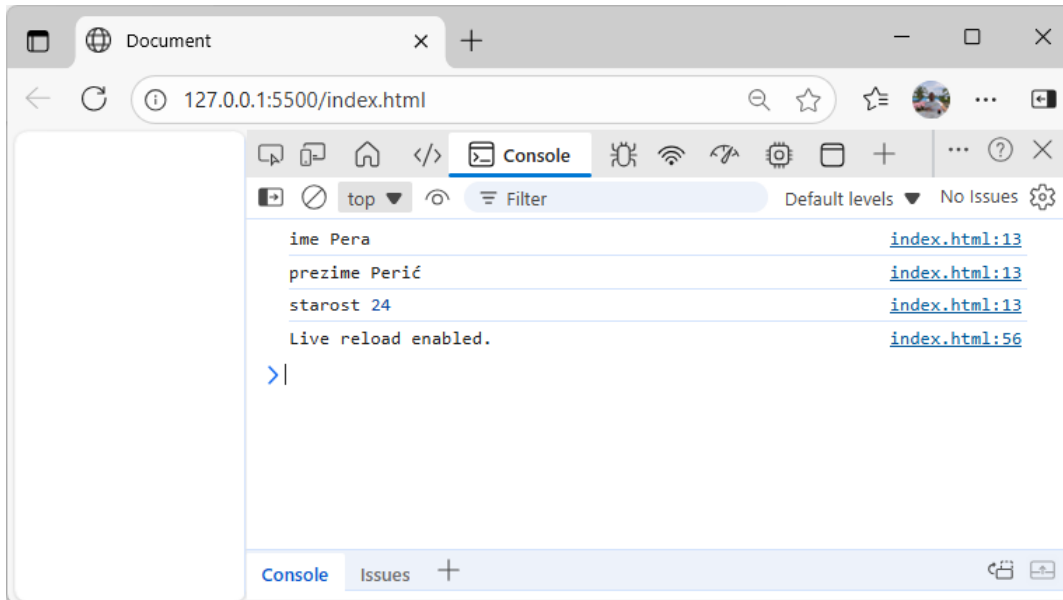
  // zagrade sa stringom
  console.log(osoba["ime"]); // ispisuje: Pera

  // zagrade sa promenljivom – kad ključ dobijamo dinamički
  const kljuc = "prezime";
  console.log(osoba[kljuc]); // ispisuje: Perić
</script>
```

JavaScript objekti

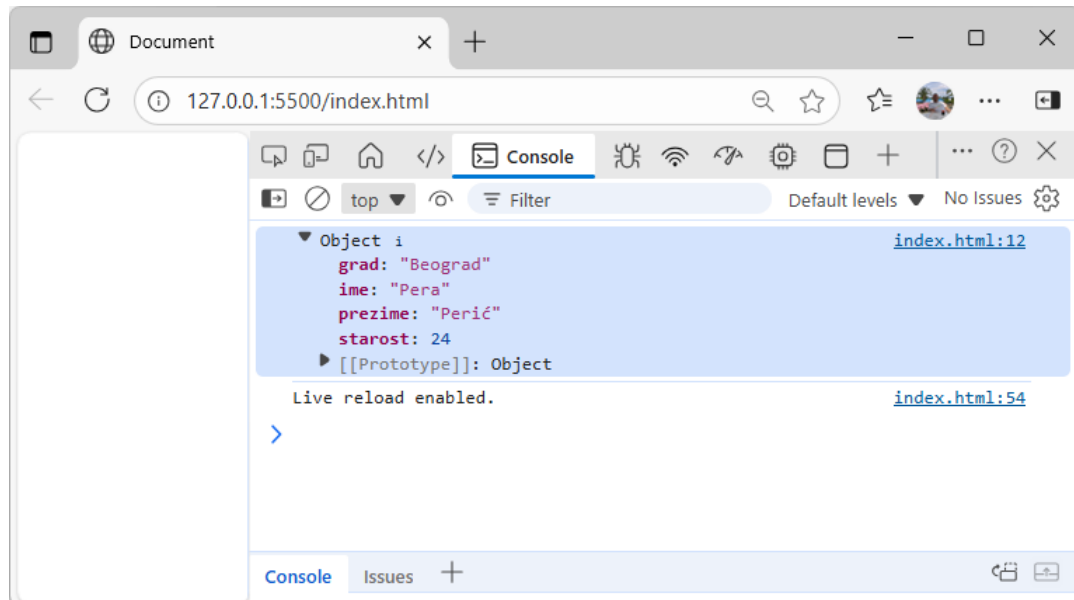
```
<script>
  const osoba = { ime: "Pera", prezime: "Perić", starost: 24 };

  for (const kljuc in osoba) {
    console.log(kljuc, osoba[kljuc]);
  }
</script>
```



Dodavanje svojstava objekta

```
<script>
  const osoba = { ime: "Pera", prezime: "Perić" };
  osoba.starost = 24;
  osoba["grad"] = "Beograd";
  console.log(osoba);
</script>
```

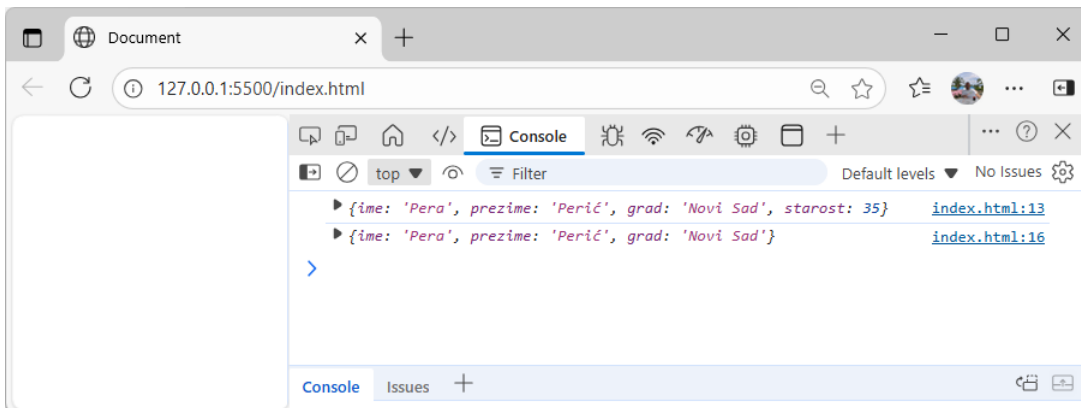


Menjenje i brisanje svojstava objekta

```
<script>
  const osoba = { ime: "Pera", prezime: "Perić", grad: "Beograd", starost:35 };

  //menjenje i brisanje svojstava objekta
  osoba.grad = "Novi Sad";
  console.log(osoba);

  delete osoba.starost;
  console.log(osoba);
</script>
```



Definisanje metode objekta

```
<script>
  const osoba = {
    ime: "Pera",
    prezime: "Perić",
    stampaj: function() {
      console.log("Zdravo, ja sam " + this.ime + " " + this.prezime);
    }
  };

  osoba.stampaj(); // Poziv metode
</script>
```

Skraćena verzija definisanje metode

```
<script>
  const osoba = {
    ime: "Pera",
    prezime: "Perić",
    stampaj() {
      console.log(`Zdravo, ja sam ${this.ime} ${this.prezime}`);
    }
  };

  osoba.stampaj();
</script>
```

Ugnježdeni objekti (objekti unutar objekata)

```
<script>
  const osoba = {
    ime: "Pera",
    prezime: "Perić",
    adresa: {
      grad: "Beograd",
      ulica: "Knez Mihailova",
      broj: 10
    },
    stampaj() {
      console.log(`${this.ime} ${this.prezime} živi u gradu ${this.adresa.grad}`);
    }
  };

  console.log(osoba.adresa.ulica); // pristup ugnježđenom objektu
  osoba.stampaj();                 // poziv metode
</script>
```

Nizovi objekata

```
<script>
  const osobe = [
    { ime: "Pera", prezime: "Perić", starost: 24 },
    { ime: "Mika", prezime: "Mikić", starost: 30 },
    { ime: "Ana", prezime: "Anić", starost: 22 }
  ];

  // Ispis svih osoba
  for (const osoba of osobe) {
    console.log(osoba.ime, osoba.prezime, osoba.starost);
  }
</script>
```

Kreiranje objekata pomoću funkcije konstruktora

```
<script>
  function Osoba(ime, prezime, starost) {
    this.ime = ime;
    this.prezime = prezime;
    this.starost = starost;

    this.stampaj = function() {
      console.log(`Zdravo, ja sam ${this.ime} ${this.prezime}`);
    };
  }

  const p1 = new Osoba("Pera", "Perić", 24);
  const p2 = new Osoba("Ana", "Anić", 22);

  p1.stampaj();
  p2.stampaj();
</script>
```

Prototip objekta

- Prototip je zajednički objekat koji dele sve instance jednog konstruktora
- U njemu se mogu definisati metode i svojstva koja su zajednička svim objektima kreiranim pomoću tog konstruktora
- Za pristup prototipu koristi se svojstvo prototype konstruktora

Naknadno dodavanje funkcije u prototip

```
<script>
  // Konstruktorska funkcija
  function Osoba(ime, prezime, starost) {
    this.ime = ime;
    this.prezime = prezime;
    this.starost = starost;
  }

  // Dodavanje metode naknadno u prototip
  Osoba.prototype.stampaj = function() {
    console.log(`Zdravo, ja sam ${this.ime} ${this.prezime}`);
  };

  const p1 = new Osoba("Pera", "Perić", 24);
  const p2 = new Osoba("Ana", "Anić", 22);

  p1.stampaj();
  p2.stampaj();
</script>
```

ES6

- ES6 (skraćeno od ECMAScript 6) je verzija JavaScript jezika koja je zvanično objavljena 2015. godine i poznata i kao ES2015
- To je jedno od najvažnijih ažuriranja u istoriji JavaScripta, jer je donelo mnoštvo novih funkcionalnosti koje su modernizovale i pojednostavile pisanje koda
- ECMAScript (ES) je standard koji definiše kako JavaScript treba da funkcioniše

Kreiranje objekata pomoću klasa (ES6 Class)

- Klase su moderniji i čitljiviji način za definisanje objekata u JavaScriptu
- Uvedene su u ES6 (2015) standardu
- Predstavljaju **sintaksni šećer** nad postojećim prototipskim nasleđivanjem, tj. funkcionalno rade isto kao konstruktorske funkcije i prototype
- **constructor()** je posebna metoda koja se automatski poziva pri kreiranju novog objekta
- Sve metode definisane u klasi automatski se nalaze u prototype delu objekta

```
<script>
  class Osoba {
    constructor(ime, prezime, starost) {
      this.ime = ime;
      this.prezime = prezime;
      this.starost = starost;
    }

    stampaj() {
      console.log(`Zdravo, ja sam ${this.ime} ${this.prezime}`);
    }
  }

  const p1 = new Osoba("Pera", "Perić", 24);
  const p2 = new Osoba("Ana", "Anić", 22);

  p1.stampaj();
  p2.stampaj();
</script>
```

JSON(JavaScript Object Notation)

- JSON je tekstualni format za razmenu podataka
- Zasniva se na JavaScript objektima, ali je jezički nezavisan – koristi se u svim modernim programskim jezicima.
- JSON format je lagan, čitljiv i pogodan za prenos podataka preko mreže

JSON sintaksa

- Podaci su zapisani kao parovi ključ–vrednost
- Ključevi i string vrednosti moraju biti u dvostrukim navodnicima
- JSON podržava tipove podataka: **string**, **number**, **boolean**, **null**, **object**, **array**
- Nisu dozvoljene funkcije ni komentari
- Elementi se odvajaju zarezom (,), dok poslednji element ne sme imati zarez
- Objekti su unutar viticastih zagrada { }
- Nizovi su unutar uglastih zagrada []

Primer JSON dokumenta

```
{  
  "ime": "Pera",  
  "prezime": "Perić",  
  "godine": 24,  
  "student": true,  
  "adresa": {  
    "grad": "Beograd",  
    "ulica": "Knez Mihailova 10"  
  },  
  "jezici": ["srpski", "engleski", "nemački"]  
}
```

Metoda JSON.stringify()

- Pretvara JavaScript objekat ili niz objekata u JSON tekstualni format
- Koristi se pri slanju podataka ka web serveru ili čuvanju podataka u fajl

```
<script>
  function Osoba(ime, prezime, starost) {
    this.ime = ime;
    this.prezime = prezime;
    this.starost = starost;
  }

  const osoba = new Osoba("Marko", "Marković", 25);

  const jsonTekst = JSON.stringify(osoba);
  console.log(jsonTekst);
</script>
```

```
{"ime":"Marko","prezime":"Marković","starost":25}
```

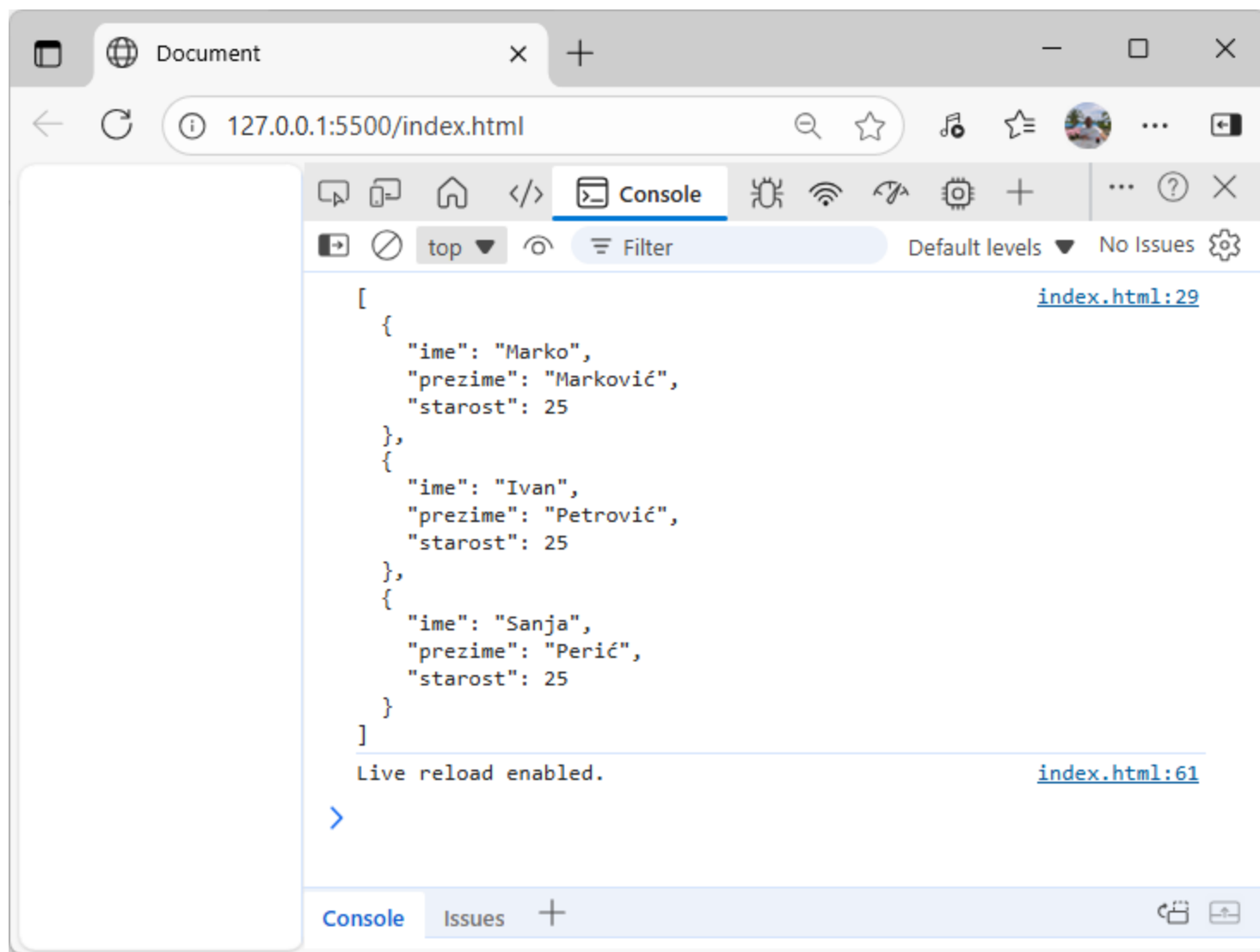
Pretvaranje javascript niza u JSON tekst

```
<script>
  // Konstruktor objekta Osoba
  function Osoba(ime, prezime, starost) {
    this.ime = ime;
    this.prezime = prezime;
    this.starost = starost;
  }

  // Kreiranje više instanci objekta
  const os1 = new Osoba("Marko", "Marković", 25);
  const os2 = new Osoba("Ivan", "Petrović", 25);
  const os3 = new Osoba("Sanja", "Perić", 25);

  // Formiranje niza objekata
  const nizOsoba = [os1, os2, os3];

  // Pretvaranje niza objekata u JSON string
  const jsonText = JSON.stringify(nizOsoba, null, 2);
  console.log(jsonText);
</script>
```



Metoda JSON.parse()

- Pretvara JSON tekst u JavaScript objekat ili niz objekata
- Koristi se prilikom prijema podataka sa web servera, jer podaci najčešće stižu u JSON formatu
- `JSON.parse(jsonString);`

Metoda JSON.parse()

```
<script>
  // JSON podatak koji stiže sa servera (jedan objekat)
  const jsonText = `
    {
      "ime": "Marko",
      "prezime": "Marković",
      "starost": 25
    }`;

  // Pretvaranje JSON stringa u JavaScript objekat
  const osoba = JSON.parse(jsonText);

  // Prikaz podataka
  console.log(osoba.ime);           // Marko
  console.log(osoba.prezime);       // Marković
  console.log(osoba.starost);       // 25
</script>
```

Metoda JSON.parse()

```
<script>
  const jsonText = `
  [
    {"ime": "Marko", "prezime": "Marković", "starost": 25},
    {"ime": "Ivan", "prezime": "Petrović", "starost": 24},
    {"ime": "Sanja", "prezime": "Perić", "starost": 23}
  ]`;

  // Pretvaranje JSON stringa u JavaScript niz objekata
  const nizOsoba = JSON.parse(jsonText);

  // Ispis svake osobe
  for (const osoba of nizOsoba) {
    console.log(osoba.ime, osoba.prezime, osoba.starost);
  }
</script>
```

Pitanje 1

Deklarisanje i inicijalizacija javascript niza je ispravno izvršena u primeru pod opcijom:

- a. `const a = [1, 2, 3, 4];`
- b. `const b = {1,2,3,4};`
- c. `const c= (1,2,3,4);`

Odgovor: a

Pitanje 2

Šta se ispisuje na konzoli nakon izvršavanja sledećeg javascript koda:

```
<script>
  const osoba = { ime: "Pera", prezime: "Peric", starost: 30 };
  let s = "";
  for (const i in osoba) {
    s += i + " ";
  }
  console.log(s);
</script>
```

- a. Pera Peric 30
- b. ime prezime starost
- c. ime:Pera prezime:Peric starost:30

Odgovor: b

Pitanje 3

Koja od sledećih sintaksi ispravno kreira JavaScript objekat?

- a. `const osoba = (ime: "Marko", starost: 30);`
- b. `const osoba = [ ime: "Marko", starost : 30 ];`
- c. `const osoba = { ime: "Marko", starost : 30 };`

Odgovor: c

Pitanje 4

Koje svojstvo JavaScript niza vraća broj elemenata u nizu?

- a. length
- b. size
- c. count

Odgovor: a

Pitanje 5

Koja petlja u JavaScript-u prolazi kroz niz i vraća indekse njegovih elemenata?

- a. for...of
- b. for...in
- c. wile

Odgovor: b

Pitanje 6

Koja petlja u JavaScript-u prolazi kroz niz i vraća vrednosti njegovih elemenata?

- a. for...of
- b. for...in
- c. wile

Odgovor: a