

Uvod u JavaScript

JavaScript varijante

- JavaScript je prvenstveno korišćen za klijentsko programiranje
 - Client-side JavaScript
 - Izvršava se u browseru klijenta
- Postoji i JavaScript kod koji se izvršava na serveru
 - Server-side JavaScript , izvršava se na serveru i kreira dinamički sadržaj koji se šalje do browsera
 - **Node.js** serverski Java script, nastao 2009. godine
 - **Node.js** je serverska platforma bazirana na script mašini Google chrome browsera
 - Godine 2010 napravljen je menadžer paketa za Node.js pod nazivom **npm**

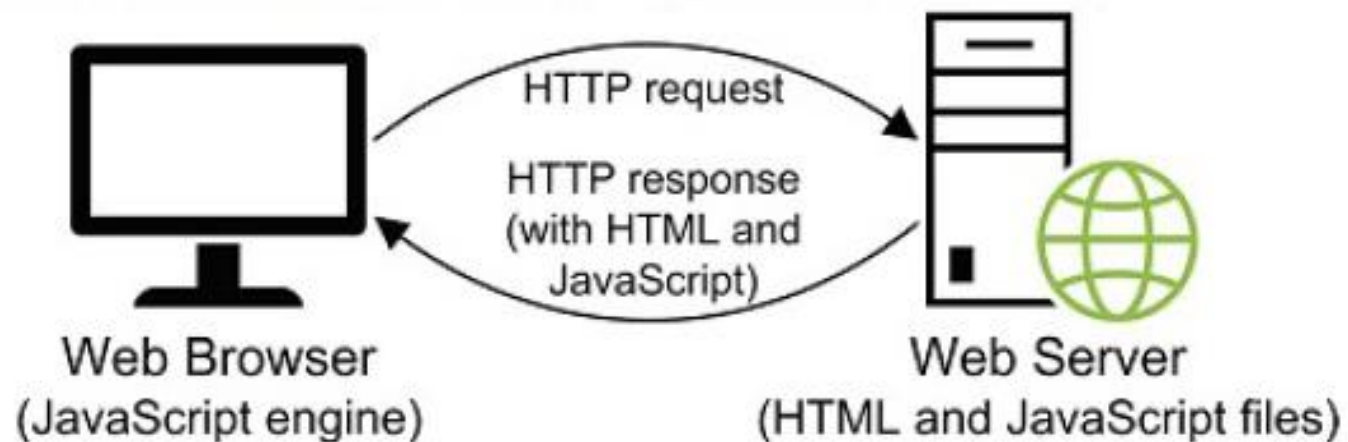
TypeScript

- TypeScript je programski jezik otvorenog koda razvijen od strane Microsofta, 2012. godine
- Kompajlira se u čist JavaScript
- Tekuća verzija TypeScript 5.6
- TypeScript je programski jezik koji predstavlja proširenje javascripta, dodaje tipove podataka, klase i module u JavaScript
- Koristi se za kreiranje javascript koda koji se može izvršavati i na klijentu i na serveru

Klijentski JavaScript

- Koristi se za izradu klijentskih delova web aplikacija
- Baziran na C sintaksi
- Java script kod se interpretira (ne kompajlira se)
- JavaScript je programski jezik koji podržava:
 - Promenljive za čuvanje informacija
 - Operatore za izvođenje operacija i poređenje
 - Funkcije za grupisanje koda u celine koje se mogu više puta pozivati
 - Uslovne izraze i programske petlje za kontrolu programskog toka
- Mogućnost kreiranja objekata sa svojstvima, metodama i događajima

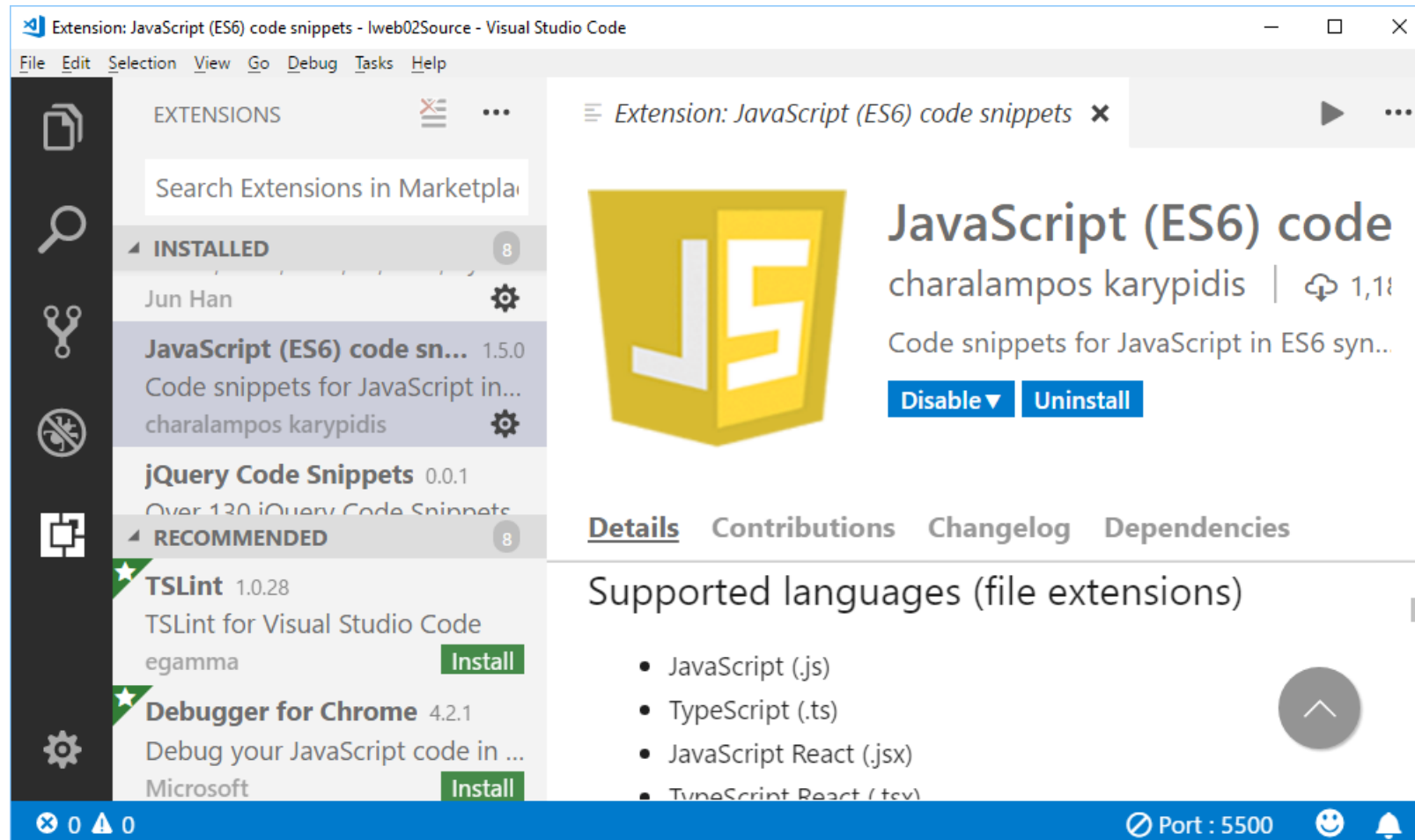
Izvršavanje klijentskog javascript koda



Funkcionisanje JavaScripta

- JavaScript se koristi u kombinaciji sa objektnim modelom dokumenta (DOM) da bi se web strana učinila dinamičkom
- Kada se HTML document učitava u browser kreira se njegov objektni model, odnosno objekat pod nazivom **document**
- Objekat document je deo objekta **window** koji predstavlja prozor browsera
- JavaScript može modifikovati web stranu kada se ona učitava u browser ili kao odgovor na akciju korisnika

Ekstenzija za JavaScript



The screenshot displays the Visual Studio Code interface with the 'JavaScript (ES6) code snippets' extension page open. The left sidebar shows the 'EXTENSIONS' view with a search bar and a list of installed and recommended extensions. The main panel shows the details for the 'JavaScript (ES6) code snippets' extension by charalampos karypidis, including its version (1.5.0), a 'Disable' button, and an 'Uninstall' button. Below the extension details, there is a section for 'Supported languages (file extensions)' which lists JavaScript (.js), TypeScript (.ts), JavaScript React (.jsx), and TypeScript React (.tsx).

Extension: JavaScript (ES6) code snippets - lweb02Source - Visual Studio Code

File Edit Selection View Go Debug Tasks Help

EXTENSIONS

Search Extensions in Marketplace

INSTALLED 8

Jun Han

JavaScript (ES6) code sn... 1.5.0
Code snippets for JavaScript in...
charalampos karypidis


jQuery Code Snippets 0.0.1
Over 130 jQuery Code Snippets

RECOMMENDED 8

TSLint 1.0.28
TSLint for Visual Studio Code
egamma **Install**

Debugger for Chrome 4.2.1
Debug your JavaScript code in ...
Microsoft **Install**

Extension: JavaScript (ES6) code snippets x

 **JavaScript (ES6) code**
charalampos karypidis | 1,18
Code snippets for JavaScript in ES6 syn..
Disable **Uninstall**

Details Contributions Changelog Dependencies

Supported languages (file extensions)

- JavaScript (.js)
- TypeScript (.ts)
- JavaScript React (.jsx)
- TypeScript React (.tsx)

Port : 5500

JavaScript sintaksa

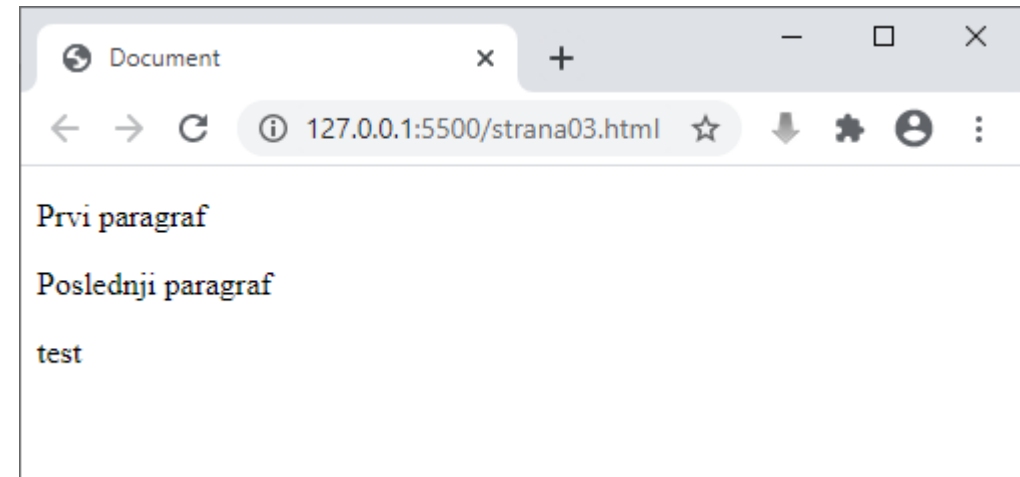
- JavaScript naredba predstavlja liniju JavaScript koda koji treba da se izvrši
- Naredba u JavaScriptu treba biti napisana u jednoj liniji i treba da se završava operatorom ;
- JavaScript naredbe se grupišu unutar bloka { }
- JavaScript je osetljiv na velika i mala slova

Pisanje javascript-a

- Ubacivanje javascript koda u html dokument
 - u telu html dokumenta
 - u head delu html dokumenta
- Pisanje javascript koda u zasebnom .js dokumentu i njegovo referenciranje upotrebom src atributa
- Pisanje javascript koda u zasebnom .js dokumentu na mreži i njegovo referenciranje upotrebom src atributa
- Preferira se pisanje javascript koda neposredno pre završnog body taga radi bržeg učitavanja strane

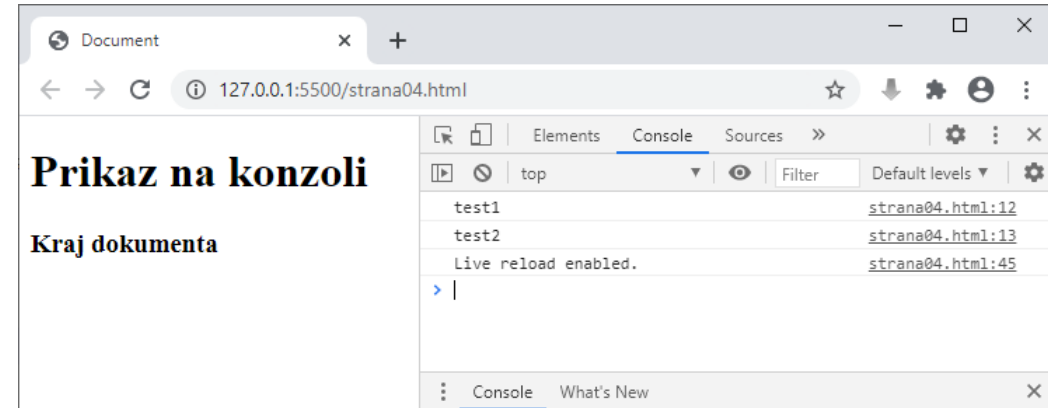
Javascript kod u body delu strane

```
<body>
  <p>Prvi paragraf</p>
  <p>Poslednji paragraf</p>
  <script>
    document.write("test");
  </script>
</body>
```



Prikazivanje rezultata na konzoli

```
<body>
  <h1>Prikaz na konzoli</h1>
  <script>
    //clg
    console.log("test1");
    console.log("test2");
  </script>
  <h3>Kraj dokumenta</h3>
</body>
```



Referenciranje eksternog js fajla

```
<body>  
<script src="Kod01.js"></script>  
<p>  
Kraj dokumenta  
</p>  
</body>
```

```
//Kod01.js  
document.write("test");
```

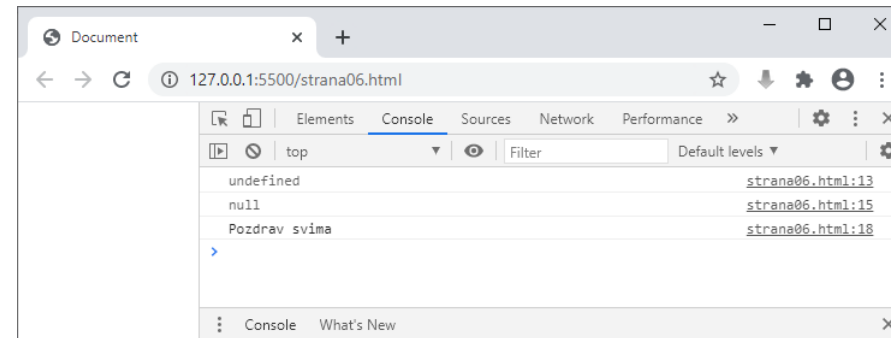
Komentari

```
<script>  
  // prikaz poruke korisniku  
  document.write("Dobar dan svima");  
  /* Mozete koristit multi-line komentar  
  da biste dodali vise informacija */  
  alert("Dobar dan svima, takodje!");  
</script>
```

Promenljive

- Promenljive u JavaScriptu počinju slovom ili donjom crtom
- Koristi se ključna reč let ili var za deklarisanje promenljivih
- Imena promenljivih su case - sensitive
- Promenljiva može biti nedefinisana a takođe može imati i null vrednost

```
<script>  
//deklarisanje promenljive  
var x;  
// promenljiva x nema vrednost  
console.log(x);  
  
var y = null; // null promenljiva  
console.log(y);  
  
// deklarisanje i inicijalizacija  
promenljive  
var poruka = "Pozdrav svima";  
console.log(poruka);  
</script>
```



Tipovi podataka

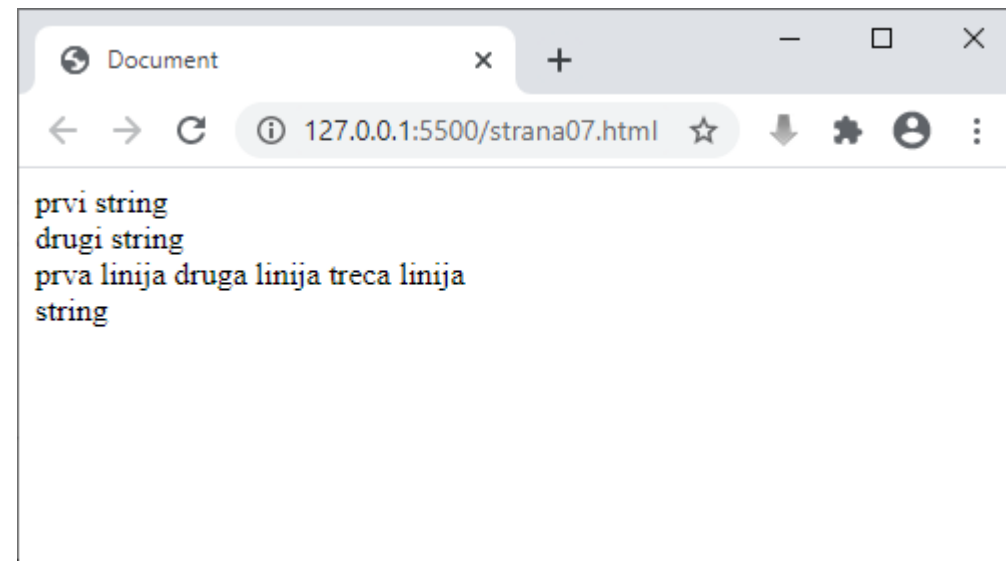
- Za razliku od C# ne može se specificirati tip promenljive u javascriptu
- Promenljiva se deklariše korišćenjem ključne reči let ili var, a zatim JavaScript sam pokušava da odredi njen tip
- JavaScript prepoznaje tri prosta tipa:
 - string
 - number
 - boolean

JavaScript stringovi

```
<script>
var s1 = "prvi string";
var s2 = 'drugi string';
var s3 = `prva linija
druga linija
treca linija
`;

document.write(s1 + "<br>");
document.write(s2 + "<br>");
document.write(s3 + "<br>");

document.write(typeof(s1));
</script>
```

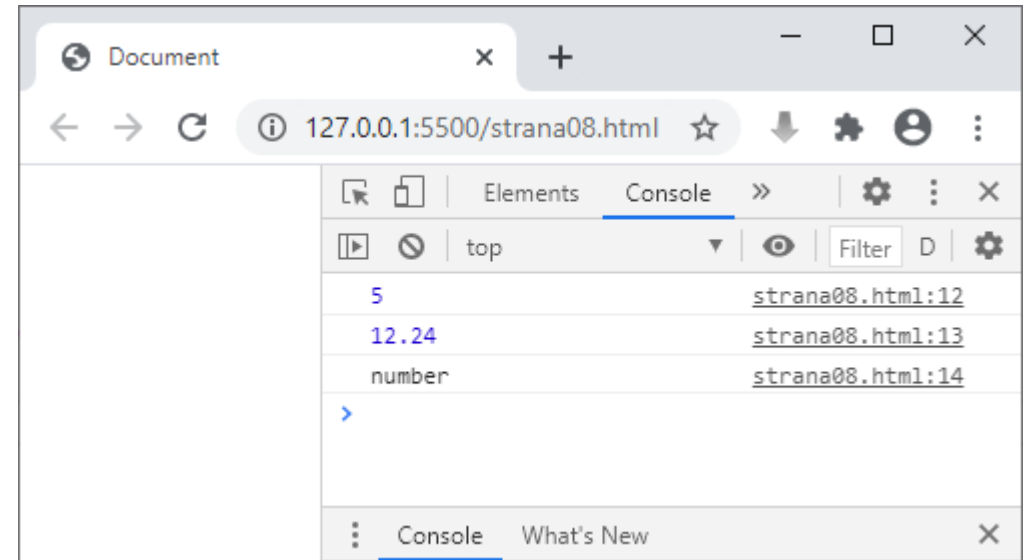


Stringovi se mogu pisati između jednostrukih navodnika, dvostrukih ili iskošenih navodnika

typeof je operator koji se koristi za određivanje tipa date vrednosti ili promenljive

JavaScript brojevi

```
<script>  
var a = 5;  
var b = 12.24;  
console.log(a);  
console.log(b);  
console.log(typeof(a));  
</script>
```



Templejt literali

```
<script>
  var a = 10;
  var b = 20;

  var s1 = `Rezultat:  ${a} + ${b} = ${a + b}`;

  var s2 = `

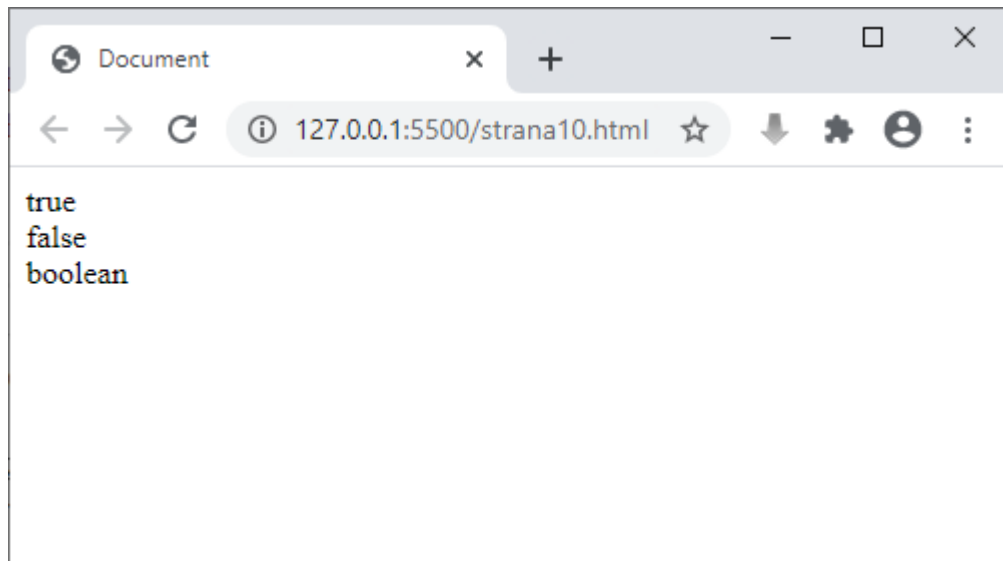



```

JavaScript logičke vrednosti

```
<script>
  var a = true;
  var b = false;

  document.write(a + "<br>");
  document.write(b + "<br>");
  document.write(typeof (a));
</script>
```



Aritmetički operatori

- Sabiranje (+)
- Oduzimanje (-)
- Množenje (*)
- Deljenje (/)
- Celobrojni ostatak (%)
- Inkerementiranje (++)
- Dekrementiranje (--)

Operatori dodeljivanja

```
<script>  
  x = y;  
  x += y; //x = x + y  
  x -= y; //x = x - y  
  x *= y; //x = x * y  
  x /= y; //x = x / y  
  x %= y; //x = x % y  
</script>
```

Operatori poređenja

- Promenljive se konvertuju u isti tip pre poređenja sledećim operatorima:
 - == (jednako)
 - != (različito)
 - > (veće)
 - < (manje)
 - >= (veće ili jednako)
 - <= (manje ili jednako)
- Promenljive se ne konvertuju u isti tip pre poređenja
 - === (jednakost po vrednosti i tipu)
 - !== (nejednakost po vrednosti ili tipu)

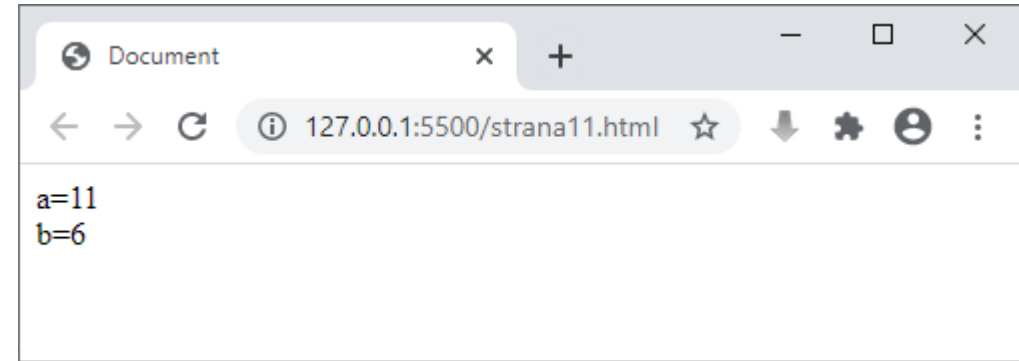
Logički operatori

- $x \ \&\& \ y$ (logičko I)
- $x \ || \ y$ (logičko ILI)
- $!x$ (logičko NE)

Rad sa operandom tipa string

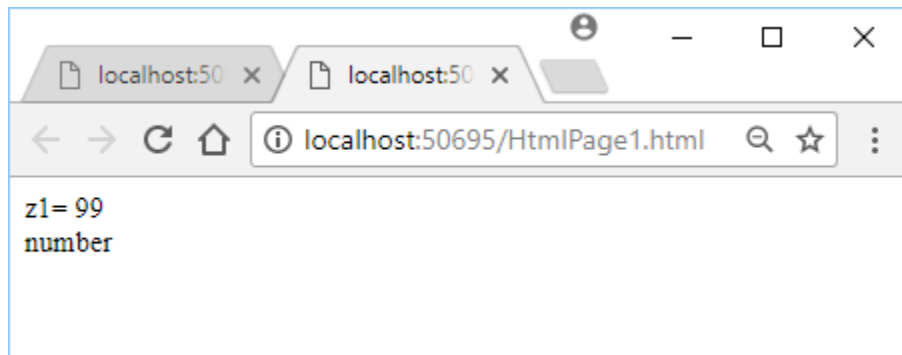
- Sabiranje – operand koji nije tipa string se konvertuje u string i radi se konkatencija (sabiranje, tj. spajanje stringova)
- Ostale operacije – string se konvertuje u broj, a zatim se obavlja operacija; ako konverzija ne uspe, dobija se vrednost **NaN** (Not a Number)

```
<script>
  var a = 1 + "1";
  var b = 2 * "3";
  document.write("a=" + a + "<br>");
  document.write("b=" + b + "<br>");
</script>
```



Rad sa brojevima

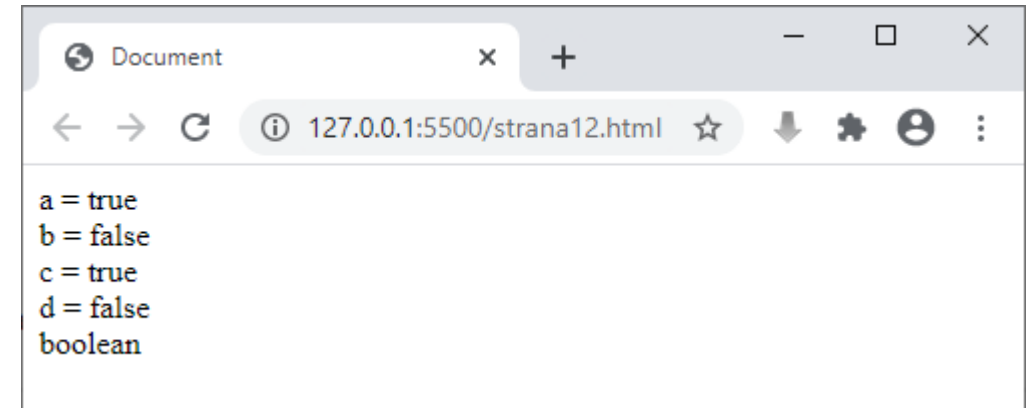
```
<script>
  var x1 = 55;
  var y1 = 44;
  var z1 = x1 + y1;
  document.write("z1= " + z1 + "<br>");
  document.write(typeof (z1));
</script>
```



Rad sa boolean vrednostima

```
<script>
  var a = true;
  var b = false;
  var c = a || b;
  var d = a && b;

  document.write("a = " + a + "<br>");
  document.write("b = " + b + "<br>");
  document.write("c = " + c + "<br>");
  document.write("d = " + d + "<br>");
  document.write(typeof (a));
</script>
```



Uslovni operator

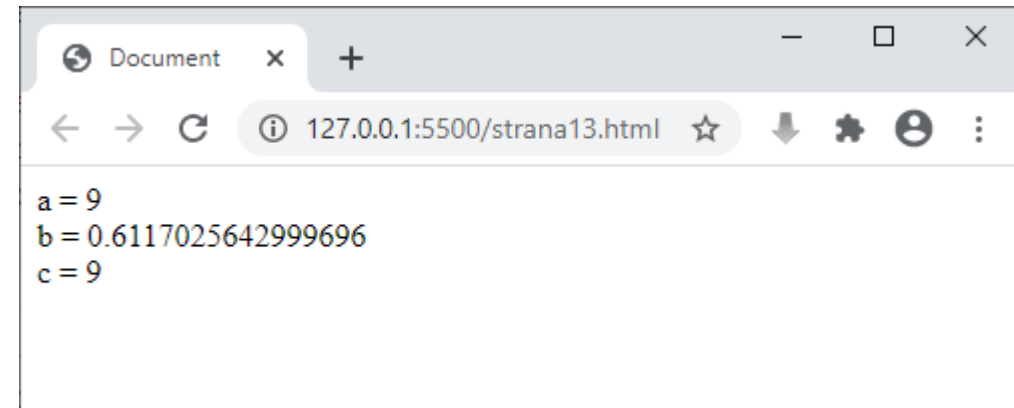
`x = (uslov) ? value1 : value2;`

`Math.random() [0, 1)`

```
<script>
  var a = Math.round(10 * Math.random());
  var b = 10 * Math.random();

  var c = a > b ? a : b;

  document.write("a = " + a + "<br>");
  document.write("b = " + b + "<br>");
  document.write("c = " + c + "<br>");
</script>
```



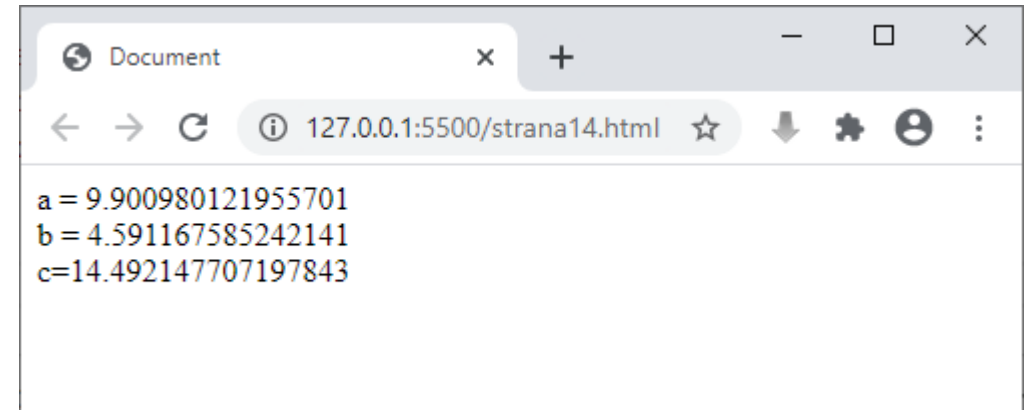
Naredba if

```
<script>

  var a = 10 * Math.random();
  var b = 10 * Math.random();
  var c=0;

  if (a > b) {
    c = a + b;
  }
  else {
    c = a * b;
  }
  document.write("a = " + a + "<br>");
  document.write("b = " + b + "<br>");
  document.write("c=" + c + "<br>");

</script>
```



Rad sa datumom

```
var d = new Date();
```

Metoda	Opis
getDate()	Get the day as a number (1-31)
getDay()	Get the weekday a number (0-6)
getFullYear()	Get the four digit year (yyyy)
getHours()	Get the hour (0-23)
getMilliseconds()	Get the milliseconds (0-999)
getMinutes()	Get the minutes (0-59)
getMonth()	Get the month (0-11)
getSeconds()	Get the seconds (0-59)
getTime()	Get the time (milliseconds since January 1, 1970)

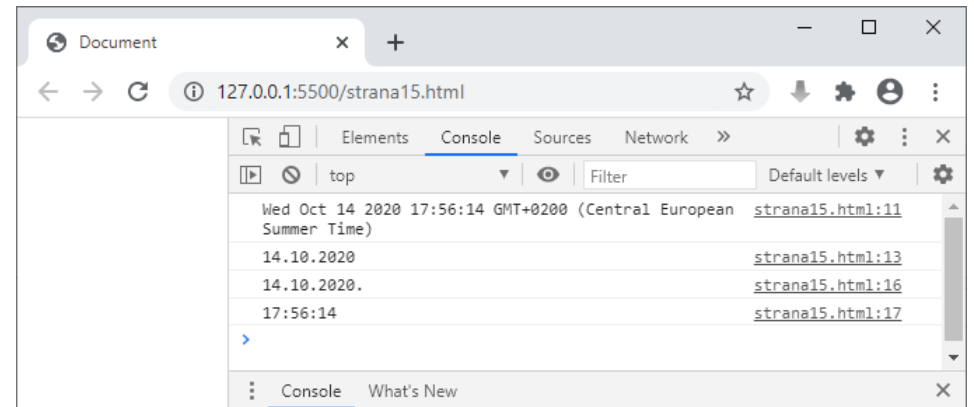
Prikaz datuma i vremena

```
<script>
var d1 = new Date();
console.log(d1);

var s1 = d1.getDate() + "." + ( d1.getMonth() +1 )
+ "." + d1.getFullYear();
console.log(s1);

var s2 = d1.toLocaleDateString("sr-Latn-RS");
var s3 = d1.toLocaleTimeString("sr-Latn-RS");

console.log(s2);
console.log(s3);
</script>
```



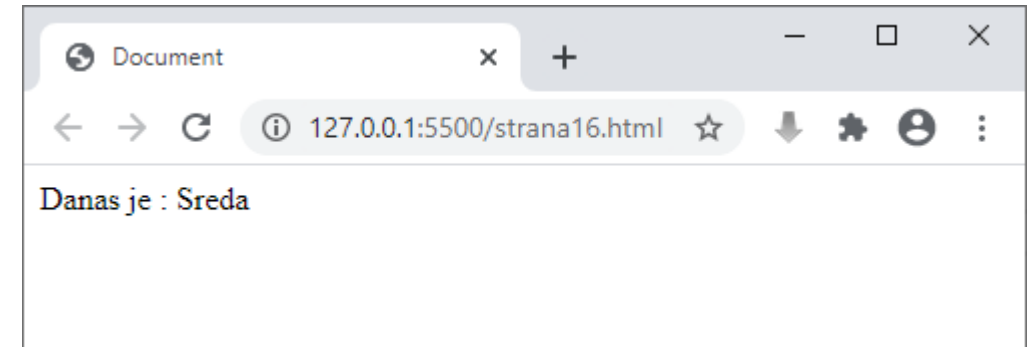
if-else if- else

```
<script>
var a = 10* Math.random();
var b = 10* Math.random();

if (a > b) {
console.log(` ${a} > ${b} `);
}
else if (a==b) {
console.log(` ${a} == ${b} `);
}
else{
console.log(` ${a} < ${b} `);
}
</script>
```

Switch selekcija

```
<script>
  var redniBroj = new Date().getDay();
  var dan = "";
  switch (redniBroj) {
    case 0:
      dan = "Nedelja";
      break;
    case 1:
      dan = "Ponedeljak";
      break;
    case 2:
      dan = "Utorak";
      break;
    case 3:
      dan = "Sreda";
      break;
    case 4:
      dan = "Cetvrtak";
      break;
    case 5:
      dan = "Petak";
      break;
    case 6:
      dan = "Subota";
      break;
  }
  document.write("Danas je : " + dan);
</script>
```



While petlja

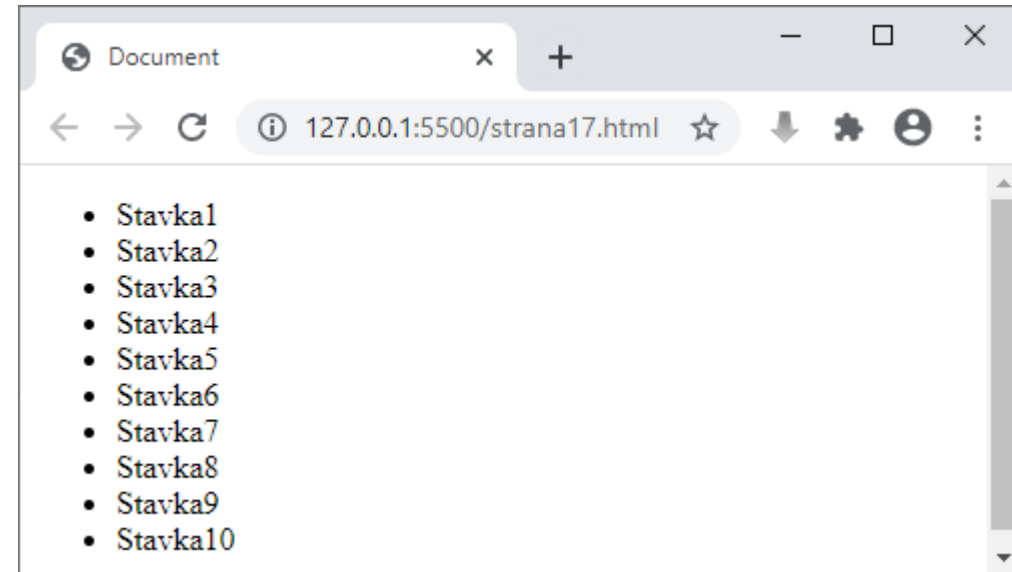
```
<script>
let i = 0;

let s = "<ul>";

while (i<10) {
s += `<li>Stavka${i+1}</li>`;
i++;
}

s += "</ul>";

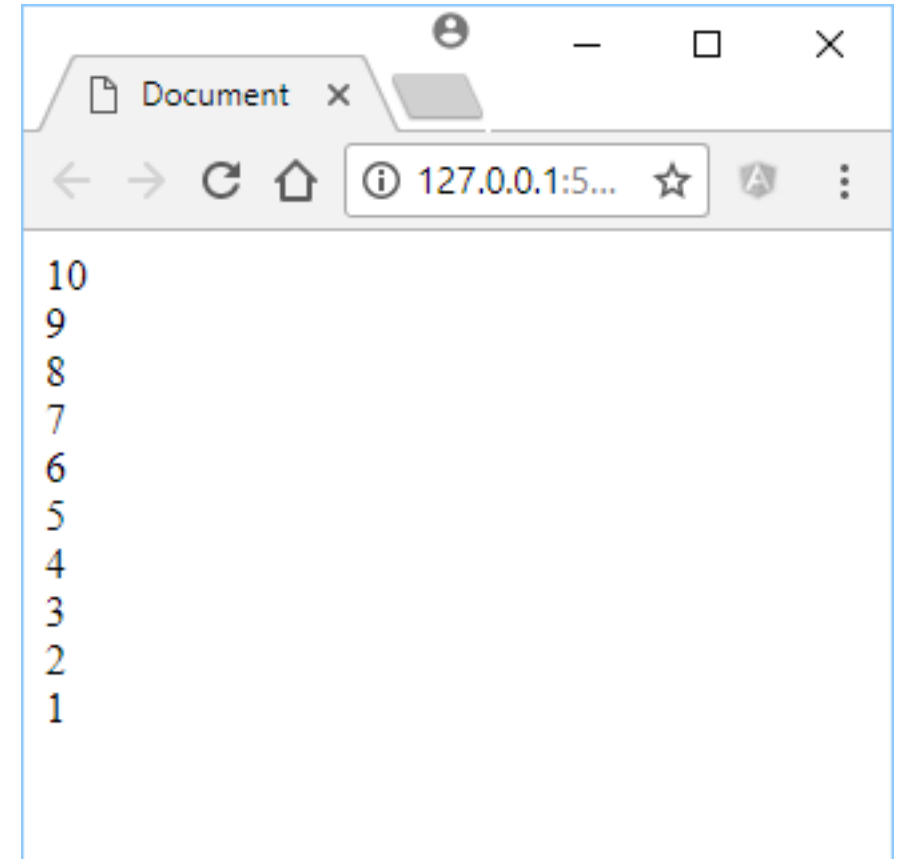
document.write(s);
</script>
```



Do-while petlja

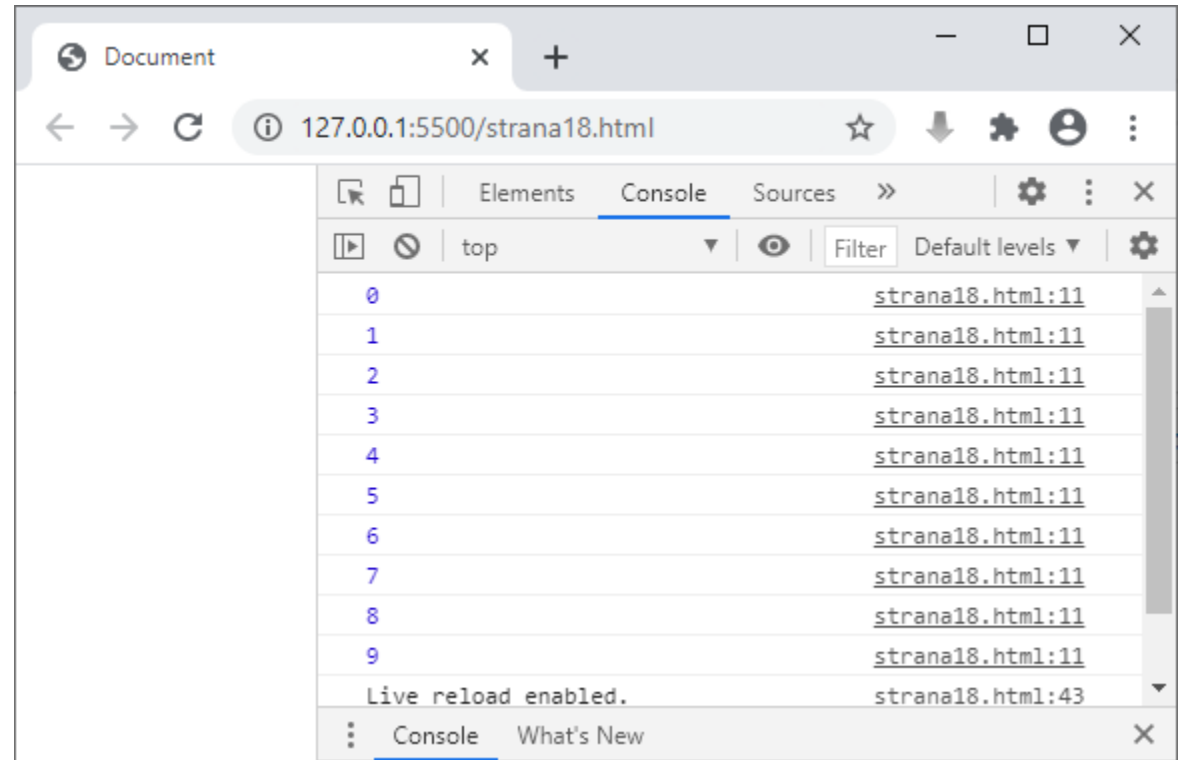
```
<script>
  let i = 10;

  do {
    document.write(i + "<br>");
    i--;
  } while (i>0);
</script>
```



For petlja

```
<script>
for(let i=0; i<10; i++){
console.log(i);
}
</script>
```



Definisanje funkcije

- Funkcija je imenovana sekvenca naredbi koja služi za izvršavanje specifičnog zadatka
- Argumentima funkcije se može pristupiti samo unutar tela funkcije
- Funkcija može vratiti vrednost
- Funkcija može definisati lokalnu promenljivu
- Globalne promenljive definisane izvan funkcije vidljive su u svim funkcijama skript sekcije

```
function aName( argument1, argument2, ...,argumentN ) {  
    statement1;  
    statement2;  
    ...  
    statementN;  
}
```

Primer funkcije

```
<script>
  function Pomnozi(a, b) {
    console.log(`a = ${a}`);
    console.log(`b = ${b}`);

    return a * b;
  }

  var a1 = 5;
  var b1 = 8;

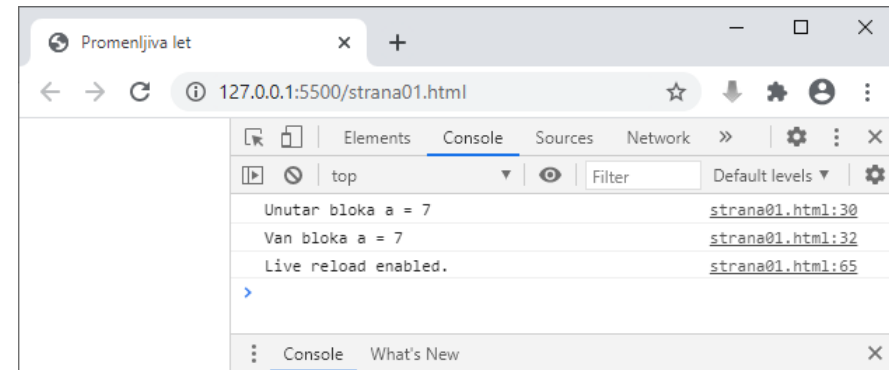
  var rezultat = Pomnozi(a1,b1);
  console.log(rezultat);

  a1 = 7;
  b1 = 12;
  rezultat = Pomnozi(a1,b1);
  console.log(rezultat);
</script>
```

Oblast važenja promenljivih

- Java Script promenljiva deklarirana sa **var** može biti globalna ili lokalna kada se definiše unutar neke funkcije
- Globalna promenljiva se definiše izvan funkcije
- Promenljiva tipa **var** ne podržava oblast važenja na nivou bloka
- Oblast važenja **var** promenljive je ograničena na funkciju u kojoj je promenljiva definisana

```
<script>
  function VarTest() {
    {
      var a = 7;
      console.log(`Unutar bloka a = ${a}`);
    }
    console.log(`Van bloka a = ${a}`);
  }
  VarTest();
</script>
```

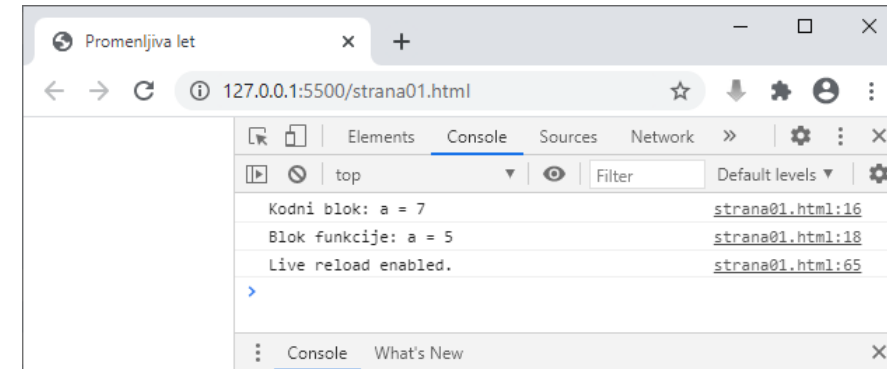


Promenljiva deklarirana sa let

```
<script>
  function LetTest() {
    let a = 5;

    //kodni blok
    {
      let a = 7;
      console.log(`Kodni blok: a = ${a}`);
    }
    console.log(`Blok funkcije: a = ${a}`);
  }

  LetTest();
</script>
```



Vidljivost let promenljive je ograničena na blok.

Funkcija alert()

Prikazuje prozor sa porukom i dugmetom OK
Mora se zatvoriti prozor da bi se koristili drugi delovi dokumenta

```
<script>  
    alert("Pozdrav svima");  
    //window.alert("Pozdrav svima");  
</script>
```



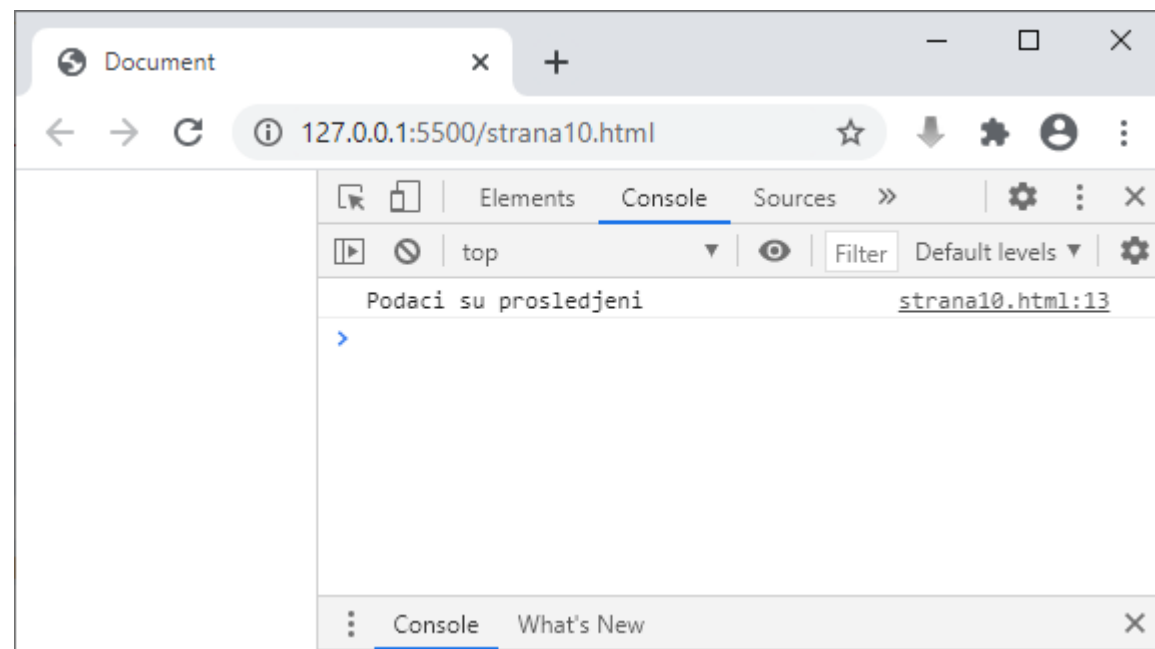
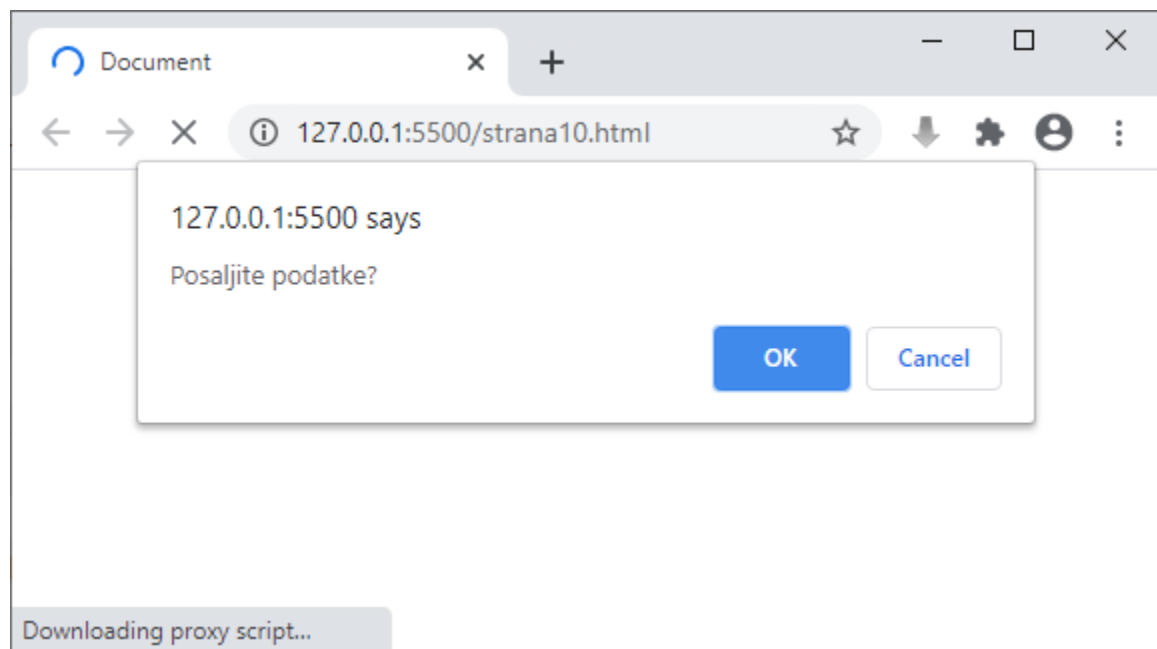
Funkcija confirm()

- Uzima u obzir akciju korisnika
- vraća rezultat (true ili false)

```
<script>
  var rezultat = confirm("Posaljite podatke?");

  if (rezultat) {
    console.log("Podaci su prosledjeni");
  }
  else{
    console.log("Nisu prosledjeni podaci");
  }
</script>
```

Prozor confirm()

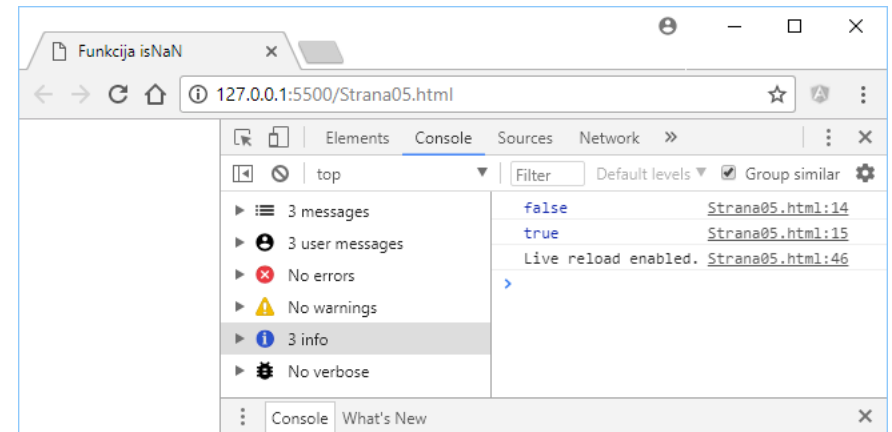


Funkcija isNaN()

- Određuje da li vrednost koja se prosleđuje funkciji predstavlja ilegalan broj NaN
- Funkcija vraća logičku vrednost true ili false

```
<script>
  var a = isNaN(123);
  var b = isNaN("Hello");

  console.log(a);
  console.log(b);
</script>
```

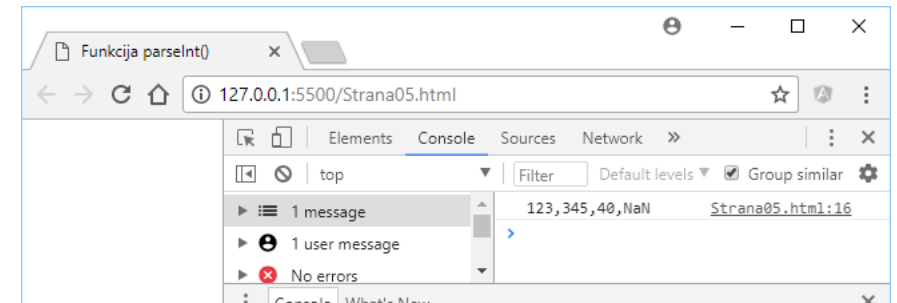


Funkcije parseInt() , parseFloat()

- Funkcija **parseInt()** parsira string i vraća ceo broj
- Ako prvi karakter ne može da se konvertuje funkcija parseInt() vraća vrednost **NaN**

```
<script>
  var c1 = parseInt("123");
  var c2 = parseInt("345.567");
  var c3 = parseInt("40godina");
  var c4 = parseInt("godina40");

  console.log(` ${c1}, ${c2}, ${c3}, ${c4} `);
</script>
```



Pitanje 1

Ukoliko html strana u svojoj **<body>** sekciji sadrži javascript kod tada se on izvršava :

- a) na serveru
- b) na klijentu
- c) i na serveru i na klijentu

Odgovor: b

Pitanje 2

Jezik javascript ne poznaje sledeći tip podataka:

- a. float
- b. number
- c. boolean

Odgovor: a

Pitanje 3

Za deklarisanje javascript promenljive čija je oblast važenja ograničena na funkciju koristi se ključna reč:

- a. variable
- b. var
- c. unknown

Odgovor: b

Pitanje 4

Ključna reč let u JavaScript-u se koristi:

- a. za deklarisanje promenljive koja je dostupna samo unutar bloka koda
- b. za deklarisanje konstantne vrednosti
- c. za deklarisanje promenljive koja je dostupna na nivou funkcija

Odgovor: a

Pitanje 5

Šta se prikazuje na konzoli nakon izvršavanja sledećeg javascript koda:

```
<script>  
  var a = 2 * "3";  
  console.log("a=", a);  
</script>
```

- a. a= Nan
- b. a= 2*3
- c. a= 6

Odgovor: c