



Internet programiranje

predavanje 08

Prof. dr Miroslav Lutovac
mlutovac@viser.edu.rs



JavaScript Objects

- U realnom životu, automobil je object
- Automobil ima properties (težina, boja), i metode (kreni i stani)
- Svi automobili imaju iste properties, ali se osobine razlikuju od auta do auta
- Svi automobili imaju iste metode, ali se metode izvršavaju u različito vreme



Object
auto

Properties
ime, model, boja

Methods
kreni, vozi



auto.ime = Fiat
auto.model = 500L
auto.tezina = 850kg
auto.boja = crvena

auto.kreni()
auto.vozi()
auto.koci()
auto.stani()



JavaScript Objects

- Primer promenljive (**variable**) imenovana kao auto

```
var auto = "Fiat"; //dodela jedne vrednosti
```

- Objects su takođe variables
- Objects mogu da sadrže brojne vrednosti
- JavaScript može da dodeljuje veći broj vrednosti (Fiat, 500L, bela) promenljivoj (**variable**) koja je imenovana kao auto:

```
var auto = {tip: "Fiat", model:"500L",boja:"bela"};
```

JavaScript objekti su kontejneri za **imenovane vrednosti**

Vrednosti su napisane kao parovi **ime:vrednost**

ime i vrednosti su razdvojeni dvotačkom :

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>Hocu da kupim:</p>
5 <p id="demo"></p>
6 <script>
7   var auto = {tip:"Fiat", model:"500L", boja:"bela"};
8   document.getElementById("demo").innerHTML =
9     ("auto marke <b>" + auto.tip +
10    "</b>, model <b>" + auto.model +
11    "</b>, boja <b>" + auto.boja + "</b>");
12 </script>
13
14 </body>
15 </html>
```

Hocu da kupim:

auto marke **Fiat**, model **500L**, boja **bela**

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p id="demo"></p>
5 <script>
6 var auto = {
7     tip:"Fiat",
8     model:"500L",
9     boja:"bela"
10 };
11 document.getElementById("demo").innerHTML =
12 ("Hocu da kupim auto marke <b>" + auto.tip +
13 "</b>, model <b>" + auto.model +
14 "</b>, boja <b>" + auto.boja + "</b>");
15 </script>
16 </body>
17 </html>
```

Hocu da kupim auto marke **Fiat**,
model **500L**, boja **bela**

Definicija objekta može da
bude u više linija koda



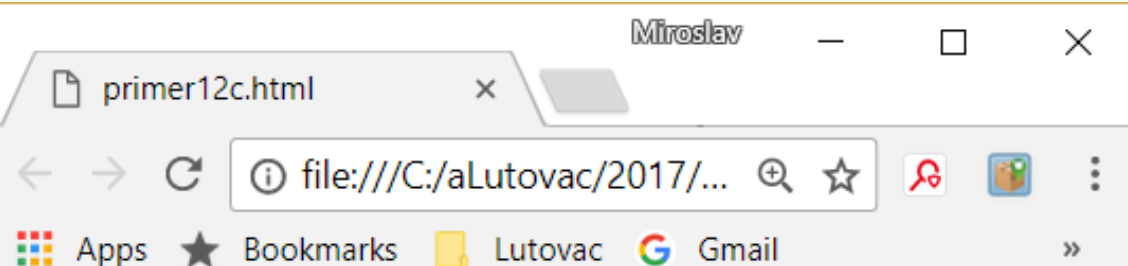
Accessing Object Properties

- Može se pristupiti properties na 2 načina:

`imeObjekta.osobinaObjekta`

`imeObjekta["osobinaObjekta"]`

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p id="demo"></p>
5 <script>
6 var auto = {
7     tip:"Fiat",
8     model:"500L",
9     boja:"bela"
10 };
11 document.getElementById("demo").innerHTML =
12 ("Pristup objektu imeObjekta.osobinaObjekta" +
13 "<br>imeObjekta je <b>auto</b>," +
14 "<br>osobinaObjekta je <b>" + auto.tip + "</b>");
15 </script>
16 </body>
17 </html>
```

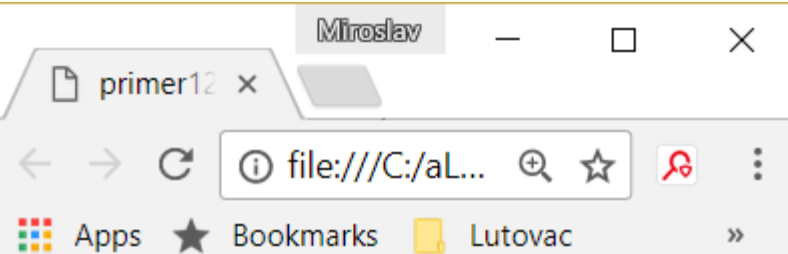


Pristup objektu imeObjekta.osobinaObjekta
imeObjekta je **auto**,
osobinaObjekta je **Fiat**

pristup JavaScript objektima
imeObjekta.osobinaObjekta



```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p id="demo"></p>
5 <script>
6 var auto = {
7     tip:"Fiat",
8     model:"500L",
9     boja:"bela"
10 };
11 document.getElementById("demo").innerHTML =
12 ('Pristup objektu imeObjekta["osobinaObjekta"]' +
13 "<br>imeObjekta je <b>auto</b>," +
14 "<br>osobinaObjekta je <b>" + auto["tip"] + "</b>");
15 </script>
16 </body>
17 </html>
```



Pristup objektu
imeObjekta["osobinaObjekta"]
imeObjekta je **auto**,
osobinaObjekta je **Fiat**

pristup JavaScript objektima
imeObjekta["osobinaObjekta"]



JavaScript Object Methods

- Metode su aktivnosti koje bi mogle da izvrše objekti
- JavaScript objekti su kontejneri za imenovane vrednosti koje nazivamo properties ili **methods**

```
function() {return auto.kreni + ... + auto.stani;}
```



Pristup Metodi objekta

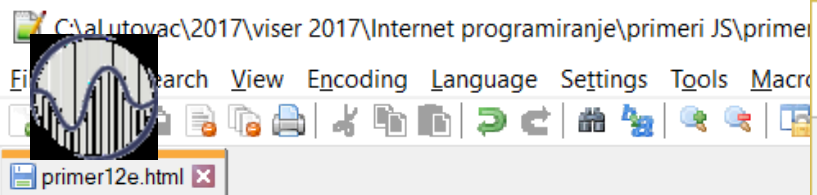
- Pristupa se metodi objekta:

`imeObjekta.imeMetode ()`

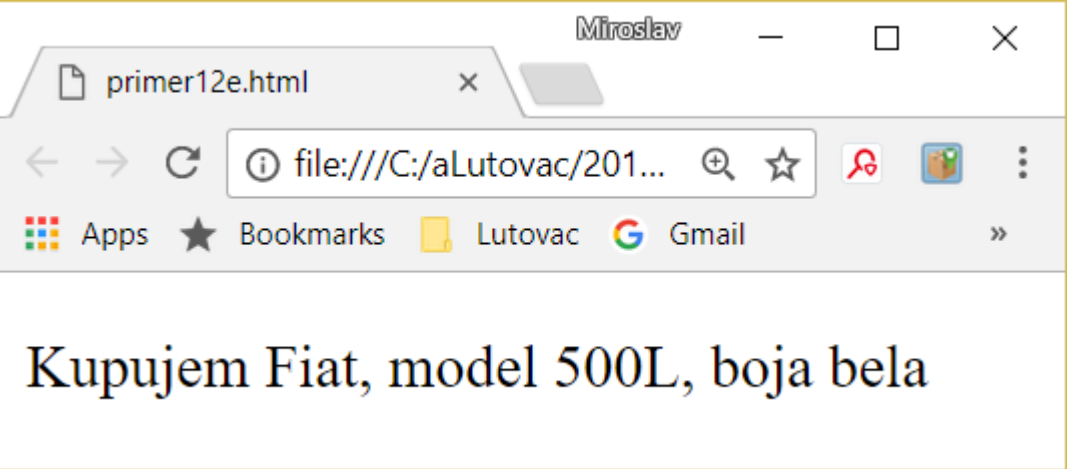
`Ime = person.fullName()`

- Ako se pristupa metodi bez zagrada (), vraća se definicija metode:

`ime = person.fullName`



```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p id="demo"></p>
5 <script>
6 var auto = {
7     tip: "Fiat",
8     model: "500L",
9     oprema1: "klima",
10    oprema2: "alat",
11    boja : "bela ",
12    id      : 5678,
13    tipModelBoja : function() {
14        return "Kupujem "+this.tip+", model "+
15        this.model+", boja "+this.boja;
16    }
17 };
18 document.getElementById("demo").innerHTML =
19 auto.tipModelBoja();
20 </script>
21 </body>
22 </html>
```



pristup JavaScript objektima
auto.tipModelBoja();

```
primer12f.html
<!DOCTYPE html>
<html>
<body>
<p id="demo"></p>
<script>
var auto = {
  tip: "Fiat",
  model: "500L",
  oprema1: "klima",
  oprema2: "alat",
  boja : "bela ",
  id      : 5678,
  tipModelBoja : function() {
    return "Kupujem "+this.tip+", model "+
    this.model+", boja "+this.boja;
  }
};
document.getElementById("demo").innerHTML =
auto.tipModelBoja;
</script>
</body>
</html>
```

pristup JavaScript objektima
bez zagrada **auto.tipModelBoja;**

```
function () { return "Kupujem  
"+this.tip+", model "+ this.model+",  
boja "+this.boja; }
```

metod je u stvari definicija funkcije
zapamćena kao vrednost property



JavaScript Scope

- Scope određuje pristupačnost (vidljivost) promenljive, objekta i funkcije iz različitih delova koda
- U JavaScript-u postoje dve vrste:
 - Local scope
 - Global scope
- JavaScript ima function scope: Svaka funkcija stvara novi scope
- Promenljive definisane unutar funkcije nisu vidljive izvan funkcije



Lokalne JavaScript promenljive

- Promenljive deklarisanе unutar JavaScript funkcije, postaju lokalne za funkciju
- Lokalne promenljive imaju lokalni scope: može im se pristupiti samo unutar funkcije
- Pošto se lokalne promenljive mogu prepoznati samo unutar funkcije, promenljive sa istim imenom mogu da se koriste u različitim funkcijama
- Lokalne promenljive nastaju kada se počinje neka funkcija, i nestaju kada se završi ta funkcija



primer12g.html

file:///C:/aLutovac/2017/viser%202017/Internet%20... ☆

Apps ★ Bookmarks Lutovac Gmail Google Translate eZaposleni »

Lokalna promenljiva **autoIme** **nije vidljiva** izvan funkcije:

Tip promenljive izvan funkcije je undefined

Lokalna promenljiva **autoIme** **vidljiva** je unutar funkcije:

Tip promenljive unutar funkcije je +string

C:\aLutovac\2017\viser 2017\Internet programiranje\primeri JS\primer12g.html - No

File Edit Search View Encoding Language Settings Tools Macro Run Plugi



primer12g.html x

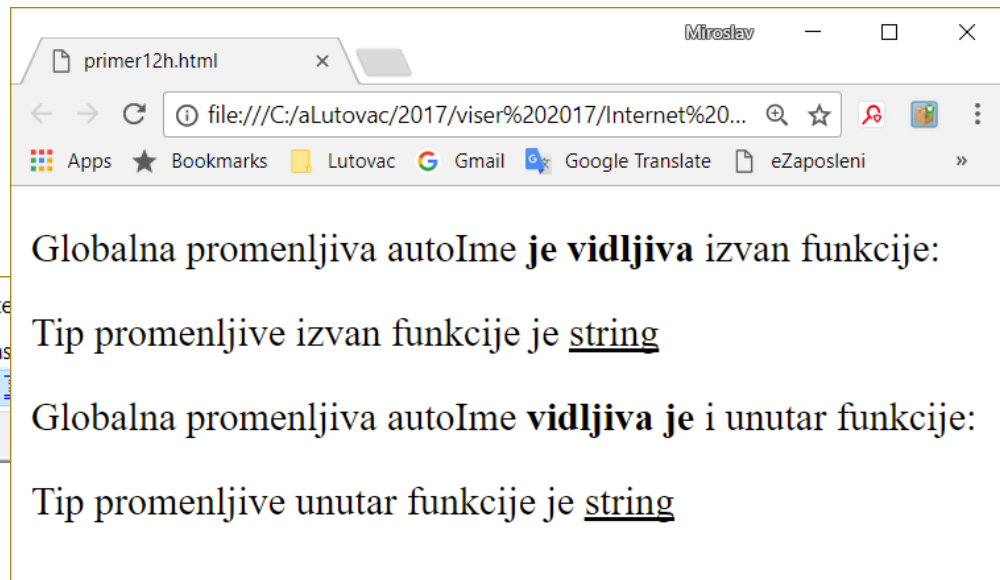
```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>Lokalna promenljiva autoIme <b>nije vidljiva</b> izvan funkcije:</p>
5 <p id="demo"></p>
6 <p>Lokalna promenljiva autoIme <b>vidljiva je</b> unutar funkcije:</p>
7 <p id="demo2"></p>
8 <script>
9 myFunction();
10 document.getElementById("demo").innerHTML =
11 "Tip promenljive izvan funkcije je <u>"+typeof autoIme+"</u>";
12 function myFunction() {
13     var autoIme = "Fiat";
14     document.getElementById("demo2").innerHTML =
15     ("Tip promenljive unutar funkcije je +<u>"+typeof autoIme+"</u>");
16 }
17 </script>
18 </body>
19 </html>
```

Lokalna promenljiva



Lokalne JavaScript promenljive

- Promenljive deklarisanе izvan funkcija postaju globalne
- Globalna promenljiva ima globalnu vidljivost
- Svi skriptovi i funkcije na veb strani mogu da pristupe promenljivoj
- U JavaScript-u, objekti i funkcije su takođe promenljive



Globalna promenljiva autoIme **je vidljiva** izvan funkcije:

Tip promenljive izvan funkcije je string

Globalna promenljiva autoIme **vidljiva je** i unutar funkcije:

Tip promenljive unutar funkcije je string

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>Globalna promenljiva autoIme je vidljiva izvan funkcije:</p>
5 <p id="demo"></p>
6 <p>Globalna promenljiva autoIme vidljiva je i unutar funkcije:</p>
7 <p id="demo2"></p>
8 <script>
9   var autoIme = "Fiat";
10  myFunction();
11  document.getElementById("demo").innerHTML =
12  "Tip promenljive izvan funkcije je <u>" + typeof autoIme + "</u>";
13  function myFunction() {
14    document.getElementById("demo2").innerHTML =
15    ("Tip promenljive unutar funkcije je <u>" + typeof autoIme + "</u>");
16  }
17 </script>
18 </body>
19 </html>
```

Globalna promenljiva



Automatski globalno promenljiva

- Ako se dodeljuje vrednost promenljivoj koja nije deklarirana, promenljiva će automatski postati globalna promenljiva
- Sledeći kod deklarira promenljivu kao globalnu iako je vrednost dodeljene unutar funkcije



primer12i.html

file:///C:/aLutovac/2017/viser%202017/Internet%20pr...

Apps ★ Bookmarks Lutovac Gmail Google Translate eZaposleni US »

Lokalna promenljiva autoIme **nije vidljiva** izvan funkcije:

Tip promenljive izvan funkcije je string

Lokalna promenljiva autoIme **vidljiva je** unutar funkcije:

Tip promenljive unutar funkcije je +string

```
C:\aLutovac\2017\viser 2017\Internet programiranje\primeri JS\primer12i.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugin
primer12i.html
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>Lokalna promenljiva autoIme <b>je vidljiva</b> izvan funkcije:</p>
5 <p id="demo"></p>
6 <p>Lokalna promenljiva autoIme <b>vidljiva je</b> i unutar funkcije:</p>
7 <p id="demo2"></p>
8 <script>
9 myFunction();
10 document.getElementById("demo").innerHTML =
11 "Tip promenljive izvan funkcije je <u>" + typeof autoIme + "</u>";
12 function myFunction() {
13 // var autoIme = "Fiat";
14 autoIme = "Fiat";
15 document.getElementById("demo2").innerHTML =
16 ("Tip promenljive unutar funkcije je +<u>" + typeof autoIme + "</u>");
17 }
18 </script>
19 </body>
20 </html>
```

dodeljena vrednost globalnoj
promenljivoj unutar funkcije



Globalne promenljive u HTML

- Sa JavaScript-om, globalna vidljivost je celokupno JavaScript okruženje
- U HTML-u, globalna vidljivost je window objekat
- Sve globalne promenljive pripadaju window objektu
- **NEMOJTE da kreirate globalne promenljive sve dok to nije neophodno**



C:\aLutovac\2017\viser 2017\Internet programira

File Edit Search View Encoding Language Se



primer12j.html

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>U HTML-u, sve globalne promenljive
5 postaju window promenljive</p>
6 <p id="demo"></p>
7 <script>
8   var autoIme = "Fiat";
9   document.getElementById("demo").innerHTML =
10   "window.autoIme koristimo za prikaz " +
11   window.autoIme;
12 </script>
13 </body>
14 </html>
```

U HTML-u, sve globalne promenljive postaju window promenljive

window.autoIme koristimo za prikaz Fiat

Globalna promenljiva u HTML-u



Upozorenja

- **NEMOJTE** da kreirate globalne promenljive sve dok to nije neophodno
- Globalne promenljive (ili funkcije) mogu da prebrišu window promenljive (ili funkcije)
- Bilo koja funkcija, uključujući window objekt, može da prebriše globalne promenljive i funkcije
- Globalne promenljive se ne kreiraju automatski u "Strict Mode"



Životni vek JavaScript promenljivih

- Životni vek JavaScript promenljive započinje njenom deklarisanjem
- Lokalne promenljive nestaju kada se završe funkcije
- U veb pregledaču, globalne promenljive se brišu (nestaju) kada se zatvori prozor pregledača, ali ostaju dostupne novim stranama u istom prozoru
- Argumenti funkcije (parametri) funkcionišu kao lokalne promenljive unutar funkcije



JavaScript događaji (Events)

- HTML events su „stvari“ koje se događaji koji se dešavaju HTML elementima
- Kada se JavaScript koristi u HTML stranama, JavaScript može da ima „reakciju“ na ovakav događaj



HTML događaji (Events)

- HTML događaj može da bude nešto što pregledač radi, ili nešto što korisnik radi
- Primeri HTML događaja:
 - HTML veb strana je završila učitavanje
 - HTML input polje je promenjeno
 - HTML dugme je kliknuto
- Često je potrebno da se uradi nešto kada se desi neki događaj



HTML događaji (Events)

- JavaScript omogućava da se izvrši kod kada se detektuje neki događaj
- HTML dozvoljava atributima za upravljanje događajima, koristeći JavaScript kod, da budu dodati HTML elementima:

`<element event='some JavaScript'>`

`<element event="some JavaScript">`

primer12k.html

file:///C:/aLutovac/2017/viser%202017...

Sada je vreme:

Sada je vreme:

Sun Nov 12 2017 22:03:11 GMT+0100 (Central Europe Standard Time)

Sada je vreme:

Sun Nov 12 2017 22:03:11 GMT+0100 (Central Europe Standard Time)

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <button onclick="document.getElementById('demo').innerHTML=Date()"
5 ">Sada je vreme: </button>
6 <p id="demo"></p>
7 </body>
8 </html>
```

onclick attribute dodat button element-u
JS kod menja sadržaj elements sa id="demo"

onclick attribute dodat button element-u
JS menja sadržaj elementa sa (**this.innerHTML**)

Sun Nov 12 2017 22:16:37 GMT+0100 (Central Europe Standard Time)

Sada je vreme:

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <button onclick="this.innerHTML=Date()">Sada je vreme:</button>
5 </body>
6 </html>
```

Klikni na dugme da prikažeš v

Sada je vreme:

Klikni na dugme da prikažeš vreme.

Sada je vreme:

Sun Nov 12 2017 22:23:42 GMT+0100 (Central Europe Standard Time)

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>Klikni na dugme da prikažeš vreme.</p>
5 <button onclick="displayDate()">Sada je vreme:</button>
6 <script>
7 function displayDate() {
8     document.getElementById("demo").innerHTML = Date();
9 }
10 </script>
11 <p id="demo"></p>
12 </body>
13 </html>
```

Da se vide atributi događaja pozvane funkcije



Najčešći HTML događaji

- Događaj - Opis
- onchange - HTML element je promenjen
- onclick – klik korisnika na HTML element
- onmouseover - korisnik je pomerio miša preko HTML elementa
- onmouseout - korisnik pomerio miša od HTML elementa
- onkeydown - korisnik pritisnuo na tastaturu
- onload - pregledač je završio sa učitavanjem strane
- [JavaScript Reference HTML DOM Events](#)



Šta može JS da uradi?

- Event handlers can be used to handle, and verify, user input, user actions, and browser actions:
 - Nešto što treba da se uradi svaki put kada se strana učita
 - Nešto što treba da se uradi kada se strana zatvori
 - Nešto što treba da se izvrši kada korisnik klikne na dugme
 - Sadržaj koji treba da se verifikuje kada korisnik unese podatak ...



Šta može JS da uradi?

- Brojne metode mogu da se koriste koje omogućavaju da JS radi sa događajima:
 - HTML event attributes može direktno da izvrši JS kod
 - HTML event attributes može da pozove JS funkciju
 - Može da se dodeli HTML elementu sopstveno upravljanje događajem
 - Može da se spreči da događaj bude poslat ili zaustavljen ...



\ escape character

- Obrnuta kosa crta se naziva escape karakter koji omogućava da se specijalni karakteri tretiraju kao stringovi

```
var y = "Mi smo "Delije" sa severa";
```

```
var y = "Mi smo \"Delije\" sa severa";
```

```
var x = 'It\'s alright';
```

```
var x = 'It\'s alright';
```

\'	single quote
\"	double quote
\\	backslash



Prekid linije koda u tekst stringu

- Programeri ne vole da je dužina kodne linije veća od 80 karaktera
- Obrnuta kosa crta \ se koristi za prekid tekst linije u JS

```
document.getElementById("demo").innerHTML = "Zdravo \nMarko!";
```

```
document.getElementById("demo").innerHTML = "Zdravo " +  
"Marko!";
```



JS string metode

- String metode i osobine
 - primitivne vrednost, kao "Marko Markov", ne mogu da imaju osobine ili metode (zato što nisu objekti)
 - sa JS, metode i osobine se mogu primeniti na primitivne vrednosti, zato što JS tretira primitivne vrednosti kao objekte kada izvršava metode i osobine
- String Length
 - length osobina vraća dužinu stringa



C:\aLutovac\2017\viser 2017\Internet programiranje\primeri JS\p

File Edit Search View Encoding Language Settings Tools



primer13a.html

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <h2>JavaScript string osobine</h2>
5 <p>Osobina length vraća duzinu stringa.</p>
6 <p id="demo"></p>
7 <script>
8   var tekst = "ABCdefGH";
9   document.getElementById("demo").innerHTML = "D\
10  uzina stringa " + tekst + " je " + tekst.length;
11 </script>
12 </body>
13 </html>
```

JavaScript string osobine

Osobina length vraća duzinu stringa.

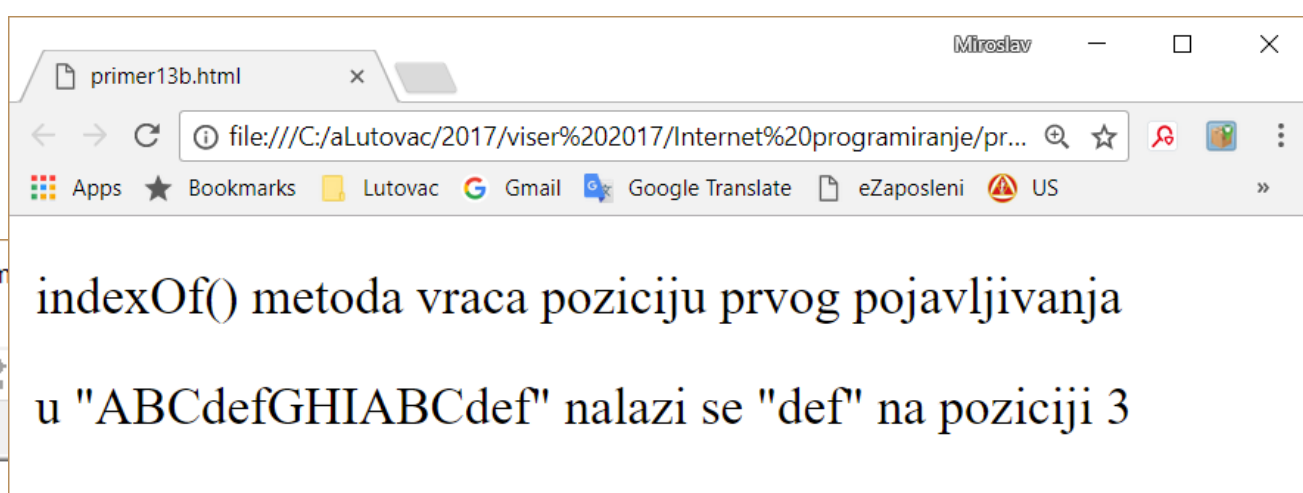
Duzina stringa ABCdefGH je 8

length property

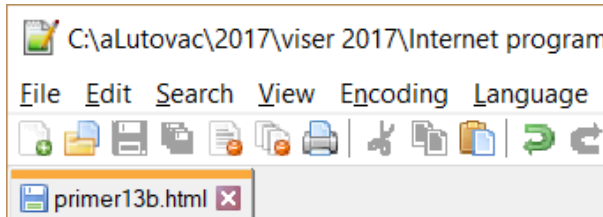


Pronalaženje stringa u stringu

- **indexOf()** metoda vraća indeks (poziciju) prvog pojavljivanja nekog teksta u stringu
- **lastIndexOf()** metoda vraća indeks poslednjeg pojavljivanja nekog teksta u stringu
- **indexOf()** i **lastIndexOf()** metode vraćaju -1 ako nije pronađen tekst
- JS broji poziciju od 0;
0 je prva pozicija u stringu, 1 je druga, 2 je treća ...
- Ako postoji drugi parametar, to je startna pozicija za pretragu



pozicija prvog pojavljivanja



```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>indexOf() metoda vraca
5 poziciju prvog pojavljivanja</p>
6 <p id="demo"></p>
7 <script>
8   var tekst = "ABCdefGHIABCdef";
9   var gdeJe = "def";
10  var pos = tekst.indexOf(gdeJe);
11  document.getElementById("demo").innerHTML = "u\
12  \""+tekst+"\" nalazi se \""+gdeJe+
13  "\" na poziciji "+pos;
14 </script>
15 </body>
16 </html>
```

length : 349 lines : 17 Ln : 17 Col : 1 Sel : 0 | 0 Windows (CR LF) UTF-8 INS



primer13c.html

file:///C:/aLutovac/2017/viser%202017/Internet%20programiranje/primeri%...

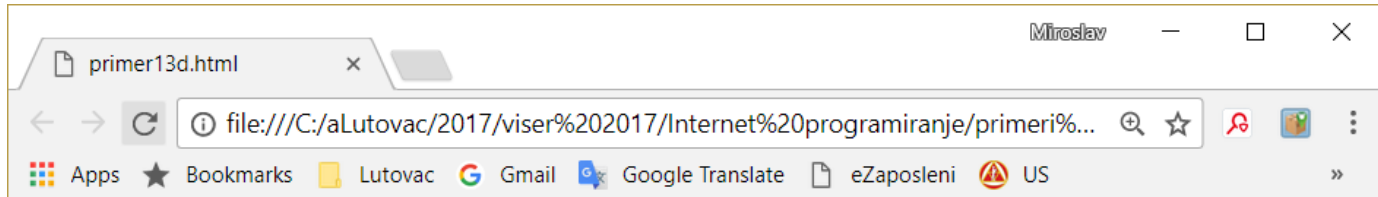
Apps Bookmarks Lutovac Gmail Google Translate eZaposleni US

lastIndexOf metoda vraća poziciju poslednjeg pojavljivanja

u "ABCdefGHIABCdef" nalazi se "def" na poziciji 12

pozicija poslednjeg
pojavljivanja

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>lastIndexOf metoda vraća
5 poziciju poslednjeg pojavljivanja</p>
6 <p id="demo"></p>
7 <script>
8   var tekst = "ABCdefGHIABCdef";
9   var gdeJe = "def";
10  var pos = tekst.lastIndexOf(gdeJe);
11  document.getElementById("demo").innerHTML = "u\
12  \""+tekst+"\" nalazi se \""+gdeJe+
13  "\" na poziciji "+pos;
14 </script>
15 </body>
16 </html>
```

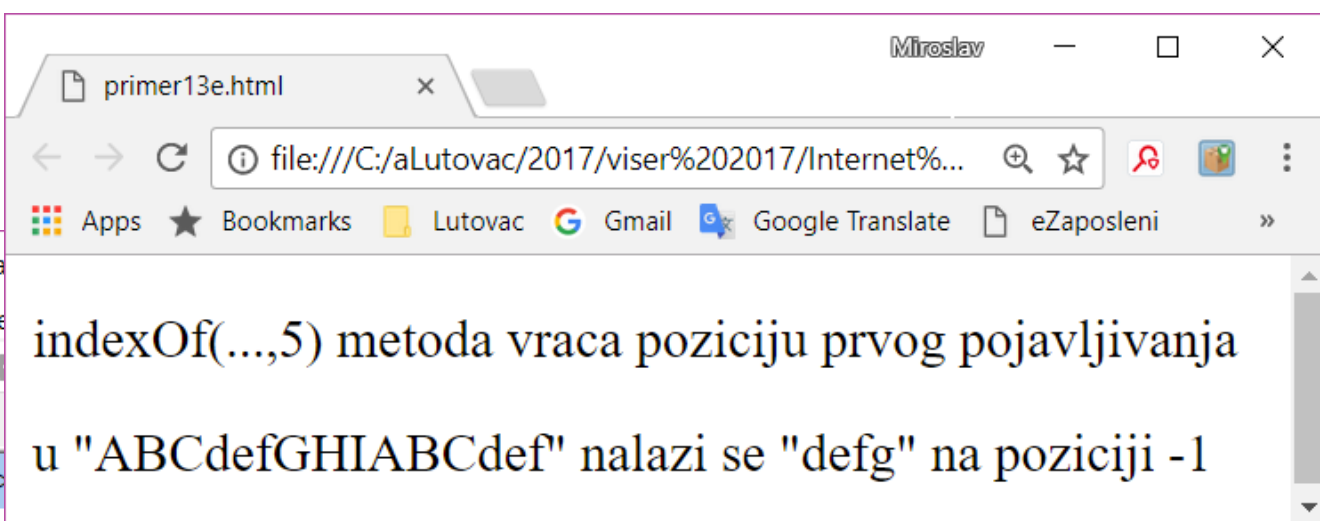



indexOf(...,5) metoda vraca poziciju prvog pojavljivanja

u "ABCdefGHIABCdef" nalazi se "def" na poziciji 12

pozicija prvog pojavljivanja
počev od pozicije 5

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>indexOf(...,5) metoda vraca
5 poziciju prvog pojavljivanja</p>
6 <p id="demo"></p>
7 <script>
8   var tekst = "ABCdefGHIABCdef";
9   var gdeJe = "def";
10  var pos = tekst.indexOf(gdeJe,5);
11  document.getElementById("demo").innerHTML = "u\
12  \""+tekst+"\" nalazi se \""+gdeJe+
13  "\" na poziciji "+pos;
14 </script>
15 </body>
16 </html>
17
```



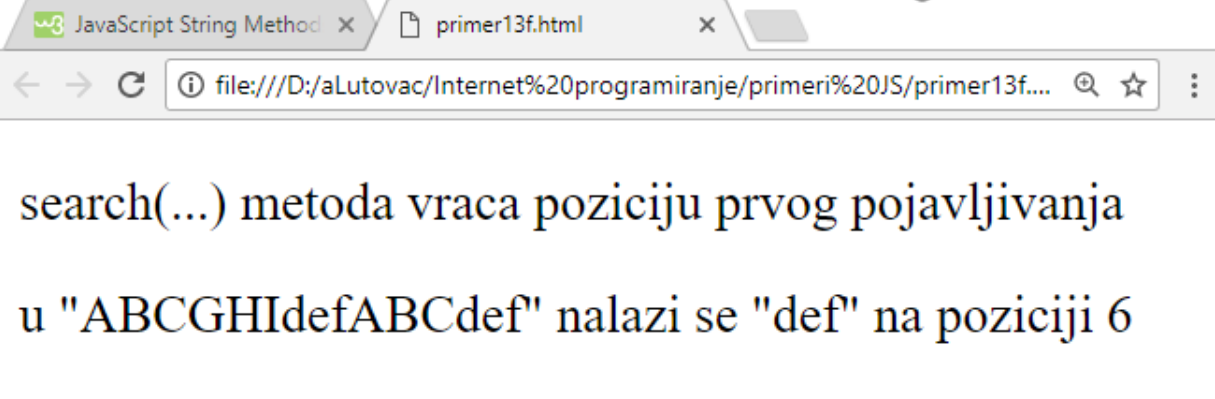
Nema stringa pa je -1

```
C:\aLutovac\2017\viser 2017\Internet programi\primjer13e.html
File Edit Search View Encoding Language
primjer13e.html x
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>indexOf(...,5) metoda vraca
5 poziciju prvog pojavljivanja</p>
6 <p id="demo"></p>
7 <script>
8 var tekst = "ABCdefGHIABCdef";
9 var gdeJe = "defg";
10 var pos = tekst.indexOf(gdeJe);
11 document.getElementById("demo").innerHTML = "u\
12 \""+tekst+"\" nalazi se \""+gdeJe+
13 "\" na poziciji "+pos;
14 </script>
15 </body>
16 </html>
17
```



Pronalaženje stringa u stringu

- metod `search()` traži string u tekstu i vraća pozicije koje su se poklopile
- Dve metode, `indexOf()` i `search()`, su iste?
- Imaju iste argumente (parametre), i vraćaju istu vrednost
- Razlike:
 - `search()` metod ne može da nađe drugu startnu poziciju
 - `search()` metod može da ima bogatiju pretragu



search isto kao indexOf

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>search(...) metoda vraca
5 poziciju prvog pojavljivanja</p>
6 <p id="demo"></p>
7 <script>
8   var tekst = "ABCGHIdefABCdef";
9   var gdeJe = "def";
10  var pos = tekst.search(gdeJe);
11  document.getElementById("demo").innerHTML = "u\
12  \"+tekst+"\" nalazi se \""+gdeJe+
13  "\" na poziciji "+pos;
14 </script>
15 </body>
16 </html>
```



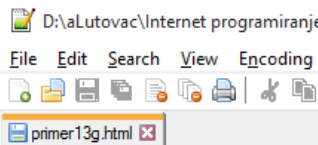
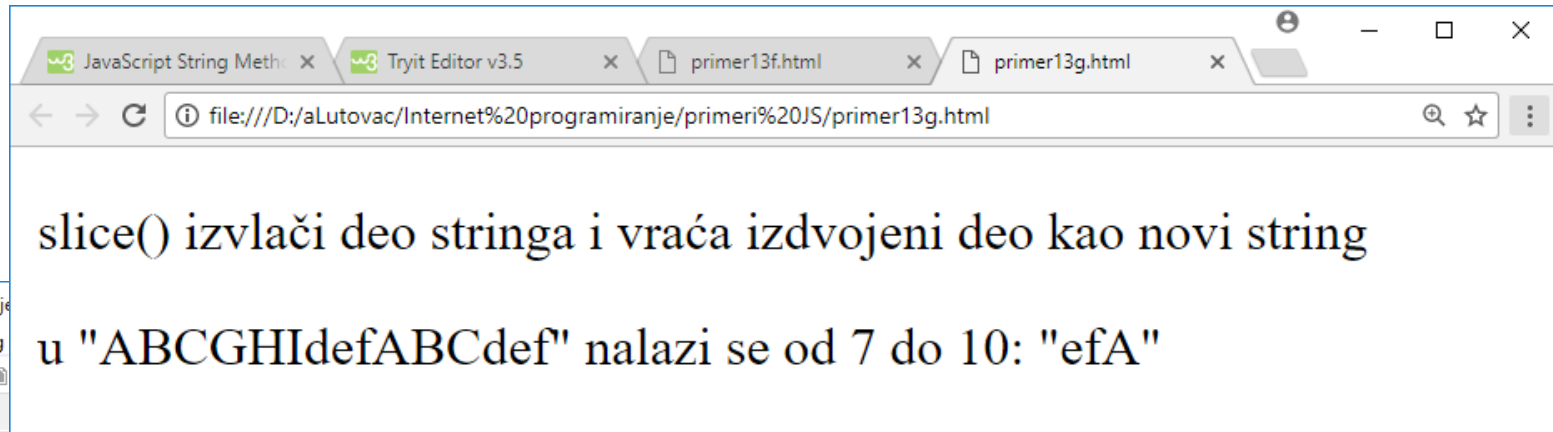
Izdvajanje delova stringa

- `slice(start, end)`
- `substring(start, end)`
- `substr(start, length)`



slice() metod

- slice() izvlači deo stringa i vraća izdvojeni deo kao string
- metod ima 2 parametra
 - startni indeks (pozicija)
 - završni indeks (pozicija)
- Ako su parametri negativni, pozicija se računa od kraja stringa
- Ako se izostavi drugi indeks, podrazumeva se do kraja stringa
- Negativne pozicije ne rade u svim pregledačima



```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>slice() izvlači deo stringa i vraća
5 izdvojeni deo kao novi string
6 </p>
7 <p id="demo"></p>
8 <script>
9   var tekst = "ABCGHIdefABCdef";
10  var pocetakstringa = 7;
11  var krajstringa = 10;
12  var pos = tekst.slice(pocetakstringa,krajstringa);
13  document.getElementById("demo").innerHTML = "u\
14  \"+tekst+"\" nalazi se od \"+pocetakstringa+
15  \" do \"+krajstringa+\": \"+pos+"\"";
16 </script>
17 </body>
18 </html>
```

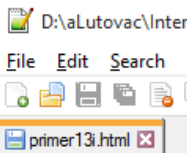
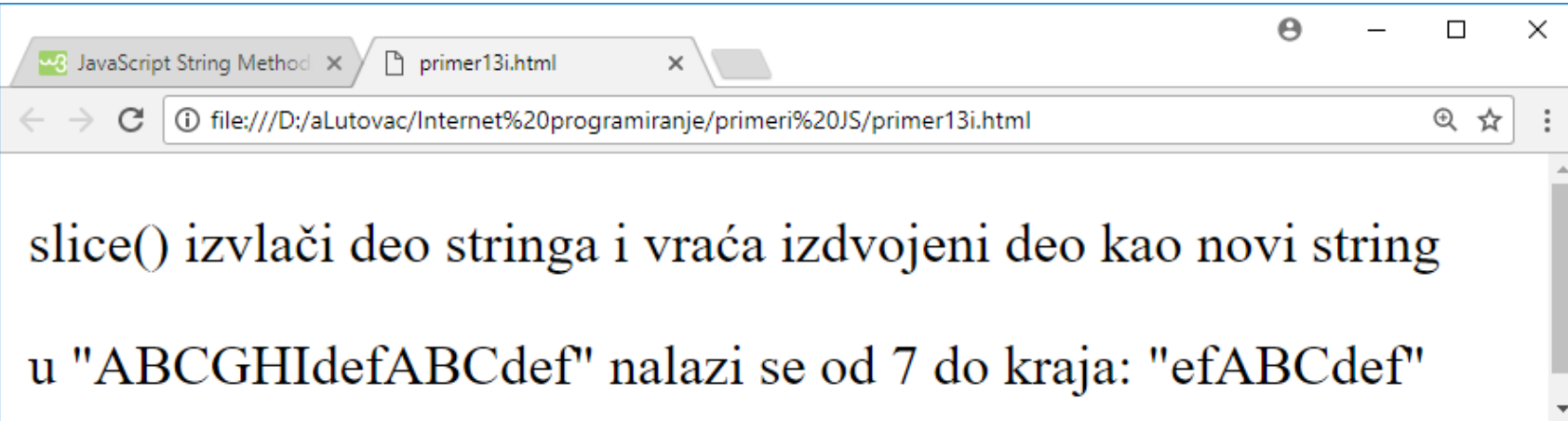
tekst.slice(pocetakstringa,krajstringa)



slice() izvlači deo stringa i vraća izdvojeni deo kao novi string
u "ABCGHIdefABCdef" nalazi se od -10 do -7: "Ide"

tekst.slice(-pocetakstringa,-krajstringa)

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>slice() izvlači deo stringa i vraća
5 izdvojeni deo kao novi string
6 </p>
7 <p id="demo"></p>
8 <script>
9 var tekst = "ABCGHIdefABCdef";
10 var pocetakstringa = -10;
11 var krajstringa = -7;
12 var pos = tekst.slice(pocetakstringa,krajstringa);
13 document.getElementById("demo").innerHTML = "u\
14 \"+tekst+\" nalazi se od \"+pocetakstringa+
15 \" do \"+krajstringa+\": \"+pos+\"\";
16 </script>
17 </body>
18 </html>
```

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>slice() izvlači deo stringa i vraća
5 izdvojeni deo kao novi string
6 </p>
7 <p id="demo"></p>
8 <script>
9 var tekst = "ABCGHIdefABCdef";
10 var pocetakstringa = 7;
11 var pos = tekst.slice(pocetakstringa);
12 document.getElementById("demo").innerHTML = "u\
13 \"+tekst+"\" nalazi se od \"+pocetakstringa+
14 \" do kraja\"+\": \"+pos+"\"";
15 </script>
16 </body>
17 </html>
```

tekst.slice(pocetakstringa)



slice() izvlači deo stringa i vraća izdvojeni deo kao novi string

u "ABCGHIdefABCdef" nalazi se od -10 do kraja: "IdefABCdef"

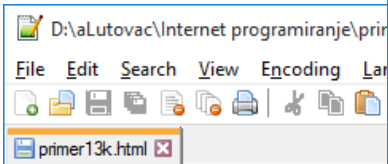
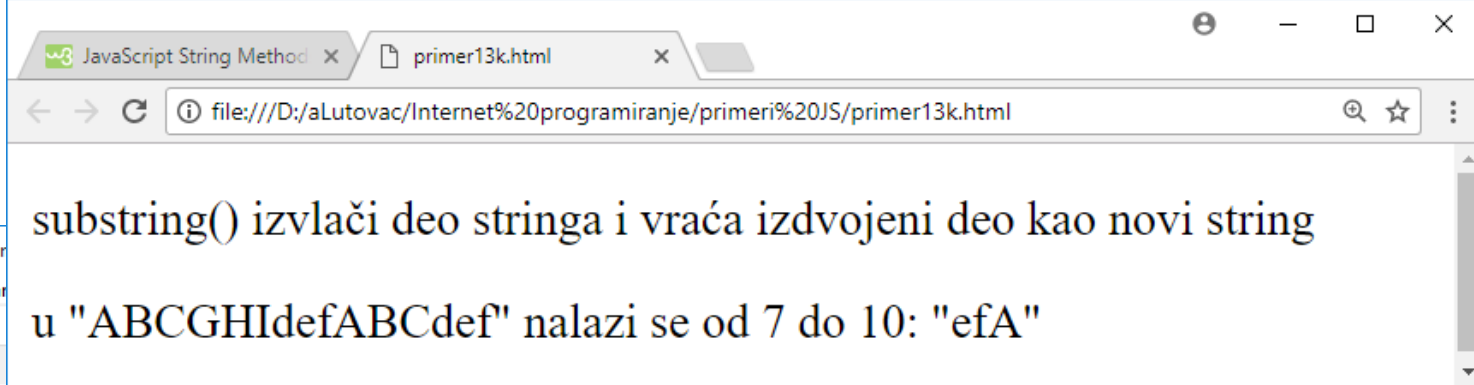
```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>slice() izvlači deo stringa i vraća
5 izdvojeni deo kao novi string
6 </p>
7 <p id="demo"></p>
8 <script>
9 var tekst = "ABCGHIdefABCdef";
10 var pocetakstringa = -10;
11 var pos = tekst.slice(pocetakstringa);
12 document.getElementById("demo").innerHTML = "u\
13 \"+tekst+"\" nalazi se od \"+pocetakstringa+
14 \" do kraja\"+\": \"+pos+"\"";
15 </script>
16 </body>
17 </html>
```

tekst.slice(-pocetakstringa)



substring() metod

- substring() je sličan metodi slice()
- substring() ne može da prihvati negativne indekse
- Ako se izostavi drugi parametar, substring() će preuzeti ostatak stringa



```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>substring() izvlači deo stringa i vraća
5 izdvojeni deo kao novi string
6 </p>
7 <p id="demo"></p>
8 <script>
9 var tekst = "ABCGHIdefABCdef";
10 var pocetakstringa = 7;
11 var krajstringa = 10;
12 var pos = tekst.substring(pocetakstringa, krajstringa);
13 document.getElementById("demo").innerHTML = "u\
14 \"+tekst+"\" nalazi se od \"+pocetakstringa+
15 \" do \"+krajstringa+": \"+pos+"\"";
16 </script>
17 </body>
18 </html>
```

tekst.substring(pocetakstringa, krajstringa)



substring() izvlači deo stringa i vraća izdvojeni deo kao novi string
u "ABCGHId efABCdef" nalazi se od 7 do kraja: "efABCdef"

tekst.substring(pocetakstringa)

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>substring() izvlači deo stringa i vraća
5 izdvojeni deo kao novi string
6 </p>
7 <p id="demo"></p>
8 <script>
9 var tekst = "ABCGHId efABCdef";
10 var pocetakstringa = 7;
11 var pos = tekst.substring(pocetakstringa);
12 document.getElementById("demo").innerHTML = "u\
13 \"+tekst+"\" nalazi se od \"+pocetakstringa+
14 \" do kraja: \"+pos+"\"";
15 </script>
16 </body>
17 </html>
```



substr() metod

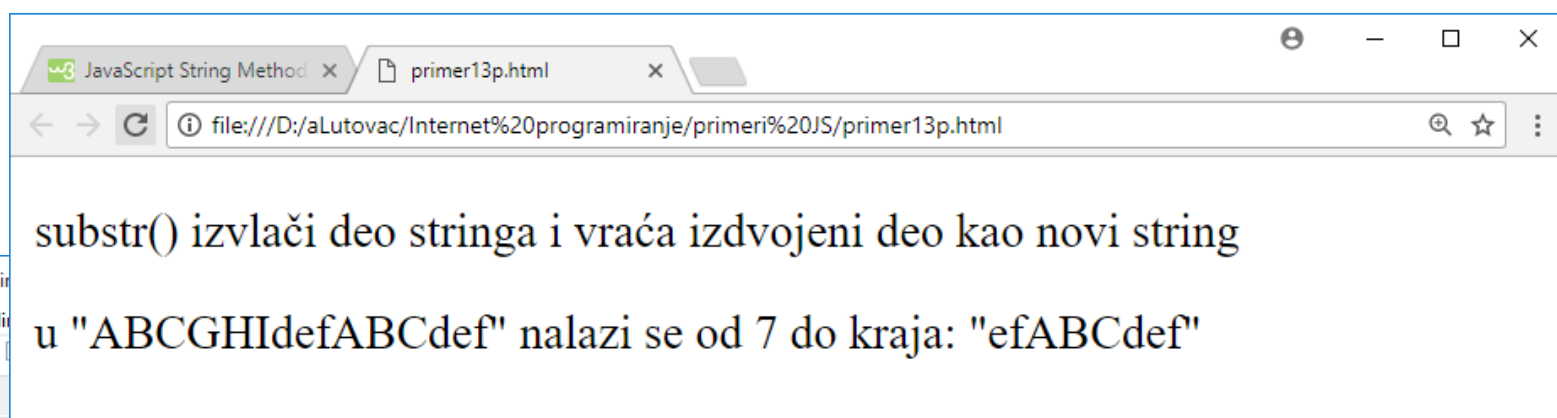
- **substr()** je sličan metodi slice()
- Drugi parametar određuje dužinu izdvojenog stringa
- Ako je prvi parametar negativan, pozicija se računa od kraja stringa
- Drugi parametar ne može da bude negativan, zato što on određuje dužinu
- Ako se izostavi drugi parametar, substr() će izdvojiti ostatak stringa



substr() izvlači deo stringa i vraća izdvojeni deo kao novi string
u "ABCGHId efABCdef" nalazi se od 7 dužine 5: "efABC"

tekst.substr(pocetakStringa,duzinaStringa)

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>substr() izvlači deo stringa i vraća
5 izdvojeni deo kao novi string
6 </p>
7 <p id="demo"></p>
8 <script>
9 var tekst = "ABCGHId efABCdef";
10 var pocetakStringa = 7;
11 var duzinaStringa = 5;
12 var pos = tekst.substr(pocetakStringa,duzinaStringa);
13 document.getElementById("demo").innerHTML = "u\
14 \"+tekst+"\" nalazi se od \"+pocetakStringa+
15 \" dužine \"+duzinaStringa+\": \"+pos+"\"";
16 </script>
17 </body>
18 </html>
```



substr() izvlači deo stringa i vraća izdvojeni deo kao novi string

u "ABCGHId efABCdef" nalazi se od 7 do kraja: "efABCdef"

tekst.substr(pocetakStringa)

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>substr() izvlači deo stringa i vraća
5 izdvojeni deo kao novi string
6 </p>
7 <p id="demo"></p>
8 <script>
9 var tekst = "ABCGHId efABCdef";
10 var pocetakStringa = 7;
11 var pos = tekst.substr(pocetakStringa);
12 document.getElementById("demo").innerHTML = "u\
13 \"+tekst+"\" nalazi se od \"+pocetakStringa+
14 \" do kraja: \"+pos+"\"";
15 </script>
16 </body>
17 </html>
```




Zamena sadržaja stringa

- Metod `replace()` specifikiranu vrednost sa drugom vrednosti stringa
- Metod `replace()` ne menja string na kome se izvršava, on vraća novi string
- Po definiciji, `replace()` funkcija zamenjuje samo prvo poklapanje
- Da bi se zamenila sva poklapanje, koristi se regularni izraz sa `/g` flag-om (globalno poklapanje)
- `replace()` je case sensitive
- Da bi se zamenila sa `/i` flag-om

Zameni "def" sa " e3Fg" i sledećem paragrafu:

zameni

Poseti ABCGHI**def**ABCdef

Zameni "def" sa

zameni

Poseti ABCGHI **e3Fg**ABCdef

Zameni "def" sa " e3Fg" i sledećem paragrafu:

zameni

Poseti ABCGHI **e3Fg**ABCe3Fg

```
1 <!DOCTYPE html>
```

```
2 <html>
```

```
3 <body>
```

```
4 <p>Zameni "def" sa " e3Fg" i sledećem paragrafu:</p>
```

```
5 <button onclick="myFunction()">zameni</button>
```

```
6 <p id="demo">Poseti ABCGHI<b>def</b>ABC<u>def</u></p>
```

```
7 <script>
```

```
8 function myFunction() {
```

```
9     var original = document.getElementById("demo").innerHTML;
```

```
10    var zamenjeno = original.replace("def", " e3Fg");
```

```
11    document.getElementById("demo").innerHTML = zamenjeno;
```

```
12 }
```

```
13 </script>
```

```
14 </body>
```

```
15 </html>
```

replace()

JavaScript String Method x primer14a.html

file:///D:/aLutovac/Internet%20programiranje/

Zameni "def" sa " e3Fg" i sled

zameni

Poseti ABCGHI**def**ABCdef

JavaScript String Method x primer14a.html

file:///D:/aLutovac/Internet%20programiranje/primeri%20JS/primer14a....

Zameni "def" sa " e3Fg" i sledećem paragrafu:

zameni

Poseti ABCGHI **e3Fg**ABC e3Fg

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>Zameni "def" sa " e3Fg" i sledećem paragrafu:</p>
5 <button onclick="myFunction()">zameni</button>
6 <p id="demo">Poseti ABCGHI<b>def</b>ABC<u>def</u></p>
7 <script>
8 function myFunction() {
9     var original = document.getElementById("demo").innerHTML;
10    var zamenjeno = original.replace(/def/g, " e3Fg");
11    document.getElementById("demo").innerHTML = zamenjeno;
12 }
13 </script>
14 </body>
15 </html>
16
```

/def/g

VISER 2018

Hyper Text Markup Language file length : 437 lines : 16 Ln: 16 Col: 1 Sel: 0 | 0 Windows (CR LF) UTF-8 INS

Zameni "DEF" sa " e3Fg" i sle

zameni

Poseti ABCGHI**def**ABCDEF

Zameni "DEF" sa " e3Fg" i sledećem paragrafu:

zameni

Poseti ABCGHI**def**ABCe3Fg

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>Zameni "DEF" sa " e3Fg" i sledećem paragrafu:</p>
5 <button onclick="myFunction()">zameni</button>
6 <p id="demo">Poseti ABCGHI<b>def</b>ABC<u>DEF</u></p>
7 <script>
8 function myFunction() {
9     var original = document.getElementById("demo").innerHTML;
10    var zamenjeno = original.replace("DEF", " e3Fg");
11    document.getElementById("demo").innerHTML = zamenjeno;
12 }
13 </script>
14 </body>
15 </html>
16
```

case sensitive



```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>Zameni "DEF" sa " e3Fg" i sledećem paragrafu:</p>
5 <button onclick="myFunction()">zameni</button>
6 <p id="demo">Poseti ABCGHI<b>def</b>ABC<u>DEF</u></p>
7 <script>
8 function myFunction() {
9     var original = document.getElementById("demo").innerHTML;
10    var zamenjeno = original.replace(/DEF/i, " e3Fg");
11    document.getElementById("demo").innerHTML = zamenjeno;
12 }
13 </script>
14 </body>
15 </html>
16
```

case insensitive, /i flag



Konverzija velikih i malih slova

- String se konvertuje u velika slova sa `toUpperCase()`
- String se konvertuje u mala slova sa `toLowerCase()`

```
var text1 = „Marko Markov”;    // String
```

```
var text2 = text1.toUpperCase(); // text2 is text1 converted to upper
```

```
var text3 = text2.toLowerCase(); // text3 is text2 converted to lower
```



concat() metod

- concat() spaja dva ili više stringova
- concat() metod može da se koristi umesto plus operatora

```
var text = "Marko" + " " + "Markov";
```

```
var text = "Marko".concat(" ", "Markov");
```

- Sve string metode vraćaju novi string
- Ove metode ne modifikuju originalni string
- Stringovi su nepromenljivi
- Stringovi ne mogu da se menjaju, mogu samo da se zamene



Izdvajanje karatera u stringu

- Postoje dva sigurna metoda za ekstrakciju karaktera u stringovima

`charAt(position)`

`charCodeAt(position)`



charAt() i charCodeAt() metode

- **charAt()** metoda vraća karakter na specificiranoj poziciji (indeksu) u stringu
- **charCodeAt()** metoda vraća unicode karaktera na specificiranom indeksu stringa

charAt() metod vraća karakter na određenoj poziciji

G

charAt()

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>charAt() metod vraća karakter na određenoj poziciji</p>
5
6 <p id="demo"></p>
7
8 <script>
9   var str = "ABCGHIdEfABCdef";
10  document.getElementById("demo").innerHTML = str.charAt(3);
11 </script>
12 </body>
13 </html>
```



charCodeAt

charCodeAt() metod vraća unicode karaktera na određenoj poziciji

71

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>charCodeAt() metod vraća unicode karaktera na određenoj poziciji</p>
5 <p id="demo"></p>
6 <script>
7   var str = "ABCGHIdefABCdef";
8   document.getElementById("demo").innerHTML = str.charCodeAt(3);
9 </script>
10 </body>
11 </html>
```



Pristup stringu kao nizu nije siguran

- Možete da vidite kod u kome se pristupa stringu kao da je niz

```
var str = "Marko Markov";  
str[0];           // returns M
```

- Ovo nije sigurno i rezultat može biti nepredvidiv:
 - Ne funkcioniše u svim pregledačima
 - Iako liči da je string niz karaktera, to nije tačno
 - `str[0] = "H"` ne generiše grešku, ali i ne radi
- Ako se želi pristup stringu kao da je niz, treba konvertovati string u niz



Konvertovanje stringa u niz

- String može da se konvertuje u niz korišćenjem metode `split()`
- Ako se izostavi separator, vraćeni niz će sadržati ceo string u indeksu `[0]`.
- Ako je separator `" "`, vraćeni niz će biti niz jednoslovnih karaktera

```
var txt = "a,b,c,d,e"; // String  
txt.split(",");        // Split on commas  
txt.split(" ");        // Split on spaces  
txt.split("|");        // Split on pipe
```

Klikni "Try it" da se prikaze prvi niz elementa, nakon što je string pretvoren u

Izaberi prvi

Klikni "Try it" da se prikaze prvi niz elementa, nakon što je string pretvoren u niz.

Izaberi prvi

G

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p>Klikni "Try it" da se prikaze prvi niz elementa,
5 nakon što je string pretvoren u niz.</p>
6 <button onclick="myFunction()">Izaberi prvi</button>
7 <p id="demo"></p>
8 <script>
9 function myFunction() {
10     var str = "A,B,C,G,H,I,d,e,f,A,B,C,d,e,f";
11     var arr = str.split(",");
12     document.getElementById("demo").innerHTML = arr[3];
13 }
14 </script>
15 </body>
16 </html>
```

VISER 2018

70



```
D:\aLutovac\Internet programiranje\primeri JS\primer14h.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
primer14h.html x
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p id="demo"></p>
5 <script>
6 var str = "ABCGHIdEfA";
7 var arr = str.split("");
8 var text = "";
9 var i;
10 for (i = 0; i < arr.length; i++) {
11     text += arr[i] + "<br>"
12 }
13 document.getElementById("demo").innerHTML = text;
14 </script>
15 </body>
16 </html>
```

Hyper Text Markup Language file length : 285 lines : 17 Ln: 17 Col: 1 Sel: 0 | 0 Windows (CR LF) UTF-8 INS

