JAVASCRIPT

DEO 2

DATE OBJEKAT

- dateObjectIme = new Date()
- dateObjectIme = new Date("month day, year hours:minutes:seconds")
- dateObjectIme = new Date(year, month, day)
- dateObjectIme = new Date(year, month, day, hours, minutes, seconds)

DATE OBJEKAT

- today = new Date()
- birthday = new Date("December 17, 2015 03:24:00")
- birthday = new Date(2015,11,17,3,24,0)
- birthday = new Date(2015,11,17,3)
- birthday = new Date(2015,11,17)
- birthday = new Date(99,11,17)

DATE OBJEKAT

- var d = new Date();
  Wed Mar 25 2020 15:28:35 GMT+0100 (Central European Standard Time)

- var d = new Date("October 13, 2014 11:13:00");
  Mon Oct 13 2014 11:13:00 GMT+0200 (Central European Summer Time)

DATE OBJEKAT

- Date.parse(datum)

  Ovaj metod vraća broj milisekundi do navedenog datuma po lokalnom vremenu (od 1.1.1970 00:00:00)

  datum.setTime(Date.parse("Aug 9, 2005")

- Date.UTC(gg,mm,dd [,hh][,mh][,sec])

  Ovaj metod vraća broj milisekundi od 1.1.1970 00:00:00 do datuma, prema Universal Coordinate Time (GMT).

  gmtDatum = new Date(Date.UTC(96, 11, 1, 0, 0, 0))

DATE OBJEKAT

- datum.getDate() - vraća dan u mesecu (1-31) za navedeni datum

```
datum = new Date("December 25, 2001 23:15:00");
dan = datum.getDate()
```

- datum.getDay() - vraća dan u nedelji (0-ned, 1-pon ... 6-sub) za navedeni datum

```
datum = new Date("December 25, 2001 23:15:00");
dan = datum.getDay()
```

DATE OBJEKAT

- datum.getHours() - vraća sat za navedeni datum, moguće vrednosti su brojevi u opsegu od 0 do 23

```
datum = new Date("December 25, 2001 23:15:00");
sati = datum.getHours()
```

- datum.getMinutes() - vraća minute za navedeni datum, moguće vrednosti su brojevi u opsegu od 0 do 59

```
datum = new Date("December 25, 2001 23:15:00");
minuti = datum.getMinutes()
```

DATE OBJEKAT

- datum.getMonth() - vraća mesec za navedeni datum (0-januar, 2-februar, ... 11-decembar)

```
datum = new Date("December 25, 2001 23:15:00");
mesec = datum.getMonth()
```

- datum.getSeconds() - vraća sekunde za navedeni datum, moguće vrednosti su brojevi u opsegu od 0 do 59

```
datum = new Date("December 25, 2001 23:15:08");
sekunde = datum.getSeconds()
```

- datum.getYear() - vraća godinu iz navedenog datuma (dvocifreno, od 1900 do 1999, u ostalim slučajevima 4 cifre)

```
datum = new Date();
godina = datum.getYear()
```

DATE OBJEKAT

- datum.setDate(brojDana) - postavlja dan u mesecu za navedeni datum. Argument metoda je broj u opsegu od 1 do 31. Primer:

  ```
  datum = new Date("July 27, 1960 23:30:00");
  datum.setDate(24)
  ```

- datum.setHours(brojSata) - postavlja broj sati za navedeni datum. Argument metoda je broj u opsegu od 0 do 23. Primer

  ```
  datum = new Date("July 27, 1960 23:30:00");
  datum.setHours(7)
  ```

- datum.setMinutes(brojMinuta) - postavlja broj minuta za navedeni datum. Argument metoda je broj u opsegu od 0 do 59

  ```
  datum = new Date("July 27, 1960 23:30:00");
  datum.setMinutes(35)
  ```

DATE OBJEKAT

- datum.setHours(brojSata) - postavlja broj sati za navedeni datum. Argument metoda je broj u opsegu od 0 do 23. Primer

```
datum = new Date("July 27, 1960 23:30:00");
datum.setHours(7)
```

- datum.setMinutes(brojMinuta) - postavlja broj minuta za navedeni datum. Argument metoda je broj u opsegu od 0 do 59. Primer:

```
datum = new Date("July 27, 1960 23:30:00");
datum.setMinutes(35)
```

DATE OBJEKAT

- datum.setMonth(brojMeseca) - postavlja broj meseci za navedeni datum. Argument metoda je broj u opsegu od 0 do 11

  datum = new Date("July 27, 1960 23:30:00");

  datum.setMonth(11)

- datum.setSeconds(brojSekundi) - postavlja sekunde za navedeni datum. Argument metoda je broj u opsegu od 0 do 59

  datum = new Date("July 27, 1960 23:30:00");

  datum.setSeconds(35)

DATE OBJEKAT

- datum.setTime(vreme) - definiše novi datum. Argument je broj milisecundi od 1.1.1970 00:00:00 do željenog datuma.

  datum.setTime(1332403882588)

  Thu Mar 22 2012 09:11:22 GMT+0100 (Central European Standard Time)

- datum.setYear(brojGodine) - postavlja godinu za navedeni datum. Argument je broj u opsegu od 0 do 99 za godine koje počinju sa 19, za ostale je 4 cifre.

  datum = new Date("July 27, 1960 23:30:00");

  datum.setYear(2010)

DATE OBJEKAT

- datum.toGMTString() - vrši konverziju datuma u GMT string iz lokalne vremenske zone

  datum = new Date("December 25, 2001 23:15:00");

  datum.toGMTString()

- datum.toLocaleString() - vrši konverziju datuma u lokalni datum string iz GMT

  datum.toLocaleString()

DATE OBJEKAT

```
<script language="JavaScript">
today = new Date();
summer= new Date();
summer.setMonth(5);
summer.setDate(22);
summer.setYear(2020);
if (today.getTime() < summer.getTime()) {
    difference = summer.getTime() - today.getTime();
    difference = Math.floor(difference / (1000 * 60 * 60 * 24));
    document.write("samo '" + difference + ' dana do leta!<p>');
}
</script>
```

STRING OBJEKAT

- escape("string") - kao rezultat vraća ASCII kôdove karaktera u okviru argumenta
  y = escape("!#")

  Nakon izvršavanja primera promenljiva y dobija vrednost „%21%23“, jer su ASCII kôdovi za simbole ! i # 21 i 23.

- x=unescape("%21%23") - kao rezultat vraća ASCII znakove navedenih kodova u okviru argumenta funkcije

  Nakon izvršavanja primera promenljiva x dobija vrednost /"!#"/, jer su simboli ! i # kodovani sa ASCIIkodovima 21 i 23.

- eval(izraz) - izračunava vrednost izraza koji je definisan kao argument funkcije
  var x = eval("4+5-8")

  Nakon izvršavanja primera promenljiva x dobija vrednost 1.

STRING OBJEKAT

```
<script>
function myFunction() {
  var x = 10;
  var y = 20;
  var a = eval("x * y") + "<br>";
  var b = eval("2 + 2") + "<br>";
  var c = eval("x + 17") + "<br>";
  var res = a + b + c;
  document.getElementById("demo").innerHTML = res;
}
</script>
```

200
4
27

STRING OBJEKAT

- linkTekst.link(linkURL) - kreira tekst linkTekst koji predstavlja HTML link na neku drugu stranicu, čiji je adrsa definisana sa argumentom linkURL

  (desjtvo kao i HTML taga <A HREF...>)

```
var naziv ="ETF sajt";
var URL = "http://www.etf.rs";
document.write("Ovo je " + naziv.link(URL))
```

STRING OBJEKAT

- parseInt(stringBroj [,*osnova*]) - kao rezultat vraća ceo broj dobijen konverzijom argumenta stringBroj koji je tipa String u brojnom sistemu sa osnovom koju definiše argument osnova

```
x = parseInt("17", 8);
y = parseInt("15", 10);
k = parseInt("10", 16);

z = parseInt("40 50 60") + "<br>";
f = parseInt("40 godina") + "<br>";
g = parseInt("On ima 40 godina") + "<br>";
```

STRING OBJEKAT

- string.big() - prikazuje string sa uvećanim slovima
  "Dobar dan!".big()

- string.bold() - prikazuje string boldovano
  "Dobar dan!".bold()

- string. italics() - prikazuje string kurziv stilom
  "Dobar dan!".italics();

- string.fontcolor() - prikazuje stringu određenoj boji
  "Dobar dan!".fontcolor("blue");

- string.fontsize() - prikazuje stringu određenoj veličini
  "Dobar dan!".fontsize(7);

STRING OBJEKAT

- string.sub() - prikazuje string kao subscript (isto dejstvo kao HTML tag <SUB>)

  "Hej!".sub()

- string.sup() - prikazuje string kao superscript (isto dejstvo kao HTML tag <SUP>)

  "Hej!".sub()

- string.toLowerCase() - vrši konverziju svih karaktera u okviru stringa u mala slova

  x = "Dobar dan!".toLowerCase()

- string.toUpperCase() - vrši konverziju svih karaktera u okviru stringa u velika slova

  x = "Dobar dan!".toUpperCase()

STRING OBJEKAT

- string.charAt(broj)
- Ovaj metod kao rezultat vraća znak na navedenoj poziciji. Pozicije unutar stringa se računaju sa leve na desnu stranu i prva pozicija ima indeks 0. U okviru svakog objekta tipa String postoji i osobina (property) length koja je jednaka broju karaktera u posmatranom stringu. Korišćenjem ovog podatka može se odrediti i indeks poslednjeg karakera u stringu, a to je vrednost string.length.

```
x= "Dobar dan!".charAt(4)    r
y= "Dobar dan!".charAt(6)    d
```

STRING OBJEKAT

- **string.indexOf(traziString, [odPozicije])**
- Ovaj metod vraća broj pozicije na kojoj je prvi put pronađen argument tipa String traziString. U slučaju da se trazeni string ne nalazi u početnom stringu kao rezultat se vraća vrednost -1.
- Ako postoji i drugi argument odPozicije, tada će se pretraga izvršavati od zadate pozicije.

```
x ="Dobar dan!".indexOf("r")
y ="Dobar dan!".indexOf("a",4)
```

- Nakon izvršavanja primera promenljiva x dobija vrednost 4, a promenljiva y je 7.

STRING OBJEKAT

- string.lastIndexOf(traziString, [doPozicije])
- Vraća broj pozicije na kojoj se poslednji put pojavljuje argument tipa String traziString. U slučaju da se trazeni string ne nalazi u početnom stringu kao rezultat se vraća vrednost -1.
- Ako postoji i drugi argument doPozicije, tada će se pretraga izvršavati do zadate pozicije.

```
x = "Dobar dan!".lastIndexOf("a")      7
y = "Dobar dan!".lastIndexOf("a", 6)    3
```

STRING OBJEKAT

- string.substring(prvi, poslednji)
- Ovaj metod vraća deo stringa počev od pozicije prvi do pozicije poslednji, t.j. uzima redom karaktere na pozicijama prvi, prvi + 1, prvi + 2, ..., poslednji -2, poslednji – 1.

  x = "Dobar dan!".substring(6,9)

- Nakon izvršavanja primera promenljiva x dobija vrednost "dan", jer su to karakteri na pozicijama 6, 7 i 8.

STRING OBJEKAT

- string.substr(prvi, duzina)
- Ovaj metod vraća deo stringa počev od pozicije prvi do pozicije poslednji, t.j. uzima redom karaktere na pozicijama prvi, prvi + 1, prvi + 2, ..., poslednji -2, poslednji – 1.

  x = "Dobar dan!".substr(6,2)

- Nakon izvršavanja primera promenljiva x dobija vrednost “da”

FUNKCIJE

```
function calculate(a,b,c) {
    d = (a+b) * c;
    return d;
}

var x = calculate(4,5,9);
document.write("x je ", x, "<br>")
var y = calculate((x/3),3,5)
document.write("y je ", y)
```

OPSEG VAŽENJA PROMENLJIVE

- Opseg važenja promenljive:
  - globalna ili lokalna

- Globalne promenljive:
  - Deklarišu se izvan funkcije
  - Može im se pristupiti iz bilo kog dela programa

- Lokalne promenljive:
  - Deklarišu se u okviru funkcije
  - Jedino su dostupne u okviru funkcije u kojoj su deklarisane

OPSEG VAŽENJA PROMENLJIVE

- Lokalne promenljive prestaju da postoje u trenutku kada se funkcija završi
- Korišćenje lokalnih promenljivih izvan funkcije dovodi do greške
- Parametri definisani u okviru funkcije su takođe lokalne promenljive

RAD SA UZORCIMA

- Česta upotreba JavaScript funkcija je provera unetih podataka od strane klijenta
- JavaScript ima razvijenu podršku za razne vrste provera i one se obavljaju na klijentskoj strani
- Zadaci se rešavaju upotrebom uzoraka (pattern) i pozivanjem određenih metoda koji uneti tekst upoređuju sa definisanim uzorkom
- Uzorak se još naziva i regularni izraz (regular expression

RAD SA UZORCIMA

- var uPrimer = new RegExp(„HTML”)
- var uPrimer = /HTML/
- Na oba načina se formira objekat uzorka koji se naziva uPrimer i kome odgovara svaki string koji u sebi sadrži podstring HTML

- var uPrimer = new RegExp(„s$”)
- var uPrimer = /s$/

- simbol $ predstavlja metakarakter koji označava kraj stringa
- Promenljiva uzorak odgovara bilo kom stringu koji se zavrsava sa s

RAD SA UZORCIMA

Karakter	Predstavlja
Slovo ili broj	Istu vrednost
\0	Specijalni NUL karakter
\t	Tab znak
\n	Nova linija
\v	Vertikalni tab znak
\f	Form feed
\r	Carriage return
\uxxxx	Unicode karakter definisan pomoću heksadecimalnog boja xxxx; na primer, \u0009 ima isti efekat kao i \t

RAD SA UZORCIMA

Karakter	Predstavlja pojavljivanje
[...]	Bilo kog karaktera od onih koji su navedeni između [i].
[^...]	Bilo kog karaktera koji nije naveden između [i].
.	Bilo kog karaktera osim nove linije
\w	Bilo kog ASCII definisanog slova.
\W	Bilo kog karaktera koji nije ASCII definisano slovo.
\d	Bilo koje ASCII definisane cifre
\D	Bilo kog karaktera koji nije ASCII definisana cifra
[\b]	Blanko znak
\s	Prazan prostor

RAD SA UZORCIMA

- /[abc]/
- predstavlja jedno pojavljivanje simbola a ili jedno pojavljivanje simbola b ili jedno pojavljivanje simbola c.

- /[^abc]/
- predstavlja karakter koji nije simbol a ili b c.

- /\d\d\d\d\d/
- Pomoću ovog uzorka se definiše broj koji se sastoji od 5 cifara

RAD SA UZORCIMA

Oznaka	Značenje
{n,m}	Ponavljanje prethodne grupe najmanje n puta, ali najviše m puta.
{n,}	Ponavljanje prethodne grupe n ili više puta.
{n}	Ponavljanje prethodne grupe tačno n puta.
?	Ponavljanje prethodne grupe jednom ili nijednom. Isto dejstvo kao i {0,1}.
+	Ponavljanje prethodne grupe jednom ili više puta. Isto dejstvo kao i {1,}.
*	Ponavljanje prethodne grupe nijednom ili više puta. Isto dejstvo kao i {0,}.
|	Alternative. Pojavljivanje dela izraza sa desne ili pojavljivanje izraza sa leve strane.
(...)	Grupisanje simbola u jedan objekat nad kojim se mogu koristiti oznake *, +, ?, |, itd.
^	Pretraga uzorka se obavlja na početku stringa
$	Pretraga uzorka se obavlja na kraju stringa

RAD SA UZORCIMA

- /\d{2,4}/
- /\w{3}\d?/
- /\s+java\s+/
- /[^"]*/
- /ab|cd|ef/
- /\d{3}|[a-z]{4}/
- /java(script)?/
- /(ab|cd)+|ef)/

RAD SA UZORCIMA

Atribut	Značenje
i	Izvršavanje case-insensitive ispitivanja.
g	Izvršava globalno ispitivanje, pronaći će se sva pojavljivanja definisanog uzorka
M	Rad sa više linija. ^ označava početak linije ili stringa, a $ predstavlja kraj linije ili stringa.

RAD SA UZORCIMA

- search( ) - Ovaj metod ispituje da li u okviru stringa postoji defnisani uzorak, i kao rezultat vraća poziciju njegovog prvog pojavljivanja

```
x = /script/
y = "JavaScript".search(x,i);
```

- replace() - Ovaj metod obavlja ispitivanje da li u stringu postoji uzorak i ako postoji vrši zamenu uzorka unutar stringa sa nekom drugom vrednošću

```
text = "JAVascriPT"
text.replace(/javascript/gi, "JavaScript");
```

RAD SA UZORCIMA

- match( )
- Rezultat izvršavanja je niz koji sadrži rezultate ispitivanja.
- Ako je u okviru uzorka definisan atribut g, rezultat je niz sa svim pojavljivanjima definisanog uzorka.

  "1 plus 2 je 3".match(/\d+/g)

- Rezultat je niz ["1", "2", "3"]

RAD SA UZORCIMA

- exec( ) – vraća tekst ako ga je pronašao, u suprotnom vraća null .
- Za razliku od match() metoda exec() vraća isti rezultat ako postoji atribut g i ako ne postoji.

```
var str = “Nema vremena!“
var patt = new RegExp(“e”);
var res = patt.exec(str);
```

- Rezultat je “e”.

```
var str = "1 plus 2 je 3!"
var patt = /\d+/g;
var res = patt.exec(str);
```

- Rezultat je “1”.

RAD SA UZORCIMA

- test( ) argument je string, a rezultat true ako string odgovra uzorku

```
var pattern = /java/i;
pattern.test("JavaScript");
```

- Rezultat je true

RAD SA UZORCIMA

- lastIndex – property RegExp objekta. Kod metode za rad sa regularnim izrazom sa indikatorom g u svojstvo lastIndex objekta klase Regex upisuje se pozicija prvog znaka posle odgovarajućeg podniza.

- Kada se metoda ponovo pozove za isti regularni izraz, počinje pretraživanje od pozicije zadate vrednošću svojstva lastIndex.

- Ovo ponašanje omogućava da ponovljene pozive metode izvršavamo kroz petlju, kako bi se pristupilo svim podnizovima u znakovnom nizu podudarnim sa regularnim izrazom.

RAD SA UZORCIMA

```
var str = "JavaScript je zabavniji nego Java";
var patt1 = /Java/g;
while (patt1.test(str)==true)
  {
  document.write("'Java' nadjena. Indeks je sada: "+patt1.lastIndex);
  document.write("<br>");
  }
```

- 'Java' nadjena. Indeks je sada: 4
  'Java' nadjena. Indeks je sada: 33

RAD SA FORMAMA

document.imeForme.imeElementa.value

- document službena reč, imeForme je ime forme u okviru koje se nalazi element čijoj se vrednosti pristupa, imeElementa je ime elementa (name ili id) i value je službena reč za vrednost tog elementa.

```
<form name="PrimerForme">
<input type="text" name="PrimerTekstPolja">
</form>
```

- x = document.PrimerForme.PrimerTekstPolja.value
- document.PrimerForme.PrimerTekstPolja.value = x

RAD SA DOGAĐAJIMA

Događaj	Nastaje kada korisnik...	Kod
blur	izađe iz fokusa elementa forme	onBlur
click	klikne na element forme ili link	onClick
change	promeni vrednost izabranog elementa forme	onChange
focus	uđe u fokus nekog elementa forme	onFocus
load	učita stranicu u browser	onLoad
mouseover	pređe pokazivačem miša preko linka i sl.	onMouseOver
mouseout	izađe kurzorom miša sa određene površine ili linka	onMouseOut
select	izabere polje elementa forme	onSelect
submit	izvrši slanje forme	onSubmit
unload	napusti stranicu	onUnload
reset	resetuje sadržaj forme	onReset
error	dobije grešku prilikom učitavanja slike ili stranice	onError
abort	prekine učitavanja slike ili stranice	onAbort

RAD SA DOGAĐAJIMA

- Registruje se prelazak miša preko linka.

```
<a href="#" onMouseOver="alert('Event: onMouseOver');">
  onMouseOver </a><p>
```

- Registruje se odlazak miša sa linka.

```
<a href="#" onMouseOut="alert('Event: onMouseOut');"> onMouseOut
  </a><p>
```

- Kada se miš pozicionira na dugme i pomeri poziva se funkcija koja broji koliko puta se desio ovakav događaj.

```
<form>
    <input type="button"  value="onMouseMove"
  onMouseMove="track_Moves();">
 </form>
```

RAD SA DOGAĐAJIMA

```
<form method="post" name="mojaforma">
Upišite prvi broj (0-10)...:   <input type="text" name="broj1" size=5>
    <br>
Upišite drugi broj (0-10)..:   <input type="text" name="broj2" size=5>
    <br>
<input type="button" value="saberi" name="dugme" onClick="Saberi()">
    <br><br>
Zbir dva broja iznosi..:   < input type="text" name="zbir" size=5>
    <br>
 Tekstualni podatak o rezultatu:     < input type="text"
    name="rezultat" size=30>
</form>
```

RAD SA DOGAĐAJIMA

```
<script language="JavaScript">
function Saberi() {
    var br1 = document.mojaforma.broj1.value - 0;
    var br2 = document.mojaforma.broj2.value - 0;
var ukupno = br1 + br2;
var poruka = “”;
if (ukupno <= 0)
        poruka = "nula ili negativan!";
    else if (ukupno > 10)
        poruka = "veći od deset!";
    else
        poruka = "između 1 i 10!";
     document.mojaforma.zbir.value = ukupno;
    document.mojaforma.tekst.value = "Zbir je " + poruka;
}
</script>
```

RAD SA DOGAĐAJIMA

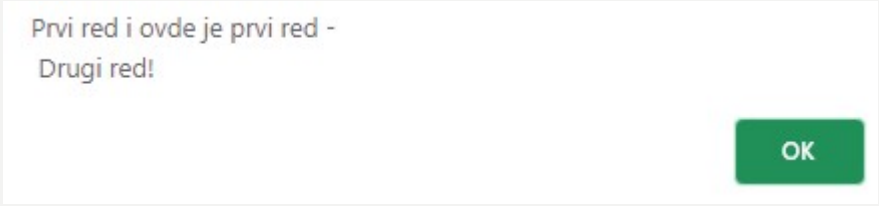
```
<html>
   <head><title>onChange Obrada dogadjaja</title>
   </head>
   <body>
  <form>
       Unesite procenat:
      <input type="text" onChange="grade=parseInt(this.value);
   if(grade < 0 || grade > 100){
              alert('Molimo unesite broj izmedju 0 i 100');
           }
           else{
              confirm('Da li je '+ grade + ' korektno?');            }
      " >
  </form>
   </body>
   </html>
```

RAD SA VIŠE PROZORA

- Alerti (novi manji prozor sa porukom) se koriste unutar HTML stranice kada se želi prikazati određeno obaveštenje

```
<form action="">
<input type="button" value="Pritisni me" onClick="alert()" />
</form>
<script type="javascript">

function alert()
{
alert ("Prvi red "+"i ovde je prvi red - \n Drugi red!");
}
</script>
```

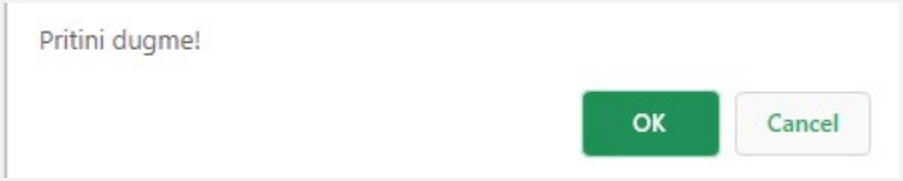


RAD SA VIŠE PROZORA

- Confirm box (novi manji prozor za potvrdu) se koristi unutar HTML stranice kada se želi prikazati određeno obaveštenje koje korisnik treba da prihvati ili odbije.

```
<button onclick="myFunction()"> Probaj </button>


<script>
function myFunction() {
  confirm(“Pritisni dugme");
}
</script>
```



RAD SA VIŠE PROZORA

- window.open() - otvara novi prozor pretraživača
- deteprozor =window.open(“url”,“ime",“lista_parametara");
- deteprozor - je promenljiva. Koristeći ovu promenljivu možemo pozivati funkcije ili pristupati elementima ovog prozora
- url - url novog prozora. Ako je prazan ništa neće biti učitano.
- Ime - je ime prozora koje se koristi pri pozivu funkcija
- lista_parametara je opciona. Čuva listu parametara odvojenih zarezima.

RAD SA VIŠE PROZORA

```
<html>
<body>
<p>Click the button to open a new browser window.</p>
<button onclick="myFunction()">Try it</button>
<script>
function myFunction() {
    window.open("https://www.etf.rs");
}
</script>
</body>
</html>
```

RAD SA VIŠE PROZORA

- window.open("https://www.etf.rs", "_blank", "toolbar=yes, scrollbars=yes, resizable=yes, top=500, left=500,width=400, height=400")

- Lista parametara - uljučuje width i height (visina i dužina prozora), top i left (pozicija prozora), parametre location, menubar, resizable, scrollbars, status, titlebar, toolbar koji mogu biti yes(1) ili no(0), u zavisnosti od toga da li će se pojaviti u novotrvorenom prozoru ili ne.

RAD SA VIŠE PROZORA

```
<script language="javascript">
function prozor( page, width, height, top, left ) {
var yes= 1;
var no= 0;
var menubar      = no;
var scrollbars   = no;
var locationbar = no;
var directories = no;
var resizable    = no;
var statusbar    = no;
var toolbar      = yes;
features = "" +
"width=" + width + "," +
"height=" + height + "," +
"top=" + top + "," +
"left=" + left + "";
```

RAD SA VIŠE PROZORA

```
features += "" +
(menubar ? ",menubars" : "") +
(scrollbars ? ",scrollbars" : "") +
(locationbar ? ",location" : "") +
(directories ? ",directories" : "") +
(resizable ? ",resizable" : "") +
(statusbar ? ",status" : "") +
(toolbar ? ",toolbar" : "");
var prozor = window.open( page, 'fullPopup', features );
}
</script>
```

RAD SA VIŠE PROZORA

- window.opener – referenca na roditeljski prozor
- window.opener.close() – zatvara roditeljski prozor

```
deteProz.deteForma.deteObjekat.value
deteProz = open("noviProzor.html","deteProz")
window.opener.document.otacForma.otacObjekat.value
```

COOKIE

- cookie - segment podataka koji web čitač pamti i koji je povezan sa određenom web stranom ili web lokacijom.
- Koristi se da bi se podaci uneti na jednoj strani koristili na drugoj, tj. da bi čitač mogao da ponovi korisničke parametre ili druge promenljive stanja kada korisnik napusti stranu i kasnije se vrati.
- Format koji cookie fajl mora da zadovolji je:

  ime=vrednost[;EXPIRES=datum][;DOMAIN=imeDomena][;PATH=putanja][;SECURE]

COOKIE

- ime - ime koje definiše upisani cookie.
- vrednost - je upravo informacija koja se želi zapamtiti
- datum - je datum koji definiše do kada cookie ostaje upisan na klijentskoj mašini.
- imeDomena - definiše jedini domen sa kog cookie može da se čita i da mu se menja vrednost.
- putanja - definiše jedinu putanju sa koje cookie može da se čita i da mu se menja vrednost.
- SECURE - upis i čitanje cookie se izvršava preko posebnih, bezbednijih linija.
- Opcije EXPIRES, DOMAIN, PATH, SECURE su opcione i nije bitan redosled u kom se pojavljuju.

COOKIE

- Čitanje

  var citamCookie=document.cookie;

- Upis na klijentskoj strani

  document.cookie =
“primerCookie=”+vrednostKojuPamtim+”;secure”

COOKIE

```
html>
<head>
<script language="javascript">
function postavljanjeCookie(){
document.cookie = 'Cookie je='+document.forma1.imeCookie.value;
}
function prikazCookie(){
alert(document.cookie);
}
</script>
</head>
```

COOKIE

```
<body>
<h1>Cookie 1</h1>
<h2>Postavljanje i pregled cookie</h2>
<form name="forma1">
  <p><input name="imeCookie" type="text" id="imeCookie" size="20">
  </p>
  <p><input type="button" value="Upisite ime" name="B1"
onClick="postavljanjeCookie()">
<input  type="button" value="Prikazi cookie" name="B2"
onClick="prikazCookie()"></p>
</form>
</body>
</html>
```

RAD SA PAUZAMA I INTERVALIMA

- setTimeout("kod", brMilSek) - izvršavanje određenog koda nakon specificiranog vremesnkog intervala
- Kod koji se definiše u okviru setTimeout() metoda, izvršava se samo jednom

```
var id= setTimeout("kod", brMilSek);
```

- clearTimeout(id) – koristi se da bi se prekino metod setTimeout("kod", brMilSek) pre nego što se izvrši. Kao argument prima identifikator koji predstavlja rezultat poziva metoda setTimeout()

RAD SA PAUZAMA I INTERVALIMA

- setInterval("kod", brMilSek) – ponavlja izvršavanje određenog koda nakon specificiranog vremesnkog intervala

```
var id= setInterval("kod", brMilSek);
```

- clearInterval(id) – Koristi se da bi se prekinulo izvršavanje specificiranog koda u setInterval("kod", brMilSek). Kao argument prima identifikator koji predstavlja rezultat poziva metoda setInterval().