

JavaScript i FORME

- Programski jezik JavaScript je svoju popularnost stekao **moćnošću da pristupa elementima forme, čita njihove vrednosti, obrađuje ih i definiše nove vrednosti elemenata**. Takođe iskorišćena je i osobina HTML jezika da prepozna korisnikovu akciju i reaguje na nju.
- **Čitač može da prepozna svaku akciju korisnika, bilo da ona potiče od miša ili tastature.**

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 kursorom 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

Primer 1.

```
<SCRIPT LANGUAGE="JavaScript">
  if (document.images) {
    var add_o = new Image();
    add_o.src = './images/add_o.gif';

    var edit_o = new Image();
    edit_o.src = './images/edit_o.gif';
    var edit = new Image();
    edit.src = './images/edit.gif';
    var add = new Image();
    add.src = './images/add.gif';
  }

  function change(id,name){
    if (document.images) {
      document.images[id].src=eval(name+"."+id);
    }
  }
</SCRIPT>
```

Primer 1.

[illegible]

Primer 2. Dogadjaji miša

```
<script language="JavaScript">
    var counter=0;
    function closeWindow(){
        alert("Gotovo je!");
        window.close();
    }
    function mouseOverCounter(){
        counter++;
        if(counter==1){
            alert(counter + " prelazak preko dugmeta!");
        }
        else{
            alert(counter + " prelaska preko dugmeta!");
        }
    }
</script>
```

Primer 2.

```
<body onClick="closeWindow();">
<p><font face="arial" size=3>
  Dva puta kliknite da bi ste zatvorili prozor!
</p>
  Regstruje se prelazak misa preko linka.
<a href="#" onMouseOver="alert('Event:
onMouseOver');">onMouseOver</a>
<p>
  Regstruje se odlazak misa sa linka.
<a href="#" onMouseOut="alert('Event:
onMouseOut');">onMouseOut</a>
<p>
  Kada se mis pozicionira na dugme i pomeri poziva se funkcija<br>
  koja broji koliko puta se desio ovakav događaj.
<form>
  <input type="button" value="onMouseMove"
    onMouseMove="mouseOverCounter();">
</form>
</body>
```

Vrednosti elementa forme

- Navedeno je da JavaScript može i da pročita vrednost proizvoljnog elementa forme. Vrednosti elementa forme se prilazi u opštem slučaju na sledeći način:

document.imeForme.imeElementa.value

- gde je document službena reč, imeForme ime forme u okviru koje se nalazi element čijoj se vrednosti pristupa, imeElementa ime elementa i value službena reč.

Primer 3.

```
<SCRIPT LANGUAGE="JavaScript">
  function Saberi() {
    var br1 = document.mojforma.X.value - 0;
    var br2 = document.mojforma.Y.value - 0;
    var ukupno = br1 + br2; //sabiranje br1 + br2
    i smeštanje rezultata u promenljivu ukupno
    document.mojforma.zbir.value = ukupno;
  }
</SCRIPT>
```

Primer 3.

```
<FORM METHOD="post" NAME="mojforma">
  X &nbsp;  = &nbsp;  
  <INPUT TYPE="text" NAME="X" SIZE=5> <br>
  Y &nbsp;  = &nbsp;  
  <INPUT TYPE="text" NAME="Y" SIZE=5> <br><br>
  <INPUT TYPE="button" VALUE="SABERI" NAME="dugme"
    onClick="Saberi()"> <br><br>
  <hr>
  REZULTAT &nbsp;  = &nbsp;  
  <INPUT TYPE="text" NAME="zbir" SIZE=5>
  <br>
</FORM>
```

RAD SA VIŠE PROZORA

JavaScript omogućava da se iz jednog prozora formira, kontroliše ili menja sadržaj u okviru drugog prozora. U ovom poglavlju se razmatraju mogućnosti takvog rada. Prvo se razmatraju specijalne vrste prozora, kao što su alerti, a kasnije se razmatraju mogućnosti rada sa uobliženim prozorima.

Alerti

□ Alerti se koriste unutar HTML stranice kada se želi prikazati određeno obaveštenje - novi manji prozor

```
<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>
```

U okviru alerta korišćena je oznaka za prelazak u novi red : "\n"

Primer 4.

- ❑ Demonstracija događaja na raznim kontrolama forme.
- ❑ Doraditi primer tako da
 - 1. pri unosu podataka za lozinku korisnik ne može napustiti polje lozinke ako je ono kraće od 5 znakova i pri tome ne sadrži ni jedan broj. O pogrešnom unosu obavestiti putem alerta
 - 2. promena boje u polju 8 (My Favorite Color) istovremeno menja i boju pozadine cele stranice

window

- ❑ U ovaj objekat su uključene window metode za manipulaciju sa istim
- ❑ `window.open()` otvara novi prozor pretraživača
- ❑ `WindowName=window.open("URL", "WindowName", "Feature List");`
- ❑ `WindowName` je promenljiva. Koristeći ovu promenljivu možemo pozivati funkcije ili pristupati elementima istog
- ❑ `"URL"` je url za novi prozor. Ako je prazan ništa neće biti učitano.
- ❑ `"WindowName"` je ime prozora koje se koristi pri pozivu nekih funkcija
- ❑ `"Feature List"` je opcioni parametar. Čuva listu parametara odvojenih zarezima.
- ❑ Ovaj metod uljučuje
- ❑ **Width, height**, zatim nekoliko veličina koje mogu biti yes(1) ili no(0)
- ❑ **toolbar, location, directories, status, menubar, scrollbars, and resizable.**
- ❑ `SmallWin = window.open("", "small", "width=100,height=120,toolbar=0,status=0");`

Rad sa više prozora

```
function prozor( page, width, height, top, left ) {  
    var yes = 1; var no = 0;  
    var menubar = no;  
    var scrollbars = yes;  
    var locationbar = no;  
    var directories = no;  
    var resizable = no;  
    var statusbar = no;  
    var toolbar = no;  
    features = "" +  
        "width=" + width + "," +  
        "height=" + height + "," +  
        "top=" + top + "," +  
        "left=" + left + "," +  
        "features=" + "" +  
        (menubar ? ",menubars" : "") +  
        (scrollbars ? ",scrollbars" : "") +  
        (locationbar ? ",locationbar" : "") +  
        (directories ? ",directories" : "") +  
        (resizable ? ",resizable" : "") +  
        (statusbar ? ",statusbar" : "") +  
        (toolbar ? ",toolbar" : "");  
    window.open( page, "FullPopup", features );  
}
```

Cookie

Cookie su tekstualni fajlovi koji se mogu zapamtiti na mašini klijenta.

Cookie

- ❑ Format koji cookie fajl mora da zadovolji je:
`ime=vrednost[;EXPIRES=datum][;DOMAIN=imeDomena][;PATH=putanja][;SECURE]`
- ❑ `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 vrednosti:

```
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.form1.imeCookie.value;
}
function prikazCookie(){
alert(document.cookie);
}
</script>
</head>
<body>
<h1>Cookie 1</h1>
<h2>Postavljanje i pregled cookie</h2>
<form name="form1">
<p><input name="imeCookie" type="text" id="imeCookie" size="20">
<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

Korišćenjem metoda objekta Window može se realizovati kod koji izvršava automatski

setTimeout()

- Koristi se u okviru JavaScript-a za izvršavanje određenog koda nakon specifikiranog vremenskog intervala
- Kod koji se definiše u okviru setTimeout() metoda, izvršava se samo jednom
- Sintaksa upotrebe setTimeout() metoda:
`var variable = setTimeout("kod", brMilSek);`

ClearTimeout ()

- Metod se koristi da bi se prekinuo metod setTimeout() pre nego što se izvrši
- ClearTimeout() sadrži jedan argument:
 - Promenljivu koja predstavlja poziv metoda setTimeout()

- Druga dva metoda JavaScripta koji automatski izvršavaju određeni kod su:
 - setInterval() metod
 - clearInterval() metod

setInterval():

Je sličan metodu setTimeout() metodu, OSIM što ponavlja izvršavanje istog koda

clearInterval()

- ❑ clearInterval():
 - Koristi se da bi prekinuo izvršavanje metoda setInterval() na isti način kao što metod clearTimeout() poništava poziv metoda setTimeout()

Objekat History

- ❑ U okviru web čitača održava se interna lista (poznata pod imenom history list) svih dokumenata koji su bili otvarani tokom trenutne sesije Web čitača
- ❑ Svaki prozor Web čitača i frejm sadrže svoj sopstveni objekat History, koji predstavlja internu listu dokumenata

URL i History

- ❑ U okviru history list ne mogu se videti posećeni URL:
 - Ali se može napisati script koji koristeći ovu listu prolazi kroz Web stranice koje su bile otvarane tokom sesije Web čitača

Neka ograničenja

- ❑ U okviru Internet Explorera, može se koristiti JavaScript kod da bi se pretraživala history list
 - Jedino ako se trenutna Web page stranica nalazi na istom domenu kao i Web stranica koja sadrži JavaScript kod koji pokušava pretražiti listu

Objekat Location

- ❑ Dozvoljava da se promeni adresa nove Web stranice pomoću JavaScript koda
- ❑ Jedan razlog za ovu promenu je i mogućnost
 - Da se izvrši redirekcija korisnika stranice na drugu stranicu ili drugi URL
- ❑ Kada se koristi metod ili property objekta Location mora se
 - Uključiti i referenca na sam Location objekat

Objekat Navigator

- ❑ Koristi se da bi se dobile informacije o trenutnom Web čitaču
- ❑ Netscape i Internet Explorer sadrže jedinstvene metode i properties objekta Navigator koje se ne mogu koristiti sa ostalim čitačima
- ❑ Najviše se koriste metodi pomoću kojih se prepoznaje tip web čitača koji se koristi

Frame i Target

- Atribut Target definiše koji frejm ili prozor Web čitača će prikazati dokument:
 - Bazira se na vrednosti prikazanoj u okviru target atributa <a> elementa ili vrednosti u okviru atributa name <frame> elementa
-

<base>

- Atribut target se koristi i sa <base> elementom
 - da specificira default target za sve linkove u okviru dokumenta
 - Koristi data imena prozora ili frejma
-