

9. JavaScript – 2.dio



Iskočni okviri, funkcije, petlje,
događaji

Iskočni okviri (eng. popup boxes)

U JavaScriptu mogu se koristiti 3 vrste iskočnih okvira:

- **Upozoravajući okviri** (eng. **alert boxes**) – koriste se kako bi se osiguralo da je informacija stigla do korisnika, pa se traži da korisnik klikne na OK dugme.
- **Okviri za potvrdu** (eng. **confirm boxes**) – koristi se ako želimo da korisnik nešto potvrdi ili provjeri
- **Okviri za unos** (eng. **prompt boxes**) – koriste se kada želimo da korisnik unese neku vrijednost, a zatim potvrdi unos klikom na dugme OK.

Upozoravajući okviri (eng. alert boxes)

- Sintaksa:
alert("sometext");
- Primjer:

```
<script type="text/javascript">  
  alert("Unijeli ste pogrešnu šifru!")  
</script>
```

Kao rezultat pokazuje se okvir samo s tipkom OK.

Za pogled na primjer kliknite na [P9_Primjer1.html](#)



M. Zekić-Sušac

3

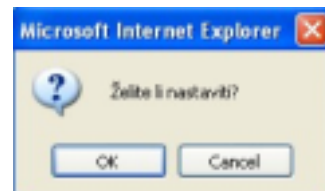
Okviri za potvrdu (eng. confirm boxes)

- Sintaksa:
confirm("sometext");
- Primjer:

```
<script type="text/javascript">  
  confirm("Želite li nastaviti?")  
</script>
```

Kao rezultat prikazuje se okvir s tipkama OK i Cancel. Tipka OK ima logičku vrijednost "Istina", a tipka "Cancel" vrijednost "Iaž".

Za pogled na primjer kliknite na [P9_Primjer2.html](#)



M. Zekić-Sušac

4

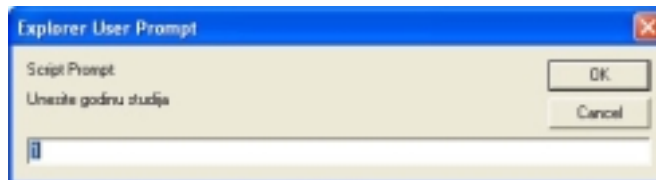
Okviri za unos (eng. prompt boxes)

- Sintaksa:
prompt("sometext","defaultvalue");

- Primjer:
`<script type="text/javascript">`
 `prompt("Unesite godinu studija", "1")`
`</script>`

Kao rezultat pokazuje se okvir s tipkama OK i Cancel. Korisnik treba unijeti podatak, a zatim potvrditi ili poništiti unos.

Za pogled na primjer kliknite na [P9_Primjer3.html](#)



M. Zekić-Sušac

5

Funkcije u JavaScript-u

- Funkcija = blok programskog koda koji se može više puta koristiti, a izvršit će se aktiviranjem nekog događaja, ili kad je pozvana
- Funkcije koristimo ako ne želimo nešto izvršiti odmah pri učitavanju stranice, nego po potrebi (pri nekom događaju ili po pozivu)
- Funkcije se mogu nalaziti unutar HTML dokumenta ili izvan njega, u posebnoj datoteci s nastavkom .js.
- Ako se nalaze unutar HTML dokumenta, one mogu biti u:
 - Zaglavlju <head> - tada se funkcija učitava prilikom učitavanja stranice, a izvršava se po pozivu
 - Tijelu stranice <body> - tada se funkcija ne učitava odmah pri učitavanju stranice, nego je potrebno naknadno je učitati i izvršiti.

M. Zekić-Sušac

6

[Primjer funkcije]

Primjer: Želimo napraviti funkciju koja će ispisivati poruku "Zdravo svijete!" kada je pozovemo.

- U zaglavlju stranice treba definirati funkciju:

```
<html>
<head>
<script type="text/javascript">
  function myfunction()
  {
    alert("Zdravo svijete!");
  }
</script>
</head>
<body>
```

Funkcija je nazvana "myfunction()"

M. Zekić-Sušac

7

[Primjer funkcije - nastavak]

- U tijelu stranice umetnut ćemo dugme "Pozovi funkciju", pri čijem će se kliku pozvati funkcija "myfunction()" na sljedeći način:

```
<form>
<input type="button" onclick="myfunction()"
  value="Pozovi funkciju">
</form>
<p>Klikom na dugme, bit će pozvana funkcija.
  Funkcija će prikazati okvir upozorenja.</p>
</body>
</html>
```

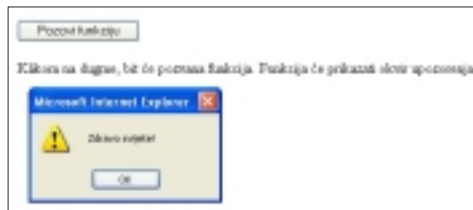
Za pogled na primjer kliknite na [P9 Primjer4.html](#)

M. Zekić-Sušac

8

Razlika između funkcije i običnog koda

- Da okvir upozorenja u prethodnom primjeru nije bio stavljen unutar funkcije, prikazao bi se odmah pri učitavanju stranice.
- S obzirom da je u funkciji, izvršava se tek nakon poziva, tj. u konkretnom slučaju na klik miša.



M. Zekić-Sušac

9

Definiranje funkcije

- Opća sintaksa za definiranje funkcije je:

```
function naziv_funkcije(var1,var2,...,varX)  
{  
  some code  
}
```

U zagradu pored naziva funkcije mogu se staviti argumenti (varijable koje se funkciji mogu proslijediti).

Ako funkcija nema parametara, zagrade ostaju prazne, ovako: naziv_funkcije().

Paziti na mala i velika slova: ključna riječ "function" mora biti napisana malim slovima, u suprotnom se funkcija neće izvršiti.

Funkcija može i vraćati neku vrijednost.

M. Zekić-Sušac

10

Vraćanje vrijednosti funkcije (return)

- Naredba "return" se koristi za specifikaciju vrijednosti koju neka funkcija vraća.
- Ukoliko funkcija vraća neku vrijednost, mora koristiti naredbu "return".

Primjer: Ova funkcija treba pomnožiti dva broja i vratiti rezultat množenja. Funkciji se prosleđuju dva broja: a i b na sljedeći način:

```
function prod(a,b)
{
    x=a*b;
    return x;
}
```

Funkciji se prosleđuju
vrijednosti varijabli
a i b

Funkcija će vraćati
vrijednost varijable
x

M. Zekić-Sušac

11

Lokalne i globalne varijable

Neka pravila u radu s funkcijama:

- Varijable u funkciji su LOKALNE. To znači:
 - Kada se deklarira neka varijabla unutar funkcije, njoj se može pristupiti samo u toj funkciji. Po izlasku iz funkcije, varijabla je uništena.
 - U različitim funkcijama moguće je imati lokalne varijable s istim nazivom, jer jedna funkcija vidi i prepoznaje samo svoje lokalne varijable, ali ne vidi lokalne varijable druge funkcije.
- Varijable se mogu deklairati i kao GLOBALNE - ako deklarirate varijablu izvan funkcije. Tada njoj mogu pristupiti sve funkcije. Takva varijabla traje sve od trenutka učitavanja stranice do trenutka zatvaranja stranice.

M. Zekić-Sušac

12

Primjer funkcije s argumentima - kojoj se proslijeđuju parametri – brožčane vrijednosti

```
<html>
<head>
<script type="text/javascript">
function pomnozi(a,b)
{
  x=a*b;
  alert("Rezultat je " + x);
}
</script>
</head><body><form>
<input type="button" onclick="pomnozi(2,3)" value="Rezultat">
</form>
<p>Klikom na dugme, bit će pozvana funkcija. Funkcija će pomnožiti dva broja.</p>
</body>
</html>
```

U zaglavlju je definirana funkcija kojoj se proslijeđuju dva broja a i b

U tijelu stranice se poziva ta funkcija i u zagradi se proslijeđuju vrijednosti.

Za izvršavanje primjera kliknite na [P9_Primjer5.html](#).

13

Primjer funkcije s argumentima - kojoj se proslijeđuju parametri - tekst

```
<html><head>
<script type="text/javascript">
  function pozdrav(txt)
  {
    alert(txt);
  }
</script>
</head>
<body>
<form>
<input type="button" onclick="pozdrav('Zdravo!')" value="Pozdravi me">
</form>
</body>
</html>
```

U zaglavlju se definira funkcija "pozdrav" koja će u okviru upozorenja ispisati neki tekst koji joj se proslijedi.

U tijelu stranice se poziva ta funkcija i u zagradi se proslijeđuje tekstualna vrijednost pod apostrofima.

Za izvršavanje primjera kliknite na [P9_Primjer6.html](#).

M. Zekić-Sušac

14

Primjer funkcije koja vraća vrijednost

```
<html>
<head>
<script type="text/javascript">
  function pozdrav()
  {
    return ("Zdravo!");
  }
</script>
</head>
<body>
<p>Skripta u tijelu stranice poziva funkciju pozdrav.</p>
<p>Funkcija vraća tekst.</p>
<script type="text/javascript">
  document.write(pozdrav());
</script>
</body>
</html>
```

Naredbom return označava se koju vrijednost funkcija vraća

Pozivom funkcije u naredbi document.write() ispisat će se vrijednost koju funkcija vraća

M. Zekić-Sušac 15

Za izvršavanje primjera kliknite na [P9_Primjer7.html](#).

Primjer funkcije kojoj se prosleđuju parametri i koja vraća vrijednost

```
<html>
<head>
<script type="text/javascript">
function product(a,b)
  {
    return a*b;
  }
</script>
</head><body>
<script type="text/javascript">
  document.write(product(4,3));
</script>
<p>Skripta u tijelu stranice poziva funkciju prosleđujući dva parametra (4 i 3). Funkcija vraća umnožak tih parametara.</p>
</body>
</html>
```

Funkcija prima argumente a i b, zatim vraća njihov umnožak

Pozivom funkcije u naredbi document.write() ispisat će se umnožak parametara koji se prosleđuju funkciji u ovoj naredbi (4 i 3)

Za izvršavanje primjera kliknite na [P9_Primjer8.html](#).

16

[Petlja for]

- Ova petlja koristi za uzastopno izvršavanje jedne ili više naredbi
- Sintaksa:

```
for (var=startvalue;var<=endvalue;var=var+increment)
{
    naredbe koje treba izvršiti
}
```

Primjer:

```
<html> <body>
<script type="text/javascript">
var i=0;
for (i=0;i<=10;i++)
{
    document.write("The number is " + i);
    document.write("<br />");
}
</script> </body> </html>
```

M. Zekić-Sušac

17

[Petlja for - nastavak]

- Kao rezultat prethodnog primjera na stranici će se ispisati poruke:

```
The number is 0
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
The number is 6
The number is 7
The number is 8
The number is 9
The number is 10
```

M. Zekić-Sušac

18

[Petlja while]

- Ova petlja koristi se za izvršavanje jedne ili više naredbi u slučaju kada je neki uvjet ispunjen

- Sintaksa:

```
while (var<=endvalue)
{
  naredbe koje treba izvršiti
}
```

Primjer:

```
<html> <body>
<script type="text/javascript">
var i=0;
while (i<=10)
{
  document.write("The number is " + i);
  document.write("<br />");
  i=i+1;
}
</script> </body> </html>
```

M. Zekić-Sušac

19

[Petlja while - nastavak]

- Rezultat prethodnog primjera isti je kao i za primjer sa "for" petljom. Ispisat će se poruke:

```
The number is 0
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
The number is 6
The number is 7
The number is 8
The number is 9
The number is 10
```

M. Zekić-Sušac

20

[Događaji u JavaScript-u]

- Događaji su akcije koje se u JavaScriptu mogu detekirati

Primjeri nekih karakterističnih događaja:

- Klik miša
- Učitavanje stranice ili slike
- Položaj miša iznad nekog kritičnog mjesta na web stranici
- Označavanje nekog okvira za unos u html formi
- Slanje podataka iz HTML forme (submit)
- Pritisnuta neka tipka na tipkovnici

[Kako se koriste događaji]

- Uz neki događaj obično se veže funkcija koja će se izvršiti kada se dogodi taj događaj. Funkcija se poziva tek kada se događaj dogodi (npr. na klik miša može se pozvati zatvaranje prozora)
- Koriste se pomoću ključnih riječi:
 - onload, onUnload
 - onFocus, onBlur i onChange
 - onSubmit
 - onMouseOver i onMouseOut

[Onload i onUnload događaji]

- Ovi događaji aktiviraju se kada korisnik učita stranicu (onload) ili kad je napusti (onUnload)
- Koriste se najčešće kako bi putem skripte provjerilo koji tip preglednika se koristi (npr. IE ili Mozilla), te da se učita ona verzija stranice koja je prilagođena tom pregledniku.
- Koriste se i za rad s kolačićima (cookies), koji se postavljaju kada korisnik napusti stranicu.
- Primjer: Ako na nekoj stranici želimo da korisnik unese svoje ime i prezime u nekom okviru. Kada napusti stranicu, želimo te podatke pohraniti u kolačić. Kad sljedeći put korisnik posjeti tu stranicu, želimo prikazati drugi okvir koji će ispisati poruku : Zdravo Pero Perić!.

M. Zekić-Sušac

23

[onFocus, onBlur i onChange]

- Ovi događaji se često koriste u kombinaciji s poljima u formi (obrascu).
- Primjer: Svaki put kad korisnik nešto promijeni u okviru za tekst u formi, pozvat će se funkcija checkEmail() koja će provjeriti mail:

```
<input type="text" size="30" id="email"  
  onchange="checkEmail()">
```

M. Zekić-Sušac

24

[onSubmit]

- Ovaj događaj koristi se za validaciju svih polja u formi koja se šalje.
- Primjer: Kada korisnik klikne na neko dugme za slanje forme, pozvat će se funkcija `checkForm()` koja će provjeriti ispravnost svih polja. Funkcija vraća vrijednost "istina" (ako je sve ispravno) ili "laž" (ako je neki podataka neispravno unesen). Ako je sve ispravno, forma će biti poslana i prikazat će se druga stranica "xxx.htm", a ako nije, forma neće biti poslana.

```
<form method="post" action="xxx.htm"  
onsubmit="return checkForm()">
```

[onMouseOver i onMouseOut]

- Ovi se događaji koriste kako bi se kreirala "animirana" dugmad ili dr. elementi stranice.
- Primjer: Kreirat ćemo animirani link – ako korisnik preko njega prijeđe mišem, aktivirat će se poruka upozorenja.

```
<html>  
<body>  
<a href="http://www.mathos.hr"  
onmouseover="alert('An onMouseOver event')">  
  
</a>  
</body>  
</html>
```

Za izvršavanje primjera kliknite na [P9_Primjer9.html](#)

[Literatura:]

- W3C, JavaScript Tutorial,
<http://www.w3schools.com/js/default.asp>, 11.04.08.