

# DEO I

## Uvod u JavaScript

---

### U OVOM DELU:

- ▶ Lekcija 1: Dodavanje JavaScript koda na web stranicu
- ▶ Lekcija 2: Promenljive i JavaScript sintaksa
- ▶ Lekcija 3: Funkcije
- ▶ Lekcija 4: Donošenje odluka
- ▶ Lekcija 5: Petlje
- ▶ Lekcija 6: Domet
- ▶ Lekcija 7: Objekti i ugrađeni tipovi
- ▶ Lekcija 8: Korisnički definisani objekti
- ▶ Lekcija 9: Prototipovi i nasleđivanje (Function objekat)



# Dodavanje JavaScript koda na web stranicu

Kada je JavaScript kod prvi put uveden u web stranice, kompanija Netscape je pokušala da nađe određeni mehanizam koji bi omogućavao da se JavaScript dodaje na web stranicu bez izazivanja određenih efekata u ostalim web čitačima. Nakon duže debate, konačno je odlučeno da se uvede `<script/>` element, koji je kasnije dodat u HTML specifikaciju.

## `<script/>` element

Primarni način dodavanja JavaScript koda u web stranicu je korišćenje `<script/>` elementa. Ovaj element, koji je uvela kompanija Netscape, postao je deo HTML 3.2 specifikacije, kao prelazni element za uvođenje HTML 4.01 standarda, u kome je ovaj element detaljno razrađen, uz dodavanje sledećih atributa:

- ▶ `src`: opcioni atribut. Specificira eksternu datoteku sa skriptom koji će se izvršavati.
- ▶ `type`: obavezan atribut. Specificira jezik koji se koristi za definisanje sadržaja `<script/>` elementa, a obično se koristi vrednost `"text/javascript"`. Iako je ovaj atribut obavezan na osnovu HTML specifikacije, svi čitači pretpostavljaju da se koristi JavaScript jezik ukoliko vrednost ovog atributa nije navedena.
- ▶ `language`: odbačeni atribut. Ovaj atribut definiše jezik koji se koristi u sadržaju `<script/>` elementa. Ovaj atribut se više ne koristi, jer ga je zamenio `type` atribut.
- ▶ `defer`: opcioni atribut. Logička vrednost (`true` ili `false`), kojom se definiše da li web čitač treba da sačeka sa izvršavanjem koda koji se nalazi u okviru `<script/>` elementa sve dok čitač ne učita celokupan HTML dokument.
- ▶ `charset`: opcioni atribut. Kodiranje karaktera eksterne datoteke sa kodom, koja je definisana pomoću `src` atributa. Ovaj atribut se retko koristi.

Postoje dva načina za korišćenje `<script/>` elemenata: direktnim ugrađivanjem JavaScript koda u stranicu, kada se radi o *inline kodu*, ili korišćenjem `src` atributa koji definiše odgovarajuću eksternu datoteku.

Da biste dodali inline kod na web stranicu, postavite kod između početnog i krajnjeg taga, što je prikazano u sledećem primeru:

```
<script type="text/javascript">
function inlineScript() {
    var message = "Hello, World!";
    alert(message);
}

inlineScript();
</script>
```

Ne razmišljajte trenutno o samom kodu - naučićete uskoro šta je rezultat izvršavanja koda prikazanog u prethodnom primeru. Iako web čitač automatski pretpostavlja da je kod u okviru `<script/>` elementa napisan korišćenjem JavaScript jezika, smatra se da je dobra programerska praksa da se uključi ovaj atribut. Čitač zaustavlja učitavanje ostatka stranice kada pristupi `<script/>` elementu; učitava kod i interpretira ga pre obrade ostatka HTML stranice.

Veoma je važno da znate da inline blokovi koda, kao onaj koji je prikazan u prethodnom primeru, treba da budu uključeni u CDATA blok u XHTML dokumentima. Ukoliko to ne učinite, doći će do pojave greške prilikom izvršavanja koda u čitaču. Zapravo, svi najpoznatiji čitači i dalje učitavaju JavaScript kod i normalno ga izvršavaju. Međutim, ne postoji garancija da će i buduće verzije čitača funkcionisati na ovaj način. Ukoliko koristite XHTML kod, obezbedite svoj inline skript tako da može da se normalno izvršava u budućnosti, tako što ćete ga inkapsulirati u CDATA sekciju, što je prikazano u sledećem primeru:

```
<script type="text/javascript">
<![CDATA[[
function inlineScript() {
    var message = "Hello, World!";
    alert(message);
}

inlineScript();
]]>
</script>
```

Dodavanje eksterne JavaScript datoteke na stranicu podrazumeva uključivanje `src` atributa. Možete da iskopirate JavaScript kod iz prethodnog primera u novu datoteku, bez navođenja `<script>` tagova. Nakon toga, korišćenjem `src` atributa, stranica može da referencira eksternu JavaScript datoteku, kao što je prikazano u sledećem primeru:

```
<script type="text/javascript" src="sample.js"></script>
```

Ovaj `<script/>` element referencira eksternu JavaScript datoteku pod nazivom `sample.js`. Korišćenje `.js` ekstenzije je konvencija koju primenjuju svi JavaScript programeri. Nije bitan naziv datoteke - čitač će preuzeti datoteku i interpretirati sadržaj datoteke kao JavaScript kod.

Identično kao i kod izvršavanja koda koji se direktno navodi na stranici, čitač zaustavlja obradu HTML stranice kada pristupi `<script>` tagu, ali je neophodno da preuzme datoteku sa kodom pre nego što počne da ga interpretira. Nakon što čitač učita JavaScript kod iz eksterne datoteke, nastavlja obradu HTML stranice.

Da li ćete navoditi skript u kodu ili koristiti eksterne skripte zavisi isključivo od vas. Međutim, većina programera koristi spoljašnje JavaScript datoteke iz nekoliko razloga:

- ▶ Eksterni skriptovi su mnogo jednostavniji za održavanje u odnosu na skripte čiji se kod navodi u HTML datoteci, posebno ukoliko koristite isti kod na nekoliko stranica. Zamislite koliko bi složeno bilo editovanje koda na više HTML stranica da bi se omogućilo izvršavanje identičnih promena blokova koda. To bi, svakako, bio vremenski veoma zahtevan zadatak. Ukoliko koristite eksternu datoteku, neophodno je da jednostavno promenite samo jednu datoteku.
- ▶ Eksterne skripte čitač kešira isto kao i slike i definicije stilova. Čitač preuzima skript samo jednom i koristi ga na svim narednim stranicama koje zahtevaju ovu datoteku, sve dok je ne budete izmenili. To garantuje određene dobitke performansi, jer čitač ne mora da preuzme isti kod više puta, a istovremeno se identičan kod ugrađuje u HTML stranicu.

Kako i gde se postavljaju `<script/>` elementi na stranici veoma je važno - kod se učitava i interpretira u redosledu u kome su postavljeni ovi elementi na stranici.

## Postavljanje tagova

Sam `<script/>` element je jedan od nekoliko elemenata koji mogu da se nađu bilo gde na HTML stranici: može da bude u okviru `<head/>` elementa ili `<body/>` elementa. Tradicionalniji način korišćenja je postavljanje u okviru `<head/>` elementa, zajedno sa `<style/>` i `<link/>` elementima. Sledeći HTML kod upravo to i prikazuje:

```
<html>
<head>
  <title>Sample Page</title>
  <script type="text/javascript" src="sample.js"></script>
</head>
<body>

</body>
</html>
```

Iako time postizete da svi resursi na HTML stranici budu na jednom mestu u dokumentu, zapamtite da čitač zaustavlja obradu HTML dokumenta nakon što pristupi `<script/>` elementu, kako bi preuzeo i učitao skript koji se referencira. Tako postavljanje `<script/>` elemenata na početku dokumenta ima i bočne efekte, vezane za zaustavljanje učitavanja sadržaja web stranice u čitaču. Posmatrano sa stanovišta korisnika, radi se o problemu vezanom za performanse, jer oni vide praznu stranicu dok čitač preuzima i učitava JavaScript kod.

Zbog toga, postala je dobra praksa da se svi `<script>` tagovi postavljaju neposredno pre završnog `<body>` taga, kao što je prikazano u sledećem primeru:

```
<html>
<head>
  <title>Sample Page</title>
</head>
<body>

  <script type="text/javascript" src="sample.js"></script>
</body>
</html>
```

Na ovaj način, čitač učitava i prikazuje celokupnu stranicu pre nego što učitava JavaScript kod. Iako je vreme preuzimanja koda identično, nezavisno od toga gde se nalaze `<script>` tagovi, korisnik će videti zahtevanu web stranicu ranije i samim tim imati osećaj da se web stranica brže učitava.

## Probajte sami

U ovoj lekciji, dodaćete eksternu JavaScript datoteku na web stranicu.

### Zahtevi vezani za lekciju

Za potrebe ove lekcije, neophodan vam je editor teksta; možete da koristite bilo koji editor teksta. Korisnici operativnog sistema Microsoft Windows mogu da koriste Notepad, jer je on raspoloživ na sistemu, a mogu da preuzmu i besplatan Visual Web Developer Express alat ([www.microsoft.com/express/web/](http://www.microsoft.com/express/web/)) kompanije Microsoft ili Web Matrix alat ([www.asp.net/web-matrix/](http://www.asp.net/web-matrix/)). Korisnici operativnog sistema Mac OS X mogu da koriste TextMate, koji dolazi uz operativni sistem, ili mogu da preuzmu probnu verziju Coda alata ([www.panic.com/coda/](http://www.panic.com/coda/)). Korisnici operativnog sistema Linux mogu da koriste ugrađeni napredni editor teksta VIM.

Neophodno je da imate i savremeni web čitač. Izaberite jedan od sledećih:

- ▶ Internet Explorer 8+
- ▶ Google Chrome
- ▶ Firefox 3.5+
- ▶ Apple Safari 4+
- ▶ Opera 10+

Na računaru kreirajte folder pod nazivom JS24Hour, a zatim kreirajte podfolder pod nazivom Lesson01. Sve datoteke koje kreirate u ovoj lekciji smeštajte u Lesson01 folder.

### Korak po korak

Dodajte eksternu JavaScript datoteku na vašu web stranicu tako što ćete uraditi sledeće:

1. Otvorite tekstualni editor i unesite sledeći JavaScript kod:

```
function inlineScript() {  
    var message = "Hello, World!";  
    alert(message);  
}  
  
inlineScript();
```

2. Snimite datoteku pod nazivom lesson01\_sample01.js.
3. Otvorite još jednu datoteku u vašem editoru teksta i unesite sledeći HTML kod:

```
<html>  
<head>  
    <title>Sample Page</title>  
</head>
```

```
<body>  
  
    <script type="text/javascript" src="lesson01_sample01.js"></script>  
</body>  
</html>
```

4. Snimite ovaj kod u datoteku pod nazivom `lesson01_sample01.html`, i to tako da se nalazi na istoj lokaciji na kojoj se nalazi i `lesson01_sample.js` datoteka.
5. Otvorite HTML datoteku u svom web čitaču, tako što dva puta kliknete odgovarajuću datoteku. Kada čitač učita stranu treba da se otvori alert box sa tekстом "Hello, World!".



*Molimo vas da selektujete Lesson 1 na pratećem DVD-u kako biste pregledali video koji je vezan za ovu lekciju. Takođe, možete da preuzmete izvorni kod za sve lekcije koje se nalaze u ovoj knjizi, tako što ćete pristupiti web stranici na adresi [www.wrox.com](http://www.wrox.com).*

