

**REGIONALNI CENTAR ZA TALENTE VRANJE**

---

**UPOTREBA JAVASCRIPTA U IZRADI WEB SAJTA OŠ  
„8.OKTOBAR”, VLASOTINCE**

**THE JAVASCRIPT USAGE IN THE PRIMERY SCHOOL'S  
"8.OKTOBAR" VLASOTINCE WEB SITE CONSTRUCTION**

**Autor:** LAZAR STAMENKOVIĆ, učenik VII razred OŠ „8.oktobar”- Vlasotince

**Mentor:** dipl.ing. IT spec.app. el. i rač. DEJAN KRSTIĆ, OŠ „8.oktobar”- Vlasotince

**Vlasotince 2014.**

**UPOTREBA JAVASCRIPTA U IZRADI WEB SAJTA OŠ  
„8.OKTOBAR”, VLASOTINCE  
THE JAVASCRIPT USAGE IN THE PRIMERY SCHOOL'S  
"8.OKTOBAR" VLASOTINCE WEB SITE CONSTRUCTION**

**Autor:** LAZAR STAMENKOVIĆ, učenik VII razred OŠ „8.oktobar”- Vlasotince  
**Mentor:** dipl.ing. IT spec.app. el. i rač. DEJAN KRSTIĆ, OŠ „8.oktobar”- Vlasotince

**REZIME**

U ovom radu je predstavljena efikasna metoda za prikaz Web stranica zasnovana na upotrebi JavaSkripta (engl. *JavaScript*). Prednost upotrebe JavaSkripta u odnosu na statičko prikazivanje Web sadržaja može se gledati kao dobitak iz ugla klijentske strane. Naime, dinamički sadržaj na Internet stranici obogaćuje kompletan vizuelni sadržaj stranice. Treba, naravno, voditi računa i u optimizaciji tog sadržaja, jer gledano sa klijentske strane klijent će biti zadovoljniji prikazom stranice koja se otvori u što kraćem roku. Programeri, radi ostvarivanja tog cilja, pri izradi Web sajta koriste i optimizaciju zasnovanu na CSS3 (engl. *Cascading Style Sheets*). Važna stavka pri izradi sajta koja se stavlja pred programerom je usklađivanje svih adekvatnih alata koji se koriste za izradu sajta, od upotrebe HTML-a (engl. *HyperText Markup Language*), XHTML-a (engl. *Extensible HyperText Markup Language*), JavaScripta, PHP-a (engl. *HyperText Preprocessor*), preko CSS3, u jednu celinu kako bi se Web stranice na klijentskoj strani u što kraćim vremenskim intervalima otvarale, tj. kako bi odziv sa servera bio što brži. Naravno, treba uzeti u obzir Web Browsers koji moraju permanentno usklađivati svoje standarde sa standardima jezika za pisanje kodova koji se koriste za Web dizajn. Na primeru izrade Web sajta pomenute škole u naslovu rada, pokazaće se prednost upotrebe JavaSkripta.

**Ključne reči:** JavaScript, Css, HTML, XHTML, PHP, Web Browser, Web server, Web client

**ABSTRACT**

This paper presents an efficient method to display the Web page based on the use of JavaScript. The advantages of using JavaScript in relation to the static presentation of Web content can be viewed from the perspective of the client side. The dynamic content on the website enriches the entire visual content of the page. However, the optimization of that content is also important, as seen from the client side, the client will be satisfied with the page that opens as soon as possible. Developers use also the optimization based on CSS3 to achieve this goal in making Web sites. An important issue in the site development is how to harmonize all the proper tools used for website developmen , the use of HTML (hypertext markup language), XML (engl. *Extensible HyperText Markup Language*), JavaScript, PHP text (the *HyperText Preprocessor*), CSS3 into one entity,the to make Web pages on the client side be opened in the shortest possible time, ie . to get the server response as fast as possible. Course, the Web browsers that must constantly adjust its standard with standard language for writing code that is used for Web design need to be taken into account.

**Key words:** JavaScript, Css, HTML, XHTML, PHP, Web Browser, Web server, Web client

## I UVOD

Kroz projekat će se prikazati upotreba JavaScripta kao Web alata za dizajniranje dinamičkih sadržaja na stranicama Web sajta. Uz kombinaciju drugih alata koji se koriste u Web dizajniranju, kroz projekat je nemoguće zaobići skriptne jezike HTML, XHTML i PHP. Konačno, istražićemo i tehnike pomoću kojih ćemo optimizovati sve aspekte Web lokacije. Programeri i Web dizajneri se trude permanentno da naprave Web lokaciju koja će biti još brža, bezbednija i lakša za održavanje. S druge strane, posetioци Web lokacija ne žele samo zanimljiv sadržaj, već i sadržaj koji se brzo preuzima. U današnje vreme je na tržištu veliki broj hosting kompanija ili manjih firmi koji iznajmljuju prostor na Web-u. Tržište diktira cenu zakupa prostora, veličinu (količinu sadržaja Web podataka), pa je i ovo tema koja se ne može zaobići. Jednom rečju, koliko novca smo spremni da izdvojimo za zakup prostora na Web-u, toliko ćemo moći da napravimo kvalitet za klijenta, tj. posetioca našeg Web sajta. Uzimajući u obzir navedene činjenice, ne možemo zaobići ili bar dotaći temu Web Browser-a. Naime, i kompanije koje su vlasnici Web Browser-a moraju standardizovati tehnologiju, kako bi pročitali ili prikazali neki segment ili sadržaj koji je napravio Web programer.

## ALATI I METODE RADA

Prilikom rada koristio sam eksperimentalne metode, metode komparacije, metode analize i sinteze, metode evaluacije (vrednovanaja), deskripcije (opisa) i praktičnu metodu rada. Od alata, korišćeni su softver za obradu slike Adobe PhotoShop CS 5.1, softver za obradu video fajlova i alati koji se koriste u Web programiranju: JavaScript, HTML, XHTML, PHP, CSS.

## II JAVASCRIPT

Dakle, šta je JavaScript? Šta on radi? Na ova pitanja će biti odgovoreno kroz ovaj rad, tekstualno i praktično. Java je u osnovi skriptni jezik koji pomaže izbacivanje XHTML u Owerdrive. Razvijen je od strane Netscape kompanije. Sa njim, elementi u dokumentu donose "dosadnu" Web stranicu. Međutim, korišćenje XHTML za kreiranje Web stranice u kombinaciji sa JavaScript-om, je u stvari "trljanje kamena sa štapovima da bi se zapalila vatra". Sa XHTML-om Web dizajneri su ograničeni na kreiranje statičkih, tj. neinteraktivnih Web stranica. Zapravo, u današnjim standardima Interneta ovo je "neprihvatljivo", kada imamo u vidu koje sve standarde podržavaju Web Browseri. Da ne bude zabune, Java je potpuno drugačiji od JavaScripta. On je mnogo moćniji, složeniji jezik i spada u rang C, C++ jezika i sl. Takođe, potrebno je sastaviti Java program pre nego ga budemo pokrenuli, dok kod

JavaScript-a, bez kompilacija, potrebno je jednostavno otvoriti editor teksta, ukucati ga, sačuvati, i naš pregledač je spreman da ga pokrene. JavaScript je programski jezik koji se koristi da bi programeri automatizovali obavljanje nekih poslova na Web stranici. Sam HTML nema mogućnosti da saopšti čitaču da izvrši niz naredaba. Bez script jezika kao što je JavaScript, web stranice nemaju mogućnost obrade ili bilo kakvog načina komunikacije s posetiocem stranice. Gde JavaScript kodove treba staviti? Pa, u osnovi bilo gde unutar <xhtml> oznaka na stranici. Početak koda počinje <script tipe="text/javascript"> i završava se sa </ script>.

JavaScript iskazi se navode unutar HTML oznaka kojima se definiše stranica. Ti iskazi mogu, na primer, proveravati ispravnost podataka koje je korisnik uneo u obrazac. Program ili script proveravaju da li svako polje ima vrednost i da li je ta vrednost ispravna, odnosno da li ima smisla. JavaScript iskazi mogu i reagovati na akcije korisnika. Na primer, pomoću JavaScripta može se napraviti padajući meni s tekstom za pomoć koji se pojavljuje kada posetilac pritisne vezu za aktiviranje sistema pomoći.

Sledeći primer prikazuje prost primer sadržaja JavaScript koda:

```
<html>

<head> <title> Ovo je primer </ title> </ head>

<body>

Dobrodošli u JavaScript naravno!

<script tipe="text/javascript">

<!--

dokument.write ("Zdravo. Ovaj tekst je napisan koristeći
JavaScript !")

// ->

</ Script>

</ body>
```

```
</ xtml>
```

Naš Browser će pokazati na stranici izlaz: Zdravo. Ovaj tekst je napisan koristeći JavaScript !

### III JAVASCRIPT I KOMPATIBILNOST SA ČITAČIMA WEB-a

Za pisanje JavaScripta nije potreban poseban alat; može da se koristi isti program za uređivanje teksta koji se koristi za izradu Web stranice. Osim toga, najpopularniji čitači Weba imaju podršku za JavaScript. Zbog toga posetioci lokacija ne treba da instaliraju dodatne programe na svoje računare; čitač može da izvrši skriptove na JavaScriptu koje umetnemo u HTML na Web stranici. To možemo odraditi tako što ćemo JavaScript iskaze staviti između početne i završne HTML oznake za script (<script></script>), i oni će naložiti Web čitaču da izvrši iskaze umesto da se prikaže tekst na ekranu. Svi poznati Web čitači prate permanentno razvoj svih jezika za Web dizajniranje, da bi mogli da se bore sa konkurencijom. Međutim, kako savremene računarske tehnologije napreduju “brzinom svetlosti”, neki Web čitači već zaostaju, tj. ne mogu da prikažu sadržaj na klijentskoj stranici koje je Web programer odradio. Tako se može desiti da neki Web čitač i ne razume sadržaj (kod), koji je napisao programer koji koristi sve najsavremenije Web alate, što će na kraju rezultirati neprikazivanjem sadržaja na klijentskoj stranici. Lako je izvesti zaključak da Web Browseri moraju pratiti standarde i praviti svoje čitače da budu kompatibilni sa svim Web alatima koji su danas dostupni za izradu Web stranica, inače ispadaju iz konkurencije. Iskusniji programeri znaju šta treba uraditi da bi njihov kod web čitač “prepoznao.” Kada u HTML dokument uvrstimo script, nije osigurano da čitač Weba posetioca podržava izvršenje skripta. Da bismo upozorili posetioca na HTML document koji sadrži skript možemo umetnuti tekst upozorenja između oznaka <noscript> i </noscript>. Oznake moraju da se nalaze u telu Web stranice, između početne i završne oznake za telo (<body></body>). Najbolje mesto za umetanje oznake <noscript> je odmah nakon oznake </script>. Primer koji prikazuje ovaj unapred navedeni postupak može izgledati ovako:

```
<html>
```

```
<head>
```

```
    <title> A JavaScript example    </title>
```

```
</head>
```

```
<body>

<script language="JavaScript">

<!--

    document. Write ("This is a JavaScript example!")

//-- >

</script>

<noscript> No JavaScript support... SORRY! </noscript>

</body>

</html>
```

Iako skript u ovom primeru saopštava posetiocu da njegov čitač nema podršku za JavaScript, trebalo bi da napišemo opširniju poruku. Na primer, možemo objasniti kako se uključuje JavaScript u čitaču ili da napravimo hipervezu koju posetilac može da upotrebi za preuzimanje čitača sa podrškom za JavaScript.

#### **IV UČITAVANJE SLIKA UNAPRED RADI USKLADIVANJA VREMENA ZA PRIKAZ**

Učitavanje slika je tehnika kojom saopštavamo čitaču weba da preuzme slike sa Web servera, koje će se možda prikazati kasnije. Saopštavanjem čitaču da preuzme slike u pozadini (tj. dok posetilac na Web stranici radi nešto drugo), možemo eliminisati kašnjenje pri učitavanju koje nastaje kada posetilac pritisne umanjenu sliku da bi prikazao veću. Ako se na Web stranici znatno koriste skriptovi koji prikazuju slike za efekat prelaza mišem ili animaciju, naravno da ćemo želiti da uskladištimo slike koje još nisu prikazane. Uskladištena slika će se odmah pojaviti na ekranu kada je čitač pozove, umesto (verovatno dugog) kašnjenja dok se čeka da Web server pošalje sliku čitaču. Na primer, ako imamo Web stranicu sa "četiri minijaturne" sličice (slika 1).



*Slika 1, minijaturne slike*

*Picture 1, miniature picture*

Slike koje će čitač prikazati kada posetilac pritisne minijaturnu sliku, možemo učitati dodavanjem atributa obrade događaja *onLoad* u oznaku `<body>` Web stranice ili u jednu od oznaka za minijaturnu sliku `<img>`. Konkretno ako dodamo atribut obrade događaja *onLoad* oznaci `<img>` za poslednju minijaturnu sliku prikazanu na slici 4.1, čitač Weba će pozvati korisnički definisanu funkciju *preCashet()*:

```

<td>a href="images/Pix4.jpg">

    </a></td

```

Učitavanje slike je jedan od onih vremenskih testiranja JavaScript tehnikama koji ostaju popularni uvek za ubacivanje slike što je pre moguće u pozadini. Ideja je da se podstakne proces čim strana počne da se učitava, a aplikacija treba da prikaže efekat prevrtanja slika (menjenja slika u određenom vremenskom razmaku). Sledeća jednostavna funkcija unapred učitava bilo koji broj slika:

```

<span
class="notranslate" onmouseover="_tipon(this)" onmouseout="_tipoff()"><span class="google-src-text" style="direction: ltr; text-align:left">function preloadimages(arr){</span>function
preloadimages (obr) {</span>

    <span
class="notranslate" onmouseover="_tipon(this)" onmouseout="_tipoff()"><span class="google-src-text" style="direction: ltr; text-align: left">var newimages=[]</span> var newimages =

```

[ ]</span>

```
<span
class="notranslate" onmouseover="_tipon(this)" onmouseout="_tipoff()"><span class="google-src-text" style="direction: ltr; text-align: left">var arr=(typeof arr!="object")?</span> var arr = (tipoff arr! = "предмет")?</span> <span class="notranslate" onmouseover="_tipon(this)" onmouseout="_tipoff()"><span class="google-src-text" style="direction: ltr; text-align: left">[arr] : arr //force arr parameter to always be an array</span> [Ap]: обр / / сила обр параметар да увек буде низ</span>
```

```
<span
class="notranslate" onmouseover="_tipon(this)" onmouseout="_tipoff()"><span class="google-src-text" style="direction: ltr; text-align: left">for (var i=0; i<arr.length; i++){</span> за (var i = 0; I <arr.lenght и + +) {</span>
```

```
<span
class="notranslate" onmouseover="_tipon(this)" onmouseout="_tipoff()"><span class="google-src-text" style="direction: ltr; text-align: left">new images[i]=new Image()</span> new Image [и] = new Image ()</span>
```

```
<span
class="notranslate" onmouseover="_tipon(this)" onmouseout="_tipoff()"><span class="google-src-text" style="direction: ltr; text-align: left">new images[i].src=arr[i]</span> new images [i]. src = obr [i]</span>
```

```
<span
class="notranslate" onmouseover="_tipon(this)" onmouseout="_tipoff()"><span class="google-src-text" style="direction: ltr; text-align: left">}</span> }</span>
```

```
<span
class="notranslate" onmouseover="_tipon(this)" onmouseout="_tipoff()">
```



```
ff() "><span class="google-src-text" style="direction: ltr; text-align: left">}</span> }</span>
```

```
<span class="notranslate" onmouseover="_tipon(this)" onmouseout="_tipoff() "><span class="google-src-text" style="direction: ltr; text-align: left"> //preload 3 images:</span> / / preload 3 images:</span>
```

```
<span class="notranslate" onmouseover="_tipon(this)" onmouseout="_tipoff() "><span class="google-src-text" style="direction: ltr; text-align: left">preloadimages(['1.gif', '2.gif', '3.gif'])</span> preloadimages(['1.gif ', '2.gif ', '3.gif '])</span>
```

Za najveći deo dodavanja željene slike ova tehnika je dovoljna. Međutim, ponekad je potrebno uraditi nešto više od navedenog. Ako je nekada potrebno da slika bude centrirana u okviru stranice potrebno je da se slike “pravilno orijentišu”. Potrebno je uraditi pre svega pikselizaciju slika i centriranje slika po slobodnom personalnom estetskom ubeđenju, tj. odrediti širinu i visinu slike. Moć JavaScript-a u kombinaciji sa PHP-om je velika. PHP (engl.) i Java mogu da rade zajedno na način koji ide u veliku korist JavaScript-a. Sintaksa za referenciranje PHP datoteke koristeći JavaScript je:

```
<script tipe="text/javascript" src="miscrypt.php"> </ script>
```

gde je "miscrypt.php" apsolutna ili relativna putanja do PHP scripta umesto uobičajene datoteke JavaScript-a. Do parametra za PHP script možemo da dođemo i kroz URL (engl.):

```
<script tipe="text/javascript" src="miscrypt.php?id=3 name=george"> </script>
```

Jedna osobina DXHTML je funkcija IE5+koja ima dinamička svojstva. Po samoj svojoj prirodi, osobine definisane XHTML-om su statične, tj. ne menjaju se (tj.:<div stile="width: 300px"> “). Dinamičke osobine nam omogućavaju da zamenimo statičke vrednosti u XHTML sa dinamičnim izrazima, pa umesto “300”, to bi moglo biti “polovina širine prozora

pregledača”. To je korisna funkcija koja služi da drastično smanji količinu napora potrebnog da se dinamički menjaju određene vrednosti XHTML-a.

## **V KONKRETAN PRIMER UPOTREBE JAVASCRIPTA NA STRANICI WEB SAJTA OŠ “8.OKTOBAR”- VLASOTINCE**

JavaScript ima široku primenu i veliku moć u Web dizajniranju. U prethodnom izlaganju sam objasnio neke tehnike i alate sa kojima se može kombinovati da bi došle do izražaja sve prednosti ovog skriptnog jezika. Na primeru školskog sajta pokazao sam kako se upotrebljava Javascript. Postavio sam script za izvršavanje na naslovnoj strani, kako bi se dobio dinamički karakter 5 slika koje se menjaju jedna za drugom u kratkim vremenskim intervalima. Slike su prethodno obrađene pomoću Photo Shop-a, pikselizacija slika isto u ovom programu, centriranje slike na Web stranici određene su u samom kodu (“width”; height”). Slike koje se menjaju prikazane su u nastavku rada (slika 2; slika 3; slika 4; slika 5; slika 6; slika 7). Iskoristio sam karakteristike JavaScripta za postavljanje dinamičkog sadržaja na naslovnoj stranici proračunato, jer smatram da naslovna strana bilo kog sajta treba da privuče pažnju posetilaca. Uzgred, naslovna strana treba da bude pregledna i upotrebio sam svoje dozajnersko iskustvo postavljajući glavne “kartice”, sa padajućim menijima.



*Slika 2, primer slike u JavaScript*

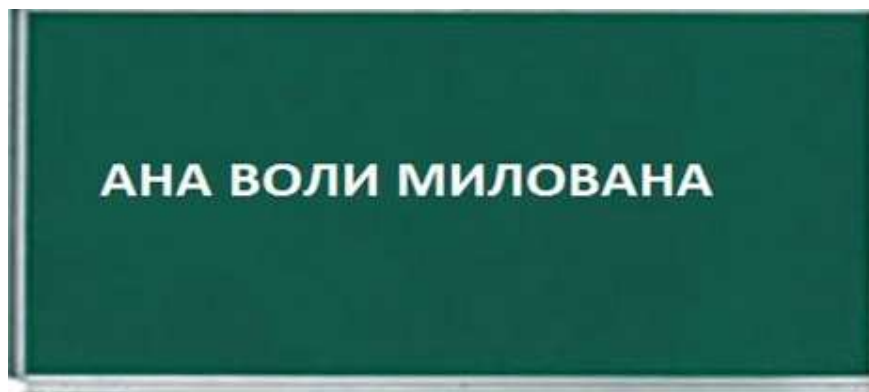
*Picture2, example images in JavaScript*



*Slika 3, primer slike u JavaScript*  
*Picture3, example images in JavaScript*



*Slika 4, primer slike u JavaScript*  
*Picture 4, example images in JavaScript*



*Slika 5, primer slike u JavaScript*  
*Picture 5, example images in JavaScript*



*Slika 6, primer slike u JavaScript*  
*Picture 6, example images in JavaScript*



*Slika 7, primer slike u JavaScript*  
*Picture 7, example images in JavaScript*

JavaScript kod koji prikazuje slike na naslovnoj strani izgleda ovako:

```
<script tipe="text/javascript">
```

```
$(function() {  
  new  
  WB_Gallery({ "id": "wb_element_instance3", "type": "slideshow", "in  
  terval": 4, "speed": 400, "width": 790, "height": 300, "border": { "colo  
  r": "#FFFFFF", "style": "solid", "weight": 5, "css": { "border": "5px  
  solid
```

```

#FFFFFF"}}, "thumbWidth":64, "thumbHeight":64, "images": [ {"thumb"
:"gallery\6-ts1396733576.jpg.thumb.jpg", "image": "gallery\6-
ts1396733576.jpg", "width":1800, "height":745}, {"thumb": "gallery
\5-ts1396733574.jpg.thumb.jpg", "image": "gallery\5-
ts1396733574.jpg", "width":1800, "height":754}, {"thumb": "gallery
\3-ts1396733569.jpg.thumb.jpg", "image": "gallery\3-
ts1396733569.jpg", "width":1800, "height":904}, {"thumb": "gallery
\4-ts1396733571.jpg.thumb.jpg", "image": "gallery\4-
ts1396733571.jpg", "width":1800, "height":801}, {"thumb": "gallery
\1-ts1396451093.jpg.thumb.jpg", "image": "gallery\1-
ts1396451093.jpg", "width":1800, "height":1276}, {"thumb": "galler
y\2-ts1396733566.jpg.thumb.jpg", "image": "gallery\2-
ts1396733566.jpg", "width":1800, "height":801}}]);
});
</script></div>

```

## VI ZAKLJUČAK

Što veći mogući broj poseta našoj Web lokaciji i sadržajima koji se nalaze na njoj je zapravo uspeh Web programera i njegovog dizajnerskog iskustva. Da bi se dočarao i što dinamičnije prikazao sadržaj na klijentskoj strani Web programer koristi sve raspoložive alate. U ovom radu je upotreba JavaScripta u kombinaciji sa drugim alatima opravdala krajnji cilj. Korišćenjem novih Web tehnologija autori Web stranica će moći da zadovolje rastuću potražnju za kvalitetnim mobilnim sadržajima bez izmene standardnog HTML koda. Najvažnija prednost dobijena korišćenjem ovih tehnologija ogleda se u tome što se osnovna i mobilna Web stranica nalaze na istoj Web adresi. Ovo ima za posledicu smanjenje troškova održavanja i olakšava posao administriranja Web stranice. Istovremeno, dizajniranje standardne i mobilne Web stranice ima prednost u tome što se unapred određuje izgled stranica na svim tipovima uređaja. Ranija rešenja koja su podrazumevala adaptaciju Web stranice od strane Web čitača nisu dala zadovoljavajuća rešenja. Upotreba JavaScripta kao dizajnerskog Web alata opravdava u potpunosti svoju ulogu u pravljenju dinamičkog sadržaja Web stranica. Kombinovanjem sa drugim modernim alatima za Web programiranje dobija se još veća dinamika prikazivanja Web sadržaja, što će u potpunosti zadovoljiti

potrebe posetilaca Web lokacije, što je i glavni cilj. Ovim radom smo prikazali neke od mogućnosti i neke moći od širokog dijapazona koje nudi upotreba JavaScripta.

## VII LITERATURA

1. Bhangal, Sham; Jankowski, Tomasz (2003): *Foundation Web Design: Essential HTML, JavaScript, CSS, PhotoShop, Fireworks, and Flash*. APress L. P. ISBN 1-59059-152-6.
2. Harris, Andy (2001): *JavaScript Programming for the Absolute Beginner*. Premier Press. ISBN 0-7615-3410-5.
3. Heinle, Nick; Koman, Richard (1997): *Designing with JavaScript*. O'Reilly & Associates. ISBN 1-56592-300-6.
4. Jamsa, K, King, K, Anderson, A. (2003): HTML i Web dizajn, Mikro knjiga, 2003. ISBN 86-7555-204-1
5. McDuffie, Tina Spain (2003): *JavaScript Concepts & Techniques: Programming Interactive Web Sites*. Franklin, Beedle & Associates. ISBN 1-887902-69-4.
6. McFarlane, Nigel (2003): *Rapid Application Development with Mozilla*. Prentice Hall Professional Technical References. ISBN 0-13-142343-6.
7. Powell, Thomas A.; Schneider, Fritz (2001): *JavaScript: The Complete Reference*. McGraw-Hill Companies. ISBN 0-07-219127-9.
8. Shelly, Gary B.; Cashman, Thomas J.; Dorin, William J.; Quasney, Jeffrey J. (2000): *JavaScript: Complete Concepts and Techniques*. Cambridge: Course Technology. ISBN 0-7895-6233-2.
9. Vander Veer, Emily A. (2004): *JavaScript For Dummies* (4th ed.). Wiley Pub. ISBN 0-7645-7659-3.
10. Watt, Andrew H.; Watt, Jonathan A.; Simon, Jinjer L. (2002): *Teach Yourself JavaScript in 21 Days*. Pearson Education. ISBN 0-672-32297-8.