

Sveučilište u Zagrebu  
**Fakultet elektrotehnike i računarstva**

Seminarski rad iz kolegija  
Ergonomija računalne i programske opreme

# **Uporaba dinamičkog HTML-a pri izradi web sučelja**

Zagreb, rujan 2004.

Autor: Josip Maslač

# Uvod

Od početaka masovnijeg korištenja interneta za izradu internet stranica koristio se jezik za označavanje HTML (Hypertext Markup Language). Njegova osnovna zadaća bila je da na jednostavan i pregledan način prikaže tekst i slike. Kako je vrijeme prolazilo tako su i zahtjevi za sadržajem sa više mogućnosti sve više rasli. Sukladno tome pojavljivale su se nove tehnologije, jezici i protokoli te razna razmišljanja o tome što sve internet stranice mogu i trebaju sadržavati. Pri tome se mnoge od tih tehnologija do danas nisu održale. U ovom seminaru govoriti ću o onim tehnologijama/jezicima koji se danas pretežno koriste pri izradi preglednih i funkcionalnih internet stranica.

## Dinamički HTML

Većina današnjih internet stranica su izrađene u većoj ili manjoj mjeri korištenjem dinamičkog HTML-a (u nastavku teksta - DHTML). Iako bi se po samom nazivu dalo zaključiti da je riječ samo o novijoj tj. naprednijoj verziji jezika HTML, DHTML predstavlja ipak nešto drugo. DHTML je samo oznaka da se pri izradi internet stranica koristi nekoliko tehnologija. To je zapravo ideja o tome koje (već postojeće) jezike i na koji način ih se treba/može koristiti pri izradi stranica. Ti jezici su:

- HTML
- CSS
- skriptni jezik: JavaScript, VBScript

Ovdje je važno napomenuti i Document object model (skraćeno DOM). DOM ne predstavlja neki konkretan jezik već je to model po kojemu se u HTML-u stranica gradi od objekata. On je bitan jer predstavlja poveznicu između HTML-a koji sadrži objekte i CSS-a i skriptnog jezika koji tim objektima na neki način upravljaju tj. dodjeljuju ili mijenjaju objektima u HTML-u neka svojstva.

## Mogućnosti DHTML-a

Ispravnom uporabom i kombinacijom jezika koji sačinjavaju DHTML pružaju se velike mogućnosti za manipuliranje stranicama. Spomenuti ću par najvažnijih (u zagradi je navedeno koji od jezika omogućava pojedinu opciju):

- interakcija s posjetiteljima stranice (JavaScript)
- mogućnost prilagodbe stranica ljudima s posebnim potrebama (CSS)
- mijenjanje izgleda stranice nakon učitavanja - "živi" elementi (JavaScript)
- lakše održavanje tj. izmjena izgleda stranica (CSS)

## Sadržaj ovog seminara

Kao što je već ukratko objašnjeno DHTML ne predstavlja konkretni jezik već označava uporabu više jezika za izradu internet stranica. Za svaki od tih jezika bi se mogao napisati poseban seminar (a neki su i **napisani**) pa ću zato u ovom seminaru dati samo kratki osvrt na mogućnosti i ulogu tih jezika za kreiranje DHTML internet stranica. Ono što ću u ovom seminaru pokušati prezentirati su mogućnosti korištenja DHTML-a te na koji način pravilno kombinirati gorespomenute jezike za dobijanje željenog rezultata - kvalitetnih internet stranica.

# HTML i DOM

## HTML

Povijest HTML-a započinje već krajem prve polovice 20-og stoljeća kada se razmišljalo na koji način skladištiti tekstualne i grafičke informacije te na koji način iste logički povezati. Ta i slične ideje (a i rješenja) su se desetljećima pojavljivali kao odgovor na razne probleme. Jedan od češćih problema koji je dosta povezan sa idejom HTML-a bio je: na koji način označiti tj. formatirati tekst i slike - rješenje tog problema primjenu je najčešće našlo za članke u novinama i knjige.

S proširenjem interneta početkom 90-ih i pojavom prvih interpretera internet stranica (među prvima Cello, Viola, MidasWWW) pojavio se i HTML. Od tada do danas HTML je doživio dosta promjena i verzija. U trenutku pisanja ovog seminara aktualna verzija je HTML 4.01 dok se polako proširuje i XHTML.

Glavna namjena HTML-a je određivanje (označavanje) koje informacije i na koji način će se one prikazati na internet stranicama. Treba napomenuti da HTML nije programski jezik već jednostavan jezik za označavanje. Sastoji se od jednostavnih označnih elemenata koji se nazivaju tag-ovi i unutar kojih se smještaju informacije. Način na koji će se pojedina informacija, bilo tekstualna ili grafička, prikazati ovisi o tag-u unutar kojega je smještena. Sama sintaksa HTML-a je dosta jednostavna a detalji se mogu pogledati na jednoj od stranica koja se nalazi među [linkovima](#). Također važno je napomenuti da je jedna od iznimno bitnih karakteristika HTML dokumenata mogućnost definiranja veze među stranicama.

Izgled pojednostavljenog HTML dokumenta dan je sljedećim primjerom:

```
<HTML>
<HEAD>
  <TITLE>Naslov dokumenta</TITLE>
</HEAD>
<BODY>
  <H1>Naslov</H1>
  <H2>Podnaslov</H2>
  <P>Text, text, text...</P>
  <a href="neka_stranica.html">sljedeća</a>
</BODY>
</HTML>
```

U počecima razvoja internet stranica HTML je služio sa ciljem, kako je navedeno, da se unutar njega smještaju informacije a ujedno i određuje na koji način će se te informacije prikazati. Danas se zbog razvoja ostalih (specifičnih) jezika teži da HTML ima malo drugačiju ulogu. Ideja je da HTML sadrži informacije no da pomoću raznih tag-ova pojedine elemente tih informacija (samo) označi, bez da se definira kako će se ti elementi prikazati. Način na koji će se ti elementi onda prikazati ili mijenjati pri interakciji sa posjetiteljem vrše se u drugim prije spomenutim jezicima: CSS-u i JavaScript-u. Ova ideja se pojavila kao rezultat težnji raslojavanja samih podataka i načina njihovog prikaza, i to ne samo za područje internet stranica već i klasičnih tekstualnih dokumenata, dokumenata namijenjenih za pregled mobilnim telefonima itd.

## DOM (Document object model)

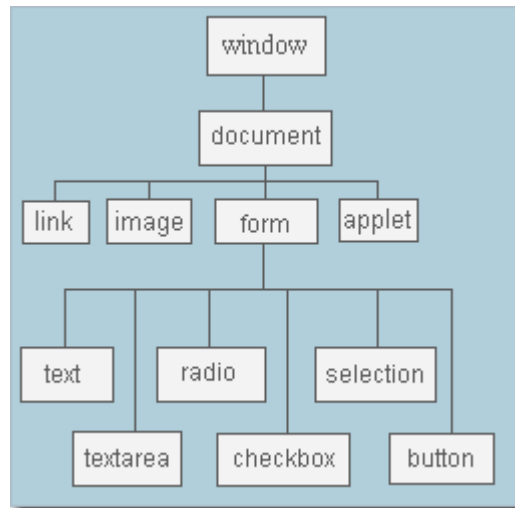
Kao što je rečeno u uvodu DOM predstavlja iznimno bitan faktor u DHTML-u. Na hrvatskom bi se DOM preveo kao objektni model dokumenta. Upravo se iz imena može vidjeti čemu on služi. Kao prvo riječ je o modelu tj. definiranju strukture HTML dokumenta. Prilikom definiranja strukture HTML dokumenta postavljaju se sljedeća pitanja:

- Koji sve tag-ovi postoje na stranici?

- Koliko ih ima?
- Na koji način su poredani?
- Koja su svojstva tih tag-ova?
- Na koji se način elementi tih tag-ova prikazuju na stranici?

DOM daje odgovor na ova pitanja. On izražava strukturu HTML dokumenta na univerzalan, sadržajno neovisan način.

Postoji više razina (level-a) DOM-a koji se međusobno razlikuju po objektima koje sadrže kao i mogućnostima mijenjanja svojstava i prikaza tih objekata. Primjer pojednostavljene sheme elemenata DOM-a dan je sljedećom slikom:



## Označavanje objekata u HTML-u

U zadnjem odsječku rečeno je da se HTML sastoji od objekata. Da bi ostala dva jezika DHTML-a tj. CSS i JavaScript mogli identificirati objekte kojima žele manipulirati potrebno je te objekte označiti. U HTML-u objekti se označavaju tako da se za HTML element koji se želi označiti eksplicitno definira ID atribut na sljedeći način:

```

<P id="naslov">
...sadržaj paragrafa naslov...
</P>
  
```

Ovako označenom HTML objektu se preko vrijednosti ID atributa lako pristupa u CSS-u i JavaScript-u. Jedino ograničenje je da se ne smije dodijeliti ista vrijednost ID atributa dvoma različitim objektima.

# CSS

Cascading style sheets - CSS se pojavio najkasnije u odnosu na ostala dva jezika koja se koriste u DHTML-u. Prva verzija je nastala 1996 godine a razlog nastajanja bile su slabe mogućnosti formatiranja i pozicioniranja sadržaja u HTML-u. CSS je donio puno dobrih stvari koje su značajno olakšale izradu internet stranica. Prije nego što spomenem prednosti CSS-a objasniti ću što to CSS uopće je. CSS je ništa drugo nego skup pravila/stilova kojima se definira kako će se pojedini elementi HTML-a prikazati. Prednosti koje CSS nudi su sljedeće:

- znatno veće mogućnosti određivanja svojstava HTML-a objekata
- mogućnost odvajanja sadržaja od prezentacije (donekle)
- izgled kompletnog web *site-a* se može promijeniti samo promjenom jedne CSS datoteke
- definiranje prezentacije ovisno o mediju (ekran, pisac, mobitel itd.)

## Kako CSS funkcionira

Način funkcioniranja CSS-a je taj da se za pojedine objekte HTML-a definiraju željena svojstva tj. pravila. Povezivanje HTML objekata i CSS stilova se vrši preko CSS selektora. Glavna uloga selektora je da izaberu(selektiraju) objekte iz HTML-a te za njih definiraju stil tj. vizualna svojstva koja će izabrani objekt posjedovati. Ima više vrsta selektora a ovdje ću navesti samo najčešće korištene:

- Selektori tipa - izabiru sve objekte koji su određenog tipa (točnije HTML tag-a)
- ID selektori - izabiru objekte na osnovi vrijednosti ID atributa
- Selektori klase - izabiru one objekte/tag-ove koji imaju atribut *class* jednak selektoru

Nakon što se pomoću selektora odrede objekti kojima se želi manipulirati potrebno je definirati željena svojstva tih objekata. Ona se definiraju pomoću deklaracijskog bloka koji sastoji od jedne ili više deklaracije. Svaka deklaracija sadrži dva dijela: oznaku svojstva te vrijednost tog svojstva. Shema jednog CSS pravila dana je sljedećom slikom:



Mjesto gdje se definiraju CSS pravila može biti:

- u samoj HTML datoteci koja koristi stilove
- u zasebnoj CSS datoteci

Ova dva načina moguće je kombinirati te je moguće imati i više CSS datoteka u kojima se definiraju stilovi. Izbor ovisi o odabiru autora no preporučljivo je stilove definirati u zasebnim CSS datotekama i to tako da je za svaki medij definirana jedan CSS datoteka.

## Odabir medija

Jedna od iznimno korisnih svojstava CSS-a je mogućnost definiranja izgleda na osnovi medija na kojem će se sadržaj stranice prikazivati. Primjeri medija su: zaslon ekrana, pisac, zaslon mobitela itd. Za svaki medij mogu se definirati posebni stilovi koji se onda primjenjuju samo ako se stranica pregledava s određenim medijem.

## Primjer korištenja

```
// HTML datoteka
..
<P>paragraf</P>
<A href="link.html" class="klasa1">link</A>
<DIV id="sloj1"></DIV>
..

// CSS datoteka
..
P{          /* selektira sve P tag-ove */
  font-size: 12pt;
}
.klasa1{   /* selektira sve tag-ove klase 'klasa1' */
  color: #ff0000;
}
#sloj1{    /* selektira objekt sa id-em 'sloj1' */
  width: 350px;
}
..
```

# JavaScript

Iz imena DHTML često se krivo tumači što u tom imenu označava atribut *dinamički*. Zbog tog atributa najčešće se očekuje da stranica napravljena uporabom DHTML-a ima neku vrstu animacije, pokretnih objekata, promjenjivih elemenata, jednom riječju očekuje se interaktivna stranica. To najčešće nije slučaj. No uporabom DHTML-a mogu se napraviti spomenute interaktivne stranice. Jezik koji to omogućava naziva se JavaScript. Osim JavaScript-a za istu namjenu može se koristiti i VBScript no on se u praksi iznimno rijetko koristi.

## Što je JavaScript

JavaScript je skriptni jezik koji služi za upravljanje/izmjenu HTML dokumenta za vrijeme pregledavanja tog dokumenta. Navest ću par osnovnih karakteristika JavaScript-a:

- JavaScript se izvršava u internet pregledniku posjetitelja
- Sintaksa je slična programskom jeziku C
- Sam programski kod može se smjestiti u samom HTML dokumentu ili zasebnoj datoteci

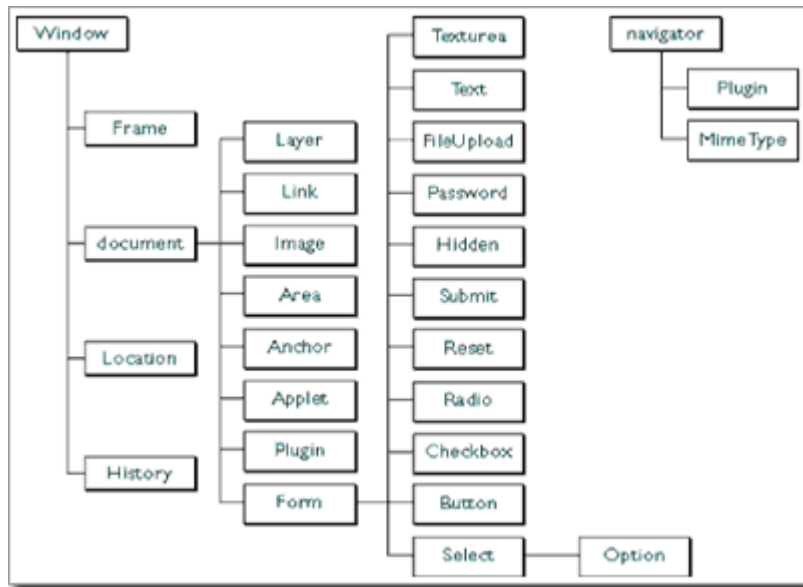
Jedan od bitnijih dijelova JavaScripta je podrška događajima (eng. events). Događaji na stranici se dešavaju kao rezultat neke akcije posjetitelja - klik mišem, prelazak mišem preko nekog objekta, učitavanje dokumenta i sl. Iznimno su korisni jer se onda može definirati što će se dogoditi ukoliko posjetitelj vrši određene radnje. Na taj način posjetitelj dobija osjećaj interakcije tj. osjećaj da su stranice 'žive'.

## Načini korištenja

Kao što je spomenuto JavaScript kod se može smjestiti u samom HTML dokumentu ili u jednoj ili više zasebnih datoteka. Ukoliko se smještava u datoteku ta datoteka bi trebala imati ekstenziju '.js' i sam kod se bez posebnih dodatnih oznaka smjesti u datoteku. Ukoliko se JavaScript kod smještava u HTML to je moguće izvršiti na dva načina:

- Unutar HTML tag-a `<script> </script>`
- Preko vrijednosti atributa događaja nekog HTML objekta

Objektima u HTML-u se u JavaScript-u pristupa preko ID atributa. Svim objektima u HTML-u se također može pristupiti preko globalnih polja koja se implicitno definiraju za svaki HTML dokument. Shema tih polja dana je sljedećom slikom:



O detaljima JavaScript-a i njegovim mogućnostima ovdje neću govoriti iz razloga što je to malo veća tema tj. tema za poseban seminar. Više informacija se može pronaći na stranicama koje se nalaze među [linkovima](#).

## Problemi

Problema u vezi korištenja JavaScripta ima više. U praksi su oni najčešće vezani uz to da razni internet preglednici ne podržavaju sve mogućnosti JavaScripta, ili ih ne podržavaju na standardom definiran način.

No veći problem od toga je mogućnost da neki internet preglednici uopće ne podržavaju JavaScript ili da je JavaScript onespособljen od strane posjetitelja. Ovaj problem se nažalost za sad nemože riješiti niti na jedan način.



# Primjeri

U ovom poglavlju navesti ću par primjera kojima ću probati ilustrirati različite mogućnosti koje DHTML pruža.

## Prilagodba izgleda stranica pristupnom mediju

Kao što je u poglavlju o CSS-u objašnjeno glavna uloga CSS-a je definiranje vizualnih svojstava stranica. Jedna od iznimno korisnih opcija DHTML-a je mogućnost definiranja izgleda stranice ovisno o mediju preko kojega se stranica pregledava. Primjeri medija su: zaslon monitora, izlaz na pisač, zaslon mobitela itd. Način na koji se ova mogućnost ostvaruje je da se jednostavno u HTML dokumentu definira koje stilove treba koristiti za željene medije:

```
<HEAD>
..
<LINK rel="stylesheet" href="stil.css" type="text/css" media="all">
<LINK rel="stylesheet" href="print.css" type="text/css" media="print">
..
</HEAD>
```

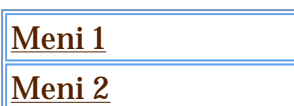
Gore je naveden konkretni primjer koji se koristi na ovim stranicama - ova stranica je posebno prilagođena za ispis na pisač. Učinci se mogu vidjeti ispisom ovih stranica na pisač (ili koristeći *print preview* opciju u *File* izborniku).

## Kontrola unosa web forme

Jedan od najčešćih načina korištenja DHTML-a (preciznije JavaScript-a) a ujedno i najkorisniji je kontrola ispunjavanja web formi. Naime prilikom ispunjavanja web formi često valja provjeriti ispravnost unešenih podataka. To se može napraviti na način da korisnik ispuni formu, pošalje je i ukoliko je korisnik nešto krivo ispunio na sljedećoj stranici se ispiše poruka o pogrešnom unosu. Nedostatak ovog pristupa je vrijeme - korisnik mora poslati podatke te onda pričekati određeno vrijeme da se učita nova stranica. Ovim pristupom ukoliko je posjetitelj sve ispravno unio ne gubi se na vremenu (jer se u svakom slučaju podaci forme moraju poslati te se učitati nova stranica). No ukoliko je posjetitelj nešto krivo ispunio učinkovitiji način je da se odmah po prihvaćanju forme obavijesti posjetitelja na krivo ispunjena polja te bez učitavanja nove stranice omogući mu se ispravak. Primjer ove mogućnosti ilustrirati ću skriptom za [validaciju email-a](#).

## Izrada dinamičkih menija

Meniji su jedan od najbitnijih dijelova svih stranica jer omogućavaju jednostavan pristup sadržajima koje stranice nude. Koristeći CSS i JavaScript moguće je izraditi iznimno efikasne i oku ugodne menije. U nastavku ću prikazati primjer jednostavnog menija koji je izrađen uporabom svih DHTML jezika:



Za izradu vizualno atraktivnih i funkcionalnih menija preporučio bih program Sothink DHTML Menu koji se nalazi među [linkovima](#).

## Definiranje više stilova za jednu stranicu

Kao što je više puta naglašeno ideja DHTML-a (zapravo samog HTML-a) je da se u HTML-u samo označe elementi koje će stranica sadržavati a potom se u CSS može definirati na koji način će se vizualno ti elementi prikazati. Iz toga proizlaze dvije korisne mogućnosti:

- Mogućnost mijenjanja kompletnog izgleda *site-a* mijenjanjem samo stilova tj. CSS datoteka. Odličan primjer je stranica [CSS Zen Garden](#).
- Mogućnost definiranja više stilova te omogućavanje korisniku da odabere stil koji mu najviše odgovara. Dobar primjer ove mogućnosti se može pogledati u [seminaru](#) Borisa Kuzmića.

# Problemi

Prilikom izrade DHTML internet stranica pojavljuje se nažalost puno (previše) problema. Glavni uzrok tih problema nenalazi se u jezicima koji se koriste pri izradi već su u pitanju internet preglednici. Iako su svi jezici standardizirani i način njihovog korištenja je jasno definiran, internet preglednici što zbog jednostavno loše izvedbe a što zbog namjernog nepridržavanja navedenih standarda često ne prikazuju ispravno razne internet stranice. Glavni razlog nepridržavanja službenih standarda je želja za vlastitim standardiziranjem protokola. U prošlosti a u velikoj mjeri još i danas najveću zastupljenost u korištenju među korisnicima ima Microsoftov internet preglednik Internet Explorer. On ujedno i prednjači u nepridržavanju službenih protokola. To je zapravo uzrok većine problema. No polako se pojavljuju drugi preglednici koji zbog inovativnih mogućnosti koje nude, privlače znatan broj korisnika. Iznimno bitno je to što se većina njih pridržava službenih standarda te time polako prisiljavaju da to čini i Internet Explorer.

## Konkretni problemi

Problemi koji su vezani uz razvoj DHTML stranica vezani su uglavnom za CSS i JavaScript. HTML kao jezik postoji već dosta dugo i tijekom vremena nije se značajno mijenjao tako da njega preglednici (većinom) ispravno prikazuju.

JavaScript kao drugi najstariji jezik iako se tijekom godina javljao u raznim verzijama (s tim da se u novijim verzijama radilo samo o nadogradnji) nije mijenjao svoj osnovni koncept i funkcionalnost pa tako da iako je on nailazio na najviše 'prepreka' kod različitih preglednika u prošlosti, danas je uglavnom dobro podržan od svih poznatijih internet preglednika. Problemi koji se javljaju kod JavaScript-a i raznih preglednika su uglavnom u razlikama pristupa HTML objektima te pristupu/izmjeni svojstava HTML objekata. Ti problemi često se rješavaju na način da se prvo provjeri sa kojim preglednikom se stranica pregledava i onda ovisno o rezultatu se izvodi kod koji je ispravan za dobijeni preglednik. Primjer jednog od najčešće korištenih pristupa je sljedeći:

```
function getObject(objectId) {
    if (document.all && !document.getElementById){// Netscape
        return document.all(objectId);
    } else{
        return document.getElementById(objectId); // ostali preglednici
    }
}
```

Zadatak navedenog primjera tj. funkcije *getObject()* je da vrati pokazivac na željeni objekt. Ovisno o internet pregledniku to se izvodi na jedan od dva gorenavedena načina.

Najmlađi od jezika koji se koriste u DHTML-u CSS sa sobom donosi i najviše problema. Najveći problemi su u različitom shvaćanju raznih preglednika na koji način treba tumačiti CSS pravila. Ovo je donekle paradoks jer je u službenom definiranju CSS-a dosta jasno prikazano što pojedina pravila znače i na koji način ih treba vizualno primjeniti. Rješavanje ovog problema u potpunosti je ostavljeno autoru stranica a najčešće se svodi na to da autor pregledava svoje stranice u najpopularnijim preglednicima (Internet explorer, Mozilla Firefox, Opera) te pokuša raznim 'trikovima' postići željeni izgled.

# Savjeti

Prilikom izrade stranica da bi iste bile kvalitetne i učinkovite valja se držati par osnovnih pravila/savjeta:

- **Pridržavanje službenih standarda**

Osim što je ovo naravno imperativ sam po sebi, dovodi do toga da je velika vjerojatnost da će se stranice (uglavnom) ispravno prikazivati u većini internet preglednika. Nakon izrade stranica pridržavanje standardima u slučaju HTML-a i CSS-a može se provjeriti validatorima dotičnih jezika. Adrese na neke od njih se nalaze među [linkovima](#). Za JavaScript nema nekog službenog rješenja već se najčešće ispravnost koda provjerava samim pregledom stranice u internet pregledniku. Stanje u vezi toga se malo popravljja jer se u nekim preglednici novijeg datuma nude dodaci kojima se izvršavanje JavaScript-a može malo detaljnije *nadgledati* - pa tako preglednici Mozilla Firefox i Opera nude *JavaScript Debug Console* dodatak koji pruža malo više detalja o eventualnim greškama u JavaScript kodu.

- **Preglednost i funkcionalnost**

Prilikom izrade stranice ovo su jedni od najvažnijih pravila na koja treba misliti. Razlog zbog kojega se stranice rade i zbog čega privlače korisnike je njihov sadržaj. Zato je bitno prije same izrade samih stranica dobro razmisliti o načinu na koji će se sadržaj organizirati i prezentirati kako stranice unatoč kvalitetnom sadržaju ne bi zbog loše izvedbe postale neupotrebljive.

- **Doziranje efekata**

Mogućnosti koje se nude uporabom DHTML-a su velike no iste ne treba krivo niti previše koristiti. Naime često (najčešće neiskusni) autori koriste neke efekte i mogućnosti DHTML-a bez da su promislili uklapaju li se oni u samu stranicu i koja je njihova svrha na stranici. Savjet s tim u vezi je da treba koristiti samo one mogućnosti koji su u funkciji stranice a nikako dodavati efekte koji su sami sebi svrha. Primjeri toga su: (animirana) slova koja prate kursor miša, ispis vremena provedenog na stranici, korištenje više od jednog dijela teksta koji se pomiče (eng. scroll text), ispis trenutnog datuma i sl.

- **Provjera izgleda u drugim preglednicima**

Unatoč činjenici da je još uvijek Microsoftov preglednik Internet Explorer daleko najkorišteniji internet preglednik svakako je preporučljivo provjeriti kako se stranice prikazuju i u drugim (korištenim) internet preglednicima te naravno pokušati ispraviti stranice ukoliko se iste ne pokazuju ispravno u tim preglednicima.

- **Obavijest o onesposobljavanju JavaScript-a**

Kao što je rečeno u poglavlju o JavaScript-u korisnicima je u preglednicima dozvoljena mogućnost onesposobljavanja JavaScript-a. Ukoliko na stranicama JavaScript predstavlja bitan faktor za pregled stranica (npr. preko njega se obavlja navigacija), valja korisnike koje imaju onesposobljen JavaScript o tome obavijestiti. To se može napraviti korištenjem <noscript> tag-a npr.:

```
<NOSCRIPT>
Zbog onemogućavanja JavaScript-a u Vašem pregledniku
nećete moći pristupiti svim dijelovima ovih stranica!
</NOSCRIPT>
```

Tekst koji se nalazi u <noscript> tag-u biti će prikazan samo u preglednicima u kojima je JavaScript onemogućen.

# Linkovi

## HTML

- <http://www.w3.org/TR/html401>  
Službena specifikacija HTML 4.01
- <http://www.htmlref.com>  
Specifikacije HTML-a, XHTML-a i nešto CSS-a
- <http://validator.w3.org/>  
HTML validator

## CSS

- <http://www.w3.org/TR/css21>  
Službena specifikacija CSS 2.1
- <http://www.csszengarden.com>  
Odličan primjer pravilnog korištenja/kombiniranja CSS-a i HTML-a
- <http://css.maxdesign.com.au/index.htm>  
Kvalitetni primjeri korištenja CSS-a pri izradi lista i menija

## JavaScript

- <http://www.javascript.com>  
Stranica sa puno sadržaja vezanih uz JavaScript
- <http://www.w3schools.com/js/default.asp>  
Tutorial o JavaScript-u

## Ostalo

- <http://www.sothink.com/product.htm>  
Odlični programi za kreiranje DHTML stranica.
- [http://www.westciv.com/style\\_master/index.html](http://www.westciv.com/style_master/index.html)  
Iznimno koristan program za kreiranje CSS stilova

# O autoru

Moje ime je Josip Maslač i student sam na Fakultetu elektrotehnike i računarstva.

Bavim se izradom uglavnom dinamičkih internet stranica pri čemu se osim jezika navedenih u ovom seminaru koristim Php-om i MySql-om.

Email putem kojega me možete kontaktirati je: [josip.maslac@fer.hr](mailto:josip.maslac@fer.hr)