

SVETOVNI SPLET IN HTML

Šola jezika HTML začnemo s skokom v zgodovino in pregledom nekaj osnovnih pojmov.

ZGODOVINA SVETOVNEGA SPLETA

Za začetek štejemo leto 1989, ko je Timothy Berners-Lee v Evropskem laboratoriju za fiziko jedrskih delcev (CERN) predlagal standarde, ki so pomenili nastanek svetovnega spleta (WWW – World Wide Web).

V začetku je za iskanje in izmenjavo podatkov prek spleta obstajal le en spletni odjemalec – Mosaic, ki so ga razvili v Nacionalnem centru za superračunalniške aplikacije (NCSA). Zdaj je na voljo množica pregledovalnikov, med njimi pa sta najbolj znana in razširjena Microsoft Internet Explorer in Netscape Communicator.

Bistvo priljubljenosti in uporabnosti svetovnega spleta je hiperbesedilo. To je nekakšno vezivo dokumentov in podatkov, ki so med seboj fizično ločeni, uredi pa jih v uporabniku prijazno, preprosto in dostopno obliko. Splet je bil ena prvih aplikacij, ki je dala tudi nekim uporabnikom možnost, da iščejo in pridobivajo velike količine podatkov, ki so bili že tedaj na voljo v internetu.

Vsekakor se ne moremo izogniti elektronski pošti. Svetovni splet je uspešno uporabil že obstoječi protokol za vključevanje različnih tipov podatkov v elektronska sporočila (MIME – Multipurpose Internet Mail Extensions). Poleg elektronske pošte so v svetovni splet vključili tudi protokol za prenos datotek (FTP – File Transfer Protocol) in protokol za oddaljeno prijavljanje Telnet. Nove možnosti, ki jih je splet ponudil skupaj s kombinacijo protokolov za nadzor oddajanja internetnega protokola (TCP/IP) so spodbudile razvoj novih jezikov in standardov.

Uporabniku spleta so najbolj znani spletni naslovi (Uniform Resource Locator – URL). URL je precej več kot le naslov, saj je v marsičem podoben imenu datoteke, ki vsebuje tudi podatke o tem, v katerem strežniku je datoteka in kateri protokol lahko uporabimo za dostop do vira informacije.

ZGLED SPLETNEGA NASLOVA

Spletni naslov je običajno v naslednji obliki:

```
<protokol>://<ime-strežnika>/<pot>
http://www.delo-revije.si/slo/moj_mikro/index.htm
```

Iz zgleda vidimo, da je uporabljen protokol http, ime strežnika je www.delo-revije.si, pot je slo/moj_mikro, datoteka, v kateri je zapisana vsebina, pa je index.htm

Najpogostejši protokoli, ki jih lahko navedemo v spletnih naslovih, so podani v naslednji tabeli:

Protokol	Opis
HTTP	protokol za prenos hiperteksta (HyperText Transfer Protocol)
HTTPS	varni protokol za prenos hiperteksta (HyperText Transfer Protocol, šifriranje prek standarda Secure Sockets Layer -SSL)
mailto	pošiljanje elektronske pošte
FTP	protokol za prenos datotek
gopher	informacijski strežnik, podoben svetovnem spletu, vendar v čisti besedilni obliki
news	novice v omrežju USENET
file	krajevna datoteka
telnet	oddaljeno prijavljanje

Tabela: Najpogostejši protokoli, ki jih lahko navedemo v spletnih naslovih.

OSNOVNE ZNAČILNOSTI PROTOKOLA HTTP

Protokol za prenos hiperbesedila (HyperText Transfer Protocol – HTTP) je glavni način prenosa podatkov med strežniki in odjemalci v svetovnem spletu. Pred njim je bil protokol za prenos datotek FTP glavni način izmenjave podatkov med oddaljenimi računalniki.

HTTP je preprost in zelo hiter besedilni protokol, ki so ga razvili za hiperbesedilno in hipermedijsko okolje svetovnega spleta. Za razliko od protokola za prenos datotek ima na voljo le nekaj ukazov in postopkov. Z večnamenskimi dodatki za elektronsko pošto (MIME) omogoča vključevanje različnih formatov, predvsem grafičnih in multimedijskih dodatkov, v hiperbesedilne dokumente.

Protokol deluje po načelu odjemalec-strežnik. Odjemalec HTTP, imenovan tudi uporabniški agent (User Agent), je najpogostejše spletni pregledovalnik, ki se priključi v strežnik HTTP in s spletnim naslovom zahteva od strežnika določen dokument ali drug vir podatkov. Strežnik mu, če je bila zahteva upravičena in jo je mogoče izvršiti, vrne zahtevano informacijo, pri čemer vključi podatke v odgovor z večnamenskimi dodatki za elektronsko pošto (MIME). Načeloma poteka komunikacija med strežnikom in odjemalcem povsem enako kot pri izmenjavi elektronske pošte, le nekoliko hitreje.

V tabeli je na drugem mestu zapisan protokol HTTPS. SSL (Secure Sockets Layer) je metoda kodiranja podatkov, ki jo je razvilo podjetje Netscape Communications posebej za uporabo prek protokola za nadzor prenosa (TCP/IP). SSL omogoča varovanje podatkov v sporočilih, ki se prenašajo s protokolom za prenos hiperbesedila.

Pri uporabi metode SSL odjemalci v spletnem naslovu zahtevajo uporabo protokola HTTPS in ne HTTP, na kar jih, ob uspešnem postopku avtorizacije, običajno opozori tudi sam spletni pregledovalnik. SSL je dandanes široko uporabljan način varovanja podatkov. Tisti, ki uporabljate bančne storitve prek SKB Neta ali Klika NLB, ta protokol gotovo poznate. Osnovne pojme svetovnega spleta smo tako spoznali, zdaj pa se lahko lotimo jezika za označevanje hiperbesedila.

Jezik za označevanje hiperbesedila

Jezik za označevanje hiperbesedila (HTML – HyperText Markup Language) je zapis podatkov, ki omogoča označevanje, oblikovanje in povezovanje podatkov v svetovnem spletu. Spletne strani, pisane v jeziku HTML, omogočajo vključevanje besedila, grafike in povezav kot kazalcev do drugih spletnih strani. Bistvena lastnost jezika za označevanje hiperbesedila naj bi bila čim večja neodvisnost od strojne in programske opreme odjemalca. S tem povezana prenosljivost naj bi znižala stroške in čas razvoja hiperbesedilnih dokumentov pri ponudnikih informacij in poenotila način uporabe in prikaza pri odjemalcih teh informacij. Tako naj bi vse naprave, sposobne sprejemanja informacij iz svetovnega spleta, od osebnih računalnikov z najrazličnejšimi grafičnimi vmesniki do delovnih postaj, mobilnih telefonov, prenosnih računalnikov in najrazličnejših drugih naprav, uporabljale enak način zapisa podatkov in povezav med njimi.

Najpogostejše so dokumenti, pisani v tem jeziku, po svoji vsebini statični, torej pripravljeni vnaprej in so v spletnih strežnikih v obliki datotek. Vendar pa je mogoče hiperbesedilne dokumente ustvariti tudi na zahtevo oziroma glede na določene pogoje in okoliščine, ali pa iz podatkov, ki jih zajamemo iz podatkovnih zbirk ob izvrstitvi zahteve.

Jezik HTML je, formalno gledano, le poenostavljena različica standardnega posplošenega jezika za označevanje (SGML – Standard Generalized Markup Language). Je pa vseeno dovolj močan in prilagodljiv, da ga je mogoče uporabljati za opis velike večine dokumentov, ki jih srečamo v svetovnem spletu.

HTML se je izredno hitro spreminjal. Njegova

šola jezika HTML (1. del)

začetna različica z 1.0 je bila v bistvu standard, ki ga je vnesel prvi spletni pregledovalnik Mosaic. Zdaj smo pri različici številka štiri, ki jo je decembra 1997 objavilo združenje W3C (World Wide Web Consortium), ki med drugim skrbi tudi za razvoj standardov na področju jezikov za označevanje hiperbesedila. Vendar je HTML 4.0 le standard na papirju, saj ga nobeden od glavnih izdelovalcev spletnih pregledovalnikov ne podpira v celoti.

V primerjavi s prejšnjimi različicami naša četrta generacija jezika HTML nekaj novosti in opaznih izboljšanj. Standardizirani so slogi pisav, kar omogoča poenotenje in poenostavitev zapisa, oblike in lastnosti hiperbesedila. Podprti so večjezični dokumenti, precej je izboljšav pri uporabi okvirjev in tabel v pregledovalnikih. Podprti so tudi matematični znaki, Braillova pisava in govor. Najpomembnejša novost jezika HTML je dinamično spreminjanje vsebine sple-



tnih dokumentov z dogodki, kar so dotlej omogočali le skriptni jeziki, kot sta JavaScript in VBScript. To je omogočilo, da je HTML postal resnični programski jezik.

V naslednji številki Mojega mikra bomo začeli z osnovami, zato zdaj posvetimo še nekaj besed urejevalnikom kode HTML. V spletu lahko najdemo precej urejevalnikov vrste freeware in shareware. Večinoma so besedilni z dodatnimi meniji in gumbi, s katerimi v besedilo vstavljamo oznake (tags) HTML. Dodatni meniji in gumbi nam pridejo prav iz dveh razlogov. Oznak nam ni treba tipkati in tudi zapomniti si nam jih ni treba, kar velja še zlasti za sintakso. Med množico urejevalnikov pa najdemo tudi take, ki podpirajo WYSIWYG, kar pomeni »kar vidiš, to dobiš« (What You See Is What You Get). Uporaba teh urejevalnikov je zelo priporočljiva, saj je delo hitrejša in preprostejša. Ta podpora pa ne velja vedno in povsod. Predvsem zadnji del,

WYG, je precej odvisen od spletnega urejevalnika in tega se moramo zavedati. Večina zgledov, ki jih bomo obravnavali, bodo takšni, da boste lahko uporabljali poljuben urejevalnik besedil.

V nadaljevanju šole HTML bomo začeli z osnovami jezika HTML in njegovo strukturo. Spoznali bomo nekaj osnovnih oznak, dotaknili pa se bomo tudi povezav, s katerimi se preusmerimo iz enega dokumenta v drugega. ■



za pisarno

ne potrebujete
pohištva

Izbirnik aplikacij z moduli

Slovenski meni

handspring™
visor™

• internet • e-mail • organizator
• WWW • Word • Excel

RAM 2

Ram 2, Janus Trade, Auto-phone,
Anni, Coming, DZS, E.Lectem,
Cankarjeva založba,
Mladinska knjiga - Trgovina,
Mladinska knjiga - Birooprema,
Grafika Soča, Big Bang Mega,
Nabatore, Rolan, Miron



Išči pametneje.
www.najdi.si

OSNOVNE OZNAKE

V prvem delu smo razjasnili nekaj osnovnih pojmov v zvezi s svetovnim spletom, zdaj pa si ogledjmo nekaj osnovnih oznak (tags).

Vsako HTML stran določajo tri osnovne oznake, ki pomagajo brskalniku prepoznati stran HTML. Te tri oznake so `<html>`, `<head>` in `<body>`. Te oznake nimajo nikakršnega vpliva na obliko in prikaz strani HTML.

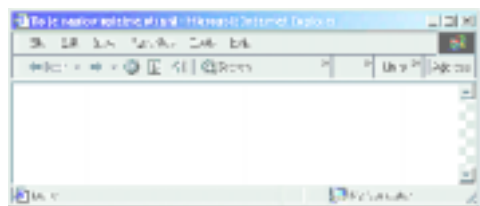
Z oznako `<html>` se začne in konča `</html>` vsaka stran HTML. Oznaka nam pove, da je vsebina datoteke zapisana v jeziku HTML. Običajno pa pred to oznako kot priporočilo konzorcija W3C za standard XHTML 1.0 velja, da napišemo še, za katero različico jezika HTML gre.

Naslednja oznaka je `<head>`, ki določa začetno in končno točko uvoda v dokument. Sem običajno vpišemo oznako `<title>`, ki nosi naslov dokumenta. Nikoli pa ne smemo vpisovati vsebine spletne strani v glavo dokumenta.

Tretja oznaka je `<body>`, znotraj katere je celotna vsebina spletne strani.

Tako je prazna stran z osnovnimi oznakami videti takole:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
http://www.w3.org/TR/xhtml1/transitional.dtd>
<html>
<head>
<title> To je naslov spletne strani </title>
</head>
<body>
...
</body>
</html>
```



Zgornji zgled nakazuje, da so oznake vedno v parih. Začetno in zaključno oznako označimo s poševno črto. Oznake se tudi gnezdijo, kar pomeni, da sta npr. obe oznaki `body` znotraj oznake `html` in podobno velja za vse ostale oznake, ki jih bomo v prihodnje še srečali. Nikoli pa ne smemo oznak prekrivati.

Oglejmo si zdaj nekaj najosnovnejših oznak.

Oznaka `<title>` rabi za določitev naslova strani. Po teh naslovih brskalniki iščejo strani v internetu, hkrati pa je ta naslov prikazan tudi v naslovni vrstici brskalnika. Oznako moramo zapisati v glavo dokumenta med oznaki `<head>`.

Priporočljivo je, da so naslovi strani jasni in nedvoumni, saj bomo tako že iz naslova razbrali, o čem govori vsebina.

Headings naslovi

Naslovi so namenjeni za razdelitev vsebine na razdelke. HTML ima v osnovi šest ravni, h1-h6.

Paragraphs – odstavki

Oznaka `<p>` določa odstavke znotraj besedila. Tudi to oznako je treba zaključiti `</p>`. Starejše različice HTML-ja tega niso zahtevale, vendar je zaradi sistematike in s tem manjše zmešnjave novost dobrodošla.

Lists – naštevanja

Omenimo še oznake za naštevanje. HTML 4.0 določa tri tipe naštevanj. Pri oštevilčenem naštevanju so posamezne alineje oštevilčene po vrstnem redu, pri naštevanju s simbolnimi oznakami imajo posamezne alineje na začetku simbol, ki je običajno pika. Tretja vrsta naštevanj je podobna slovarju, saj je najprej beseda ali termin, sledi pa razlaga, ki se lahko razprostira čez več vrstic.

```
<p> Tri vrste naštevanj </p>
<ol>
<li> Oštevilčeno naštevanje. </li>
</ol>
```



```
<p> Tri vrste naštevanj </p>
<ul>
<li> Naštevane s piko na začetku. </li>
</ul>
```



```
<p> Tri vrste naštevanj </p>
<dl>
<dt> Pojem </dt>
<dd> Razlaga pojma. </dd>
</dl>
```

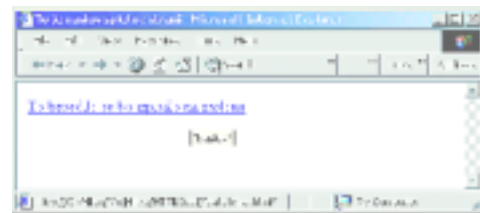
Na kratko smo si ogledali nekaj najosnovnejših oznak, s katerimi že lahko naredimo stran HTML, kjer je besedilo razdeljeno po odstavkih in ustrezno naštetu. Velika prednost interneta pa so povezave. Oglejmo si torej oznako, ki nam to omogoča.

POVEZAVE

Za povezavo potrebujemo v osnovi dvojico: datoteko ali naslov URL do datoteke, kamor želimo narediti povezavo, in besedilo, ki nam bo odskočna deska do datoteke. Besedilo je običajno posebej označeno glede na drugo besedilo v brskalniku in se nam ponuja, da ga kliknemo.

Za izdelavo povezav znotraj dokumentov HTML uporabljamo oznako `<a>`, ki se uporablja tudi za ustvarjanje sider (anchor) pri povezavah, kar si bomo ogledali nekoliko pozneje. Za razliko od oznak, ki smo si jih doslej ogledali, je oznaka `<a>` malce kompleksnejša, saj vsebuje še dodatne attribute. V splošnem povezave HTML opišemo takole:

```
<a name=ime href=datoteka.html title=Naslov>
To besedilo se bo izpisalo na zaslonu </a>
```



Oznaka `name` rabi za poimenovanje povezave, `datoteka.html` pa je tista datoteka, ki se naloži, ko izberemo povezavo. V `title` napišemo besedilo, ki se bo izpisalo, ko smo z miško na povezavi. Pri povezavah moramo paziti predvsem na dvojico. HTML ni občutljiv na velike in male črke, vendar moramo pri imenu datoteke paziti, da jo napišemo tako, kot je shranjena v operacijskem sistemu. Pozorni moramo biti tudi pri ustvarjanju relativnih in absolutnih povezav. Kaj to pomeni? Pri izdelavi posamezne povezave moramo navesti pot do datoteke, za katero želimo, da se izvede ob kliku na povezavo. To pot pa lahko navedemo v absolutni ali relativni obliki. Absolutna oblika je manj primerna, saj v tem primeru izgubimo prenosljivost, po drugi strani pa je ta oblika enostavnejša za uporabo. Če pozneje iz kakršnegakoli razloga datoteke premikamo po trdem disku, se bodo povezave prekinile, kar seveda pomeni, da

ne bodo delovale. V teh primerih si pomagamo s spremenljivkami v posameznih datotekah, pot v absolutni obliki pa navedemo le na enem mestu. Popravljanje je tako precej lažje. Vse to pa odpade, če uporabljamo relativne povezave. Tabeli prikazujeta različne oblike povezav.

Zapis relativne povezave HTML	Pomen
<code>href="datoteka.html"</code>	datoteka.html je v trenutni mapi
<code>href="mapa/datoteka.html"</code>	datoteka.html je v mapi mapa, ki je v trenutni mapi
<code>href="mapa1/mapa2/datoteka.html"</code>	datoteka.html je v mapi mapa1, ki je v mapi mapa2 v trenutni mapi
<code>href="../datoteka.html"</code>	datoteka.html je za raven višje od trenutne mape
<code>href="../../datoteka.html"</code>	datoteka.html je za dve ravni višje od tekoče mape

Tabela 1: Relativne povezave

Zapis absolutne povezave HTML	Pomen
<code>href="/mapa1/mapa2/datoteka.html"</code>	datoteka.html je v mapi /mapa1/mapa2
<code>href="/d:/mapa/datoteka.html"</code>	datoteka.html je na disku D v imeniku mapa

Tabela 2: Absolutne povezave

Za ustvarjanje povezav do dokumentov v spletu uporabimo isto sintakso, kot smo jo opisali zgoraj. Razlika je le v poti, kjer v tem primeru navedemo celoten naslov URL do dokumenta, kot kaže zgled:

```
<a href=http://.http://www.mojmikro.delo-revije.si/index.html title=Naslov> Domaca stran mojega mikra </a>
```

Zelo uporabne so povezave do določenega mesta znotraj dokumenta. Običajna povezava nam namreč vedno odpre stran na začetku. Če so strani daljše, postane iskanje posameznega podatka znotraj dokumenta precej zamudno. V ta namen uporabimo povezave, ki jih naredimo s sidrom. Povezavo naredimo podobno, kot smo doslej že vajeni, s pomočjo oznake <a>. Na tem mestu postane tudi jasno, zakaj se za oznako <a> (anchor – sidro) ne uporablja npr. <l> (link – povezava). Še enkrat si oglejmo sintakso, s katero ustvarimo povezavo. Razdeljena je na dva dela, in sicer href atribut in besedilo, ki rabi za klik na povezavo. Sidro določimo na enak način, vendar namesto atributa href uporabimo atribut name, ki vsebuje ime sidra.

```
<a name=ime_sidra>To besedilo bo na vrhu zaslona</a>
```

Povezavo naredimo z naslednjo sintakso:

```
<a href=datoteka.html#ime_sidra>Vsebinska</a>
```

Kot vidimo iz sintakse, se običajni povezavi doda še #ime_sidra. Brskalnik odpre datoteko.html in naloži stran, ki se odpre na mestu, ki je določeno s sidrom.

Zaenkrat bomo končali z povezavami. Oglejmo si še ukaze za oblikovanje besedila znotraj dokumentov HTML. Z začetek si oglejmo znakovne sloge, ki jih delimo na logične in fizične.

Logični slogi določajo, kakšna bo uporaba označenega besedila, in ne, kako bo to besedilo izpisano na zaslonu. Sem sodijo poudarki (emphasis), navajanja (citation) in podobno. V tabeli 3 so našteje najpogostejše oznake, ki sodijo v skupino logičnih slogov.

	Znaki znotraj oznake bodo poudarjeni. V grafičnih pregledovalnikih to običajno pomeni ležečo pisavo (italic)
	Besedilo s to oznako je še bolj poudarjeno kot pri . Največkrat to pomeni krepko <bold>.
<code>	S to oznako zapisujemo programsko kodo. Videz je podoben pisavi Courier, pri čemer je metrika vseh znakov enaka.
<samp>	Podobno kot oznaka <code>. Namenjena je primeru besedila (sample).
<kbd>	Besedilo, ki ga vpišemo prek tipkovnice
<var>	Ime spremenljivke, ki zamenja neko realno vrednost
<dfn>	Definicija
<cite>	Navajanje

Tabela 3: Logični znakovni slogi

Fizični slogi določajo natančno obliko oblikovanja znakov. Tabela 4 pojasnjuje najpogostejše fizične znakovne sloge.

	Krepko (bold)
<i>	Ležeče (italic)
<tt>	Pisava pisalnega stroja
<u>	Podčrtano (underline)
<s>	Prečrtano (strike - through)
<big>	Večji izpis kot preostali del besedila v tej pisavi
<small>	Manjši izpis kot preostali del besedila v tej pisavi
<sub>	Uporaba za indekse. Spodnji indeks (subscript)
<sup>	Uporaba za indekse. Zgornji indeks (superscript)

Tabela 4: Fizični znakovni slogi

Vse zgoraj omenjen sloge lahko tudi gnezdimo, kar pomeni, da je besedilo lahko hkrati krepko in kurzivno.

tudi že, da nekateri uporabljajo to oznako za oblikovanje slik iz znakov ASCII.

Vodoravne črte naredimo z oznako <hr>. Oznaka nima zaključne oznake </hr>. Najpogostejše se uporabljata atributa za debelino in širino črte.

```
<hr align="center" size="4" width="200" />
```

Oznaka **Line break**
 oziroma prelom vrstice prekine besedilo, kar pomeni, da brskalnik besedilo za to oznako prikaže v novi vrsti skrajno levo. Poudariti velja, da oznaka
 ne vnaša dodatnega praznega prostora med vrstice.

Naslednjič si bomo ogledali še poravnavo besedila in pisave. Začeli pa bomo tudi z grafiko v spletu. ■



PISAVA, GRAFIKA, BARVE ...

Tokrat si oglejmo nekaj možnosti oblikovanja pisave ter uporabe slik, barv in ozadja.

PORAVNAVA BESEDILA

Standard HTML 3.2 je vpeljal možnost poravnave besedila, kar dotlej pri HTML-ju ni bilo mogoče.

Za poravnavo je poskrbel pregledovalnik kar sam. Največkrat je to pomenilo, da je bila vsebina poravnana levo, kar je še zdaj privzeta vrednost. S standardom HTML 4.0 se ni dosti spremenilo, čeprav je priporočljivo uporabljati prekrivne (kaskadne) predloge, ki jih bomo opisali v enem izmed naslednjih nadaljevanj šole.

Za poravnavo posameznih delov besedila uporabljamo atribut align, ki ima lahko tri vrednosti (left, right, center).

```
<h1 align="center"> Naslov, poravnani na sredino strani </h1>
<p align="left"> Besedilo, poravnano levo</p>
<p align="right"> Besedilo, poravnano desno</p>
```



Bolj priporočljiva metoda za poravnavo besedila je uporaba oznake <div>, ki izhaja iz besede division (oddelek). Prednosti te oznake se pokažejo pri večjih segmentih besedila, v katerih smo uporabili različne oznake, kot so <h1>, <p> in podobne. Če uporabimo oznako <div>, je zadeva povsem preprosta. Vse besedilo znotraj oznake bo poravnano tako, kot smo določili na začetku. Za isti učinek in izpis brez uporabe oznake <div> bi morali pri vsaki izmed oznak dopisati atribut align="xxxx", kar pa je seveda zamudno, pa še dokument za prenos po omrežju bo daljši.

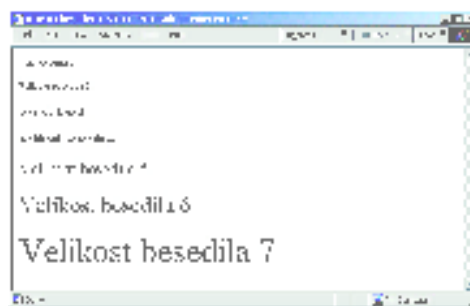
PISAVE IN NJIHOVE VELIKOSTI

Oznaka je bila v zelo razširjena v standardu HTML 3.2. standard HTML 4.0, pa ima do uporabe te oznake nekoliko odklonilen odnos, saj predlaga uporabo prekrivnih slogov. Oznaka se je prvotno uporabljala le za spreminjanje velikosti pisave, pozneje pa se je uporaba razširila tudi na vrsto in barvo pisave.

Velikost pisave določimo z atributom size, ki ima lahko vrednosti med 1 in 7. Privzeta vrednost je 3. Zanimivo je, da lahko povečujemo velikost pisave tudi relativno glede na osnovno

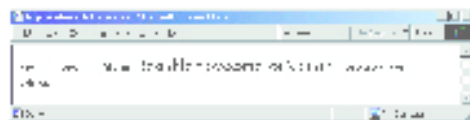
vrednost. To je seveda posledica prekrivnih predlog.

```
<p> <font size="1"> Velikost besedila 1 </font> </p>
<p> <font size="2"> Velikost besedila 2 </font> </p>
<p> <font size="3"> Velikost besedila 3 </font> </p>
<p> <font size="4"> Velikost besedila 4 </font> </p>
<p> <font size="5"> Velikost besedila 5 </font> </p>
<p> <font size="6"> Velikost besedila 6 </font> </p>
<p> <font size="7"> Velikost besedila 7 </font> </p>
```



Relativno povečevanje velikosti besedila je bolj priporočljivo, postopek pa je naslednji:

```
<p> Besedilo privzete velikosti. <font size="+2"> Besedilo povečano za 2 enoti.</font> Ponovno normalna velikost </p>
```



Spreminjanje vrste pisave poteka z oznako in atributom face. Zavedati se moramo, da vrsto, ki jo določimo z atributom, brskalnik išče v operacijskem sistemu uporabnika. Če te pisave sistem nima nameščene, potem brskalnik išče naslednjo vrsto. Če tudi tega ne najde, potem uporabi za prikaz privzeto pisavo. Sintaksa je naslednja:

```
<p> <font face="Futura, Helvetica"> Besedilo </font> </p>
```

GRAFIKA V SPLETU

Drugače kot v prejšnjem delu, tokrat ne bomo spoznali toliko novih oznak, temveč si bomo ogledali, katere vrste slik lahko uporabimo v spletu in kako jih vključimo na spletno stran, nekaj stavkov pa bomo posvetili tudi barvam in ozadjem.

Slike, ki jih srečujemo na spletnih straneh, delimo na dve kategoriji. V prvo sodijo tiste, ki jih vključimo neposredno na spletno stran.

To pomeni, da se slika naloži in prikaže hkrati z vsebino dokumenta in morebitnimi povezavami. V drugo skupino pa sodijo tiste slike, ki se ne prikažejo sočasno z vsebino strani. Naložijo se šele na zahtevo uporabnika. Prednost je zmanjšanje prometa v omrežju, saj se slika ne naloži samodejno. Poleg tega, lahko take strani pregledujemo tudi s pregledovalniki, ki ne podpirajo grafike. Seveda pa lahko slike klub temu shranimo v računalnik in jih pozneje pogledamo z ustreznim grafičnim orodjem. Oglejmo si najprej prvo skupino nekoliko podrobneje.

Ne glede na to, v katero skupino sodijo slike, morajo biti v ustreznem formatu. Za prvo skupino velja, da mora biti slika zapisana v formatu GIF ali JPEG. Standard GIF je bolj razširjen, poleg tega ga podpirajo skoraj vsi grafični pregledovalniki. V zadnjem času je vse bolj priljubljen tudi zapis JPEG, saj ga podpirajo skoraj vsi digitalni fotoaparati, kar pomeni, da sliko, ki smo jo posneli, brez težav prenesemo na spletno stran. Prednost je v tem, da za pretvorbo iz enega v drug zapis v splošnem ne potrebujemo posebnega grafičnega urejevalnika slik. To je žal le deloma res, saj se izkaže, da je treba sliko vsaj malce popraviti (odstranitev rdečih oči, posvetlitev ali potemnitev fotografije in podobno), preden jo objavimo v spletu. Tukaj lahko omenimo nekaj programskih paketov, ki se uporabljajo v ta namen. Med njimi so na prvem mestu Adobe Paint Shop Pro, Adobe Photoshop in Micrografx Picture Publisher. Nekoliko več o urejanju fotografij pa v enem od naslednjih nadaljevanj šole HTML.

 oznaka

Ko imamo sliko v ustreznem formatu, jo lahko vključimo na spletno stran. Za to uporabimo oznako , ki nima zaključne oznake, podobno kot <hr>. Za zapis XHTML, pa je treba na koncu dodati še en presledek in znak /. Med najpomembnejšimi atributi je src, ki nam pove ime in pot do slike, ki jo želimo vključiti. Tako bi za vključitev slike v formatu GIF, ki je v isti mapi kot dokument HTML zapisali:

```

```

Naslednji pomemben atribut je alt, ki postane nepogrešljiv v primerih, ko imamo le besedilni pregledovalnik ali pa izklopljeno samodejno prikazovanje slik. Za zgornji primer pomeni, da bi se na mestu, kjer bi morala biti slika, izpisalo besedilo »Ta slika je zelo lepa«.

Oznaka `<img>` pozna tudi attribute za poravnavo, ki so naštetih v spodnji tabeli.

<code>align="top"</code>	poravnava zgornjega roba slike z najvišjim delom v vrstici
<code>align="middle"</code>	poravnava sredine slike s sredino vrstice (baseline)
<code>align="bottom"</code>	poravnava spodnjega roba slike z najnižjim delom vrstice

Tabela 1: Atributi za poravnavo slik

V tabeli 2 pa so naštetih tudi atributi, ki so v uporabi, a jih vsi pregledovalniki ne podpirajo enako in tudi niso specifikirani v standardih HTML 3.2 in HTML 4.0. Seveda jih ne najdemo tudi v standardu XHTML 1.0. Omenili jih bomo zgolj zato, ker jih srečamo precej pogosto.

<code>align="texttop"</code>	poravnava zgornjega roba slike z zgornjim robom besedila v vrstici
<code>align="absmiddle"</code>	poravnava sredine slike s sredino najvišjega elementa v vrstici
<code>align="baseline"</code>	poravnava spodnjega roba slike z osnovno vrstico (baseline) besedila
<code>align="absbottom"</code>	poravnava spodnjega roba slike z najnižjim elementom v vrstici

Tabela 2: Nestandardni atributi za poravnavo slik

Za oblikanje besedila okoli slike poznamo od HTML 2.0 naprej dva atributa.

`align="left"` in `align="right"`, ki poravnata sliko vsak na svojo stran. Atributa tudi povzročita, da se bo celotno besedilo, ki sledi atributom, izpisalo levo ali desno na prostoru poleg slike, odvisno od vrednosti atributa. Atribut `clear`, ki ima lahko vrednosti `left`, `right` ali `all`, pa konča oblikanje besedila okoli slike.

Atributa za določanje praznega prostora okoli slik sta `vspace` in `hspace`. Vrednosti navedemo v pikah (piksljih).

Slike in povezave

Vsaka slika nam lahko, podobno kot besedilo, rabi za povezavo. Sliko preprosto kliknemo z miško in povezava nas pripelje do nove strani. Sintaksa je naslednja:

```
<a href="dokument.html"></a>
```

Torej s klikom na sliko `slika.gif` odpremo nov dokument z imenom `dokument.html`.

Slike, ki se ne naložijo

Poglejmo si na kratko še slike, ki sodijo v drugo kategorijo. Te slike v pregledovalniku

ne prikažejo samodejno. Običajno se prikaže le slika z nizko ločljivostjo (thumbnail), le za »okus«. Večjo sliko pa je treba v računalnik naložiti z dodatnim klikom.

BARVE

Barve vsekakor poživijo spletno stran. Najprej si pogledajmo, kako barve poimenujemo. Običajno se uporablja šestnajstiško število, lahko pa uporabimo tudi angleško ime za posamezne barve. Poimenovane niso vse barve, le nekatere (16). (Black, White, Green, Maroon, Olive, Navy, Purple, Gray, Red, Yellow, Blue, Teal, Lime, Aqua, Fuchsia in Silver). Najzanesljivejša je uporaba šestnajstiške številke. V rabi je model RGB, pri katerem dobimo za vsako izmed osnovnih barv številko med 0 in 255. Tako dobimo za #0000 črno in 255 255 255 (ffffff) belo barvo.

Spreminjanje barve ozadja

Za spremembo barve ozadja uporabljamo oznako `<body>` in atribut `bgcolor`.

```
<body bgcolor="#ffffff"> ali <body bgcolor="white">
```

Spreminjanje barve besedila

Skoraj vedno, kadar spremenimo barvo ozadja, moramo spremeniti tudi barvo besedila. Tudi pri tem nam pomaga oznaka `<body>` z naslednjimi atributi:

<code>text</code>	Z njim določimo barvo (body) besedila. Nima vpliva na barvo povezav
<code>link</code>	Z njim določimo barvo povezav
<code>vlink</code>	Z njim določimo barvo povezav, ki smo jih že obiskali (visited links)
<code>alink</code>	Z njim določimo barvo povezave, ki je trenutno aktivna. To pomeni, da je miška nad povezavo in gumb pritisnjen in zadržan.

Tabela 3: Atributi za barvo besedila

SLIKE KOT OZADJE STRANI

Omenili smo že, da barve prinesejo na spletne strani precej svežine. Še zlasti pa te zaživijo, če uporabimo namesto posameznih barv kar slike. Žal pa uporaba slik, še posebej če so obsežne, podaljša čas nalaganja strani. Dobro je, če se za vsako sliko posebej vprašamo, ali jo res nujno potrebujemo na strani ali pa je tam samo za okrasek. Podobno kot velja za drugo spreminjanje barv tudi tokrat uporabimo oznako

```
<body>. Atribut pa je background.
<body background="ozadje.gif">
```



RADIO KRANJ
97.3 MHz

GORENJSKI MEGASRČEK

NAJBOLJ POSLUŠAN
RADIALSKA POSTAJA NA GORENJSKEM

RADIO KRANJ d.o.o.
Slovenski trg 1, KRANJ

TELEFON:
(04) 2022-825 REDAKCIJA
(04) 2021-186 REŽIJE
(04) 2022-222 PROJEKCIJA

FAX:
(04) 2021-866 REDAKCIJA
(04) 2025-290 PROJEKCIJA

E-pošta:
radio@kranj@radio-kranj.si

Spletna stran:
http://www.radio-kranj.si

brother
laserski tiskalniki



KAKO DO PRIMERNIH SLIK ZA SPLET?

V prejšnjem delu naše šole smo se omejili le na vključevanje slik v spletne strani. Zdaj pa si oglejmo še podrobnosti v zvezi s slikami.

FORMATI ZAPISA GIF

Graphics Interchange Format ali Compu-Serve GIF je oblika zapisa slik, ki se najpogosteje uporablja na spletnih straneh. Zapis je bil razvit, kot govori že ime, za izmenjavo slik. Najprej je bil GIF87, ki je tudi izvirni format, sledil mu je GIF89a, ki vsebuje nekatere zelo dobrodošle dopolnitve. Naj omenim le prosojnost in možnost zapisa več okvirjev v eni sliki (multi-frame), kar nam rabi za pravo preprostih animacij. Edina večja slabost zapisa GIF je, da ima lahko slika, zapisana v tem formatu največ 256 barv, kar pa ni primerno za fotografije. Naslednja slabost tega formata pa ni tehnične narave. Problem je plačilo razvijalcev in nosilcu patenta, podjetju UniSys. V tem je razlog, da se zapis GIF ne razvija več. Pričakovati je, da ga bo zamenjal drug zapis.

JPEG

Joint Photographic Experts Group je oblika zapisa, ki je bila razvit z namenom shranjevanja fotografskih slik. Za razliko do formata GIF lahko pri formatu JPEG uporabimo poljubno barvno globino. Zaradi metode stiskanja je primeren za fotografije, precej slabše pa se obnese pri slikah z ostrimi robovi (logotipi, risbe).

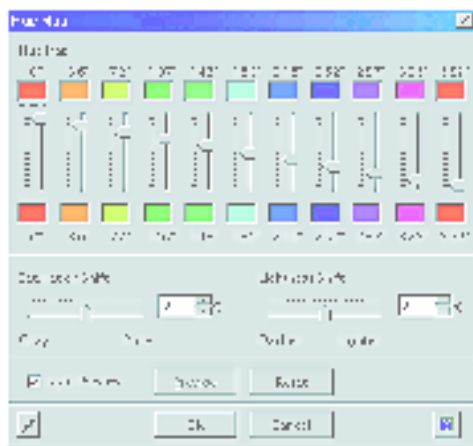
BARVE

V tehniko barv in v nasvete, kako jih uporabljati na spletnih straneh se ne bomo spuščali, saj je to povsem subjektivna zadeva, ko gre za všečnost. Poglejmo si najpogosteje uporabljana barvna modela, HSB in RGB.

Model HSB (Hue, Saturation, Brightness)

Ta model je subjektiven, saj intuitivno opisuje, kako ljudje dojemajo barve in njihove spremembe. Vsaka barva je opisana s tremi številskimi, ki predstavljajo barvni odtenek (hue), intenziteto (saturation) in svetlost (brightness). S tem modelom lahko predstavimo poljubno barvo, ki jo lahko vidimo.

Barvni odtenek (hue) določimo z naslednjim krogom. Vrednosti tega parametra so med 0 in 360.



Odtinki v barvnem modelu HSB

Intenziteta barve (saturation) pomeni, koliko je posamezne barve v mešanici barv. Intenziteto merimo v odstotkih, zato parameter zavzame vrednosti med 0 in 100, pri čemer 100 pomeni, da je prisotna samo ena barva (solid).

Parameter svetlost (brightness) nam pove, kako svetla ali temna je barva. Tudi svetlost se meri v odstotkih, zato so vrednosti parametra med 0 in 100.

Model RGB (Red, Green, Blue)

Model RGB uporabljajo računalniški monitorji za opis slike. Vsako izmed pik na zaslonu naredijo trije laserski žarki, ki so rdeče, zelene in modre barve. Z barvno kombinacijo omenjenih pik in njihovo intenzivnostjo dobimo posamezno barvo na zaslon.

Barvna paleta WEB safe

Poglejmo nekoliko v zgodovino, ko je podjetje Netscape poslalo na trg svoj prvi brskalnik. Takrat je bila večina računalnikov zmožna prikazati le 256 barv, kar danes ni več nikakršen problem. Tedanji brskalnik je deloval tudi na različnih operacijskih sistemih in vsak izmed njih je imel drugačno sistemsko barvno paleto. Za uporabnika pa je to pomenilo, da barva na vsakem računalniku ni bila povsem enaka. Problem so pri podjetju Netscape rešili tako, da so določili barvno paleto, ki vsebuje 216 barv. Če v spletu uporabimo katerokoli izmed teh 216 barv, potem ne bomo imeli ne mi ne uporabniki

na drugi strani nikakršnih težav. Vsekakor pa je tudi dandanes priporočljiva uporaba varne barvne palete.

STISKANJE SLIK

Zamislimo si opis bitne slike s 24-bitno barvno globino. Pomnožiti moramo vse pike na sliki (višina [pika] x širina [pika]). Skupno število pik pomnožimo s številom bitov, s katerimi so zapisane barve. V našem primeru je to 24 bitov. Tako dobimo velikost slike v bitih, ki je pretvorimo v bajte tako, da jo delimo z 8. V tabeli je to v zadnjem stolpcu. Številke so v kilobajtih (KB).

Velikost slike [pika]	Število pik	Velikost slike, 24-bitne barve [pika]	Velikost slike, 24-bitne barve [KB]
160 x 120	19.200	460.800	57.600
240 x 180	43.200	1.036.800	129.600
320 x 240	76.800	1.843.200	230.400
384 x 288	110.592	2.654.208	331.776
640 x 480	307.200	7.372.800	921.600

Povezava med velikostjo slike v pikah in bajtih pri 24-bitni barvni globini

Seveda ni treba posebej poudarjati, da je treba vsako tako sliko prenesti po omrežju. Zamislimo si lahko, kakšen promet po omrežju povzročamo s tako velikimi slikami. Poleg tega, pa je zadeva precej zamudna tudi za uporabnika na drugi strani, saj mora počakati, da se stran s tako sliko naloži. Iz svojih izkušenj lahko povem, da proces nalaganja take strani kaj hitro prekinem in se na tako stran ne vračam več. Torej, moj nasvet je, da na spletne strani vključite le slike, ki so po količini bajtov majhne – največ do nekaj deset kilobajtov.

Zgornja tabela velja za slike, ki niso stisnjene. Za prenos slikovnega gradiva po omrežju, so razvili različne metode in algoritme stiskanja slik. Stiskanje pomeni, da zmanjšamo število bajtov, s katerimi je slika zapisana. Ohrani pa se vsebina slike. Slika je dimenzijsko še vedno enako velika in ima isto število barv, kot jih je imela pred stiskanjem. Na zaslonu bo videti popolnoma enako.

Mnogi med vami zagotovo poznate program

za stiskanje WinZip. Za primer vzemite sliko, shranjeno v formatu BMP. Poskusite jo shraniti s tem algoritmom. Opazili boste, da bo nova datoteka precej krajša, vsebina slike pa se bo ohranila. Že zgoraj smo omenili formata GIF



in JPEG. Uporabljata različni metodi stiskanja podatkov, ki imata svoje prednosti in slabosti. Slike GIF uporabljajo algoritem stiskanja LZW. LZW so kratice avtorjev algoritma Lempel, Ziv in Welch. Metoda deluje po načelu iskanja sorodnih pik z isto barvo. Več kot je takih pik, bolj bo slika stisnjena. Ta metoda je najprimernejša za slike z manjšim številom barv, ki niso razgibane. Prav nasprotna pa je metoda JPEG, ki shrani razlike med posameznimi pikami na sliki. Ta algoritem doseže najboljše rezultate pri fotografijah in slikah, ki imajo ogromno barv. Slabše pa se obnese pri slikah z večjimi površine enake barve ali ostrimi robovi. Omeniti pa moram, da algoritem JPEG tudi med shranjevanjem zgublja nekaj informacij o sliki. To pomeni, da reprodukcija take slike ne bo enaka izvorniku. Pri shranjevanju slike JPEG, pa imamo možnost nastaviti dovoljeno količino informacije, ki se lahko izgubi. Upravičeno se lahko zdaj vprašate, kako pa je s prikazovanjem stisnjenih slik na uporabnikovem računalniku. Vsi verjetno nimajo algoritma, ki bi znal prebrati stisnjeno sliko. Seveda, stisnjene slike ne moremo prikazati na zaslonu, dokler ne uporabimo algoritma, ki zna prebrati tako stisnjene podatke. Vsi brskalniki imajo vdelane take algoritme za slike formata GIF in JPEG, zato smo lahko glede tega brez skrbi. Kako hitro pa se bo algoritem izvedel, je odvisno od zmogljivosti procesorske enote v uporabnikovem računalniku. Današnji računalniki pa so vsi dovolj zmogljivi, da se to zgodi praktično v trenutku.

Za primerjavo si oglejmo **sliki zgoraj**, zapisani v formatu JPEG. Leva, velikosti 28 KB je kakovostna. Desna pa je zelo slabe kakovosti, vendar je velika le 4 KB.

Malce nas je zaneslo in smo se oddaljili od snovi, a se mi zdi pomembno, da si razjasnimo nekaj osnovnih pojmov, ki jih običajen bralec ne sreča prav pogosto. Vrnimo se k snovi in se ustavimo pri formatu GIF.

FORMAT GIF Prosojnost

Značilnost prosojnih slik GIF je, da so določena mesta na sliki prosojna. To pomeni, da lahko vidimo vse, kar je za tistim delom slike. To možnost lahko s pridom izkoristimo, da so slike poljubnih oblik za razliko od običajnih kvadratnih. Take slike lahko naredimo s posebnimi grafičnimi orodji, kot so Micrografx Picture Publisher, Adobe Image Styler, Paint Shop Pro, Macromedia Fireworks in druga. Seveda pa lahko najdemo v spletu precej programčkov shareware, s katerimi si lahko pomagamo precej ceneje. Najlažje naredimo prosojno ozadje, če je to enobarvno. Največ težav je, če je barva ozadja vsebovana tudi v delu slike, za katerega ne želimo, da je prosojen. Izdelava take prosojne slike GIF zahteva kar precej truda. Običajno v takih primerih izrežemo del slike in ga postavimo na drugo, enobarvno ozadje in nadaljujemo s postopkom.

Prepletanje

Zanimiva lastnost, ali bolje rečeno učinek, ki ga omogočajo slike GIF, je prepletanje (in-

terlacing). Ta lastnost ne vpliva na kakovost slike ali kaj podobnega. Prepletanje je način shranjevanja slike, ki se pokaže šele pri nalaganju slike v brskalniku. Gotovo ste že opazili, da se običajne slike nalagajo od zgoraj navzdol.

Če smo sliko shranili s prepletanjem, pa se slike shranjujejo in seveda tudi nalagajo v štirih pasovih, pri čemer je v enem pasu vsaka osma vrstica. Uporabnost tega vidim le pri večjih slikah, saj lahko že dokaj hitro vidimo obris slike in prekinemo nadaljnje nalaganje, če se nam slika ne zdi zanimiva.

Animacije

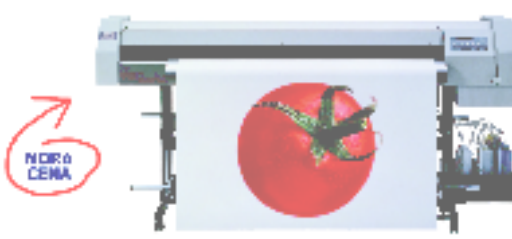
Format GIF omogoča izdelavo preprostih animacij, zato imenujemo tak zapis tudi animirani GIF. V tako datoteko zapišemo več slik GIF in določimo zakasnitev med eno in drugo. Tako dobimo preprosto animacijo. V brskalniku vidimo tak GIF kot zaporedje shranjenih slik, ki se ponavlja v nedogled. Uporaba animiranih datotek GIF je zelo široka. Največkrat jih srečamo kot ikone, logotipe, črte, gumbe in podobno. Animirane datoteke GIF naredimo s posebnimi orodji. Med najbolj znanimi je GIF Construction Set podjetja Alchemy Mindworks.

FORMAT JPEG

Nekaj podobnega, kar smo srečali pri formatu GIF kot prepletanje, omogoča tudi JPEG. Imenuje se napredni (progressive) JPEG. Tudi v tem primeru gre za način shranjevanja slike. Taka slika se v brskalniku najprej prikaže zelo nenatančno. Z nadaljnjim nalaganjem pa je slika vedno ostrejša, dokler ni naložena do konca.




N.C.R. Ljubljana d.o.o. / Brnčičeva 13 / tel. 01 561 17 13





Tiskalniki velikega formata z solventnimi barvami

Strojna oprema
Programska oprema
Potrošni material
Consulting
Novosti
Storitve

ANIMIRANA GRAFIKA IN TABELE

Doslej smo se ukvarjali le z mirujočimi slikami. V svetovnem spletu pa lahko opazimo ogromno najrazličnejših animacij, v katerih predmeti utripajo in skačejo naokoli. Pojasnimo si najprej, kaj pomeni izraz animacija.

ANIMACIJA

Predstavljajmo si, da imamo slike zložene eno vrh druge. Vsaka slika se malenkostno razlikuje od prejšnje. Če so slike smiselno zložene, potem pri hitrem pogledu na zaporedje slik dobimo vtis gibanja. Hitrost animacije pa je odvisna od tega, kako hitro pregledujemo posamezne slike. V osnovi so vse datoteke, pa naj so GIF, AVI, QTM (Quick Time Movie) ali MPEG, sestavljene iz zaporedja slik, ki jih nato pregledujemo.

UPORABA NA SPLETNIH STRANEH

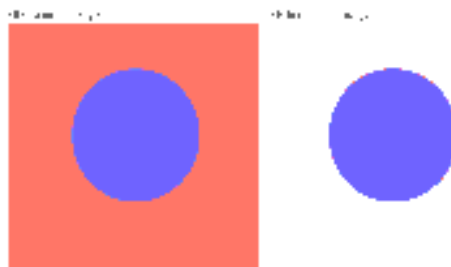
Glavni namen animacije je pritegniti pozornost spletnega obiskovalca. Ko na določeni strani opazimo gibanje, nas to pritegne. S tem je cilj, ki si ga je zastavil oblikovalec strani, dosežen. Važno je le še sporočilo, ki ga dobimo ob animaciji. Seveda pa moramo biti pri uporabi animacij previdni, kajti preveč animacij na posamezni strani ljudi odvrča, saj je taka stran moteča. Poleg tega je treba omeniti tudi to, da se taka stran precej dolgo nalaga, kar tudi negativno vpliva na obisk strani. Zamislimo si uporabnike, ki deskajo prek analognih modemov. Takih je zaenkrat še precej. Preveč načičkana stran tudi zmede obiskovalca, ki ne ve, kam naj najprej pogleda, katera informacija je najbolj pomembna. Zamislimo si strani, na katerih kar mrgoli oglaševalskih pasic (banners). Praviloma nikoli ne kliknemo na tako pasico, saj nam odpre še kopico novih strani. Če jih želimo zapreti, se odprejo nove itd. Poleg pasic srečujemo tudi animirane gumbе, ki popestrijo stran in so dobrodošli. Primerna je tudi uporaba animiranih logotipov izdelkov in podjetij, saj morajo pritegniti obiskovalca.

DATOTEKA GIF BREZ OZADJA

Brez dvoma ste že zasledili spletne strani, na katerih so slike nepravilnih oblik. Običajne slike so pravokotnih oblik. Odgovor je v datotekah GIF, ki nimajo ozadja, ali pa je ozadje iste barve kot ozadje spletne strani.

Večina programske opreme za obdelavo bitnih slik omogoča izdelavo datotek GIF brez ozadja. Zamislimo si preprost zgled. Vzemimo, da je slika rdeče barve. Na sredino narišemo moder krog. Sliko nato shranimo kot datoteko GIF. Če želimo na spletni strani prikazati le moder krog, bomo pri shranjevanju datoteke GIF izbrali, da je rdeča barva v naši datoteki GIF prozorna. Tako shranjeni GIF se bo na

spletni strani prikazal kot moder krog brez ozadja. Okoli kroga bo ista barva, kot je ozadje spletne strani. Primer ponazarja slika 1.



Programov, ki nam pomagajo pri izdelavi animiranih datotek GIF, je v spletu precej. Med tistimi, ki se najpogosteje pojavljajo, je GIF Construction Set, ki deluje v okolju Windows. Programček je vrste shareware. Za zahtevnejše pa sta gotovo zanimiva paketa Paint Shop Pro in Ulead PhotoImpact.

Na koncu je pomembna tudi optimizacija animacije GIF. Omenili smo že, da datoteke GIF vsebujejo v osnovi 256 barv. Običajno lahko z barvno redukcijo privarčujemo nekaj bajtov pri dolžini datoteke. To se še posebej pozna pri času nalaganja strani. Velikokrat se izkaže, da ni opaznih razlik, če zmanjšamo število barv z 256 na 32 ali celo 16.

Vrnimo se zdaj k jeziku HTML in si oglejmo tabele.

TABELE

Doslej smo za postavljanje besedila na strani uporabljali ukaze HTML (ukaze, naštevanja in podobno). S tabelami lahko razdelimo besedilo na vrstice in stolpce, pri tem pa ni nujno, da uporabimo okvir okoli tabele, kar pomeni, da so celice tabele nevidne. Seveda pa lahko v posamezne celice tabele vključimo vso vsebino HTML, od besedila do tabel, slik, povezav in podobno.

Tabele so se prvič pojavile v standardu HTML 3.2. Od takrat jih urejevalci spletnih strani uporabljajo pri oblikovanju videza spletnih strani. Njihova prednost pred posebnimi ukazi HTML je v tem, da jih vsi pregledovalniki tolmačijo enako, kar ne moremo reči za nekatere ukaze, ki uporabljajo plasti (layers).

Pri izdelavi tabel nam je v izdatno pomoč urejevalnik HTML, s katerim naredimo poljubno tabelo tako, da le povemo, koliko vrstic in stolpcev želimo. Drugo naredi urejevalnik sam. Precej za-

mudno bi bilo, če bi vsakič sproti pisali tabelo na roke. Kodo popravljamo le v posebnih primerih, ko želimo doseči nekaj posebnega, kar ni zajeto v urejevalniku. Kljub temu si pogledjmo, kako je videti tabela in iz katerih oznak je sestavljena.

Kratek zgled tabele, v katerem so imena, priimki in spol treh članov uredništva Mojega mikra.

Zgled 1:

```
<html>
<head>
<title>Tabela</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1250">
</head>
<body bgcolor="#FFFFFF"
<table width="120" border="0" bordercolor="#CCCCFF"
  bgcolor="#CCCCCC">
  <caption>Tabela </caption>
  <tr>
    <th>Ime</th>
    <th>Priimek</th>
    <th>Spol</th>
  </tr>
  <tr>
    <td>Marjan</td>
    <td>Kodelja</td>
    <td>M</td>
  </tr>
  <tr>
    <td>Zoran</td>
    <td>Banovič</td>
    <td>M</td>
  </tr>
  <tr>
    <td>Milan</td>
    <td>Simčič</td>
    <td>M</td>
  </tr>
</table>
</body>
</html>
```

Tabela

Ime	Priimek	Spol
Marjan	Kodelja	M
Zoran	Banovič	M
Milan	Simčič	M

Slika 2 prikazuje rezultat v brskalniku Microsoft Internet Explorer.

VRSTICE IN STOLPCI

Znotraj oznake <table> ... </table> imamo v tabeli vrstice. Torej določamo le vrstice, kar pomeni, da določimo najprej prvo polje v vrstici

in njegovo vsebino, nato drugo polje in tako naprej do konca vrstice. Tako si vrstice sledijo druga za drugo, odvisno od tega, koliko kolon želimo imeti. Vsaka vrstica se začne z oznako `<tr>` in konča z oznako `</tr>`.

Oznaka `<th> ... </th>` se uporablja za osnovno vrstico tabele, v kateri je vsebina prikazana odeljeno. `<th>` pomeni table heading (naslov tabele)

Oznaka `<td> ... </td>` se uporablja za označevanje posameznih celic. `<td>` pomeni table data (vsebina tabele).

Pri oblikovanju strani včasih potrebujemo tabele, v kateri so posamezne celice prazne. Težava se pojavi, ker nekateri brskalniki preprosto ignorirajo prazne celice v tabeli in jih sploh ne prikažejo. Temu se izognemo tako, da vstavimo v celico, za katero želimo, da je prazna, presledek.

Primer 2:

```
<html>
<head>
<title>Tabela</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1250">
</head>
<body bgcolor="#FFFFFF">
<p>V prvo polje v tabeli smo vpisali presledek</p>
<table width="120" border="1">
<tr>
<td>&nbsp;</td>
<td>1</td>
<td>2</td>
</tr>
</table>
<p>Prazno prvo polje v tabeli</p>
<table width="120" border="1">
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
</table>
<p>&nbsp;</p>
</body>
</html>
```

Slika 3 ponazarja zgled 2

V prvo polje v tabeli smo vpisali presledek.

	1	2
--	---	---

Prazno prvo polje v tabeli

	1	2
--	---	---

Parametri, ki vplivajo na videz tabele:

VELIKOST TABELE

V večini primerov je najbolje, da odločitev glede velikosti tabele prepustimo kar brskal-

niku. Za posebne primere pa imamo na voljo posebne attribute. Za širino tabele se uporablja atribut `width`. Vrednost lahko podamo v odstotkih, ki se nanašajo na velikost zaslona ali pa določimo natančno velikost v pikah (pikslih). Razlika je v tem, da bo pri natančni določitvi širine tabele v točkah tabela povsod enako velika. Če podamo širino v odstotkih, pa bo ta različno velika, a vedno določen odstotek velikosti zaslona. Velikost tabele se bo spremenila, če bomo nastavili drugo ločljivost zaslona. Z atributom `width` lahko določimo tudi širino posameznega stolpca v tabeli. Uporaba atributa je najbolj smiselna takrat, ko imamo več stolpcev, za katere želimo, da so enake širine.

OKVIR TABELE

Z ukazom `<table width="40%" border="10">` ustvarimo tabelo širine 40% zaslona. Okvir tabele bo debel 10 točk. Zgled prikazuje slika 4.

	1	2
--	---	---

Atribut **Cell Padding** določa količino praznega prostora med vsebino celice in okvirjem celice. Privzeta vrednost je 1. V našem primeru bomo to vrednost povečali na 10.

```
<table width="40%" cellpadding="10" border="10">
```

Rezultat vidimo na sliki 5.

	1	2
--	---	---

Cell Spacing je podoben ukaz kot `cell padding`. Ta ukaz poveča razmak med posameznimi celicami. Privzeta vrednost je 2. Za naš primer bomo to vrednost povečali na 10.

```
<table width="40%" cellpadding="10" cellspacing="10" border="10">
```

Rezultat prikazuje slika 6.

	1	2
--	---	---

Prelome vrstic znotraj posamezne celice v tabeli določamo z ukazoma `<br />` in `nowrap`. Pri uporabi teh atributov moramo biti previdni in skrbno preveriti videz tabele v brskalniku. Poizkusiti moramo tudi tako, da spreminjamo velikost okna.

TABELA IN BARVE

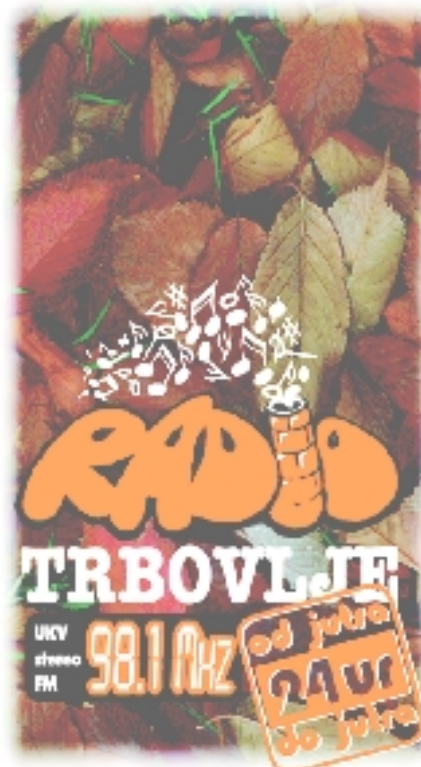
Za določanje barve znotraj tabele, vrstice in celice uporabljamo atribut `bgcolor`. Vrednosti

lahko zapišemo kot šestnajstiški trojček števil po standardu RGB. Brskalnika MS Internet Explorer in Netscape Navigator pa prepoznata tudi 16 imen barv (Black, White, Green, Maroon, Olive, Navy, Purple, Gray, Red, Yellow, Blue, Teal, Lime, Aqua, Fuchsia in Silver). Pomembno je, da si zapomnimo, da barva ozadja, ki jo določimo npr. znotraj vrstice, razveljavi barvo, ki smo jo določili za ozadje tabele. Podobno velja tudi za barvo ozadja celice, ki ima višjo prioriteto od barve vrstice in tabele. Ne smemo pa pozabiti zamenjati tudi barve besedila, če smo že zamenjali barvo ozadja, saj se nam lahko zgodi, da besedila ne bomo mogli prebrati. Za spremembo barve besedila uporabimo ukaz `<font color ...>`. Zamenjamo lahko tudi barvo okvirja. To naredimo z ukazom `<bordercolor>`, pri tem pa izgubimo 3D videz okvirja. Internet Explorer pozna tudi ukaza `<bordercolor-dark>` in `<bordercolor-light>`. Prvi določi temno komponento 3D okvirja. To barvo postavi na desno in spodnjo stran okvirja. Drugi ukaz pa postavi na zgornjo in levo stran okvirja.

Zgled lahko vidimo na sliki 7.

	1	2
--	---	---

Danes smo se naučili izdelati tabelo, jo obarvati, naslednjič pa bomo začeli s poravnavo besedila v tabeli. ■



<http://www.radio-trbovlje.si>

TABELE, JAVA, OBRAZCI ...

Nadaljujmo našo šolo HTML z oblikovanjem tabel in se posvetimo še drugim temam.

Poravnava tabel

Poravnava tabel na strani in poravnava vsebine tabel sta dve različni zadevi. Tabela poravnamo z ukazom `align="left"` ali `"right"`

```
<table border="1" align="left" width="70">
```

Usredinjanje tabele na zgornji način je podprto šele od 4. različice brskalnika dalje. Do takrat pa je bil v uporabi ukaz `<center>` ali pa `<div align="center">`.

Vsebine tabele oziroma posamezne celice znotraj tabel lahko poravnavamo na različne načine. In sicer poznamo vodoravno (`left`, `right`, `center`) in navpično (`top`, `bottom`, `middle`) poravnavo.

HTML pozna tudi poravnavo naslovov tabel, slik (`caption`). Uporabimo naslednji ukaz:

```
<caption align="bottom">vsebina naslova</caption>
```

Zapletene tabele

Poglejmo si še nekoliko bolj zapleten zgled tabele, pri katerem so vrstice čez več stolpcev in stolpci čez več vrstic.

```
<html>
<head>
<title>Tabela</title>
</head>

<body>
<table border="1">
  <tr>
    <th colspan="2">&nbsp;</th>
    <th>Moj<br />mikro</th>
  </tr>

  <tr align="center">
    <th rowspan="2">Stran</th>
    <th>Zgoraj</th>
    <td>Prispevek 1</td>
  </tr>

  <tr align="center">
    <th>Spodaj</th>
    <td>Prispevek 2</td>
  </tr>
```

```
</table>
</body>
</html>
```

		Moj Mikro
Stran	Zgoraj	Prispevek 1
	Spodaj	Prispevek 2

JAVA

Najprej malo zgodovine. Java se je v začetku imenovala OAK in razvili so jo v podjetju Sun Microsystems. Programski jezik naj bi bil primeren za naprave PDA (Portable Data Assistant) in za interaktivno televizijo. Javo odlikuje predvsem to, da jo lahko vključimo v skoraj poljubno napravo, ki jo sprogramiramo po naših željah. Po nekajletnem životarjenju je soustanovitelj podjetja Sun Bill Joy prišel do zamisli, da bi lahko Javo uporabili tudi v internetu.

Java je objektno usmerjen programski jezik, ki je podoben jeziku C++. Eden glavnih razlogov, da so napisali programski jezik Java pa je bila želja in potreba po programskem jeziku, ki bi bil neodvisen od programske opreme, na kateri deluje. Samo predstavljajmo si, koliko različnih računalnikov z različnimi operacijskimi sistemi je priključenih v internet. Žal programa, ki ga napišemo v jeziku C++ za operacijski sistem Windows ne moremo brez težav prenesti v računalnik Macintosh. Programski jezik Java pa je bil razvit ravno s tem namenom, da program, ki ga napišemo, deluje v različnih operacijskih sistemih, brez posebnih pretvorb. Vendar javanski prevajalnik ne prevaja izvorne programske kode označene s končnico `.java` v strojno kodo, ki je odvisna od strojne in programske opreme uporabnika programa, temveč v bajtno kodo s končnico `.class`, ki je od tega neodvisna. Programček se nato izvede v programskem okolju javanskega navideznega stroja (Java Virtual Machine – Java VM), ki ga vsebujejo javanski pregledovalniki (Java Enabled Browsers).

Programski jezik Java vsebuje vrsto programskih vmesnikov, katerih glavna prednost je v prenosljivosti prek cele vrste programskih in strojnih okolij. Ti programski vmesniki omogočajo delo s pomnilnikom, uporabniškimi okni, obravnavanje uporabniških vnosov in podobno. Sami programski vmesniki pa so združeni v posamezne pakete, ki omogočajo njihovo preprosto nadgradnjo ter modularnost njihove zasnove in uporabe.

Preidimo sedaj na tisti del Jave, ki nas v tem trenutku zanima, saj pišemo o kodi HTML in spletu.

Javanski programčki

Javanski programčki (applets) so le majhni deli programske kode, lahko bi rekli nekakšna okrnjena vrsta javanskih programov, in niso

namenjeni samostojnemu delovanju znotraj uporabniškega operacijskega sistema. Programčki morajo biti, da lahko delujejo, vstavljeni v drugo aplikacijo (običajno spletni pregledovalnik). Zaradi varnostnih razlogov imajo tudi zelo omejene komunikacijske sposobnosti, saj ne morejo neposredno dostopati do uporabnikovega operacijskega sistema, prav tako pa lahko navzven komunicirajo le s strežnikom, iz katerega so bili nameščeni. Rabijo za vključevanje dinamičnih podatkov, običajno zvoka, slike, podatkov iz preglednic in zbirke podatkov, v statične hipertekstualne dokumente. Javanski programčki so shranjeni v spletnih strežnikih in pregledovalnik jih naloži po potrebi skupaj z dokumentom, ki se nanje sklicuje.

Ustvarjalci hipertekstualnih dokumentov vstavijo javanske programčke v kodo HTML s pomočjo oznake `<APPLET>`, kateri dodajo še dodatne parametre, kot so spletni naslov programčka in njegove morebitne dodatne parametre in možnosti.

Javanska razvojna orodja

Preden začnemo pisati svoje javanske programčke, potrebujemo katero izmed javanskih razvojnih orodij. Med bolj znanimi komercialnimi razvojnimi orodji so JDK podjetja Sun, Java Workshop podjetja JavaSoft, Symantecov Visual Cafe, Microsoftov Visual J++, IBM-ov Visual Age for Java in še bi lahko naštevati. Ta razvojna orodja vsebujejo vsa potrebna orodja za prevajanje javanskih programčkov, različne knjižnice, ki jim pravijo `classes`, samostojen pregledovalnik javanskih programčkov, v katerem lahko rezultate svojega dela preverimo in na koncu še razhroščevalnik (angl. debugger) za odkrivanje in lažje odpravljanje napak. Vesela novica za tiste, ki nimajo denarja za nakup enega izmed zgoraj omenjenih komercialnih orodij. Podjetje Sun omogoča, da njihovo razvojno orodje pretočimo tudi prek spleta.

Znotraj šole HTML se ne bomo spuščali v pisanje javanskih programčkov. Omeniti želimo le, kako take prevedene programčke nato vključimo v dokument HTML. Za to uporabimo oznako `<applet>`. Pravilna sintaksa pa je naslednja

```
<applet code="name.class" width=točke
height=točke></applet>
```

Programček mora biti v isti mapi kot dokument HTML iz katerega programček poganjamo. Obvezno moramo navesti širino (wi-

dth) in višino (height), saj se sicer programček ne bo prikazal na strani.

V spletu najdemo ogromno javanskih programčkov, ki jih lahko uporabimo in vključimo v svoje spletne strani oziroma dokumente HTML. Nekaterih niti ni potrebno prekopirati v svoj računalnik, saj jih lahko poženemo kar iz spleta. Ta možnost sicer ni najbolj priporočljiva, saj nikoli ne vemo, kdaj tisti strežnik bo oziroma ne bo deloval. Poleg tega lahko avtor programček tudi umakne s tiste strani, naša stran potem ne bo pravilno delovala. Brskalnik običajno sporoči napako. Torej vsega ni treba napisati od samega začetka. Večinoma se spleta malce pobrskati po spletu in gotovo bomo našli že napisan programček, ki ga potem lahko zelo preprosto z ukazom, opisanim zgoraj, vključimo na svojo stran. Potrebovali bomo le še oznako `<param>`, s katero določimo morebitne parametre javanskega programčka.

OBRAZCI

Doslej smo spoznali skoraj vse osnove izdelovanja spletnih strani oziroma dokumentov HTML. Vključili smo tudi interaktivnost, a če dobro pomislimo, smo se doslej naučili informacije le posredovati. Obrazci so namenjeni temu, da lahko dobimo tudi povratno informacijo od obiskovalcev spletnih strani. Informacije, ki jih dobimo, so lahko poljubne. Običajno rezultate zapisujemo v zbirke podatkov, da jih pozneje lažje obdelamo z drugimi programskimi paketi (Microsoft Excel). Za to pa potrebujemo tudi nekaj predznanja skriptnih jezikov.

Na spletnih straneh srečujemo najrazličnejše oblike obrazcev, vsem pa je skupna bistvena lastnost: od obiskovalca spletne strani želijo dobiti povratno informacijo. To je lahko le uporabniško ime in geslo, ki ju moramo vtipkati, ali pa iskalnik po spletnih straneh, elektronska pošta ali kaj podobnega. Lahko napišemo celo zgodnico poljubne dolžine ali pa preprosto odključamo posamezne možne odgovore v spletni anketi.

Obrazec je del spletne strani, zgrajen iz elementov HTML. Vsak obrazec je sestavljen iz elementov, kot so gumbi, besedilna polja, meniji ali potrdilna polja (check box). Poglejmo si preprost zgled, sestavljen iz nekaj elementov.

```
<html>
<head>
<title>Primer</title>
<body>
<form method="post" action="www.spletniurl.si">
  <p>Vpiši svoje ime
    <input type="text" name="textfield">
  </p>
  <p>Vpiši svoj priimek
    <input type="text" name="textfield2">
```

```
</p>
<p>Vozniški izpit
  <input type="checkbox" name="checkbox"
value="checkbox">
</p>
<p>Moški
  <input type="radio" name="radiobutton"
value="radiobutton">
</p>
<p>Ženska
  <input type="radio" name="radiobutton"
value="radiobutton">
</p>
<p>Starost
  <select name="Starost">
    <option>20 - 30</option>
    <option>30-40</option>
    <option>&lt; 20</option>
    <option>&gt; 40</option>
  </select>
</p>
<p align="center">
  <input type="submit" name="Submit" value="
Pošlji">
  <input type="submit" name="Submit2"
value="Pobriši">
</p>
</form>
</body>
</html>
```



MOJ MIKRO



www.gambit.si

Microsoft
GOLD CERTIFIED
Partner

KODAK
PRINT LITHOGRAPHY

EuroTeam
Software
ALLIED BUSINESS

IBM
Business
Partner

Razvoj
poslovnih
rešitev za
obvladovanje
dokumentov,
poslovnih
procesov,
arhiviranje in
elektronsko
trgovanje.



DINAMIČNI HTML

DHTML ni nov jezik, temveč je le nekakšna nadgradnja jezika HTML. Pojavil se je z različico HTML 4.0. Lahko bi celo rekli, da so to nekakšni dodatki običajnemu jeziku HTML, ki omogočajo spreminjanje vsebin spletnih strani.

Vsebine strani pa ne spreminja spletni strežnik, kot je običaj, temveč je to posledica programske kode jezika DHTML. Takega spreminjanja vsebine strani smo bili vajeni le prek skriptnih jezikov Java in Visual Basic.

NAJPOGOSTEJŠI UČINKI DHTML

Naj naštejemo le nekaj najpogostejših učinkov, ki jih zmora DHTML. Objekt na spletni strani se lahko prosto giblje po strani. Podobno kot let čebele. Posamezna vsebina lahko pod določenimi pogoji izgine s strani. Spletne strani niso le dvorazsežne, saj so lahko posamezni objekti pred ali za drugimi. Vsebine strani se spreminja glede na klikanje miške. Posamezni deli strani se tako razkrijejo ali pa skrijejo. Oblikovanje spletnih strani je postalo precej lažje, saj je možno posamezne objekte na strani postaviti do točke natančno tja, kamor želimo, brez zapletenih tabel in točkovnih mašil. Na kratko pa bil lahko novosti strnili v pet točk.

- prekrivne (kaskadne) slogovne predloge, ki omogočajo spletnemu oblikovalcu, da na enem mestu določi vse značilnosti spletnih strani, ki temeljijo na tej predlogi;
- absolutno pozicioniranje, ki omogoča, da posamezni objekt postavimo na točno določeno mesto, do točke natančno;
- dinamična vsebina strani; pomeni, da se vsebina strani lahko spreminja tudi po tem, ko je že prenesena iz spletnega strežnika; strani ni potrebno na novo osveževati;
- posebne pisave, kar pomeni, da lahko spletni oblikovalec vključi na strani posebne pisave, ki se po potrebi prenesejo s stranjo iz strežnika k odjemalcu;
- povezava podatkov, pomeni, da so podatki v spletnem strežniku povezani z tabelo ali obrazcem. Stran se v tem primeru spreminja dinamično brez ponovnega izrisovanja celotne strani.

Lahko si predstavljate koliko možnosti ponuja zgornjih pet točk in koliko novosti to prinaša v spletno oblikovanje oziroma pisanje spletnih strani. Seveda pa vse stvari ne potekajo tako gladko, kot je morda videti na prvi pogled. Oba največja brskalnika Microsoft Internet Explorer in Netscape Navigator sta z

veseljem podprla novi standard. Zaradi odprtega standarda pa sta si zadeve predstavljala vsak po svoje, nekaj pa ima gotovo zraven tudi komercialna konkurenca. Tako uporabljata za iste zadeve različne ukaze in sintakso. Zmedo si lahko predstavljate. Tu pa tiči razlog, da je večina spletnih strani še vedno narejena po starem, kjer je pozicioniranje izvedeno z zapletenimi tabelami in ne s pomočjo plasti (layers).

Prekrivne slogovne predloge

Značilnost jezika HTML je, da sta oblika in vsebina tesno povezani. Za posamezne dokumente to morda niti ni tako slabo, a stvari postanejo precej nepregledne, če želimo tako oblikovati spletno stran. Vsebine običajno ne piše ista oseba, ki skrbi za obliko spletnih strani, zato se delo zamudno podvaja. S prekrivnimi slogovnimi predlogami (cascading style sheets) pa se je položaj precej spremenil. V slogu lahko določimo posamezne parametre oznak HTML z enim ukazom. To pa potem lahko velja za celotno spletno stran. Prekrivne sloge lahko vključujemo v dokument HTML na **tri načine**, in sicer kot:

- zunanjo datoteko; v tem primeru je to datoteka ASCII s podaljškom.css. Prednost uporabe tega načina je v tem, da lahko tako predlogo uporabimo v več različnih dokumentih. Klic datoteke poteka po naslednji sintaksi:

```
<link rel="stylesheet" href="predloga.css">
```
- del dokumenta HTML; na začetku dokumenta HTML napišemo vsa pravila, ki jih želimo določiti s predlogo:

```
<! BODY {background: blue; font-family: VERDANA;}
P {font-size: 12pt; margin-left: 10pt;}
H3 {color:orange;}->
```
- slogovno predlogo, ki vpliva samo na določeno oznako znotraj dokumenta. Poglejmo si preprost in kratek zgled prekrivne slogovne predloge, v kateri bomo določili barvo ozadja strani, družino pisave, velikost in poravnavo običajne pisave in barvo tretjega naslova (Heading 3):

```
<STYLE>
BODY {background: blue; font-family: VERDANA;}
```

```
P {font-size: 12pt; margin-left: 10pt;}
H3 {color:orange;}
</STYLE>
```

Morda ste opazili, da lahko velikost pisave določimo do pike (pt) natančno. Običajni HTML določa velikost pisave s številkami od 1-7. Podobno velja za velikost vrstice, ki jo tudi lahko spreminjamo. Seveda prekrivne slogovne predloge dajejo še ogromno drugih možnosti, ki jih pridobimo, če dodajamo svoje oznake v svoje slogovne predloge. To pa že presega okvir znanj, ki jih želimo podati v okviru te šole HTML. Bistveno je, da se s prekrivnimi slogovnimi predlogami poenoteno in sistematizirano lotimo urejanja oblike spletnih strani. Oblika je torej povsem ločena od vsebine, kar je velika prednost.

Absolutno pozicioniranje

Kdor se je vsaj malo ukvarjal z izdelovanjem spletnih strani, ve, kako težko je postaviti posamezen element spletne strani na točno določeno mesto. Običajno se je treba ozirati tudi na druge elemente spletne strani in tako lahko postane pozicioniranje prava mora. Običajno so oblikovalci reševali težave s tabelami, ki jih pozna vsak spletni brskalnik. Te tabele pa so lahko precej zapletene, vgnezdene in nepregledne, še zlasti, če opazujemo izvorno kodo. Poleg tega je treba vključevati slike velikosti ene točke, da pridemo do želenega rezultata.

DHTML je prinesel rešitev v plasteh. To so nevidne škatle, v katere lahko vstavimo poljubno spletno vsebino. Kar pa je najvažnejše, postavimo jih lahko v poljubno točko na strani. Vsaki plasti lahko določimo tudi nekaj parametrov, med katerimi je najuporabnejši nevidnost (invisible). Posamezne plasti lahko povezujemo v skupine, ki imajo svoje parametre, vsi člani skupine pa privzamejo parametre višje definirane skupine. S takim gnezdenjem lahko upravljamo tudi z več plastmi hkrati in le domišljija oblikovalca je meja med mogočim in nemogočim. V potrditev te teze naj omenim le to, da lahko s pomočjo plasti naredimo tudi preprosto animacijo. Plast lahko dinamično premikamo po zaslonu v določenem časovnem zaporedju. Tako dobimo občutek animacije, ki pa ni posledica animirane datoteke gif ali česa podobnega.

Dinamična vsebina strani

Zaenkrat podpira dinamično vsebino le Microsoft Internet Explorer. Razlog tiči v tem, da ima le Internet Explorer popoln nadzor nad modelom DOM (Document Object Model). Tako je lahko posamezna oznaka spremenjena kadarkoli.

Dinamična vsebina nam pride prav, ko prikazujemo določene vsebine. Na začetku prikazemo le naslov. Šele po kliku na naslov se nam prikaže še ostala vsebina. Ponoven klik pa vsebino skrije. Tako lahko listamo po kazalu in hkrati pregledujemo vsebino dokumenta.

Posebne pisave

Večina spletnih strani prikazuje svojo vsebino v osnovnih pisavah, kot so Arial, Times in Helvetica. Razlog je v tem, da te pisave delujejo v vseh sistemih. Če si spletni oblikovalec izbere drugačno pisavo, se nam lahko zgodi, da bo stran pri obiskovalcu spletne strani videti drugače, če ta nima nameščene tiste pisave v svojem osebem računalniku.

Podobno kot še marsikje drugod sta tudi v tem primeru Microsoft in Netscape izbrala vsak svojo, seveda diametralno pot, tako da stvari

ne delujejo, kot bi morale. Žal bomo morali na možnost prenašanja pisav prek spleta še nekoliko počakati. Rešitev, ki jo ponujata tekmeča, deluje za vsakega posebej, skupne rešitve pa žal še ni. Škoda.

Povezava podatkov

Povezati bazo podatkov s spletno stranjo ni preprosta zadeva. Običajno se povezava izvede v spletnem strežniku z enim izmed skriptnih jezikov. Prek jezika DHTML pa lahko povezujemo podatke tudi s tabelami in obrazci. Vsebinska pa se lahko spreminja dinamično. Zaenkrat le Microsoft Internet Explorer podpira možnost povezovanja podatkov. S skriptnim jezikom Java lahko dodajamo, brišemo in spreminjamo povezave v realnem času. Precej inovacij je tudi znotraj tabel in polj znotraj tabel. Med drugim se lahko tabela zgradi sama na podlagi podatkov, ki jih imamo, vendar to že presega okvir tega prispevka.

NEKAJ BESED OB KONCU

S tem se šola HTML v Mojem mikru končuje, za bralce pa se prava šola šele začneja. Podobno kot pri vsaki stvari tudi tukaj velja,

da vaja dela mojstra, in nasprotno, mojster dela vajo. Izkušnje se pridobivajo le z rednim delom in spremljanjem razvoja jezika. Veliko dela in preizkušanja, kako se določena zadeva naredi, je nujno potrebno. Veliko rešitev je moč najti tudi v svetovnem spletu, saj nima smisla, da bi odkrivali toplo vodo. Po drugi strani pa je dobro, da stvari, ki jih dobimo v spletu ali od prijatelja, tudi sami dobro preučimo, da bomo vedeli za drugič. Izkušnja več. S šolo HTML smo želeli bralcem podati le osnove in opozarjati na določene napake, ki se pogosto pojavljajo. Namenoma smo se izognili poglavju o postavitvi spletne strani v spletni strežnik. Tema je precej široka in obširna. Precej stvari je specifičnih in veljajo za točno določen strežnik, tako da bi težko napisali splošno navodilo. V tem primeru bi bilo vse skupaj preveč površno in nedorečeno.

Ob koncu bi se rad zahvalil vsem, ki ste šolo spremljali. Zahvaljujem se tudi vsem, ki ste se oglasili prek elektronskega naslova ali telefona za dodatna pojasnila in vprašanja. Želim vam veliko uspešnega dela pri izdelavi spletnih strani.



Tektronix
COLOUR PRINTERS BY
XEROX



Phaser 7700

22 barvnih strani/minuto

1200 dpi, A3 format

PowerPC G4 500 MHz procesor

128 MB RAM (max 512 MB)

5 GB trdi disk serijsko

duplex enota serijsko

CenterWare Web Server

PhaserMatch ICC 7700



First Look
Phaser 7700
10/2001



PANNA

družba informacijskih tehnologij

Kolezilska 7, 1000 Ljubljana

Telefon 01 429 2000

Internet www.panna.si