

UNIVERZA V MARIBORU
FAKULTETA ZA ELEKTROTEHNIKO,
RAČUNALNIŠTVO IN INFORMATIKO

Sara Tušar Suhadolc

ZNAČILNOSTI IN OMEJITVE SPLETNEGA OBLIKOVANJA

Diplomska naloga

Maribor, december 2008



UNIVERZA V MARIBORU



FAKULTETA ZA ELEKTROTEHNIKO,
RAČUNALNIŠTVO IN INFORMATIKO
2000 Maribor, Smetanova ul. 17

Diplomska naloga univerzitetnega študijskega programa

ZNAČILNOSTI IN OMEJITVE SPLETNEGA OBLIKOVANJA

Študent: Sara TUŠAR SUHADOLC
Študijski program: Univerzitetni, Medijske komunikacije
Smer: Interaktivna grafična komunikacija

Mentor: izr. prof. dr. Matjaž B. JURIČ
Somentor: doc. dr. Marjan DRUŽOVEC

Maribor, december 2008

ZNAČILNOSTI IN OMEJITVE SPLETNEGA OBLIKOVANJA

Ključne besede: spletno oblikovanje, značilnosti spleta, tipografija na spletu, podpora
transparence, cross-browser support, trendi spletnega oblikovanja

UDK: 004.774.6(043.2)

Povzetek

V diplomskem delu so opisani dejavniki, ki vplivajo na načrtovanje, oblikovanje in razvoj spletnih vsebin. Pasovna širina, zaslonska ločljivost in vrsta brskalnika so le nekatere izmed njih. Predstavljeni sta glavni tehniki zasnove spletnega mesta, še posebej izpostavljen pa je problem uporabe poljubne tipografije na spletu. Besedilo analizira podporo transparence, izpostavlja tehnologije za povečevanje interaktivnosti kot so Flash in Ajax, preučuje značilnosti in omejitve oblikovanja spletnih pasic in promocijskih e-sporočil, razkriva značilnosti Web 2.0, 3.0 in 4.0, za konec pa se poda v odkrivanje prihajajočih trendov spletnega oblikovanja. Diplomsko delo poleg pregleda dejavnikov, ki služijo kot izhodišče za uspešno spletno oblikovanje, izpostavlja nekatere aktualne probleme spleta in za njih ponuja možne rešitve.

CHARACTERISTICS AND LIMITATIONS OF WEB DESIGN

Ključne besede: web design, web characteristics, typography on the web, transparency support, cross-browser support, web design trends

UDK: 004.774.6(043.2)

Abstract

The final thesis describes factors (such as bandwidth, screen resolution, type of a browser, etc...), that influence the planning, designing and developing process of a web site. We compare two main web layout techniques and highlight the problem of using an optional font on the web. The work analyzes transparency support, points out the technologies used to increase interactivity, such as Flash and Ajax, and studies characteristics and limitations of banner and newsletter design. It exposes characteristics of Web 2.0, 3.0 and 4.0 and finishes with uncovering the future web design trends. Not only is the thesis an overview of the factors, that serve as a starting-point for successful web design, it is also highlighting some of the current web design problems for which it offers possible solutions.

KAZALO VSEBINE

1. UVOD	2
1.1. Metodologija	4
2. ZNAČILNOSTI SPLETA	5
2.1. Pasovna širina	6
2.1.1. Optimizacija slikovnega gradiva	7
2.1.2. Optimizacija kode	12
2.2. Ločljivost in barvna globina zaslonov	14
2.3. Format	18
2.4. Cross-browser support	21
2.4.1. Orodja za testiranje	23
3. ZASNOVA SPLETNE STRANI	26
4. POMEN TIPOGRAFIJE NA SPLETU	28
4.1. Primer uporabe zaloge pisav	35
4.2. Predstavitev skupin pisav	37
4.2.1. Sans-serifne pisave – pisave brez serifov:	37
4.2.2. Serifne pisave	38
4.2.3. Monospace – zaslonske pisave za pisanje kode	39
4.3. Tipografija na strežniku	40
4.4. Zamenjava s slikami	42
4.5. SiFR	43
5. PODPORA TRANSPARENCE	45
5.1. Transparentni (X)HTML elementi	45
5.2. Transparentne slike	47
5.2.1. Filter	47
5.2.2. Unit PNG fix	50
5.2.3. Twin Helix Behavior	50
6. IZBOLJŠANJE INTERAKTIVNOSTI	53
6.1. jQuery	53
6.2. MooTools	54
6.3. Flash	55
7. PROMOCIJSKA E-SPOROČILA	57
7.1. Omejitve	57
8. ZNAČILNOSTI OBLIKOVANJA SPLETNIH PASIC	60
8.1. Dimenzije	60
8.2. Omejitve velikosti	61
8.3. Verzija Flasha	62
8.4. Kreativna izraba pasic	62
9. WEB 2.0	64
9.1. Ajax	65
9.2. Oblikovne značilnosti Web 2.0	65
9.2.1. Živahne barve	66
9.2.2. Zaobljeni koti	66
9.2.3. Prehodi	66
9.2.4. Odsevi, sijaji na gumbih	67

9.2.5. Sence	67
9.2.6. Izpostavljeni elementi	67
9.2.7. Ikone.....	68
9.2.8. Visoke noge.....	68
9.2.9. Interaktivnost in dinamičnost.....	69
9.2.10. Preprostost.....	69
10. WEB 3.0, WEB 4.0	70
11. PRIHAJAJOČI TRENDI SPLETNEGA OBLIKOVANJA	73
11.1.1. Umirjene, nežne, pastelne barve.....	73
11.1.2. Veliko, še večje	74
11.1.3. Prehodi ali enotne barve	74
11.1.4. Tipografija.....	74
11.1.5. Video	74
12. PRAKTIČNI DEL	76
12.1. Naloga.....	76
12.2. Zahteve in cilji.....	77
12.2.1. Vsebinske zahteve	77
12.2.2. Tehnične zahteve.....	77
12.2.3. Poslovni cilji.....	77
12.2.4. Komunikacijski cilji	77
12.3. Izhodišča za oblikovanje	78
12.3.1. Ime penziona	78
12.3.2. CGP	78
12.3.3. Načrt strukture vsebin	78
12.4. Rezultati.....	79
12.4.1. Pregled uporabljenih tehnologij in principov oblikovanja.....	79
12.4.2. Opis postopka izvedbe in diskusija rezultatov	80
13. SKLEP.....	84
LITERATURA	86
PRILOGE	88
Statistični podatki.....	88
Prvo spletno mesto	88
Drugo spletno mesto.....	90
Tretje spletno mesto	91

SEZNAM SLIK

Slika 2.1: Test slikovnih formatov	10
Slika 2.2: Test slikovnih formatov	11
Slika 2.3: Barvna globina zaslonov (globalna raven, avgust 2008)	15
Slika 2.4: Barvna globina zaslonov (Slovenija, avgust 2008).....	15
Slika 2.5: Zaslonske ločljivosti (globalna raven, avgust 2008).....	16
Slika 2.6: Zaslonske ločljivosti (Slovenija, avgust 2008)	17
Slika 2.7: Prikaz višine 550 pixlov.....	18
Slika 2.8: Ozadja HTML dokumenta	19
Slika 2.9: Podatki o razširjenosti brskalnikov (globalna raven, avgust 2008)	22
Slika 2.10: Podatki o razširjenosti brskalnikov (Slovenija, avgust 2008).....	22
Slika 4.1: Primer slabo izbrane zaloge pisav	35
Slika 4.2: Primer slabo izbrane zaloge pisav	36
Slika 4.3: Primer slabo izbrane zaloge pisav.....	36
Slika 4.4: Arial, Verdana in Trebuchet MS.....	37
Slika 4.5: Lucida Grande, Century Gothic in Tahoma.....	37
Slika 4.6: Calibri, Candara in Corbel	37
Slika 4.7: Arial Black, Comic Sans MS in Impact.....	38
Slika 4.8: Times New Roman, Times in Georgia	38
Slika 4.9: Book Antiqua, Palatino	38
Slika 4.10: Cambria in Constantia.....	39
Slika 4.11: Andale Mono, Courier New in Courier	39
Slika 4.12: Consolas, Lucida Console in Monaco	39
Slika 4.13: Zamenjava besedila s slikami	42
Slika 5.1: Transparenca HTML elementa	46
Slika 5.2: Transparenca HTML elementa	46
Slika 5.3: Prikaz v IE7 in Firefox-u	48
Slika 5.4: Prikaz v IE6.....	48
Slika 5.5: Transparenca pri ponavljajočem slikovnem formatu PNG.....	49
Slika 6.1: jQuery primeri.....	54
Slika 6.2: MooTools primeri	54
Slika 9.1: Oblikovne značilnosti Web 2.0: Barve	66
Slika 9.2: Oblikovne značilnosti Web 2.0.....	67
Slika 9.3: Oblikovne značilnosti Web 2.0.....	68
Slika 9.4: Oblikovne značilnosti Web 2.0.....	68
Slika 11.1: Prihajajoči trendi spletnega oblikovanja: barve	73
Slika 12.1: Stara spletna predstavitev in znak penziona	76
Slika 12.2: vsebinska zasnova in navigacijski model.....	78
Slika 12.3: Oblikovni elementi spletnega mesta Vitapark	80
Slika 12.4: Oblikovno / tehnični elementi spletnega mesta Vitapark	81
Slika 12.6: Prva tipična stran: naslovnica	82
Slika 12.7: Druga tipična stran: podstran	83

SEZNAM PREGLEDNIC

Tabela 2.1: Test slikovnih formatov	10
Tabela 2.2: Barvna globina zaslonov (globalna raven, avgust 2008)	14
Tabela 2.3: Barvna globina zaslonov (Slovenija, avgust 2008)	15
Tabela 2.4: Zaslonske ločljivosti (globalna raven, avgust 2008)	16
Tabela 2.5: Zaslonske ločljivosti (Slovenija, avgust 2008)	17
Tabela 2.6: Podatki o razširjenosti brskalnikov (globalna raven, avgust 2008)	22
Tabela 2.7: Podatki o razširjenosti brskalnikov (Slovenija, avgust 2008)	22
Tabela 4.1: Pisave različnih operacijskih sistemov	34
Tabela 4.2: Zaloge pisav	40
Tabela 7.1: CSS podpora v različnih poštnih odjemalcih	59

SLOVAR TUJK

Spletni brskalnik – web browser

Vtičnik – plugin

Promocijsko e-obvestilo – newsletter

Zaslonski bralnik – screen reader

Postavitev – layout

Lastnost – property

Vrednost – value

Element – tag

HTTP Zahteva – HTTP request

Spletni pajek – Web Crawler

SLOVAR KRATIC

(X)HTML – (Extensible) Hypertext Markup Language

CSS – Cascading Style Sheets

SiFR – Scalable Inman Flash Replacement

MooTools, jQuery, Prototype, ... – Object oriented JavaScript libraries

CMS – Content Management System

SEO – Search Engine Optimization

1. UVOD

Splet je medij nešteti možnosti, je forum izmenjevanja znanja, prožilec političnih in socialnih sprememb, nepogrešljiv vir informacij in nenazadnje prostor, ki posamezniku omogoča skoraj neomejeno kreativno izražanje. Zaradi spleta so se korenito spremenile naše navade, način dela in izobraževanja, zato lahko z gotovostjo rečemo, da je splet eden izmed večjih izumov človeštva nasploh.

Splet je medij z najhitreje rastočim številom uporabnikov, kar vedno bolj privlači oglaševalce, saj temu mediju iz leta v leto namenijo večji delež oglaševalskega kolača. The Interactive Advertising Bureau (IAB) poroča, da je bilo za spletno oglaševanje v prvi četrtini leta 2008 porabljenih za 18,2% več sredstev kot v enakem obdobju lani [1]. Po poročanju MediaPosta pa je spletno oglaševanje v nekaterih delih Evropske unije celo že dohitelo in naj bi v naslednjem letu tudi preseglo televizijskega [2].

O pomembnosti spleta priča tudi izjava Kennetha Goldsmitha: »*If It Doesn't Exist on the Internet, It Doesn't Exist*« [3]. In res je, splet je nedvomno postal medij z velikim vplivom, ki ga sodobna podjetja ne morejo več ignorirati. *Biti ali ne biti prisotni na spletu danes sploh ni več vprašanje. Bolj smiselno se je vprašati: kako načrtovati in oblikovati spletne vsebine, da bi karseda učinkovito in kreativno izkoristili potencial tega medija?* Naloga ni tako zelo preprosta kot se morda zdi na prvi pogled. Splet je medij s specifičnimi lastnostmi in omejitvami, ki izhajajo iz njih. Oblikovanje vsebin za druge medije se močno razlikuje od oblikovanja spletnih vsebin, kjer veljajo povsem drugačne zakonitosti. Pri oblikovanju za tisk smo na primer omejeni s številom barv, kvaliteto tiska, papirjem, itd, pri oblikovanju spletnih vsebin pa naletimo na povsem druge omejitve, ki jih je, po mojih izkušnjah, več kot pri oblikovanju za katerikoli drug medij. Nekdo, ki je izjemno dober oblikovalec, še ni nujno dober spletni oblikovalec in obratno. Zavedati se moramo, da je splet medij s posebnimi zmožnostmi in omejitvami, zato nanj ne moremo neposredno prenašati principov oblikovanja s področja tiska, televizije, itd. To seveda ne pomeni, da iz teh področij ne moremo črpati inspiracij, le izvedba mora biti premišljena in prilagojena ciljnemu mediju.

Kako se torej spletno oblikovanje loči od klasičnega oblikovanja? Kaj vse je treba upoštevati pri oblikovanju spletnih vsebin? Lahko na spletu uporabimo poljubno tipografijo? Kako so podprti transparentni elementi? S kakšnimi omejitvami se srečujemo in kako vse to vpliva na našo kreativnost?

Da bi lahko čim boljše odgovorili na zgornja in podobna vprašanja, si bomo v diplomskem delu postavili naslednje cilje:

- Posvetili se bomo karakteristikam in omejitvam spleta,
- pogledali kako sta podprti tipografija in transparenca,
- pregledali in primerjali tehnologije, ki omogočajo večjo interaktivnost,
- se posvetili karakteristikam oblikovanja spletnih pasic in promocijskih e-obvestil,
- ovrednotili in primerjali opisane pristope v povezavi z Web 2.0,
- se zazrli v prihodnost Web-a 3.0 in 4.0. in
- poizkusili predvideti prihajajoče trende spletnega oblikovanja.

Omejili se bomo predvsem na raziskovanje značilnosti spleta z vidika spletnega oblikovalca, načrtovalca in delno razvijalca. V času spletnih aplikacij in orodij, ki so izredno enostavna za uporabo (brezplačna CMS orodja) in vse večje računalniške pismenosti se mnogo posameznikov znajde v vlogi oblikovalca, načrtovalca in razvijalca v enem. Za produciranje kvalitetnih spletnih vsebin danes ne potrebujemo več celotne ekipe strokovnjakov, kot je bila to praksa včasih. Četudi so naloge kot so načrtovanje, oblikovanje in implementacija ločene, je za spletnega oblikovalca, kot tudi načrtovalca, obvezno dobro poznavanje tehničnega ozadja. Le z natančnim vpogledom v zmožnosti in omejitve spleta na vseh ravneh produkcije lahko ustvarimo presežne spletne vsebine.

Diplomska naloga bo služila kot odgovor na vprašanje *kaj je v tem trenutku možno narediti*, delno tudi *kako priti do rešitev nekaterih aktualnih problemov spleta*, s katerimi se srečujem že vrsto let, še bolj skoncentrirano pa v zadnjem letu dela v industriji (Innovatif, skupina Futura).

Želim vam prijetno branje.

1.1. Metodologija

Metoda raziskovanja je pregled literature in obstoječih primerov kot tudi razvoj lastnih metod in testiranja na računalnikih z različno programsko in strojno opremo.

Metodologija zbiranja podatkov za statistične primere:

Podatki o globalni razširjenosti zaslonih ločljivosti in brskalnikov za mesec avgust 2008 izhajajo iz vseh spletnih mest na katere je umeščen [w3counter](#) (kar zajame 13,708 spletnih mest in 27.797.911 unikatnih obiskovalcev mesečno). Podatke o globalni barvni globini zaslonov sem pridobila iz strani thecounter.com, tudi zelo razširjenem števcu obiskov spletnih strani.

Podatke za zaslonske ločljivosti, brskalnike in barvno globino zaslonov za slovenski prostor sem zbrala sama. Izbrala sem tri dobro obiskana in po tematiki dovolj splošna spletna mesta, tako da bi zbrane statistike čim boljše odsevale celoten slovenski prostor. Prvo spletno mesto predstavlja eno izmed največjih slovenskih mest, drugo popularno radijsko postajo, tretje pa izredno dobro obiskano slovensko gledališče.

Metodologija testiranja:

Za testiranje različnih slikovnih formatov (JPG, PNG, PNG8 in GIF) sem uporabila orodje Adobe Photoshop na operacijskem sistemu Windows XP.

Za testiranje izgleda spletnih mest v različnih brskalnikih sem uporabila: Mozilla Firefox 2.0 in 3.0, Internet Explorer 7 in »stand-alone« verzijo Internet Explorer 6, kot tudi aplikacijo IE tester, ki je zmožna prikazovanja spletnih strani v verzijah IE5, IE5.5, IE6.

2. ZNAČILNOSTI SPLETA

V poglavju karakteristike spleta bomo predstavili značilnosti, ki so specifične za splet in vplivajo na proces snovanja spletnih vsebin, zaradi česar bi jih moral poznati vsak načrtovalec, kot tudi spletni oblikovalec. Pri načrtovanju se je pomembno zavedati, da na prikaz našega izdelka vpliva kopica različnih dejavnikov.

Eden izmed glavnih problemov pri oblikovanju spletnih strani se glasi: *kako zasnovati in oblikovati spletno stran, ki bo pri vseh odjemalcih pravilno delovala in kar je še težje izgledala kar se da podobno*. Popolnoma enak izgled je zaradi več dejavnikov skoraj nemogoč in morda tudi nesmiseln, a več o tem kasneje.

Že samo dejstvo, da obstaja množica spletnih brskalnikov (pet glavnih in vse njihove verzije, ki so v uporabi) nam zelo oteži zgoraj zastavljeno nalogo, da sploh ne omenjamo hitrosti povezave, različnih ločljivosti zaslonov, št barv, ki so jih sposobni prikazati, nastavitve v brskalnikih, zmogljivosti strojne opreme, itd... Ko snujemo za splet, vsebino ponavadi prilagodimo tako, da jo bo optimalno videla večina ljudi, še vedno pa mora biti uporabna tudi za tiste s starejšo programsko in strojno opremo, počasno povezavo itd.

V nadaljevanju bomo videli, da je naša kreativnost v mnogih primerih odvisna od omenjenih značilnosti in omejitev, ki iz njih izhajajo. Omejitve niso vedno slaba stvar, saj določajo okvir našega delovanja. Občasno se nam posreči prebiti izven okvirov in takrat obstaja velika možnost, da smo ustvarili presežen izdelek. Snovanje in izvedba spletnih vsebin je mnogokrat težavno in naporno, vendar vedno dinamično in vznemirljivo. Zavedati se je treba, da spletno oblikovanje ni zgolj risanje lepih podob ampak reševanje konkretnih problemov.

2.1. Pasovna širina

Pasovna širina je ena izmed ključnih lastnosti spleta, ki jo je treba upoštevati pri načrtovanju in izvedbi spletnih vsebin. Izraža relativno hitrost internetne povezave oziroma koliko podatkov je mogoče prenesti v nekem časovnem obdobju. Najpogosteje je izražena v kilobitih na sekundo (kbit/s, kbps) [4]. Večja kot je pasovna širina, več podatkov lahko prenesemo v nekem časovnem obdobju. Povedano drugače, večja kot je pasovna širina, hitreje se bo naložila neka spletna, kar seveda pozitivno vpliva na uporabniško izkušnjo.

Pasovne širine so iz leta v leto večje - predstavniki direktorata za elektronske komunikacije so februarja 2007 predstavili program gradnje odprtih širokopasovnih omrežij. Med cilji projekta je zapisano, da bodo imeli do leta 2010 vsi končni uporabniki v Sloveniji možnost dostopa do širokopasovnih povezav, do leta 2020 pa naj bi imelo 90 odstotkov prebivalstva optične povezave do doma [5]. Kljub vedno večjim pasovnim širinam pa moramo pri načrtovanju upoštevati tudi tiste uporabnike, ki do spleta še vedno dostopajo s počasnimi »dial-up« povezavami, morda z mobilnimi napravami. Raziskava, ki sta jo leta 2006 izvedla Akamai and JupiterResearch kaže, da naj bi bil najdaljši priporočljiv čas nalaganja spletnega mesta pičle 4 sekunde [6]. Po tem času naj bi se mnogi uporabniki naveličali čakanja in spletno mesto zapustili. Včasih, ko so bile povezave počasnejše, smo bili uporabniki navajeni počakati malo dlje, sedaj pa postajamo vedno bolj razvajeni, poleg tega pa je tudi spletnih vsebin toliko, da stvar, ki jo iščemo, zlahka najdemo tudi drugje. Sama sem bila že v postopku nakupa zaslona prek spleta in ker potrditveno sporočilo o registraciji v poštni nabiralnik ni prišlo v roku nekaj minut, sem nakup opravila drugje, spletni trgovini pa sporočila na kakšen način izgubljajo stranke.

Pasovna širina uporabnika je nekaj na kar nimamo vpliva. Vse kar lahko naredimo, da bi pripomogli k boljši uporabniški izkušnji, je optimizacija elementov spletne strani (slikovno gradivo, izvorno kodo, CSS datoteke). Najbolje je, da je skupna velikost datotek naše spletne predstavitev in število HTTP zahtev kar se da majhno.

2.1.1. Optimizacija slikovnega gradiva

Slikovno gradivo je priporočljivo pripraviti na tak način, da bo le to zasedlo čim manj prostora, obenem pa še vedno izgledalo dovolj dobro in s tem služilo svojemu namenu. Obstaja mnogo formatov za shranjevanje slikovnega materiala, nekaj izmed njih pa je takšnih, ki so še posebej primerni za objavljanje na spletu. Slikovni elementi so ponavadi največji porabniki pasovne širine, zato iščemo takšne formate, ki jih odlikuje najboljše razmerje med kompresijo in kvaliteto. Vsak izmed formatov, ki jih bomo predstavili v nadaljevanju je bil narejen za točno določen namen. Če ne vemo katerega izbrati, lahko po nepotrebnem izgubimo na kvaliteti ali pridobimo na velikosti datoteke. *Katerega izmed formatov torej izbrati, če želimo shraniti logotip? Katerega za shranjevanje fotografije? Katerega za grafiko s transparentnim ozadjem?*

JPG

JPEG ali JPG (Joint Photographic Experts Group) je rastrski slikovni format, ki je najpogosteje uporabljen na spletu. JPEG format ponavadi uporablja izgubno (baseline lossy) kompresijo, kar pomeni da določene informacije v sliki izgubimo. S tem ko sliko shranimo v JPEG format lahko dosežemo 3 ali celo 100 krat manjšo velikost datoteke v primerjavi z brezizgubnim formatom BMP [7]. Če s stopnjo kompresije, ki se giblje med 1 in 100, ne pretiravamo, razlika med originalom in kompresirano sliko s prostim očesom ni opazna.

JPEG algoritem s katerim kompresiramo sliko je dokaj zapleten. Vključuje lahko postopke kot so:

- rezanje višje-frekvenčnih komponent
- zmanjševanje barvnih komponent (downsampling)
- kodiranje po metodi ponavljajočih vrednosti (Run Length Encoding)
- kodiranje po Huffmanu
- ...

[8]

Format JPEG je idealen za shranjevanje fotografij, za kar je bil tudi narejen. Podpira namreč 24 bitno barvno globino, kar pomeni okoli 17 milijonov različnih barvnih odtenkov.

Format JPG, poleg normalnega zapisa, podpira tudi zapis v progresivnem načinu, ki v brskalniku omogoča predogled slike v slabši kvaliteti, še preden se ta popolnoma naloži.

GIF

Format GIF (Graphics Interchange Format) uporablja brezizgubno LZW (Lempel-Ziv-Welch) kompresijo in podpira 8 bitno barvno globino, kar pomeni, da lahko v sliki prikažemo 256 različnih barvnih odtenkov [9]. Format GIF temelji na indeksiranju točkovnih elementov slike, kar pomeni da se podatek o barvi ne zapiše za vsako točko posebej (kot pri formatu BMP) ampak se ustvari tabela barv, ki so prisotne v sliki. Vsaki točki se nato dodeli številko (index), ki označuje kje v tabeli se zapis nahaja. Na ta način lahko zelo zmanjšamo velikost slik, ki vsebujejo majhno število barv – tem manj je barv tem večja je kompresija. Format GIF se torej uporablja za shranjevanje slik, v katerih ni več kot 256 barv ali odtenkov sivine, na primer logotipov in tehničnih risb, kjer je pomembno ohraniti čiste linije in robove ploskev. Če bi za shranjevanje logotipa uporabili format JPG bi pri povečavi opazili zmazke pri mejnih ploskvah in linijah.

Format GIF podpira shranjevanje slik s prozornim ozadjem, česar format JPEG ne. Transparenca je dosežena z definiranjem barve v tabeli, ki ni vidna.

Slike v tem formatu lahko z ustrezno programsko opremo zložimo v zaporedje in jih shranimo kot animacijo – animiran GIF. Uporabe tovrstnih animacij, ki naj bi služile kot popestritev spletne strani ne priporočamo, saj so lahko izjemno nadležne. Še posebej se izogibajmo animacijam, nastavljenim kot ponavljajoče ozadje dokumenta. V preteklosti je kar mrgolelo spletnih strani, ki so bile polne takšnih elementov:

http://users.nac.net/falken/annoying/ani_gif.html

Kreativno uporabo tovrstnega elementa sem zasledila na starem portalu Marketing Magazin, kjer so ga domiselno uporabili za označevanje sklopov. Stran je delno še vedno dosegljiva na naslovu: <http://web.archive.org/web/20021128120756/www.mmportal.delo.si/> Ikone, ki označujejo sklope se ob dotiku kurzorja zamenjajo z animirano sliko:

<http://www.mmportal.delo.si/images/desno/avodnik.gif>

http://www.mmportal.delo.si/images/desno/avodnik_over.gif

Animiran GIF bi načeloma lahko uporabili za kreiranje enostavnih spletnih pasic, vendar danes za to nalogo uporabljamo Adobe Flash, ki je sposoben proizvesti datoteke manjših velikosti.

PNG

Začetek razvoja formata PNG (Portable Network Graphic) sega v leto 1995. Nastala je potreba po novem formatu z večjo barvno globino, ki bi nadomestil GIF. V tistem času so se namreč začeli pojavljati računalniški zasloni, ki so bili zmožni prikazati več kot 256 barv, poleg tega pa se je izvedelo, da je Unisys patentiral LZW algoritem, ki se je do tedaj uporabljal pri kompresiji formata GIF. Format PNG podpira 24 bitno barvno globino (lahko tudi 8 bitno) za stiskanje pa se uporablja nepatentirana, brezizgubna metoda kompresije DEFLATE, ki izvira iz LZ77, ki se uporablja za kompresijo ZIP datotek. Poleg metode DEFLATE se nad vrstico točk izvaja predvidevanje vrednosti točke na podlagi vrednosti prejšnje vrstice [10].

Format PNG s posebnim alpha kanalom podpira 8 bitno transparenco, kar pomeni, da točki lahko priredimo vrednost vidnosti. Vsaka točka je lahko vidna od 0 do 100% (opacity), kar odpravi nazobčane robove prisotne pri formatu GIF, ki s svojo 2 bitno transparenco omogoča zgolj vrednosti vidna ali nevidna.

Format PNG uporabljamo tako za shranjevanje slik z majhnim številom barv (8 bitni PNG), kot tudi fotografij. 8 bitni PNG je v nekaterih primerih sposoben večje kompresije kot format GIF.

Za nadomestitev animiranih GIF-ov je bil leta 2001 razvit format MNG (Multiple-image Network Graphics), ki deluje na podoben način kot PNG, vendar njegova uporaba še vedno ni razširjena. Format podpira brskalnik Konqueror in novejši Sonny Ericson telefoni. Nekoč je obstajala podpora tudi v Mozillinih brskalnikih, vendar so jo kasneje zaradi optimizacije kode in premajhne uporabe umaknili [11].

Test slikovnih formatov

S spodnjim testom bomo na primerih poizkusili prikazati kateri izmed formatov je najprimernejši za shranjevanje različnih motivov. Za prvi primer smo vzeli fotografijo barvic velikosti 493x321px in jo shranili v formate JPG, GIF, PNG (8bit) in PNG (24bit). Prav tako smo storili z logotipom formata 300x300px.

V spodnji razpredelnici lahko vidimo podatke o velikostih slik in nastavitvah s katerimi so bile shranjene (razen pri 24 bitnem PNG-ju, kjer ni možnosti kompresije).

	JPG	PNG(24bit)	PNG (8bit)	GIF
logotip	4,265 KB (kvaliteta 18)	13,71KB	4,18KB (32 barv)	4,32 KB (32 barv)
fotografija	37,67KB (kvaliteta 67)	219,14KB	62,93KB (256 barv)	74,51KB (256 barv)

Tabela 2.1: Test slikovnih formatov

Na spodnjih primerih je prikazana razlika v izgledu izrezov testiranih slik:



Slika 2.1: Test slikovnih formatov

Vidimo lahko, da je za shranjevanje slike z majhnim številom barv, kot je na primer logotip, ikona ali tehnična risba najmanj primeren format JPG. Na ploskvah so namreč opazni nekakšni zmazki na robovih ploskev pa je videti oblake pik¹. Sliki shranjeni v formatih PNG

¹ Če bi sliko shranili z nižjo stopnjo kompresije zmazki ne bi bili tako zelo očitni, bi pa se povečala velikost datoteke. Stopnja kompresije je bila nastavljena tako, da smo dobili približno enake velikosti datotek JPG, GIF in PNG8, s čimer smo bolj nazorno prikazali razliko med formati.

(8bit) in GIF se po videzu med seboj v ničemer ne razlikujeta, kar ni presenetljivo saj sta obe shranjeni z enakim številom barv. Slika shranjena v formatu PNG (24bit) je od njiju za malenkost kvalitetnejša, kar je moč opaziti šele pri povečavi slike. Zadnje tri slike (PNG, PNG (8 bit) in GIF) izgledajo identično, zato je smiselno izbrati tisto, ki zasede najmanj prostora. V našem primeru je to slika shranjena v formatu PNG (8bit), ki je za malenkost (za 140B) manjša od slike formata GIF.

V primeru shranjevanja fotografije je situacija nekoliko spremenjena. Sliki shranjeni v formatih GIF in PNG (8 bit) sta se odrezali najslabše, saj je med njima in ostalima dvema videti opazno razliko v kvaliteti. Na območjih, kjer so prisotni prehodi med barvami lahko vidimo pike velikosti 1x1px. Med slikama formatov PNG in JPG je opaziti minimalno razliko v kvaliteti, vendar ta ni vredna dodatnih 181,47KB kolikor je razlika v velikosti datoteke.



Slika 2.2: Test slikovnih formatov

Zaključimo lahko, da je za shranjevanje logotipov, ikon ali tehničnih risb najprimernejši format PNG (8 bit) za shranjevanje fotografij pa nedvomno JPG. 8-bitni PNG je odlična zamenjava formata GIF saj ob enaki kvaliteti postreže z malenkost manjšo velikostjo datoteke.

Da bi karseda zmanjšali čas nalaganja spletne strani moramo torej izbrati pravilne formate za shranjevanje slikovnega materiala. Poleg tega je potrebno zmanjšati tudi število HTTP zahtev. Več slik se nalaga dalj časa kot ena, ki je enako velika kot vse majhne skupaj [12]. Iz tega dejstva izhaja priljubljen princip združevanja več slik v eno samo in njihovo prikazovanje s CSS pozicioniranjem. CSS sprites, kot se imenuje tehnika, je odlična za zmanjševanje HTTP zahtev, poleg tega pa jo lahko uporabimo tudi za »rollover« efekte, za katere smo poprej potrebovali JavaScript.

```
#gumb a {  
    background: url(gumb.gif) 0 0 no-repeat;}  
  
#gumb a:hover {  
    background: url(gumb.gif) 0 -50px no-repeat;}
```

2.1.2. Optimizacija kode

Za čim hitrejšo nalaganje naše spletne strani, je poleg optimizacije grafičnih elementov, ključna tudi optimizacija kode. Z leti popravljanja in dopolnjevanja se v naši kodi lahko nabere kopica nepotrebnih elementov, ki le upočasnijo nalaganje spletne strani in kar je še huje, poslabša se lahko celo rangiranje v spletnih iskalnikih. Kodo je potrebno optimizirati tudi v primeru uporabe WYSIWYG (What You See Is What You Get) urejevalnikov kot sta Frontpage, Dreamweaver in drugi. Takšna orodja nam mnogokrat olajšajo proces izgradnje spletne strani, vendar v mnogih primerih v kodo zapišejo veliko redundantnih elementov. Optimizirati je treba HTML kot tudi CSS kodo.

Nekaj nasvetov kako do optimalne (X)HTML kode:

- Ne uporabimo več elementov kot jih potrebujemo
- Počistimo komentarje
- Počistimo odvečne presledke
- Za določanje izgleda besedila v nobenem primeru ne uporabimo elementa z pripadajočimi atributi face, color in size. To vse nadomestimo s CSS atributi font-family, color in font-size
- Za oblikovanje na splošno uporabljajmo CSS
- Izogibajmo se prekomernemu gnezdenju HTML elementov
- Izogibajmo se uporabi tabel, razen ko gre za tabelarične podatke
- CSS in JavaScript shranimo v zunanji dokument. Za najvišjo stopnjo optimizacije vse CSS dokumente združimo v en dokument, enako storimo z JavaScript datotekami, saj na tak način zmanjšamo število HTTP zahtev

- Slikam določimo višino in širino. S tem pohitrimo nalaganje spletne strani, saj spletni brskalnik dobi podatke o dimenziji slike, brez da bi jo moral naložiti. Tako se dokument lahko »renderira« naprej, slike pa se prikažejo, ko se naložijo [13]
- Vedeti moramo, da se med nalaganjem <script> elementov nalaganje vsega drugega preneha [12]. Če je le možno te elemente postavimo pred zaključkom elementa <body> (skripte za štetje števila obiska, itd..)

Kako do optimalne CSS kode?

- Izogibajmo se ponavljanju. Če vemo, da bo na spletni strani prevladovala določena pisava jo s CSS-om določimo najvišjemu elementu v strukturi (html ali body) in ne vsakemu elementu posebej. Dedovanje lastnosti je glavna značilnost CSS-a, izkoristimo jo.
- Imena class-ov naj ne bodo predolga, vendar smiselna. Ime naj opisuje funkcijo elementa, ne njegovega videza, saj se ta lahko spremeni. [14]
- Izogibajmo se prevelikemu številu selektorjev. Vsak element ne potrebuje svojega selektorja, saj se po njem lahko sklicujemo glede na prejšnji element, ki mu je določen selektor.
- Po elementu se sklicujemo po najkrajši poti. Pri sklicevanju #ovoj #desna_stran #vsebina p sta odveč #ovoj in #desna_stran.
- Če se le da barve zapisujemo na krajši način. #222222 je enako kot #222.
- Pri lastnostih margin, padding, background, font in border združujemo vrednosti. Margin-top:5px; margin-right:10px; margin-bottom:5px; margin-left:10px; je enako kot margin: 5px 10px.
- Zavedajmo se možnosti uporabe več class-ov na enem elementu (class='nav left') in združujemo ponavljajoče lastnosti.
- Nekateri zagovarjajo pisanje CSS atributov v eni vrstici, vendar je to odločitev posameznika. Sama menim, da je tak način pisanja veliko bolj nepregleden.

2.2. Ločljivost in barvna globina zaslonov

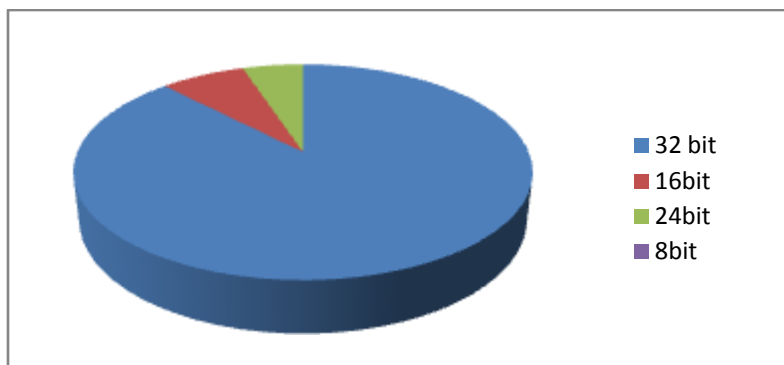
Ločljivost zaslona je za spletnega oblikovalca še ena neznanka in hkrati posebnost spletnega oblikovanja. Pri oblikovanju tiskovin je natančno znano na kakšni dimenziji bo sporočilo videno, pri pripravi TV materiala nam je znano vsaj razmerje (4:3 ali 16:9). *Več kot imamo podatkov večjo kontrolo imamo nad izgledom. Kako pa je s tem na spletu?*

Pri oblikovanju spletnih vsebin, moramo vsebino pripraviti tako, da bo dobro izgledala na različnih zaslonih z različnimi ločljivostmi in razmerji stranic. Prav tako se moramo zavedati, da različni zasloni različno prikazujejo barve, imajo drugače nastavljen kontrast, svetlost, itd. Včasih gre lahko za majhne razlike, spet drugič za ogromne. Sama sem imela dokaj neprijetno izkušnjo z izdelavo spletne strani, ki se je na zaslonih naročnika ne le videla popolnoma drugače, ampak je bila skoraj neuporabna. Meni, ki smo ga predvideli v svetlo sivi barvi, se na zaslonih naročnika sploh ni prikazal. Vzrok, kot smo izvedeli kasneje je bil ta, da so bili njihovi cenejši zasloni sposobni prikazati veliko manjše število barv kot običajni. Ker je bila naša siva nekje vmes med belo in prvim odtenkom sive, ki jo je bil zaslon zmožen prikazati, se je ta vrednost zaokrožila na belo in tako smo dobili povezave, ki so lebdele v zraku. Da bi se izognili takšnim situacijam moramo poskrbeti, da bo naša spletna predstavitev dovolj kontrastna in tako uporabna na kateremkoli zaslonu. Pa si pogledjmo kakšna je statistika razširjenosti zaslonov različnih barvnih globin:

Barvna globina zaslonov (globalna raven, avgust 2008)

32bit (16.8 milijona barv + alpha kanal)	86%
16bit (65.534 barv)	7%
24bit (16.8 milijona barv)	5%
8bit (256 barv)	0%

Tabela 2.2: Barvna globina zaslonov (globalna raven, avgust 2008) [15]

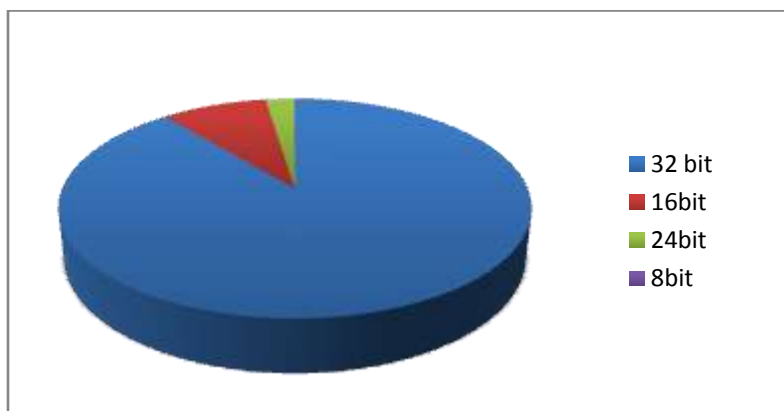


Slika 2.3: Barvna globina zaslonov (globalna raven, avgust 2008)

Barvna globina zaslonov (Slovenija, avgust 2008)

32bit (16.8 milijona barv + alpha kanal)	89%
16bit (65.534 barv)	9%
24bit (16.8 milijona barv)	2%
8bit (256 barv)	0%

Tabela 2.3: Barvna globina zaslonov (Slovenija, avgust 2008)



Slika 2.4: Barvna globina zaslonov (Slovenija, avgust 2008)

Iz zgornjih statistik lahko vidimo, da so tako po svetu, kot tudi pri nas najbolj razširjeni zasloni, ki podpirajo 32 bitno barvno globino (pri nas za 3% več kot po svetu), sledijo jim zasloni s 16 bitno barvno globino (teh je pri nas za 2% manj kot po svetu), nato tisti s 24 bitno barvno globino in zanemarljivo število takšnih, ki so zmožni prikazati zgolj 256 barv. V zadnjo skupino štejemo uporabnike, ki uporabljajo zastarele zaslone kot tudi tiste, ki so operacijski sistem Windows zagnali v varnem načinu.

Ko govorimo o ločljivosti zaslonov moramo vedeti, da le teh obstaja ogromno. Poznamo zaslone različnih velikosti, različnih formatov, takšne, ki imajo razmerje stranic 4:3 (zastarela 640x460 in 800x600 pa vse do 1920x1200) in široke »widescreen« zaslone, ki prevladujejo predvsem na prenosnih računalnikih (1280x800, 1152x864, ..), v zadnjem času pa tudi na namiznih zaslonih. Upoštevati je treba tudi vse večje število mobilnih naprav, kot so mobilni telefoni in dlančniki, s katerimi uporabniki vse pogosteje dostopajo do spleta. Kot oblikovalci ne moremo vedeti s kakšno zaslonsko ločljivostjo si obiskovalec ogleduje naš izdelek, saj le teh obstaja več kot 25! Zanesemo se lahko le na statistiko razširjenosti različnih ločljivosti in se na podlagi tega odločimo kako zasnovati spletno stran. Ločljivosti, ki so v uporabi so kriterij predvsem za odločitev o širini spletne strani, če je le-ta fiksna. Lahko se odločimo tudi za fluidno postavitev, kjer se vsebina spletne strani razteza s širino okna. V tem primeru je oblikovanje veliko bolj zahtevno, če seveda želimo izdelek, ki bo ne le uporaben ampak tudi prijeten za oko. Sama nisem preveč navdušena nad fluidnimi postavitvami, če pa se za njih vendarle odločimo, je po mojem mnenju vseeno koristno določiti največjo širino strani, že zaradi same berljivosti besedila, ki je najboljša pri dolžini 39 znakov na vrstico [16].

Pogledali si bomo statistiko trenutne razširjenosti ločljivosti in poskušali ugotoviti kakšna bi bila najprimernejša širina strani naše spletne predstavitve. Zavedati se moramo, da globalne statistike, še ne nujno odražajo stanja za prostor v katerem delujejo naši naročniki, zato je smiselno opraviti dodatne raziskavo o razširjenosti zaslonskih ločljivosti, v našem primeru za državo Slovenija.

Zaslonske ločljivosti (globalna raven, avgust 2008)

1024x768	44,54%
1280x1024	16,25%
1280x800	13,08%
800x600	6,50%
1440x900	4,98%
1680x1050	3,57%
1152x864	3,46%
1280x768	1,36%
1920x1200	1,17%
1280x960	1,04%

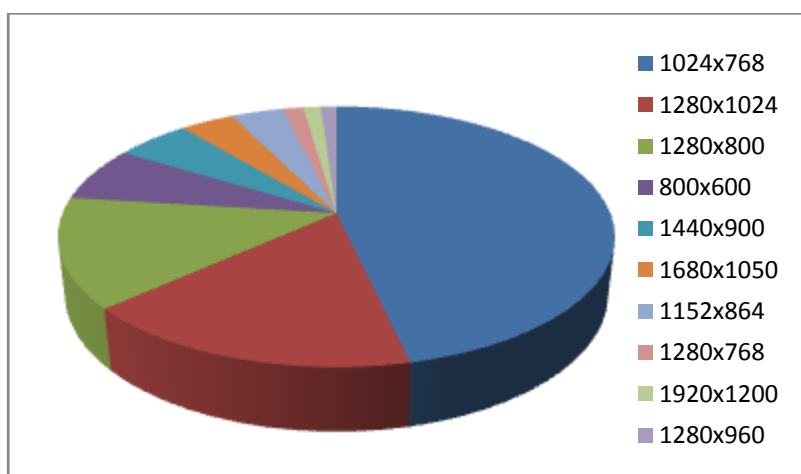


Tabela 2.4: Zaslonske ločljivosti (globalna raven, avgust 2008) [17]

Slika 2.5: Zaslonske ločljivosti (globalna raven, avgust 2008)

Zaslonske ločljivosti (Slovenija, avgust 2008)

1024x768	42,00%
1280x1024	27,13%
1280x800	8,30%
1680x1050	5,06%
1440x900	4,40%
1152x864	3,41%
800x600	2,47%
1280x960	1,31%
1600x1200	1,08%
1400x1050	0,98%

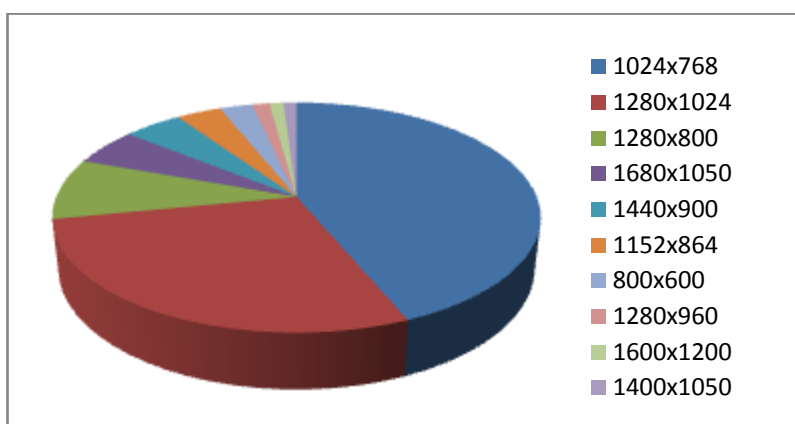


Tabela 2.5: Zaslonske ločljivosti (Slovenija, avgust 2008)

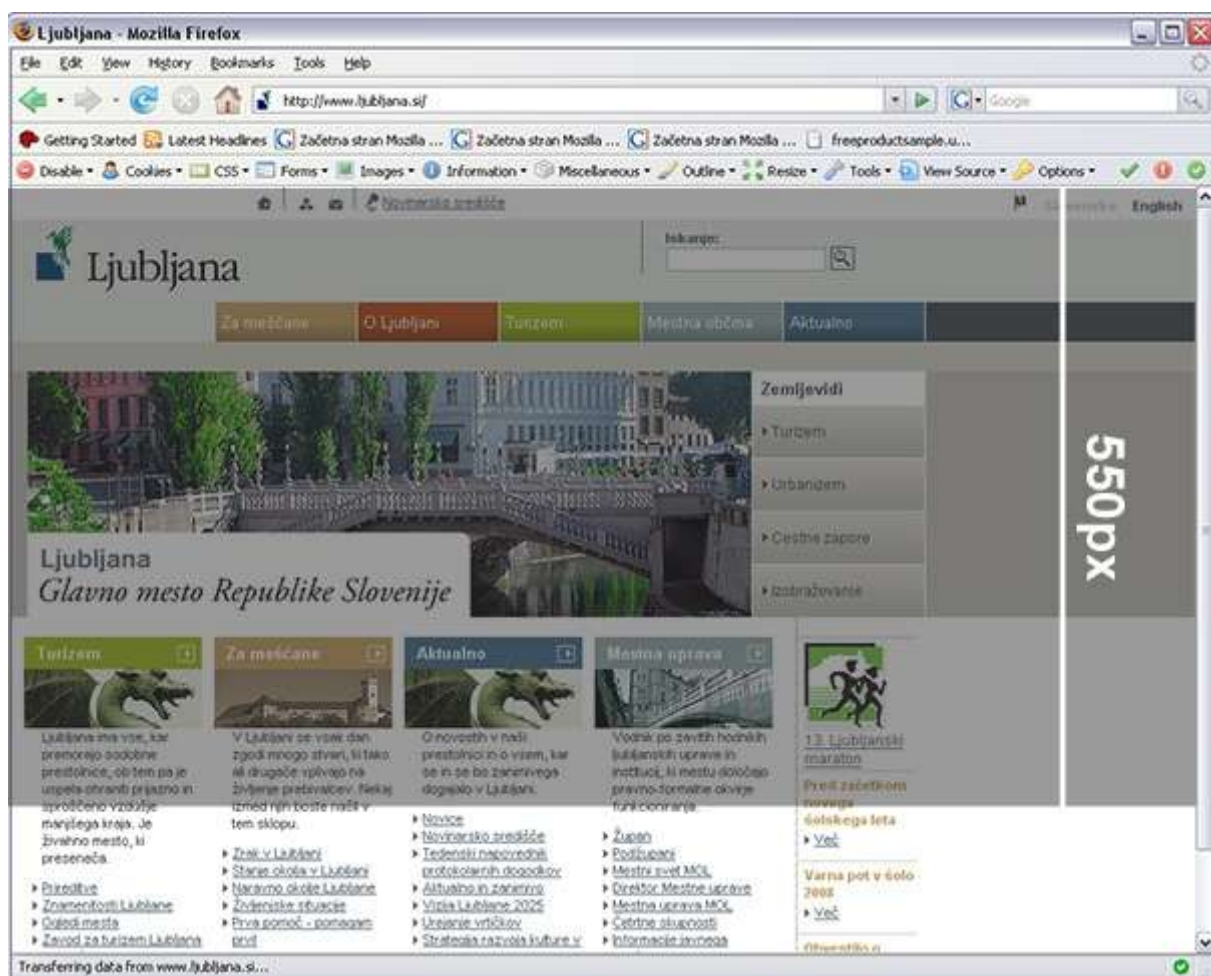
Slika 2.6: Zaslonske ločljivosti (Slovenija, avgust 2008)

Iz zgornjih podatkov lahko vidimo, da je prevladujoča ločljivost v obeh primerih 1024x768, sledi pa ji 1280x1024. Ker je odstotek uporabe zaslonov, ki so sposobni prikazati širino večjo od 800px v Sloveniji 97% v tujini pa 93%, danes spletne strani prilagajamo za širino 1024px in ne za 800px, kot smo to delali pred leti. Ko rečemo, da spletno stran prilagajamo za širino 1024px to ne pomeni, da naj bo naša spletna stran široka 1024px. Če je postavitev centralna je dobro na levi in desni strani pustiti vsaj malo prostora. Poleg tega je treba odšteti robove okna brskalnika in širino drsnika in tako dobimo širino strani nekje med 900px in 960px, lahko tudi manj. Morda se bo ta številka čez nekaj let spremenila, morda se bo to zgodilo že prej. Kakorkoli že, statistike je treba pogosto preverjati, dobro je, da to storimo vsaj enkrat letno.

Opozoriti moramo še na problem oblikovanja z velikimi zasloni, ki jih imajo ponavadi v lasti oblikovalci. Spletna stran lahko pri njih izgleda odlično, dovolj zračno, ko pa enak izdelek pogledamo na zaslonu z manjšo ločljivostjo, pa so elementi natlačeni, včasih preveliki. V procesu oblikovanja je skoraj obvezno je vsake toliko časa pogledati, kako naš izdelek izgleda na manjših zaslonih, z večjo velikostjo točke, ki jih uporablja večina ljudi. Primerjavo je dobro narediti že v fazi skiciranja, da na koncu ni preveč nepotrebnega popravljanja.

Naslednja pomembna stvar je upoštevanje višine območja, na katerega je dobro vključiti vse bistvene elemente spletne strani, saj je le-to vidno večini ljudi brez uporabe vertikalnega drsnika. Pri zaslonski ločljivosti 1280x1024 je to območje visoko od 550 do 600px, odvisno od števila orodnih vrstic (Yahoo, Google toolbars, itd), ki jih ima uporabnik nameščene v

spletnem brskalniku. Višina od 550 do 600px velja ob predpostavki, da je spletni brskalnik odprt čez cel zaslon.



Slika 2.7: Prikaz višine 550 pixlov

Spletna stran www.ljubljana.si upošteva pravilo 550-ih pixlov, saj so na tem območju vsi glavni elementi kot so navigacija in sklopi spletnega mesta.

2.3. Format

Sedaj ko smo obdelali ločljivost in barvno globino, pa si pogledjmo kakšen je značilen spletni format. Zdi se, da je naš format določen z merami zaslona (4:3 ali widescreen), se pravi v vsak primeru ležeč. To drži, če ne upoštevamo ene izmed glavnih značilnosti in to je »scrollanje« navzdol (scrollanje levo in desno iz meni neznanih razlogov ni zaželeno). V tem primeru vidimo, da dobimo izrazito pokončen format, kjer se naša vsebina spušča v

neskončnost (vsaj v teoriji). Zanimivo je, da naj bi dejansko obstajala omejitev višine spletne strani. V Firefox-u 2 je ta omejitev 18939em (kar znaša dobrih 18km), v Firefox-u 3 pa 2219em. Če zunanjemu elementu zapišemo vrednost, ki presega zgoraj napisane omejitve se element skrči na višino vsebine, ki jo vsebuje [17]. Imamo torej format, ki ima variabilno širino in višino. *Kako torej postaviti elemente, kako oblikovati ozadje?*

Kar se tiče ozadja spletnih strani, imamo pri oblikovanju le-tega več možnosti.

- Določimo ga lahko kot enotno barvno podlago, ki jo s CSS-om zapišemo kot

```
body {background-color: #000;}
```

- Uporabimo lahko sliko in jo ponavljamo v X in Y smeri (Slika 2.8, primer 1)

```
body {background: url(roze.gif) repeat;}
```

- Za ozadje določimo horizontalno ponavljajoč vzorec, ki ga kombiniramo z enotno barvno podlago (Slika 2.8, primer 2)

```
body {background: url(vzorec.gif) repeat-x #000;}
```



Slika 2.8: Ozadja HTML dokumenta

- Uporabimo lahko tudi sliko, ki se ne ponavlja, kar nam nudi več fleksibilnosti pri oblikovanju. Pogoj je, da mora biti dovolj široka, da pokrije tudi največje ločljivosti. Problem tega pristopa je lahko velikost slike in pa vedno večje ločljivosti zaslonov, na katere nismo bili pripravljeni. V zgornjem primeru (Slika 2.8, primer 3) je za ozadje uporabljena slika širine 2400px, v formatu JPG, ki je močno kompresiran. Velikost 52KB je še sprejemljiva. <http://www.friskdesign.com/>.
- Ozadje spletne strani lahko tudi fiksiramo, tako da s premikanjem drsnika ozadje ostaja na mestu. V tem primeru mora biti ozadje dovolj široko in visoko, da pokrije vse možne ločljivosti zaslonov. Tovrstne uporabe ozadja na spletu ne srečamo pogosto, vendar lahko v nekaterih primerih izpade zelo dobro. Ozadje fiksiramo s spodnjo vrednostjo CSS lastnosti background:

```
body {background:#d9cab7 url(image.jpg) top center repeat-x fixed;}
```

Primer uporabe fiksnega ozadja: <http://futureofwebdesign.com/2008/newyork/>.

Postavitev spletne strani je lahko leva (privzeto), centralna `#ovoj{margin:0 auto;}` ali desna `#ovoj{float:right}`. Če je vsebine manj kot za višino zaslona, jo lahko centriramo tudi vertikalno. Pri oblikovanju s tabelami je bila ta naloga precej preprosta. Vsebino smo postavili v tabelo, katere višina je bila nastavljena na 100%, celici pa določili centralno vertikalno poravnavo: `vertical-align:middle` (kar je privzeta nastavitve celice). Zaradi novih spletnih standardov ta rešitev ni več mogoča, saj višina tabele ni več dovoljena lastnost. Do enakega efekta lahko sicer še vedno pridemo če HTML dokumentu odstranimo DOCTYPE, in tako dosežemo, da se stran renderira po starejših pravilih (quirks mode), vendar to ni priporočljivo. Vsebino bi lahko centrirali tudi z uporabo JavaScripta vendar obstaja elegantnejša rešitev s CSS-om brez uporabe tabel.

```
<STYLE type="text/css">
html,body {height:100%; margin:0}
```

```
#ovoj {height: 100%; display: table;}  
#pozicija {display: table-cell; vertical-align: middle;}  
</STYLE>
```

Z zgornjo kodo smo rekli naj se naš vsebinski element obnaša kot tabela in se centrira vertikalno. Težava je zgolj v tem, da rešitev deluje v vseh Gecko brskalnikih kot so Firefox, Safari in Netscape vendar ne v Internet Explorerju.

S spodnjo kodo dosežemo delovanje v vseh brskalnikih:

```
<STYLE type="text/css">  
#ovoj {position: relative; overflow: hidden;}  
#pozicija {position: absolute; top: 50%;}  
#vsebina {position: relative; top: -50%;}  
</STYLE>
```

2.4. Cross-browser support

Termin cross-browser support izhaja iz poznih 90ih, časa vojn med brskalnikoma Netscape in Internet Explorer. Takrat je to pomenilo, da je spletna stran podprta za prikazovanje v obeh brskalnikih [18]. Namreč prihajalo je do ogromnih razlik med delovanjem enega in drugega brskalnika, saj so razvijalci dodajali funkcionalnosti brez medsebojnega sodelovanja in nedoslednega upoštevanja priporočil W3C.

Cross-browser support ima danes podoben pomen, le da spletne strani ne prilagajamo zgolj za dva brskalnika. Idealno je, da stran deluje v vseh brskalnikih, ki so trenutno v uporabi. Načini kako različni brskalniki interpretirajo izvorno kodo, se med seboj še danes zelo razlikujejo. Poglejmo kako je z razširjenostjo spletnih brskalnikov po svetu kot tudi pri nas.

Podatki o razširjenosti brskalnikov (globalna raven, avgust 2008)

IE 6	34,62%
Firefox	29,68%
IE 7	26,03%
Safari	1,90%
Opera	1,08%
Mozilla	0,58%

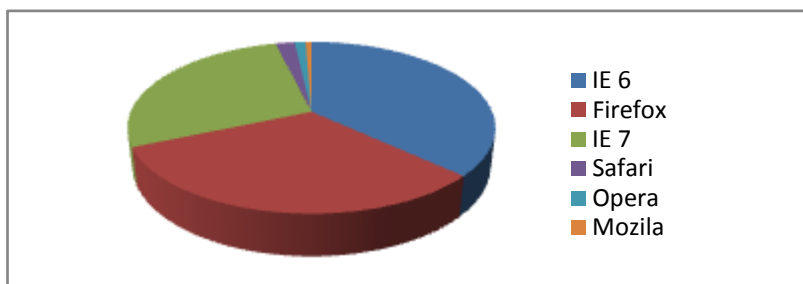


Tabela 2.6: Podatki o razširjenosti brskalnikov (globalna raven, avgust 2008) [20]

Slika 2.9: Podatki o razširjenosti brskalnikov (globalna raven, avgust 2008)

Podatki o razširjenosti brskalnikov (Slovenija, avgust 2008)

IE 6	38,86%
Firefox	32,87%
IE 7	25,70%
Safari	1,25%
Opera	0,87%
Mozilla	0,15%

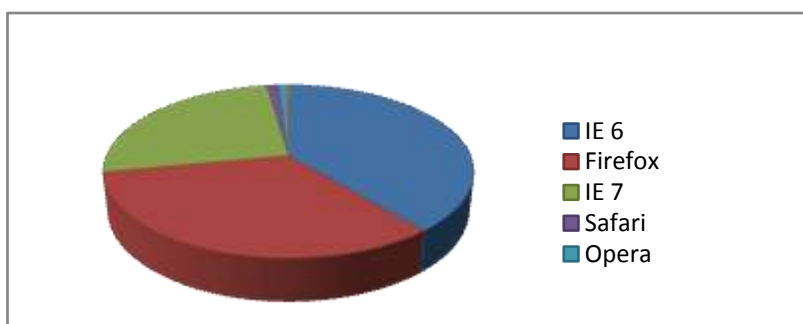


Tabela 2.7: Podatki o razširjenosti brskalnikov (Slovenija, avgust 2008)

Slika 2.10: Podatki o razširjenosti brskalnikov (Slovenija, avgust 2008)

Število spletnih brskalnikov, za katere moramo prilagoditi našo predstavitev je danes večje kot kadar koli prej. Pogled na zgornjo tabelo razkriva zgolj šest najbolj razširjenih, vendar to še zdaleč ni celoten seznam. Upoštevati je treba tudi vse različne verzije brskalnikov, ki so hkrati v uporabi (Firefox 1.5, Firefox 2, Firefox 3...).

Kopica spletnih brskalnikov razvijalcem nemalokrat povzroča preglavice. Njihova naloga ne bi bila tako težka, če bi proizvajalci upoštevali standarde, vendar jih še vedno ne popolnoma. Najbolj znan po neupoštevanju standardov je zagotovo Internet Explorer. Izredno problematična in osovražena verzija tega brskalnika, je verzija 6, ki je, kot smo videli, še zelo razširjena med uporabniki, kljub temu, da je verzija 7 prisotna že dolgo časa. Zavedati se moramo, da splošne javnosti ne moremo enačiti s spletnimi navdušenci, ki si takoj priskrbijo novo verzijo brskalnika, v trenutku ko se le-ta pojavi na trgu. K ogromnemu številu uporabnikov stare različice zagotovo prispeva tudi dejstvo, da Microsoft nove verzije

brskalnika ni ponudil kot samodejno nadgradnjo, ampak so ga v začetku na svoje sisteme lahko posneli le tisti z originalnim operacijskim sistemov. Internet Explorer 7 v marsikaterih pogledih odpravlja napake svojega predhodnika, vendar ima kot njegov predhodnik tudi sam kar nekaj hroščev. Znani hrošči predhodnikov IE7 so: [Three pixel text jog](#), [Peekaboo Bug](#), [Line-height bug](#), [Duplicate Characters Bug](#) in mnogi drugi [21].

V nadaljevanju se bomo še posebej posvetili zelo pogostemu, v nekaterih primerih kar usodnemu hrošču, ki sliši na ime *IE double margin bug*. Na omenjeno težavo sem sama naletela že večkrat in prav vsakič sem jo uspela rešiti s spodaj opisano metodo, ki jo je razvil Steve Clason. Eden izmed hujših napak Internet Explorerja (verzije manjše od 7) je zagotovo njegovo podvajanje lastnosti `margin-left` in `margin-right` v primeru, da je elementu dodeljena lastnost `float`. Kaj to pomeni v praksi? V najboljšem primeru dobimo dvakrat večji razmik med elementi kot smo določili v CSS-u, v najhujšem primeru pa lahko stran popolnoma razpade. V situaciji, ko lastnost `float-left` uporabljamo za razdelitev strani v dva ali več stolpcev okoli njiju pa je element s fiksno širino, bo desnemu elementu zmanjkalo prostora in bo zato porinjen navzdol pod levi element, namesto da bi se pozicioniral poleg njega.

Rešitev problema je povsem preprosta. Vse kar moramo storiti je, da elementu, ki ima podvojen odmik, dodamo naslednjo CSS kodo: `display: inline`. Rešitev je nekoliko nerazumljiva, saj so elementi, ki imajo določeno lastnost `float block` elementi. Tudi če jim priredimo vrednost `inline`, se bodo ti še vedno obnašali enako kot prej, le da se bodo zaradi neznanih razlogov izničili dvojni odmiki v brskalnikih Internet Explorer verzije manjše od 7.

2.4.1. Orodja za testiranje

Pri testiranju in razvijanju spletnega mesta za različne brskalnike so nam v veliko pomoč naslednja orodja:

Firebug je vtičnik za Firefox, ki spletnim razvijalcem mnogokrat pride še kako prav.

Z njim lahko hitro ugotovimo kakšna je struktura strani, kakšne CSS lastnosti imajo določeni elementi. Poleg tega omogoča spreminjanje izvorne kode in CSS dokumenta, zaradi česar lahko hitreje odkrijemo morebitne napake.

Podobno orodje, ki je bilo napisano za Internet Explorer, sliši na ime **IE developer toolbar**, ki je malce bolj okorno in manj uporabno kot Firebug.

Za preverjanje izgleda v različnih brskalnikih, lahko za začetek na svoj sistem posnamemo Firefox, Opera, Safari, itd.. V težavah se znajdemo v trenutku, ko bi spletno stran radi testirali v IE7 in IE6, saj novejša različica odstrani staro. Obstaja več načinov kako zaobiti problem. **Multiple IE's** je aplikacija, ki nam na sistem posname več različic Internet Explorerja in sicer 3, 4, 5, 5.5 in 6. Pri testiranju z omenjenim brezplačnim produktom sem opazila nedoslednost prikazovanja Flash objektov in nekaj drugih napak pri »renderiranju«, ki nam lahko zagrenijo življenje saj se lahko znajdemo v situaciji reševanja problema, ki se pri normalno posnetem brskalniku sploh ne pojavi.

Druga, nekoliko bolj zanesljiva aplikacija se imenuje **IEtester**, ki jo naložimo na sistem in testiramo v verzijah IE5.5 do IE8beta.

Uporabniki Apple računalnikov lahko na strani <http://www.iecapture.com/> naročijo zajem slike svoje spletne strani z IE7. Podobno kot v operacijskem sistemu Windows, lahko tudi na OS X posnamemo le eno verzijo Safarija. [Multi-Safari](#) je aplikacija, ki rešuje to omejitev.

Zelo uporabna in še vedno brezplačna stran je **browsershots**, ki podobno kot IEcapture zajame sliko naše spletne strani v vseh mogočih brskalnikih, (tudi tistih manj razširjenih kot so Dillo, Epiphany, Flock, Galeon, Konqueror in Seamonkey), poleg tega pa je pri predogledu mogoče določiti še različne zaslonske ločljivosti, barvne globine in izbrati ali sta Flash in JavaScript omogočena ali ne. Ponavadi je na slike treba počakati nekaj minut do pol ure, saj so strežniki zaradi velikega števila zahtev pogosto zasedeni, za mesečno plačilo 10€ pa si lahko zagotovimo prednostno procesiranje.

Zadnja storitev, ki jo bomo predstavili se imenuje **Browsercam** in je plačljiva. Za 20\$ do 1000\$ mesečno lahko našo spletno stran testiramo na več kot 70 brskalnikih, različnih ločljivostih, barvnih globinah, itd. Storitev se od prejšnjih razlikuje po tem, da nam na vrne zgolj zajetih slik spletne strani ampak se lahko z oddaljeno povezavo povežemo na njihov sistem, ki nam med testiranjem celo predlaga možne rešitve.

Vprašanje, ki se na tej točki zastavlja samo od sebe, se glasi: *Ali je spletne strani smiselno načrtovati, oblikovati in razvijati tako, da bodo le-te v vseh brskalnikih izgledale identično? Želimo ostajati na zmogljivostih starejših brskalnikov (kot je IE6) dokler bodo le-ti v uporabi ali pa se premaknemo naprej in uporabnikom bolj zmogljivih brskalnikov ponudimo vizualno bogatejše izdelke?*

Pri tem vprašanju obstajata dve skupini ljudi. Prva zagovarja popolno prilagajanje spletne strani za vse brskalnike in s tem enako vidnost informacij vsem, druga pa obsoja ljudi, ki uporabljajo zastarelo programsko opremo in jih želi zaradi tega kaznovati. Pred kratkim sem naletela na vizualno zanimivo spletno stran in njen izgled preverila v IE6, kjer se transparentni PNG elementi niso pravilno prikazali, nekateri drugi elementi spletne strani pa so razpadli. Takoj pod glavo dokumenta sem uzrla napis: Izgleda da uporabljate IE6, na žalost naša spletna stran ni podprta za ta brskalnik. Poizkusite z brskalnikom Safari ali Firefox.

Kar je v te situaciji še posebej čudno je dejstvo, da sem omenjen napis našla na spletni strani znane konference o prihodnosti spletnega oblikovanja: [Future of web design](#).

Možna sta oba zgoraj opisana pristopa, čeprav sama ne zagovarjam nobenega. Prvi je sicer tisti, ki ga ponavadi prakticiramo in spletno stran prilagodimo za vse glavne brskalnike. Drugi pristop, kjer je uporabniku z IE6 onemogočen ogled spletne strani, pa je za moj okus preveč radikalen. Prava rešitev je po mojem mnenju nekje vmes: Spletne strani bi morale načrtovati in izvesti za novejše brskalnike in jo prilagoditi za starejše tako, da bo vsebina še vedno berljiva. S pogojnimi komentarji lahko namreč brez večjega problema odstranimo ali zamenjamo elemente, ki se v starejših brskalnikih povzročajo težave. Za razvoj spletnega oblikovanja je vsekakor slabo, da starejši brskalniki predstavljajo standard in tako postavljajo omejitve, ki jih pri novejših brskalnikih ni več.

3. ZASNOVA SPLETNE STRANI

Za postavitev zasnove spletne strani imamo na voljo dve povsem različni metodi. Uberemo lahko nekemu na prvi pogled lažjo pot in jo postavimo s tabelami, kot je bila praksa včasih ali pa se poslužimo CSS-a in s tem ustvarimo, tako zvani, »Tableless CSS based layout« in se tako tabelam popolnoma izognemo. Trend se je že pred leti obrnil v smer oblikovanja s CSS-om, saj ima ta način ogromno prednosti. Poglejmo si zakaj je oblikovanje s pomočjo tabel že kar lep čas del spletne zgodovine.

Slabosti oblikovanja s tabelami.

1. Oblikovanje s tabelami se lahko nekemu, ki je na njih navajen, zdi enostavnejše, vendar postane kasneje veliko bolj zapleteno za urejanje.
2. Pri oblikovanju s tabelami nastane veliko več kode (v povprečju do 50% več [22]), kar lahko upočasni nalaganje spletne strani. Samo pogledjmo dolžino naslednjih kod:
`<table><tr><td>vsebina</td></tr></table>` ali pa `<div>vsebina</div>`
v primeru oblikovanja s CSS-om.
3. Tabele omejujejo kreativnost, saj smo prisiljeni ostati v okvirih mreže. Nasprotno lahko elemente `<div>`, ki so najpogostejši gradniki strani v primeru oblikovanja s CSS-om, postavimo kamorkoli. Določamo jim lahko negativne odmike (`margin-top:-20px;`), pozicijo (`position:absolute;`) in (`position:relative;`), globino (`z-index:2;`) in jih tako postavimo enega čez drugega, česar s celicami v tabeli ne moremo.
4. Zasnova, ki bazira na tabelah se lahko poruši v trenutku, ko pomotoma odstranimo eno samo celico.

Prednosti oblikovanja s CSS-om.

1. Vsebina je ločena od oblike. Za spletno stran lahko naredimo več CSS dokumentov in tako brez težav preklapljammo med izgledi. Preoblikovanje spletne strani narejene s tabelami bi bilo mnogo zahtevnejše, saj bi se morali spustiti v zamudno popravljanje HTML kode.
2. Izgled in postavitev vseh elementov nekega tipa je definiran na enem mestu, zato je naloga kot je menjava barvne sheme opravljena v trenutku.

3. Oblikovanje s CSS-om omogoča večjo dostopnost. Z različnimi CSS datotekami lahko določimo izgled spletne strani za različne naprave.
4. Naša spletna stran bo ob uporabi pravih HTML oznak kljub onemogočenemu CSS-u še vedno uporabna za ljudi, ki uporabljajo zaslonke bralnike, mobilnike in dlančnike. Stran postavljena z uporabo tabel je, če ji odstranimo oblikovanje, za tovrstne naprave zelo težko berljiva, če ne celo neuporabna.

Odgovor na vprašanje, zakaj spletnih strani že od nekdaj ne oblikujemo s pomočjo CSS-a, je dokaj preprost. CSS včasih v spletnih brskalnikih ni bil tako dobro podprt kot je sedaj, zato so razvijalci iskali svoje rešitve. Tipograf David Siegl je leta 1997 ugotovil, da tabelam lahko odstrani rob in tako je nastalo oblikovanje s tabelami, s katerimi je lažje dosegel podoben izgled v večini brskalnikov, poleg tega pa so mu omogočile postavitev elementov, na podoben način kot je bilo to možno v tiskanih medijih [14]. Oblikovalci in razvijalci so rešitev nemudoma posvojili, kar je prineslo grozovite posledice. Siegl je kmalu ugotovil, da je storil ogromno napako, kar je lepo razvidno iz njegovih besed: »Splet je uničen in jaz sem ga uničil«, »Storil sem ogromno napako mešanja vsebine in oblike«. In res je, element `<table>`, ki ga je izumil Dave Ragget ni bil nikoli mišljen za zasnovo spletne strani, temveč zgolj za tabelarične podatke [14]. Največja težava oblikovanja s tabelami je v tem, da vsiljujejo način razvrščanja vsebine z namenom doseganja vizualnih rezultatov. Če takšni strani odstranimo obliko postane ta neuporabna.

Dandanes spletni brskalniki dobro podpirajo CSS, zato nimamo več razloga da bi za zasnovo spletne strani izbrali tabele. Če želimo upoštevati spletne standarde uporabljamo elemente na način kot je bilo zamišljeno: `<p>` za odstavke, `<table>` za tabelarične podatke, `` in `` za naštevanje, itd...

Zanimivost: Dustin Diaz je prišel na posrečeno idejo, kako promovirati spletne standarde. *CSS naked day* je projekt s katerim si Diaz prizadeva, da bi čim več spletnih strani (v letu 2008 jih je bilo natanko 2019) za en dan v letu »sleklo« CSS in pokazalo svoj `<body>`, kar seveda razkrije princip postavitve spletne strani.

4. POMEN TIPOGRAFIJE NA SPLETU

Tipografija je brez dvoma eden izmed pomembnejših elementov oblikovanja, kot tudi spletnega oblikovanja. Tipografija ni, kot mislijo mnogi, zgolj pisava, ampak mnogo več – je določanje velikosti, razmika med vrsticami, besedami in znaki. Na spletu imamo na nekaterih področjih večjo, drugje manjšo kontrolo. S pomočjo CSS-a lahko zelo natančno nastavljamo razmike med vrsticami (`line-height`), besedami (`word-spacing`) in črkami (`letter-spacing`), velikost pisave (`font-size`) in barvo (`color`), vendar ne poznamo preprostega načina, kako prikazati poljubno pisavo. Izbiramo lahko namreč le med zelo omejenim naborom »web safe« pisav.

Situacija dandanes je takšna, da mora imeti uporabnik pisavo, ki jo je določil oblikovalec / razvijalec posneto na računalniku, da jo brskalnik lahko prikaže. Kot oblikovalci ne moremo vedeti katero tipografijo ima v lasti obiskovalec in katere ne, zato je najbolje, da se poslužimo tistih, ki so del večine operacijskih sistemov.

Microsoft Windows 98	Microsoft Windows XP	Microsoft Windows Vista	OS X (10 do 10.4)
Abadi MT Condensed Light	Arial	Aharoni Bold	Al Bayan
Arial	Arial Black	Andalus	American Typewriter
Arial Black	Comic Sans MS	Angsana New/AngsanaUPC	Andale Mono
Book Antiqua	Courier	Arabic Typesetting	Apple Casual
Calisto MT	Courier New	Arial	Apple Chancery
Century Gothic	Estrangelo Edessa	Arial Black	Apple Garamond
Comic Sans MS	Franklin Gothic Medium	Batang/BatangChe	Apple Gothic
Copperplate Gothic Bold	Gautami	Browallia New/BrowalliaUPC	Apple LiGothic
Copperplate Gothic Light	Georgia	Calibri	Apple LiSung

Courier New	Impact	Cambria	Apple Myungjo
Georgia	Latha	Candara	Apple Symbols
Impact	Lucida Console	Consolas	.AquaKana
Lucida Console	Lucida Sans Unicode	Constantias	Arial
Lucida Handwriting	Microsoft Sans Serif	Corbel	Arial Hebrew
Lucida Sans	Modern MS Sans Serif	Cordia New/CordiaUPC	Ayuthaya
Lucida Sans Unicode	MS Serif	Courier New	Baghdad
Marlett	Mv Boli	DaunPenh	Baskerville
Matisse ITC	Palatino Linotype	David	Beijing
News Gothic MT	Script	DFKai-SB	BiauKai
OCR A Extended	Small Fonts	DilleniaUPC	Big Caslon
Symbol	Symbol	DokChampa	Brush Script
Tahoma	Tahoma	Dotum/DotumChe	Chalkboard
Tempus Sans ITC	Times New Roman	Estrangelo Edessa	Charcoal
Times New Roman	Trebuchet MS	EucrosiUPC	Charcoal CY
Trebuchet MS ¹	Tunga	Euphemia	Chicago
Verdana	Verdana	Fangsong	Cochin
Webdings	Webdings	Franklin Gothic Medium	Comic Sans
Westminster	WingDings	FrankRuehl	Cooper
Wingdings	WST_Czech	FreesiaUPC	Copperplate

	WST_Engl	Gautami	Corsiva Hebrew
	WST_Fren	Georgia	Courier
	WST_Germ	Gisha	Courier New
	WST_Ital	Gulim/GulimChe	DecoType Naskh
	WST_Span	Gungsuh/GungsuhChe	Devanagari
	WST_Swed	Impact	Didot
		IrisUPC	Eupheima UCAS
		Iskoola Pota	Fang Song
		JasmineUPC	Futura
		KaiTi	Gadget
		Kalinga	Geeza Pro
		Kartika	Geezah
		KodchiangUPC	Geneva
		Latha	Geneva CY
		Leelawadee	Georgia
		Levenim	Gill Sans
		LilyUPC	Gujarati
		Lucida Console	Gung Seouche
		Lucida Sans Console	Gurmukhi
		Malgun Gothic	Hangangche

		Mangal	HeadlineA
		Marlett	Hei
		Meiryo	Helvetica
		Microsoft Himalaya	Helvetica CY
		Microsoft JhengHei	Helvetica Neue
		Microsoft Sans Serif	Herculanum
		Microsoft Uighur	Hiragino Kaku Gothic Pro
		Microsoft YaHei	Hiragino Kaku Gothic Std
		Microsoft Yi Baiti	Hiragino Maru Gothic Pro
		MingLiU_HKSCS/MingLiU_HKSCS-ExtB	Hiragino Mincho Pro
		MingLiU-ExtB/PMingLiU-ExtB	Hoefler Text
		Miriam	Inai Mathi
		Mongolian Baiti	Impact
		MS Gothic/MS PGothic/MS UI Gothic	Jung Gothic
		MS Mincho/MS PMincho	Kai
		MV Boli	.Keyboard
		Narkisim	Krungthep
		Nyala	KufiStandard GK
		Palatino Linotype	.LastResort

		Plantagenet Cherokee	LiHei Pro
		Raavi	LiSong Pro
		Rod	Lucida Grande
		Segoe Print	Marker Felt
		Segoe Script	Monaco
		Segoe UI	Monaco CY
		Shruti	Mshtakan
		SimHei	Nadeem
		Simplified Arabic	New Peninim
		Simsun/NSimsun	New York
		SimSun-ExtB	NISC GB18030
		Sylfaen	Optima
		Symbol	Osaka
		Tahoma	Palatino
		Times New Roman	Papyrus
		Traditional Arabic	PC Myungjo
		Trebuchet MS	Pilgiche
		Tunga	Plantagenet Cherokee
		Verdana	Raanana
		Vrinda	Sand

		Webdings	Sathu
		Wingdings	Seoul
			Shin Myungjo Neue
			Silom
			Skia
			Song
			ST FangSong
			ST Heiti
			ST Kaiti
			ST Song
			Symbol
			Tae Graphic
			Tahoma
			Taipei
			Techno
			Textile
			Thonburi
			Times
			Times CY
			Times New Roman

			Trebuchet
			Verdana
			Zapf Chancery
			Zapf Dingbats
			Zapfino

Tabela 4.1: Pisave različnih operacijskih sistemov [23][24]

Če primerjamo skupine pisav in pogledamo katere so vsem operacijskih sistemom skupne, dobimo naslednji presek:

Serifen **Tymes New Roman**, nevtralen **Arial**, poudarjen **Arial Black**, spletu prijazna široka brezserifna **Verdana**, stilska serifna **Georgia**, ožji brezserifen **Trebuchet MS**, večinoma neuporaben **Impact**, in **Courier New**.

Zgornji seznam na srečo še ne pomeni končne omejitve uporabe pisav na spletu. S CSS lastnostjo `font-family` lahko namreč določimo zalogo pisav. Prva naštetja je tista, ki jo želimo prikazati, vsaka naslednja pa se uporabi, če njena predhodnica ni naložena na sistemu. Kot zadnjo naštejemo družino kateri določena skupina pisav pripada.

```
h1 {
font-family: 'Lucida Grande', 'Lucida Sans Unicode', Lucida, Arial,
Helvetica, sans-serif;
}
```

Da bi torej naši besedilni elementi na različnih računalnikih izgledali čim bolj podobno, moramo `font-family` premišljeno definirati, da pa to lahko storimo, je obvezno dobro poznavanje podobnosti med skupinami sistemskih pisav različnih operacijskih sistemov.

4.1. Primer uporabe zaloge pisav

Na spodnji spletni strani lahko vidimo primer slabe definicije zaloge pisave in skoraj neberljivo besedilo kot rezultat tega. Najhuje je, da spletna stran pripada znani skupini [Hoefler & Frere-Jones](#), ki se ukvarja s prodajo tipografij.



Slika 4.1: Primer slabo izbrane zaloge pisav

Z rdečo barvo so na sliki (Slika 4.1) označeni problematični elementi. Pa si poglejmo kaj je vzrok njihove težave. Iz njihovega CSS dokumenta lahko razberemo, da se za element `<body>` uporablja naslednja zaloga pisav:

```
body {  
  font-family: 'Lucida Grande', 'Lucida Sans', Verdana, Georgia, Helvetica,  
  Arial;  
}
```

Najverjetneje je, da avtor spletne strani uporabljajo Apple-ov operacijski sistem, saj je prva določena pisava Lucida Grande del sistema OS X. Verjamem, da stran pri njih izgleda odlično. Težava se pojavi pri uporabnikih PC-jev, pri katerih se uporabi druga definirana pisava in sicer Lucida Sans, ki ni najbolj primerna za uporabo na spletu (brez uporabe opcije ClearType, ki je mnogi nimajo vklopljene, je uporabna šele od velikosti 18px naprej). Spodnja slika (Slika 4.2) prikazuje kako besedilo izgleda z vklopljeno opcijo ClearType, ki odpravi večino težav, vendar besedilo v desnem stolpcu kljub temu ne izgleda dobro.



Slika 4.2: Primer slabo izbrane zaloge pisav

Z uporabo orodja Firebug lahko poizkusimo namesto pisave Lucida Sans na drugo mesto postaviti pisavo Lucida Sans Unicode, ki je del operacijskega sistema Microsoft Windows 98 in XP. Vidimo lahko, da je kljub izklopljenemu ClearTypeu besedilo sedaj berljivo.



Slika 4.3: Primer slabo izbrane zaloge pisav

4.2. Predstavitev skupin pisav

V nadaljevanju bomo predstavili skupine pisav, ki so zaradi razširjenosti primerne (nekatero bolj druge manj) za spletno predstavitev.

4.2.1. Sans-serifne pisave – pisave brez serifov:

Arial, Verdana in Trebuchet MS so najbolj pogosto uporabljene pisave na spletu. Verdana je zaradi širokih znakov dobro berljiva tudi pri majhnih velikostih in zato primerna za uporabo v menijih kot tudi besedilu. Arial je zelo podoben Helvetici, pisavi za uporabo katere je treba plačati.

The image shows three instances of the word "Ag" in a large, bold, black sans-serif font. The first instance is Arial, the second is Verdana, and the third is Trebuchet MS. The letters are clean and modern, with no serifs.

Slika 4.4: Arial, Verdana in Trebuchet MS

Spodnje pisave so svojo zaobljenostjo in širino primerne za razne poudarke in naslove.

The image shows three instances of the word "Ag" in a large, bold, black rounded sans-serif font. The first instance is Lucida Grande, the second is Century Gothic, and the third is Tahoma. The letters have a soft, rounded appearance compared to the previous examples.

Slika 4.5: Lucida Grande, Century Gothic in Tahoma

Calibri, *Candara* in *Corbel* so nove sistemske pisave operacijskega sistema Windows Vista. Narejene so bile posebej za uporabo na spletu, zamenjale pa naj bi Arial, Verdano in Trebuchet MS.

The image shows three instances of the word "Ag" in a large, bold, black modern sans-serif font. The first instance is Calibri, the second is Candara, and the third is Corbel. The letters are clean and modern, with a slightly more rounded and friendly feel than the previous examples.

Slika 4.6: Calibri, Candara in Corbel

Arial Black, Comic Sans in Impact so popolnoma neprimerne za daljša besedila, pogojno bi se jih dalo uporabiti za naslove. Zaradi prepogoste uporabe/zlorabe pisave Comic Sans je le ta med oblikovalci ponavadi nezaželena. Impact je pri majhnih velikostih izjemno nazobčan, zato njegovo uporabo priporočam od velikosti 17px naprej.

The image shows three instances of the word "Ag" in large, bold, black font. The first is in Arial Black, the second is in Comic Sans MS, and the third is in Impact. The Comic Sans version has a noticeably more rounded and informal appearance compared to the other two.

Slika 4.7: Arial Black, Comic Sans MS in Impact

4.2.2. Serifne pisave

Serifni *Times New Roman* in njegov ekvivalent *Times* na Linuxu in Mac-u so primerni za izražanje občutka tradicionalnosti – spletna stran, ki posnema izgled časopisa (naslovi na www.vest.si). Uporaba *Georgie* je pogosta na blogih, še posebej lepo izpade v načinu italics pri večjih velikostih. Sama *Georgie* zelo rada uporabljam za prikazovanje števil, ki so v tej pisavi res nekaj posebnega.

The image shows three instances of the word "Ag" in large, black serif font. The first is in Times New Roman, the second is in Times, and the third is in Georgia. All three fonts exhibit classic serif characteristics with distinct serifs on the letters.

Slika 4.8: Times New Roman, Times in Georgia

Alternativa uporabe zgornjih serifnih pisav so *Book Antiqua* (Windows) in *Palatino*.

The image shows two instances of the word "Ag" in large, black serif font. The first is in Book Antiqua and the second is in Palatino. Both fonts are highly stylized serif fonts with prominent serifs.

Slika 4.9: Book Antiqua, Palatino

Cambria in *Constantia*, še dve "C" pisavi del Windows Viste. Predvideni sta bili za zamenjavo pisav Times New Roman in Palatino.

Slika 4.10: Cambria in Constantia

4.2.3. Monospace – zaslonske pisave za pisanje kode

Slika 4.11: Andale Mono, Courier New in Courier

Slika 4.12: Consolas, Lucida Console in Monaco

Naredili smo pregled pisav, ki so zaradi svoje razširjenosti primerne za uporabo na spletu, spodaj pa so predstavljene nekatere zgoraj opisane pisave in njihovi ekvivalenti na sistemu OS X. Spodnja razpredelnica nam je lahko v pomoč pri sestavljanju font-family zalog.

Pisave na sistemu Windows	Ekvivalentne pisave na sistemu OS X	Družina
Arial	Arial, Helvetica	<i>sans-serif</i>
Arial Black	Arial Black, Gadget	<i>sans-serif</i>
Comic Sans MS	Comic Sans MS	<i>cursive</i>
Courier New	Courier New, Courier	<i>monospace</i>
Georgia	Georgia	<i>serif</i>
Impact	Impact, Charcoal	<i>sans-serif</i>
Lucida Console	Monaco	<i>monospace</i>
Lucida Sans Unicode	Lucida Grande	<i>sans-serif</i>
Palatino Linotype	Palatino	<i>serif</i>

Book Antiqua		
Tahoma	Geneva	sans-serif
Times New Roman	Times	<i>serif</i>
Trebuchet MS	Trebuchet MS, Helvetica	sans-serif
Verdana	Verdana, Geneva	sans-serif
MS Sans Serif	Geneva	sans-serif
MS Serif	New York	<i>serif</i>

Tabela 4.2: Zaloge pisav [25]

Kljub zgoraj opisanim zalogam smo pri izbiranju pisave še vedno zelo omejeni, zato ni čudno, da so mnogi začeli premišljevati kako se te omejitve znebiti. Obstaja že kar nekaj načinov, kako zaobiti problem, vendar noben od njih ni optimalen. Na pravo rešitev, ki bo z nami ostala dalj časa po mojem mnenju še čakamo.

4.3. Tipografija na strežniku

Eden izmed preprostejših načinov je ta, da želeno tipografijo posnamemo na strežnik. Obiskovalcu tipografije ni potrebno imeti na svojem računalniku, saj jo brskalnik naloži kar iz strežnika. To storimo z naslednjim CSS pravilom:

```
@font-face {
    font-family: "Swiss 721";
    src: url(swiss721.ttf);
}
```

Kasneje jo uporabimo na način:

```
body {
    font-family: "Swiss 721", Arial, sans-serif;
}
```

[26]

Enostavno in hitro, torej kje je tu problem? Problem je zgolj ta, da zgoraj opisana metoda deluje le brskalniku Internet Explorer in Safari 3.1. Poleg tega se pri tej rešitvi zastavlja tudi vprašanje avtorskih pravic. Tipografija, ki je posneta na strežniku je zlahka dostopna

vsakomur. Rešitev? Uporaba le brezplačnih tipografij ali razvoj metode, ki bi preprečila presnemavanje tipografije s strežnika. Govori se, da bo novi CSS3 podpiral to pravilo.

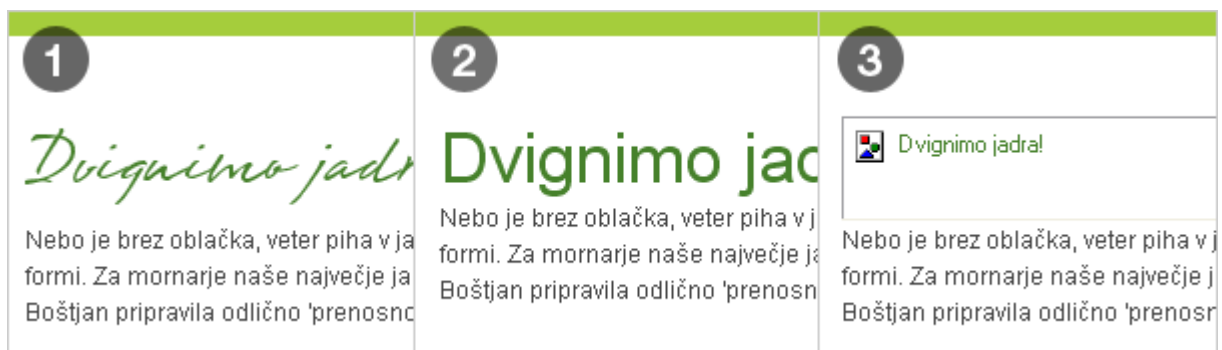
4.4. Zamenjava s slikami

Način, ki se ga poslužujejo mnogi, je zamenjava dela besedila s slikovnim elementom. Slabi strani te metode je vsaj nekaj in sicer:

- Stran se zaradi slik, ki zasedejo mnogo več prostora kot besedilo, nalaga dlje
- Besedila ni možno kopirati
- Proces pretvarjanja je lahko zamuden (ročno ali pa bolj napredno z generatorji slik).
- Če aplikacija ni preiščljena, lahko izgubimo semantično vrednost elementa, kar pomeni nižje rangiranje v spletnih iskalnikih in posledično manj obiska.

Zadnja točka je zelo problematična iz vidika optimizacije za spletne iskalnike (SEO). Besedilo naslova seveda lahko zapišemo kot ALT besedilo, vendar izgubimo pomembnost, ki bi jo sicer nosil element `<h1>`. Iskalnik Google višje rangira besede zapisane v elementih `<h1>`, `<h2>`.. kot tiste v elementu `<p>` ali ``. Rešitev je, da element `` postavimo znotraj elementa `<h1>`.

Kako je z dostopnostjo? Na spletni strani www.desetka.tv sem naredila manjši preizkus situacije, ko uporabnik izklopi prikazovanje slikovnih elementov. Ugotovila sem, da alt besedilo slike v brskalniku Firefox in Opera prevzame vlogo besedila znotraj elementa `<h1>`, in je tako tudi formatirano (Slika 4.13, primer 2). V IE6 in IE7 je situacija nekoliko drugačna. Besedilo se izpiše z majhno pisavo znotraj okvira kjer bi se morala naložiti slika (Slika 4.13, primer 3).



Slika 4.13: Zamenjava besedila s slikami

4.5. SiFR

SiFR (Scalable Inman Flash Replacement) je nekoliko bolj zapletena rešitev problema prikazovanja poljubne tipografije. Uporablja se predvsem za naslove in ostale poudarke kot so citati, itn, nikakor pa ne za celotno besedilo na spletni strani. SiFR so razvili Mike Davidson, Mark Wubben in Shaun Inman, po katerem je SiFR tudi dobil ime [27]. Tehnologija SiFR se nenehno razvija. Trenutno je na voljo že različica 3.0, ki v marsičem prekaša različico 2.0.

Kaj torej je SiFR in kako deluje? SiFR je odprtokodna tehnologija, ki bazira na Flashu, JavaScriptu in CSS-u. Povedano na kratko gre za zamenjavo krajših odsekov besedila z Flash ekvivalenti.

Potek delovanja:

- Z orodjem Adobe Flash naredimo SWF datoteko v katero vključimo poljubno tipografijo
- S CSS-om določimo elemente, ki jim želimo zamenjati tipografijo. Poleg tega na tem mestu določimo velikost, barvo in druge lastnosti pisave.
- Javascript funkcija preveri če ima uporabnik naložen Flash predvajalnik
- Če predvajalnik ni naložen se stran naloži normalno (pisava se prikaže v tisti sistemski pisavi, ki smo jo določili v glavnem CSS dokumentu)
- V nasprotnem primeru funkcija proizvede Flash objekte in jih postavi na mesto, kjer je prej stalo besedilo, originalno besedilo pa se skriva [27].

Prednosti prikazovanja poljubne tipografije s tehnologijo SiFR

- Semantična vrednost elementa ostaja enaka.
- Tipografija je zaščitena pred redistribucijo.
- SWF datoteka, v katero so shranjeni vsi znaki, se naloži le enkrat. Flash elementi se nato generirajo iz nje. Besedilo je še vedno vektorsko.
- Besedilo je možno izbrati, kopirati in prilepiti.
- Deluje v vseh glavnih brskalnikih.
- S CSS-om imamo popoln nadzor nad izgledom.

Slabosti:

- V poljubni pisavi ni priporočljivo prikazovati dolgih odsekov besedila.
- V nekaterih primerih obremeni procesor, kar povzroči, da se naša stran nalaga dlje kot bi se sicer.
- Na računalniku mora biti omogočen JavaScript.
- Na računalniku mora biti posnet Flash predvajalnik, vendar pa zaradi dejstva, da ima predvajalnik po raziskavah Adobe naloženih 98,8% ljudi [28], SiFR deluje v večini primerov.
- Pri kompleksnih spletnih mestih, ki uporabljajo JavaScript knjižnice kot so jQuery in MooTools lahko pride do ne pravilnega delovanja zaradi nekompatibilnosti med knjižnicami.

Odločila sem se, da tudi sama preizkusim opisano metodo prikazovanja poljubne pisave. Na spodnjem naslovu lahko vidite enostavno predstavitev delovanja tehnologije SiFR, ki je sestavljena iz treh HTML dokumentov med katerimi navigiramo z uporabo menija v zgornjem desnem kotu.

<http://www.emivega.com/00/diploma/sifr/>

Vidimo lahko, da se izgled naslovov in izvlečkov na desni strani z navigacijo med stranmi spreminja. Na osnovni strani je uporabljen klasičen Times New Roman, na ostalih dveh HTML dokumentih pa se naslovi zamenjajo s Flash ekvivalenti v pisavah Agendia, Gunny Handwriting in Vandenkeere za izvlečke.

5. PODPORA TRANSPARENCE

Oblikovalci s(m)o v veliki večini navajeni dela v orodjih kot so Adobe Photoshop, Adobe Illustrator, Adobe InDesign, Adobe Flash, in drugih, ki nam omogočajo skoraj neomejeno izražanje naše kreativnosti. Med mnogimi funkcijami so tu plasti (layers), ki jim lahko nastavimo stopnjo vidnosti (opacity). Z združevanjem in kombiniranjem takšnih plasti je mogoče priti do zanimivih vizualnih efektov. *Kako pa je s tem na spletu? Ali spletni brskalniki podpirajo transparentne elemente?* Z vsem tem se bomo ukvarjali v trenutnem poglavju.

5.1. Transparentni (X)HTML elementi

Predstavljajmo si naslednjo nalogo. Spletni strani smo določili ponavljajoče ozadje z vzorcem, nanj pa bi radi postavili stolpec z vsebino, katere ozadje je pol transparentno, tako da je vzorec za njim delno viden. Zdi se, da naloga ne bo pretežka saj ukaz za prosojnost poznajo vsi spletni brskalniki, tako Firefox, Opera kot Internet Explorer (celo verzija 6). Sintaksa je za različne brskalnike nekoliko drugačna vendar stvar deluje.

```
#ovoj{
  opacity: 0.65; /* za brskalnike Firefox, Safari, in Opera */
  -moz-opacity: 0.65; /* za starejše Gecko brskalnike kot je Netscape */
  filter:alpha(opacity=65); /* za IE6 in IE7 */}
```

Z zgornjo CSS kodo dobimo element, ki je 65% viden. To smo storili z zapisom treh lastnosti in sicer: Opacity, moz-opacity in filter:alpha. Tri zapise potrebujemo zato, ker se načini kako spletni brskalniki prikazujejo transparenco med seboj razlikujejo. Dejstvo, da imamo za isto stvar tri ukaze brskalnike ne motijo, saj tiste lastnosti, ki jih ne poznajo preprosto ignorirajo.

Na problem naletimo pri oblikovanju vsebine tega elementa, saj hitro opazimo, da se transparentnost starševskega elementa deduje na vse njegove otroke, kar je logično, saj je dedovanje ena glavnih lastnosti CSS-a. Po principu specifičnosti se pri zapisu nove vrednosti bolj natančno določenega elementa stara lastnost izniči [29]. Tako bi pričakovali bi da bo otrok postal 100% viden, če mu bomo to določili s CSS-om, vendar se to ne zgodi. Besedilo znotraj vsebinskega elementa ostaja slabo berljivo, kot je razvidno iz spodnjega primera.



Slika 5.1: Transparenca HTML elementa

Obstaja druga pot do rešitve. Element z vsebino moramo, čeprav je tehnično znotraj tistega s transparentnim ozadjem, postaviti ven, tako da le ta ne bo več obravnavan kot njegov otrok. To storimo tako, da elementu dodelimo relativno pozicijo glede na prejšnji element, in ga tako postavimo v ospredje: `#vsebina{position: relative;}`



Slika 5.2: Transparenca HTML elementa

Sedaj je vsebina notranjega elementa 100% vidna, besedilo pa je splet berljivo.

5.2. Transparentne slike

Videli smo, da so transparentni HTML elementi kar dobro podprti, kako pa je s tem pri slikah? *Je varno uporabiti sliko formata PNG s transparentnim ozadjem? Jo lahko uporabimo kot ozadje, kaj pa kot ponavljajoče ozadje?*

Kot smo že povedali v prvem poglavju, poznamo dva spletna slikovna formata, ki podpirata transparenco. To sta GIF in PNG. Format GIF je s svojo binarno transparenco podprt v vseh spletnih brskalnikih, vendar bi nam v mnogih situacijah bolj prav prišel format PNG, ki s svojo 8 bitno transparenco omogoča lepše prehode v primerjavi z GIF-ovimi nazobčanimi robovi.

Netransparenten format PNG je podprt v vseh brskalnikih, njegova transparentna različica prav tako, razen v IE6. Ta brskalnik sliko sicer prikaže, vendar ne razume njegovega alpha kanala. Področje, ki naj bi bilo transparentno je v tem brskalniku svetlo sive barve [30].

Dobra novica je, da imamo na voljo kar nekaj različnih načinov kako zaobiti problem. Sama sem testirala tri različne rešitve in vsaka izmed njih ima svoje prednosti in slabosti. Pa si jih pogledjmo.

5.2.1. Filter

Microsoft je za svoj brskalnik Internet Explorer ustvaril filtre, CSS lastnosti, ki omogočajo razne efekte nad slikami. Ti filtri v drugih brskalnikih ne delujejo. Eden izmed njih je AlphaImageLoader, ki naredi prav to kar potrebujemo v našem primeru – pravilno interpretira alpha kanal PNG slike [31]. Poseben CSS, definiran izključno za brskalnik IE6 izgleda tako:

```
.transparent {  
  filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(src='ozadje.png',  
  sizingMethod='crop');}
```

Slika, kateri smo priredili selektor transparent se sedaj prikaže pravilno. Težava je v tem, da moramo zaradi zapisa poti do datoteke zgornjo kodo ponoviti tolikokrat, kolikor PNG slik s prozornim ozadjem imamo na strani. Poleg tega zaradi lastnosti filter naš CSS dokument ne bo uspešno validiran, kar niti ni tako hudo. Ugotovila sem, da se prave težave začnejo šele, ko

želimo transparenten PNG uporabiti za ozadje elementa. Zaradi neznanega razloga postanejo povezave, ki ležijo na sliki mrtve, poleg tega pa se na povezavah onemogoči `:hover`. Težava ni nerešljiva, saj lahko element z vsebino zopet pozicioniramo relativno in ga tako dvignemo v ospredje, kot smo to naredili v prejšnjem primeru.

Ob nadaljnjem testiranju vidimo, da sliki, ki je uporabljena za ozadje ne moremo nastavljati pozicije, težave pa imamo tudi pri njenem ponavljanju (CSS vrednosti `repeat`). Filter pozna tri metode prikaza slike (`SizingMethod`) in sicer: *image*, *crop* in *scale* [32]. Pri metodi *crop* ali *image* se slika kljub CSS nastavitvi `repeat` pokaže le enkrat. Pri metodi *scale* pa se zgodi razširitev slike na širino in višino elementa namesto njenega ponavljanja.



Slika 5.3: Prikaz v IE7 in Firefox-u



Slika 5.4: Prikaz v IE6

To lastnost, oz napako v delovanju lahko v nekaterih primerih celo izkoristimo nam v prid. Recimo, da bi želeli oblikovati element, ki bi na desno in levo stran metal senco. Za ozadje bi uporabiti spodnjo pol transparentno sliko, ki bi jo želeli ponavljali v smeri y.



Če vnaprej premislimo kako deluje filter in predvidimo kakšen efekt bomo dobili z njegovo aplikacijo, lahko vidimo da je izvedba dejansko mogoča (z enakim izgledom v vseh brskalnikih). V večini brskalnikov bo CSS vrednost: *repeat-y* sliko ponovila po višini elementa, v IE6 pa se bo zgodilo naslednje: slika se bo raztegnila po višini in tako napolnila prostor elementa. Širina bo v tem primeru ostala nespremenjena, saj je širina slike enaka širini elementa. Če element nima določene širine se slika v ozadju ne bo prikazala, zato moramo obvezno določiti širino elementa.

V tem primeru dobimo v vseh spletnih brskalnikih enako sliko



Slika 5.5: Transparenca pri ponavljajočem slikovnem formatu PNG

Vidimo, da lahko slike, katerih izgled ostane z raztegovanjem nespremenjen, brez težav ponavljamo v eno smer. Elementi znotraj katerih ponavljamo sliko pa morajo imeti obvezno določeno širino.

5.2.2. Unit PNG fix

Unit PNG fix so razvili pri Unit interactive. Gre za JavaScript kodo, ki na vse slike formata PNG na strani aplicira zgoraj opisani AlphaImageLoader filter. Aplikacija je povsem preprosta saj moramo v glavo dokumenta vključiti le povezavo do JavaScript kode [33].

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

Prednosti te metode pred prejšnjo so naslednje:

- Osnovni CSS uspešno prestane preizkus z validatorjem, saj se filter slikam doda kasneje.
- Povezave delujejo brez ročnega nastavljanja relativne pozicije vsebinskim elementom, saj za to poskrbi že skripta sama.

Slabost

- Če uporabnik nima vklopljenega JavaScripta se slike ne bodo prikazale pravilno

Datoteke in navodila za uporabo so dostopna na naslovu:

<http://labs.unitinteractive.com/unitpngfix.php>

5.2.3. Twin Helix Behavior

Zadnja rešitev deluje na podoben princip kot prejšnja, le da jo apliciramo na nekoliko drugačen način. V CSS dokument vključimo spodnjo kodo, ki kliče datoteko iepngfix.htc.

```
img { behavior: url("iepngfix.htc") }
```

Prednosti pred zgornjima rešitvama

- Deluje v IE 5.5, zgornji metodi samo v IE6
- Za delovanje ni potreben JavaScript
- Povezave delujejo brez relativnega pozicioniranja
- Selektor :hover na povezavah deluje brez relativnega pozicioniranja

Slabosti

- Naš CSS dokument ni validen (lastnost behavior je na seznamu priporočil za CSS3)
- Na območjih kjer je v ozadju slika se nad besedilom ne prikazuje kurzor ampak pointer (rešimo z relativnim pozicioniranjem)

Datoteke in navodila za uporabo so dostopna na naslovu:

<http://www.twinhelix.com/css/iepngfix/>

Zgornje rešitve pridejo prav v primeru, ko želimo konsistenten izgled v vseh spletnih brskalnikih. Če nam enoten izgled ne pomeni veliko, lahko s pomočjo pogojnih komentarjev (ki jih razumejo le IE brskalniki) za IE6 in manj določimo poseben CSS.

```
<!--[if IE 6]>
  <link rel="stylesheet" type="text/css" href="ie6.css" />
<![endif]-->
```

Pri testiranju zgornjih metod sem opazila še en problem. Izgled slik formata PNG s transparentnim ozadjem v različnih brskalnikih ni konsistenten. V Firefoxu so bile slike za malenkost temnejše kot v Internet Explorerju 7. Malo sem pobrskala po spletu in razlog naj bi se skrival v tem, da se v format PNG zapiše gama vrednost, ki jo različni brskalniki interpretirajo drugače. Avtor članka zaključí, da ne pozna načina, kako doseči popolnoma enak izgled slikovnega formata PNG v vseh brskalnikih. Najbolj se približamo pravim barvam če iz PNG zapisa odstranimo gama informacije, vendar za to potrebujemo poseben program po imenu TweakPNG [34]. Poleg zgornjih problemov, je tu le še en razlog proti uporabi, drugače odličnega spletnega formata.

Zaključimo lahko, da format PNG s transparentnim ozadjem prinaša svojo porcijo težav. Razmisliti je treba kako pomembni so tovrstni slikovni elementi za naš projekt in ali jih je v resnici vredno implementirati. Sodelovala sem pri projektih, kjer brez njih enostavno ni šlo, spet druge pa ji je bilo možno zamenjati s formatom GIF.

Slikovni elementi, ki jih ne uporabimo kot ozadje načeloma niso tako zelo problematični, zakomplicira se, ko jih postavimo v ozadje, zlasti če bi jih želeli ponavljati. V praksi smo se

temu do sedaj izogibali. Po temeljitem pregledu možnih rešitev vidimo kakšne so naše možnosti, zato se bomo pri prihodnjih projektih lažje odločili o uporabi ali neuporabi tovrstnih slikovnih elementov.

6. IZBOLJŠANJE INTERAKTIVNOSTI

Splet je medij, ki v svojem bistvu omogoča interakcijo, feedback v trenutku ko uporabnik stori neko akcijo. V realnosti je situacija nekoliko drugačna, saj uporabnika vse prevečkrat pustimo čakati. Vzroki za čakanje so pošiljanje zahteve na strežnik ob vsaki akciji uporabnika in nato čakanje na odgovor. Že klasičen proces izpolnjevanja obrazca (preverjanje ali je uporabniško ime na voljo, itd) zna biti zamuden. Težavo rešuje AJAX (Asynchronous JavaScript and XML), a več o tem v poglavju Web 2.0. Na tem mestu bomo izpustili tehnične podrobnosti AJAX-a in se raje posvetili različnim JavaScript knjižnicam, ki ne le lajšajo delo z AJAX-om, temveč omogočajo fantastične efekte in animacije, za katere smo poprej potrebovali Flash.

6.1. jQuery

Je knjižnica vnaprej pripravljenih JavaScript funkcij, ki jo je v začetku leta 2006 razvil John Resig. Filozofija jQuery-ja se glasi: Write less, do more. jQuery kot tudi druge knjižnice razvijalcem močno olajšajo delo. Sedaj lahko ljudje brez ali osnovnim znanjem JavaScripta proizvedejo fantastične efekte kot so: show, hide, slide, fade, animate, itd. Za boljšo predstavo si pogledjmo spodnje primere, ki so le peščica tega kar obstaja:

Drevesna struktura datotek (Slika 6.1, primer 1)

http://abeautifulsite.net/notebook_files/58/demo/

LavaLamp drseči meni (Slika 6.1, primer 2)

<http://www.gmarwaha.com/blog/?p=7>

Vertikalno drseča vsebina (Slika 6.1, primer 3)

<http://www.arcinspirations.com/kobe/>

LightBox (Slika 6.1, primer 4)

Pogosto uporabljen, že skoraj standarden prikazovalnik slikovnih datotek

<http://leandrovieira.com/projects/jquery/lightbox/>



Slika 6.1: jQuery primeri

6.2. MooTools

Slidedown login obrazec (Slika 6.2, primer 1)

http://www.hotajax.org/demo/mootools/login_form_mootools_1.2/login_form_mootools_1.2/index.html

Generiran odsev (Slika 6.2, primer 2)

<http://www.digitalia.be/software/reflectionjs-for-mootools#demo>

Meni, v 3D prostoru (Slika 6.2, primer 3)

<http://www.hotajax.org/download/mootools/3d-carousel/index.html>

Premikajoči se meni (Slika 6.2, primer 4)

Za zadnji primer sem se odločila sama implementirati enega izmed efektov in sicer

Fx.elements. Rezultat je viden na naslovu: <http://www.emivega.com/00/diploma/meni>

JavaScript koda, ki sem jo uporabila je dostopna na:

<http://demos111.mootools.net/Fx.Elements>, ikone za meni pa sem si sposodila iz naslova:

<http://tea00.deviantart.com/art/heart-PNG-48675249>



Slika 6.2: MooTools primeri

Poleg predstavljenih knjižnic obstajajo še *Scriptaculous*, *YUI* in nekatere druge. Z vsemi zgoraj omenjenimi knjižnicami z lahkoto dosežemo efekte s katerimi naša spletna stran dobi bolj interaktiven izgled, ne da bi za to žrtvovali dostopnost in semantičnost, kot se to mnogokrat zgodi pri Flashu. Pomembno je vedeti, da raba več knjižnic naenkrat ni priporočljiva, saj so nekatere med seboj nekompatibilne.

6.3. Flash

Vse zgornje primere in še mnogo več bi z lahkoto naredili tudi z orodjem Adobe Flash. Flash nam s svojim lastnim skriptnim jezikom omogoča mnogo več kot le animacije, z njim lahko ustvarimo interaktivne igre, omogoča nam vključevanje videa in avdia, poleg tega pa rešuje problem omejitve uporabe poljubne pisave. Zakaj torej vseh spletnih mest ne oblikujemo na ta način?

Flash je vsekakor zelo zmogljivo orodje, vendar nikakor ni primerno za vse vrste spletnih mest. Uporabimo ga na primer za izdelavo predstavitvene strani z malo vsebine, ne pa za izgradnjo vsebinsko bogatega portala. V nadaljevanju bomo pogledali zakaj uporaba Flasha za vsako ceno (kljub vsem njegovim pozitivnim platem) ni priporočljiva.

Čas nalaganja

Pogosto moramo na prikaz spletne strani narejene s tehnologijo Flash čakati dolgo časa, saj se mora celotna vsebina naložiti na začetku, nasprotno kot pri X(HTML)-ju kjer se vsebine nalagajo sproti. Za nekoga, ki išče zgolj informacije in mu ni mar za impresivne animacije, je čakanje lahko dovolj velik razlog, da spletno stran predčasno zapusti. Avtorji takšnih spletnih strani obiskovalca včasih poizkušajo zadržati z domiselnimi »preloaderji«, katerih funkcija je zabavati obiskovalca ko le-ta čaka na vsebino.

Nedostopnost

Informacije v Flashu so same po sebi nedostopne zaslonskim bralnikom. Od različice Flash MX naprej obstaja v Flashu možnost opisovanja vsebine, kar malo izboljša dostopnost, vendar je vse to le dodatno delo. (X)HTML je dostopen sam po sebi [35].

Težka optimizacija za spletne iskalnike

Spletne strani narejene v Flashu je zelo težko optimizirati za spletne iskalnike. Še nedavno Googlovi spletni pajki niso znali prebrati vsebine iz SWF datotek, zato je bilo za dobro rangiranje nujno postaviti še HTML različico strani. 30. Junija 2008 se je to spremenilo, saj je Google naznanil, da je sedaj sposoben indeksiranja tekstovnih elementov Flasha. S klikom na zadetek iskanja bo še vedno nemogoče navigirati neposredno na podstran, temveč bo obiskovalec usmerjen na vstopno stran, kar je očitna slabost. [36]

Težave z navigacijo

Slabost Flasha je vsekakor tudi nepričakovano delovanje gumbov naprej in nazaj. URL strani ostaja namreč vedno enak in se z našim navigiranjem v globino ne menjuje. To povzroči, da s pritiskom gumba nazaj zapustimo spletno mesto, namesto, da bi bili usmerjeni na stran, ki smo si jo ogledovali pred tem. Obstajajo rešitve tega problema, kot so ločevanje SWF datotek na glavne sklope, itd. Spletne strani narejene s Flashom imajo ponavadi svojo lastno navigacijo, ki pa ni standardizirana in je zato včasih težko razumljiva.

Če se le odločimo za Flash z njim ponavadi ustvarimo aplikacije z dodano vrednostjo, kot so na primer igre, vmesniki za izdelavo stripa, itd. Z izjavo Jakoba Nielsena, da je Flash 99% slab [37] se ne strinjam, menim pa, da potrebujemo za uporabo Flasha res dober razlog.

7. PROMOCIJSKA E-SPOROČILA

Ko pripravljamo promocijsko e-sporočilo se lahko odločimo za golo »plain text« sporočilo, sporočilo postavljeno v HTML jeziku ali oboje.

Pomembno je, vedeti, da e-sporočila ne smemo zasnovati na enak način kot spletno stran, saj takšna sporočila ljudje obravnavajo drugače. V poštnem nabiralniku je namreč že ogromno sporočil, tako zasebnih kot komercialnih, s katerimi tekmuje za naslovnikovo pozornost. Sporočilo mora zato že v začetku jasno izražati vsebino – bodisi že v »subject« vrstici bodisi v nagovoru. V nasprotnem primeru obstaja velika verjetnost, da smo že izgubili pozornost naslovnika.

7.1. Omejitve

Oblikovanje in izvedba promocijskih e-sporočil je poglavje zase. Pri načrtovanju moramo biti izjemno pazljivi, saj obstaja pri izvedbi e-sporočil ogromno omejitev. Izvor omejitev je podobno kot pri spletnih straneh odvisen od okolja kjer se sporočila prikazujejo. Obstaja veliko število odjemalcev spletne pošte in sicer 21 glavnih in množica ostalih manjših. Delimo jih na tiste ki so naloženi na našem računalniku (desktop clients) in pa druge, ki delujejo preko spleta (web clients). Pri vsakem od teh odjemalcev prihaja do velikih odstopanj prikazovanja HTML-ja in še posebej pri podpori CSS lastnosti. Eudora naprimer ne prikaže nobene CSS lastnosti, med tem ko so pri Thunderbirdu podprte prav vse [38].

Pri skupini CampaignMonitor so naredili odlično raziskavo in serijo testov, s katerimi so ugotovili katere CSS lastnosti so podprte v posameznih poštnih odjemalcih. Raziskava je zajela 21 odjemalcev. Mi se bomo osredotočili le na tiste, za katere menim, da so najpogosteje uporabljeni pri nas (Outlook '03, Outlook '07, Windows Mail, Mac Mail, Thunder-bird 2, Eudora, Yahoo mail, Gmail in Hotmail).

V spodnji tabeli (Tabela 2.1) so predstavljeni izjemno zanimivi rezultati o stopnji podpore različnih CSS lastnosti:

- Podprta CSS lastnost
- Nepodprta CSS lastnost

cursor	●	●	●	●	●	●	●	●	●
list-style-image	●	●	●	●	●	●	●	●	●
list-style-position	●	●	●	●	●	●	●	●	●
margin	●	●	●	●	●	●	●	●	●
z-index	●	●	●	●	●	●	●	●	●
left	●	●	●	●	●	●	●	●	●
right	●	●	●	●	●	●	●	●	●
top	●	●	●	●	●	●	●	●	●
background-position	●	●	●	●	●	●	●	●	●
border-spacing	●	●	●	●	●	●	●	●	●
bottom	●	●	●	●	●	●	●	●	●
empty-cells	●	●	●	●	●	●	●	●	●
position	●	●	●	●	●	●	●	●	●
caption-side	●	●	●	●	●	●	●	●	●
opacity	●	●	●	●	●	●	●	●	●

Tabela 7.1: CSS podpora v različnih poštnih odjemalcih [38]

Kako torej oblikovati promocijsko elektronsko sporočilo, da bo le-to enako vidno v vseh odjemalcih? Pisanje kode, kot da smo v letu 1999, je po vsej verjetnosti najboljši nasvet Mathew Pattersona [39]. Najbolj zanesljiv način, da bo naše sporočilo enako vidno v večini primerov je, da HTML oblikujemo s tabelami, uporabljamo HTML formatiranje in »inline« CSS. Nekateri spletni poštni odjemalci kot so Gmail in Hotmail CSS, ki je določen v glavi dokumenta enostavno odstranijo, zato je obvezno zapisovanje CSS lastnosti za vsak element posebej. Takšen način formatiranja zna biti zelo zamuden, da sploh ne omenjamo težavnega popravljanja, v primeru spremembe oblike. Poleg naštetega nam težave povzročajo nekateri odjemalci s slabo podprtimi CSS lastnostmi. Pri promocijskih e-obvestilih ne moremo določiti izgleda za vsak odjemalec posebej (tako kot to s pogojnimi komentarji lahko storimo za brskalnike), zato odjemalec z najslabšo podporo začrta meje izvedljivega in s tem omeji možnosti oblikovanja sporočila. Če se osredotočimo na zgornjo tabelo, lahko vidimo, kako različno so v odjemalcih podprte CSS lastnosti. Najhuje je videti, da so proizvajalci Microsoft Outlooka 2007 močno oklestili podporo CSS-a v primerjavi z njegovim predhodnikom odjemalcem Outlook 2003. Skoraj težko je verjeti, da ne moremo uporabljati osnovnih CSS lastnosti kot so width, height in background image. Poleg naštetega, sem pri testiranju v odjemalcu Outlook 2007 ugotovila, da se na elementih kot so <p>, <div> in nekaterih drugih ne upošteva lastnost padding. Rešitev je, da namesto padding-a, uporabimo margin.

8. ZNAČILNOSTI OBLIKOVANJA SPLETNIH PASIC

Spletne pasice predstavljajo eno izmed oblik oglaševanja na spletu. Pasica je lahko statična ali animirana, vanjo pa je vključena povezava, ki vodi na spletno stran, ki ponavadi vsebuje še več informacij kot pasica sama. Vključimo jih na spletne strani z zanimivo vsebino in velikim obiskom, da bi nanjo kliknilo čim več obiskovalcev.

Spletne pasice, kot smo že povedali, delimo na statične in animirane. Animirane pasice ponavadi pripravimo z orodjem Adobe Flash v formatu SWF, statične pasice pa v slikovnem formatu JPG, GIF ali animiran GIF (za preproste animacije sestavljene iz nekaj sličic). Statične pasice so lahko samostojna enota ali pa služijo kot podpora dinamični pasici in sicer za prikaz uporabnikom, ki nimajo naloženega Flash predvajalnika.

Izziv oblikovanja spletnih pasic, kot bomo videli v nadaljevanju, leži v njihovih zanimivih dimenzijah, še posebej pa nam delo otežijo omejitve njihovih velikosti in verzije Flasha v kateri morajo biti shranjene.

8.1. Dimenzije

Ime pasica izvira iz njene tipično podolgovate oblike, ki je primerna za umeščanje nad ali zraven vsebine. Formati pasic so do neke mere standardizirani. Najpogosteje uporabljeni so naslednji:

- 728 x 90 (Leaderboard)
- 468 x 60 (Full banner)
- 120 by 600 (Skyscraper)
- 160 x 600 (Wide skyscraper)
- 300x600 (Half-page)
- 300x250 (Medium rectangle)
- 500x500 (Floater)

[40]

Izziv oblikovanja spletnih pasic je vsekakor njihova nenavadna dimenzija. V začetku so bile uporabljene predvsem za izpisovanje besedila, za kar je ležeč format kot nalašč. Ko pa se enkrat lotimo pozicioniranja različnih grafičnih elementov pa kaj kmalu naletimo na različne ovire. Nekatere materiale, ki jih oblikovalec prejme, je včasih zelo težko pripraviti za potrebe oglaševanja s pasicami. Težava mnogokrat nastane tudi pri zakupu medijev, kjer zakupljene dimenzije niso v skladu s kreativno akcijo. Zgodi se na primer, da za promocijo nove plastenke neke pijače zakupijo pasico dimenzije 468x60, pri čemer je jasno, da bi pokončna oblika prišla kot naročena.

8.2. Omejitev velikosti

Omejitev velikosti datoteke spletne pasice je še en dejavnik, ki nam velikokrat prekriža pot do kvalitetne izvedbe. Omejitev velikosti oglasnih pasic na najdi.si je bila še pred kratkim pičlih 17kB, ki pa se je pred kratkim spremenila na 25kB. Omejitve na drugih slovenskih portalih se gibljejo od 25 do 40kB. Omejitev velikosti sama po sebi še ne pomeni nujno slabše kvalitete izvedbe pasice - v kombinaciji z zanimivim formatom pa je naloga včasih kar zahtevna.

SWF je format, ki je bil prvotno namenjen animiranju vektorskih elementov, omogoča pa tudi vključevanje bitnih slik in zvoka. Vektorska grafika sama po sebi ne zasede prav veliko prostora, večje težave se pojavijo, ko hočemo vključiti sliko ali zaporedje slik. Takrat je omejitev velikosti tisti dejavnik zaradi katerega se moramo zateči k improvizaciji. Sama sem izdelala že ogromno pasic, zato poznam kar nekaj trikov kako zmanjšati velikost SWF datoteke.

Nekaj nasvetov, kako zmanjšati velikost SWF datoteke.

- Kar je možno nadomestimo z vektorskimi elementi (prelivajoča ozadja, razne objekte)
- Sliko lahko razdelimo na področja, ki so bolj ali manj pomembna. Tiste z večjo pomembnostjo (obrazi, itd) pri izvozu kompresiramo manj kot tiste, ki ne nosijo tako pomembnih informacij.
- Slikovni elementi shranjeni v formatu PNG, ki vsebujejo velike poltransparentne površine so opazno večjih od tistih brez transparence. Če se, da se transparentci

izogibamo. Včasih (če imamo dokaj enotno ozadje) je namesto 24bit PNG-ja možno uporabiti GIF ali pa 8bit PNG, s čimer prihranimo kakšen kilobajt.

- Kompleksne vektorske forme lahko včasih poenostavimo, brez izrazito opaznih sprememb.
- Če je besedila zelo malo ali pa se le-to pojavlja v različnih pisavah, prihranimo nekaj prostora če ga razbijemo v krivulje.
- Če je besedila več ga je smiselno pustiti v pisavi, saj se tako vsaka črka zapiše le enkrat
- En simbol uporabimo večkrat. Da izrišemo cvet potrebujemo le en cvetni list, ki ga dupliciramo, po potrebi skaliramo in rotiramo. Naš cvet tako ne bo zasedel nič več prostora kot en listič, ki smo ga uporabili večkrat.
- Število okvirjev povečuje velikost pasice. Prazne okvirje, ki jih uporabimo zgolj za podaljševanje nekega odseka animacije, lahko nadomestimo z ActionScript kodo.

8.3. Verzija Flasha

Na večini slovenskih portalov zahtevajo pasice shranjene v Flash verziji 6 (sedaj obstaja že verzija 10). Verzija 6 ne omogoča nekaterih funkcij kot so metanje sence, zamegljenost (blur), gladkejši prehodi, itd, kar nam še dodatno oteži delo. Namesto vektorske sence moramo uporabiti sliko, kar zasede dodaten prostor, s katerim smo že tako ali tako precej omejeni. Omenjene funkcije so na voljo od Flash verzije 8 naprej.

8.4. Kreativna izraba pasic

Mnogi spletne pasice jemljejo le kot okrnjeno verzijo enosmernega televizijskega oglasa ali pa kot animiran plakat. Navodila za izvedbo se prepogosto glasijo: izpiše naj se naslednje besedilo, prikaže ta slika, na koncu naš logotip in vse naj se ponavlja v neskončnost. S takšnim pristopom izgubimo glavno prednost spleta in to je njegova interaktivnost. Prepogosto pozabimo, da uporabnik nadzira kaj, kako dolgo in na kakšne način bo neko vsebino spremljal. Ni le goba, ki vase vsrkava vsebino, ki jo je nekdo drug pripravil zanj.

Statistike kažejo, da na spletne pasice ljudje čedalje manj klikajo, saj jih pogosto sploh ne opazijo več. Pasice so videne kot nadležni elementi, ki jih ovirajo pri branju vsebin in zato jih podzavestno ignorirajo. Sama velikokrat skrčim okno brskalnika in tako skrijem oglase ob strani. Rezultat tega je proizvodjanje še bolj opaznih pasic, ki naj bi prinesle več klikov in tako

se vrtimo v začaranem krogu. Nekateri se poslužijo motečih utripanj, ki pa so na večini slovenskih portalov na srečo prepovedana. V pravilih portala vest.si piše: *oglasil ne smejo uporabljati hitre 'stroboskopske' animacije*. Drugi na pasico namestijo gumbe, ki so na las podobni tem v operacijskem sistemu in s tem zavajajo uporabnike. Na srečo so tudi na tem področju določena pravila: *Ne dovoljujemo rabe zavajajočih elementov vmesnika, kot so lažni gumbi 'zapri', 'išči' in podobno* [41].

Pasice se ljudem vsiljuje, ravno tako kot letake, za katere so že zdavnaj ugotovili, da jih povprečen človek dobi veliko preveč, da bi jih sploh pregledal, kaj šele si jih zapomnil.

Mislím, da bi bilo smiselno razmisliti o konceptu pasice, ki za uporabnika ni nadležna in ima obenem zanj neko dodano vrednost. Pasice z interaktivnimi igrami na primer nudijo zabavo, oglaševanje varčevanja z informativnim izračunom končnega privarčevanega zneska nudi informacijo. Kaj je skupnega zgoraj opisanima predlogoma? Izkoriščanje možnosti interakcije in s tem aktivno vključevanje uporabnika.

Sama sem sodelovala pri izdelavi pasice, ki naj bi poizkusila vzpodbuditi aktiven odnos do preprečevanja nasilja. S Sašom Petkom sva posnela nasilneža, ki uporabniku spletne strani grozi, ga izziva. Uporabnik ima možnost nasilneža spraviti za zapahe z enim samim premikom miške. *Tako preprosto je. Prijavi in ustavi nasilje*: Se glasi sporočilo. Nasilnež ni ves čas ujet v svojem prostoru 300x250px ampak v nekem trenutku poseže v osebni prostor obiskovalca – razbijati začne navigacijo nad sabo. S tem sva želela poudariti, da je nasilje nekaj kar se tiče prav vsakega izmed nas. Pasica je vidna na naslednjem naslovu: <http://www.emivega.com/00/diploma/nasilnez/>

Ključna se mi zdi kontrola in privoljenje obiskovalca nad zaganjanjem animacije, česar najin zgornji primer nima. Če obiskovalca vprašamo ali mu lahko nekaj pokažemo, bo le ta, če bo odgovoril pritrdilno, vsebino spremljal z večjo pozornostjo, saj smo z njim že vstopili v dialog. Res je, da bo naš oglas tako videlo manj ljudi, obstaja pa bistveno večja možnost, da bo pri njih naredil vtis, morda sprožil akcijo. Pasica na spodnji povezavi se mi zdi iz tega vidika naravnost odlična. http://www.bannerblog.com.au/2006/12/virgin_money_balleri.php

Še ena možnost kreativne izrabe pasic so pasice, ki so med seboj povezane. Akcija v prvi pasici povzroči reakcijo v drugi.

9. WEB 2.0

Izraz web 2.0 je prvi uporabil Tim O'Reilly, ki sedaj že nekaj let prireja Web 2.0 Summit (konferenco o Web 2.0). Z njim je označil novo generacijo spleta. Nekateri so menja, da je Web 2.0 le marketinški izraz, »buzzword« in da se splet ni bistveni spremenil. Spet drugi mislijo, da Web 2.0 pomeni zgolj sence, zaobljene kote in živahne barve. Videli bomo, da je Web 2.0 mnogo več, kot le njegove oblikovne značilnosti, katerim se bomo posvetili nekoliko kasneje.

Kaj se je torej tako drastično spremenilo, da je Tim O'Riley pred splet zapisal verzija št. 2.0? Gre za razvoj novih tehnologij? Do neke mere že, vendar je bolj pomembno nekaj drugega. WEB 2.0 je nov način razumevanja in uporabe tehnologije, je poizkus gradnje spleta okoli ljudi, namesto okoli tehnologije [42]. Podjetja so v proces razvoja spletnih orodij v vedno večji meri začela vključevati uporabnika in s tem proizvajati orodja, ki so povsem preprosta za uporabo, se pravi pisana na kožo povprečnemu uporabniku.

Orodja uporabnikom omogočajo komunikacijo, sodelovanje, objavljanje in izmenjavo besedil, slik in videa na način, ki ga prej niso poznali. Še včeraj smo za postavitev lastne spletne strani potrebovali ogromno znanja, sedaj pa obstajajo orodja s katerimi lahko to dosežemo na enostaven in hiter način. Če smo Web 1.0 uporabljali kot vir informacij, Web 2.0 uporabljamo za povezovanje ljudi. Web 2.0 je socialen splet, kjer je posameznik viden kot aktivni člen spleta. S spletom, kjer ima vsak posameznik realno možnost sodelovanja, je ena temeljnih vrednost sodobne družbe, svoboda govora, dobila svojo tehnološko platformo, svoj medij skozi katerega se lahko realizira.

Z WEB 2.0 se je dramatično spremenil tudi pogled na uporabnika. Če je imela včasih naša spletna stran funkcijo predstavitve produkta, mora imeti danes funkcijo vključevanja uporabnika. Včasih so bila v ospredju podjetja, sedaj pa se v ospredje postavlja uporabnika, ki postaja vse bolj aktiven člen v procesu nastajanja izdelka.

Že pred leti je Nike obiskovalcem spletne strani omogočil, da si prek spletnega vmesnika sami oblikujejo obutev (Nike ID). Uspeh je bil velik, Od leta 2004 se je prodaja prek spleta namreč kar potrojila, zabeležili so več kot 3 milijone unikatnih obiskovalcev na mesec [43].

9.1. Ajax

Izvor Ajax-a (asynchronous JavaScript and XML) sega v leto 2005, vendar tehnika asinhronnega nalaganja podatkov ni novost. Ajax ni izraz za novo tehnologijo, temveč za skupino tehnologij, uporabljenih na poseben način [44]. Ajax uporabljamo za izdelavo bogatih, interaktivnih spletnih aplikacij [45]. Kot smo omenili že v poglavju povečevanje interaktivnosti, uporabnika vse prevečkrat pustimo čakati. Ajax rešuje to težavo saj omogoča izvajanje operacij na odjemalčevi strani, zaradi česar se znatno poveča odzivnost na uporabnikove inpute in s tem izboljša uporabniška izkušnja. Čeprav Ajax prinaša večjo odzivnost, njegova uporaba povzroča nekatere težave v zvezi z dostopnostjo, ki niso nerešljive), saj za svoje delovanje uporablja JavaScript. Za naprave, ki ne podpirajo JavaScripta je zato smiselno narediti alternativno verzijo spletne strani brez Ajaxa [46].

9.2. Oblikovne značilnosti Web 2.0

Se še spomnimo starih spletnih strani, poplavljenih z GIF animacijami, naslovi v pisavi Comic Seans, nemogočimi ozadji, vidnimi števci obiska in vstopnimi Flash animacijami? Če jih primerjamo s spletnimi stranmi, ki prevladujejo danes, opazimo velike spremembe. *Katere so torej oblikovne značilnosti spletnih strani današnjega časa?*

Da bi odgovorili na zgornje vprašanje smo pregledali, med seboj primerjali in iskali skupne točke kopice spletnih mest kot so www.flickr.com, www.vimeo.com, www.tumblr.com, www.last.fm, www.apple.com, www.bbc.co.uk, www.dropsend.com, www.youtube.com, www.skype.com, wordpress.org, strawpollnow.com, 9rules.com, twitter.com, www.koornk.com (slovenska verzija twitterja), www.noovo.com in drugih.

9.2.1. Živahne barve

Za Web 2.0 oblikovanje so značilne živahne barve, ki uporabnika asociirajo na ključne besede kot so »novo«, »prijazno« in »zabavno«. Najbolj uporabljena barva je zagotovo rumenozelena (Slika 9.1, primer 2), ki naj bi bila po mnenju nekaterih celo »uradna« barva Web 2.0 [47]. Barve, ki so prav tako pogoste so: žareče oranžna (Slika 9.1, primer 1), cyan modra (Slika 9.1, primer 3) in magenta (Slika 9.1, primer 4).



Slika 9.1: Oblikovne značilnosti Web 2.0: Barve

9.2.2. Zaobljeni koti

Zaobljeni koti (Slika 9.2, primer 5) so še ena oblikovna značilnost, ki najdemo praktično povsod – uporablja jih celo nov portal BBC (<http://bbc.co.uk>). Zaobljeni koti dajejo občutek sproščenosti, mladostne energije, med tem ko ostri koti izgledajo bolj strogo in profesionalno. Zaobljene kote dosežemo z naslednjimi načini:

- S statičnimi slikami zaključkov
- Ali pa dinamično z uporabo JavaScripta, ki proizvede vrsto <div> elementov, pri čemer je vsak naslednji za 2px večji od prejšnjega, na zaključku pa obratno: <http://www.malsup.com/jquery/corner/>

9.2.3. Prehodi

Prehodi med barvami so priljubljeni že kar dolgo časa, saj popestrijo izgled spletne strani in dajo občutek globine. Uporabljamo jih na slikah, za ozadje strani, glavah, gumbih in ostalih elementih. V zgodnji fazi spleta smo lahko videli ogromno neokusnih prehodov sestavljenih iz vseh možnih barv, ki so bili v velikih primerih uporabljeni napačno in so zmanjševali berljivost besedila. Sedaj prevladujejo rahli prehodi iz določene barve v njeno temnejšo ali

svetlejšo različico (Slika 9.2, primer 6), prav tako pa je opaziti vedno več drznejših prehodov, kot so prehod iz oranžne v roza, zelene v modro itd.

9.2.4. Odsevi, sijaji na gumbih

Za množično uporabo odsevov na logotipih, slikah in drugih elementih, kot tudi sijaje na gumbih lahko okrivimo Apple, ki je začetnik tega trenda (Slika 9.2, primer 6 in 7). Gumbi, ki dajejo steklen videz so trenutno neverjetno priljubljeni, še bolj pogosto pa lahko vidimo že nekoliko izpet efekt odseva logotipov in ostalih elementov. Obstajajo celo skripte, ki avtomatsko proizvedejo opisan efekt. Zanimivo je videti, da se je odsev logotipa uporabljen celo na spletnih straneh slovenskih političnih strank: <http://www.zares.si/> in <http://www.slssms.si/>.

9.2.5. Sence

Tako kot prehodi, nam sence (Slika 9.2, primer 8) vzbudijo občutek prostora. Prostor z vsebino ni več v enaki ravnini z ozadjem ampak se nenadoma dvigne v ospredje. Sence so v spletnem oblikovanju prisotne že kar nekaj časa, vendar njihova priljubljenost s časom slabi.



Slika 9.2: Oblikovne značilnosti Web 2.0

9.2.6. Izpostavljeni elementi

Na mnogih referenčnih straneh smo v zadnjem času lahko opazili pojav (včasih prekomerno) izpostavljenih elementov kot so: naslovi, citati, besedni oblaki (vizualni prikaz pogostosti ključnih besed, ker se velikost besede s številom obiskov dinamično povečuje ali zmanjšuje) (Slika 9.3, primer 9), kot tudi ostalih, kot so gumbi, značke (Slika 9.3, primer 10) in vpisna polja (Slika 9.3, primer 12).

9.2.7. Ikone

Besedilo mnogokrat dopolnjujejo razne ilustracije ali ikone, ki vizualno ponazarjajo temo, obenem pa popestrijo izgled strani (Slika 9.3, primer 11).



Slika 9.3: Oblikovne značilnosti Web 2.0

9.2.8. Visoke noge

Po principu klasičnega spletnega oblikovanja se vse pomembne elemente pozicionira na vrh strani, da so le ti obiskovalcu takoj vidni. Informacije na dnu spletne strani, kot so informacije v nogi, pa naj bi bile po tej logiki manj pomembne. Ponavadi tja dokaj neopazno zapišemo letnico izdelave, avtorske pravice in morda avtorja spletne strani. Z Web 2.0 oblikovanjem je tega konec. Noga spletne strani se spreminja iz dolgočasnega neopaznega elementa, v področje, ki uporabniku bodisi poda nove informacije bodisi pomaga pri navigaciji. Zakaj bi na kocu dolgega besedila postavili povezavo na vrh strani, ko pa lahko v nogi ponovimo navigacijo (Slika 9.4, primer 15), jo morda še bolj razčlenimo. Noga je tudi prostor kamor lahko umestimo zadnje komentarje iz foruma, objave za prosta delovna mesta (Slika 9.4, primer 13), praktično karkoli, kar je vredno izpostaviti (Slika 9.4, primer 14).



Slika 9.4: Oblikovne značilnosti Web 2.0

9.2.9. Interaktivnost in dinamičnost

Novejša spletna mesta mnogokrat vključujejo premikajoče vizualne priboljške, ki jih lahko dosežemo z (X)HTML, CSS in Javascript-om (opisano v poglavju povečevanje interaktivnosti in dinamičnosti). Uporaba orodja Flash, zgolj kot sredstvo animacije ni več opravičljivo, saj ga lahko v mnogih primerih brez opazne razlike nadomestimo z bolj optimalnim JavaScriptom (meniji, galerije, itd).

9.2.10. Preprostost

Poleg vseh zgoraj opisanih elementih je treba povedati, da Web 2.0 spletna mesta praviloma niso prenasočena, saj vsebujejo veliko beline. Poravnava strani je ponavi sredinjska.

Ugotovimo lahko, da je celotna filozofija Web 2.0 sloni na prijaznosti do uporabnika, ki je podlaga za njegovo aktivno vključevanje. Opravila na spletu so kar naenkrat postala otročje lahka, kar nam sporočajo tudi oblikovne značilnosti Web 2.0. Besedilo je postalo večje in bolj berljivo, strukture strani bolj pregledne, pomembni elementi (ki uporabnika vodijo do ključnih informacij) velikanski, koti mehko zaobljeni, barve pa sijoče in vabljive, kot še nikoli poprej.

10. WEB 3.0, WEB 4.0

Splet je korenito spremenil naše navade in način dela, kot smo že zapisali v uvodu, vendar splet kot ga poznamo danes, še zdaleč ne izkorišča svojih potencialov. Zagotovo je, da lahko v obdobju po Web 2.0 pričakujemo velike spremembe.

Izraz Web 3.0 je prvi uporabil John Markoff, novinar časopisa The New York Times in sicer za poimenovanje prihajajoče tretje generacije spleta. Kdaj se bo torej zgodil web 3.0? Nova Spivack pravi, da v letih 2010 do 2020 [48]. Obstaja več predvidevanj kaj bo značilno za tretjo generacijo spleta. Dean Giustini pravi, da bo to obdobje semantičnega spleta, metapodatkov (podatkov o podatkih), s tem učinkovitejšega obdelovanje podatkov in razvoja avtomatizacije, kar bi odprlo nove poti k razvoju umetne inteligence [49]. Nekaterim se ideja o inteligentnem spletu zdi neverjetna, vendar IBM in Google že sedaj sklepata o novih glasbenih uspešnicah z obdelovanjem podatkov iz študentskih spletnih strani [50].

Situacija s katero se soočamo danes je takšna, da so podatki na spletu razumljivi ljudem, ne pa tudi strojem. Semantičen splet bo splet razumljiv računalnikom. In kaj pravzaprav je semantika? Mooly E. Holzschlag pravi: Semantika se ukvarja s pomenom elementov in tem, kako dobro ti elementi opisuje vsebino znotraj njih [14].

V (X)HTML jeziku so vsebine znotraj elementov kot so `<div>` ali `` poljubne zato ne moremo vedeti kaj se v njih skriva, dokler jih ne preberemo. Je to mogoče naslov, ime ali kaj tretjega? Problem jezika (X)HTML je, da je v svoji osnovi premalo semantičen. Z elementi kot so `<h1>`, `<h2>`, itd nam poda okvirno strukturo spletne strani, vendar o njeni vsebini še vedno ne vemo prav dosti. Ključne besede znotraj glave (X)HTML-ja, ki jih določimo v elementu `<meta>` so majhen korak naprej (vključen šele leta 1999). Prav tako je pomembno ločevanje oblike in vsebine, ki jo dosežemo z zunanji CSS datotekami. Poznamo tudi elemente `<address>` in `<cite>`, ki kar dobro opišejo svojo vsebino, vendar takšnih elementov (vsega skupaj jih je 40) še zdaleč ni dovolj [14].

Ideja o semantičnem spletu je pravzaprav že precej stara. Že Tim Berners-Lee je imel vizijo o spletu, ki bi računalnikom omogočal razumevanje podatkov. »Sanjam o spletu, v katerem bodo računalniki sposobni analizirati vse podatke, ki krožijo po internetu: vsebino, povezave in interakcije med ljudmi in računalniki. Semantični splet, ki bi to omogočal, še mora nastati,

a ko bo prišel ta dan, bodo v vsakodnevnem življenju, prometu, birokraciji, stroji komunicirali s stroji.« [51] Splet danes dejansko predstavlja ogromno odlagališče podatkov (besedila, slikovnega materiala, zvočnih zapisov), ki za računalnike nimajo pomena. Samo predstavljajmo si kaj vse bi bilo mogoče, če bi lahko vse te podatke združili v celoto in jih uporabili. V tistem trenutku bi se znanje človeštva dvignilo na zavidljivo raven.

Kako torej do semantičnega spleta, podatkov, ki jih razumejo računalniki? Rešitev je v že prej omenjenih metapodatkih. V prihodnosti bomo zagotovo priča velikanskemu porastu podatkov o podatkih, saj tehnologije pomembne za razvoj semantičnega spleta že obstajajo. XML (eXtensible Markup Language), RDF (Resource Description Framework) in OWL (Web Ontology language) so nekatere izmed njih.

Za začetek se za novi označevalni jezik *HTML 5* napoveduje dodatne elemente, ki bi povečali semantično vrednost vsebine znotraj njih. Nekateri izmed novih elementov bodo <footer>, <video>, <audio> in <nav>, medtem ko se bo opustilo elemente kot so , <center> in <strike>, ki jih več kot odlično pokrijemo s CSS-om.

Nekakšna vmesna rešitev med zgoraj naštetimi tehnologijami in sedanjim stanjem so *Microformati*, ki elementom obstoječe HTML kode, dodajo pomen. S primerno programske opreme (vtičnik za FireFox) lahko kontaktne informacije o podjetju v obliki vizitke enostavno prenesemo v poštni odjemalec (microformat hCard), informacije o dogodku pa v naš koledar (microformat hCalendar). Obstajajo še drugi microformati kot so hResume, za življenjepise, hReview za kritike in mnogi drugi. Microformati delujejo po principu razširjajo obstoječega (X)HTML-ja, namesto uvajanja novega jezika. Implementacija je povsem preprosta saj zahteva le dodatne class oznake. [52]

Primer za hCard:

```
<div class="vcard">
  <div class="fn">Janez Novak</div>
  <div class="org">Čebelarstvo Novak</div>
  <div class="tel">00386 1 555 8953</div>
  <a class="url" href="http://www.cebela.si/"> http://www.cebela.si/</a>
</div>
```

Pomembne vloge Microformatov pri ustvarjanju semantičnega spleta se zaveda sam Bill Gates, ki je izjavil »Potrebujemo microformate« [14]

Pri Mozilla labs so pred kratkim predstavili *Ubiquity*, orodje s katerim so naloge, kot so vstavljanje zemljevida v e-sporočilo, kritik, objava na Twitter in prevajanje besedila stvar klika ali dveh. Prej smo za takšne naloge porabili ogromno časa, sedaj pa nam pri tem pomaga intuitivni uporabniški vmesnik, ki »razume« preproste jezikovne ukaze kot so email, map, twitter, translate itd. Ustvarjalci aplikacije pravijo, da pri spletnih brskalnikih uporabnik vpiše kaj bi rad našel, v Ubiquity pa kaj bi rad počel. In prav za to tudi gre. Uporabniški vmesnik nam služi kot nekakšen inteligentni agent, ki prepozna kaj z informacijami na spletu želimo početi in s tem znatno izboljša uporabniško izkušnjo.[53]

Več o Ubiquity na: <http://labs.mozilla.com/2008/08/introducing-ubiquity/>

Resnično revolucionarne bodo šele aplikacije, ki bodo znale uporabljati mikroformate. Pri Ubiquity uporabnik z vpisom ukaza izbranim podatkom določi njihov pomen (če napiše map orodje ve, da je v označenem besedilu informacija o lokaciji) v prihodnosti pa niti to ne bo več potrebno.

V prihodnosti se nam po vsej verjetnosti obeta popolna selitev aplikacij, kot tudi datotek uporabnikov na splet. Web 4.0 naj bi bil WebOS, ogromen operacijski sistem, organizem, do katerega bodo uporabniki dostopali prek terminalov. V prihodnosti bo v omrežju poleg računalnikov vedno več drugih naprav kot so prenosni telefoni, fotoaparati, avtomobili, stereo-naprave, hladilniki, itd.[54]

11. PRIHAJAJOČI TRENDI SPLETNEGA OBLIKOVANJA

Trendi spletnega oblikovanja razkrivajo dobro sprejete pristope in vzorce oblikovanja pri spletni publiki.[55] Takšni pristopi so v uporabi tako dolgo dokler ne pride val novih, ki se s starimi pomeša in jih naposled izpodrine.

V poglavju Web 2.0 smo predstavili trenutne trende spletnega oblikovanja, v nadaljevanju pa bomo poizkusili napovedati trende, ki k nam šele prihajajo in jih primerjali s tistimi, ki so bili prisotni v letih pred tem.

11.1.1. Umirjene, nežne, pastelne barve

Kot smo že zapisali, nam je Web 2.0 revolucija postregla z živahnimi, intenzivnimi barvami, ki so spletna mesta navdihnile z življenjem. Oblikovanje z barvami kot so žareče oranžna, magenta, rumeno zelena in cyan modra je postalo norma in predvidevamo lahko, da bodo oblikovalci kmalu začeli iskati nekaj drugega, nekaj s čimer bi izstopili iz poplave že prevečkrat videnih živahnih barv. Primer oblikovanja z umirjenimi barvami je zagotovo novi Wordpress (Slika 11.1, primer 1). Brezplačne aplikacije, so mnogokrat videne kot nezanesljive. Resni, umirjeni toni zagotovo pomagajo razbiti to stigmo.

Spletni mesti <http://designbyninjas.com/> (Slika 11.1, primer 2) in <http://www.studiojwal.com/> (Slika 11.1, primer 3) uporabljata zamolke pastelne tone v modri in zeleni kombinaciji.



Slika 11.1: Prihajajoči trendi spletnega oblikovanja: barve

11.1.2. Veliko, še večje

Pri uporabi velikih elementov gre za princip vizualnega izpostavljanja pomembne informacije z velikostjo kot tudi barvo. Uporaba velikih elementov je na drugih področjih oblikovanja kot so plakati in platnice knjig prisotna že dolgo, sedaj pa se je končno naselila tudi na splet, kjer bo po vsej verjetnosti ostala tudi v prihodnje.

11.1.3. Prehodi ali enotne barve

Dejstvo je, da so prehodi trenutno zelo priljubljeni. Jih bomo v prihodnosti videvali še več ali pa jih bodo oblikovalci opustili in začeli v večji meri uporabljati enotne barve, kot napovedujejo nekateri, bo pokazal čas.[55] Sama menim, da prehodi ne bodo kar izginili čez noč, saj spletnim stranem mnogokrat dajejo bolj prijazen, naraven izgled, ki so ga ljudje navajeni iz realnega sveta, kjer so objekti redko enotnih barv zaradi svetlobe in senc, ki padajo na njih.

11.1.4. Tipografija

Do pred kratkim smo bili na področju uporabe tipografije na spletu še zelo omejeni. S tehnologijo kot je SiFR in raznimi orodji, ki omogočajo zamenjavo besedila s slikami, se ta omejitev zmanjša. Poleg tega prihajajo z novimi operacijskimi nove pisave, ki bodo nekoč morda tako razširjene, da jih bomo lahko brez zadržkov uporabili. Mogoče bo prihodnost presenetila še s kakšnim novim načinom, kako rešiti ta problem, kar bi omogočilo večje eksperimentiranje s tipografijo, kot je to mogoče sedaj.

11.1.5. Video

Z vedno hitrejšimi povezavami bomo priča tudi bolj pogosti uporabi videa na spletnih straneh, ki bo služil kot del uporabniškega vmesnika.

Predstavili smo trenutne trende spletnega oblikovanja, kot tudi nekatere, ki k nam šele prihajajo. Za konec se nam zastavlja pomembno vprašanje: *Naj se pri spletnem oblikovanju oziramo na trende ali naj naš izdelek izžareva odsotnost trendov in s tem izraža našo individualnost? Naj sledimo čredi ali gremo svojo pot? Poplava vizualnih podob na spletu zahteva drugačnost, ki je v nekaterih primerih edino zagotovilo za vidnost, zato je cilj mnogih spletnih oblikovalcev ustvariti nekaj popolnoma novega, nekaj kar ni ustvaril še nihče. To je*

kar težka naloga, saj je bilo narejenega že ogromno. Po drugi strani pa je splet še mlad medij, ki se spreminja iz dneva v dan. Njegov neprestan tehnološki razvoj nam omogoča veliko manevrskega prostora, da ustvarimo nekaj unikatnega. Pozitivna stran trendov je, da nam prikazujejo pristope in vzorce, ki so med ljudmi dobro sprejeti. Izkoristimo jih lahko tudi kot primer kaj delajo drugi in to nadgradimo z lastno kreativnostjo.

Za uspeh je ključno natančno poznavanje principov dobrega oblikovanja, ki nikoli ne zastarajo. Če temu dodamo kakšen element oblikovanja, ki je trenutno v modi ne moremo zgrešiti. Zanašanje zgolj na trende pa ni priporočljivo, saj bodo v tem primeru naši izdelki v nekaj letih, če ne še prej, videti zastareli. Trendov ne smemo jemati kot seznam stvari za kopiranje, temveč jih moramo ovrednotiti kot možne principe oblikovanja in tako priti do lastnih zaključkov. Trendi so trenutni, dobro oblikovanje je večno.

12. PRAKTIČNI DEL

Kot praktični del diplomske naloge bo predstavljen postopek oblikovanja in izvedbe spletnega mesta Vitapark. Izpostavila bom elemente in principe oblikovanja in izvedbe opisane v diplomskem delu, ki sem jih uporabila na omenjenem projektu.

12.1. Naloga

Ekipa sodelavcev podjetja Innovatif je dobila nalogo osvežitve podobe Penziona Park, ki na trgu deluje že 14 let. Sama sem bila zadolžena za oblikovanje nove spletne strani in »front end« izvedbo. Na projektu so sodelovali: Barbara Kvas (načrt spletnega mesta), John Alao in Danijel Kurinčič (programiranje), Tadej Šulman (Izbor imena), Alenka Rovan (Oblikovanje nobega CGP-ja) in Marko Kadič (mentorstvo).

Na spodnji sliki lahko vidimo izgled stare spletne strani in znak penziona:



Slika 12.1: Stara spletna predstavitev in znak penziona

12.2. Zahteve in cilji

12.2.1. Vsebinske zahteve

Spletno mesto mora imeti naslednje karakteristike:

1. Nuditi mora vse informacije, ki so povezane s penzionom in bivanjem v njem.
2. Ponuditi mora enostaven način rezerviranja storitev.
3. Spletno mesto mora ponuditi informacije, ki opisujejo kraje in storitve v okolici penziona.

12.2.2. Tehnične zahteve

1. Zagotovljena mora biti interaktivnost med uporabniki spletnega mesta in penzionom. Omogočale jo bodo funkcionalnosti kot so rezervacija s pomočjo spletnega obrazca.
2. Spletno mesto mora biti podprto z vmesnikom za upravljanje z vsebino, ki bo omogočalo urednikom, da v večji meri samostojno upravljajo z vsebinami spletnega mesta.
3. Omogočena mora biti uredniška politika (urednik, pisec, ...).
4. Spletno mesto mora zagotavljati podporo objavi vsebin v različnih formatih (besedila, dokumenti, video in avdio datoteke). Spletno mesto mora vsebovati galerijo slik.
5. Na spletnem mestu mora obstajati možnost izpostavljanja napovednikov.
6. Omogočena mora biti večjezikovnost (slovenski, angleški, nemški in italijanski jezik)

12.2.3. Poslovni cilji

1. Cilj spletnega mesta je povečati prepoznavnost penziona.
2. Povečati mora povpraševanje po storitvah penziona, posredno pa se mora povečati tudi njihova prodaja.
3. Spletno mesto mora obiskovalce pritegniti k izkoriščanju storitev prenočišča, gostinske ponudbe, okoliških objektov in naravnih danosti v okolici penziona.

12.2.4. Komunikacijski cilji

1. Gostom, potencialnim gostom in naključnim uporabnikom spletnega mesta bo spletno mesto nudilo informacije, ki jih potrebujejo tudi že pred obiskom penziona.
2. Vsebine bodo zasnovane tako, da bodo spletnim uporabnikom pomagale pri organizaciji svojega prostega časa v času bivanja v penzionu.

12.3. Izhodišča za oblikovanje

12.3.1. Ime penziona

V prvi fazi je Tadej Šulman izbral novo ime penziona: Vitapark. O njegovem pomenu je zapisal:

Iz dveh besed sestavljeno ime. »Vita« – iz latinščine privzeta tujka je široko znana in uporabljana in se navezuje na življenje oz. živost. V kombinaciji z besedo »park« nakazuje oz. asociira na večnivojski prostor narave, zdravja, ugodja, sproščenosti in življenjske energije. »Park« se obenem lepo poveže s predhodnim imenom in s tem ohranja del že pridobljene prepoznavnosti. Sestavljanje za razliko od starega imena nudi širšo možnost komuniciranja, ta pa je glede na vizijo o širitvi ponudbe zaželjena. Ime je lahko izgovorljivo, pomenskost pa je razumljiva tudi potencialnim gostom iz tujine.

Simbolni oz. asociativni atributi imena:

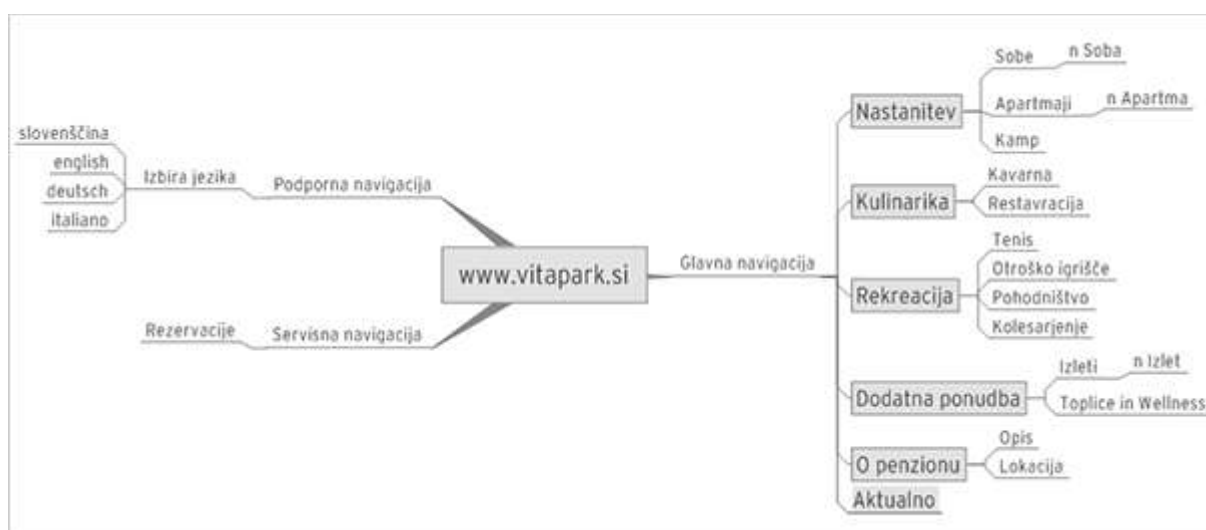
življenje, živost, narava, zdravje, zeleno, aktivnost, prostranost, večnamembnost.....

12.3.2. CGP

Alenka Rovan je izdelala novo celotno grafično podobo (Slika 12.3, primer 1), pri čemer sem pomagala izbrati barve. Nova celotna grafična podoba mi je služila kot izhodišče za oblikovno podobo spletnega mesta. Podoba metulja, ki je prisoten v znaku, s svojimi mehкими linijami izraža nežnost, lahкотnost, in igrivost. Ta občutek sem želela prenesti v oblikovanje spletne strani.

12.3.3. Načrt strukture vsebin

Na spodnji sliki (Slika 12.2) sta predstavljeni vsebinska zasnova in navigacijski model spletnega mesta Vitapark.



Slika 12.2: vsebinska zasnova in navigacijski model

12.4. Rezultati

12.4.1. Pregled uporabljenih tehnologij in principov oblikovanja

Za izvedbo spletnega mesta Vitapark sem uporabila tehnologijo (X)HTML v kombinaciji s CSS-om. Stran je zgrajena s »tableless, CSS based« tehniko, ki za osnovne gradnike uporablja elemente <div>, oziroma tiste, ki so bili zasnovani za določeno funkcijo. Tehnologija Flash je bila uporabljena za vizualne poudarke kot so animacija logotipa in prikazovanje različnih fotografij na naslovnici. Za spletnim mestom stoji CMS sistem Ažur 2 (izdelek podjetja Inovativ), ki deluje na tehnologiji PHP. Za prikazovanje poljubne pisave v naslovih in citatu na naslovnici sem se poslužila tehnologije SiFR. Galerija slik in razširitev besedila (ki se zgodi brez dodatnega pošiljanja zahteve na strežnik) je narejena z JavaScriptom, ki uporablja knjižnico jQuery.

Tehnologije, ki smo jih uporabili na projektu:

- (X)HTML
- CSS
- PHP
- Flash
- SiFR 3.0 Novemberborn
- JavaScript (knjižnica jQuery – Lightbox vtičnik)

Principi oblikovanja vsebin:

- »Tableless CSS based« postavitev
- CSS Sprites

12.4.2. Opis postopka izvedbe in diskusija rezultatov

Pri oblikovanju grafične podobe spletnega mesta sem uporabila veliko beline, oz negativnega prostora, ki daje spletni strani zračnost. Občutek igrivost sem poskušala doseči z animacijo metulja v znaku, ki na vsake toliko časa zamahne s krili. Barvo za poudarke sem črpala iz znakovnega dela logotipa (#449ead). Za ozadje sem uporabila vzorec v umirjeno svetlo sivi barvi (#cec6c6) s pridihom oranžne (komplementarna barva modri iz logotipa). Vzorec na prvi pogled sestavljajo različno veliki krogi, ob natančnejšem pogledu pa vidimo, da kroge v resnici tvorijo metulji iz znaka. Vzorec sem ponovila v X in Y smeri (Slika 12.3, primer 2). Za označevanje novosti ponudbe sem uporabila »trendy Web 2.0« zvezdasto obliko v katero sem zapisala *sveže* in *okusno* (Slika 12.3, primer 3).



Slika 12.3: Oblikovni elementi spletnega mesta Vitapark

Za »rollover« efekt na gumbu pošlji sem se poslužila metode *sprites*, kjer obe stanji gumba shranimo v eno sliko in ju prikazujemo s CSS pozicioniranjem (Slika 12.4, primer 4).

```
#gumb_poslji {
    background: url(/Static/images/gumb_poslji.gif) no-repeat 0 -27px;
    height: 27px;
    margin-bottom: 10px;
    width: 84px;
}
#gumb_poslji:hover {
    background: url(/Static/images/gumb_poslji.gif) no-repeat 0 0;
}
```




Slika 12.4: Oblikovno / tehnični elementi spletnega mesta Vitapark

Za prikaz večjih slik v galeriji sem uporabila Lightbox, ki uporablja JavaScript knjižnico jQuery (Slika 12.4, primer 5). Za pravilno delovanje je bilo v glavo dokumenta potrebno vključiti naslednji datoteki.

```
<script type="text/javascript" src="Static/js/jquery-1.2.3.pack.js"/>
<script type="text/javascript" src="Static/js/jquery.lightbox-
0.4.pack.js"/>
```

Izgled Lightboxa je bilo potrebno malenkost prilagoditi, kar smo storili z spreminjanjem CSS-a in zamenjavo nekaterih slikovnih elementov (angleški napisi na gumbih). jQuery je bil uporabljen tudi za razširitev besedila ob kliku na povezavo več v poglavju aktualno (/si/aktualno/). Za naslove in citate je uporabljena tehnologija SiFR, s pomočjo katere se besedilo prikaže v pisavi Lisboa Light (pisava, ki je uporabljena v logotipu). Uporabila sem zadnjo različico SiFR 3 Novemberborn, ki mi je omogočila več fleksibilnosti pri formatiranju besedila kot različica SiFR 2 (uporabljena za demonstracijo v poglavju SiFR). Pri aplikaciji SiFR-a smo naleteli na manjše zaplete pri prikazovanju šumnikov in drugih znakov značilnih za nemški in italijanski jezik. Pisavo Lisboa je bilo potrebno najprej ošumiti nato pa izbrati pravilne nastavitve za izvoz SWF datoteke. Vključiti je bilo treba večje število znakov kot je privzeto, zaradi česar se je znatno povečala velikost SWF datoteke (prej 25KB, sedaj 100KB).

Rezultat dela je delujoče spletno mesto, ki je vidno na naslovu: <http://www.vitapark.si>.



Slika 12.5: Prva tipična stran: naslovnica

Na zgornji sliki (Slika 12.5) je prikazan izgled prve tipične strani (naslovnice), na naslednji (Slika 12.6) pa je predstavljena druga tipična stran (podstrani).



Slika 12.6: Druga tipična stran: podstran

Menim, da spletno mesto Vitapark s svojo kvalitetno izvedbo in premišljeno zasnovo izpolnjuje naročnikove zahteve in dobro služi svojemu namenu, kar pa še ne pomeni da ni prostora za določene izboljšave. Če bi se znova lotila podobnega projekta bi namesto Flasha, ki na naslovnici prikazuje različne fotografije, uporabila eno izmed JavaScript knjižnic, s katero bi prišla do identičnega efekta. Poleg tega pa bi zmanjšala višino fotografije na naslovnici, saj zaradi te območje 550ih pixlov ni dovolj izkoriščeno. Proces oblikovanja je potekal na zaslonu z ločljivostjo 1600x1200, kjer je bilo brez »scrollanja« vidno večje območje kot na večini zaslonov, ki so v uporabi.

13. SKLEP

V diplomskem delu smo videli, da se na področju spletnega oblikovanja soočamo z veliko omejitvami, ki mnogokrat izhajajo iz značilnosti spleta (pasovna širina, barvna globina zaslonov in različne zaslonske ločljivosti, in nepredvidljiv format kot posledica neznane ločljivosti). Težave se nato pojavijo tudi pri izbiri poljubne tipografije, kot tudi pri uporabi formata PNG s transparentnim ozadjem. Prilagajanje izgleda spletne strani za vse brskalnike je včasih zelo težka naloga, produkcija promocijskih e-sporočil pa je sploh poglavje zase.

Ugotovili smo, da mora biti skupna velikost spletne predstavitev, zaradi neznane hitrosti povezave uporabnika, kar se da majhna, čemur pripomore pravilen izbor in kompresija slikovnih formatov kot tudi optimizacija HTML in CSS kode. Pri množici zaslonskih ločljivosti se ravnamo po tisti, ki je najbolj razširjena in za njo prilagodimo spletno predstavitev. Prevladujoča ločljivost je trenutno 1024x786 (44,5% globalno in 42% pri nas), pri čemer je lahko širina spletne strani 960px ali manj, v območje višine 550px pa je priporočljivo vključiti vse glavne elemente spletnega mesta. Zaradi slabših zaslonov s 16 bitno barvno globino, ki so še vedno relativno razširjeni (globalno 7%, pri nas 9%) in neznanih nastavitev zaslonov moramo poskrbeti za dovolj kontrastno spletno predstavitev. Ugotovili smo, da CSS zasnova spletne strani prinaša mnoge prednosti pred zasnovo s tabelami, zato jo uporabimo brez razmišljanja. Na področju tipografije smo kljub možnosti definiranja zaloge sistemskih pisav še vedno zelo omejeni. Obstajajo metode prikazovanja poljubne pisave, kot so tipografija na strežniku, ki ne deluje v vseh brskalnikih, SiFR in zamenjava besedila s slikami, ki lahko ob nepravilni implementaciji pomeni izgubo semantične vrednosti elementov. Uporabni sta obe zadnje naštetih rešitvi, vendar z njima ne rešujemo problema uporabe poljubne pisave na splošno, temveč ju uporabljamo le prikazovanje krajših odsekov besedila. Kar se tiče podpore transparence smo zaključili, da lahko transparentne HTML elemente uporabljamo brez večjih težav. Transparenten format GIF deluje povsod, transparenten PNG pa v vseh novejših brskalnikih, v Internet Explorer 6 pa s pomočjo ene izmed opisanih metod. Transparentne slike, ki jo uporabimo za ozadje v omenjenem brskalniku ni mogoče pozicionirati in ponavljati. Po pregledu tehnologij za povečevanje interaktivnosti vidimo, da poleg Flasha obstaja tudi Ajax in razne JavaScript knjižnice, s katerimi lahko proizvedemo impresivne efekte, obenem pa ne ogrozimo semantične vrednosti in dostopnosti podatkov. Na področju promocijskih e-sporočil se soočamo s slabo podprtimi CSS lastnostmi, zato naj bo oblikovanje kar se da preprosto. Pri

izvedbi promocijskih e-sporočil lahko pozabimo vse kar smo povedali o napredni CSS postavitvi in se v veliki meri poslužimo tabel in HTML formatiranja elementov. Oblikovanje in izvedba spletnih pasic je mnogokrat velik izziv zaradi kombinacije formata, omejitve velikosti in verzije Flasha v kateri morajo biti shranjene. Prav tako je v velikih primerih neustrezno zasnovan koncept pasice, kjer uporabniku le podajamo informacije namesto, da bi izkoristili glavno prednost spleta – interaktivnost, in ga aktivno vključili. Če je Web 2.0 revolucija na oblikovnem področju postregla z živahnimi barvami, zaobljenimi koti, prehodi, odsevi, sencami, izpostavljenimi elementi in visokimi nogami, lahko v prihodnje pričakujemo umirjene, pastelne barve, še več velikih elementov, enotne barve kot tudi prehode in porast uporabe videa kot del uporabniškega vmesnika.

Dejstvo je, da smo soočeni z velikim številom omejitev in morda je prav zaradi njih spletno oblikovanje tako vznemirljivo. Uspešno oblikovanje je nenazadnje le reševanje problemov in včasih je treba le pomisliti izven okvirjev in kaj kmalu se najde rešitev. Omejitev ne moremo vedno enačiti z omejevanjem kreativnosti, saj so mnogokrat lahko celo vzrok zanjo.

Razmišljanje Kennetha Goldsmitha iz uvoda, »*If It Doesn't Exist on the Internet, It Doesn't Exist*«, bi se še ne dolgo nazaj marsikomu zdelo smešno, vendar je čas pokazal, da je imel Goldsmith še kako prav. Splet je medij nešteti možnosti in priložnosti, česar se ljudje vse bolj zavedajo. V prihodnosti lahko pričakujemo samo še povečevanje pomembnosti spletne prisotnosti zaradi vse večjega števila uporabnikov in neizogibne konvergence medijev. Splet je šel od svojega začetka pa vse do danes skozi korenite spremembe in še sanja se nam ne kaj se bo z njim dogajalo v prihodnosti. Morda bo splet, ob vse večjem upoštevanju standardov, postal bolj prijazno okolje za spletne oblikovalce in načrtovalce, morda tudi ne. Zagotovo je le eno in to je: spremembe so neizogibne. Splet se bo spremenil in z njim spletno oblikovanje. Diplomsko delo je dalo odgovor na nekatera aktualna vprašanja, ki že čez par let morda ne bodo več relevantna. Če želimo ustvarjati odlične spletne vsebine, se proces učenja ne sme nikoli končati. Nenehno moramo biti na preži in spremljati razvoj spletnih tehnologij. Kitajski pregovor pravi: »*When the winds of change are blowing, some people are building shelters, and others are building windmills*«. In kateri izmed njih želimo biti mi?

LITERATURA

- [1] http://www.iab.net/about_the_iab/recent_press_releases/press_release_archive/press_release/339821
- [2] <http://slo-tech.com/script/forum/izpisitemo.php?threadID=298892>
- [3] http://epc.buffalo.edu/authors/goldsmith/if_it_doesnt_exist.html
- [4] <http://www.senca.net/digitalni-marketing/slovar/>
- [5] http://www.finance.si/manager/177735/Anketa_Kak%B9ne_spremembe_lahko_pri%E8akujem_o_do_leta_2020
- [6] http://www.akamai.com/html/about/press/releases/2006/press_110606.html
- [7] <http://www.faqs.org/faqs/jpeg-faq/part1/>
- [8] <http://sl.wikipedia.org/wiki/JPEG>
- [9] <http://en.wikipedia.org/wiki/GIF>
- [10] http://en.wikipedia.org/wiki/Portable_Network_Graphics
- [11] http://en.wikipedia.org/wiki/Multiple-image_Network_Graphics#Software_that_supports_MNG
- [12] <http://www.websiteoptimization.com/speed/tweak/size/>
- [13] C Schmitt, *Adapting to Web Standards: CSS and Ajax for Big Sites*, New Riders, Berkley, California, 2008
- [14] A Clarke, *Transcending CSS*, New Riders, Berkley, California, 2007
- [15] <http://www.thecounter.com/stats/2008/July/colors.php>
- [16] <http://desktoppub.about.com/od/emailclasses/ig/Rules-of-DTP-Illustrated/Line-Length.htm>
- [17] <http://www.w3counter.com/globalstats.php>
- [18] <http://worlds-highest-website.com/>
- [19] <http://en.wikipedia.org/wiki/Cross-browser>
- [20] <http://www.w3counter.com/globalstats.php>
- [21] <http://blogs.msdn.com/ie/archive/2006/08/22/712830.aspx>
- [22] M Holzschlag, *250 HTML and Web Design Secrets*, Wiley Publishing, Inc., Indianapolis, Indiana, 2004
- [23] <http://www.ampsoft.net/webdesign-l/windows-fonts-by-version.html>
- [24] <http://support.apple.com/kb/HT1538>
- [25] <http://www.ampsoft.net/webdesign-l/WindowsMacFonts.html>
- [26] <http://www.w3.org/TR/CSS2/fonts.html>
- [27] <http://www.mikeindustries.com/blog/sifr/>
- [28] http://www.adobe.com/products/player_census/flashplayer/
- [29] <http://www.smashingmagazine.com/2007/07/27/css-specificity-things-you-should-know/>
- [30] <http://www.alistapart.com/stories/pngopacity/>
- [31] <http://support.microsoft.com/kb/294714>

- [32] <http://reference.sitepoint.com/css/filter>
- [33] <http://labs.unitinteractive.com/unitpngfix.php>
- [34] <http://hsivonen.iki.fi/png-gamma/>
- [35] A Campbell, Access all areas, *Practical web design*, 197, (2008), 1/1, str. 105.
- [36] <http://googleblog.blogspot.com/2008/06/google-learns-to-crawl-flash.html>
- [37] <http://www.useit.com/alertbox/20001029.html>
- [38] <http://www.campaignmonitor.com/css/>
- [39] <http://www.sitepoint.com/print/principles-beautiful-html-email/>
- [40] http://www.iab.net/iab_products_and_industry_services/1421/1443/1452
- [41] <http://www.vest.si/oglasevanje/splosni-pogoji/>
- [42] <http://www.rastko.cc/?m=200803>
- [43] http://www.businessweek.com/innovate/next/archives/2007/10/nikes_new_publi.html
- [44] <http://www.webaim.org/techniques/ajax/>
- [45] <http://en.wikipedia.org/wiki/AJAX>
- [46] <http://www.webaim.org/techniques/ajax/>
- [47] <http://www.web2inspiration.com/effective-web-20-font-colors/>
- [48] http://en.wikipedia.org/wiki/Web_3.0
- [49] D Giustini, Web 3.0 and Medicine, *British Medical Journal*, (2007)
- [50] J Markoff, Entrepreneurs See a Web Guided by Common Sense, *New York Times*, (2006)
- [51] T Berners-Lee. J Hendler. O Lassila, The Semantic Web, *Scientific American*, 2001
(www.si.umich.edu/~rfrost/courses/SII10/readings/In_Out_and_Beyond/Semantic_Web.pdf)
- [52] <http://microformats.org/>
- [53] <http://labs.mozilla.com/2008/08/introducing-ubiquity/>
- [54] <http://blogs.zdnet.com/BTL/?p=4499>
- [55] P McNeil, Switch on to the latest styles, *Practical web design*, 197, (2008), 1/1, str. 42 - 47.

PRILOGE

Statistični podatki

Zbrane statistike razširjenosti zaslonskih ločljivosti, spletnih brskalnikov in barvnih globin zaslonov za slovenski prostor.

Ločljivosti SI	1. spletno mesto	2. spletno mesto	3. spletno mesto	
1024x768	50,63%	37,49%	37,89%	42,00%
1280x1024	20,41%	30,63%	30,36%	27,13%
1280x800	11,20%	8,45%	5,24%	8,30%
1680x1050	4,26%	5,79%	5,12%	5,06%
1440x900	3,67%	4,41%	5,12%	4,40%
1152x864	2,47%	4,19%	3,56%	3,41%
800x600	1,92%	3,38%	2,11%	2,47%
1280x960	0,91%	1,67%	1,36%	1,31%
1600x1200	0,97%	1,20%	1,06%	1,08%
1400x1050	1,05%	0,81%	1,09%	0,98%

Brskalniki SI	1. spletno mesto	2. spletno mesto	3. spletno mesto	
IE 6	45,10%	36,57%	34,90%	38,86%
Firefox	26,97%	34,40%	37,23%	32,87%
IE 7	24,48%	27,51%	25,10%	25,70%
Safari	1,95%	0,43%	1,36%	1,25%
Opera	0,98%	0,80%	0,82%	0,87%
Mozilla	0,26%	0,10%	0,10%	0,15%

Barvna globina SI	1. spletno mesto	2. spletno mesto	3. spletno mesto	
32 bit	80,19%	94,11%	92,66%	88,99%
16bit	16,48%	4,46%	4,93%	8,62%
24bit	3,27%	1,41%	2,34%	2,34%
8bit	0,05%	0,01%	0,07%	0,04%

Prvo spletno mesto

Barvna globina zaslonov

1. 32-bit 44,996 80.19%
2. 16-bit 9,245 16.48%
3. 24-bit 1,837 3.27%
4. 8-bit 29 0.05%
5. 12-bit 5 0.01%
6. 0-bit 2 > 0.00%
7. 1-bit 1 > 0.00%

Zaslonske ločljivosti

1. 1024x768 28,412 50.63%
2. 1280x1024 11,454 20.41%
3. 1280x800 6,283 11.20%
4. 1680x1050 2,390 4.26%
5. 1440x900 2,061 3.67%
6. 1152x864 1,384 2.47%
7. 800x600 1,075 1.92%
8. 1400x1050 590 1.05%
9. 1600x1200 543 0.97%
10. 1280x960 509 0.91%
11. 1920x1200 474 0.84%
12. 1280x768 345 0.61%
13. 1280x720 87 0.16%
14. 1280x854 49 0.09%
15. 800x480 30 0.05%
16. 1366x768 27 0.05%
17. 1360x768 26 0.05%
18. 240x320 26 0.05%
19. 320x396 26 0.05%
20. 1024x600 21 0.04%
21. 1600x1024 21 0.04%

Spletni brskalniki

1. Internet Explorer 39,122 69.72%
2. Firefox 15,136 26.97%
3. Safari 1,095 1.95%
4. Opera 550 0.98%
5. Mozilla 147 0.26%
6. Netscape 26 0.05%
7. Konqueror 13 0.02%
8. Camino 6 0.01%
9. Mozilla Compatible Agent 4 0.01%
10. HTC_Hermes Mozilla 3 0.01%
11. HTC_TyTN_II Mozilla 3 0.01%
12. HTC_TouchDual Mozilla 2 > 0.00%
13. NetFront 2 > 0.00%
14. Playstation 3 2 > 0.00%
15. (not set) 1 > 0.00%
16. BSHSHKIQF 1 > 0.00%
17. HTC_P3700 Opera 1 > 0.00%
18. PPC; 240x320; HTC_P3450 1 > 0.00%

IE

1. 6.0 25,307 64.69%
2. 7.0 13,740 35.12%
3. 5.5 54 0.14%
4. 8.0 14 0.04%
5. 5.01 6 0.02%
6. 5.0 1 > 0.00%

FF

1. 2.0.0.16 8,088 53.44%
2. 3.0.1 4,886 32.28%
3. 3.0 350 2.31%
4. 2.0.0.14 341 2.25%
5. 1.5.0.12 160 1.06%
6. 2.0.0.11 146 0.96%
7. 2.0 129 0.85%
8. 2.0.0.12 109 0.72%
9. 1.0.7 104 0.69%
10. 2.0.0.15 98 0.65%

11. 2.0.0.13 91 0.60%
12. 2.0.0.6 75 0.50%
13. 2.0.0.9 65 0.43%
14. 1.0.6 50 0.33%
15. 2.0.0.7 49 0.32%
16. 1.0 41 0.27%
17. 2.0.0.4 41 0.27%
18. 2.0.0.8 41 0.27%
19. 2.0.0.3 36 0.24%
20. 2.0.0.1 35 0.23%
21. 1.5.0.6 20 0.13%
22. 1.0.4 18 0.12%
23. 1.5.0.4 17 0.11%
24. 1.5.0.7 15 0.10%
25. 1.5.0.3 14 0.09%
26. 2.0.0.2 13 0.09%
27. 1.5.0.9 12 0.08%
28. 1.5.0.1 11 0.07%
29. 1.5.0.5 11 0.07%
30. 2.0.0.5 11 0.07%

Drugo spletno mesto

Barvna globina zaslonov

1. 32-bit 98,303 94.11%
2. 16-bit 4,654 4.46%
3. 24-bit 1,474 1.41%
4. 8-bit 12 0.01%
5. 0-bit 7 0.01%

Zaslonske ločljivosti

1. 1024x768 39,158 37.49%
2. 1280x1024 31,995 30.63%
3. 1280x800 8,822 8.45%
4. 1680x1050 6,045 5.79%
5. 1440x900 4,604 4.41%
6. 1152x864 4,376 4.19%
7. 800x600 3,528 3.38%
8. 1280x960 1,743 1.67%
9. 1600x1200 1,255 1.20%
10. 1400x1050 851 0.81%
11. 1920x1200 844 0.81%
12. 1280x768 489 0.47%
13. 1280x720 112 0.11%
14. 1366x768 81 0.08%
15. 240x320 67 0.06%
16. 1360x768 44 0.04%
17. 1360x1024 39 0.04%
18. 2048x768 37 0.04%
19. 2560x1024 27 0.03%
20. 1120x840 25 0.02%
21. 960x600 24 0.02%
22. 800x480 19 0.02%
23. 1600x900 18 0.02%
24. 1280x975 17 0.02%
25. 1024x1280 16 0.02%

Spletni brskalniki

1. Internet Explorer 67,057 64.20%

2. Firefox 35,929 34.40%
3. Opera 832 0.80%
4. Safari 446 0.43%
5. Mozilla 103 0.10%
6. Netscape 52 0.05%
7. NetFront 20 0.02%
8. HTC_TyTN_II Mozilla 2 0.00%
9. Konqueror 2 0.00%
10. Mozilla Compatible Agent 2 0.00%

IE

1. 6.0 38,201 56.97%
2. 7.0 28,740 42.86%
3. 5.5 91 0.14%
4. 8.0 23 0.03%
5. 5.01 2 > 0.00%

FF

1. 2.0.0.16 17,917 49.87%
2. 3.0.1 13,306 37.03%
3. 3.0 789 2.20%
4. 1.0.7 531 1.48%
5. 1.5.0.12 497 1.38%
6. 2.0.0.14 409 1.14%
7. 2.0.0.13 318 0.89%
8. 1.0 304 0.85%
9. 2.0.0.11 275 0.77%
10. 2.0 262 0.73%

Tretje spletno mesto

Barvna globina zaslonov

1. 32-bit 15,389 92.66%
2. 16-bit 818 4.93%
3. 24-bit 388 2.34%
4. 8-bit 12 0.07%
5. 0-bit 1 0.01%

Zaslonske ločljivosti

1. 1024x768 6,292 37.89%
2. 1280x1024 5,043 30.36%
3. 1280x800 1,647 9.92%
4. 1680x1050 870 5.24%
5. 1440x900 851 5.12%
6. 1152x864 591 3.56%
7. 800x600 351 2.11%
8. 1280x960 226 1.36%
9. 1400x1050 181 1.09%
10. 1600x1200 176 1.06%
11. 1920x1200 141 0.85%
12. 1280x768 82 0.49%
13. 1280x720 33 0.20%
14. 1360x1024 19 0.11%
15. 240x320 9 0.05%
16. 1360x768 8 0.05%
17. 1200x1600 7 0.04%
18. 320x396 7 0.04%
19. 840x525 7 0.04%
20. 1024x763 5 0.03%
21. 1272x1015 5 0.03%
22. 1344x840 4 0.02%

23. 2560x1600 4 0.02%
24. 1120x840 3 0.02%
25. 1440x960 3 0.02%

Spletni brskalniki

1. Internet Explorer 3,453 60.36%
2. Firefox 2,130 37.23%
3. Safari 78 1.36%
4. Opera 47 0.82%
5. Mozilla 6 0.10%
6. Netscape 6 0.10%
7. SAMSUNG-SGH-i780 1 0.02%

IE

1. 6.0 1,996 57.80%
2. 7.0 1,433 41.50%
3. 5.5 13 0.38%
4. 5.0 6 0.17%
5. 5.01 4 0.12%
6. 8.0 1 0.03%

FF

1. 3.0.1 3,032 54.63%
2. 2.0.0.16 1,168 21.05%
3. 3.0.2 309 5.57%
4. 3.0.3 229 4.13%
5. 2.0.0.17 180 3.24%
6. 2.0.0.14 71 1.28%
7. 1.5.0.12 70 1.26%
8. 1.0.7 61 1.10%
9. 2.0 58 1.05%
10. 3.0 50 0.90%