

Povijesni razvoj CSS-a

Škalec, Domagoj

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, University of Zagreb, Faculty of Humanities and Social Sciences / Sveučilište u Zagrebu, Filozofski fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:131:573456>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-06-22**



Sveučilište u Zagrebu
Filozofski fakultet
University of Zagreb
Faculty of Humanities
and Social Sciences

Repository / Repozitorij:

[ODRAZ - open repository of the University of Zagreb
Faculty of Humanities and Social Sciences](#)



SVEUČILIŠTE U ZAGREBU
FILOZOFSKI FAKULTET
ODSJEK ZA INFORMACIJSKE I KOMUNIKACIJSKE ZNANOSTI
Ak. god. 2019./ 2020.

Domagoj Škalec

POVIJESNI RAZVOJ CSS-A

Završni rad

Mentor: dr.sc. Kristina Kocijan, izv.prof.

Zagreb 2020.

Izjava o akademskoj čestitosti

Izjavljujem i svojim potpisom potvrđujem da je ovaj rad rezultat mog vlastitog rada koji se temelji na istraživanjima te objavljenoj i citiranoj literaturi. Izjavljujem da nijedan dio rada nije napisan na nedozvoljen način, odnosno da je prepisan iz necitiranog rada, te da nijedan dio rada ne krši bilo čija autorska prava. Također izjavljujem da nijedan dio rada nije korišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

(potpis)

Sadržaj

Izjava o akademskoj čestitosti	1
Sadržaj.....	2
Uvod.....	3
1. Osnove CSS-a	4
2. Povijest CSS-a.....	6
2.1. Statusi dokumenata	10
2.2. CSS 1	12
2.3. CSS 2.....	13
2.3.1. Nove funkcionalnosti	13
2.3.2. Ažurirani opisi	16
2.3.3. Semantičke promjene iz CSS-a 1.....	16
2.4. CSS 2.1	17
2.5. CSS 3	18
3. Konkurencija CSS-u.....	20
3.1. DSSSL	20
3.2. XSL	21
4. Principi dizajna u CSS-u.....	23
5. Zaključak.....	26
6. Literatura.....	28
Sažetak.....	30

Uvod

Ideja za stilski jezik CSS (engl. *Cascading Style Sheets*, približan prijevod na hrvatski bio bi *stranice s opisom kaskadnih stilova*) nastala je 1994. godine. CSS je od onda više puta mijenjan, a 1996. godine je dovršena prva njegova iteracija. CSS se koristi za definiranje formatiranja na nekoj mrežnoj stranici (jednostavnije rečeno, njen izgled).

CSS je jedna od ključnih tehnologija u kreiranju suvremenog sadržaja na Internetu. Dopušta prilagodbu mrežne stranice ekranima uređaja različitih veličina – stolna te prijenosna računala, mobilni telefoni, tableti itd. Najčešće se koristi zajedno s HTML-om (engl. *HyperText Markup Language*), jezikom za označavanje (engl. *markup language*) i JavaScriptom, programskim jezikom koji mrežnoj stranici dodaje funkcionalnost. CSS je vrlo kompatibilan i intuitivan za korištenje s HTML-om jer implicitno sadrži neke HTML elemente (van Ossenbruggen et al., 2002). Iako postoje ostali (npr. DSSSL i XSL), CSS je najrašireniji stilski jezik. CSS omogućava programerima da uz promjenu samo nekoliko linija kôda promijene grafički dizajn cijelog dokumenta ili stranice – primjerice, promjena veličine margina stranice izmjenom samo jednog broja u vrijednosti svojstva margina ili pozadinsku boju cijele stranice promjenom vrijednosti svojstva boja.

U prvom poglavlju ovog rada će ukratko biti opisane osnove korištenja CSS-a u svrhu lakšeg rukovanja terminologijom i jezikom koji će se koristiti kasnije u radu.

Nakon tog opisa korištenja CSS-a, svi čitatelji bi trebali imati pozadinu za čitanje drugog, glavnog, poglavlja, gdje će se pokriti povijesni razvoj tog stilskog jezika, od CSS-a 1, preko CSS-a 2 i 2.1 do konačnog CSS-a 3, koji se koristi danas.

Zadnja dva poglavlja pričaju o ostalim stilskim jezicima koji čine konkurenciju CSS-u te o principima dizajna kojih se sve razine CSS-a pridržavaju u što je većoj mjeri moguće.

1. Osnove CSS-a

Najbolje bi bilo prvo objasniti osnove korištenja CSS-a i terminologiju jezika prije daljnjeg istraživanja i korištenja raznih elemenata CSS-a, kako bi se olakšao prolazak kroz rad čitateljima koji nisu prije koristili CSS. Svako *pravilo* (engl. *rule*) u *stranici s opisom stilova* se sastoji od dva glavna dijela: **selektora** (engl. *selector*) i **deklaracijskog bloka** (engl. *declaration block*). Primjer selektora bi bio *body* element, dok bi primjer deklaracijskog bloka bio:

```
{background-color: lightblue;}
```

Selektor je opisan kao poveznica „između HTML dokumenta i stranice s opisom stilova, a svi tipovi HTML elemenata su mogući selektori“¹ (Wium Lie & Bos, 1996, poglavlje 1, para. 3). U toj definiciji vidimo u kolikoj je mjeri CSS napravljen s HTML-om na umu, jer su autori CSS-a eksplicitno spomenuli HTML, a ne neki drugi jezik za označavanje.

Deklaracijski blok u CSS-u se sastoji od jedne ili više deklaracija, koje se sastoje od svojstava i vrijednosti tih svojstava. U gornjem primjeru bi *background-color* bilo svojstvo, a *lightblue* vrijednost tog svojstva. Svojstva se od svojih vrijednosti odvajaju dvotočkom (simbol :), dok se više parova svojstava-vrijednosti u deklaracijskom bloku međusobno razdvaja točkom sa zarezom (simbol ;). Točka sa zarezom je obavezna u deklaracijskom bloku čak i kad se on sastoji od samo jedne deklaracije. Deklaracijski blok se uvijek nalazi unutar vitičastih zagrada.

Ta osnovna sintaksa CSS-a je ostala nepromijenjena do danas. Jedini noviteti koje programeri u CSS-u moraju savladati su nova svojstva koja se dodaju, njihovi atributi i slični elementi. CSS 1 je imao 53² različita svojstva (Wium Lie & Bos, 1996), dok ih je CSS 2 imao 98 (Bos et al., 2011).

¹ Ovo je slobodni prijevod autora koji u originalu na engleskom jeziku glasi: “*The selector is the link between the HTML document and the style sheet, and all HTML element types are possible selectors.*” (Wium Lie & Bos, 1996, poglavlje *Basic concepts*, para. 3).

² Prebrojano iz tablice sadržaja.

Korištenje CSS-a nije obavezno pri kreiranju jednostavnih mrežnih stranica. Stranice bez dodatnog CSS-a neće sadržavati samo običan tekst bez ikakvog formatiranja jer mrežni preglednici imaju vlastite zadane postavke o tome kako se nositi s nekim HTML elementima.

No unatoč tomu što bez definiranja stranica s opisom stilova svejedno postoji nekakvo formatiranje u dokumentu, bilo kakvo složenije vizualno formatiranje mrežnih stranica zahtijeva korištenje CSS-a kako stranica ne bi izgledala previše jednostavno te kako bi lakše i uspješnije privukla potencijalne korisnike.

Jako dobar primjer korištenja CSS-a, pogotovo u njegovim ranijim danima, je internetski preglednik za mobilne telefone *Opera Mini*. Prije nego što su mobilni uređaji postali rašireni kao što su danas, rijetko koja mrežna stranica je bila napravljena s prikazom na mobilnim uređajima u planu. Vrijedno je spomena da su mobilni uređaji tada imali puno manje zaslone nego danas. Razvojni programeri preglednika *Opera Mini* riješili su taj problem. Na uređajima širine zaslona manje od 146 piksela cijelu mrežnu stranicu prikazuju u jednom dugačkom vertikalnom stupcu, bez potrebe za horizontalnim listanjem stranice (Opera, 2007). CSS 2 je uveo tehnologiju koja je omogućila takve inovacije.

2. Povijest CSS-a

CSS je započeo svoj životni vijek 1994. godine kao ideja CERN-ovog zaposlenika Håkona Wiuma Lieja u njegovoj prvoj skici objavljenoj 10. listopada (Wium Lie, 1994). Svjetska mreža (engl. *World Wide Web*) se u to vrijeme polako razvijala kao platforma za elektronsko izdavanje, no nedostajao je način za korištenje rasporeda sličnog stranicama novina na mrežnim mjestima. Autori mrežnih stranica u svojim dokumentima nisu mogli niti mijenjati boju teksta, tako da su imali jako malo utjecaja na to kako će njihova stranica izgledati (Bos, 2016). Stranicu su stilistički mogli urediti samo dodavanjem HTML oznaka poput `<hr>` oznake koja djeluje kao pregrada između tema na mrežnoj stranici.

Netko tko nije koristio CSS, nego samo HTML, bi mogao pomisliti da je CSS nepotreban jer neke stvari koje se mogu mijenjati u CSS-u mogu jednako tako promijeniti koristeći samo HTML, primjerice porodicu fonta oznakom ``. Ako je trenutni trend u mrežnom dizajnu korištenje sans-serifnih fontova poput Ariala, no prema zadanim postavkama većina internetskih pretraživača koristi serifne fontove poput *Times New Romana*, to može dovesti do nepotrebno velikog broja `` oznaka na mrežnoj stranici kako bi se svi fontovi izjednačili. To je zato što bi svaki paragraf gdje postoji neka razlika u fontu (npr. veličina) zahtijevao novu `` oznaku (Andrew, 2004).

CSS je osmišljen kako bi to promijenio. Wium Lie predstavlja CSS kao poveznicu između elemenata HTML-a i naputaka za prezentaciju internetskog sadržaja na različitim uređajima. Opisao je mogućnosti CSS-a u svom prijedlogu na sljedeći način: „Svojstva poput porodice fonta i veličine prozora mogu biti predložene pomoću stranica s opisom stilova, koji također može pružiti logiku za odluke prezentiranja sadržaja s obzirom na korisnikovu okolinu; npr. veličinu ekrana ili trenutni datum“³ (Wium Lie, 1994: Abstract, para. 1).

³ Ovo je slobodni prijevod autora koji u originalu na engleskom jeziku glasi: “*Properties like font family and window size can be suggested by the style sheet, and it can also provide logic to make presentation decisions based on the user's environment; e.g. the size of the screen or the current date.*” (Wium Lie, 1994, Abstract, para. 1).

Bert Bos je bio među prvima koji je odgovorio na Wium Liejov prijedlog CSS-a. CSS je tada još bio zvan **HTML** stranice s opisom kaskadnih stilova (engl. *Cascading HTML Style Sheets*), što je kasnije promijenjeno u *Stranice s opisom kaskadnih stilova* jer je neovisan od HTML-a te se može koristiti s bilo kojim jezikom za označavanje baziranom na XML-u (W3C, 2016). Bos je u to vrijeme radio na pretraživaču *Argo* koji je koristio stranice s opisom stilova te se stoga odlučio udružiti s Wium Liejom. Stilski jezik pretraživača *Argo* je bio dovoljno općenit da bi se mogao koristiti i s drugim jezicima za označavanje uz HTML. Wium Lie je objavio prvu verziju svog prijedloga tri dana prije najave *Netscapeovog* internetskog preglednika, a sam prijedlog je javno prezentiran na mrežnoj konferenciji *Mosaic and the Web* u studenom 1994. u Chicagu. Novost u svijetu mrežnog dizajna je bila CSS-ova ravnoteža između želja korisnika i želja autora⁴. Dio publike je smatrao da je CSS prejednostavan za zadatak koji mu je namijenjen te da je za to potrebna snaga punog jezika za programiranje (Bos, 2016). No upravo je ta jednostavnost postala CSS-ova najveća snaga.

Najviše nesuglasica među programerima je bilo oko toga koliko utjecaja na izgled stranice treba imati autor, a koliko korisnik. Wium Lie je u svom prijedlogu CSS-a rekao: „(...) ako dođe do sukoba, korisnikova riječ bi trebala biti zadnja, no također bi se trebalo dopustiti autorima da dodaju naputke za stiliziranje“⁵ (Wium Lie, 1994, poglavlje *Introduction*, para. 1). Rekao je da je to najveći problem u poljima koja se inače bave objavljivanjem u papirnatom formatu, kao što su novine, jer tamo čitatelji uopće nemaju izbora oko izgleda sadržaja kojeg čitaju (Wium Lie, 1994). Bos i Wium Lie su bili strogi pobornici toga da korisnik mora biti glavni u odlučivanju stiliziranja stranice, pošto korisnikove oči i uši moraju dekodirati sadržaj mrežne stranice. Oni koji se s njima nisu slagali su tvrdili da bi autor morao biti taj koji odlučuje kako stilizirati stranicu. Primjer koji

⁴ Korisnici su krajnji čitatelji na Internetu, a autori su oni koji su napravili stranicu. Tada je još bilo uobičajeno da korisnici koriste vlastite stranice s opisom stilova.

⁵ Ovo je slobodni prijevod autora koji u originalu na engleskom jeziku glasi: „(...) *if conflicts arise the user should have the last word, but one should also allow the author to attach style hints*” (Wium Lie, 1994, poglavlje *Introduction*, para. 1).

su dali je bio taj da korisnik ne bi smio moći smanjiti veličinu fonta na oznakama za upozorenje (Bos, 2016).

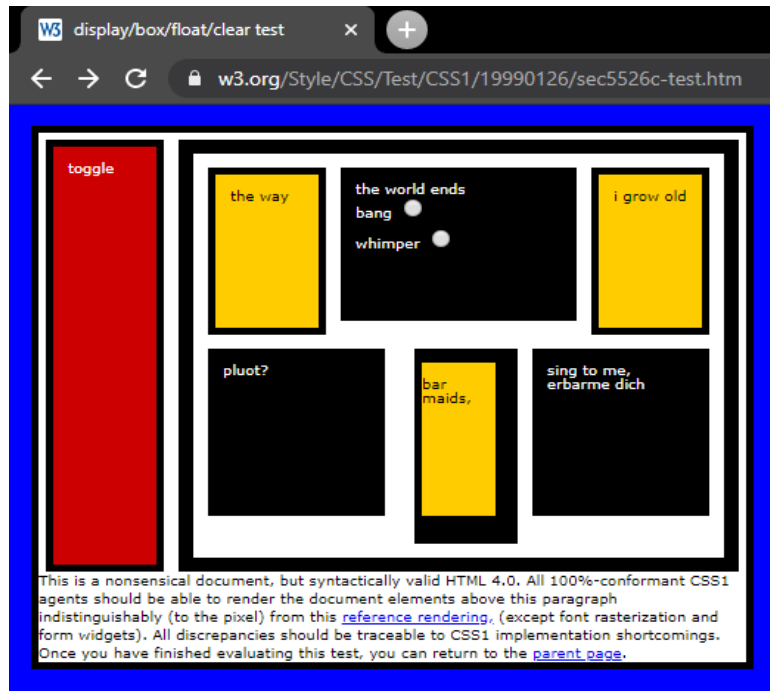
1995. je *Konzorcij Svjetske mreže* (engl. *World Wide Web Consortium*, odnosno *W3C*, što je također najčešći naziv za organizaciju) postao aktivan (Bos, 2016). Konzorcij je 1994. osnovao izumitelj Svjetske mreže Tim Berners-Lee na Institutu za tehnologiju u Massachusettsu (W3C, 2020). Organizaciji su se velikom brzinom priključivale razne tvrtke i organizacija se postavila kao vodeća u poboljšavanju Interneta i inovacijama u tom polju. W3C je odlučio održati radionicu o stilskim stranicama. Radionici je prisustvovao i Thomas Reardon, Microsoftov zaposlenik koji je obećao potporu za CSS u nadolazećim verzijama *Internet Explorera*, *Microsoftovog* Internet preglednika. W3C-u se također pridružio i *Netscapeov* Lou Montulli, što je vrlo važno jer bi se bez *Netscapeovog* sudjelovanja u W3C-u Internet razvijao u različitim smjerovima za različite preglednike. Bilo je mnogo prepirki unutar W3C-ovog Odbora urednika za HTML (engl. *HTML Editorial Review Board*, odnosno *HTML ERB*) koji se, među ostalim, bavio i specifikacijama CSS-a. No te prepirke su nakon nekog vremena razriješene te je prva razina CSS-a dobila status preporuke W3C-a u prosincu 1996. (Bos, 2016).

U veljači sljedeće godine, CSS je dobio radnu grupu unutar W3C-a koja se odmah fokusirala na zadatke koji nisu bili pokriveni u CSS-u 1. Na čelu grupe je bio Chris Lilley. CSS 2 je postao preporuka W3C-a u svibnju 1998. godine, te je grupa od tada radila paralelno na novim modulima za CSS 1 i popravljala greške na CSS-u 2 (Bos, 2016).

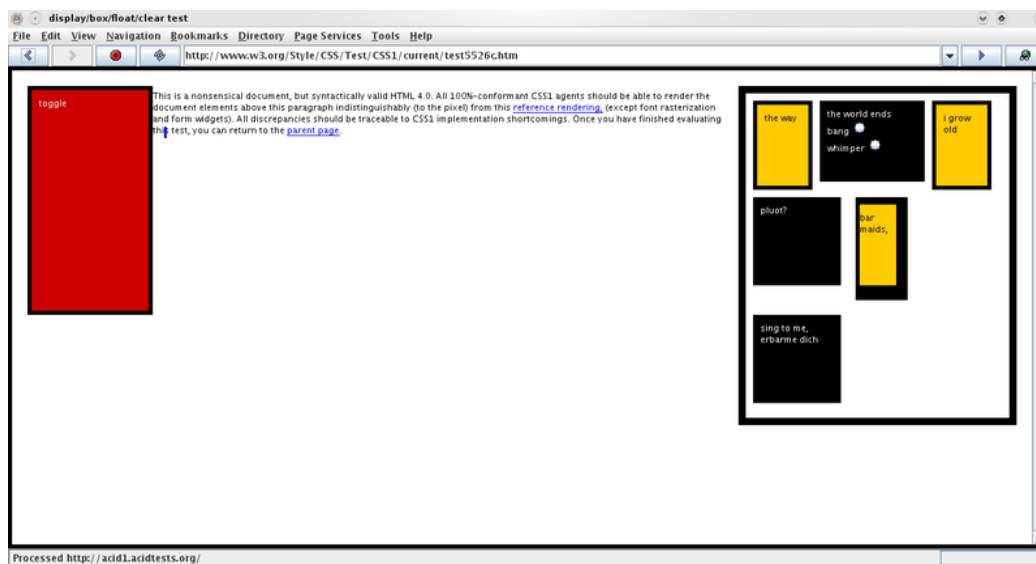
Postojanje obje verzije CSS-a u isto vrijeme je značilo da se mogu uspoređivati, no to je razotkrilo problem: ako je stranica s opisom stilova bila testirana samo u jednom pregledniku, ne može se od njega očekivati da funkcionira i u ostalim preglednicima. Rješenje na taj problem je osmislio Eric Meyer, koji je razvio testni program u kojem bi programeri tražili probleme i greške (engl. *bugs*) dok još imaju vremena popraviti ih (Bos, 2016).

U listopadu 1998. Todd Fahrner je osmislio *Acid1* test (Bos, 2016) koji je služio za ispitivanje sposobnosti internetskih preglednika, da se vidi je li kompatibilan s CSS-om 1.

Možemo vidjeti kako uspješno izveden *Acid1* test izgleda na slici 1, a na slici 2 vidimo *Acid1* test u pregledniku koji ne podržava CSS 1 u potpunosti. Do 2008. godine su svi popularniji preglednici prošli *Acid1* test (Sciactive, 2008).



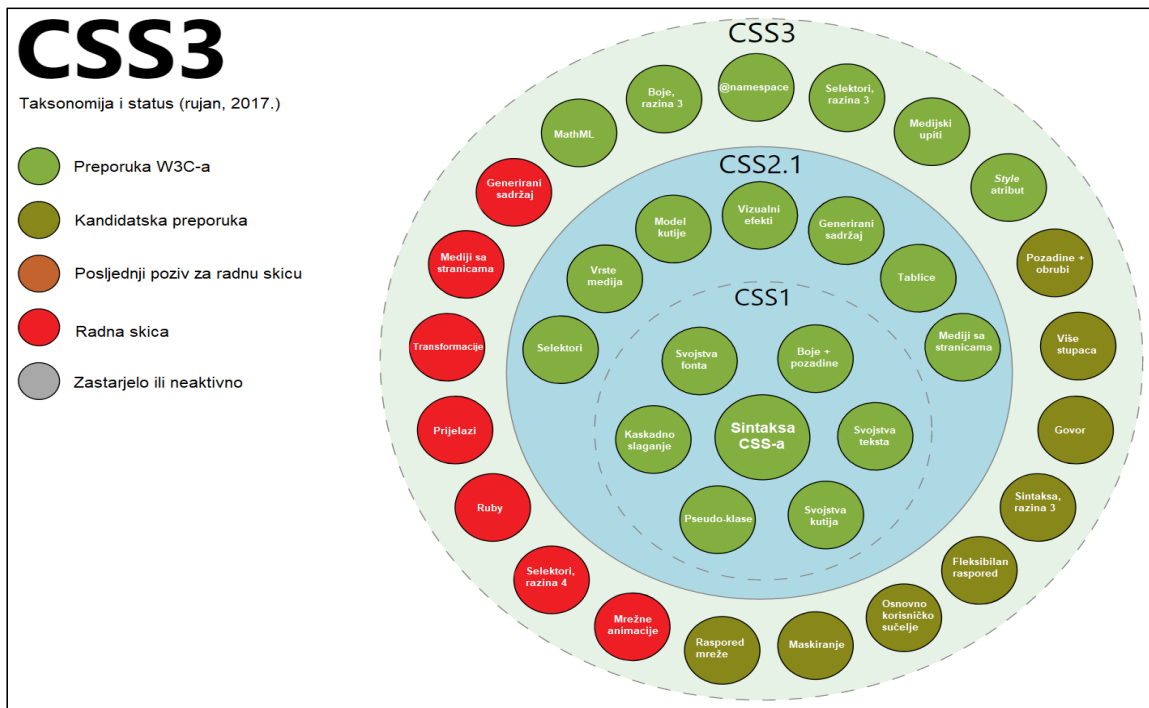
Slika 1: Prikaz stranice nakon što je "Acid1" test uspješno izveden na pregledniku *Google Chrome*, verzija 85.0.4183.83. (vlastiti rad)



Slika 2: Prikaz stranice na kojoj *Acid1* test nije uspješno izveden u pregledniku *Lobo* na operativnom sustavu *Linux*. Nedostaje plava pozadina, crvena kutija nije na pravom mjestu, kao ni tekst te posljednja crna kutija. (izvor: The Lobo Project, 2009, <https://commons.wikimedia.org/wiki/File:Lobo-0.98.4-linux.png>)

2.1. Statusi dokumenata

CSS se često mijenjao kroz godine te su mu dodavane mnoge značajke.



Slika 3: Graf koji prikazuje razlike između različitih specifikacija CSS-a (točnije CSS 1, CSS 2.1, CSS 3) u rujnu 2017. godine. Graf je prilagođen prema radu Sergeya Mavrodyja, rujna, 2017. (https://commons.wikimedia.org/wiki/File:CSS3_taxonomy_and_status_by_Sergey_Mavrody.svg)

Uz graf na slici 3 možemo vidjeti nazive različitih statusa neke značajke.⁶ Ovo su službeni nazivi za status dokumenata W3C-a:

- **radna skica (engl. working draft)** – „Radne skice se objavljuju tijekom procesa skiciranja specifikacije. Svrha je stvoriti sliku trenutnog stanja specifikacije da bi se zatražila povratna informacija od W3C-a i javnosti. Specifikacija se još uvijek smatra nestabilnom i često je nedovršena“⁷ (Bos, 2020, poglavlje *Document statuses*, para. 2).

⁶ Status *Last Call* nije službeno ime nekog W3C statusa.

⁷ Ovo je slobodni prijevod autora koji u originalu na engleskom jeziku glasi: “*WDs are published during the process of drafting the specification. The purpose is to create a snapshot of the specification's current state and to solicit input from the W3C and the public. The specification is still known to be unstable, and is often incomplete.*” (Bos, 2020, poglavlje *Document statuses*, para. 2).

Prva službena radna skica se zove *Prva javna radna skica* (engl. *First Public Working Draft*, odnosno *FPWD*). Objava *FPWD*-a znači da se CSS radna grupa složila da rade na nekoj značajki, otprilike onako kako je predložena u skici urednika. Prijelaz u sljedeću fazu se ponekad naziva *Posljednji poziv za radnu skicu* (engl. *Last Call Working Draft*, odnosno *LCWD*). Do nje specifikacija dolazi kad CSS radna grupa razriješi sve poznate probleme te više ne može dalje napredovati bez povratnih informacija. S tom fazom dolazi i rok za prijavu svih znatnijih problema te radna grupa mora s posebnom pozornošću pratiti nadolazeće povratne informacije (Atkins et al., 2019);

- **kandidatska preporuka (engl. *candidate recommendation*)** – „Objavom kandidatske preporuke radna grupa smatra specifikaciju dovršenom te da je razriješila sve poznate probleme. Specifikacija je onda spremna za testiranje u praksi. Status kandidatske preporuke je poziv za implementacije“⁸ (Bos, 2020, poglavlje *Document statuses*, para. 3). Proces testiranja često otkrije još problema sa specifikacijom, tako da će se kandidatska preporuka kroz vrijeme mijenjati sukladno s povratnim informacijama (Atkins et al., 2019);
- **ponuđena preporuka (engl. *proposed recommendation*)** – „Da bi iz statusa kandidatske preporuke došla do statusa ponuđene preporuke, specifikacija zahtijeva opsežni testni program i izvještaje o implementaciji koji dokazuju da je svaka značajka kompatibilna s ostalima u barem dvije proslijeđene implementacije. Članove W3C-a se traži da po posljednji put recenziraju specifikaciju“⁹ (Bos, 2020, poglavlje *Document statuses*, para. 4);

⁸ Ovo je slobodni prijevod autora koji u originalu na engleskom jeziku glasi: “By publishing a Candidate Recommendation, a working group is expressing that it considers the specification complete and that it has resolved all known issues. It is ready for testing in practice. A CR is a call for implementations.” (Bos, 2020, poglavlje *Document statuses*, para. 3).

⁹ Ovo je slobodni prijevod autora koji u originalu na engleskom jeziku glasi: “To exit CR and enter this stage, the specification needs a comprehensive test suite and implementation reports proving that every feature is interoperably implemented in at least two shipping implementations. W3C Members are asked to review the specification one last time.” (Bos, 2020, poglavlje *Document statuses*, para. 4).

- **preporuka (engl. *recommendation*)** – „Ovo je posljednja faza; standard. U ovom trenutku ne bi trebalo biti više promjena, no naravno da se može dogoditi da se pronađu neke greške zbog kojih bi se status preporuke trebao izmijeniti. Preporuka s vremenom također može postati zastarjela“¹⁰ (Bos, 2020, poglavlje *Document statuses*, para. 5).

CSS nema verzije u tradicionalnom smislu te riječi, nego umjesto njih ima razine. Svaka razina CSS-a se nadograđuje na prethodnu, razrađujući definicije i dodajući značajke. Set značajki od kojih se svaka viša verzija sastoji sadrži sve značajke nižih razina, a isto vrijedi i za ponašanje određenih značajki (Atkins et al., 2019).

Sljedećih nekoliko potpoglavlja će pobliže opisati razlike između razina te nove značajke koje su bile dodavane.

2.2. CSS 1

CSS je osmišljen da bude jednostavan i intuitivan za korištenje. To je u njegovim začecima bilo nešto što se stalno spominjalo kao mana, neki su ga smatrali "preslabim" za ono što treba postići (Bos, 2016). No baš zato je postao standardom za stiliziranje mrežnih stranica. Uzmimo za primjer sljedeći primjer CSS koda:

```
body {  
    background-color: lightblue;  
}
```

Čak i osoba koja nije tehnološki pismena, a zna engleski jezik, vidi o čemu se ovdje radi. Razumljivo je da taj kod ima veze s tekстом u glavnom tijelu teksta (a ne u zaglavlju, primjerice), te da će mu dati svijetlo-plavu pozadinu, čak i ako osoba ne razumije točnu sintaksu CSS-a (u ovom slučaju su to vitičaste zagrade, dvotočke i točke-zarezi).

¹⁰ Ovo je slobodni prijevod autora koji u originalu na engleskom jeziku glasi: “*This is the final stage: a standard. At this point there should be no more changes, although it may of course happen that errata are found that eventually require a REC to be revised. A REC may eventually also become obsolete.*” (Bos, 2020, poglavlje *Document statuses*, para. 5).

Čak su i Håkon Wium Lie i Bert Bos u W3C-ovoj preporuci CSS-a 1 rekli da je korištenje CSS-a lako te da za to treba poznavati samo malo HTML-a i osnovnu terminologiju digitalnog izdavaštva (Wium Lie & Bos, 1996).

Razina 1 CSS-a se definira kao sve značajke koje su definirane u specifikaciji CSS-a 1 (svojstva, vrijednosti, *at-pravila*¹¹), no koristeći sintaksu i definicije iz specifikacije CSS-a 2.1. Dokument *CSS Style Attributes* definira njegovo uključivanje u stilske atribute za specifične elemente (Bos, 2020). Prva specifikacija CSS-a je zastarjela jer su svi elementi iz te specifikacije nadograđeni ili zamijenjeni elementima iz kasnijih specifikacija.

2.3. CSS 2

Iako je specifikacija CSS 2 tehnički službena preporuka W3C-a, to je postala prije nego je W3C definirao status kandidatske preporuke. CSS 2 se nadovezuje na CSS 1 te je, uz vrlo rijetke iznimke, svaka stranica s opisom stilova koja je bila važeća u CSS-u 1, važeća u CSS-u 2 (Bos et al., 1998).

2.3.1. Nove funkcionalnosti

Uz sve stare mogućnosti koje je CSS 1 imao, CSS 2 podržava i sljedeće značajke (Bos et al., 1998):

- koncept vrsta medija (engl. *media types*):
 - dozvoljavaju da se stranica s opisom stilova uredi drukčije ovisno o uređaju na kojem se prikazuje.
 - medijski uređaji koje CSS 2 podržava, odnosno vrijednosti koje CSS prihvaća u *@media* upitu su (Bos et al., 1998):
 - *all* – prikladno za sve uređaje;

¹¹ CSS izjave koje upućuju CSS kako da se ponaša. Neki primjeri *at-pravila* bili bi *@charset*, *@import*, *@namespace* (Mozilla Developer Network, 2019).

- *aural* – koristi se u uređajima koji imaju sposobnost reproduciranja govora;
- *braille* – koristi se u uređajima koji imaju povratne informacije u Brailleovom pismu;
- *embossed* – koristi se u uređajima koji ispisuju papir Brailleovim pismom;
- *handheld* – koristi se u malim, prijenosnim uređajima manjih brzina mreže;
- *print* – koristi se za sadržaj podijeljen u više stranica koji se pregledava u načinu pretpregleda prije ispisa (engl. *print preview mode*);
- *projection* – koristi se za projektirane prezentacije;
- *screen* – primarno se koristi za računalne zaslone u boji;
- *tty* – koristi se u uređajima koji imaju mrežu od zadanog broja znakova (terminali, teleprinteri itd.);
- *tv* – koristi se u uređajima sličnim televizorima (niska razlučivost, boja, zvuk);

- Primjer *@media* upita bi bio:

```
@media screen and (min-width: 480px) {  
  body {  
    background-color: lightgreen;  
  }  
}
```

[ako se stranica prikazuje na ekranu minimalne širine od 480 piksela, pozadinska boja *body* elementa bit će *lightgreen*]

- vrijednost *inherit* za sva svojstva:
 - To znači da će element naslijediti vrijednost svojstva od svog „roditeljskog“ elementa (Bos et al., 1998);
- mediji na stranicama (engl. *paged media*):
 - dokument je podijeljen na jednu ili više diskretnih stranica (Bos et al., 1998);

- auralne stranice s opisom stilova:
 - pružaju stranice s opisom stilova za čitače ekrana za slijepe i slabovidne osobe (Bos et al., 1998);
- više međunarodnih značajki, uključujući stilove za brojive liste, potporu za dvosmjerni tekst i potporu za navodne znakove osjetljive na jezik;
- proširen mehanizam za biranje fontova, uključujući inteligentno poklapanje, sintezu te fontove za preuzimanje. Također, uveden je koncept sistemskih fontova te je dodano novo svojstvo, *font-size-adjust*;
- tablice, uključujući nove vrijednosti za *display* i *vertical-align* svojstva;
- relativno i apsolutno pozicioniranje, uključujući nepomično pozicioniranje:
 - relativno pozicioniranje predstavlja pozicioniranje u odnosu na svoju uobičajenu poziciju, dok apsolutno pozicioniranje predstavlja pozicioniranje samo u odnosu na neki njegov ne-statični „predak“ element (Bos et al., 1998);
- nove vrste kutija (engl. *box*) (zajedno sa *block* i *inline*): *compact* i *run-in*;
- sposobnost kontroliranja prelijevanja (engl. *overflow*), preklapanja (engl. *clipping*) i vidljivosti (engl. *visibility*);
- sposobnost određivanja minimalne i maksimalne širine i visine;
- prošireni mehanizam za selektore, uključujući „djecu“ selektore, obližnje selektore i atributne selektore;
- generirani sadržaj, brojače te automatsko numeriranje, kao i označivače (engl. *marker*);
- sjene teksta, kroz novo svojstvo *text-shadow*;
- nekoliko novih pseudo-klasa - *:first-child*, *:hover*, *:focus*, *:lang*;
- sistemske boje i fontove;
- pokazivače;
- dinamične obrube.

2.3.2. Ažurirani opisi

Specifikacija CSS-a 1 je bila kratka i sažeta. Specifikacija CSS-a 2 je mnogo opsežnija i puno lakše čitljiva. Većina dodatnog sadržaja opisuje nove funkcionalnosti, no opis značajki CSS-a 1 u specifikaciji CSS-a 2 je također proširen. Čak i na mjestima gdje su opisi izmijenjeni, ne znači da je došlo do promjene u sintaksi ili semantici (Bos et al., 1998).

2.3.3. Semantičke promjene iz CSS-a 1

Iako su većinom kompatibilni s CSS 1 stranicama s opisom stilova, CSS 1 stranice s opisom stilova u nekim slučajevima imaju drukčije značenje kad se koriste kao CSS 2 stranice s opisom stilova. Većina promjena koje su se razvile između CSS-a 1 i CSS-a 2 razvile su se zbog novog iskustva radne grupe, no također ima i ispravaka nekih pogrešaka (Bos et al., 1998):

- značenje *!important* oznake je promijenjeno. U CSS-u 1, *!important* oznaka autora je imala veći prioritet od *!important* oznake korisnika u njegovoj stranici s opisom stilova. U CSS-u 2 je suprotan slučaj;
- u CSS-u 2 se vrijednosti boja preklapaju s obzirom na raspon boja raspoloživ ekranu, umjesto standardnog RGB sustava, što je bio slučaj u CSS-u 1;
- u CSS-u 1 je postojala greška da se vrijednost nekoliko svojstava (primjerice, *padding*, odnosno razmak sadržaja od unutarnjih granica elementa u kojem se taj sadržaj nalazi) odnosila na širinu „roditeljskog“ elementa. Vrijednost se zapravo uvijek trebala odnositi na širinu elementa na blok razini, te se u specifikaciji CSS-a 2 to odražava izrazom *sadržajni blok* (engl. *containing block*);
- početna vrijednost *display* svojstva je *inline* u CSS-u 2, a ne *block* kao u CSS-u 1;
- u CSS-u 1 je svojstvo *clear* bilo primijenjeno na sve elemente, što je bilo pogrešno te se u CSS-u 2 primjenjuje samo na elemente na blok razini;
- u CSS-u 1 su se *:link*, *:visited* i *:active* selektori međusobno isključivali, dok u CSS-u 2 *:active* može postojati zajedno sa *:link* ili *:visited* selektorima;

- predloženi faktor omjera između obližnjih *font-size* indeksa u tablici veličina fontova je smanjen sa 1,5 na 1,2;
- izračunata vrijednost *font-size* svojstva se nasljeđuje umjesto prave vrijednosti;
- opis vrijednosti *inside* (svojstva *list-style-position*) u CSS-u 1 je dozvolilo interpretaciju stranice s opisom stilova gdje je utjecalo na lijevu marginu teksta umjesto na poziciju točke. Te interpretacije nema u CSS-u 2.

S vremenom su iskustvo implementiranja specifikacije i daljnji osvrti razotkrili mnoštvo problema s drugom specifikacijom CSS-a, pa je CSS radna grupa odlučila definirati *CSS Razinu 2, Ispravak 1* (engl. CSS Level 2 Revision 1), odnosno CSS 2.1 umjesto proširivanja već nezgrapne liste pogrešaka (Atkins et al., 2019).

2.4. CSS 2.1

CSS 2.1 potječe od CSS-a 2 te je bio namijenjen da ga zamijeni. Neki dijelovi CSS-a 2 su ostali nepromijenjeni, neki su doživjeli izmjene, a neki su bili uklonjeni. Neki od uklonjenih dijelova se koriste u razini 3 CSS-a (Bos et al., 2011).

Specifikacija CSS-a 2.1 je dobila broj manje verzije u nazivu zbog svoje sličnosti CSS-u 2 (odnosno, nije dobila broj veće verzije, kao što je to CSS 3) (Wium Lie & Bos, 2005). Pošto nema sjedinjene specifikacije CSS-a 3, CSS 2.1 je zajedno sa svojim ispravcima i proširenjima dovršenih modula službena preporuka W3C-a.

Iako su neke pogreške u specifikaciji CSS-a 2 bile relativno brzo popravljene, specifikacija u cijelosti je ostala nepromijenjena. CSS 2.1 je definiran kako bi se obratila pažnja na problem implementacije i interoperabilnosti CSS-a 2 s tada nadolazećim CSS-om 3. U tu svrhu CSS 2.1 čini sljedeće (Bos et al., 2011):

- održava kompatibilnost dijelova CSS-a 2 koji nisu općeprihvaćeni i implementirani;
- sadržava u sebi sve objavljene ispravke CSS-a 2;

- na mjestima gdje se implementacija uvelike razlikuje od onoga kako je specificirana u CSS-u 2, izmjenjuje se kako bi bila u skladu s općeprihvaćenom praksom;
- uklanja značajke CSS-a 2 koje je CSS zajednica odbila tako što ih nije implementirala. Cilj CSS-a 2.1 je odraziti značajke CSS-a koje su u razumnoj mjeri šire implementirane za HTML te XML jezike općenito (a ne samo za neki određeni XML jezik, ili samo za HTML);
- uklanja značajke CSS-a 2 koje će, objavom CSS razine 3, postati zastarjele. Tako potiče pojačano korištenje značajki CSS-a 3 umjesto njih;
- dodaje (vrlo mali) broj novih vrijednosti svojstava tamo gdje je iskustvo implementacije pokazalo da su potrebna. Nove vrijednosti su:
 - *color: orange*
 - *display: inline-block*
 - *content: none* i *content: normal* (*none* i *normal* imaju jednako značenje u CSS-u 2.1)
 - *white-space: pre-wrap* i *white-space: pre-line*
 - *cursor: progress*.

Stranice s opisom stilova u CSS-u 2 nisu nužno uvijek kompatibilni unaprijed sa stranicama s opisom stilova u CSS-u 2.1. Iako prekidanje kompatibilnosti unaprijed nije baš poželjno, CSS radna grupa je vjerovala da su prednosti ispravaka CSS-a 2.1 vrijedne toga (Bos et al., 2011).

2.5. CSS 3

CSS radna grupa je umjesto definiranja jedne ogromne specifikacije odlučila primijeniti pristup u modulima, gdje svaki modul definira dio CSS-a. Moduli su CSS datoteke koje po zadanim postavkama obuhvaćaju sva imena klasa i animacija. Takav pristup razdvaja specifikaciju u lakše probavljive dijelove i pruža brže, djelomične nadogradnje CSS-u

(Atkins et al., 2019). CSS 3 je donio neke dugo-očekivane novosti, poput zaobljenih rubova, sjena, gradijenata, prijelaza ili animacija. Također su s njim došli i novi rasporedi, poput rasporeda u više stupaca (engl. *multi-column layout*) i rasporeda fleksibilne kutije (engl. *flexible box module*, odnosno *flexbox*). Raspored u više stupaca proširuje raspored u bloku kako bi olakšao čitanje dugih redova (Mozilla Developer Network, 2020).

Neki moduli su dosegli status W3C preporuke, no postoje i oni koji su još u ranom statusu radne skice. Moduli CSS-a 3 neće biti proturječni specifikaciji CSS-a 2.1, već će samo dodavati funkcionalnost i razrađivati definicije. Takav individualan pristup znači da razina 4 CSS-a nikada neće biti stvorena, unatoč tome što individualni moduli mogu prijeći i do viših razina (Atkins et al., 2019). Na primjer, u ožujku 2020. godine je objavljena prva javna radna skica pete razine modula za boje. U tom su modulu dodane tri nove funkcije za upravljanje bojama: *color-mix*, *color-contrast* i *color-adjust*. (Lilley, 2020).

3. Konkurencija CSS-u

CSS, naravno, nije jedini stilski jezik danas, kao što nije bio niti 90-ih godina 20. stoljeća, iako je postao najrasprostranjeniji. *WorldWideWeb* pretraživač Tima Bernersa-Leeja je, primjerice, „koristio stranice s opisom stilova, više fontova, veličina i vrsta poravnanja“¹² (Berners-Lee, FAQ, poglavlje *WorldWideWeb*, para. 3). U povijesti CSS-a važno je i razumjeti ostale jezike za stranice s opisom stilova.

3.1. DSSSL

Jezik za semantiku i specifikaciju stilova dokumenata (engl. *Document Style Semantics and Specification Language*), odnosno DSSSL, je još jedan stilski jezik koji se može koristiti za definiranje stila mrežne stranice. DSSSL je međunarodni standard, razvijen da bi pružio jezik za stranice s opisom stilova za dokumente za SGML, odnosno jezik za označavanje generalizacije standarda (engl. *Standard Generalizer Markup Language*). Sastoji se od dva dijela: slobodnog transformacijskog procesa koji može manipulirati razgranatom strukturom dokumenata prije prezentacije te od procesa formatiranja koji povezuje elemente u izvornom dokumentu s određenim točkama u prikazu ciljanog dokumenta, što zovemo Stablo toka objekata (engl. *flow object tree*, odnosno *FOT*) (van Ossenbruggen et al., 2002). DSSSL je kompatibilan s bilo kojim dokumentom definiranim kao SGML, uključujući i HTML (van Ossenbruggen et al., 2002).

Kod DSSSL-a je pozitivno što daje stranici s opisom stilova puni pristup i sadržaju i strukturi dokumenta pomoću snažnog jezika za upite. Stranice s opisom stilova DSSSL-a su najčešće neovisni o platformi i vrsti uređaja (van Ossenbruggen et al., 2002), što je još jedna prednost DSSSL-a.

Mane DSSSL-a su to što, primjerice, koristi programerski jezik *Scheme* („narječje“ jezika *Lisp*) za određivanje rasporeda dokumenta, što je mnogim autorima stran i nepoznat

¹² Ovo je slobodni prijevod autora koji u originalu na engleskom jeziku glasi: “*It used style sheets, and multiple fonts, sizes, and justification styles.*” (Berners-Lee, FAQ, potpoglavljje *WorldWideWeb*, para. 3)

pristup. Zbog toga što je DSSSL usko vezan uz SGML, potrebno je pozamašno znanje tog jezika za označavanje. Nadalje, teško je koristiti DSSSL za stvaranje HTML dokumenata. Zbog toga što DSSSL-ov jezik za transformaciju nije dovoljno rasprostranjen, DSSSL-ov jezik za formatiranje koristi se za stvaranje HTML dokumenata iz drugih SGML dokumenata. HTML-ovi strukturalni elementi nisu uključeni u standardni DSSSL-ov *flow object tree*. Zbog toga proces stvaranja HTML dokumenata u DSSSL-u zahtijeva stvaranje ekstenzija za standardni set objekata u *flowu* koje ovise o implementaciji (van Ossenbruggen et al., 2002).

3.2. XSL

XSL, odnosno Proširivi jezik za stranice s opisom stilova (engl. *eXtensible Style Sheet Language*), je stilski jezik za XML, predložen W3C-u 1997. godine. CSS nije dovoljno snažan za neke od funkcija koje XSL može pružiti, kao što je primjerice mapiranje (engl. *mapping*). XSL je većinski baziran na DSSSL-u, no ima neke ključne razlike (van Ossenbruggen et al., 2002):

- XSL će koristiti samo dio DSSSL-a koji mu je potreban da bi mogao specificirati online prikaz, za npr. najmanju moguću količinu mogućnosti ispisa XML dokumenata na Internetu;
- XSL pruža alternativu za DSSSL-ov *Scheme* jezik zahvaljujući svojoj sintaksi baziranoj na XML-u;
- XSL proširuje skup *flow* objekata koje DSSSL nudi, dodajući mu set HTML objekata. Oni imaju karakteristike bazirane na atributima HTML objekata, proširene kako je to definirano u CSS-u.

Najveća mana XSL-a je ta što ne podržava interaktivne dokumente (Leventhal, 1999), što znači da je na Internetu beskoristan, jer je jedna od glavnih odlika Interneta njegova interaktivnost, odnosno mogućnost da povezuje dokumente (mrežne stranice). XSL je

koristan jedino ako želimo koristiti isti jezik za stvaranje mrežnih stranica i tiskanih listova, s istim rasporedom sadržaja (Leventhal, 1999).

4. Principi dizajna u CSS-u

Sve razine CSS-a su građene na istim principima, a njih je W3C u svojoj preporuci za razinu CSS-a 2.1 definirao na sljedeći način (Bos et al., 2011):

- **kompatibilnost unaprijed i unatrag** – internetski preglednici koji koriste CSS 2.1 mogu razumjeti stranice s opisom stilova CSS-a 1 i CSS-a 2 te ih pravilno prikazati. Preglednici koji koriste CSS 1 mogu uglavnom razumjeti kasnije specifikacije i odbaciti dijelove koje ne mogu. CSS dijelovi koji nisu prikazani ne utječu na sam sadržaj stranice;
- **komplementarnost strukturiranim dokumentima** – stranice s opisom stilova nadopunjavaju strukturirane dokumente (npr. HTML i XML dokumente) tako što pružaju informaciju o stilu označenog (engl. *marked-up*) teksta. To nadopunjavanje dopušta da se stranica s opisom stilova izmjenjuje s vrlo malo ili čak nimalo utjecaja na sadržaj;
- **neovisnost o proizvođaču, platformi i uređaju** – stranice s opisom stilova brinu se za to da dokumenti ostanu neovisni o proizvođaču, platformi i uređaju te su i oni sami po sebi neovisni o proizvođaču i platformi, no CSS 2.1 (i CSS 3) nam dopuštaju da se bavimo stranicama s opisom stilova samo za jednu grupu uređaja, primjerice mobilne telefone;
- **održivost** – spajajući stranice s opisom stilova s dokumentima, mrežni dizajneri si olakšavaju buduće održavanje mrežnih stranica te čuvaju jednak izgled i osjećaj kroz cijelu stranicu;
- **jednostavnost** – CSS je jednostavni stilski jezik koji ljudi mogu čitati i pisati. Svojstva se u CSS-u čuvaju međusobno neovisnima, što znači da u većini slučajeva postoji samo jedan način za ostvarenje nekog efekta;
- **performanse mreže** – CSS pruža kompaktno kodiranje toga kako prezentirati sadržaj. Korištenje stranica s opisom stilova često smanjuje veličinu sadržaja na

disku pa se zato brže učitava i na mreži. Također, manje mrežnih veza mora biti otvoreno što dodatno poboljšava performanse mreže;

- **fleksibilnost** – CSS se može spojiti sa sadržajem na mnoštvo načina. Ključna osobina CSS-a je sposobnost kaskadnog slaganja informacija o stilu koje su pohranjene u zadanim postavkama preglednika, eksternim stranicama s opisom stilova, internim stranicama s opisom stilova te u CSS-u unutar linije;
- **bogatstvo** – to što mrežni dizajneri imaju ogroman set mogućnosti za svoje mrežne stranice povećava bogatstvo Interneta kao medija za izražavanje. CSS 2.1 je dizajnerima pružio dugo tražene mogućnosti oblikovanja izgleda stranice koje često nalazimo u programima poput *Microsoftovog Power Pointa*;
- **veze s alternativnim jezicima** – CSS je moguće povezati i s drugim jezicima za programiranje. Na primjer, program u JavaScriptu može dinamično mijenjati vrijednost *color* svojstva nekog elementa;
- **pristupačnost:**
 - svojstva za kontrolu izgleda fonta dopuštaju autorima da uklone tekst koji nije hipertekst (bit-mape, primjerice);
 - svojstva za pozicioniranje daju autorima mogućnost da prestanu s korištenjem trikova u jezicima za označavanje kao što su ubacivanje nevidljivih slika da bi forsirali određeni raspored na stranici;
 - korištenje pravila označenih oznakom važnosti *!important* znači da korisnici sa specifičnim zahtjevima za prezentiranje mogu premostiti autorove stranice s opisom stilova (npr. povećanje veličine fonta za slabovidne osobe, promjena fonta u sans-serifni za disleksične osobe itd.);
 - vrijednost *inherit* za razna svojstva pomaže u ostvarivanju više konzistentnog prilagođavanja stilova;
 - poboljšana potpora za medije dopušta autorima da svoje stranice kroje prema tim uređajima.

Svi navedeni principi se mogu naći u razinama CSS-a. Primjerice to da CSS pravilno funkcionira na svim Internet preglednicima i uređajima, za razliku od ranijih stranica s opisom stilova, koje su često radile samo na jednom pregledniku. Tako je preglednik *Viola* programera Peija Weija iz 1992. imao svoj vlastiti jezik za stranice s opisom stilova (Bos, 2016).

5. Zaključak

Svrha ovog rada je bila upoznati čitatelja sa stilskim jezikom CSS, koji se koristi za formatiranje neke mrežne-stranice, a zatim s njegovim povijesnim razvojem - od njegovih začetaka 1994. godine do danas. U prvom poglavlju su pokriveno osnove korištenja CSS-a u svrhu lakšeg razumijevanja rada za čitatelje koji nisu otprije upoznati s CSS-om, primjerice, sintaksa i osnovna terminologija (selektori, svojstva, vrijednosti, pravila).

Drugo poglavlje se prvo bavi općom povijesti CSS-a (rad Wiuma Lieja i Bosa), a zatim time kako Konzorcij Svjetske mreže dijeli razne faze u životnom vijeku implementacija CSS-a (radna skica, kandidatska preporuka, ponuđena preporuka te preporuka). Potpoglavlja koja slijede nakon toga se bave pojedinačnim razinama CSS-a te ih uspoređuju. Pretposljednje potpoglavlje se bavi drugim stilskim jezicima (DSSSL, XSL) i time što te jezike čini drukčijima od CSS-a, dok zadnje poglavlje pokriva principe dizajna kojih se pridržavaju sve razine CSS-a.

Internet je u današnjem digitalnom svijetu od neprocjenjive važnosti zbog svih svojih mogućnosti i snaga. Internetski preglednici nam donose sav njegov sadržaj, no HTML, glavni pokretač skoro svih informacija na Internetu, ne može to prikazati na oku-ugodan način. To je svrha CSS-a. Zbog toga je osmišljen i razvijen do ove razine na kojoj ga danas vidimo. Lijep Internet postoji zbog CSS-a.

Važnost rada Håkona Wiuma Lieja i Berta Bosa u povijesti, ne samo CSS-a, već i samog Interneta u cijelosti, se ne može precijeniti. Da već skoro 30 godina W3C-ova CSS radna grupa ne radi na poboljšavanju stranica s opisom stilova, Internet bi izgledao mnogo drukčije. Što kad bi primjerice svi gumbi na mrežnim stranicama morali izgledati jednako jer ne postoji stilski jezik koji može urediti HTML-ovu `<button>` oznaku? Mrežne stranice bi bile mnogo monotonije te bi bilo teže nekoj kompaniji razviti vlastiti stil na svojim mrežnih mjestima.

Od prve ideje za CSS 1994. godine do najnovijih modula CSS-a 3 koji postaju službenim preporukama W3C-a, smatram da je CSS dokaz da ljudi ne mogu biti samo roboti koji mogu bez problema gledati u tone podataka i informacija bez ikakve vizualne varijacije.

Ljudima treba vizualna stimulacija koju CSS pruža dok čitaju (bilo veće ili manje količine teksta), ili će izgubiti koncentraciju i volju za radom. Moje je mišljenje da su mogućnost i sposobnost CSS-a da prikažu našu kreativnost i viziju na računalnim ekranima upravo onako kako smo zamislili od neprocjenjive važnosti.

6. Literatura

1. Andrew, Rachel. (2004). The CSS Anthology – 101 Essential Tips, Tricks & Hacks.
2. Atkins, Tab Jr; J. Etemad, Elika; Rivoal, Florian. (2019. 1. 22). *CSS Snapshot 2018*. <https://www.w3.org/TR/CSS/>
3. Berners-Lee, Tim. Frequently Asked Questions.
<https://www.w3.org/People/Berners-Lee/FAQ.html>
4. Bos, Bert. (2016. 12. 17). A brief history of CSS until 2016.
<https://www.w3.org/Style/CSS 20/history.html>
5. Bos, Bert. (2020. 2. 18). The CSS Standardization Process.
<https://www.w3.org/Style/2011/CSS-process>
6. Bos, Bert; Çelik, Tantek; Hickson, Ian; Wium Lie, Håkon. (2011. 6. 7). *Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification*.
<https://www.w3.org/TR/CSS2/>
7. Bos, Bert; Wium Lie, Håkon; Lilley, Chris; Jacobs, Ian. (1998. 5. 12). *Cascading Style Sheets, level 2*. <https://www.w3.org/TR/1998/REC-CSS2-19980512/>
8. Leventhal, Michael. (1999. 5. 20). *XSL Considered Harmful*.
9. Lilley, Chris; Kravets, Una; Verou, Lea; Argyle, Adam (2020. 3. 3). *CSS Color Module Level 5*. <https://www.w3.org/TR/css-color-5/>
10. Mozilla Developer Network. (2019. 11. 7). At-rules.
<https://developer.mozilla.org/en-US/docs/Web/CSS/At-rule>
11. Mozilla Developer Network. (2020. 4. 19). CSS3.
<https://developer.mozilla.org/en-US/docs/Archive/CSS3>
12. Opera. (2004. 10. 24). *Opera's Small-Screen Rendering™*.
<https://web.archive.org/web/20071204034833/http://www.opera.com/products/mobile/smallscreen/>
13. Sciactive. (2008. 2. 12). *Acid Test Results on Popular Browsers*.
https://web.archive.org/web/20080505035123/http://www.sciactive.com/main/index.php?Itemid=1&id=133&option=com_content&task=view

14. van Ossenbruggen, Jacco; Hardman, Lynda; Rutledge, Lloyd; Eliens, Anton. (2002). *Style Sheet Languages for Hypertext*.
<https://homepages.cwi.nl/~jrvosse/publications/1997/dv:siglink.pdf>
15. W3C. (2020). *Facts About W3C*. <https://www.w3.org/Consortium/facts>
16. W3C. (2016). *HTML & CSS*. <https://www.w3.org/standards/webdesign/htmlcss>
17. Wium Lie, Håkon. (1994. 10. 10). Cascading HTML style sheets - a proposal.
<https://www.w3.org/People/howcome/p/cascade.html>
18. Wium Lie, Håkon; Bos, Bert. (1996. 12. 17). *Cascading Style Sheets, level 1*.
<https://www.w3.org/TR/CSS 1/>
19. Wium Lie, Hakon; Bos, Bert. (2005. 5. 5). *Cascading Style Sheets: Designing for the Web* (3rd ed). Addison-Wesley Professional.

Sažetak

Svrha ovog rada je upoznati čitatelja sa stilskim jezikom CSS (*Cascading Style Sheets*), koji se koristi za definiranje formatiranja na nekoj Web-stranici; točnije s njegovim povijesnim razvojem - od njegovih začetaka 1994. do danas. Što se tiče korištenja CSS-a, pokrivene su osnove sintakse i terminologije CSS-a; a što se tiče povijesti tog stilskog jezika, čitatelj je upoznat s verzijama CSS-a, od CSS-a 1, preko CSS-a 2 i 2.1 do konačnog CSS-a 3, koji se koristi danas.

Ključne riječi: *CSS, Web, Web-dizajn, Internet, HTML, povijesni razvoj*

Historical Development of CSS

Abstract

The purpose of this bachelor's thesis is to introduce the reader to the CSS (Cascading Style Sheets) style language which is used to define the formatting of a Website, more specifically its historical development – from its inception in 1994 until today. Regarding the use of CSS, the basics of its syntax and terminology are covered; and regarding the history of the style language, the reader is introduced to versions of CSS, from CSS 1, through CSS 2 and CSS 2.1, to the final CSS 3 version, which is still used today.

Key words: *CSS, Web, Web-design, Internet, HTML, historical development*