

Usporedba CSS-a i XSL-a za prikaz XML dokumenata

Ageljić, Monika

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Humanities and Social Sciences / Sveučilište Josipa Jurja Strossmayera u Osijeku, Filozofski fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:142:926490>

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

Download date / Datum preuzimanja: **2024-12-31**



FILOZOFSKI FAKULTET
SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

Repository / Repozitorij:

[FFOS-repository - Repository of the Faculty of Humanities and Social Sciences Osijek](#)



Sveučilište J. J. Strossmayera u Osijeku

Filozofski fakultet Osijek

Sveučilišni prijediplomski jednopredmetni studij Informatologija

Monika Ageljić

Usporedba CSS-a i XSL-a za prikaz XML dokumenata

Završni rad

Mentor: prof. dr. sc. Boris Bosančić

Osijek, 2024.

Sveučilište J. J. Strossmayera u Osijeku

Filozofski fakultet Osijek

Odsjek za informacijske znanosti

Sveučilišni prijediplomski jednopredmetni studij Informatologija

Monika Ageljić

Usporedba CSS-a i XSL-a za prikaz XML dokumenata

Završni rad

Društvene znanosti, informacijske i komunikacijske znanosti, informacijski sustavi
i informatologija

Mentor: prof. dr. sc. Boris Bosančić

Osijek, 2024.

Prilog: Izjava o akademskoj čestitosti i o suglasnosti za javno objavljivanje

Obveza je studenta da donju Izjavu vlastoručno potpiše i umetne kao treću stranicu završnoga, odnosno diplomskog rada.

IZJAVA

Izjavljujem s punom materijalnom i moralnom odgovornošću da sam ovaj rad samostalno napisao/napisala te da u njemu nema kopiranih ili prepisanih dijelova teksta tuđih radova, a da nisu označeni kao citati s navođenjem izvora odakle su preneseni.

Svojim vlastoručnim potpisom potvrđujem da sam suglasan/suglasna da Filozofski fakultet u Osijeku trajno pohrani i javno objavi ovaj moj rad u internetskoj bazi završnih i diplomskih radova knjižnice Filozofskog fakulteta u Osijeku, knjižnice Sveučilišta Josipa Jurja Strossmayera u Osijeku i Nacionalne i sveučilišne knjižnice u Zagrebu.

U Osijeku 12.09.2024.

Monika Ageljić, 0122238920

Ime i prezime studenta, JMBAG

SAŽETAK

U ovom radu analiziraju se različiti pristupi stiliziranju i transformiranju XML dokumenata putem CSS-a (Cascading Style Sheets) i XSL-a (eXtensible Stylesheet Language). XML, koji se pojavio 1998. godine, postao je ključan za pohranu i razmjenu podataka na webu, no mrežni preglednici u početku nisu mogli prikazivati XML dokumente osim kao obične tekstualne zapise. Kao privremeno rješenje, CSS je korišten za stiliziranje XML dokumenata, ali budući da je CSS izvorno razvijen za rad s HTML-om, nedostajale su mu mogućnosti potrebne za rad s XML-om, osobito u kontekstu transformacija. Kako bi se taj problem riješio, W3C je razvio XSL, alat koji omogućuje ne samo stiliziranje nego i transformaciju XML dokumenata u različite formate. Ovaj rad detaljno razmatra povijest i razvoj XML-a, CSS-a i XSL-a te pruža pregled njihovih ključnih značajki uz primjere iz prakse. Poseban fokus stavljen je na usporedbu CSS-a i XSL-a te njihove prednosti i nedostatke u radu s XML dokumentima. Na kraju rada, na temelju provedene analize, donose se zaključci o optimalnoj primjeni CSS-a i/ili XSL-a, ovisno o specifičnim potrebama i ciljevima u radu s XML-om.

Ključne riječi: označiteljski jezici, XML dokumenti, XSL, XSLT, CSS

Sadržaj

SAŽETAK.....	4
1. UVOD	6
2. Općenito o CSS-u.....	8
2.1. Povijest CSS-a	8
2.2. CSS3	9
2.3. Sintaksa CSS-a	10
3. Općenito o XSL-u	12
3.1. Povijest XSL-a.....	13
3.2. XSLT	14
4. CSS vs. XSL.....	17
4.1. Liste stilova za XML	17
4.2. CSS za prikaz XML dokumenata	18
4.3. XSL za prikaz XML dokumenata.....	20
4.4. CSS vs. XSL.....	21
5. ZAKLJUČAK	24
POPIS LITERATURE	25

1. UVOD

Označiteljski jezici ključan su dio weba još od 1960-ih godina. Jedan od poznatijih označiteljskih jezika je XML (*eXtensible Markup Language*). Osim što je XML označiteljski jezik, on je također i metajezik. Zbog toga što je istovremeno i označiteljski i metajezik, XML se ne sastoji od unaprijed definiranog skupa elemenata.¹ Kreirao ga je W3C (*World Wide Web Consortium*) 1998. godine, kao pojednostavljenu verziju SGML-a (*Standard Generalized Markup Language*). Glavna svrha XML-a je pohrana podataka radi lakšeg čitanja i dijeljenja podataka.² Osim toga, XML je omogućio predstavljanje strukturiranih informacija na webu, dosegao nove razine automatizacije i brojne druge mogućnosti.³ Nakon objavljivanja XML-a, gotovo nijedan mrežni preglednik nije imao mogućnost prikazivanja XML dokumenata osim kao tekstualnih dokumenata. Nakon nekog vremena podrška za XML u mrežnim preglednicima je rasla, a *Internet Explorer* bio je prvi mrežni preglednik koji je prikazivao XML dokumente u obliku stabla.

XSL (*eXtensible Stylesheet Language*) se pojavio dvije godine nakon XML-a, pa su do tog vremena mnogi kao alternativu za prikaz sadržaja XML dokumenata u pregledniku koristili CSS (*Cascading Style Sheets*). Mrežni preglednici su već podržavali CSS i uzimajući u obzir kako su i HTML (*HyperText Markup Language*) i XML jezici za označavanje, nije bio težak izazov prikazati XML koristeći CSS.⁴ Iako se CSS koristio već neko vrijeme, W3C objavljuje XSL kao preporuku za rad s XML dokumentima. XSL je kreiran kako bi omogućio različita formatiranja istog sadržaja XML dokumenata i njihov prikaz u pregledniku.. CSS je kreiran za prikaz HTML dokumenata u pregledniku, ali unatoč tomu, zbog jednostavnosti korištenja, mnogi i dalje preferiraju korištenje CSS-a za označavanje XML dokumenata umjesto XSL-a.⁵ Osim korištenja XSL-a ili CSS-a za prikaz XML dokumenata, moguće je koristiti XSL i CSS u kombinaciji. S obzirom na brojne mogućnosti transformiranja XML dokumenata pomoću XSL-a, moguće je XML dokumente konvertirati u HTML format, a zatim dobiveni HTML dokument prikazati putem CSS-a. Mišljenja stručnjaka o tome koji jezik se preporučuje za rad s XML dokumentima i danas

¹ Usp. Morrison, Michael. Comparing XSL and CSS // XML Unleashed. // O'Reilly, 1999., URL: https://learning.oreilly.com/library/view/xml-unleashed/0672315149/0672315149_ch08lev1sec3.html (2024-08-24)

² Usp. Schmidt, Desmond. The Role of Markup in the Digital Humanities. // Historical Social Research 3 (141), 37 (2012), str. 127 URL: <http://www.jstor.org/stable/41636601> (2024-08-23)

³ Usp. Morrison, Michael, Nav. dj.

⁴ Usp. Gulbransen, David. Using XML [Google books]. Indianapolis, Ind.: Que Pub., 2002., str. 174

URL: <https://books.google.hr/books?id=aqfvLQzR-GkC&lpg=PA1&ots=-8sUc6tzjB&dq=Comparison%20of%20CSS%20and%20XSL%20for%20displaying%20XML%20documents&lr&hl=hr&pg=PA218#v=onepage&q=Comparison%20of%20CSS%20and%20XSL%20for%20displaying%20XML%20documents&f=false> (2024-08-23)

⁵ Isto.

su podijeljena. Izbor između CSS-a i XSL-a za rad s XML dokumentima uvelike ovisi o samim ciljevima koji se žele postići.

Svrha rada je opisati usporedbu CSS-a i XSL-a za prikaz XML dokumenata. U skladu sa svrhom ovog rada, ciljevi rada su prikazati povijest označiteljskih jezika, proći kroz sintaksu i strukturu obje tehnologije te prikazati praktične primjere. Prvo poglavlje rada odnosi se na prikaz CSS-a, ključne značajke, povijest te primjere kako se koristi. U drugom poglavlju govori se o XSL-u te kao i u prošlom poglavlju, obradit će se osnovne informacije o jeziku, povijest, sintaksa te praktična primjena. Treće poglavlje pružit će usporedbu CSS-a i XSL-a, čime će se omogućiti dublje razumijevanje njihove međusobne komplementarnosti i prednosti u radu s XML sadržajem. U Zaključku rada će se sumirati ključne razlike i prednosti ovih tehnologija, te razmotriti njihova uloga i primjena u suvremenom web razvoju.

2. Općenito o CSS-u

CSS je stilski jezik koji služi za opisivanje prikaza dokumenata, uglavnom onih izrađenih u HTML-u, ali i u drugim srodnim jezicima poput XML-a.⁶ Srž CSS-a je u određivanju izgleda i prikaza različitih elemenata dokumenta u mrežnim preglednicima, ali i u drugim postojećim medijima. CSS određuje način na koji će se elementi HTML-a prikazati što se u prijašnjim verzijama HTML-a postizalo atributima i oznakama unutar samog HTML kôda.⁷ Osmišljen je kako bi se omogućilo jednostavno uređivanje i stiliziranje HTML dokumenata, čime se omogućilo i željeno odvajanje sadržaja od prezentacije.⁸

Sâm naziv sugerira da se radi o listama stilova kojima se definira izgled HTML elemenata.⁹ Pomoću ovog jezika moguće je definirati izgled teksta, veličinu i dimenzije elemenata, odrediti boje, pozicije elemenata i brojne druge karakteristike koje su ključne za izgled svakog HTML dokumenta. Prva riječ u nazivu, „*cascading*“, ukazuje na mogućnost primjene novih stilova preko postojećih, stvarajući time dinamičan i prilagodljiv dizajn mrežnih stranica. Za ovu riječ ne postoji hrvatska riječ, osim „kaskadni“, ali najbliži prijevod je „padati u slapovima“.

Ključno pravilo CSS-a jest pravilo nasljeđivanja. Ako postoji „sukob“ između stilova, stilovi koji su naknadno definirani imaju prioritet te zamjenjuju ranije definirane stilove. U konačnici, „*cascading*“ ukazuje na mogućnosti, kao što su jednostavne prilagodbe i nadogradnje dizajna, koje doprinose boljoj i jednostavnijoj organizaciji te održivosti stila HTML dokumenata.

2.1. Povijest CSS-a

Kao što je već spomenuto, prije pojave CSS-a, izgled i dizajn HTML elemenata se određivao u samom HTML kôdu. Treba imati na umu kako je HTML prvotno napravljen kao sredstvo za predstavljanje statičkih informacija, a dodavanje interaktivnosti i dinamičnosti u HTML kôd značilo je promjenu same njegove prirode. S vremenom se HTML razvijao i unaprjeđivao kako bi se njime moglo ostvariti što više mogućnosti, ali je također postajala i sve jasnija potreba za novim jezikom. U tim trenucima, na tržištu je već postojalo nekoliko stilskih jezika koje se moglo koristiti, ali nije postojalo standardno rješenje.

⁶ Usp. Vlašić, Goran. CSS. Zagreb: Algebra, 2016., str. 4

⁷ Usp. Maleš, Lada. Osnove programiranja za web: (HTML, JavaScript, XML i XSL). Split: Filozofski fakultet Sveučilišta, 2007., str. 11

⁸Usp. Morrison, Michael. Comparing XSL and CSS // XML Unleashed. // O'Reilly, 1999., URL: https://learning.oreilly.com/library/view/xml-unleashed/0672315149/0672315149_ch08lev1sec3.html (2024-08-24)

⁹ Usp. Vlašić, Goran. CSS. Zagreb: Algebra, 2016., str. 4

CSS je, zapravo, nastao spajanjem CHSS-a (*Cascading HTML Style Sheets*) i SSP-a (*Stream-based Style Sheet Proposal*). Prva skica CSS-a nastala je 1994. godine, a nakon toga W3C preuzima zadatak daljnjeg razvoja CSS-a. Kao standardni jezik, CSS je objavljen 1996. godine, točnije u prosincu te godine, te je prva verzija dobila naziv CSS1. Već sljedeće godine pojavljuje se i prva radna skica CSS2, a 1998. godine CSS postaje službeni standard. Uvođenjem CSS2 kao standardnog jezika predstavljene su brojne nove mogućnosti koje su vidno nedostajale u CSS1 te je zahvaljujući tomu CSS ušao u širu primjenu.¹⁰ Već 1999. godine, objavljene su prve radne skice novog standarda - CSS3. Značajna razlika kod CSS3 je ta što se, za razliku od CSS2 standarda, koji je bio cjelovita specifikacija, CSS3 podijelio na više odvojenih specifikacije nazvanih moduli. Svaki od tih modula omogućio je nove promjene i proširio već postojeće značajke CSS2 standarda.

U lipnju 2012. godine, CSS3 sadrži već više od pedeset modula koje razvija i objavljuje radna skupina koja se bavi razvojem CSS-a. Od pedeset modula, četiri su dobila službenu preporuku za korištenje: *Media Queries*, *Namespaces*, *Selectors Level 3* i *Color*. Svaki od navedenih modula imao je značajnu ulogu u osiguravanju funkcionalnosti CSS-a.¹¹ Danas na službenoj stranici W3C, svi CSS moduli mogu se svrstati u tri kategorije: u prvoj se nalaze moduli koji se smatraju stabilnima, u drugoj kategoriji su moduli za koje se smatra da su dovoljno stabilni, ali nisu u dovoljnoj mjeri testirani, a u trećoj se nalaze moduli koji su također vrlo korišteni, ali njihovi detalji nisu dovoljno razrađeni te se time ne može jamčiti njihova stabilnost.¹²

2.2. CSS3

CSS3 je posljednja službena verzija CSS-a, osmišljena s ciljem proširivanja mogućnosti prethodnog standarda, uz zadržavanje kompatibilnosti s ranijim verzijama, CSS1 i CSS2. Uvođenjem CSS3 otvorile su se brojne nove mogućnosti za dizajn koje su značajno proširile tadašnje granice web dizajniranja, a neke od novina su *CSS Flexbox Layout*, *CSS Grid Layout*, *Media Queries* i brojne druge.¹³ Kao što je već spomenuto, CSS3 je prva verzija CSS-a čije su specifikacije podijeljene na više zasebnih modula od kojih svaki pokriva specifične funkcionalnosti ili proširuje već postojeće značajke iz CSS2. Takav pristup je omogućio brži razvoj i jednostavniju implementaciju novih značajki.

Jedan od najvećih problema s kojima se CSS3 standard susreo, pogotovo u ranijim fazama njegovog razvoja, bio je nedostatak podrške u starijim mrežnim preglednicima. No i tom problemu

¹⁰ Usp. Vlašić, Goran. CSS. Zagreb: Algebra, 2016., str. 5

¹¹ Isto.

¹² Usp. CSS Snapshot 2023. URL: <https://www.w3.org/TR/CSS/#css> (2024-08-22)

¹³ Usp. CSS Snapshot 2023. URL: <https://www.w3.org/TR/CSS/#css> (2024-08-22)

je ubrzo pronađeno rješenje. Kako bi se osigurala kompatibilnost sa starijim mrežnim preglednicima te omogućilo korištenje novih značajki CSS3, uvedeno je takozvano korištenje „vendor prefiksa“ odnosno oznaka proizvođača.¹⁴ Danas su takvi problemi rijetki, odnosno većina modernih mrežnih preglednika ima vrlo dobru podršku za sve CSS3 module. Problemi s kompatibilnošću su u manjoj mjeri prisutni nego što su to bili ranije, ali i dalje postoje, osobito kod novijih značajki koje nisu u potpunosti standardizirane i stabilne.

Unatoč svim izazovima s kojima se CSS3 susreo, i s kojima se i dalje suočava, on ostaje ključan alat u modernom web dizajnu te je bez njega nezamislivo stvaranje vizualno bogatih, dinamičnih i prilagodljivih mrežnih stranica. Zbog stalnog razvoja tehnologije, i sâm CSS3 je konstantno u razvoju te radi osiguranja stabilnosti i kompatibilnosti mrežnih stranica s preglednicima preporučuje se praćenje najnovijih specifikacija.

2.3. Sintaksa CSS-a

CSS je prilično jednostavan za korištenje, zapravo, radi se o pisanju blokova kôda koji se nazivaju CSS pravila ili CSS stilovi. Ta pravila ili stilovi, HTML elementima određuju specifična svojstva, kao što je boja, font, veličina, kako bi ih mrežni preglednici prikazali na ciljani način. Svaki CSS kôd se sastoji od selektora, svojstva i vrijednosti. Selektor je srž svakog kôda, on određuje na koje će se HTML elemente primijeniti pravilo, npr. hoće li se željeni stil primijeniti na cijeli tekst ili samo određeni odlomak teksta. Svojstvo je, pak, stilsko obilježje koje će se primijeniti na željeni element, npr. širina elementa, boja, font i dr. Posljednja komponenta predstavlja vrijednost koja određuje koja je vrijednost stilskog obilježja koje je definirano u svojstvu, npr. koja boja će se primijeniti u CSS stilu.¹⁵ Istim redoslijedom kojim su ovdje navedeni, navode se i u CSS kôdu. Pravilnu strukturu i redoslijed CSS kôda koji se sastoji od jednog selektora te jednog svojstva i vrijednosti moguće je vidjeti na Slici 1.

```
selektor {svojstvo: vrijednost;}
```

Slika 1. Struktura CSS kôda.

Naravno unutar jednog selektora (h1) moguće je napisati više od jednog svojstva (*font-size*, *color* itd.) odnosno vrijednosti; važno je svojstva odvojiti točka-zarezom (;), kao što je prikazano na slici (Slika 2).

¹⁴ Usp. Vlašić, Goran. CSS. Zagreb: Algebra, 2016., str. 122

¹⁵ Isto, str. 10

```
selektor {svojstvo: vrijednost; svojstvo2: vrijednost;}
```

Slika 2. Struktura CSS kôda s više svojstava.

Snaga CSS-a krije se upravo u beskonačnim mogućnostima koje pruža, primjerice, jedan selektor može imati neograničenu količinu svojstava koja se mogu kombinirati kako bi se postigao željeni izgled i dizajn elementa. Jedan h1 naslov može se urediti na brojne načine, a samo neke od osnovnih mogućnosti prikazane su na primjeru (Slika 3). U svrhu bolje preglednosti, preporučuje se odvajanje kôda u različite redove. Također nije nužno stavljati točku-zarez (;) nakon svakog reda u CSS-u, ali to se ipak preporučuje kao dobra praksa radi održavanja konzistentnosti i izbjegavanja mogućih grešaka.¹⁶

```
h1 {  
  font-size: 32px;  
  color: ■ #333333;  
  text-align: center;  
  background-color: □ #f0f0f0;  
  padding: 10px;  
}
```

Slika 3. Primjer CSS kôda za h1 element.

Može se zaključiti kako je CSS imao velik utjecaj na razvoj mrežnih stranica te uveo brojne novine u oblikovanju i prikazivanju elemenata na mrežnim stranicama. Prije uvođenja CSS-a, elementi i stilovi morali su se definirati unutar samog HTML kôda što je u brojnim slučajevima dovodilo do nepreglednog kôda koji je bio teško održiv. CSS je omogućio vrlo poželjno odvajanje sadržaja od prezentacije te je time automatski ostvarena veća fleksibilnost i jednostavnost u održavanju kôda. Iako je CSS kreiran specifično za HTML kôd, kasnije je postao primjenjiv i na XML dokumente, o čemu će biti više riječi kasnije u radu. CSS je postao ključan alat u modernom web dizajnu te je zadržao taj značaj sve do danas. S obzirom na činjenicu da se uspio prilagoditi razvoju tehnologije i HTML-a kroz vrijeme, pretpostavlja se da će nastaviti evoluirati kako bi pratio sve složenije zahtjeve modernog web dizajna. CSS će i dalje imati središnju ulogu ne samo u dizajnu mrežnih stranica, nego i u raznim oblicima digitalnih medija i struktura podataka.

¹⁶ Isto.

3. Općenito o XSL-u

XSL je označiteljski jezik koji je nastao s jednim ciljem, a to je transformacija XML dokumenata u različite druge formate, a i sâm je baziran na XML-u.¹⁷ Njegov primarni zadatak je isporučiti XML dokumente u različitim formatima, preglednicima i poslužiteljima. Prikaz XML dokumenata prije pojave XSL-a predstavljao je problem za mrežne preglednike. Pojavom XSL-a, postalo je moguće promijeniti format XML dokumentima kako bi ih mrežni preglednici mogli prepoznati i prikazati, a jedan od formata u koje je moguće transformirati XML dokumente je i HTML. Osim transformacije, XSL nudi opcije testiranja, dodavanja, brisanja, razne načine prikaza elemenata i brojne druge opcije.¹⁸

Prvotno, XSL se sastojao od tri različita standarda: XSLT-a (*XSL Transformations*), XPath-a i XSL-FO-a (*XSL Formatting Objects*). XSL-FO je jedini XSL standard koji se službeno više ne razvija i njega je u potpunosti zamijenio CSS3. XSLT je, kao što i sam naziv govori, jezik kojim se transformira XML dokument. Ovaj jezik je utemeljen na predlošcima (engl. *templates*) koji određuju kako će pojedini XML dokument biti obrađen, a što upravo odvaja ovaj jezik od ostalih programskih jezika koji su sekvencijalni.¹⁹ Već je navedeno kako je XSL utemeljen na XML-u, ali za razliku od XML-a, XSL koristi vlastiti rječnik elemenata. Sukladno tomu, XSLT koristi već definirane nazive elemenata koji imaju određeno značenje. Sam proces transformacije, obradit će se detaljnije u potpoglavlju posvećenom XSLT-u.

XPath je neizostavan dio XSL-a, budući da omogućava selektiranje, odnosno odabir i pronalazak željenih dijelova XML dokumenta. XPath definira sintaksu koja se može smatrati putanjom koja vodi do svih dijelova XML dokumenta. Najnovija verzija XPath-a je 3.1 iz 2017. godine, no najkorištenija verzija i dalje je XPath-a 1.0. I XPath osmislio je W3C tako da se može integrirati u razne jezike; XPath koristi ne samo XSLT, već i drugi jezici kada rade s XML dokumentima. Posljednji dio XSL-a je XSL-FO, jezik koji se koristio za definiranje izgleda podataka iz XML dokumenata u PDF dokumentima.²⁰

¹⁷ Usp. Brčina, Josip. Primjena XSL-a u digitalnoj knjižnici hrvatske tiskane baštine Edicija : diplomski rad. Osijek: Filozofski fakultet, 2014., str. 15 URL: <https://repozitorij.ffos.hr/islandora/object/ffos%3A1497/datastream/PDF/view> (2024-08-20)

¹⁸ Usp. Maleš, Lada. Osnove programiranja za web: (HTML, JavaScript, XML i XSL). Split: Filozofski fakultet Sveučilišta, 2007., str. 12

¹⁹ Usp. Brčina, Josip. Primjena XSL-a u digitalnoj knjižnici hrvatske tiskane baštine Edicija : diplomski rad. Osijek: Filozofski fakultet, 2014., str. 16 URL: <https://repozitorij.ffos.hr/islandora/object/ffos%3A1497/datastream/PDF/view> (2024-08-20)

²⁰ Usp. Sheldon, Robert. XSL (Extensible Stylesheet Language). URL: <https://www.techtarget.com/whatis/definition/XSL-Extensible-Stylesheet-Language> (2024-08-23)

3.1. Povijest XSL-a

W3C započeo je razvoj jezika za transformaciju i uređivanje XML dokumenata sredinom 1990-ih. Kao preporuka, XSL je prvi put objavljen 1998. godine, i već prva verzija XSL-a bila je usmjerena prema omogućavanju prikaza XML dokumenata u preglednicima. Ubrzo nakon toga, 1999. godine, XSL se razdvaja u dva dijela: XSLT i XSL-FO. XSLT 1.0 objavljen je u studenom 1999. godine i predstavlja prvu verziju ovog jezika koja je omogućila transformaciju XML dokumenata u razne formate, uključujući i HTML, koju je objavio *XSLT Working Group*.²¹ Zanimljivo da je druga po redu objavljena verzija XSLT 3.0 bila objavljena u lipnju 2017. godine. Nakon XSLT 3.0, objavljena je i verzija XSLT 2.0 u ožujku 2021. godine, te se u opisu ove verzije navodi kako je ovo prerađena verzija originalne odnosno prve verzije, XSLT 1.0 iz 1999. godine. XSLT 2.0 uveo je brojne napredne funkcionalnosti, a neke od njih su unaprijeđeno upravljanje podacima, proširena podrška za vrste podataka itd.²²

Iako se XSL u početku dijelio na dva dijela, razvoju XSLT-a pridružio se i treći dio, XPath. Spomenuti XPath zapravo je sastavni dio XSLT-a, jer omogućuje navigaciju kroz XML dokumente. Prva verzija XPath 1.0 objavljena je kao preporuka u studenom 1999. godine, a objavila ga je *XML Linking Working Group*. Drugu verziju XPath-a, XPath 2.0, objavio je *XML Query Working Group* i *XSLT Working Group* u prosincu 2010. godine. Konačno, u travnju 2014. godine, objavljena je i treća verzija, XPath 3.0, koja je prvotno nosila naziv XPath 2.1, dok je najnovija verzija XPath 3.1 objavljena u ožujku 2017. godine, a objavili su ju također *XML Query Working Group* i *XSLT Working Group*.²³

XSL je od trenutka svoje objave izvršio veliki utjecaj na način na koji se pristupa XML dokumentima i na koje načine ih se može transformirati. Integracijom s drugim XML tehnologijama, XSL je uistinu postao nezamjenjiv alat kada se radi s XML-om. XSL i danas ima važnu ulogu u obradi XML podataka, unatoč razvoju i pojavi novih tehnologija kao što je JSON, i brojne druge. Bitno je napomenuti kako se XSL intenzivno koristi u području digitalne humanistike, jer bi bez upotrebe XSL-a bilo vrlo teško osmisliti i razviti novi pristup koji omogućava jedinstvenu prezentaciju XML sadržaja i njegovo pretraživanje.

²¹ Usp. Gulbransen, David. Using XML [Google books]. Indianapolis, Ind.: Que Pub., 2002., str. 190

URL: <https://books.google.hr/books?id=aqfvLQzR-GkC&lpg=PA1&ots=-8sUc6tzjB&dq=Comparison%20of%20CSS%20and%20XSL%20for%20displaying%20XML%20documents&lr&hl=hr&pg=PA218#v=onepage&q=Comparison%20of%20CSS%20and%20XSL%20for%20displaying%20XML%20documents&f=false> (2024-08-23)

²² Usp. The Extensible Stylesheet Language Family (XSL). URL: <https://www.w3.org/Style/XSL/> (2024-08-20)

²³ Usp. The Extensible Stylesheet Language Family (XSL). URL: <https://www.w3.org/Style/XSL/> (2024-08-20)

3.2. XSLT

Kao što je već navedeno, XSLT kreiran je radi transformacije XML dokumenata u druge formate kao što su HTML, PDF i slično, ali i u druge XML dokumente. XSLT preuzima izvorno XML stablo (engl. *XML source-tree*) i od njega stvara rezultirajuće stablo (engl. *XML result-tree*). Osim svoje glavne funkcije transformiranja XML dokumenata, XSLT ima i brojne druge mogućnosti. XSLT ima mogućnost odlučivanja koje će elemente prikazati, a koje ne, a osim toga ima i mogućnost sortiranja elemenata. Važno je napomenuti da XSLT zapravo posjeduje vlastite elemente koji su ključni za transformaciju XML dokumenata prema zadanim pravilima i standardima.²⁴

Svaki XSL dokument sadrži predloške koji definiraju načine na koje se sadržaj XML dokumenata prikazuje u mrežnim preglednicima. XSL procesor prvo „čita“ XML dokument te za svaki element pokušava odrediti odgovarajući XSLT predložak. Kada procesor pronađe predložak, njegov sadržaj se obrađuje kako bi se prikazao u mrežnom pregledniku. Taj proces se ponavlja za svaki sljedeći pronađeni element i odgovarajući predložak. Unatoč brojnim mogućnostima koje XSLT posjeduje, poput pretvaranja XML dokumenata u formate poput PDF-a, njegova glavna svrha je upravljanje podacima za potrebe prikazivanja u mrežnim preglednicima.²⁵

XSLT se sastoji od brojnih elemenata, a svi elementi svrstani su u različite grupe s obzirom na njihovu funkciju. Neke od glavnih kategorija su: *Template Elements*, *Control Elements*, *Output Elements*, *Variable and Parameter Elements* i druge. Za potrebe ovog rada, поближе će se obraditi četiri XSLT elementa. Jedan od osnovnih elemenata je `<xsl:value-of select=“./“>`, a služi za prikazivanje sadržaja jednog XML elementa ili atributa. Vrlo sličan element prethodnom je `<xsl:for-each select=“./“>` koji služi za prikazivanje sadržaja elemenata koji se ponavljaju, a primjer oba elementa nalazi se na slici (Slika 4).²⁶

²⁴ Isto.

²⁵ Usp. Brčina, Josip. Primjena XSL-a u digitalnoj knjižnici hrvatske tiskane baštine Edicija : diplomski rad. Osijek: Filozofski fakultet, 2014., str. 16 URL: <https://repozitorij.ffos.hr/islandora/object/ffos%3A1497/datastream/PDF/view> (2024-08-20)

²⁶ Usp. The Extensible Stylesheet Language Family (XSL). URL: <https://www.w3.org/Style/XSL/> (2024-08-20)

```

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:template match="/">
<html>
<body>
<h2>Detalji o pjesmama</h2>
<xsl:for-each select="pjesma/popis_pjesama">
<hr/>
<p>Izvođač: <xsl:value-of select="izvodac" /></p>
<p>Album: <xsl:value-of select="album"/></p>
<p>Trajanje: <xsl:value-of select="trajanje" /></p>
<hr/>
</xsl:for-each>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Slika 4. Uporaba <xsl:for-each> i <xsl:value-of> elemenata.

Sljedeća grupa elemenata odnosi se na prikazivanje elemenata samo onda kada su određeni uvjeti ispunjeni. <xsl:if test="."> koristi se za prikazivanje elemenata kada je zadovoljen jedan određeni uvjet. Sljedeći element, <xsl:choose>, koristi se za prikazivanje svih elemenata kada se ispune višestruki uvjeti, a pruža i mogućnost grafičkog oblikovanja zapisa koji su zadovoljili neki od uvjeta.²⁷ Na primjeru vidljivom na slici (Slika 5) element <xsl:if> koristi se za prikaz samo studenata koji imaju ocjenu izvrstan (5), a korištenjem elementa <xsl:choose> svi će se studenti prikazati, ali će student s ocjenom 5 biti označen zelenom bojom.

²⁷ Usp. The Extensible Stylesheet Language Family (XSL). URL: <https://www.w3.org/Style/XSL/> (2024-08-20)


```

<body>
  <h2>Podaci o studentima</h2>

  <!-- Primjer korištenja <xsl:if> -->
  <h3>Studenti s najboljim ocjenama:</h3>
  <xsl:for-each select="fakultet/student">
    <xsl:if test="ocjena=5">
      <p><xsl:value-of select="ime"/> - Ocjena: <xsl:value-of select="ocjena"/></p>
    </xsl:if>
  </xsl:for-each>

  <!-- Primjer korištenja <xsl:choose> -->
  <h3>Pregled svih studenata:</h3>
  <xsl:for-each select="fakultet/student">
    <p>
      <xsl:value-of select="ime"/> - Ocjena: <xsl:value-of select="ocjena"/>
      <xsl:choose>
        <xsl:when test="ocjena=5">
          <strong style="color:green;">(Izvrstan)</strong>
        </xsl:when>
      </xsl:choose>
    </p>
  </xsl:for-each>
</body>

```

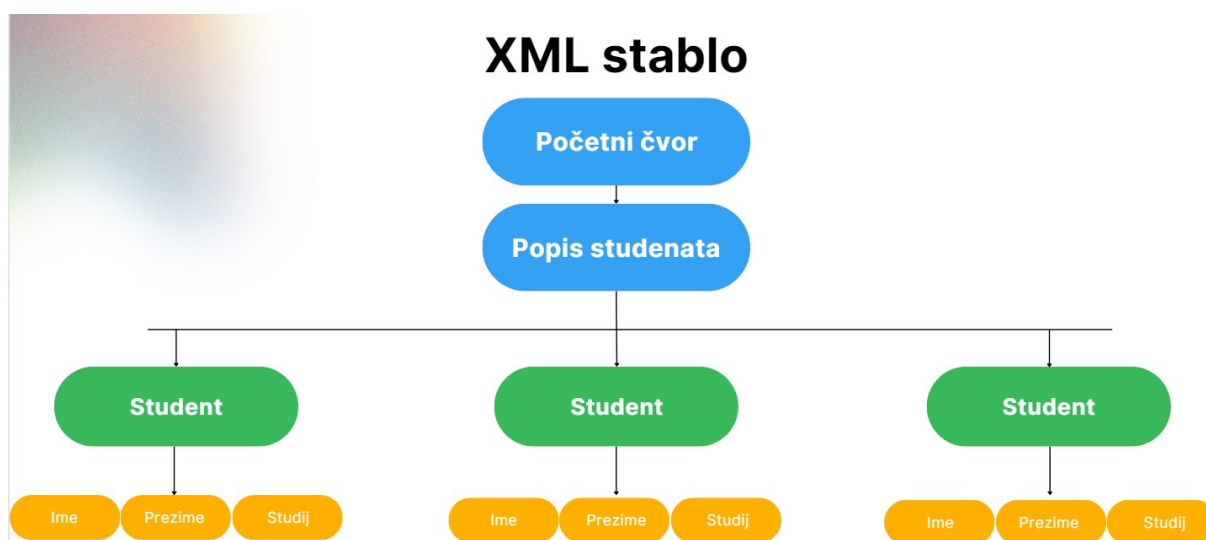
Slika 5. Uporaba <xsl:if> i <xsl:choose> elemenata.

Nakon obrade svih detalja i dijelova XSL-a, može se zaključiti kako je XSL neizostavan dio u radu s bilo kojim XML dokumentom. XSL se nametnuo kao glavni jezik za transformiranje XML dokumenata, što je bilo i očekivano s obzirom da mu je to bila glavna namjera od nastanka, a i sam je baziran na XML-u. Osim glavne namjere, transformiranja XML dokumenata, brojne su druge mogućnosti XSL-a što je samo dodatno učvrstilo njegovu poziciju na tržištu. Međutim, zbog opsega ovog rada, nismo u mogućnosti detaljno ih obraditi.

4. CSS vs. XSL

4.1. Liste stilova za XML

Svaki pravilno napisani XML dokument je u obliku stabla, XML stablo je zapravo struktura sastavljena od povezanih čvorova (engl. *node*), a na vrhu stabla nalazi se početni čvor (engl. *root node*). Čvorovi koji proizlaze iz početnog čvora nazivaju se potomci (engl. *child node*), a svaki od potomaka može imati svoje vlastite potomke i tako dalje. Najzanimljivija karakteristika strukture XML stabla je da svaki početni čvor sa svojim potomcima predstavlja zasebnu strukturu stabla, tako da je XML stablo hijerarhijska struktura sastavljena od stabala od kojih je svako zasebno stablo sastavljeno od više manjih stabala, kao što je to vidljivo na primjeru popisa studenata koji će se koristiti za usporedbu CSS i XSL prikaza, a koji je na Slici 6 prikazan u grafičkom obliku.²⁸



Slika 6. Hijerarhijska struktura XML dokumenta.

Iako je XML imao značajan utjecaj od početka te su mnogi odmah to i prepoznali, sa sobom je predstavio i nove probleme. Uzimajući u obzir da je XML i jezik za označavanje te njegovu karakterističnu strukturu stabla, za prikazivanje XML dokumenata potreban je određeni alat pomoću kojeg će se pružiti informacije o formatiranju, drugim riječima ubrzo se javila potreba za listama stilova.²⁹ Većina preglednika nije podržavala XML dokumente na način da prikaže njegovu strukturu stabla, nego su bili prikazani kao obični tekstualni dokumenti. Iz prethodnog poglavlja već se opisao XSL koji je napravljen isključivo za potrebe XML dokumenata, ali XSL

²⁸ Usp. Maleš, Lada. Osnove programiranja za web: (HTML, JavaScript, XML i XSL). Split: Filozofski fakultet Sveučilišta, 2007., str. 100-101

²⁹ Usp. Pitts-Moultis, Natanya. Osnove XML: brz i lak pristup ključnim vladajućim veštinama i tehnologijama. Čačak: Kompjuter biblioteka, 2000., str. 122

stupa na scenu tek dvije godine nakon XML-ovog postojanja. U to vrijeme CSS je već bio neko vrijeme na sceni te je imao prilično dobru podršku mrežnih preglednika.

Unatoč postojanju i korištenju CSS-a za XML dokumente, W3C je prepoznao kako CSS ne posjeduje brojne značajke koje su zapravo ključne za rad s XML dokumentima. Zbog tih nedostataka CSS-a, razvijen je XSL, ali unatoč pojavi jezika specifičnog za XML dokumente, mnogi su nastavili koristiti CSS za rad s XML dokumentima te se postavlja pitanje koji od ta dva jezika, CSS-a i XSL-a, je korisniji i bolji za rad s XML dokumentima.

U sljedećim poglavljima analizirat će se primjeri prikaza oglednog XML dokumenata prikazanog na Slici 7. pomoću CSS-a i XSL-a, a nakon toga izvršiti njihova usporedba.

4.2. CSS za prikaz XML dokumenata

Iako se CSS pokazao kao stabilan alat za rad s XML dokumentima, ima brojne nedostatke što nije iznenađujuće s obzirom na to da je izvorno namijenjen za rad s HTML-om, te nikada nije bio namijenjen za rad s XML-om.³⁰ Najveći nedostatak CSS-a je taj što ne podržava nikakve transformacije. Iako ga je moguće koristiti za druge procese, čak i tada će vjerojatno biti potrebno koristiti jednu od XSL verzija.³¹ Zašto onda mnogi i dalje više preferiraju rad s CSS-om nego s XSL-om? Ne postoji jednostavan odgovor na to pitanje, jer sve ovisi o tome na koje se sve načine želi manipulirati s XML dokumentima. No, vjerojatno i jednostavnost CSS-a ima velik utjecaj na izbor između ova dva jezika. Osim jednostavnosti i lakoće korištenja CSS-a, više mrežnih preglednika podržava CSS nego XSL.³²

Na sljedećem primjeru, prikazuje se stiliziranje navedenog primjera popisa studenata u XML-u koji je vidljiv na slici (Slika 7) pomoću CSS-a čiji je kôd vidljiv na Slici 8. Rezultat samog stiliziranja vidljiv je na Slici 9.

³⁰Usp. Gulbransen, David. Using XML [Google books]. Indianapolis, Ind.: Que Pub., 2002., str. 188

URL: <https://books.google.hr/books?id=aqfvLQzR-GkC&lpg=PA1&ots=-8sUc6tzjB&dq=Comparison%20of%20CSS%20and%20XSL%20for%20displaying%20XML%20documents&lr&hl=hr&pg=PA218#v=onepage&q=Comparison%20of%20CSS%20and%20XSL%20for%20displaying%20XML%20documents&f=false> (2024-08-23)

³¹ Usp. Graham, Tony. Formatting from XML. // XML Prague 2014: conference proceedings: University of Economics, Prague, Czech Republic, February 14-16, 2014. Czech Republic: Jiří Kosek, Oldřichov v Hájích, 2014 ; str. 266. URL: https://web.archive.org/web/20180721191520id_/http://archive.xmlprague.cz/2014/files/xmlprague-2014-proceedings.pdf#page=279 (2024-08-22)

³² Usp. Gulbransen, David. Using XML [Google books]. Indianapolis, Ind.: Que Pub., 2002., str. 174

URL: <https://books.google.hr/books?id=aqfvLQzR-GkC&lpg=PA1&ots=-8sUc6tzjB&dq=Comparison%20of%20CSS%20and%20XSL%20for%20displaying%20XML%20documents&lr&hl=hr&pg=PA218#v=onepage&q=Comparison%20of%20CSS%20and%20XSL%20for%20displaying%20XML%20documents&f=false> (2024-08-23)

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="style.css"?>
<popis_studenata>
  <student>
    <ime>Ivan</ime>
    <prezime>Horvat</prezime>
    <studij>Informatologija</studij>
  </student>
  <student>
    <ime>Ivana</ime>
    <prezime>Kovač</prezime>
    <studij>Pedagogija</studij>
  </student>
  <student>
    <ime>Marko</ime>
    <prezime>Kovačić</prezime>
    <studij>Psihologija</studij>
  </student>
</popis_studenata>

```

Slika 7. Popis studenata u XML obliku.

```

student {
  display: block;
  margin: 10px 0;
  padding: 10px;
  border: 1px solid #ccc;
  background-color: #f9f9f9;
}

ime {
  font-weight: bold;
}

studij {
  font-style: italic;
  color: #555;
}

```

Slika 8. CSS kôd za prikaz XML dokument u pregledniku.

Ivan Horvat <i>Informatologija</i>
Ivana Kovač <i>Pedagogija</i>
Marko Kovačić <i>Psihologija</i>

Slika 9. Prikaz XML dokumenta pomoću CSS-a na pregledniku.

4.3. XSL za prikaz XML dokumenata

Kao što je već objašnjeno u radu, XSL je jezik koji je nastao specifično za XML dokumente u svrhu prikazivanja njihovih podataka, a glavna svrha mu je omogućiti prikazivanje XML dokumenata u preglednicima. Svaki XML dokument kojeg je trebalo prikazati, bilo na ekranu, pisaču ili u drugim formatima, imao bi prateći XSL dokument.³³ XSL dokument, koristeći predloške, definirao je pravila i načine koji su određivali kako će se XML podaci transformirati čime se omogućilo prikazivanje strukture XML dokumenata, za razliku od ranijih prikaza kada su bili prikazivani kao obični tekstualni dokumenti. Iako je XSL nešto kompliciraniji od CSS-a, olakšao je transformacije XML dokumenata u razne formate, omogućio selektivno izdvajanje podataka iz samih XML dokumenata te omogućio manipulaciju XML elementima.³⁴ Fleksibilnost i moć XSL-a učinili su ga ključnim alatom u radu s XML dokumentima. Pritom je nadomjestio nedostatke CSS-a koji je u brojnim slučajevima doveo do ozbiljnih problema pri prikazivanju XML dokumenata na preglednicima.

Na primjeru vidljivom na Slici 10. elementi `<xsl:value-of>` i `<xsl:for-each>` koriste se kako bi se prikazale informacije svakog studenta iz XML datoteke. U XSL datoteku dodatno je ubačen sadržaj formatiran HTML elementima (u H2 tagu naslov 'Informacije o studentu') koji podatke u XML-u obogaćuje dodatnim značenjima.

³³ Usp. Ciancarini, Paolo; Vitali, Fabio; Mascolo, Cecilia. Managing complex documents over the WWW: a case study for XML. // IEEE Transactions on Knowledge and Data Engineering 4, 11 (1999), str. 6 URL: https://www.researchgate.net/publication/3296828_Managing_complex_documents_over_the_WWW_A_case_study_for_XML (2024-08-22)

³⁴ Usp. Cole, Timothy W. ... [et. al.]. XML Technologies for the Digital Library, 1997. URL: https://www.academia.edu/74772015/XML_technologies_for_the_digital_library?uc-sb-sw=93516020 (2024-08-24)

```

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:template match="/">
<html>
<body>
<h2>Informacije o studentu</h2>
<xsl:for-each select="popis_studenata/student">
<hr/>
<p>Ime: <xsl:value-of select="ime" /></p>
<p>Prezime: <xsl:value-of select="prezime"/></p>
<p>Studij: <xsl:value-of select="studij" /></p>
<hr/>
</xsl:for-each>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Slika 10. XSL kôd za prikaz XML dokumenta u pregledniku.

Kako to izgleda u pregledniku prikazano je na Slici 11.

Informacije o studentu

Ime: Ivan

Prezime: Horvat

Studij: Informatologija

Ime: Ivana

Prezime: Kovač

Studij: Pedagogija

Ime: Marko

Prezime: Kovačić

Studij: Psihologija

Slika 11. Prikaz XML dokumenta pomoću XSL-a u pregledniku.

4.4. CSS vs. XSL

Dakle, postoje dvije ključne razlike između XSL-a i CSS-a: CSS se može koristiti za uređivanje HTML dokumenata (što mu je i glavna svrha), a XSL se koristi isključivo za XML dokumente. Druga razlika je ta što XSL ima mogućnost transformiranja XML dokumenata, a CSS nema tu

moгуćnost. Taj nedostatak moгуćnosti transformiranja XML dokumenata zapravo je ključan, jer ako se koristi isključivo CSS u radu s XML dokumentima, transformacija nije moguća. Vrlo je očit nedostatak CSS-a taj što je on isključivo jezik za oblikovanje, što ne predstavlja problem za dodjelu stilskih svojstava XML ili HTML elementima, ali predstavlja problem kada je cilj prikazivanje XML dokumenata. Osim nedostatka moгуćnosti transformacije, CSS ima još nekoliko značajnih ograničenja koja uvelike utječu na rad s XML dokumentima: ne 'shvaća' odnose među čvorovima u dokumentu, ne može pohraniti vrijednosti u varijablama, ne može ponovno upotrijebiti dio podataka iz dokumenta na drugom mjestu, te ne podržava strukture odlučivanja.³⁵ XSL je svakako moćniji od CSS-a kada se uzmu u obzir sve njegove moгуćnosti, ali te dodatne i moćnije moгуćnosti XSL-a također imaju nedostatak, a to je njihova složenost. Moгуćnosti stiliziranja u XSL-u su slične onima u CSS-u, pa se složenost XSL-a uglavnom odnosi na njegove sposobnosti transformiranja XML dokumenta.

Kada se CSS koristi u radu s XML dokumentima, kao osnovu za primjenu stilskih pravila koristi se struktura XML stabla što može biti korisno za određene aplikacije. Postoje aplikacije kojima odgovara stiliziranje dokumenta unutar njegove već postojeće strukture, a određene aplikacije imat će potrebu transformirati sadržaj XML dokumenta prije nego što se krene s procesom stiliziranja i prikazivanja. Ako je nužna transformacija podataka, CSS to ne može omogućiti i tada će se koristiti XSL. CSS-ova nemogućnost transformiranja ga ne čini beskorisnim, on je odličan izbor ako je cilj jednostavan prikaz XML dokumenata bez potrebe za mijenjanjem postojeće strukture. Također, CSS se može koristiti za stiliziranje nakon što je dokument transformiran putem XSL-a.³⁶

Na praktičnom primjeru XML dokumenta s popisom studenata, napravljena je usporedba moгуćnosti CSS-a i XSL-a. Prikaz XML dokumenta u pregledniku koji je označen pomoću CSS-a vidljiv je na Slici 9., a XSL-a na Slici 11.

Pitanje koji jezik koristiti u radu s XML dokumentima ostaje, ali je i pitanje postoji li odgovor na takvo pitanje. I CSS i XSL imaju svoje nedostatke, ali kada se konkretno govori o XML dokumentima, XSL se čini boljim izborom zbog svojih naprednih moгуćnosti. U konačnici sve ovisi o krajnjem cilju, npr. ako postoji potreba za transformacijom XML dokumenta u drugi format, CSS nije opcija, pa je u tom slučaju XSL jedini izbor. Ako je potrebna manipulacija XML-om i njegovom strukturom, također je XSL jedini izbor. Ali ako su cilj jednostavnije promjene u

³⁵ Usp. Morrison, Michael. Comparing XSL and CSS // XML Unleashed. // O'Reilly, 1999., URL: https://learning.oreilly.com/library/view/xml-unleashed/0672315149/0672315149_ch08lev1sec3.html (2024-08-24)

³⁶ Isto.

oblikovanju ili dizajnu, moguće je koristiti i XSL i CSS, no, u tom slučaju, zbog svoje jednostavnosti, većina će se okrenuti CSS-u.³⁷ Međutim, CSS i XSL nisu u međusobnom sukobu te ih je moguće koristiti u kombinaciji. Moguće je pomoću XSL-a pretvoriti XML dokument u HTML dokument te nakon toga koristiti CSS za stiliziranje, moguće je XSL-om transformirati postojeće XML dokumente u XML dokumente koji su stilizirani CSS-om itd.³⁸ I CSS i XSL imaju svoj značaj u radu s XML dokumentima, ali s obzirom na njihove značajke, XSL se postavio kao glavni alat za rad s XML dokumentima, dok je CSS pomoćni alat za stiliziranje.

³⁷ Isto.

³⁸ Isto.

5. ZAKLJUČAK

Nakon detaljne obrade CSS-a i XSL-a može se zaključiti kako oba jezika imaju ključnu ulogu u oblikovanju weba te da svaki ima svoju svrhu i specifičnosti. CSS je uistinu promijenio način stiliziranja mrežnih stranica omogućujući dizajnerima i programerima precizno definiranje izgleda mrežnih stranica. Uvođenjem CSS-a omogućila se izrada fleksibilnih i dinamičnih mrežnih stranica čime je CSS postao nezamjenjiv alat u modernom web dizajnu. S obzirom na to da je kreiran za rad s HTML-om, CSS-u nedostaju brojne značajke kada ga se koristi za rad s XML dokumentima, iako ga je moguće samostalno koristiti u jednostavnijim procesima. Glavni nedostatak korištenja CSS-a u radu s XML dokumentima odnosi se na nemogućnost transformacije dokumenata. CSS nije dizajniran za manipuliranje sadržajem XML dokumenata što ga čini neprikladnim izborom za složenije operacije koje zahtijevaju manipuliranje i transformacije XML dokumenata. Tomu služi upravo XSL koji pruža napredne mogućnosti transformacije i manipuliranja XML podacima. XSL se sastoji od nekoliko dijelova, uključujući XSLT koji omogućuje transformacije XML dokumenata u druge formate, XPath i XSL-FO. XSL se nametnuo kao nezamjenjiv alat u radu s XML dokumentima zbog svojih naprednih mogućnosti koje su potrebne u složenijim operacijama nad XML strukturama. No uspoređujući XSL i CSS, CSS se ističe kao jednostavniji za učenje, zbog čega ga mnogi preferiraju. Unatoč raznim mogućnostima CSS-a, za brojne operacije jedini izbor je XSL. Kroz navedene praktične primjere u radu, opisani su načini rada s XML dokumentima koristeći i CSS i XSL. Konačni izbor između CSS-a i XSL-a ne ovisi toliko o njihovim tehničkim značajkama koliko ovisi o specifičnim potrebama i ciljevima određenog projekta. Svaki od njih nudi jedinstvene prednosti, ali ne isključuju jedan drugog tako da ih je moguće koristiti i u kombinaciji koja je često optimalno rješenje. Često će se najbolji rezultati ostvariti njihovom kombiniranom upotrebom, gdje će jedan drugog nadopunjavati na način koji najbolje odgovara zahtjevima projekta. U konačnici, CSS i XSL, svaki sa svojim jedinstvenim prednostima, nezamjenjivi su alati na današnjem tržištu, omogućujući stvaranje, transformaciju i stiliziranje funkcionalnog i vizualno bogatog mrežnog sadržaja.

POPIS LITERATURE

1. Brčina, Josip. Primjena XSL-a u digitalnoj knjižnici hrvatske tiskane baštine Edicija : diplomski rad. Osijek: Filozofski fakultet, 2014. URL: <https://repozitorij.ffos.hr/islandora/object/ffos%3A1497/datastream/PDF/view> (2024-08-20)
2. Ciancarini, Paolo; Vitali, Fabio; Mascolo, Cecilia. Managing complex documents over the WWW: a case study for XML. // IEEE Transactions on Knowledge and Data Engineering 4, 11 (1999), str. 1-12. URL: https://www.researchgate.net/publication/3296828_Managing_complex_documents_over_the_WWW_A_case_study_for_XML (2024-08-22)
3. Cole, Timothy W. ... [et. al.]. XML Technologies for the Digital Library, 1997. URL: https://www.academia.edu/74772015/XML_technologies_for_the_digital_library?uc-sb-sw=93516020 (2024-08-24)
4. CSS Snapshot 2023. URL: <https://www.w3.org/TR/CSS/#css> (2024-08-22)
5. Graham, Tony. Formatting from XML. // XML Prague 2014: conference proceedings: University of Economics, Prague, Czech Republic, February 14-16, 2014. Czech Republic: Jiří Kosek, Oldřichov v Hájích, 2014 ; str. 265-272. URL: https://web.archive.org/web/20180721191520id_/http://archive.xmlprague.cz/2014/files/xmlprague-2014-proceedings.pdf#page=279 (2024-08-22)
6. Gulbransen, David. Using XML [Google books]. Indianapolis, Ind.: Que Pub., 2002. URL: <https://books.google.hr/books?id=aqfvLQzR-GkC&lpg=PA1&ots=-8sUc6tzjB&dq=Comparison%20of%20CSS%20and%20XSL%20for%20displaying%20XML%20documents&lr&hl=hr&pg=PA218#v=onepage&q=Comparison%20of%20CSS%20and%20XSL%20for%20displaying%20XML%20documents&f=false> (2024-08-23)
7. Maleš, Lada. Osnove programiranja za web: (HTML, JavaScript, XML i XSL). Split: Filozofski fakultet Sveučilišta, 2007.
8. Morrison, Michael. Comparing XSL and CSS // XML Unleashed. // O'Reilly, 1999., URL: https://learning.oreilly.com/library/view/xml-unleashed/0672315149/0672315149_ch08lev1sec3.html (2024-08-24)
9. Pitts-Moultis, Natanya. Osnove XML: brz i lak pristup ključnim vladajućim vještinama i tehnologijama. Čačak: Kompjuter biblioteka, 2000.

10. Schmidt, Desmond. The Role of Markup in the Digital Humanities. // Historical Social Research 3 (141), 37 (2012), str. 125-146. URL: <http://www.jstor.org/stable/41636601> (2024-08-23)
11. Sheldon, Robert. XSL (Extensible Stylesheet Language). URL: <https://www.techtarget.com/whatis/definition/XSL-Extensible-Stylesheet-Language> (2024-08-23)
12. The Extensible Stylesheet Language Family (XSL). URL: <https://www.w3.org/Style/XSL/> (2024-08-20)
13. Vlašić, Goran. CSS. Zagreb: Algebra, 2016.