

Formati za razmjenu podataka u mrežnom okruženju

Mataija, Katarina

Undergraduate thesis / Završni rad

2021

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:142:617277>

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

Download date / Datum preuzimanja: **2024-12-27**



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

Preddiplomski studij informatologije

Katarina Mataija

Formati za razmjenu podataka u mrežnom okruženju

Završni rad

Mentor: izv. prof. dr. sc. Boris Bosančić

Osijek, 2021.

Sveučilište J.J. Strossmayera u Osijeku
Filozofski fakultet Osijek
Odsjek za informacijske znanosti
Preddiplomski studij informatologije

Katarina Mataija

Formati za razmjenu podataka u mrežnom okruženju

Završni rad

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

Mentor: izv. prof. dr. sc. Boris Bosančić

Osijek, 2021.

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

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

IZJAVA

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

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

U Osijeku, datum
2. rujna 2024.

Katerina Matija, 0122226015
ime i prezime studenta, JMBAG

Sažetak

Rad se bavi temom formata za razmjenu podataka u mrežnom okruženju. Cilj ovoga rada jest predstaviti dva najznačajnija i najpopularnija formata za razmjenu podataka: XML (*Extensible Markup Language*) i JSON (*JavaScript Object Notation*) te usporediti njihove značajke, odnosno prednosti i nedostatke. U radu su najprije naznačene definicije te same karakteristike oba formata. Zatim je prikazan i kraći pregled povijesnoga razvoja oba formata. Također, opisani su i osnovni dijelovi XML i JSON formata te njihova sintaksa i struktura, uključujući DTD, XML Schema-u te JSON Schema-u. Ujedno, prikazani su i primjeri kako bi se lakše razumjela struktura te sintaksa XML i JSON dokumenata. Na koncu, formati su uspoređeni na temelju opisanih karakteristika koje iskazuju prednosti i nedostatke svakog formata.

Ključne riječi: formati podataka, XML, JSON, razmjena podataka.

Sadržaj

1. Uvod	1
2. XML	2
2.1. Povijest XML-a	2
2.2. Sintaksa XML-a.....	4
2.3. Struktura XML dokumenta.....	6
2.3.1 DTD	6
2.3.2. XML Schema	7
3. JSON	9
3.1. Povijest JSON-a.....	9
3.2. Sintaksa JSON-a.....	10
3.3. JSON Schema	11
4. Usporedba XML-a i JSON-a.....	12
5. Zaključak.....	14
Popis literature.....	15

1. Uvod

Svijet se preselio iz industrijskoga u informacijsko doba i praktički svaki aspekt našeg života je postao digitaliziran. Kombinacija tehnologije i digitalizacije navela je mnoge industrije poput bankarstva ili e-trgovina na stvaranje i razmjenu ogromnih količina korisnih podataka. U posljednje dvije godine generirano je 90% svih podataka na svijetu.¹

Format za razmjenu podataka je tekstualni format koji se upotrebljava za razmjenu podataka među platformama. Brojni formati za razmjenu podataka podupiru opise raznih vrsta podataka, a posjeduju sposobnost prilagodbe ovisno o potrebama opisa. Kako bi razmjena podataka bila moguća, neophodno je poznavanje sintakse formata u kojemu su podaci pohranjeni te korištena shema za njihov opis. Dakle, ključno je razumjeti kako su podaci pohranjeni te koja je sintaksa upotrijebljena i hoće li ju sustav koji koristimo razumjeti.

Razmjena podataka između različitih uređaja i aplikacija danas je postala nužna. Podaci se više ne pohranjuju lokalno na uređaju, već u oblaku. Za komunikaciju s oblakom i razmjenu podataka koriste se web usluge. Kako bi se komunikacija na različitim uređajima i platformama dosljedno provodila, podatke je potrebno formatirati standardnim formatima podataka, kao što su JSON ili XML. Navedena dva formata u ovom trenutku predstavljaju najzastupljenije formate za pohranu i posebice razmjenu podataka. XML i JSON su formati stvoreni ne tako davno, nakon što je informacijska tehnologija umnogome evoluirala, stoga su primjereni zapisivanju većih količina podataka.

Svrha ovoga završnoga rada jest prikaz formata za razmjenu podataka u mrežnom okruženju. U skladu sa svrhom rada, ciljevi rada se odnose na prikaz najzastupljenijih formata za razmjenu podataka u mrežnom okruženju, dakle, XML-a i JSON-a, prikaz strukture i sintakse oba formata, kao i na njihovu usporedbu. U skladu sa svrhom i ciljevima rada, u prvome poglavlju govori se o XML-u, povijesnom razvoju ovoga formata, sintaksi te strukturi formata, uključujući DTD i W3C XML Schemu. Drugo poglavlje je posvećeno JSON formatu za razmjenu podataka, njegovom povijesnom razvoju te sintaksi ovoga formata. Također, navedena je i JSON Schema koja se temelji na JSON-u te definira strukturu podataka sustava za provjeru valjanosti. Na koncu, treće poglavlje je posvećeno usporedbi ova dva formata za razmjenu podataka u mrežnom okruženju te će se navesti prednosti i nedostaci oba formata.

¹ Goyal, Gaurav; Singh, Karanjit; Ramkumar, K. R. A detailed analysis of data consistency concepts in data exchange formats (JSON & XML).// International Conference on Computing, Communication and Automation, 2017. Str. 72. URL: <https://ieeexplore.ieee.org/document/8229774> (2021-08-12)

2. XML

XML (*eXtensible Markup Language*) je inovativan format stvoren za pohranu, označavanje te razmjenu podataka u mrežnom okruženju; u prijevodu, proširivi jezik za označavanje kojemu je cilj jednostavniji i pregledniji opis podataka korištenih u mrežnom okruženju. Zbog svoje specifičnosti i mogućnosti prilagodbe, XML se danas koristi u razne svrhe, tj. za pohranu i razmjenjivanje raznih vrsta informacija. Neophodno je napomenuti da se XML, za razliku od brojnih prethodnika, može s lakoćom savladati te se može pokrenuti na bilo kojem operativnom sustavu.² Pored toga, XML se također može koristiti za kodiranje i postavljanje upita u bazama podataka što omogućuje različite prikaze istog sadržaja te odvajanje sadržaja od njegovog prikaza. Zbog toga je XML kompatibilan i s upitnim jezicima kao što je SQL (*Structured Query Language*).³

Nadalje, XML karakteriziraju i određena svojstva po kojima se razlikuje od drugih označiteljskih jezika. Primjerice, XML razlikuje koncepte semantičke valjanosti i sintaktičke ispravnosti s obzirom na pripadnu XML Schemu, zatim, ne ovisi ni o hardveru ni o softveru te ne provodi proceduralno označavanje teksta, već opisno.⁴ S obzirom da se XML može koristiti za oblikovanje teksta, ali i za strukturiranje temeljnih podataka, jedan je od ciljeva XML-a da njegovi dokumenti budu ljudski čitljivi, sažeti, formalni te jednostavni za kreiranje i obradu. Isto tako, XML bi trebao osigurati podršku radu različitih aplikacija te biti interoperabilan sa svojim prethodnikom - SGML-om.⁵

2.1. Povijest XML-a

XML je, dakle, 'potomak' ili nasljednik SGML-a (*Standard Generalized Markup Language*), strukturnog i semantičkog jezika za označavanje tekstualnih dokumenata. Sedamdesetih godina prošloga stoljeća, Charles F. Goldfarb, Ed Mosher i Ray Lorie u IBM-u su izumili označiteljski jezik koji je na koncu dobio naziv SGML. Razvijanje ovoga jezika od strane nekoliko stotina ljudi diljem svijeta potrajalo je sve do 1986. godine kada je konačno usvojen standard ISO 8879. U

² Harold, E. R.; Means, W. S. XML in a nutshell : a desktop quick reference. 2nd edition. Sebastopol: O'Reilly, 2002. Str. 3.

³ Bosančić, Boris. Označavanje teksta starih knjiga na hrvatskom jeziku pomoću TEI standarda [doktorski rad]. Zadar: Odjel za informacijske znanosti, 2011. Str. 96-98. URL: <https://www.bib.irb.hr/561774> (2021-07-09)

⁴ A Gentle Introduction to XML. URL: <https://tei-c.org/release/doc/tei-p5-doc/en/html/SG.html> (2021-07-02)

⁵ Blyth, A.; Cunliffe, D.; Sutherland, I. Security analysis of XML usage and XML parsing. // Computer & Security 22, 6(2003), str. 494-495. URL: <https://www.sciencedirect.com/science/article/pii/S0167404803006072?via%3Dihub> (2021-07-02)

temeljnom smislu, SGML i XML nisu bili predviđeni da postanu označiteljski jezici sa svojim, točno određenim, skupom elemenata, već označiteljski standardi koji bi preko svojih pravila propisivali kako kreirati pojedini označiteljski jezik. Dakle, XML-om se željela smanjiti komunikacija između kreatora XML dokumenata i programera koji programiraju programe za obradu istih.

Najuspješnija primjena SGML-a jest HTML (*Hyper Text Markup Language*), označiteljski jezik baziran na SGML-u koji se počeo koristiti kao temeljni format mrežnih stranica. Problem SGML-a je što je vrlo kompliciran i složen da ga gotovo niti jedan softver nikada nije u potpunosti implementirao.⁶ Uspoređujući HTML s XML-om, cilj HTML-a odnosi se na prikazivanje teksta i multimedije, a cilj XML-a jest opisivanje podataka. Dakle, XML pruža informacije o podacima koji su zapisani unutar oznaka proizvoljnih elemenata, a HTML pruža informacije o tekstu i formatu u okviru svojih vlastitih oznaka HTML elemenata. Prema tome, XML ne pruža unaprijed definirane oznake kao što to čini HTML.⁷ Razlika između HTML-a i XML-a ogleda se i u tomu što je HTML specijalizirani jezik za označavanje dijelova dokumenata u svrhu hijerarhijskoga prikaza te je nepraktičan za pohranu i razmjenu strukturiranih podataka, a XML je jezik za metaoznakačivanje te podržava hijerarhijsku strukturu podataka.⁸ XML dokument, za razliku od HTML-a, obrađuje softver pod nazivom XML Parser koji provjerava XML dokumente prema strogim sintaktičkim pravilima na kojima se XML zasniva. Naime, ključno je naglasiti da XML Parseri mogu obrađivati XML dokument i bez pripadnog DTD-ja ili XML Scheme. Ta činjenica je uzrokovala veoma stroga sintaktička pravila, na primjer, u XML dokumentu nije dozvoljeno izostaviti završnu oznaku elementa itd.⁹

Jon Bosak, Tim Bray, C. M. Sperberg-McQueen, James Clark i nekolicina znanstvenika 1996. godine započela je rad na takozvanoj „lite“ verziji SGML-a pritom smanjujući značajke koje su se pokazale suvišnima i složenima za implementaciju te bile nekorisne i zbunjujuće za krajnje korisnike tijekom prethodnih godina iskustva sa SGML-om. Rezultat njihova uloženog truda bio je XML 1.0 koji je nastao u veljači 1998. godine. Uspjeh novog formata bio je trenutačan. Programeri kojima je nedostajao strukturni označiteljski jezik putem kojeg bi razmjenjivali

⁶ Harold, E. R.; Means, W. S. Nav.dj., str. 8-10.

⁷ Introduction to XML. URL: https://www.w3schools.com/xml/xml_what_is.asp (2021-07-03)

⁸ Pitts-Moultis, Natanya. Osnove XML : [brz i lak pristup ključnim vladajućim vještinama i tehnologijama]. Čačak: Kompjuter biblioteka, 2000. Str. 9.

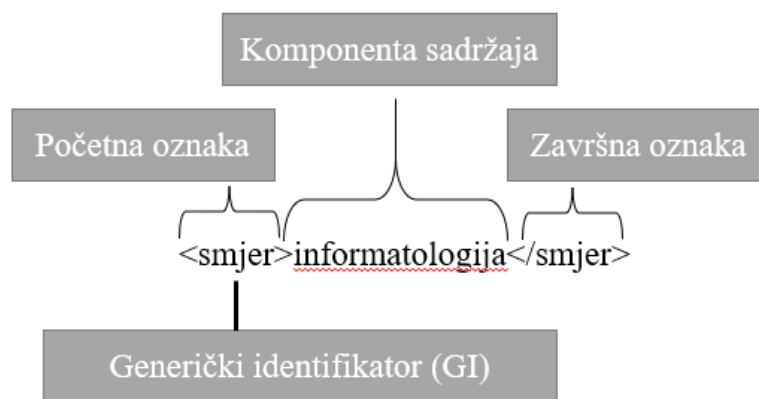
⁹ Bosančić, Boris. Uloga opisnih označiteljskih jezika u razvoju digitalne humanistike. // Libellarium, 4 (1), 2011, str. 76-77. URL: https://hrcak.srce.hr/index.php?show=clanak&id_clanak_jezik=136172 (2021-07-03)

podatke, odmah su ga usvojili, a s obzirom da je SGML bio složen jezik koji nije bio predviđen za takvo što.¹⁰

Međutim, XML 1.0 bio je samo početak, razvoj na XML-u se nastavio, da bi konačna XML specifikacija danas imala dvije verzije: XML 1.0 i XML 1.1. Kao što je već rečeno, XML 1.0 verzija nastala je 1998. godine, a svoje peto izdanje bilježi 2008. godine. XML 1.1 verzija objavljena je prvi put 2004. godine, a drugo i posljednje izdanje objavljeno je 2006. godine. Razlika između ove dvije verzije XML-a sastoji u ovisnosti odnosno neovisnosti o Unicode-u, standardu za predstavljanje simbola i znakova većine pisama svijeta u digitalnom okruženju. Naime, XML 1.1 verzija ne ovisi o trenutnoj inačici Unicode-a, već uvijek koristi njegovu najnoviju inačicu.¹¹

2.2. Sintaksa XML-a

Posebna namjena XML-a ogleda se u mogućnosti strukturiranja dokumenta po odvojenim cjelinama kao što su npr. naslov, poglavlje i sl. Kao što je već spomenuto, XML ne posjeduje predefinirane oznake (engl. *tag*), već njih stvara kreator dokumenta.



Slika 1. XML element.

Svaki element se sastoji od početne (engl. *start tag*) i završne oznake (engl. *end tag*) koje se nalaze između znaka manje (<) i znaka veće (>) te ti znakovi zajedno predstavljaju generički identifikator,

¹⁰ Harold, E. R.; Means, W. S. Nav.dj., str. 8-10.

¹¹ Bosančić, Boris. Uloga opisnih označiteljskih jezika u razvoju digitalne humanistike. // Libellarium, 4 (1), 2011, str. 76-77. URL: https://hrcak.srce.hr/index.php?show=clanak&id_clanak_jezik=136172 (2021-07-03)

što je moguće vidjeti na Slici 1. Unutar početnih i završnih oznaka smješten je pripadajući podatak, vrijednost ili tekst XML elementa.¹²

```
1  <?xml version="1.0" encoding="utf-8" ?>
2
3  <student>
4    <ime_prezime>Ana Anić</ime_prezime>
5    <kontakt>aanic@ffos.hr</kontakt>
6    <studij godina="3">preddiplomski</studij>
7    <smjer>informatologija</smjer>
8  </student>
9
```

Slika 2. Primjer XML zapisa.

Pri imenovanju XML elemenata neophodno je poštivati definirana pravila. Prvo pravilo određuje da naziv XML elementa može obuhvaćati slova, brojeve i druge znakove, ali ne može započeti znakom zarez, brojem ili slovima *xml*. Također, nazivi XML elemenata ne smiju sadržavati ni prazne prostore, a znakove poput crtice (–) i točke (.) uobičajeno je izbjegavati. Za potrebe korištenja dvije ili više riječi u nazivu XML elementa zbog opisnog karaktera XML-a, preporuča se korištenje donje crte (_) kao zamjene za razmak između riječi, što je vidljivo na Slici 2. na kojoj je drugi element odvojen donjom crtom.

Nadalje, XML elementi mogu sadržavati atribute, a atributi su predviđeni da sadrže podatke koji se vežu uz određeni element, te njihova vrijednost uvijek mora biti naznačena unutar navodnih znakova ili jednostrukih navodnika. Elementi se koriste za označavanje osnovnog teksta, dok se atributi većinom upotrebljavaju za označavanje metapodataka. Prilikom korištenja atributa, treba uzeti u obzir da atributi ne mogu sadržavati strukturu stabla, odnosno da se ne mogu ugnježdjivati jedan u drugi. Također, atributi ne mogu sadržavati višestruke vrijednosti, dok elementi mogu, jer se mogu ponavljati, pa se može zaključiti da atributi nisu prilagodljivi za buduće razmjene informacija. Svaki XML dokument trebao bi započeti XML deklaracijom, koja zapravo predstavlja instrukciju za obradu (engl. *processing instruction*), s informacijom o XML verziji dokumenta, samostalnim atributima i atributima kodiranja.¹³

¹² Bosančić, Boris. Uvod u XML. Označiteljski jezici za prikaz i opis sadržaja. Sveučilište J. J. Strossmayer, Filozofski fakultet, Odsjek za informacijske znanosti. Osijek, 2019 [predavanje]

¹³ Harold, E. R.; Means, W. S. Nav.dj., str. 21.

Svaki XML dokument, bez iznimke, mora biti dobro oblikovan. To znači da se mora pridržavati i određenih sintaktičkih pravila: prvo pravilo propisuje da svaka početna oznaka mora imati odgovarajuću krajnju oznaku. Nadalje, sintaktička pravila propisuju da svaki XML dokument mora i može imati samo jedan korijenski element u kojem su propisano ugniježđeni svi drugi elementi. Isto tako, pravila nalažu i da su XML oznake odnosno elementi osjetljivi na velika i mala slova. Konačno, posljednje pravilo propisuje da vrijednost atributa XML elementa mora biti navedena u navodnim znakovima i da element ne smije imati dva atributa istog naziva.¹⁴

2.3. Struktura XML dokumenta

XML dokument ispravno je napisan ili dobro oformljen ako je sintaksa u dokumentu pravilno napisana, dok je XML valjan dokument ako slijedi pravila određenog XML Schema jezika za opis strukture XML dokumenta. Dakle, XML Schema jezik definira koje elemente XML dokument može sadržavati te njihov broj i slijed.

Valjanost XML dokumenta u najvećem broju slučajeva testira se u odnosu na odgovarajući Document Type Definition (DTD) ili W3C XML Schemu. U zaglavlju XML dokumenta konstatira se koji je XML Schema jezik korišten za deklaraciju elemenata i atributa.¹⁵ DTD i W3C XML Schema su jezici putem kojih korisnici mogu izričito odrediti formate zapisa podataka i dokumenata. Dakle, DTD i W3C XML Schema su važni za strojnu obradu razmijenjenih podataka jer zahvaljujući njima programi mogu utvrditi je li primljena informacija u zadanom formatu i tako suzbiti eventualne pogreške.¹⁶

2.3.1 DTD

DTD (*Document Type Definition*) određuje strukturu XML dokumenta, dakle, koji se elementi i atributi mogu javiti u XML dokumentu i pod kojim uvjetima. To znači da kreiranje elemenata u XML dokumentu nije u potpunosti slobodno već se ravna prema odgovarajućem DTD-ju. Naime, XML dokument i DTD predstavljaju dva dijela jedne nerazdvojive cjeline. Nadalje, DTD se definira deklaracijama koje se mogu nalaziti ili unutar XML dokumenta ili biti pohranjene u zasebnom dokumentu. DTD se koristi kako bi neovisna grupa ljudi ili istraživača unutar jedne

¹⁴ Harold, E. R.; Means, W. S. Nav.dj., str. 23.

¹⁵ Chaudhri, Akmal B.; Rashid, Awais; Zicari, Roberto. XML data management: native XML and XML-enabled database systems. Boston: MA: Addison-Wesley, 2003. Str. 2.

¹⁶ Kirasić, D. XML tehnologija i primjena u sustavima procesne informatike. // Rijeka: Proceedings of the 28th International Convention MIPRO, 2005. Str. 85. URL: <https://www.bib.irb.hr/198358> (2021-07-11)

domene ili područja mogla dogovoriti odgovarajući standard (DTD dokument) prema kojemu bi razmjenjivali sadržaj u XML-u.¹⁷

```
1 <!ELEMENT student (ime_prezime, kontakt, studij, smjer)>
2 <!ELEMENT ime_prezime (#PCDATA)>
3 <!ELEMENT kontakt (#PCDATA)>
4 <!ELEMENT studij (#PCDATA)>
5 <!ELEMENT smjer (#PCDATA)>
```

Slika 3. Primjer DTD-ja.

Sa stajališta DTD-ja svaki XML dokument sastoji se od sljedećih tipova gradivnih blokova sadržaja: elemenata, atributa, entiteta, PCDATA te CDATA (Slika 3). Elementi su osnovni blokovi XML dokumenta te oni mogu sadržavati druge elemente, sadržaj ili biti prazni, a atributi sadrže dodatne informacije o elementima te se navode unutar početne oznake elementa. S druge strane, entiteti su varijable koje definiraju često korišteni tekst, odnosno predefimirane simbole s posebnim značenjem. PCDATA predstavlja vrstu sadržaja ili tekst XML datoteke u kojem XML *parser* interpretira znakove manje (<) i veće (>) kao dijelove XML oznaka. Suprotno tomu, CDATA predstavlja tekst koji XML *parser* neće parsirati. XML oznake koje *parser* registrira, tumačit će kao običan tekst, a ne kao XML oznake.¹⁸

Osnovni nedostatak DTD-ja je što sintaksna pravila kreiranja DTD dokumenta nisu usklađena sa sintaksom XML dokumenta. Zbog toga program koji opisuje XML dokumente mora ovladati parsiranjem pravila oba jezika. To znatno komplicira izgradnju takvih programa. Također i ljudi koji kreiraju DTD dokument moraju pored sintaktičkih pravila XML dokumenata savladati i sintaktička pravila za formiranje DTD dokumenata.

2.3.2. W3C XML Schema

XML Schema ili XSD (*XML Schema Definition*) predstavlja W3C preporuku (*W3C Recommendation*) web standarda iz 2001. godine. XML Schema opisuje strukturu XML dokumenta te definira elemente i attribute koji se mogu javiti u XML dokumentu. XML Schema definira i podelemente te njihov redoslijed i broj u nadređenom elementu, tipove podataka,

¹⁷ DTD Introduction. URL: https://www.w3schools.com/xml/xml_dtd_intro.asp (2021-07-04)

¹⁸ DTD Building blocks. URL: https://www.w3schools.com/xml/xml_dtd_building.asp (2021-07-04)

vrijednosti atributa i sadržaja elemenata, početne i fiksne vrijednosti sadržaja elemenata, kao i atributa, te definira je li neki element prazan ili pohranjuje nekakav sadržaj.¹⁹

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
3 targetNamespace="http://www.w3schools.com"
4 xmlns="http://www.w3schools.com"
5 elementFormDefault="qualified">
6 <xs:element name="student">
7 <xs:complexType>
8 <xs:sequence>
9 <xs:element name="ime_prezime" type="xs:string"/>
10 <xs:element name="kontakt" type="xs:string"/>
11 <xs:element name="studij" type="xs:string"/>
12 <xs:element name="smjer" type="xs:string"/>
13 </xs:sequence>
14 </xs:complexType>
15 </xs:element>
16 </xs:schema>
```

Slika 4. Primjer XML Schema dokumenta.

W3C XML Schema koristi slijedeće najčešće tipove podataka: niz znakova, cijeli broj, decimalni broj, datum, vrijeme i dr.²⁰ Na primjer, Slika 4 prikazuje XSD dokument u kojemu svi elementi imaju za vrstu teksta niz znakova (engl. *string*).

XML Schema opisuje strukturu XML dokumenta te se koristi kao zamjena za DTD. XML Schema je efikasnija i utjecajnija od DTD-ja, jer je XML Schema pisana u XML-u, te nije potrebno učiti novi jezik. 3C XML Schema ima identičnu zadaću kao i DTD, ali je detaljnija i suvremenija od DTD-ja. W3C XML Schema kao i DTD definira pravila i kriterije sadržaja u okviru strukture XML dokumenata, a to znači, definira pravila kojih se svaki kreator XML dokumenata mora pridržavati kako bi isti bio valjan. Također, propisuje i rječnik oznaka koje se mogu nalaziti u dokumentu. Iako je W3C XML Schema kompliciranija, ona kreatoru XML dokumenata pruža puno više mogućnosti. Na primjer, u W3C XML Schemi može se odrediti da li sadržaj nekog elementa mora biti broj, dok takva mogućnost ne postoji u DTD-ju.²¹

¹⁹ Bosančić, Boris. Uvod u XML Schemu i imenske prostore. Označiteljski jezici za prikaz i opis sadržaja. Sveučilište J. J. Strossmayer, Filozofski fakultet, Odsjek za informacijske znanosti. Osijek, 2019. [Predavanje]

²⁰ XSD Elements. URL: https://www.w3schools.com/xml/schema_simple.asp (2021-07-05)

²¹ Kirasić, D. Nav.dj., str. 85.

3. JSON

JSON (*JavaScript Object Notation*) je suvremeni format za razmjenu podataka u mrežnom okruženju koji se temelji na JavaScript programskom jeziku. Glavna karakteristika ovoga formata je jednostavnost čitanja i pisanja, ali i laka strojna čitljivost što ga čini sve popularnijim formatom.²² Popularnost je vidljiva u tome što je JSON trenutačno prevladavajući format za slanje zahtjeva Aplikacijskom programskom sučelju (*Application Programming Interface* - API) i dobivanje odgovora putem HTTP protokola. API je skup određenih specifikacija i pravila koje programeri slijede kako bi se mogli služiti resursima i uslugama određenog složenog programa ili operacijskog sustava. Osim toga, JSON se koristi i u sustavima NoSQL baza podataka.²³ Naime, baze podataka orijentirane na dokumente (eng. *document oriented databases*) su specijalizirane za pohranu JSON dokumenata a jedan od utjecajnijih primjera takve baze jest MongoDB.²⁴ Također, usvajanjem Ajax-a (*Asynchronous JavaScript and XML*) popularnost JSON-a počela je rasti, jer su korisnici započeli zamjećivati jednostavnost njegove implementacije u odnosu na XML.²⁵

JSON je potpuno zaseban standard, a najčešće se koristi zajedno s programskim jezicima poput PHP-a, Python-a, C#, C++ i Jave. JSON je formiran na dvjema strukturama: objektu i popisu vrijednosti. Objekt označava skup parova vrijednosti koji se mogu definirati nazivom, odnosno imenom. Popis vrijednosti se definira kao struktura koja sadrži individualne vrijednosti u određenom redosljedu.²⁶

3.1. Povijest JSON-a

Stvaranje JSON-a 2001. godine pripisuje se Douglasu Crockfordu. Iako, on priznaje da nije bio prvi koji je kreirao JSON, ipak je bio prvi koji ga je imenovao i formalizirao u okviru RFC 3627. Zatim je 2006. godine napisana formalizacija JSON-a u okviru RFC-a 4627 koja je uvela ekstenziju datoteke .json te definirala samu kompoziciju JSON-a. Naime, JSON je službeno priznat kao norma ECMA-404 tek u prosincu 2009. godine. *Internet Engineering Task Force* (IETF) je 2014. godine objavio vlastiti standard JSON-a - RFC 7159, kojim se sada nastoji očistiti izvorna specifikacija. Primarna razlika između ta dva standarda jest ta što standard ECMA-e upućuje na to da se valjani tekst JSON-a može pojaviti u obliku bilo koje vrijednosti JSON-a, dok

²² JSON. URL: <https://www.json.org/json-en.html> (2021-07-05)

²³ Bourhis, Pierre; Reutter, Juan L; Suárez, Fernando; Vrgoč, Domagoj. JSON: Data model, Query languages and Schema specification.// New York: Association for Computing Machinery, 2017. Str. 123. URL: <https://dl.acm.org/doi/10.1145/3034786.3056120> (2021-07-17)

²⁴ MongoDB. URL: <https://www.mongodb.com/> (2021-07-18)

²⁵ Smith, Ben. Beginning JSON. New york: Apress, 2015. Str. 110.

²⁶ JSON. URL: <https://www.json.org/json-en.html> (2021-07-05)

standard IETF-a navodi da valjani tekst JSON-a mora obuhvaćati sve valjane vrijednosti unutar početnoga niza ili objekta.²⁷

3.2. Sintaksa JSON-a

JSON dokumenti se temelje na tekstu te, kako je već navedeno, ne zahtijevaju veliki diskovni prostor za pohranu; osim toga, lako su čitljivi. Podaci su prikazani u obliku ključ/vrijednost parova, pa tako ključevi predstavljaju nizove (engl. *strings*), a vrijednost označava JSON oblike podataka. Ključ i vrijednost su međusobno odijeljeni dvotočkom, a uređeni parovi zareziima.²⁸

```
1  {"student":[
2  |         "ime_prezime":"Ana Anić",
3  |         "kontakt":"aanic@ffos.hr",
4  |         "studij":"preddiplomski",
5  |         "smjer":"informatologija",
6  |     ]}
```

Slika 5. Primjer JSON zapisa.

Na Slici 5. može se uočiti sintaksa JSON zapisa, ali i struktura koju obilježavaju objekti. Zapis se sastoji od četiri objekta, a svaki objekt se sastoji od jedne kombinacije ključ-vrijednost parova. Na primjer, "ime_prezime" je ključ koji je tekstualni podatak, a vrijednost je "Ana Anić", što je u ovom slučaju tekst.

Temeljni tipovi podataka su niz znakova (*string*), broj, objekt, lista vrijednosti (*arrays*), Booleove vrijednosti i *null* vrijednost. Takav model izravno nudi mogućnost hijerarhijskog modela podataka te se podudara s modelima zastupljenima u velikom broju suvremenih programskih jezika koji upravljaju podacima.²⁹ U nekim implementacijama JSON uvrštava i vrste podataka poput enkodiranih binarnih vrijednosti, uobičajenih izraza ili datuma objekta. Općenito, zbog jednostavnosti JSON-a, njegova uporaba zahtijeva određene kompromise kao što su 'efektivna propusnost' (engl. *bandwidth efficiency*) ili nelinearno dekodiranje.³⁰ JSON vrijednosti ne mogu

²⁷ Smith, Ben. Nav.dj., str.37-38.

²⁸ Edelman, Jason; Lowe, Scott S; Oswalt, Matt. Network Programmability and Automation. Boston: O'Reilly Media, 2018. Str. 169.

²⁹ Sriparasa, Sai. JavaScript and JSON Essentials. Olton: Packt Publishing, 2013. Str. 37-38. URL: <https://www-proquest-com.ezproxy.nsk.hr/legacydocview/EBC/1481127?accountid=47665> (2021-07-21)

³⁰ White, Alexei. JavaScript Programmer's Reference. Hoboken: John Wiley & Sons, 2009. Str. 555-556. URL: <https://www-proquest-com.ezproxy.nsk.hr/legacydocview/EBC/456275?accountid=47665> (2021-07-20)

biti jedna od sljedećih vrsta podataka: datum, funkcija ili nedefinirano.³¹ Važno je napomenuti da su strukture koje se koriste u JSON-u tzv. univerzalne podatkovne strukture, koje se koriste u većini programskih jezika i na većini platformi. Stoga je logično da se JSON temelji na ove dvije vrste podatkovnih struktura, kako bi svaki jezik koji podržava korištenje JSON-a mogao raščlaniti ili dekonstruirati prikazivanje JSON-a u objekte.³²

Pravila za sintaksu JSON-a temelje se na JavaScript sintaksi. Prvo pravilo nalaže da se podaci navode u kombinaciji ime-vrijednost i da se odvajaju zarezima. Zatim, postoji pravilo koje ključeve definira kao nizove znakova u navodnicima. Naposljetku, tu je i pravilo koje striktno propisuje da se u uglatim zagradama nalazi lista vrijednosti, a u vitičastim zagradama objekti.³³

3.3. JSON Schema

JSON Schema predstavlja rječnik za definiranje strukture podataka sustava za provjeru njegove valjanosti, dokumentiranja i kontrolu interakcije. JSON Schemom se osiguravaju pravila strukture JSON podataka koji se zahtijevaju određenom aplikacijom te način na koji se ti podaci mogu izmijeniti.³⁴ JSON Schema jedini je opći pokušaj definiranja jezika za propisivanje strukture JSON dokumenata. JSON Schema još uvijek nije norma, ali postoji rastući broj aplikacija koji ju podupiru te velik broj alata koji omogućuju provjeru dokumenata u odnosu na nju. Postojale su i druge alternative za provjeru JSON dokumenata, koje su se temeljile ili na samom JSON-u ili su dizajnirane s posebnim modelom.

Unatoč svim prednostima JSON Scheme, razvoj iste odvijao se prilično sporo. Jedno od pitanja koje je priječilo rašireno priznavanje JSON Scheme kao norme za JSON metapodatke jest nejasnoća njezine specifikacije. Nedostatak formalne definicije također je sprječavalo znanstvenu zajednicu da se znatnije uključi u razvoj JSON Scheme. Formalna specifikacija također bi pomogla u razvoju automatizacijskih alata za API-je. Budući da se većina koristi za prijenos podataka između web aplikacija, od ključne je važnosti razvoj učinkovitih algoritama za provjeru JSON Scheme. Stoga je važno razumjeti proračunsku složenost problema validacije JSON Scheme jer to daje dobru polazišnu točku za projektiranje učinkovitih algoritama validacije JSON dokumenata.³⁵

³¹ JSON Data Types. URL: https://www.w3schools.com/js/js_json_datatypes.asp (2021-07-21)

³² Jackson, Wallace. JSON Quick Syntax Reference. New York: Apress, 2016. Str. 17.

³³ JSON Syntax. URL: https://www.w3schools.com/js/js_json_syntax.asp (2021-07-18)

³⁴ JSON Schema. URL: <https://json-schema.org/> (2021-07-19)

³⁵ Peoza, Felipe et al. Foundations of JSON Schema. International World Wide Web Conferences Steering Committee, 2016. Str. 263-268. URL: <https://doi.org/10.1145/2872427.2883029> (2021-07-19)

4. Usporedba XML-a i JSON-a

Istražujući i proučavajući XML i JSON mogu se uočiti sve prednosti formata za razmjenu, zapis te obradu podataka. Naime, oba formata imaju jednostavnu i razumljivu sintaksu, odnosno strukturu, što ih čini i ljudski i strojno čitljivima. I XML i JSON nemaju unaprijed definiranu semantiku, već su otvoreni za slobodno kreiranje elemenata i ključeva. Pritom, u slučaju JSON-a semantiku zapisa određuju pojedinačni sustavi, dok XML posjeduje vlastite aplikacije koje propisuju semantičke elemente u ovisnosti o području u kojem se rabe (npr. MathML, MusicXML itd.). Na ovaj način se pojednostavljuje i skraćuje sam postupak bilježenja podataka u navedenim formatima, kao i njihova sintaktička pravila. U brojnim situacijama nije neophodno posjedovati određeno znanje o tome koja identifikacijska oznaka što predstavlja, doduše, iznimno je jednostavno identificiranje podataka, unatoč činjenici da su nazivi JSON ključeva i XML elemenata u većini slučajeva na prirodnom jeziku te im je značenje razumljivo. Pri složenijim slučajevima je neophodno proučiti specifikaciju koja propisuje vjerodostojan opseg polja.

Oba formata su stvorena s ciljem jednostavnijeg i bržeg prijenosa podataka i omogućavanja interoperabilnosti sustava. Također, neusklađenost sustava bi mogla učiniti podatke nepouzdanima, odnosno postoji mogućnost da šum u sustavu naruši konzistentnost podataka koje preuzima. Dobra strana je da XML i JSON ne ovise o jedinstvenoj vrsti softvera pa ih i ova činjenica čini optimalnim formatima za razmjenu podataka u mrežnom okruženju.

Mrežna sučelja u najvećem broju slučajeva primjenjuju jedan od dva spomenuta formata, a ponekad koriste i oba te su korisnici i sustavi naviknuti da rabe podatke zabilježene u ovim formatima. Također, korisnici su naviknuti da obrađuju podatke u JSON-u ili XML-u pomoću brojnih suvremenih tehnologija. S obzirom na brzinu prijenosa podataka, uspoređujući učinkovitost XML-a i JSON-a pri razmjeni podataka, ustanovljeno je da podaci zapisani u JSON formatu u prosjeku nadilaze brzinu prijenosa podataka zapisanih u XML-u. Naime, JSON za obradu upotrebljava korisnikov procesor u većoj mjeri nego što to čini XML. Stoga, pored različite sintakse, jedna od razlika JSON-a u odnosu na XML leži u činjenici da JSON rasterećuje poslužitelj, a opterećuje klijenta. U isto vrijeme, oba formata podjednako koriste memoriju.³⁶

³⁶ Nurseitov, Nurzhan; Paulson, Michael; Reynolds, Randall; Izurieta, Clemente. Comparison of JSON and XML Data Interchange Formats: A Case Study. // Caine, 2009. Str. 4. URL: <https://www.semanticscholar.org/paper/Comparison-of-JSON-and-XML-Data-Interchange-A-Case-Nurseitov-Paulson/84321e662b24363e032d680901627aa1bfd6088f> (2021-08-11)

Po pitanju razlike u strukturi između JSON-a i XML-a, JSON posjeduje jednostavnu strukturu što ga čini pogodnim za jednostavan prijenos podataka. S druge strane, jedna od glavnih prednosti XML-a je njegova fleksibilnost, s obzirom na mogućnost pohrane, u teoriji, svih mogućih vrsta podataka, za razliku od JSON-a čija je pohrana ograničena na uobičajene vrste podataka. Takva fleksibilnost očituje se u jednostavnosti JSON formata, no mnogo je teže raščlaniti XML format jer posjeduje strogu definiciju strukture sličnu stablu.³⁷

Nadalje, XML obrađuje podatke u tekstualnom formatu, dok JSON obrađuje podatke u mapiranom formatu te mu je struktura jednostavnija i razumljivija od strukture XML-a. Naime, JSON podaci se zapisuju u obliku brojeva, objekata, null oznaka, običnog sadržaja, Booleovih vrijednosti te nizova, dok se XML podaci zapisuju kao elementi koje je tek potrebno parsirati i za računalne potrebe pretvoriti u brojeve, običan sadržaj, Booleove oznake, null oznake i nizove prije same uporabe u aplikaciji. Jedna od prednosti XML-a je i velika dostupnost tehnologija koje se mogu koristiti za provjeru valjanosti XML dokumenta kao što su to, primjerice, W3C XML Schema i W3C XML Schema validatori. Iako su se posljednjih godina pojavile slične tehnologije za JSON format, kao što je npr. JSON Schema, raspon funkcija i dalje nije usporediv s onima koje posjeduje XML.³⁸ Osnovno ograničenje JSON Scheme i W3C XML Scheme leži u činjenici da oba formata jedino propisuju strukturu dokumenta, ali ne provjeravaju podatke pohranjene u toj strukturi. Podaci se smatraju vjerodostojnima samo ako se vremenom ne mijenjaju i ako se njihova točnost uvijek održava.³⁹

³⁷ Breje, A; Györödi, Robert; Györödi Cornelia; Zmaranda Doina; Pecherle, George. Comparative Study of Data Sending Methods for XML and JSON Models. // International Journal of Advanced Computer Science and Applications 12, 9(2018), str. 198-200. URL: <https://thesai.org/Publications/ViewPaper?Volume=9&Issue=12&Code=IJACSA&SerialNo=29> (2021-08-12)

³⁸ Isto.

³⁹ Goyal, Gaurav; Singh, Karanjit; Ramkumar, K. R. A detailed analysis of data consistency concepts in data exchange formats (JSON & XML).// International Conference on Computing, Communication and Automation, 2017. Str. 73. URL: <https://ieeexplore.ieee.org/document/8229774> (2021-08-12)

5. Zaključak

U ovome radu prikazani su formati za razmjenu podataka u mrežnom okruženju, XML i JSON. U prvome poglavlju govorilo se o XML-u te o povijesnom razvoju XML formata. Zatim je dan osvrt na sintaksu i strukturu XML formata za pohranu i razmjenu podataka. S obzirom da se valjanost XML dokumenata provjerava u odnosu na odgovarajuću W3C XML Schemu ili DTD, ti su jezici umnogome i objašnjeni u ovome radu. Nadalje, drugo poglavlje je bilo posvećeno JSON formatu za razmjenu podataka te povijesnom razvoju i sintaksi JSON-a. Također, bila je navedena i JSON Schema koja definira strukturu podataka sustava za provjeru valjanosti. Naposljetku, treće poglavlje rada bilo je posvećeno usporedbi XML-a i JSON-a, njihovim prednostima i nedostacima, kao i razlikama i sličnostima.

XML i JSON danas su najčešće upotrebljavani formati za razmjenu podataka u mrežnom okruženju. Ključan čimbenik tomu jest neovisnost oba formata o jedinstvenoj vrsti softvera. Iz gore navedene usporedbe jasno je da JSON i XML formati za razmjenu podataka u mrežnom okruženju imaju svoje vlastite prednosti i nedostatke te je također jasno da se oba formata mogu koristiti u skladu s potrebama pojedinca ili sustava. S obzirom na pohranu podataka, XML je u prednosti, jer je fleksibilniji po pitanju formata pohrane podataka, dok je JSON ograničen na uobičajene vrste podataka. JSON može izgledati kompleksno zbog svoje strukture, no ako je dokument pravilno napisan sa svim zagradama i zarezima, jednostavan je za čitanje i razumijevanje.

Znanstvenici smatraju da JSON neće u potpunosti zamijeniti XML kao format za razmjenu podataka u mrežnom okruženju. XML zbog svojih vrijednih značajki ima svoje mjesto u prijenosu podataka i provjeri valjanosti dokumenta. Također, složili su se kako je JSON prikladniji za razmjenu podataka i da bi se trebao koristiti umjesto XML-a pri razmjeni podataka između poslužitelja i mrežnih aplikacija.

Popis literature

1. A Gentle Introduction to XML. URL: <https://tei-c.org/release/doc/tei-p5-doc/en/html/SG.html> (2021-07-02)
2. Blyth, A.; Cunliffe, D.; Sutherland, I. Security analysis of XML usage and XML parsing. // *Computer & Security* 22, 6(2003), str. 494-495. URL: <https://www.sciencedirect.com/science/article/pii/S0167404803006072?via%3Dihub> (2021-07-02)
3. Bosančić, Boris. Označavanje teksta starih knjiga na hrvatskom jeziku pomoću TEI standarda [doktorski rad]. Zadar: Odjel za informacijske znanosti, 2011. Str. 96-98. URL: <https://www.bib.irb.hr/561774> (2021-07-09)
4. Bosančić, Boris. Uloga opisnih označiteljskih jezika u razvoju digitalne humanistike. // *Libellarium*, 4 (1), 2011, str. 65-8 URL: https://hrcak.srce.hr/index.php?show=clanak&id_clanak_jezik=136172 (2021-07-03)
5. Bosančić, Boris. Uvod u XML Schemu i imenske prostore. Označiteljski jezici za prikaz i opis sadržaja. Sveučilište J. J. Strossmayer, Filozofski fakultet, Odsjek za informacijske znanosti. Osijek, 2019. [Predavanje]
6. Bosančić, Boris. Uvod u XML. Označiteljski jezici za prikaz i opis sadržaja. Sveučilište J. J. Strossmayer, Filozofski fakultet, Odsjek za informacijske znanosti. Osijek, 2019 [predavanje]
7. Bourhis, Pierre; Reutter, Juan L; Suárez, Fernando; Vrgoč, Domagoj. JSON: Data model, Query languages and Schema specification.// New York: Association for Computing Machinery, 2017. Str. 123. URL: <https://dl.acm.org/doi/10.1145/3034786.3056120> (2021-07-17)
8. Breje, A.; Győrödi, Robert; Győrödi Cornelia; Zmaranda Doina; Pecherle, George. Comparative Study of Data Sending Methods for XML and JSON Models. // *International Journal of Advanced Computer Science and Applications* 12, 9(2018), str. 198-200. URL: <https://thesai.org/Publications/ViewPaper?Volume=9&Issue=12&Code=IJACSA&SerialNo=29> (2021-08-12)
9. Chaudhri, Akmal B.; Rashid, Awais; Zicari, Roberto. XML data management: native XML and XML-enabled database systems. Boston: MA: Addison-Wesley, 2003. Str. 2.
10. DTD Building blocks. URL: https://www.w3schools.com/xml/xml_dtd_building.asp (2021-07-04)

11. DTD Introduction. URL: https://www.w3schools.com/xml/xml_dtd_intro.asp (2021-07-04)
12. Edelman, Jason; Lowe, Scott S; Oswalt, Matt. Network Programmability and Automation. Boston: O'Reilly Media, 2018.
13. Goyal, Gaurav; Singh, Karanjit; Ramkumar, K. R. A detailed analysis of data consistency concepts in data exchange formats (JSON & XML).// International Conference on Computing, Communication and Automation, 2017. Str. 72-73. URL: <https://ieeexplore.ieee.org/document/8229774> (2021-08-12)
14. Harold, E. R.; Means, W. S. XML in a nutshell : a desktop quick reference. 2nd edition. Sebastopol: O'Reilly, 2002. Str. 3-23.
15. Introduction to XML. URL: https://www.w3schools.com/xml/xml_whatism.asp (2021-07-03)
16. Jackson, Wallace. JSON Quick Syntax Reference. New York: Apress, 2016.
17. JSON Data Types. URL: https://www.w3schools.com/js/js_json_datatypes.asp (2021-07-21)
18. JSON Schema. URL: <https://json-schema.org/> (2021-07-19)
19. JSON Syntax. URL: https://www.w3schools.com/js/js_json_syntax.asp (2021-07-18)
20. JSON. URL: <https://www.json.org/json-en.html> (2021-07-05)
21. Kirasić, D. XML tehnologija i primjena u sustavima procesne informatike.// Rijeka: Proceedings of the 28th International Convention MIPRO, 2005. Str. 85. URL: <https://www.bib.irb.hr/198358> (2021-07-11)
22. MongoDB. URL: <https://www.mongodb.com/> (2021-07-18)
23. Nurseitov, Nurzhan; Paulson, Michael; Reynolds, Randall; Izurieta, Clemente. Comparison of JSON and XML Data Interchange Formats: A Case Study. // Caine, 2009. Str. 4. URL: <https://www.semanticscholar.org/paper/Comparison-of-JSON-and-XML-Data-Interchange-A-Case-Nurseitov-Paulson/84321e662b24363e032d680901627aa1bfd6088f> (2021-08-11)
24. Peoza, Felipe et al. Foundations of JSON Schema. International World Wide Web Conferences Steering Committee, 2016. Str. 263-268. URL: <https://doi.org/10.1145/2872427.2883029> (2021-07-19)
25. Pitts-Moultis, Natanya. Osnove XML : [brz i lak pristup ključnim vladajućim veštinama i tehnologijama]. Čačak: Kompjuter biblioteka, 2000.
26. Smith, Ben. Beginning JSON. New york: Apress, 2015. Str. 37-110.

27. Sriparasa, Sai. JavaScript and JSON Essentials. Olton: Packt Publishing, 2013. Str. 37-38. URL: <https://www-proquest-com.ezproxy.nsk.hr/legacydocview/EBC/1481127?accountid=47665> (2021-07-21)
28. White, Alexei. JavaScript Programmer's Reference. Hoboken: John Wiley & Sons, 2009. Str. 555-556. URL: <https://www-proquest-com.ezproxy.nsk.hr/legacydocview/EBC/456275?accountid=47665> (2021-07-20)
29. XML Schema. URL: https://www.w3schools.com/xml/schema_intro.asp (2021-07-11)
30. XSD Elements. URL: https://www.w3schools.com/xml/schema_simple.asp (2021-07-05)