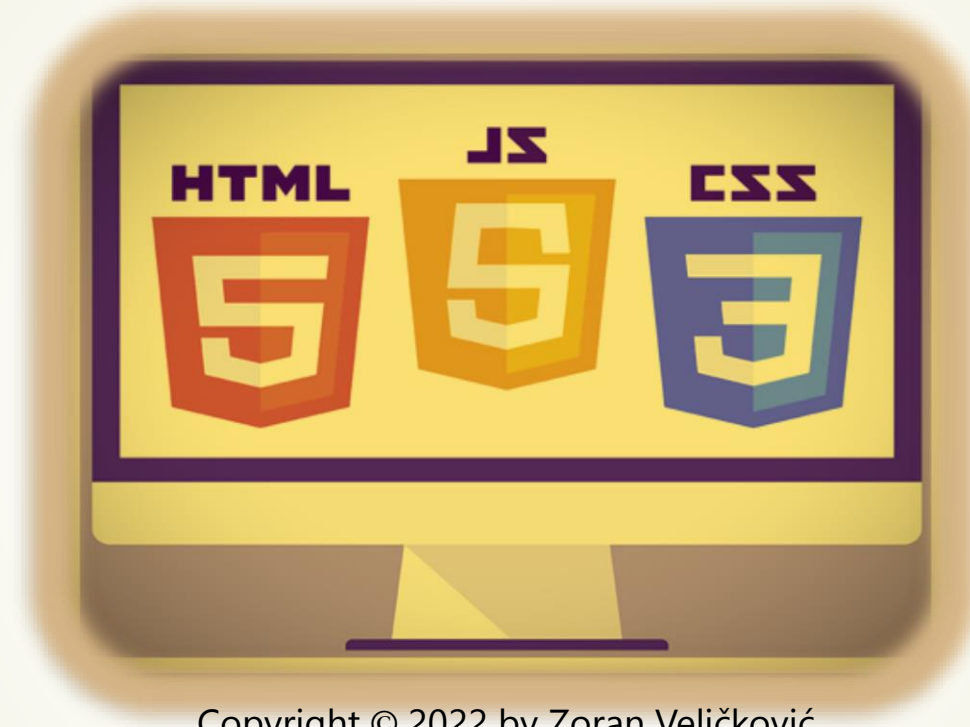




Akademija tehničko-vaspitačkih strukovnih studija



Copyright © 2022 by Zoran Veličković

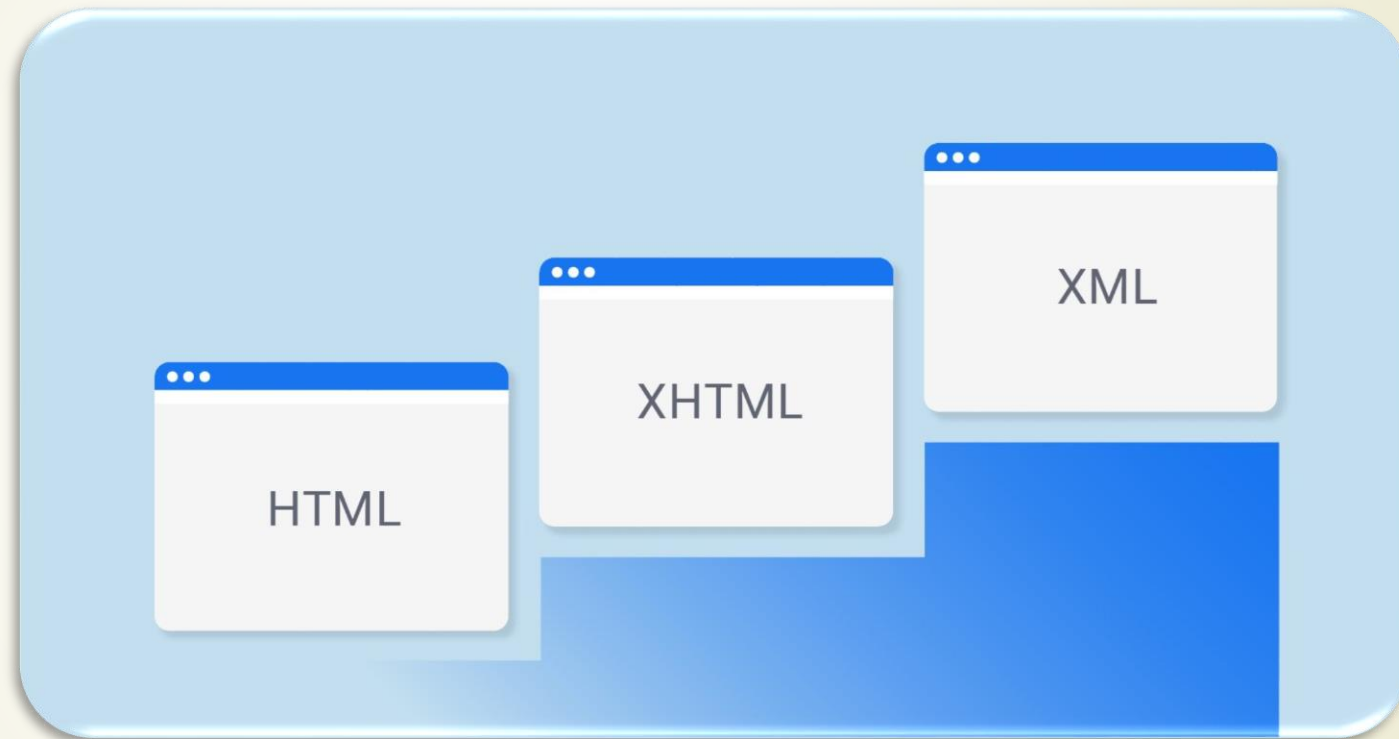
INTERNET TEHNOLOGIJE

Prof. dr Zoran Veličković, dipl. inž. el.

2022/23.

Prof. dr Zoran Veličković, dipl. inž. el.

Internet tehnologije



**Proširivi jezici za označavanje i stilizovanje dokumenata:
XML, XSLT i XHTML**

Sadržaj

- ▶ XML - PROŠIRIVI JEZIK ZA OZNAČAVANJE
 - ▶ XML – olakšava razmenu podataka
 - ▶ XML je ASCII fajl
 - ▶ XML rečnici
 - ▶ Postojeći XML rečnici
 - ▶ HTML kao XML rečnik
 - ▶ NEDOSTACI HTML-a
 - ▶ Primer intuitivnih XML oznaka
 - ▶ Primena XML-a u VisualStudio-u
 - ▶ W3C primer: XMLNews dokument
 - ▶ Konverzija XML i JSON formata
 - ▶ XML OZNAKE
 - ▶ XML i razdvajanje strukture od prikaza
 - ▶ DTD i XML
 - ▶ Interni DTD
 - ▶ Primer internog DTD-a
 - ▶ Web app i XML
 - ▶ Elementi i entiteti
 - ▶ Stablo XML dokumenta
- ▶ PRIKAZ XML DOKUMENTA U WEB ČITAČU
 - ▶ Struktura XML dokumenta
 - ▶ Dobro formatirani i validni dokumenti
 - ▶ Dekleracija XML dokumenta
- ▶ Jezik za transformisanje XML dokumenata – XSLT
- ▶ XML + HTML = XHTML
 - ▶ XHTML pravila
 - ▶ XHTML i CDATA
 - ▶ XHTML u odnosu na HTML 5

XML - proširivi jezik za označavanje

- **XML** (engl. e**X**tensible **M**arkup **L**anguage) je **PROŠIRIVI JEZIK** za **OZNAČAVANJE DOKUMENATA**.
- **XML** je kreiran za **SKLADIŠTENJE** i **PRENOS** podataka na mreži, uz osobinu da je istovremeno čitljiv i ljudima i mašinama.
- **XML** je samoopisujući jezik za označavanje **SLIČAN HTML**-u.
- Međutim, **XML** i **HTML** su kreirani sa različitim zadacima (ciljevima):
 - XML je dizajniran da **prenosi podatke** (fokus je na tome šta su podaci)
 - HTML je dizajniran da **prikaže podatke** (fokus je na prikazu podataka)
 - **XML** oznake **NISU UNAPRED DEFINISANE** kao HTML-a.
- Kod **XML**-a, sam autor mora da **DEFINIŠE** i **OZNAKE** i **STRUKTURU** dokumenta.



XML – olakšava razmenu podataka

- Mnogi računarski sistemi sadrže podatke u međusobom **NEKOMPATIBILNIM FORMATIMA**.
- **RAZMENA PODATAKA** između nekompatibilnih sistema (ili nadograđenih sistema) je zadatak koji oduzima **MNOGO VREMENA** Web programerima.
- Velike količine podataka moraju da se **KONVERTUJU**, a nekompatibilni podaci se često **GUBE**.
- Ovo je razlog zbog čega XML **ČUVA PODATKE** u formatu **ASCII** (engl. *American Standard Code for Information Interchange*) teksta.
- Na taj način se obezbeđuje softverski i hardverski **NEZAVISAN** način **ČUVANJA, TRANSPORTA i DELJENJA** podataka.
- XML takođe olakšava **PROŠIRENJE** ili **NADOGRAĐNJU** na nove operativne sisteme, nove aplikacije ili nove pretraživače, **BEZ GUBITKA PODATAKA**.
- Sa **XML**-om, **PODACI MOGU BITI DOSTUPNI** ljudima, računarima, glasovnim mašinama, izvorima vesti, itd.



XML je ASCII fajl

- Slično kao i HTML-u, oznake (XML labele) postoje **UPOREDO SA SADRŽAJEM**, ali su instrukcije zadužene za **IZGLED** dokumenta **SKRIVENE OD KORISNIKA**.
- **XML** donosi rešenja problem **PRELASKA SA PLATFORME** na **PLATFORMU** i pitanje korišćenja specifičnih aplikacija.
- Čitanje i tumačenje označenog dokumenta svodi se na rad sa **ASCII KARAKTERIMA**, što sve računarske platforme **DOBRO POZNAJU**.
- Jezik za označavanje se usredsređuje na pravljenje DOKUMENATA **NEZAVISNIH OD PLATFORME** čime se omogućava pristup SVIM APLIKACIJAMA koje razumeju jezik za označavanje.
- Različite specifikacije HTML-a i dodatni **ELEMENTI** i **ATRIBUTI** su zapravo **DTD**-ovi koji detaljno govore **KOJE OZNAKE** mogu da se koriste pri pravljenju Web stranice.

XML rečnici

- **XML** je prvobitno bio razvijen da razreši probleme **IZDAVAŠTVA VELIKOG OBIMA**, a tek je kasnije počeo da se primenjuje u označavanju raznolikih **PODATAKA NA WEBU**.
- Upotrebom XML-a se pojednostavljuje **PRETRAŽIVANJE INFORMACIJE NA WEB-u** kao i tzv. **METAPODATAKA** na Webu - informacije o informacijama.
- Poslovi **PRETRAŽIVANJA** ali i **RAZMENE** informacija između računarskih sistema se pojednostavljaju i **AUTOMATIZUJU** primenom XML-a.
- Kada se **XML** koristi za opis **SPECIFIČNO ORIJENTISANIH PROBLEMA**, nazivaju se **XML REČNICI**.
- Ideja da se HTML **PROŠIRITI** za svaki pojedinačni **TIP PROBLEMA** nije prihvatljiva jer bi on postao **PREOBIMAN!**
- **XML** treba da **OTKLONI NEDOSTATKE** primećene korišćenjem HTML-a.
- **REŠENJE** je da se definiše **META JEZIK** za **KREIRANJE JEZIKA** za obeležavanje orijentisanog ka Web-u.



Postojeći XML rečnici*

- Već su razvijeni **XML** rečnici, a samim tim i odgovarajući **DTD**-ovi:
 - **CDF** (engl. *Channel Definition Format*) - Postavlja serverske kanale za prosleđivanje i rutinsku isporuku informacija zasnovanih na Web-u.
 - **CML** (engl. *Chemical Markup Langue*) – Opisuje hemijske formule i podatke.
 - **MathML** (engl. *Mathematical Markup Langue*) – opisuje matematičke podatke.
 - **OSD** (engl. *Open Software Description*) – Opisuje softverske pakete za daljinsko postavljanje i instalaciju softvera i komponenata intraneta i Interneta.
 - **WIDL** (engl. *Web Interface Definition*) – Definiše interfejse za programiranje aplikacije (API) za Web servise i informacije.
 - **SMIL** (engl. *Synchronized Multimedia Integration Language*) – Opisuje skupove multimedijalnih izvora (audio i video fajlovi) koji se pokreću zajedno u prezentaciji.

Nedostaci HTML-a

- ▶ HTML ima **FIKSNI SKUP OZNAKA**, samim tim, ne omogućava se kreiranje **SOPSTVENIH OZNAKA** koje bi **neki drugi program** mogao da interpretira.
- ▶ **HTML** tehnologija je razvijena za **PREZENTACIJU** i ne nosi informaciju **O ZNAČAJU SADRŽAJA** koji se nalazi između HTML oznaka.
- ▶ HTML je **RAVAN** (linearan) – ne poseduje definiciju **VAŽNOSTI OZNAKA** tako da nije moguće prikazati **HIJERARHIJU PODATAKA**.
- ▶ Iako su Web aplikacijama često potrebni samo neki delovi, šalju se **ČITAVI DOKUMENTI** što ima za posledicu povećanje mrežnog saobraćaja što izaziva **ZAGUŠENJE SAOBRAĆAJA**.
- ▶ Dodavanjem **OZNAKA TEKSTU**, kao na sledećem primeru, formiraće **MNOGO VIŠE** informacija o samom tekstu a da pritom one nisu vezane za način prikazivanja.

Primer intuitivnih XML oznaka

<knjiga>

<naslov> Internet tehnologije </ naslov>

<poglavlje BROJ="1">

<paragraf>

<prvo> Osnove HTML-a </ prvo>

</ paragraf>

</ poglavlje>

</ knjiga>

Primena XML-a u VisualStudio-u (1)

XML oznake

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <configuration>
3   <startup>
4     <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
5   </startup>
6 </configuration>
```

Opis konfiguracije .NET Framework aplikacije

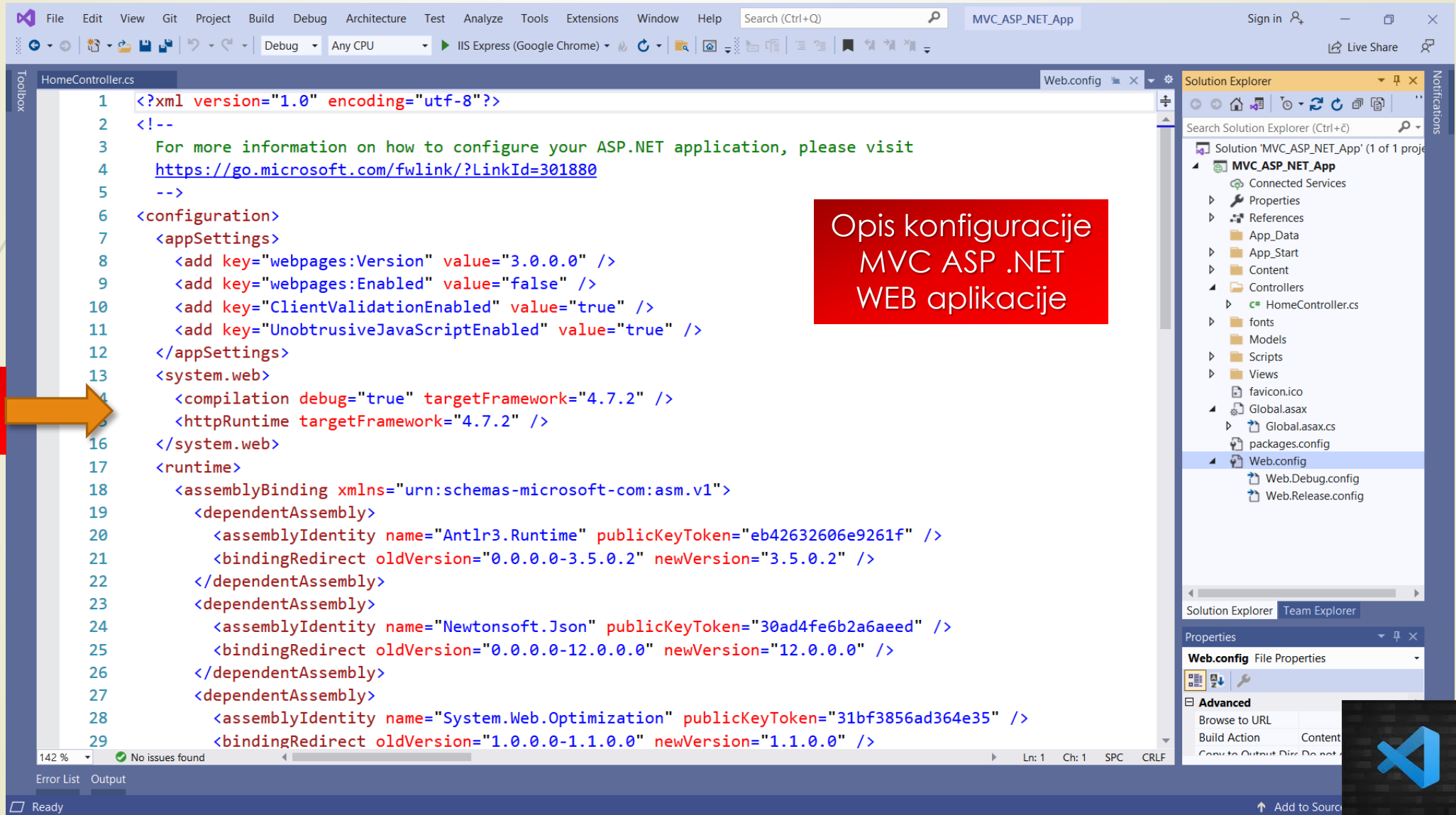
The screenshot displays the Visual Studio IDE with the 'App.config' file open. The code in the editor is as follows:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
  </startup>
</configuration>
```

The Solution Explorer on the right shows the project structure for 'Vezba_1_Z6', including 'App.config' and 'Program.cs'. The Properties window at the bottom right shows the file properties for 'App.config', including 'File Name' and 'Full Path'. The status bar at the bottom indicates '172 %' zoom, 'No issues found', and the current line and column positions (Ln: 6, Ch: 17).



Primena XML-a u VisualStudio-u (2)



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <!--
3   For more information on how to configure your ASP.NET application, please visit
4   https://go.microsoft.com/fwlink/?LinkId=301880
5   -->
6 <configuration>
7   <appSettings>
8     <add key="webpages:Version" value="3.0.0.0" />
9     <add key="webpages:Enabled" value="false" />
10    <add key="ClientValidationEnabled" value="true" />
11    <add key="UnobtrusiveJavaScriptEnabled" value="true" />
12  </appSettings>
13  <system.web>
14    <compilation debug="true" targetFramework="4.7.2" />
15    <httpRuntime targetFramework="4.7.2" />
16  </system.web>
17  <runtime>
18    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
19      <dependentAssembly>
20        <assemblyIdentity name="Antlr3.Runtime" publicKeyToken="eb42632606e9261f" />
21        <bindingRedirect oldVersion="0.0.0.0-3.5.0.2" newVersion="3.5.0.2" />
22      </dependentAssembly>
23      <dependentAssembly>
24        <assemblyIdentity name="Newtonsoft.Json" publicKeyToken="30ad4fe6b2a6aeed" />
25        <bindingRedirect oldVersion="0.0.0.0-12.0.0.0" newVersion="12.0.0.0" />
26      </dependentAssembly>
27      <dependentAssembly>
28        <assemblyIdentity name="System.Web.Optimization" publicKeyToken="31bf3856ad364e35" />
29        <bindingRedirect oldVersion="1.0.0.0-1.1.0.0" newVersion="1.1.0.0" />
30      </dependentAssembly>
31    </assemblyBinding>
32  </runtime>
33 </configuration>
```

Opis konfiguracije
MVC ASP .NET
WEB aplikacije

Niz XML
oznaka



W3C primer: XMLNews dokument

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<nitf>
```

```
  <head>
```

```
    <title> Colombia Earthquake </title>
```

```
  </head>
```

```
  <body>
```

```
    <headline>
```

```
      <h1> 143 Dead in Colombia Earthquake </h1>
```

```
    </headline>
```

```
    <byline>
```

```
      <bytag> By Jared Kotler, Associated Press Writer </bytag>
```

```
    </byline>
```

```
    <dateline>
```

```
      <location> Bogota, Colombia </location>
```

```
      <date> Monday January 25 1999 7:28 ET </date>
```

```
    </dateline>
```

```
  </body>
```

```
</nitf>
```


XMLNews: Konverzija XML i JSON formata

- Podaci skladišteni u **XML** fajlu se mogu predstaviti i u **JSON**-u (JavaScript notifikacija).

The screenshot displays the 'JSON formatter' website interface. The main heading is 'XML to JSON'. On the left, an XML editor shows the following code:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <nitf>
3 <head>
4 <title>Colombia Earthquake</title>
5 </head>
6 <body>
7 <headline>
8 <h1>143 Dead in Colombia Earthquake</h1>
9 </headline>
10 <byline>
11 <bytag>By Jared Kotler, Associated Press Writer
12 </bytag>
13 </byline>
14 <dateline>
15 <location>Bogota, Colombia</location>
16 <date>Monday January 25 1999 7:28 ET</date>
17 </dateline>
18 </body>
19 </nitf>
```

A red 'XML' stamp is overlaid on the XML code. In the center, there are four buttons: 'Load Data', 'XML to JSON', 'Plain JSON', and 'Download'. A dropdown menu shows '2 Tab Space'. On the right, a JSON editor shows the converted output:

```
1 {
2   "nitf": {
3     "head": {
4       "title": "Colombia Earthquake"
5     },
6     "body": {
7       "headline": {
8         "h1": "143 Dead in Colombia Earthquake"
9       },
10      "byline": {
11        "bytag": "By Jared Kotler, Associated
12          Press Writer"
13      },
14      "dateline": {
15        "location": "Bogota, Colombia",
16        "date": "Monday January 25 1999 7:28 ET"
17      }
18    }
19  }
```

A red 'JSON' stamp is overlaid on the JSON code.

<https://jsonformatter.org/xml-to-json>

XML oznake

- ▶ **DODATNE OZNAKE** u tekstu pomažu da se sazna **MNOGO VIŠE O SAMOM DOKUMENTU**, ali ne pomažu da se sazna o čemu se radi!
- ▶ Oznake dodeljuju **LABELE ZA OZNAČAVANJE** delova dokumenta i prave se korišćenjem određene **NOTACIJE**.
- ▶ **PAROVI OZNAKA** upotrebljeni za opis sadržaja teksta dodaju odgovarajuću, **SPECIFIČNU LABELU U TEKSTU** koji okružuju.
- ▶ **OZNAČAVANJEM** i **UGREŽĐAVANJEM** se definišu delovi dokumenta i preciznije se opisuje njihova **ULOGA**.
- ▶ Tekst između oznaka **<naslov>...</naslov>** predstavlja samo naslov, ali kada se oznake **<naslov>...</naslov>** ugnezde unutar oznaka **<knjiga>...</ knjiga>** ili **<poglavlje>...</ poglavlje>**, uloga je očigledna!
- ▶ Definisanje **IZGLEDA DOKUMENTA** je zadatak **UREĐAJA ZA PRIKAZ!**

XML i razdvajanje strukture od prikaza

- Ključni zadatak jezika za označavanje je da **DODA LABELE** (oznake) delovima teksta.
- Na osnovu dodeljenih labela **IZLAZNI UREĐAJ SAM** odlučuje **KAKO** najbolje da formatira sadržaj.
- Ovaj pristup kreiranja dokumenta treba da podrži **RAZDVAJANJE STRUKTURE OD PRIKAZA!**
- Jezik za označavanje služi za **OPISIVANJE SADRŽAJ DOKUMENTA**, tako da kreator dokumenata treba da dobro **OPIŠE SVOJ DOKUMENT** najtatačnije moguće **BEZ OBZIRA** na **KONAČAN IZGLED DOKUMENTA**.
- Za razliku od **XML**-a, **HTML** kreatori **PRILAGODAVAJU OZNAČAVANJE** da bi postigli **ODGOVARAJUĆI IZGLED** svojih Web stranica!
- Otuda potiče IZVOR **POZNATIH PROBLEMA NEKOMPATIBILNOSTI** Web pretraživača.

DTD i XML

- Poznato je da u HTML-u **NE POSTOJI** oznaka tipa `<bold>...</bold>`, tako da se i **NE MOGU** KORISTITI za boldiranje slova!
- Da bi mogla da se primenjuje ova oznaka (i stotine drugih) **MORA** se **OPISATI** u **DTD**-u za HTML 4.0.
- Nove verzije HTML-a, ustvari predstavljaju **JEDAN OBLIK DTD**-a, što znači da je dokument pisan po pravilima HTML 4.0.
- **DTD** specifikacija može da ima u sebi elemente poput **FREJMOVA** koji nisu podržani u ranijim DTD-ovima.
- Razlike među Web čitačima su sadržane u NJIHOVIM **DTD**-ovima!
- Da bi se napisao kompatibilan kod sa HTML-om 4.0, postavlja se **DEKLERACIJA** U DTD-a na SAMOM **POČETKU DOKUMENTA**.

Interni DTD

- Da se podsetimo, DEFINICIJA TIPA DOKUMENTA – **DTD** opisuje skupove:

- **OZNAKA,**

- **ATRIBUTA** i

- **PRAVILA**

kojima se definišu **ELEMENTI** za opisivanje dokumenta.

- Da bi mogle da se koriste **nove - dodate XML OZNAKE**, neophodno ih je definisati u **INTERNOM DTD**-u i treba da opišu:

- **SADRŽAJ DOKUMENTA** i

- **METAPODATKE.**

- Jezik za označavanje koristi **OZNAKE** ugrađene **DIREKTNO U SAM TEKST** da bi opisao različite delove teksta.

Primer internog DTD-a

```
<?xml version = "1.0" ?>
```

```
<!DOCTYPE osoba  
[  
  <!ELEMENT osoba(dob, pol, ime)>  
  <!ELEMENT dob(#PCDATA)>  
  <!ELEMENT pol(#PCDATA)>  
  <!ELEMENT ime(#PCDATA)>  
]>
```

Interni DTD,
deklaracija korišćenih elemenata

Jezik za pisanje DTD-a: proširena
Bakus-Naurova forma – **EBNF**
(engl. *Extended Backus-Naur
Form*).

```
<osoba>  
  <dob>40</dob>  
  <pol>muški</pol>  
  <ime>Ime</ime>  
</osoba>
```

Povezivanje se može ostvariti i sa
eksternim DTD-om na sledeći način:
<!DOCTYPE person SYSTEM "person.dtd">

PCDATA - Parsed Character DATA

Web app i XML

- ▶ Jezik za označavanje i **APLIKACIJE** koje ga podržavaju rešavaju ovaj problem **ODVAJANJEM STRUKTURE** od **PRIKAZA I IZLAZA**.
- ▶ **APLIKACIJE** koje razumeju jezik za označavanje **OBRAĐUJU** i **TUMAČE** oznake i generišu **VIŠE RAZLIČITIH IZLAZA** u obliku pogodnom za odgovarajuću primenu.
- ▶ Iako je generisano **VIŠE** različitih izlaza, **SVI** oni su zasnovani **NA JEDNOM XML DOKUMENTU**.
- ▶ Da bi se potencirao OPIS DOKUMENTA, a istovremeno omogućio uticaj na konačan izgled, razvijene su **KASKADNE DATOTEKE SA STILOVIMA - CSS** (engl. *Cascading Style Sheets*) - već smo ih obradili.
- ▶ Jezik za označavanje ODREĐUJE **SADRŽAJ**, dok CSS datoteke određuju **KONAČNI PRIKAZ**.

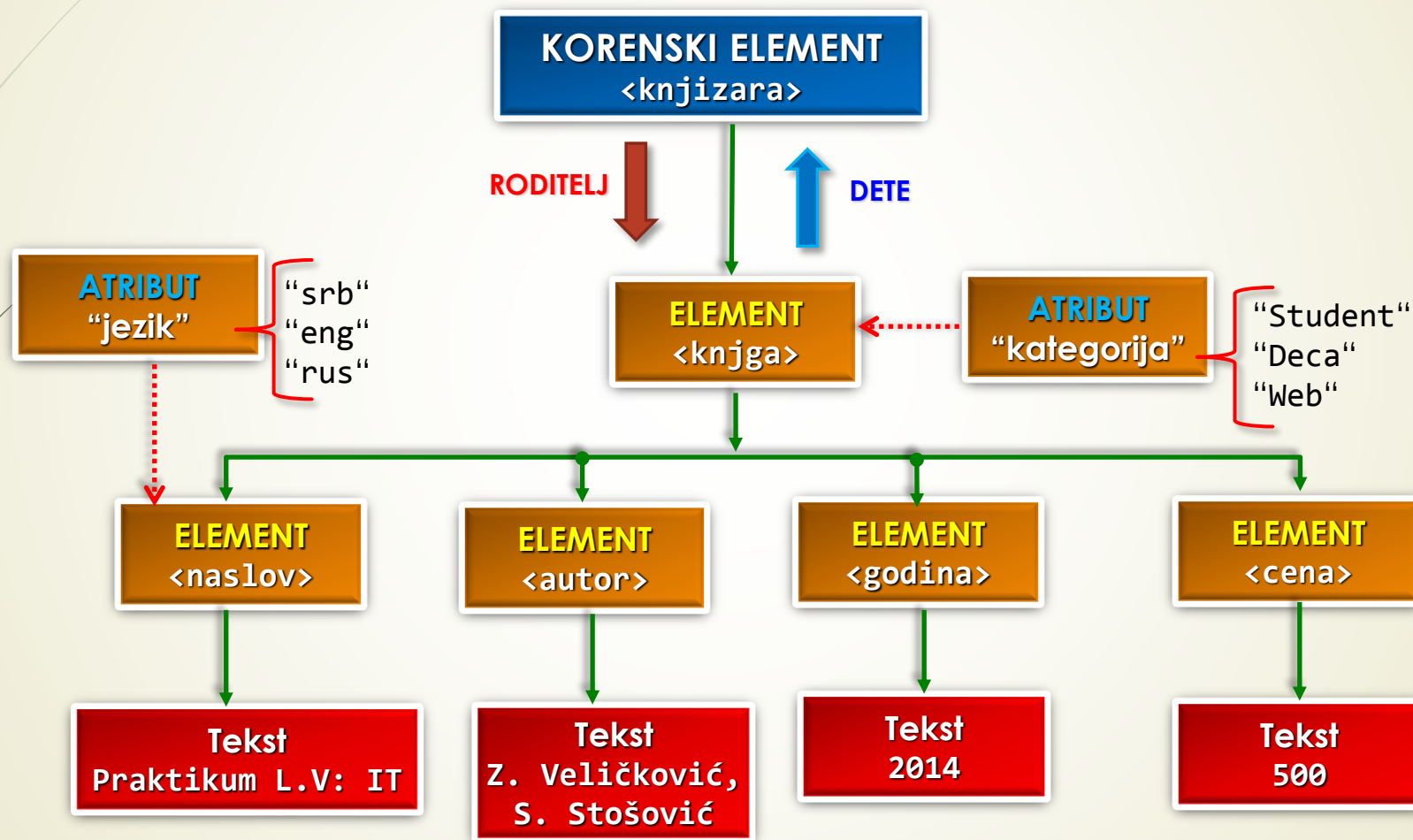
XML elementi i entiteti

- ▶ Slično kao kod HTML-a, **XML DOKUMENTI** su podeljeni na **KONTEJNERE** koji se nazivaju **XML ELEMENTI**.
- ▶ **XML ELEMENTI** su ključne strukture u XML dokumentu, a **DTD**-ovi određuju **KOJI** elementi **MOGU BITI UPOTREBLJENI** u dokumentima pisanim za dati DTD.
- ▶ **XML KONSTRUKTORI**, kao što su **ATRIBUTI** i **DEKLARACIJE SADRŽAJA**, pružaju dodatne informacije o elementima i načinu njihovog međusobnog uticaja.
- ▶ Dakle, **ELEMENTI** su definisani u DTD-ovima, a **OZNAKE** se koriste da predstavljaju te elemente u dokumentima.
- ▶ **XML ENTITET** je, uopšte, **JEDINICA PODATAKA** koja se koristi kod XML-a i mogu da sadrže:
 - ▶ **BINARNE** PODATKE,
 - ▶ da imaju formu **GRAFIČKIH** ili **ZVUČNIH** fajlova ili
 - ▶ **JAVA APLETE**, ...

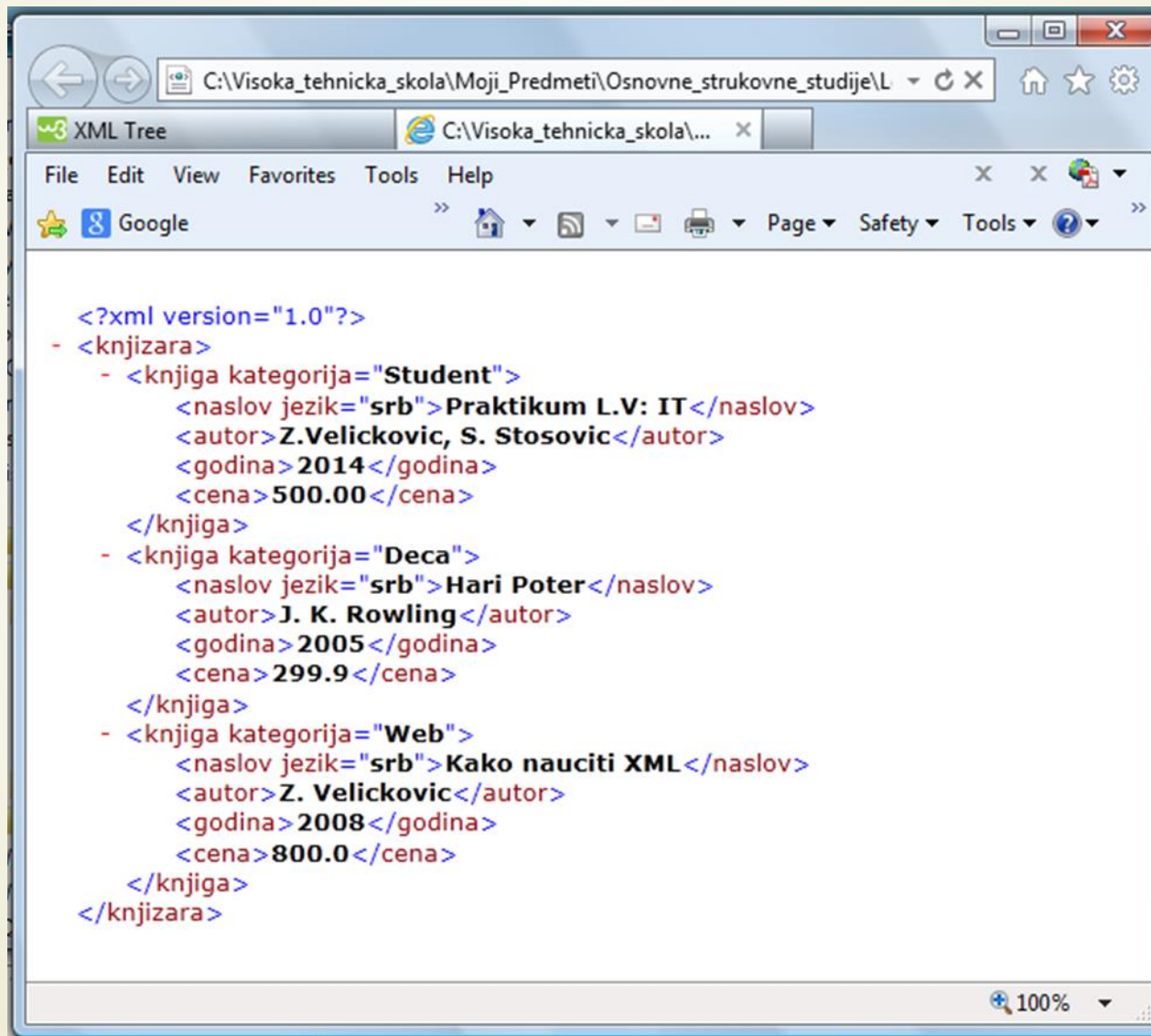
Stablo XML dokumenta

- ▶ Svaki XML dokument može imati **EKSTERNI** i **INTERNI** DTD.
 - ▶ Eksterni DTD sadrži sve **OPŠTE DEKLARACIJE** - kao što su **ELEMENTI** i **ATRIBUTI**.
- ▶ **MEĐUSOBNI ODNOS** XML elemenata definisan je **DTD**-u.
 - ▶ SVAKI XML DOKUMENT mora da poseduje **KORENSKI ELEMENT**, iz koga se izvode svi ostali.
 - ▶ Princip **RODITELJ-DETE** je primenjen u strukturi XML dokumenta.
- ▶ Kretanje kroz **STABLO XML DOKUMENTA** omogućava pristup svakom XML elementu korišćenjem:
 - ▶ **JAVASCRIPT FUNKCIJA:**
 - ▶ `getElementsByTagName()`, `appendChild()`, `getAttribute()`, `removeChild()`, `setAttribute()`, ...
 - ▶ **SVOJSTAVA:**
 - ▶ `nodeName`, `nodeType`, `nodeValue`, `childNodes`, `attributes`, `parentNode`, `tagName`, ...

Strabalo XML dokumenta - DOM



Prikaz XML dokumenta u Web čitaču



```
<?xml version="1.0"?>
- <knjizara>
  - <knjiga kategorija="Student">
    <naslov jezik="srb">Praktikum L.V: IT</naslov>
    <autor>Z.Velickovic, S. Stosovic</autor>
    <godina>2014</godina>
    <cena>500.00</cena>
  </knjiga>
  - <knjiga kategorija="Deca">
    <naslov jezik="srb">Hari Potter</naslov>
    <autor>J. K. Rowling</autor>
    <godina>2005</godina>
    <cena>299.9</cena>
  </knjiga>
  - <knjiga kategorija="Web">
    <naslov jezik="srb">Kako nauciti XML</naslov>
    <autor>Z. Velickovic</autor>
    <godina>2008</godina>
    <cena>800.0</cena>
  </knjiga>
</knjizara>
```

Dobro formatirani i validni dokumenti

- XML specifikacija poznaje **DVA TIPA** dokumenta:
 - **DOBRO FORMATIRAN** XML dokument – u skladu sa pravilima **XML SINTAKSE** (svi elementi moraju imati odgovarajuću završnu oznaku, završavaju se kosom crtom kod praznih elemenata, ...)
 - **VALIDNI DOKUMENTI** su dobro formatirani i **U SKALADU** su sa **DTD**-om.
- **VALIDACIONI PARSERI** su programi koji se koriste za **PROVERU DOKUMENTA** u odnosu na njegov **DTD**.
- Ako dokument **NE PRATI** strukturu definisani DTD-om parser će prijaviti **GREŠKU**.
- Da bi neki **RAČUNARSKI SISTEM** mogao da koristi XML, potrebno je da poseduje **DVE** softverske komponente:
 1. **XML PARSER** i
 2. **APLIKACIJU** (koja koristi podatke iz stabla dokumenta DOM).

Deklaracija XML dokumenta

- ▶ **INTERNI DTD** sadrži deklaracije koje su **SPECIFIČNE ZA DATI DOKUMENT**, kao što su recimo deklaracije za grafiku.
- ▶ **DTD PRAVILA** govore **KAKO** oznake mogu da budu **UGNEŽĐENE** nazivaju se - **MODELI SADRŽAJA**.
- ▶ **XML DEKLARACIJA** ima **DVA** dela i obavlja dva važna zadatka:
 1. **IDENTIFIKUJE** dokument kao XML dokument;
 2. **SAOPŠTAVA** APLIKACIJI (koja obrađuje dokument) da li treba da brađuje **SAMO** dokument ili i dokument i DTD.

XML PROLOG

```
<?xml version = "1.0" rmd = "ALL" encoding = "UTF-8"?>
```

Oznaka XML dokumenta (1)

Atributi

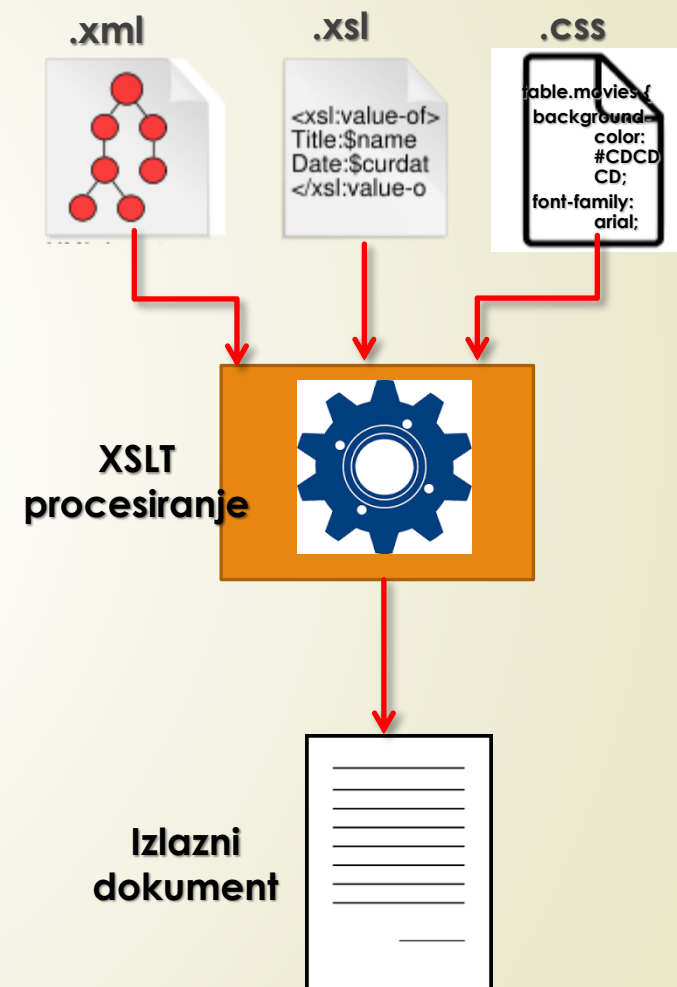
RMD (engl. *Required Markup Declaration*)

Dekleracija i atributi XML-a

- Deklaracija počinje znakom pitanja (?).
- Atribut **version** određuje **VERZIJU** XML-a koja je korišćena prilikom pisanja dokumenta.
- Atribut **RMD** (engl. *Required Markup Declaration*) - deklaracija zahtevanog označavanja, pruža mogućnost saopštavanja aplikaciji da li ona **MORA** da obradi i **PRATEĆI DTD**.
 - Vrednost **RMD = "ALL"** ukazuje da **SVI PRIDODATI DTD**-ovi treba da budu obrađeni zajedno sa dokumentom.

Jezik za transformisanje XML dokumenata - XSLT

- **XSLT** (engl. e**X**tensible **S**tylesheet **L**anguage *Transformations*) je jezik za transformisanje XML dokumenata u **DRUGE XML DOKUMENTE** ili u **DRUGE FORMATE** kao što je:
 - **HTML** za Web stranicu,
 - **PDF** dokument,
 - **PNG**, ...
- Originalni dokument prilikom transformacije ostaje **NEPROMENJEN**, dok se **NOVI DOKUMENT** kreira na bazi sadržaja postojećeg dokumenta.
- Uobičajeno je da je ulazni dokument XML fajl, a izlazni - **ZAVISI OD POTREBA!**



Primer: movies.xml* (1)

```
<?xml version = "1.0" encoding = "UTF-8"?>
<?xml-stylesheet type = "text/xsl" href = "movies.xsl"?>
<movies>
  <movie>
    <title>Aliens</title>
    <year>1986</year>
    <rank>8.2</rank>
  </movie>
  <movie>
    <title>Apollo 13</title>
    <year>1995</year>
    <rank>7.5</rank>
  </movie>
  <movie>
    <title>Pi</title>
    <year>1998</year>
    <rank>7.1</rank>
  </movie>
</movies>
```

Povezivanje sa programom
za transformaciju

Primer: movies.xml* (2)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xsl:stylesheet version = "1.0"
```

```
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">
```

```
<xsl:output method = "html"
```

```
  doctype-system = "about:legacy-compat"
```

```
  encoding = "UTF-8"
```

```
  indent = "yes" />
```

```
<xsl:template match="/">
```

```
  <html>
```

```
    <head>
```

```
      <title>List of Movies.</title>
```

```
      <link rel = "stylesheet" href = "movies.css" />
```

```
    </head>
```

Atribut `xmlns` definiše imenski prostor u cilju sprečavanja konflikata imena!

XSLT KOMANDE
jezika za transformaciju

Kreira se HTML fajl

Povezivanje sa fajlom za stilizaciju

Koristi CSS fajl

Napomena: XSLT tehnologija se NE PROUČAVAJU na ovom kursu !

Primer: movies.xml* (3)

```
<body>
  <table class="movies">
    <thead>
      <tr>
        <th>Title</th>
        <th>Year</th>
        <th>Rank</th>
      </tr>
    </thead>
    <tbody>
      <xsl:for-each select="movies/movie">
        <xsl:sort select="rank"/>
        <tr>
          <td><xsl:value-of select="title"/></td>
          <td><xsl:value-of select="year"/></td>
          <td><xsl:value-of select="rank"/></td>
        </tr>
      </xsl:for-each>
    </tbody>
  </table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

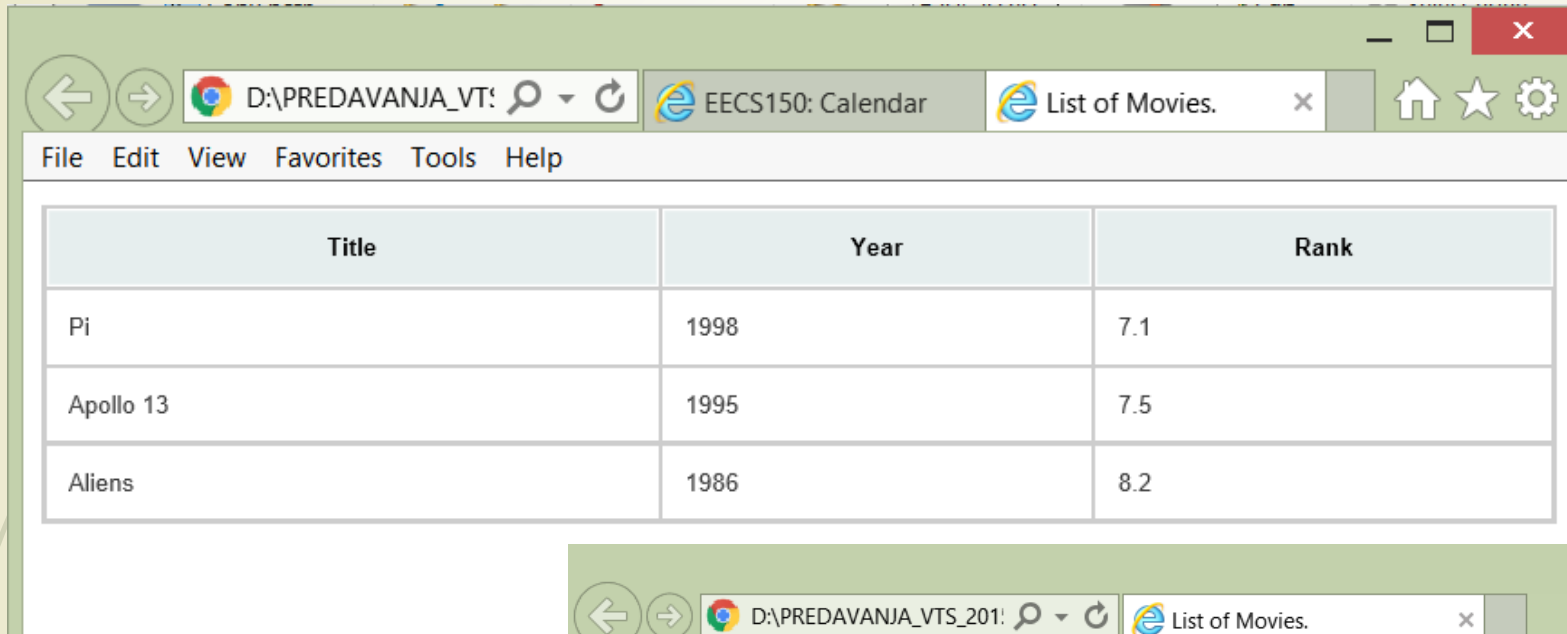
XSLT komande za
ispisivanje
pronađenih
podataka



Primer: movies.css*

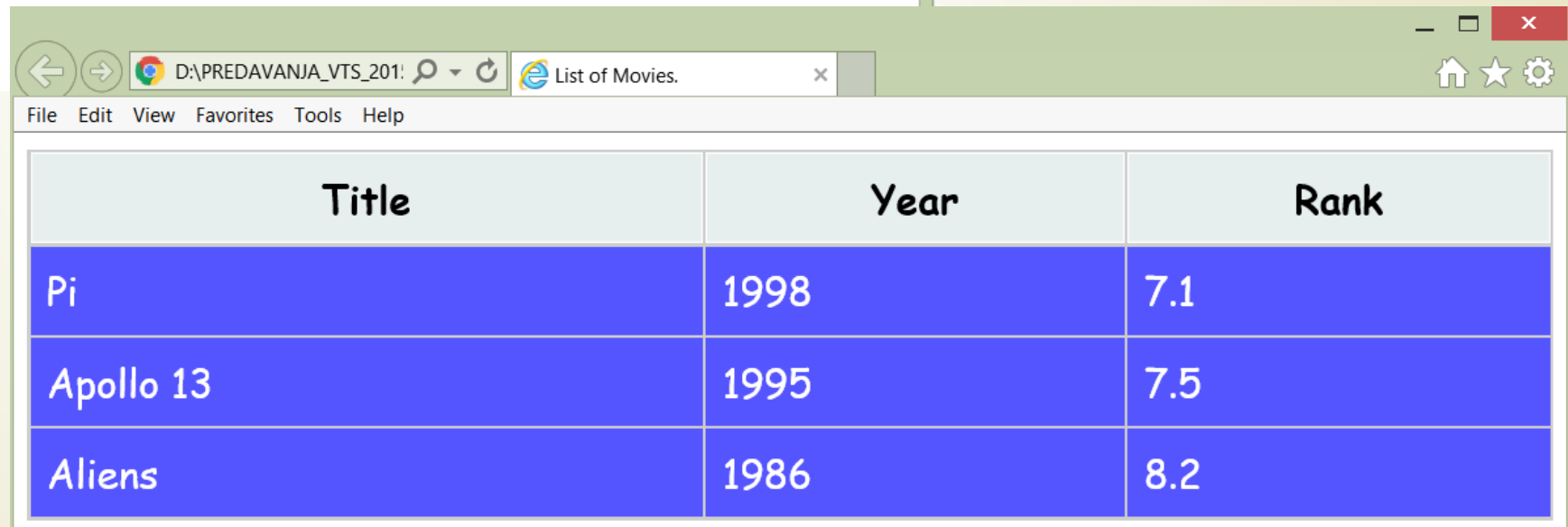
```
table.movies {  
    background-color: #CDCDCD;  
    font-family: Comic Sans MS;  
    margin: 10px 0pt 15px;  
    font-size: 8pt;  
    width: 100%;  
    text-align: left;  
}  
table.movies thead tr th {  
    background-color: #E6EEEE;  
    border: 1px solid #FFF;  
    font-size: 18pt;  
    padding: 10px;  
}  
table.movies tbody tr td {  
    background-color: ##5555FF;  
    color: white;  
    vertical-align: top;  
    padding: 10px; }
```

Movies.xml → movies.html (1)



A screenshot of a web browser window showing a table with three columns: Title, Year, and Rank. The table contains three rows of movie data.

Title	Year	Rank
Pi	1998	7.1
Apollo 13	1995	7.5
Aliens	1986	8.2



A screenshot of a web browser window showing a table with three columns: Title, Year, and Rank. The table contains three rows of movie data, with the rows highlighted in blue.

Title	Year	Rank
Pi	1998	7.1
Apollo 13	1995	7.5
Aliens	1986	8.2

Movies.xml → movies.html (2)

The screenshot shows a web browser displaying a table of movies. The table has three columns: Title, Year, and Rank. The data rows are Pi (1998, 7.1), Apollo 13 (1995, 7.5), and Aliens (1986, 8.2). Below the browser window, the DOM Explorer is open, showing the HTML structure of the page. The table is highlighted in the DOM tree, and the Styles pane on the right shows the styles applied to the table.

Title	Year	Rank
Pi	1998	7.1
Apollo 13	1995	7.5
Aliens	1986	8.2

```
<!DOCTYPE html SYSTEM "about:legacy-compat">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"></meta>
    <title>List of Movies.</title>
    <link href="movies.css" rel="stylesheet"></link>
  </head>
  <body>
    <table class="movies">
      <thead>
        <tr>
          <th>Title</th>
          <th>Year</th>
          <th>Rank</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>Pi</td>
          <td>1998</td>
          <td>7.1</td>
        </tr>
        <tr>
          <td>Apollo 13</td>
          <td>1995</td>
          <td>7.5</td>
        </tr>
        <tr>
          <td>Aliens</td>
          <td>1986</td>
          <td>8.2</td>
        </tr>
      </tbody>
    </table>
  </body>
</html>
```

Styles pane:

- table.movies {
 - font-family: Comic Sans MS;
 - font-size: 18pt;
 - text-align: left;

XML + HTML = XHTML

- **W3C** je preradio standardni HTML **U APLIKACIJU** sa proširivim jezikom za označavanje:
XML + HTML = XHTML.
- **XHTML** je **PROŠIRIV** (**X** u nazivu) i **MODULARAN** što znači da pruža mogućnost DODAVANJEM novih oznaka.
- **MODULARANOST** se odnosi na to da se iz Web čitača mogu **UKLONITI XHTML oznake** koje nisu podržane određenom platformom.
- U XHTML datotekama će se **SVI UGRAĐENI SKRIPTOVI, KOMENTARI** i **OPISI STILOVA** nalaziti u posebnim **XML ODELJCIMA CDATA** (engl. *Character DATA*).
- XHTML dokumenti **MORAJU** biti dobro formirani.
- Oznake **DELOVA DOKUMENTA** se koriste za raspoređivanje XHTML koda Web stranice u **DVA KONTEJNERA**:
 - **head**
 - **body**

koji se stavljaju unutar trećeg, **HTML** kontejnera.

XHTML pravila (1)

- Svaki od kontejnera u **XHTML**-u ima različitu namenu - isto kao i HTML-u.
- Elementi se moraju **UGNEZDITI**, ali NE i **PREKLAPATI**.
- Sva imena **ELEMENATA** i **ATRIBUTA** moraju biti napisana **MALIM SLOVIMA**.
- Takođe, otvorene uparene oznake moraju biti **ZATVORENE**, a svaka prazna oznaka mora imati KRAJ.
- **XHTML** dokumenti moraju imati element **<title>** u zaglavlju.
- **XHTML** dokumenti **MORAJU** počinjati **DOCTYPE** definicijom, VREDNOSTI SVIH ATRIBUTA moraju biti zadate između **znakova navoda**.
- Primer je dat na sledećem slajdu.

XHTML primer

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html>
```

```
  <head>
```

```
    <title> Moja prva ispravna XHTML stranica </title>
```

```
  </head>
```

```
  <body bgcolor= "white">
```

```
    <!--Sadržaj koji čitač treba da prikaže ide ovde. -->
```

```
  </body>
```

```
    ...
```

```
</html>
```

XHTML pravila (2)

- **VREDNOSTI ATRIBUTA** se mogu pisati i malim i velikim slovima - ali se moraju staviti pod navodnike:

`<table border = "0">` //ispravno

`<table border = 0>` //pogrešno!

- **XHTML** (za razliku od HTML-a) traži da se i oznake koje NISU KONTEJNERI (prazne oznake) **ZATVORE**.
- **MORAJU SE ZATVORITI** prazne oznake `
`, ``, `<frame>`, `<meta>`, `<option>`, `<param>` ... tako što će se iza naziva elementa staviti blanko (razmak) `[" "]` i kosa crta `[/]`.

`

`

`<applet code = "appl.clas" width = "10" hwidth = "20" />`

XHTML pravila (3)

- ▶ Uobičajeno je da se **DOCTYPE** definiciju upisuje na **POČETAK** HTML dokumenata.
- ▶ Svaki čitač Weba ima DTD ugrađen u svoj kod.
- ▶ DTD govori čitaču **KAKO DA ČITA** HTML oznake (a sada i XHTML oznake) i attribute koji opisuju Web stranicu.
- ▶ Čitač proverava svoj **interni DTD**,
 - ▶ ako posle proveravanja svog koda čitač utvrdi da njegov DTD nije taj koji je neophodan za stranicu, on traženi DTD može **PREUZETI** sa URL adrese naznačene u **DOCTYPE** definiciji.
- ▶ Čitač neće videti skriptove jer se **IGNORIŠU** svi komentari pa i oni unutar elemenata **<style>** i **<script>** !

XHTML i odeljak CDATA

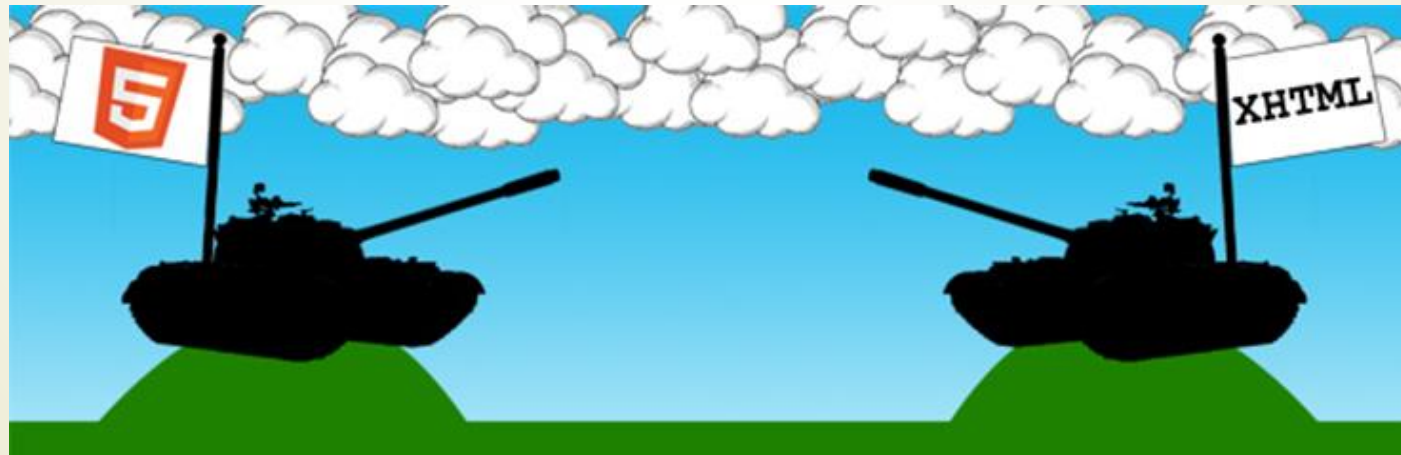
- U ovakvim slučajevima se koriste **SPOLJAŠNJI OPISI STILOVA** i **SKRIPTOVI**, ili umetnuti opisi i skriptove treba smestite u **ODELJKE - CDATA**.
- **CDATA** (eng1. *Character DATA*) znakovni niz koji čitač treba da posmatra kao **TEKSTUALNI ZNAKOVNI NIZ**.
- CDATA **NIJE** XHTML oznaka, već **ODVAJA ODELJAK** ili grupu iskaza koje čitač treba da čita **DRUGAČIJE**.
- Ako se hoće da opisi stilova i skriptovi budu ugrađeni u **XHTML** dokument, definišu se u novim **ODELJCIMA CDATA**.
- **CDATA** blok počinje isto kao **DOCTYPE** definicija: znakom manje iza koga sledi uzvičnik (" <! "):

```
<! [CDATA [If 5 < 10 method.write ("<p> 5 je i dalje manje od 10!!! </p>") ; ]]
```

JavaScript u telu odeljka CDATA

XHTML u odnosu na HTML 5

- Dugo se vodila bitka za prevlast između **HTML 5** i **XHTML**-a.
- **HTML 5** je nova verzija HTML jezika, dok je XHTML zasnovan na kombinaciji HTML 4 i XML 1.0 specifikacije (lansiran je posle 2000-te posle standardizacije HTML 4).
- **HTML 5** je naslednik HTML 4 i smatra se da će dugoročno gledano zameniti XHTML.
- Planirano je da HTML 5 objedini ne samo HTML 4 nego i XHTML 1 i DOM Level 2 specifikacije.
- HTML 5 je kompatibilan sa svim tipovima čitača, dok je održavanje XHTML-a jednostavnije.



XHTML i tip medija (1)

- **MENJANJE IZGLEDA** XHTML stranice se može adaptirati na osnovu **TIPA MEDIJA**.
- Pored ekrana prenosivih i stonih računara, posetioci za prikazivanje stranica mogu koristiti mobilni telefon, PDA ili neki drugi bežični uređaj.
- Korisnički uređaji se razlikuju **PO VELIČINI I NAČINU PRIKAZA!**
- Neophodno je menjati izgled sadržaja na osnovu **TIPA MEDIJA** pomoću koga se prikazuje.
- **CSS** omogućava pisanje pravila koja na osnovu tipa prikazivanja govore **KOJI STIL** treba da ima prikazani sadržaj.
- Da bi se na ekranu računara prikazala stranicu koja ima naslov od 36 tačaka i tekst veličine 18 tačaka, a na ručnom uređaju (PDA) prikazali istu stranicu sa tekstom od 12 tačaka, u Web dokument dodajte opise stila kao što su sledeći:

XHTML (2)

```
<head>
  <style type = "text/css" media = "screen">
    <!--
      h1 {font-size:36pt; text-align:center; font-weight:bold}
      body {font-size:18pt}
    -->
  </style>

  <style type = "text/css" media = "handheld">
    <!--
      h1 {font-size:12pt; font-weight:bold}
      body {font-size:12pt}
    -->
  </style>
</head>
```


XHTML i zastupnički server (1)

```
<head>
  <style type='text/css'>
  <!--
    @media screen, print {
      h1 {font-size:36pt; text-align:center; font-weight:bold}
      body {font-size:18pt}
    }
    @media handheld {
      h1 {font-size:12pt; font-weight:bold}
      body {font-size:12pt}
    }
  -->
  </style>
</head>
```

XHTML i tip medija (2)

- ▶ Ako se opisi stilova zapamte u spoljašnjim datotekama, čitač klijenta će PREUZETI **SAMO SPOLJAŠNJU DATOTEKU OPISA STILA** sa čijim tipom (ili tipovima) medija može da radi!

```
@import (/StyleSheets/NormalDisplay.css) screen, print;
```

```
@import (/StyleSheets/SmallDisplay.css) handheld;
```

- ▶ Zastupnički server čitača na **MOBILNIM TELEFONIMA** će preuzeti SAMO **SmallDisplay.css**.
- ▶ Pogledajte sledeće zajedničke radove studenata na Spec. studijama:
 - ▶ Bojan Krstić, Zoran Veličković, "Optimizacija prikaza mobilnih Web stranica zasnovana na media upitima", INFORMACIONE TEHNOLOGIJE, pp. 186-189, Žabljak 2011.
 - ▶ Bojan Krstić, Zoran Veličković, „Straničenje web stranica zasnovano na CSS 3 specifikaciji i primena „getusermedia“ metode u Opera reader web čitaču, Zbornik radova VTŠ NIŠ, pp. 17-20, 2012.