

Markup jezici i XML

Šta je markup

- oznaka (tag, code) koja opisuje deo sadržaja

The Supply of Money

CT:
(chapter title)

To understand chronic inflation and, in general, to learn what determines prices and why they change, we must now focus on the behavior of the two basic causal factors: the supply of and the demand for money.

The supply of money is the total number of currency units in the economy. Originally, when each currency unit was defined strictly as a certain weight of gold or silver, the name and the weight were simply interchangeable. Thus, if there are \$100 billion in the economy, and the dollar is defined as 1/20 of a gold ounce, then M can be equally considered to be \$100 billion or 5 billion gold ounces. As monetary standards became lightened and debased by governments, however, the money supply increased as the same number of gold ounces were represented by an increased supply of francs, marks or dollars.

Debasement was a relatively slow process. Kings could not easily have explained continuous changes in their solemnly defined standards. Traditionally, a new king ordered a recoinage with his own likeness stamped on the coins and, in the process, [p. 44] often redefined the unit so as to divert some much needed revenue into his own coffers. But this variety of increased money supply did not usually occur more than once in a generation. Since paper currency did not yet exist, kings had to be content with debasement and its hidden taxation of their subjects.

H1:
(first-level
heading)

1. What Should the Supply of Money Be?

What should the supply of money be? What is the “optimal” supply of money? Should M increase, decrease, or remain constant, and why?

Šta je markup

- markup je korišćen i na papiru prilikom slaganja teksta za označavanje pojedinih delova
 - naslovi, citati, brojevi poglavlja, itd.
- markup ne predstavlja sadržaj,
- već uputstvo kako neki sadržaj obraditi ili prikazati

Šta je markup

- markup se najčešće ne prikazuje eksplicitno
- krajnji čitalac ne vidi markup tagove
- ako se markup koristi za definisanje prezentacije dokumenta on je vidljiv iz grafičkog izgleda pojedinih elemenata
 - broj poglavlja, glavni naslov, podnaslov, citat

Ko definiše markup tagove

- firme
 - definicije su u vlasništvu firme
 - korisnik ne mora da zna sve detalje o markupu
 - Microsoft: Word .doc format
 - Adobe: PostScript, PDF
- otvoreni standardi
 - definišu ih tela za standardizaciju (ISO, W3C, ...)
 - svi detalji su poznati i javno dostupni
 - SGML, HTML, XML

Rad sa više markup sistema istovremeno

- tehnički komplikovan
 - traži puno ručnih korekcija
 - primer
 - autor koristi Microsoft Word (.doc)
 - lektor koristi OpenOffice Writer (.odt)
 - priprema i korektura koriste Adobe InDesign (.indd)
 - štampa koristi PDF (.pdf)
 - web izdanje koristi HTML (.html)
- šanse da dođe do problema u konverziji formata su vrlo velike

Otvoreni standardi za markup

- korišćenje otvorenih standarda bi trebalo da pojednostavi razmenu podataka između
 - različitih sistema
 - različitog softvera
 - različitih učesnika u proizvodnji sadržaja

Primer markupa

- Quark XPress

@LFI: Prva stavka

@LMI: Druga stavka

@LMI: Treća stavka

@LLI: Četvrta stavka

- HTML

```
<ul>
```

```
<li>Prva stavka</li>
```

```
<li>Druga stavka</li>
```

```
<li>Treća stavka</li>
```

```
<li>Četvrta stavka</li>
```

```
</ul>
```

Razlike:

- HTML ima hijerarhiju tagova, Quark nema
- Quark mora posebno da označava prvi i poslednji element liste zbog formatiranja
- Quark ne može da definiše listu u listi

Nejasan markup

- nejasan (nedorečen, višeznačan) markup
 - u formatiranju: oslanjanje na vizuelni izgled delova teksta kao opis značenja
 - u strukturi: vidi prethodni primer
 - nedostajući tagovi
 - “na slici ispod je prikazano...”

Razdvajanje strukture i prikaza

- osnovni razlozi za pojavu sistema sa nejasnim markupom
 - fokus na prikaz dokumenta
 - ograničenost na sopstveni, zatvoreni sistem
 -
- otvoreni standardi za markup se zasnivaju na konceptima
 - markiranja strukture
 - markiranja značenja

Označavanje strukture i značenja

- označavanje strukture

```
<chapter>
  <section>
    <title>...</title>
    ...
  </section>
  <section>
    <title>...</title>
    ...
  </section>
</chapter>
```

- označavanje značenja

```
<opis>
  Vlade Divac je centar i
  visok je <visina>211
  cm</visina> i težak je
  <masa>120 kg</masa>. ...
</opis>
```

Ovi koncepti nisu suprotstavljeni i mogu se kombinovati

Istorija standarda za strukturirani markup

- 1969: Generic Markup Language (GML)
 - definisanje različitih tipova dokumenata (skupova tagova)
 - dokument kao hijerarhija elemenata
 - razvijen u okviru IBM-a kao komercijalan proizvod
- 1980-1986: Standard Generalized Markup Language (SGML)
 - razvijen na bazi GML-a kao otvoreni standard
 - ANSI: American National Standards Institute
 - GCA: Graphic Communications Association
 - ISO: International Organization for Standardization
 - vrlo fleksibilan, sa širokim područjem primene
 - Association of American Publishers (AAP): The Manuscript Project

Istorija standarda za strukturirani markup

- 1989-1992: HyperText Markup Language (HTML)
 - razvijen u CERN-u kao jednostavan, fiksni skup tagova
 - kasnije formalno definisan kao jedna SGML gramatika
 - jednostavnost HTML-a je jedan od razloga za brzi rast Interneta
- 1998: Extensible Markup Language (XML)
 - HTML nije mogao da posluži svrsi u svim primenama na webu
 - SGML je suviše složen (fleksibilan) standard
 - XML je definisan kao podskup SGML-a koji je znatno jednostavnije implementirati
 - izbaciti 10% funkcionalnosti SGML-a (koje se ionako retko koriste) i smanjiti složenost za 90%

HTML

- dokument je hijerarhija elemenata
- mogu se koristiti samo unapred definisani tagovi

```
<html>
```

```
  <head>...</head>
```

```
  <body>
```

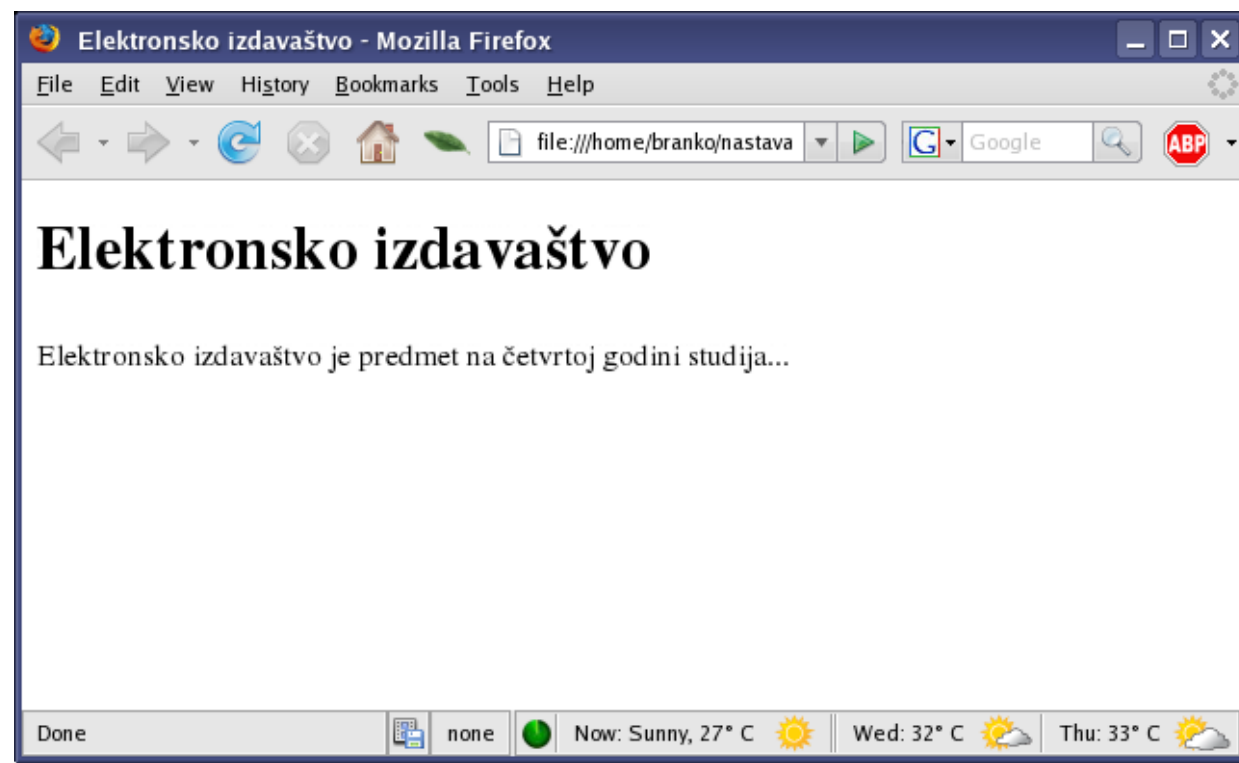
```
    <h1>Elektronsko izdavaštvo</h1>
```

```
    <p>...</p>
```

```
    ...
```

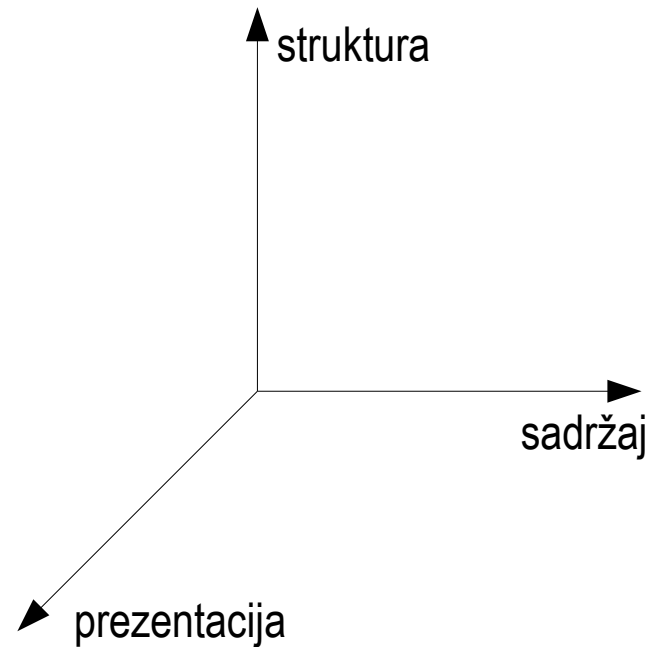
```
  </body>
```

```
</html>
```



Pojam dokumenta

- osnovni nosilac informacija
- osnovna jedinica razmene informacija
- tri karakteristike (dimenzije) dokumenta
 - sadržaj
 - struktura
 - prezentacija



XML

- ne definiše nikakve tagove unapred, već ih korisnik sam definiše
- XML je jezik za definisanje markup jezika, tj. metajezik
- HTML se može posmatrati kao jedan od jezika definisanih pomoću XML-a
- drugi primeri
 - CrossRef (povezivanje bibliografskih podataka)
 - CML (opis hemijskih jedinjenja)
 - OFX (razmena finansijskih dokumenata i elektronsko plaćanje)
 - NewsML (novinsko izdavaštvo)
 - SVG (vektorska grafika)
 - MathML (matematičke formule)

XML podrazumeva familiju standarda

- XSL (Extensible Stylesheet Language) - vizuelizacija
 - XSLT: XSL Transformations
 - XSL-FO: XSL Formatting Objects
- XPath: označavanje strukture dokumenata
- XLink: povezivanje dokumenata
- XQuery: pretraživanje dokumenata (po sadržaju i strukturi)

Ciljevi XML-a

- da odgovara upotrebi na Internetu
- mogućnost korišćenja od strane različitih aplikacija
- (jednosmerna) kompatibilnost sa SGML-om
- da se programi za obradu XML dokumenata pišu lako
- jednoznačnost prilikom obrade XML dokumenata
- XML dokumenti čitki i rezonski jasni
- dizajn XML-a se vrši brzo
- dizajn XML-a je formalan i koncizan
- kreiranje XML dokumenata je lako
- konciznost je od minimalne važnosti

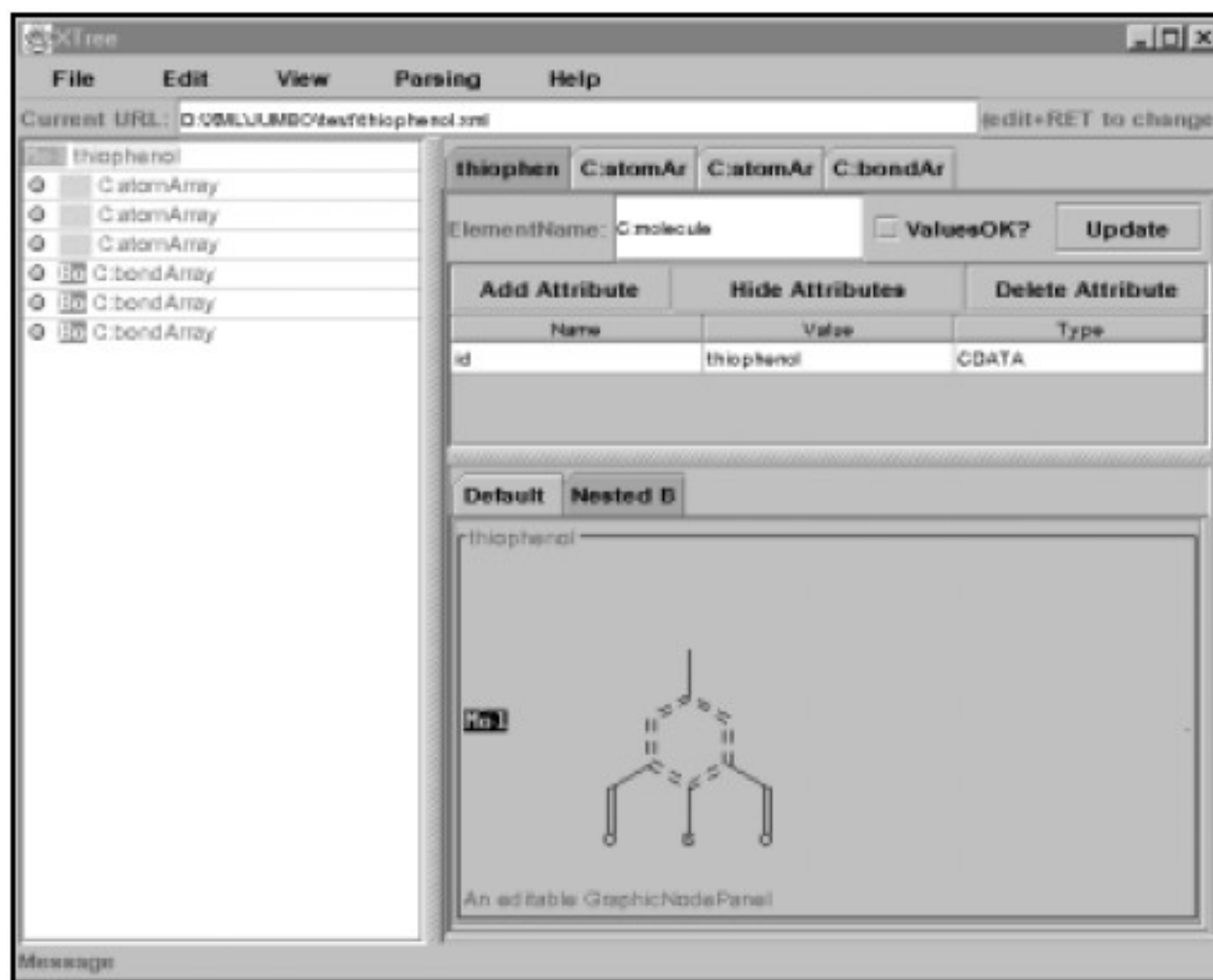
Primer XML dokumenta

- CML opis molekula H₂O

```
<CML>
  <MOL TITLE="Water">
    <ATOMS>
      <ARRAY BUILTIN="ELSYM">H O H</ARRAY>
    </ATOMS>
    <BONDS>
      <ARRAY BUILTIN="ATID1">1 2</ARRAY>
      <ARRAY BUILTIN="ATID2">2 3</ARRAY>
      <ARRAY BUILTIN="ORDER">1 1</ARRAY>
    </BONDS>
  </MOL>
</CML>
```

Primer XML dokumenta

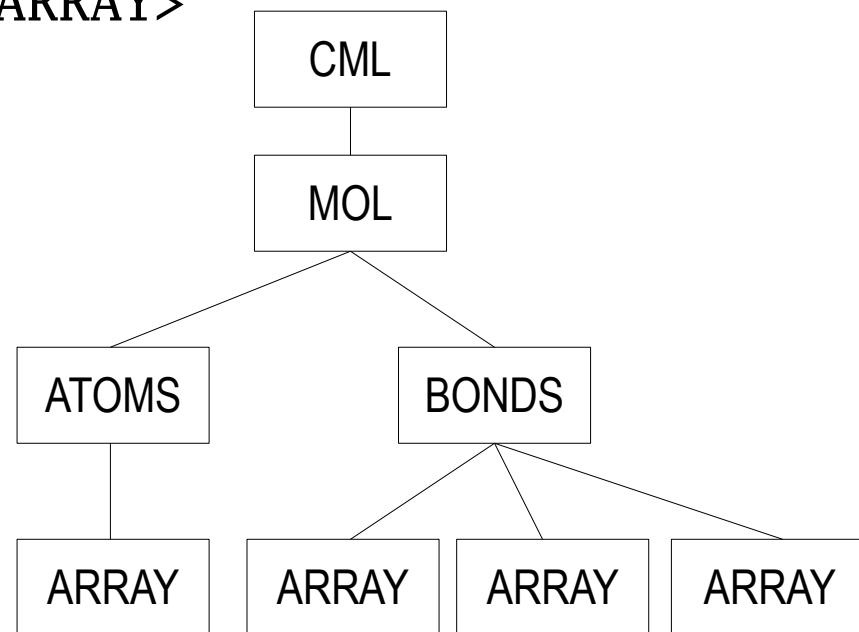
- prethodni dokument nema prikaz sam po sebi
- može se vizuelizovati na razne načine
- prikaz prethodnog dokumenta u CML browseru



Struktura XML dokumenta

- XML dokument predstavlja hijerarhiju elemenata

```
<CML>  
  <MOL TITLE="Water">  
    <ATOMS>  
      <ARRAY BUILTIN="ELSYM">H O H</ARRAY>  
    </ATOMS>  
    <BONDS>  
      <ARRAY BUILTIN="ATID1">1 2</ARRAY>  
      <ARRAY BUILTIN="ATID2">2 3</ARRAY>  
      <ARRAY BUILTIN="ORDER">1 1</ARRAY>  
    </BONDS>  
  </MOL>  
</CML>
```



Element i tag

- element je čvor u hijerarhijskoj strukturi dokumenta
 - on može sadržati druge čvorove
- tag je tekstualna oznaka (markup) za početak ili kraj elementa
 - početni (otvarajući) tag: `<tag>`
 - završni (zatvarajući tag): `</tag>`
- sadržaj elementa nalazi se između početnog i završnog taga:
 - `<tag>sadržaj</tag>`
- ako je element prazan, može da se piše dvojako:
 - `<tag/>`
 - `<tag></tag>`

Tagovi su način da se hijerarhijska struktura (stablo elemenata) prikaže pomoću linearne strukture (tekstualnog dokumenta, tj. niza slova).

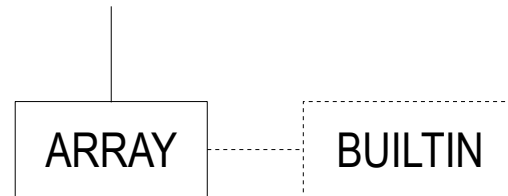
Sadržaj elementa

- sadržaj može biti
 - tekst
`<A>trlababalan`
 - podelementi
`<A>`
 `...`
 `<C>...</C>`
``
 - mešavina teksta i podelemenata
`<A>`
 `trlababalan trlbrl trlababalan`
``
 - prazan sadržaj
`<A>`
`<A/>`

Atribut

- element može da ima attribute
- atribut ima naziv i sadržaj
- sadržaj atributa je nestrukturiran tekst
- atribut se (najčešće) ne smatra posebnim čvorom stabla
- primer

<ARRAY **BUILTIN="ELSYM"**>H O H</ARRAY>



Dobro-formirani XML dokument

- #1 ima jedan korenski element
- #2 elementi se mogu ugnježdavati ali ne i preklapati
- #3 vrednosti atributa moraju biti unutar navodnika
- #4 element ne može imati dva atributa sa istim imenom
- ... *(biće još kasnije)*

Dobro-formirani XML dokument

- #1 ima jedan korenski element

```
<CONTACT>
  <NAME>Petar Petrović</NAME>
  <ADDRESS>Dunavska 1, Novi Sad</ADDRESS>
  <COMPANY>Gradska biblioteka u Novom Sadu</COMPANY>
  <TEL_NUM>444-333</TEL_NUM>
  <EMAIL>pp@yahoo.com</EMAIL>
</CONTACT>
```

- ne bi bilo ispravno:

```
<CONTACT>
...
</CONTACT>
<CONTACT>
...
</CONTACT>
```

```
neki tekst
<CONTACT>
...
</CONTACT>
još neki tekst
```

Dobro-formirani XML dokument

- #2 elementi se mogu ugnježdavati ali ne i preklapati

pravilno

`<A> `



nepravilno

`<A> `



Dobro-formiran XML dokument

- #3 vrednosti atributa moraju biti unutar navodnika

```
<telnum type="mobile">444-333</telnum>
```

```
<telnum type='mobile'>444-333</telnum>
```

- nije ispravno

```
<telnum type=mobile>444-333</telnum>
```

Dobro-formirani XML dokument

- #4 element ne može imati dva elementa sa istim imenom
- primer
`<osoba ime="..." telefon="021/..." telefon="064/...">`

Dobro-formiran XML dokument

- XML dokument mora biti dobro formiran da bi mogao biti mašinski obrađivan!



HTML je dopuštao svašta

Strukturiranje dokumenta

- šta je bolje?

<CONTACT>

<NAME>Petar Petrović</NAME>

<ADDRESS>Dunavska 1, Novi Sad</ADDRESS>

<COMPANY>Gradska biblioteka</COMPANY>

<TEL_NUM>444-333</TEL_NUM>

<EMAIL>pp@yahoo.com</EMAIL>

</CONTACT>

<CONTACT>

<NAME>

<FIRST>Petar</FIRST>

<LAST>Petrović</LAST>

</NAME>

<ADDRESS>Dunavska 1, Novi Sad</ADDRESS>

<COMPANY>Gradska biblioteka</COMPANY>

<TEL_NUM>444-333</TEL_NUM>

<EMAIL>pp@yahoo.com</EMAIL>

</CONTACT>

Strukturiranje dokumenta

- šta je bolje?

<NAME>

<FIRST>Petar</FIRST>

<LAST>Petrović</LAST>

</NAME>

- ili

<NAME FIRST="Petar" LAST="Petrović"/>

Strukturiranje dokumenta


- da li koristiti attribute ili elemente?
- nema univerzalnog odgovora
- neka pravila
 - podaci bi trebalo da budu smešteni u elementima
 - podaci o podacima (metapodaci) u atributima
 - kada nismo sigurni, onda u elementima
- podaci u atributima se ne mogu strukturirati
- elementi se mogu lakše prilagođavati izmenama u budućnosti

Strukturiranje dokumenta

- šta mogu biti metapodaci?
- primer

`<poet language="english">Homer</poet>`

`<poet language="greek">Ὅμηρος</poet>`



jezik na kome je
navedeno ime pesnika



ime pesnika

Davanje imena elementima i atributima

- ime elemenata i atributa mora biti tzv. XML ime:
 - pravi se razlika između velikih i malih slova
 - za davanje imena mogu da se koriste:
 - slova, cifre, donja crta _, crtica -, dvotačka : i tačka .
 - ime može početi slovom ili donjom crtom
 - ne može početi slovima xml (rezervisano)
 - primeri
 - <BrojMotora>123456789</BrojMotora>
 - <broj_sasije>123456789</broj_sasije>
 - <регистарскиБрој>NS 123-456</регистарскиБрој>

Ostali delovi XML dokumenta

- komentari
- procesne instrukcije
- entiteti
- CDATA sekcije

XML komentari

- navode se između `<!-- i -->`
- predstavljaju komentar autora XML dokumenta
- ignorišu se kao sadržaj, ne prikazuju se, ne obrađuju se programski
- mogu se nalaziti bilo gde izvan taga
- primer

```
<contact>
<!-- ovaj kontakt je zastareo -->
  <name>
    <first>...</first>
    ...
  ...
```

XML procesne instrukcije

- navode se između `<? i ?>`
- predstavljaju instrukcije softveru
- nisu namenjene krajnjem korisniku (čoveku)
- nisu deo sadržaja dokumenta
- mogu se nalaziti bilo gde izvan taga
- primeri
 - `<?xml version="1.0" encoding="utf-8" standalone="yes"?>`
 - `<?xml-stylesheet href="main.css" type="text/css"?>`

XML deklaracija

- XML deklaracija `<?xml ... ?>` se nalazi na početku dokumenta
- ima tri atributa
 - `version`: verzija XML jezika koja se koristi u dokumentu
 - `encoding`: kodni raspored koji se koristi u dokumentu
 - `standalone`: da li će biti neophodno učitati i DTD
- nije obavezna, ali je poželjna (obavezna od XML v1.1)
- primer
`<?xml version="1.0" encoding="utf-8" standalone="yes"?>`

Dobro-formirani XML dokument

- #5 komentari i procesne instrukcije se ne smeju nalaziti unutar taga
- primeri
`<osoba <!-- ovaj kontakt je zastareo --> ime="...">`
`<osoba <?ignore?> ime="...">`

Entiteti

- ugrađeni (unapred definisani) entiteti
 - < zamenjuje znak <
 - & zamenjuje znak &
 - > zamenjuje znak >
 - " zamenjuje znak "
 - ' zamenjuje znak '



referenca na entitet



entitet



novi entiteti se mogu posebno definisati, biće reči o tome kasnije

CDATA sekcije

- navode se između `<![CDATA[i]]>`
- sadrži tekst koji se interpretira direktno, bez zamene entiteta
- primer

```
<p>You can use a default <code>xmlns</code> attribute to  
avoid having to add the svg prefix to all your  
elements:</p>  
  <![CDATA[  
    <svg xmlns="http://www.w3.org/2000/svg"  
      width="12cm" height="10cm">  
      <ellipse rx="110" ry="130" />  
      <rect x="4cm" y="1cm" width="3cm" height="6cm" />  
    </svg>  
  ]]>
```

SVG slika ugrađena u HTML dokument;

- ako je stavimo u CDATA sekciju, biće prikazan SVG listing
- ako je ne stavimo u CDATA sekciju, biće prikazana slika