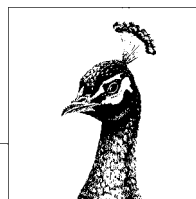


# 2

## Osnove XML-a



U ovom poglavlju naučićete kako se pišu jednostavni XML dokumenti. Saznaćete da se XML dokument sastoji od tekstualnog sadržaja obeleženog tekstualnim oznakama kao što su `<SKU>`, `<identifikator_zapisa>` i `<autor>`, koje pomalo liče na HTML oznake. Međutim, HTML sadrži oko stotinu unapred definisanih oznaka koje opisuju formatiranje Web stranice. U XML-u pravite proizvoljan broj oznaka prema svom nahođenju. Štaviše, te oznake uglavnom opisuju tip sadržaja dokumenta, a ne njegove formate i strukturu. U XML-u se ne kaže da je nešto ispisano kurzivom ili reljefno ili polucrno, nego da je to knjiga ili biografija ili kalendar.

Premda je XML opušteniji od HTML-a kada se radi o oznakama koje dopušta, mnogo je stroži što se tiče njihovog položaja i načina na koji su napisane. Konkretno, svi XML dokumenti moraju biti dobro oblikovani. Pravila dobrog oblikovanja postavljaju ograničenja kao što su „Svaka početna oznaka mora imati odgovarajuću završnu oznaku“ i „Vrednosti atributa moraju biti napisane u navodnicima“. Ta pravila se ne smeju kršiti, pa je XML dokumente nešto teže napisati, ali ih je zato lakše analizirati; oni ipak omogućavaju gotovo neograničenu fleksibilnost izražavanja.

### XML dokumenti i XML datoteke

XML dokument sadrži tekst. Pošto on nikada ne sadrži binarne podatke, može se otvoriti u svakom programu koji ume da čita tekstualne datoteke. Primer 2-1 sadrži otprilike najjednostavniji XML dokument koji se može zamisliti. Bez obzira na to, reč je o dobro oblikovanom XML dokumentu, pa ga XML analizatori mogu čitati i shvatiti (ukoliko se za računarski program može reći da nešto shvata).

*Primer 2-1. Krajnje jednostavan, ali potpun XML dokument*

```
<osoba>  
  Alen Tjuring  
</osoba>
```

U najčešćem scenariju, ovaj dokument bi predstavljao celokupan sadržaj datoteke nazvane *osoba.xml* ili možda *2-1.xml*. Međutim, XML nije sitničav u pogledu imena datoteke. Što se tiče analizatora, ova datoteka bi se mogla zvati *osoba.txt*, *osoba* ili *Ej ti, u ovoj datoteci je XML!* Možda se vašem operativnom sistemu takva imena ne bi svidela, ali se XML analizador neće buniti. Dokument uopšte ne mora biti smešten u datoteku – može i u zapis ili u polje baze podataka. Mogao ga je u letu generisati određeni CGI program kao odgovor na upit nekog čitača. S druge strane, mogao bi biti smešten u više datoteka, premda je to malo verovatno za tako jednostavan dokument. Ako se nalazi na Web serveru, verovatno će mu dodeliti MIME tip medija `application/xml` ili `text/xml`. Međutim, XML aplikacije bi mu mogle dodeliti određeniji MIME tip medija, kao što su `application/mathml+xml`, `application/xslt+xml`, `image/svg+xml`, `text/vnd.wap.wml`, pa čak i `text/html` (u veoma specijalnim slučajevima).



U generičkim XML dokumentima, prednost treba dati MIME tipu `application/xml` umesto `text/xml`, premda su mnogi Web serveri fabrički podešeni da koriste `text/xml`. Tip `text/xml` podrazumevano upotrebljava ASCII skup znakova, što je u većini XML dokumenata netačno.

## Elementi, oznake i znakovni podaci

Dokument u primeru 2-1 sadrži samo jedan *element* nazvan *osoba*. Taj element je razgraničen *početnom oznakom* (engl. *start-tag*) `<osoba>` i *završnom oznakom* (engl. *end-tag*) `</osoba>`. Sve što se nalazi između početne i završne oznake elementa (isključujući njih) naziva se *sadržaj* (engl. *content*) elementa. Sadržaj pomenutog elementa je tekst:

Alen Tjuring

Razmaci (beline) predstavljaju deo sadržaja, mada ih mnoge aplikacije zanemaruju. *Markiranje* (engl. *markup*) dokumenta čine oznake `<osoba>` i `</osoba>`. Znakovni niz „Alen Tjuring“ i razmaci koji ga okružuju jesu *znakovni podaci* (engl. *character data*). Oznaka je najčešći oblik markiranja XML dokumenta, mada postoje i druge vrste koje ćemo razmotriti kasnije.

## Sintaksa oznaka

XML oznake površno podsećaju na HTML oznake. Početna oznaka počinje znakom `<` a završava se znakom `>`; završna oznaka počinje znakom `</` a završava se sa `>`; između oznaka stoji ime elementa. Međutim, za razliku od HTML oznaka, nove XML oznake možete praviti tokom pisanja dokumenta. Da biste opisali osobu, upotrebite oznake `<osoba>` i `</osoba>`. Da biste opisali kalendar, upotrebite oznake `<kalendar>` i `</kalendar>`. Imena oznaka po pravilu odražavaju tip sadržaja unutar elementa, a ne način formatiranja tog sadržaja.

## Prazni elementi

Postoji i specijalna sintaksa za prazne elemente, one koji nemaju sadržaja. Takav element može biti predstavljen jednom *oznakom praznog elementa* (engl. *empty-element tag*), koja počinje znakom `<`, a završava se znakovima `/>`. Na primer, u jeziku XHTML, što je „XML-izovana“ verzija standardnog HTML-a, elementi prekid reda

(engl. *line break*) i horizontalna linija (engl. *horizontal rule*) opisuju se oznakama `<br />` i `<hr />`. Te oznake su ekvivalentne parovima oznaka `<br></br>` odnosno `<hr></hr>`. Sami odlučite koji oblik oznaka ćete upotrebljavati za prazne elemente. U XML-u i XHTML-u (za razliku od HTML-a) ne možete upotrebiti samo početnu oznaku – na primer, `<br>` ili `<hr>` – a da ne upotrebite i odgovarajuću završnu oznaku. To bi bila greška u pogledu dobrog oblikovanja.

### Uzimanje u obzir razlike između malih i velikih slova

Za razliku od HTML-a, XML uzima u obzir razliku između malih i velikih slova. Element `OSOBA` nije jednak elementu `osoba`, niti elementu `Osoba`. Ako element otvorite oznakom `<osoba>`, ne možete ga zatvoriti oznakom `</OSOBA>`. Možete upotrebljavati i mala i velika slova po svom izboru, ali svaki element morate dosledno ispisati.

## XML stabla

Pogledajmo nešto složeniji XML dokument. U primeru 2-2 imamo element `osoba` koji sadrži više podesno označenih podataka, kako bi se videlo njihovo značenje.

*Primer 2-2. Složeniji XML dokument koji opisuje određenu osobu*

```
<osoba>
  <ime_i_prezime>
    <ime>Alen</ime>
    <prezime>Tjuring</prezime>
  </ime_i_prezime>
  <zanimanje>naučnik u oblasti računarstva</zanimanje>
  <zanimanje>matematičar</zanimanje>
  <zanimanje>kriptograf</zanimanje>
</osoba>
```

### Roditelji i potomci

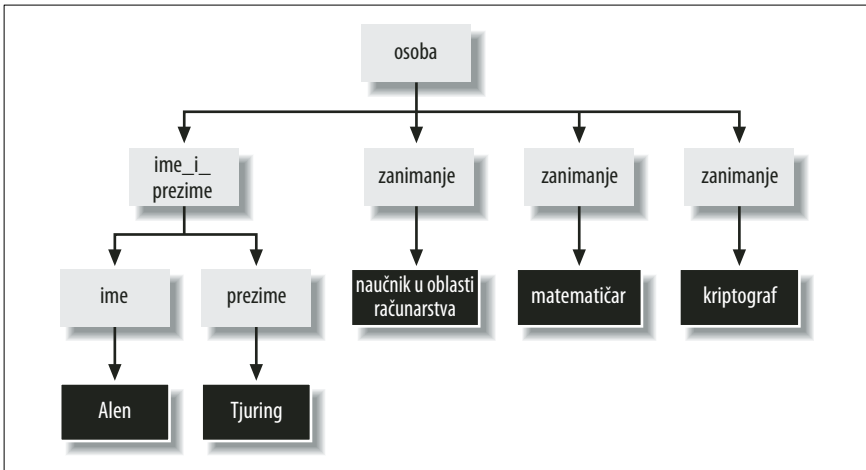
XML dokument u primeru 2-2 još uvek sadrži samo jedan element `osoba`. Međutim, sada taj element ne sadrži samo nediferencirane znakovne podatke. On ima četiri *elementa potomka* (engl. *child elements*): jedan element `ime_i_prezime` i tri elementa `zanimanje`. Element `ime_i_prezime` ima dva sopstvena elementa potomka, element `ime` i element `prezime`.

Za element `osoba` kažemo da je *roditelj* (roditeljski element) elementa `ime_i_prezime` i triju elemenata `zanimanje`. Element `ime_i_prezime` je roditelj elementa `ime` i elementa `prezime`. Katkada se kaže da su element `ime_i_prezime` i elementi `zanimanje` *braća* (engl. *siblings*) ili bratski elementi. Međusobno bratski su i elementi `ime` i `prezime`.

Kao i u ljudskom društvu, roditelji mogu imati više potomaka. Za razliku od ljudskog društva, XML svakom potomku daje tačno jednog, a ne dva ili više roditelja. Svakom elementu (uz jedan izuzetak koji ćemo uskoro objasniti) roditelj je tačno jedan element. Drugim rečima, svaki element se nalazi unutar određenog drugog elementa. Ako je početna oznaka elementa unutar određenog elementa, onda i njegova završna oznaka mora biti unutar istog elementa. U XML-u su zabranjene oznake koje se preklapaju, kao u `<strong><em>ovo je uobičajeno u HTML-u</strong></em>`. Pošto element `em` počinje unutar elementa `strong`, on se mora i završiti unutar istog elementa.

## Korenski element

Svaki XML dokument ima jedan element koji nema roditelja. To je prvi element u dokumentu i element koji sadrži sve druge elemente. U primerima 2-1 i 2-2, tu ulogu je imao element *osoba*. Njega nazivamo *korenski element* (engl. *root element*) dokumenta. Katkada ga nazivaju i *element (celog) dokumenta* (engl. *document element*). Svaki dobro oblikovan XML dokument ima jedan korenski element. Pošto se elementi ne smeju preklapati, i pošto svi elementi osim korenskog imaju jednog roditelja, XML dokumenti tvore strukturu podataka koju programeri nazivaju *stablo* (engl. *tree*). Na slici 2-1 prikazan je taj odnos za primer 2-2. Svaki sivi pravougaonik predstavlja jedan element. Svaki crni pravougaonik predstavlja znakovne podatke. Svaka strelica predstavlja odnos sadržavanja.



Slika 2-1. Dijagram stabla za primer 2-2.

## Mešovit sadržaj

U primeru 2-2, sadržaj elemenata *ime*, *prezime* i *zanimanje* bili su znakovni podaci; dakle, tekst bez ikakvih oznaka. Sadržaj elemenata *osoba* i *ime\_i\_prezime* bili su elementi potomci i razmaci (beline), koje većina aplikacija zanemaruje. Takva razlika između elemenata koji sadrže samo znakovne podatke i elemenata koji sadrže samo elemente potomke (i eventualno nekoliko razmaka – belina) uobičajena je u dokumentima sličnim zapisima. Međutim, XML se može upotrebiti i za narativne dokumente slobodnijeg oblika, kao što su poslovni izveštaji, članci za časopise, studentski eseji, kratke priče, Web stranice itd.; ilustraciju daje primer 2-3.

Primer 2-3. Narativno organizovan XML dokument

```

<biografija>
  <pasus>
    <ime_i_prezime><ime>Alen</ime> <prezime>Tjuring</prezime>
  </ime_i_prezime> bio je jedan od prvih ljudi koji su zaista zaslužili titulu
  <istaknuto>naučnika u oblasti računarstva</istaknuto>. Njegovi doprinosi
  toj oblasti previše su brojni da bismo ih ovde sve naveli,
  ali najpoznatiji su epohalni: <istaknuto>Tjuringov test</istaknuto>

```

### Primer 2-3. Narativno organizovan XML dokument (nastavak)

```
i <istaknuto>Tjuringova mašina</istaknuto>.  
</pasus>
```

```
</definicija><termin>Tjuringova mašina</termin> je do dana današnjeg  
standardan test za utvrđivanje da li je neki računar zaista inteligentan.  
Još ga nijedan računar nije položio. </definicija>
```

```
<definicija><termin>Tjuringova mašina</termin> je apstraktan automat  
koji se može nalaziti u konačno mnogo stanja, ima beskonačnu memoriju,  
i za njega se može dokazati da je ekvivalentan svakom drugom automatu  
s proizvoljno velikom memorijom, koji se može nalaziti u konačno mnogo stanja.  
Dakle, ono što važi za jednu Tjuringovu mašinu, važi za sve njih,  
bez obzira na to kako su napravljene.  
</definicija>
```

```
<pasus>  
<ime_i_prezime><prezime>Tjuring</prezime></ime_i_prezime> je bio i perfekten  
<zanimanje>matematičar</zanimanje> i <zanimanje>kriptograf</zanimanje>.  
U savezničkom razbijanju nemačkog šifratora Enigma, njegov doprinos  
je bio odlučujući. Tjuring se ubio <datum><dan>7</dan>. <mesec>juna</mesec>  
<godina>1954</godina></datum>, nakon što je bio osuđen kao homoseksualac  
i prisiljen da prima injekcije ženskih hormona.  
</pasus>
```

```
</biografija>
```

Korenski element ovog dokumenta jeste biografija. Ta biografija ima elemente potomke pasus i definicija, kao i razmake (beline). I elementi pasus i definicija sadrže druge elemente: termin, istaknuto, ime\_i\_prezime i zanimanje, a i neke neoznačene znakovne podatke. Za elemente kao što su pasus i definicija, koji sadrže elemente potomke i znakovne podatke koji nisu samo razmaci (beline), kažemo da imaju *mešovit* sadržaj (engl. *mixed content*). Mešovit sadržaj je uobičajen u XML dokumentima u kojima su članci, eseji, priče, knjige, romani, izveštaji, Web stranice i sve drugo što je organizovano kao pisano pripovedanje. Mešovit sadržaj je ređi (a s njim je i teže raditi) u računarski generisanim i obrađenim XML dokumentima, koji se upotrebljavaju za razmenu baza podataka, serijalizaciju objekata, trajne formate datoteka itd. Jedna od prednosti XML-a je to da ga lako možete prilagoditi veoma različitim zahtevima računarski generisanih dokumenata i dokumenata koje su napisali ljudi.

## Atributi

XML elementi mogu imati atribute. Atribut je par ime–vrednost pridruženo početnoj oznaci elementa. Imena atributa su od njihovih vrednosti razdvojena znakom jednakosti i opcionim razmakom. Vrednosti atributa moraju biti zatvorene u navodnike ili polunavodnike. Primera radi, sledeći element *osoba* ima atribut *rođena* čija je vrednost 1912-06-23 i atribut *umrla* čija je vrednost 1954-06-07:

```
<osoba rođena="1912-06-23" umrla="1954-06-07">  
  Alen Tjuring  
</osoba>
```

Što se tiče XML analizatora, prethodni element je identičan sledećem. U njemu su umesto navodnika upotrebljeni polunavodnici, redosled atributa je drugačiji i dodato je nekoliko razmaka oko znaka jednakosti.

```
<osoba umrla = '1954-06-07' rođena = '1912-06-23'>
  Alen Tjuring
</osoba>
```

Razmake oko znaka jednakosti svako dodaje po svom nahođenju. Polunavodnici su korisni u slučajevima kada vrednost atributa sadrži navodnik. Redosled atributa nije važan.

U primeru 2-4 prikazano je kako se atributi mogu upotrebiti za kodiranje velikog dela istih podataka navedenih u dokumentu sličnom zapisu iz primera 2-2.

*Primer 2-4. XML dokument koji osobu opisuje pomoću atributa*

```
<osoba>
  <ime_i_prezime ime="Alen" prezime="Tjuring"/>
  <zanimanje vrednost="naučnik u oblasti računarstva"/>
  <zanimanje vrednost="matematičar"/>
  <zanimanje vrednost="kriptograf"/>
</osoba>
```

Ovime se otvara pitanje treba li i kada treba za čuvanje podataka upotrebljavati elemente potomke, a kada attribute. To je predmet žestoke rasprave. Neki informatičari tvrde da su atributi podesni za metapodatke oko elementa, dok su elementi za same podatke. Drugi ukazuju na to da nije uvek očigledno šta su podaci, a šta metapodaci. Zaista, odgovor može zavistiti od toga gde se podaci koriste.

Nesporno je da svaki element može imati samo jedan atribut datog imena. Malo je verovatno da će to postati problem za datum rođenja ili smrti; moglo bi da bude problem za zanimanje, ime, adresu i sve drugo od čega element može imati više primera. Nadalje, struktura atributa je sasvim ograničena. Vrednost atributa je prosto nediferenciran tekst. Podela datuma crticama na godinu, mesec i dan u prethodnim odlomcima koda predstavlja maksimalnu podstrukturu koja je prikladna za ugradnju u atribut. Mnogo je fleksibilnija i proširivija struktura zasnovana na elementima. Uprkos tome, za neke primene atributi su sigurno podesniji. Najzad, ako sami pišete svoj XML rečnik, na vama je da odlučite kada ćete upotrebiti element, a kada atribut.

Atributi su korisni i u narativnim dokumentima, kao što je pokazano u primeru 2-5. U njemu je možda nešto očiglednije šta pripada elementima, a šta atributima. Sirov narativni tekst predstavljen je u obliku znakovnih podataka unutar elemenata. Dodatne informacije koje objašnjavaju te podatke predstavljene su u obliku atributa. Među tim informacijama su reference izvora, URL adrese slika, hiperveze, i datumi rođenja i smrti. Međutim, čak i ovo se moglo uraditi drugačije. Na primer, brojevi fusnota mogu biti atributi elementa *fusnota*, a ne znakovni podaci.

## Primer 2-5. Narativni dokument u kome su upotrebljeni atributi

```
<biografija xmlns:xlink="http://www.w3.org/1999/xlink/">

  <slika izvor="http://www.turing.org.uk/turing/pi1/busgroup.jpg"
    širina="152" visina="345"/>

  <pasus><osoba rodena='1912-06-23'
    umrla='1954-06-07'> <ime>Alen</ime>
  <prezime>Tjuring</prezime> </osoba> je bio jedan od prvih ljudi koji su
  zaista zaslužili titulu <istaknuto>naučnika u oblasti računarstva</istaknuto>.
  Njegovi doprinosi toj oblasti previše su brojni da bismo ih ovde sve naveli,
  ali najpoznatiji su epohalni:
  <istaknuto xlink:type="simple"
    xlink:href="http://cogsci.ucsd.edu/~asaygin/tt/ttest.html">Tjuringov
  test</istaknuto> i <istaknuto xlink:type="simple"
    xlink:href="http://mathworld.wolfram.com/TuringMachine.html">Tjuringova
  mašina</istaknuto>.</pasus>

  <pasus><prezime>Tjuring</prezime> je bio i perfektan <zanimanje>matematičar
  </zanimanje> i <zanimanje>kriptograf</zanimanje>. U savezničkom razbijanju
  nemačkog šifratora Enigma, njegov doprinos bio je odlučujući.
  <fusnota izvor="The Ultra Secret, F.W. Winterbotham, 1974">1</fusnota></pasus>

  <pasus>
  <prezime>Tjuring</prezime>se ubio <datum><dan>7</dan>. <mesec>juna</mesec>
  <godina>1954</godina>.</datum> nakon što je bio osuđen kao homoseksualac
  i prisiljen da prima injekcije ženskih hormona.<fusnota izvor="Alan Turing:
  the Enigma, Andrew Hodges, 1983">2</fusnota>
  </pasus>
</biografija>
```

## XML imena

Specifikacija XML-a ume da bude sitničava i izbirljiva što se tiče imena. Bez obzira na to, ona pokušava da bude efikasna kada je moguće. Jedan od načina da to postigne jeste, po mogućstvu, korišćenje istih pravila za različite stavke. Na primer, pravila za imena XML elemenata jednaka su pravilima za imena XML atributa, kao i za imena nekih ređe upotrebljivanih komponenata. Sve njih skupno nazivamo *XML imena* (engl. *XML names*).

U suštini, XML imena mogu sadržati sve alfanumeričke znakove. Među njima su standardna engleska slova od A do Z i od a do z, kao i cifre od 0 do 9. XML imena mogu sadržati i neengleska slova, brojeve i ideograme, kao što su ö, ç, Ω, 串. U imenima se mogu koristiti i sledeća tri znaka interpunkcije:

- \_ donja crta
- crtica
- . tačka

U XML imenima ne smeju se javljati drugi znakovi interpunkcije, kao što su navodnici, polunavodnici, znak za dolar, kapica (^), znak za procenat i tačka i zarez (;).

Dvotačka je dozvoljena, ali je rezervisana za prostore imena, što je objašnjeno u poglavlju 4. XML imena ne smeju sadržati beline bilo koje vrste, bez obzira na to da li se radi o razmaku, znaku za vraćanje na početak reda, znaku za prelazak u novi red, nelomivom razmaku itd. Najzad, sva imena koja počinju znakovnim nizom „XML“ (u svim kombinacijama malih i velikih slova) rezervisana su za standardizaciju u XML specifikacijama organizacije W3C.



Primarni novitet u XML-u 1.1 jeste da XML imena mogu sadržati samo znakove definisane u standardu Unicode 3.0 i njegovim novijim verzijama. XML 1.0 je ograničen na znakove definisane u standardu Unicode 2.0. XML 1.1 dozvoljava u imenima i znakove iz sledećih dodatnih pisama: burmanskog, mongolskog, Thaana, kambodžanskog, Yi i amharskog. (U XML-u 1.0 sva su ta pisma bila dozvoljena u tekstualnom sadržaju. Tada se nisu smela upotrebljavati za imena elemenata, atributa i entiteta.) XML 1.1 ne pruža gotovo ništa novo programerima koji u svojim oznakama ne upotrebljavaju navedena pisma.

XML 1.1 dozvoljava da imena sadrže i neuobičajene simbole, kao što je muzički simbol za šestostruno cimbalo, pa čak i približno milion kodova kojima još nisu dodeljeni znakovi. Međutim, korišćenje tih simbola u imenima bilo bi veoma nerazumno. Toplo vam preporučujemo da, čak i u XML-u 1.1, imena ograničite na slova, cifre, ideograme i izričito dozvoljene ASCII znakove interpunkcije.

XML imena smeju da počnu isključivo slovom, ideogramom ili znakom donja crta. Ne mogu početi cifrom, crticom niti tačkom. Dužina XML imena nije ograničena. Zato su sledeći elementi dobro oblikovani:

- `<Drivers_License_Number>98 NY 32</Drivers_License_Number>`
- `<month-day-year>7/23/2001</month-day-year>`
- `<ime_i_prezime>Alen Tjuring</ime_i_prezime >`
- `<_4-lane>I-610</_4-lane>`
- `<téléphone>011 33 91 55 27 55 27</téléphone>`
- `<персна>Га.ЛІННА°Нванов</персна>`

Sledeći elementi su neprihvatljivi:

- `<Driver's_License_Number>98 NY 32</Driver's_License_Number>`
- `<month/day/year>7/23/2001</month/day/year>`
- `<ime i prezime>Alen Tjuring</ime i prezime>`
- `<4-lane>I-610</4-lane>`

## Reference

Znakovni podaci unutar elementa ne smeju sadržati znak za manje od (<) koji nema odgovarajuću izlaznu (engl. *escape*) sekvencu (</). Znak < uvek se tumači kao početak oznake. Ukoliko vam zatreba u tekstu, pretvorite ga u izlaznu sekvencu pomoću *reference entiteta* (engl. *entity reference*) `&lt;`; *numericke reference znaka* (engl. *numeric*

*character reference* &#60; ili *hexadecimalne numeričke reference znaka* (engl. *hexadecimal numeric character reference*) &#x3C;. Kada analizator bude čitao dokument, zamene sve reference &lt;, &#60 ili &#x3C znakom <, a neće se zbuniti i protumačiti < kao početak nove oznake. Na primer:

```
<SCRIPT LANGUAGE="JavaScript">
  if (location.host.toLowerCase().indexOf("ibiblio") &lt; 0) {
    location.href="http//&&ibiblio.org/xml/"<;
  }
</SCRIPT>
```

Znakovni podaci ne smeju sadržati ni sirov znak ampersend (&) koji nema svoju izlaznu sekvencu. Taj znak se uvek tumači kao početak reference entiteta. Ovakvo se ampersend referencom entiteta &amp; pretvara u izlaznu sekvencu:

```
<company>W.L. Gore &amp; Associates</company>
```

Pošto je znak ampersend u Unicode sistemu definisan kao kôd 38, mogli smo upotrebiti i numeričku referencu znaka, &#38;:

```
<company>W.L. Gore &#38; Associates</company>
```

Reference entiteta kao što je &amp; i reference znakova kao što je &#60; spadaju u markiranja. Kada aplikacija raščlanjuje i analizira XML dokument, ta markiranja se zamjenjuju znakom ili znakovima na koje referenca upućuje. XML unapred definiše pet referenci entiteta. To su:

&lt;

Znak „manje od“, poznat i kao otvorena ugaona zagrada (<)

&amp;

Ampersend (&)

&gt;

Znak „veće od“, poznat i kao zatvorena ugaona zagrada (>)

&quot;

Ravan navodnik (")

&apos;

Ravan polunavodnik, poznat i kao apostrof (')

U sadržaju elemenata, samo se &lt; i &amp; moraju upotrebljavati umesto doslovno napisanih znakova < i &. Ostale reference su opcione. &quot; i &apos; su korisne unutar vrednosti atributa, gde bi sirov " ili ' mogao biti pogrešno protumačen kao završetak vrednosti atributa. Na primer, u sledećoj XML oznaci slike upotrebljena je referenca entiteta &apos; da bi se ubacio polunavodnik u ime O'Reilly:

```
<slika izvor='oreilly_koala3.gif' širina='122' visina='66'
  alt='Powered by O&apos;Reilly Books'
/>
```

Iako se znak „veće od“ ne može pogrešno protumačiti kao da završava oznaku koju nije trebalo da zatvori, referenca &gt; je dozvoljena najviše zbog simetrije s referencom &lt;.



Postoji jedan neobičan slučaj kada znak „veće od“ zaista morate pretvoriti u izlaznu sekvencu. U znakovnim podacima ne sme se pojaviti niz ]]>. Umesto njega morate pisati ]]&gt;.

Pored pet unapred definisanih referenci entiteta, možete koristiti i druge koje ste sami naveli u definiciji tipa dokumenta. U poglavlju 3 objasnimo kako se to radi.

Reference entiteta i znakova možete upotrebljavati samo u sadržaju elemenata i vrednostima atributa. Ne smete ih koristiti u imenima elemenata, imenima atributa, niti u drugim vrstama markiranja. Tekst kao što je `&amp;` ili `&#60;`; može se pojaviti unutar komentara ili instrukcije za obradu. Međutim, na tim mestima reference neće biti razrešene. Analizator zamenjuje samo reference koje pronade u sadržaju elemenata i vrednostima atributa. On ne prepoznaje reference na drugim mestima.

## Odeljci CDATA

Kada XML dokument sadrži uzorke XML ili HTML izvornog koda, znakovi `<` i `&` u tim uzorcima moraju biti napisani kao `&lt;` odnosno `&amp;`. Što više odeljaka s doslovno navedenim kodom dokument sadrži i što su oni duži, to kodiranje postaje zamornije. Umesto da se gnjavite s tim, svaki uzorak doslovno navedenog koda možete zatvoriti u *odeljak CDATA* (engl. *CDATA section*). Odeljak CDATA razgraničavate znakovima `<![CDATA[ i ]]>`. Sve što se zatekne između graničnika `<![CDATA[ i ]]>` tretira se kao sirov znakovni podatak. Zato u odeljku CDATA znak `<` ne otvara XML oznaku, znak `&` ne započinje referencu entiteta itd. Svi znakovi unutar odeljka CDATA tumače se kao obični znakovni podaci, a ne kao delovi markiranja.

Primeru radi, u udžbeniku za Scalable Vector Graphics (SVG), napisanom na XHTML-u, možete naići na ovako nešto:

```
<p>You can use a default <code>xmlns</code> attribute to avoid
having to add the svg prefix to all your elements:</p>
<pre><![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>
]]></pre>
```

SVG izvorni kôd uključen je neposredno u XHTML datoteku, a da nismo morali pažljivo da zamenimo svaki znak `<` referencom `&lt;`. Tako smo dobili SVG dokument, a ne ugrađenu SVG sliku, što bi se u ovom primeru desilo da SVG kôd nismo smestili u odeljak CDATA.

Jedino što se u odeljku CDATA ne sme pojaviti jeste završni graničnik odeljka CDATA, `]]>`.

Odeljci CDATA postoje kao pogodnost za programere, a ne za programe. Analizatori vas ne moraju obavestiti da li je određen blok teksta potekao iz odeljka CDATA, od normalnih znakovnih podataka ili od znakovnih podataka koji sadrže reference entiteta kao što su `&lt;` i `&amp;`. Kada vam podaci postanu dostupni, te razlike će već biti izbrisane. Kôd koji pišete ne sme zavisiti od tih razlika.

## Komentari

XML dokumenti mogu sadržati komentare, tako da koautori mogu ostavljati napomene jedni drugima i sebi, dokumentujući zašto su uradili to što su uradili i šta je još ostalo da se uradi. XML komentari su sintaktički jednaki HTML komentarima. Kao u HTML-u, komentari počinju sa `<!--` i završavaju se prvim primerkom niza `-->`. Na primer:

```
<!-- Ove hiperveze treba da proverim i ažuriram kada stignem. -->
```

Dve crtice se ne smeju pojaviti unutar komentara pre završnog `-->`. Izričito je zabranjen završetak komentara s tri crtice, `---`.

Komentare možete stavljati bilo gde unutar znakovnih podataka u dokumentu. Možete ih stavljati pre i posle korenskog elementa. (Komentari nisu elementi, pa se time ne krši XML pravilo o strukturi stabla niti ono o jedinstvenom korenskom elementu.) Međutim, komentari se ne smeju pojaviti unutar oznake ili drugog komentara.

Aplikacije koje čitaju i obrađuju XML dokumente mogu, ali ne moraju proslediti informacije iz komentara. Svaka aplikacija slobodno može da ispusti sve komentare, ako se tako sviđa njenom autoru. Ne pišite dokumente ili aplikacije koji zavise od dostupnosti komentara. Oni služe isključivo tome da sirov izvorni kôd XML dokumenta učine razumljivijim ljudima. Komentari nisu namenjeni računarskim programima. U te svrhe upotrebite *instrukciju za obradu* (engl. *processing instruction*).

## Instrukcije za obradu

U HTML-u se komentari ponekad zloupotrebljavaju za podršku nestandardnim proširenjima. Primera radi, sadržaj elementa `script` katkada se zatvara u komentar da bi se zaštitio od prikazivanja u čitaču koji ne ume da radi sa skriptovima. Web server Apache raščlanjuje i analizira komentare u `.shtml` datotekama da bi prepoznao datoteke za umetanje na serverskoj strani. Nažalost, ti dokumenti, nakon obrade u raznim HTML editorima, ponekad ne prežive s netaknutim komentarima i njima pridruženom semantikom. Što je još gore, bezazlen komentar može biti pogrešno protumačen kao ulaz u aplikaciju.

Kao alternativno sredstvo prosleđivanja informacija određenim aplikacijama koje će čitati dokument, XML ima *instrukciju za obradu*. Instrukcija za obradu počinje sa `<?`, a završava sa `?>`. Neposredno posle `<?` dolazi XML ime *cilja*, što može biti ime aplikacije kojoj je ta instrukcija namenjena ili identifikator instrukcije. Ostatak instrukcije za obradu sadrži tekst u formatu koji odgovara ciljnim aplikacijama.

Na primer, u HTML-u se robotskom oznakom `META` saopštava pretraživačima i ostalim robotima treba li i kako da indeksiraju stranicu. Za XML dokumente predložena je kao ekvivalentna sledeća instrukcija za obradu:

```
<?robots index="yes" follow="no"?>
```

Cilj ove instrukcije za obradu je `robots`. Sintaktički, ova instrukcija sadrži dva pseudoatributa, jedan nazvan `index`, a drugi `follow`. Njihove vrednosti su `yes` ili `no`. Semantika ove instrukcije je sledeća: ako atribut `index` ima vrednost `yes`, onda bi

roboti pretraživača trebalo da indeksiraju stranicu. Slično tome, ako atribut `follow` ima vrednost `yes`, onda će roboti pretraživača posetiti lokacije na koje upućuju hiperveze stranice; ukoliko ima vrednost `no`, to neće biti učinjeno.

Druge instrukcije za obradu mogu imati potpuno različite sintakse i semantike. Primera radi, instrukcije za obradu mogu sadržati praktično neograničenu količinu teksta. PHP smešta velike programe u instrukcije za obradu. Na primer:

```
<?php
mysql_connect("database.unc.edu", "clerk", "password");
$result = mysql("HR", "SELECT LastName, FirstName FROM Employees
ORDER BY LastName, Firstname");
$i = 0;
while ($i < mysql_numrows ($result)) {
    $fields = mysql_fetch_row($result);
    echo "<person>$fields[1] $fields[0] </person>\r\n";
    $i++;
}
mysql_close( );
?>
```

Instrukcije za obradu spadaju u označavanje, ali nisu elementi. Stoga ih možete pisati na bilo kom mestu u XML dokumentu osim unutar oznaka, kao komentare. Smete ih pisati i pre i posle korenskog elementa. Najčešća instrukcija za obradu, `xml-style-sheet`, dokumentu pridružuje opis stilova. Ona se uvek piše pre korenskog elementa, kao u primeru 2-6. U njemu instrukcija za obradu `xml-stylesheet` saopštava čitaču da na dokument, pre nego što ga prikaže čitaocu, primeni CSS opis stilova `osoba.css`.

*Primer 2-6. XML dokument sa instrukcijom za obradu smeštenom u prolog*

```
<?xml-stylesheet href="osoba.css" type="text/css"?>

<osoba>
    Alen Tjuring
</osoba>
```

Da bi se izbegla zabuna s deklaracijom XML-a, zabranjene su instrukcije za obradu nazvane `xml`, `XML`, `Xml` itd., u bilo kojoj kombinaciji malih i velikih slova. Instrukcijama za obradu možete davati sva druga imena koja zadovoljavaju pravila XML-a.

## Deklaracija XML-a

XML dokumenti bi trebalo da otpočinu deklaracijom XML-a (ali ne moraju). Deklaracija XML-a izgleda kao instrukcija za obradu nazvana `xml`, koja sadrži pseudoatribute `version`, `standalone` i `encoding`. Strogo uzev, to nije instrukcija za obradu, nego deklaracija XML-a – ni više ni manje od toga. Ilustraciju daje primer 2-7.

*Primer 2-7. Veoma jednostavan XML dokument s deklaracijom XML-a*

```
<?xml version="1.0" encoding="ASCII" standalone="yes"?>
<osoba>
    Alen Tjuring
</osoba>
```

XML dokumenti ne moraju imati deklaraciju XML-a, ali ako je imaju, ona mora biti prva stavka dokumenta. Pre nje ne sme biti komentara, razmaka (belina), instrukcija za obradu itd. Razlog za to je što XML analizador na osnovu prvih pet znakova (`<?xml`) zaključuje kakvo je kodiranje znakova u dokumentu – recimo, da li je upotrebljen jednobajtni ili višebajtni skup znakova. Pre deklaracije XML-a sme biti samo nevidljiva Unicode oznaka redosleda bajtova. Razmotrićemo to ponovo u poglavlju 5.

## Atribut version

Atribut `version` bi trebalo da ima vrednost 1.0. Pod veoma neuobičajenim okolnostima možete mu dati vrednost 1.1. Pošto zadavanje verzije 1.1 ograničava dokument na najnovije verzije malog broja analizatora, a svi analizatori za XML 1.1 moraju podržavati i XML 1.0, ne bi trebalo da olako zadajete verziju 1.1.

Ne verujete? Najpre odgovorite na nekoliko pitanja:

1. Govorite li burmanski, mongolski, kambodžanski, amharski ili divehi?
2. Sadrže li vaši podaci zastarele, netekstualne C0 kontrolne znakove kao što su vertikalni tabulator, prelazak na novu stranicu ili znak za zvonce?

Ukoliko ste na oba pitanja odgovorili „ne“, korišćenjem XML-a 1.1 ne dobijate apsolutno ništa. Ako ste na jedno pitanje odgovorili „da“, možda imate razloga za korišćenje XML-a 1.1. XML 1.0 dozvoljava da se burmanski, mongolski, kambodžanski itd. upotrebljavaju u znakovnim podacima i vrednostima atributa. XML 1.1 dozvoljava da se ta pisma upotrebljavaju i u imenima elemenata i atributa, što XML 1.0 ne dopušta. XML 1.1 dozvoljava i da se C0 kontrolni znakovi (sem znaka null) upotrebljavaju u znakovnim podacima i vrednostima atributa (ukoliko su pretvoreni u numeričke reference znakova poput `&#x07;`), što XML 1.0 ne dopušta. Ako ijedan od ova dva uslova važi za vas, mogli biste upotrebiti XML 1.1 (premda bi trebalo da budete svesni da time znatno sužavate potencijalnu publiku svog XML dokumenta). U protivnom, trebalo bi da upotrebljavate isključivo XML 1.0.

## Atribut encoding

Dosad smo bili neodređeni u pogledu skupova znakova i kodiranja znakova. Rekli smo da su XML dokumenti sastavljeni od čistog teksta, ali nismo rekli kako su znakovi tog teksta kodirani. Po standardu ASCII? Latin-1? Unicode? Nekom četvrtom?

Kratak odgovor na ovo pitanje je „da“. Dugačak odgovor je da su XML dokumenti podrazumevano kodirani UTF-8 kodovima promenljive dužine u skupu znakova Unicode. Pošto se radi o nadskupu skupa znakova ASCII, tekstualne datoteke napisane u čistom ASCII-ju automatski su kodirane i po standardu UTF-8. Međutim, većina programa za obradu XML-a (naročito oni napisani na Javi) mogu obrađivati mnogo veći broj skupova znakova. Analizatoru treba reći samo koji je standard za kodiranje znakova upotrebljen u dokumentu. To je najbolje uraditi preko metapodataka, koji su smešteni u sistem datoteka ili ih daje server. Međutim, ne pružaju svi sistemi podatke o skupu znakova, pa XML dozvoljava dokumentima da sami naznače svoj skup znakova pomoću *deklaracije kodiranja* (engl. *encoding declaration*) unutar deklaracije XML-a. U primeru 2-8 pokazano je kako biste naznačili da je dokument napisan u skupu znakova ISO-8859-1 (Latin-1), koji obuhvata i znakove kao što su ö i ç, potrebne u mnogim zapadnoevropskim jezicima.

Primer 2-8. XML dokument kodiran u skupu znakova Latin-1

```
>?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<osoba>
  Erwin Schrödinger
</osoba>
```

U deklaraciji XML-a atribut `encoding` je opcion. Ako se on izostavi, a metapodaci su nedostupni, analizator pretpostavlja da je upotrebljen skup znakova Unicode. Na osnovu prvih nekoliko bajtova datoteke, analizator može pokušati da utvrdi koje je Unicode kodiranje upotrebljeno. Ako su metapodaci dostupni, ali su suprotni deklaraciji o kodiranju, onda analizator veruje metapodacima. Na primer, ukoliko HTTP čitač kaže da je dokument kodiran u ASCII-ju, a deklaracija o kodiranju kaže da je kodiran po standardu UTF-8, analizator će izabrati ASCII.

Razna kodiranja i pravilno rukovanje neengleskim XML dokumentima razmotrićemo detaljnije u poglavlju 5.

## Atribut `standalone`

Ukoliko atribut `standalone` (samostalan) ima vrednost `no`, onda aplikacija može učitati spoljni DTD (tj. DTD smešten u neku drugu datoteku, a ne u onu koja sadrži dokument) da bi utvrdila prave vrednosti određenih delova dokumenta. Primera radi, DTD može sadržati podrazumevane vrednosti atributa koje analizator treba da prijavi, premda one u dokumentu ne postoje.

Dokument koji nema DTD, a takvi su svi dokumenti u ovom poglavlju, može imati `yes` kao vrednost atributa `standalone`. I dokument koji ima DTD može imati `yes` kao vrednost atributa `standalone`, ukoliko taj DTD ni na koji način ne menja sadržaj dokumenta ili je DTD potpuno interni. Pojednosti o dokumentima s DTD-ovima objašnjene su u poglavlju 3.

U deklaraciji XML-a, atribut `standalone` je neobavezan. Ako je izostavljen, pretpostavlja se da ima vrednost `no`.

## Provera dobre oblikovanosti dokumenta

Svaki XML dokument, bez izuzetka, mora biti dobro oblikovan. To znači da mora zadovoljiti više pravila, među kojima i sledeća:

1. Svaka početna oznaka mora imati odgovarajuću završnu oznaku.
2. Elementi mogu biti ugnežđeni, ali se ne smeju preklapati.
3. Mora postojati tačno jedan korenski element.
4. Vrednosti atributa moraju biti zatvorene u navodnike.
5. Element ne sme imati dva istoimena atributa.
6. Unutar oznaka ne sme biti komentara i instrukcija za obradu.
7. U znakovnim podacima elemenata i atributa ne sme biti znakova `<` ili `&` koji nisu pretvoreni u izlaznu sekvencu.

Ovo nije celokupna lista pravila. Dokument može biti loše oblikovan na veoma mnogo načina. Celokupna lista pravila navedena je u poglavlju 21. Neka od njih se odnose na strukture koje još nismo razmatrali, kao što su DTD-ovi. Druga pravila ćete veoma retko kršiti ako budete sledili primere iz ovog poglavlja (na primer, ne-  
mojte stavljati razmak između početnog < i imena elementa u oznaci).

Bez obzira na to da li je greška mala ili velika i da li se sreće retko ili često, XML anali-  
zator koji čita dokument mora da je prijavi. Analizator može, ali ne mora da prijavi  
sve greške u dobrom oblikovanju koje pronađe u dokumentu. Međutim, analizator  
ne sme pokušati da popravi dokument i da doda ono što smatra da je autor doku-  
menta izostavio. Ne sme dodati izostavljene navodnike oko vrednosti atributa ili izo-  
stavljenu završnu oznaku, niti sme zanemariti komentar napisan unutar početne  
oznake. Analizator mora prijaviti grešku. Tako je napravljeno da bi se izbegli ratovi  
kompatibilnosti („vaš čitač prijavljuje 5 naših grešaka, pa će i naš čitač prijavljivati 5  
vaših grešaka“), koji čitače Weba prate od njihovih početaka do danas. Zato proverite  
da li je vaš XML dobro oblikovan pre nego što ga objavite, bez obzira na to da li se  
radi o Web stranici, ulazu u bazu podataka ili nečem trećem.

Najjednostavniji način da to uradite jeste da učitate dokument u čitač Weba koji  
ume da radi s XML dokumentima, kao što je Mozilla. Ukoliko je dokument dobro  
oblikovan, čitač će ga prikazati. Ako nije, prikazaće poruku o grešci.

Umesto učitavanja dokumenta u čitač Weba, možete neposredno upotrebiti XML  
analizator. Većina XML analizatora nije namenjena krajnjim korisnicima. Oni su za-  
pravo biblioteke klasa projektovane za ugradnju u program koji se lakše koristi, kao  
što je Mozilla. Njihov interfejs komandne linije je minimalan, ako uopšte postoji; če-  
sto nije dobro dokumentovan. Bez obzira na sve to, ponekad je brže provući grupu  
datoteka kroz interfejs komandne linije nego ih jednu po jednu učitati u čitač  
Weba. Nadalje, kada naučite da radite s DTD-ovima i šemama, iste alatke moći ćete  
da upotrebljavate za proveru validnosti dokumenata, što većina čitača Weba ne radi.

Postoji mnogo XML analizatora dostupnih u raznim jezicima. Ovde ćemo prikazati  
proveru dobre oblikovanosti analizatorom *libxml* kompanije Gnome Project, koji  
možete preuzeti na lokaciji <http://xmlsoft.org>. Taj paket otvorenog izvornog koda na-  
pisan je na prilično prenosivom C-u i radi na većini glavnih platformi, uključujući  
Windows, Linux i Mac OS X. (Unapred je instaliran u mnogim distribucijama Linu-  
xa.) Postupak bi trebalo da bude sličan i s drugim analizatorima, premda se pojeđi-  
nosti mogu razlikovati.

*libxml* je zapravo biblioteka, ali sadrži i program *xmllint* koji pomoću te biblioteke  
proverava dobru oblikovanost datoteka. *xmllint* se pokreće pisanjem njegovog ime-  
na u Unixovom komandnom okruženju (engl. *Unix shell*) ili posle DOS-ovog odziv-  
nika (engl. *DOS prompt*), kao i svaki drugi program koji ima komandnu liniju.  
Njegovi argumenti su URL adrese ili imena datoteka dokumenta koji treba proveriti.  
Evo rezultata koje je *xmllint* dao za jednu od ranijih verzija primera 2-5. Već prvi  
red ispisa saopštava gde je prvi problem u datoteci:

```
% xmllint 2-5.xml
2-5.xml:5: error: Unescaped '<' not allowed in attribute values
  <osoba rodena='1912/06/23'
  ^
2-5.xml:5: error: attributes construct error
  <osoba rodena='1912/06/23'
  ^
```

```

2-5.xml:5: error: error parsing attribute name
  <osoba rodena='1912/06/23'
  ^
2-5.xml:5: error: attributes construct error
  <osoba rodena='1912/06/23'
  ^
2-5.xml:5: error:xmlParseStartTag: problem parsing attributes
  <osoba rodena='1912/06/23'
  ^
2-5.xml:5: error: Coludn't find end of Start Tag image line 3
  <osoba rodena='1912/06/23'
  ^

```

Kao što vidite, analizator je pronašao grešku. U ovom slučaju, poruka o grešci nam nije naročito pomogla. Stvarni problem nije bio u tome što je vrednost atributa sadržala znak <, nego u tome što je izostavljen završni navodnik u vrednosti atributa *visina*. Ipak smo pomoću datih podataka uspjeli da pronađemo i otklonimo problem. Uprkos dugačkom ispisu, *xmllint* je prijavio samo prvu grešku u dokumentu, pa ćete morati da ga pokrećete više puta dok ne pronađete i ne ispravite sve greške. Kada smo primer 2-5 popravili tako da bude dobro oblikovan, *xmllint* je samo odštampao datoteku koju je pročitao:

```

% xmllint 2-5.xml
<biografija xmlns:xlink="http://www.w3.org/1999/xlink/">

  <slika izvor="http://www.turing.org.uk/turing/pi1/busgroup.jpg"
  širina="152" visina="345"/>

  <pasus><osoba rodena='1912-06-23'
  umrla='1954-06-07'> <ime>Alen</ime>
...

```

Pošto je dokument ispravljen i dobro oblikovan, možete ga proslediti čitaču Weba, bazi podataka ili nekom drugom programu koji ga očekuje. Gotovo svi netrivialni ručno pisani dokumenti na početku nisu dobro oblikovani, pa je važno da proverite šta ste uradili pre nego što to objavite.