

Validacija - DTD (engl. Document Type Definition)

Koja je svrha uvođenja DTD?

1. Da bi se uvela **ograničenja** tj. opis strukture podataka, a time dobile nove mogućnosti. Opis strukture je javan i potpuno razumljiv tj. standardizovan
 2. **Automatizacija** provere ispravnosti dokumenta.
 3. **Višestruka upotreba** definisanih delova dokumenta.
- DTD je dodatni dokument koji može biti pridružen nekom XML dokumentu i pruža gore navedene osobine.

Definicija DTD?

- ▶ Skraćenica od engl. naziva: **Document Type Definition**
- ▶ DTD definiše sledeće:
 1. **elemente** koji se pojavljuju u dokumentu
 2. **broj elemenata** u dokumentu
 3. **način ugnježdavanja**
 4. **skup** dozvoljenih, zahtevanih i podrazumevanih **atributa**

▶ 3

18:11

Šta DTD može da deklarise?

- ❑ **Tip elementa**
 - ▶ Opisuje tip elementa, kao i tipove podataka koje on sadrži.
- ❑ **Tip atributa**
 - ▶ Opisuje tip atributa, kao i tipove podataka koje on može da sadrži.
- ❑ **Listu atributa**
 - ▶ Elementi imaju attribute navedene u listi. Liste omogućavaju grupisanje svih srodnih atributa elementa.

▶ 4

18:11

Šta DTD **ne može** da deklariše?

Na primer:

- ❑ Koji je korenski element.
- ❑ Koliko primeraka elemenata pojedinih vrsta ima u dokumentu.
- ❑ Kako izgledaju znakovni podaci unutar elemenata.
- ❑ Semantiku, tj. značenje elemenata; na primer, da li određeni element sadrži datum ili ime neke osobe.

► 5

18:11

Jedan DTD fajl

```
<person>
  <name>
    <first_name>Alan</first_name>
    <last_name>Turing</last_name>
  </name>
  <profession>computer scientist</profession>
  <profession>mathematician</profession>
  <profession>cryptographer</profession>
</person>
```

```
<!ELEMENT person (name, profession*)>
<!ELEMENT name (first_name, last_name)>
<!ELEMENT first_name (#PCDATA)>
<!ELEMENT last_name (#PCDATA)>
<!ELEMENT profession (#PCDATA)>
```

► 6

18:11

Kako se DTD pridružuje XML fajlu?

Na više načina:

- Kao **poseban fajl** u odnosu na XML
 - Može biti naknadno više korišćen
- Kao **deo XML dokumenta**
- **kombinacijom** prethodna dva slučaja, tj. deo DTD-a je u posebnom dokumentu, a deo u XML dokumentu

<!DOCTYPE root-element [element-declarations]>

► 7

18:11

Spoljašnji DTD

XML i DTD su zasebni dokumenti, prosleđuje se URI na kome je DTD fajl

hello.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE message SYSTEM
    "http://www.viser.edu.rs/dtd/message.dtd">
<poruka>
    <tema>Opis DTD formata!</tema>
    <potpis>Perica P!</potpis>
</poruka>
```

http://www.viser.edu.rs/dtd/message.dtd

```
<!ELEMENT poruka (tema, potpis)>
<!ELEMENT tema (#PCDATA)>
<!ELEMENT potpis (#PCDATA)>
```

► 8

18:11

Relativna putanja do fajla

- ▶ Ako se dokument nalazi na istoj osnovnoj lokaciji kao i DTD, može se koristiti **relativna putanja** (adresa) umesto apsolutne, na primer:
 - ▶ `<!DOCTYPE message SYSTEM "/dtd/message.dtd">`
- ▶ Ako se dokument nalazi u istom direktorijumu može se zadati samo ime datoteke:
 - ▶ `<!DOCTYPE message SYSTEM "message.dtd">`

▶ 9

18:11

Javni identifikatori

`<!DOCTYPE root_element PUBLIC "publicID" "OpcioniURL">`

- ▶ Standardni DTD-ovi mogu biti smešteni na više URL adresa. Ime javnog identifikatora (*public ID*) jednoznačno određuje XML aplikaciju za koju se koristi. U isto vreme se navodi i rezervna URL adresa, za slučaj da analizator validnosti ne prepozna javni identifikator.
- ▶ *Primer: javni DTD Rich Site Summary (Netscape)*
- ▶ `<!DOCTYPE rss PUBLIC "-//Netscape Communications//DTD RSS 0.91//EN" "http://my.netscape.com/publish/formats/rss-0.91.dtd">`

▶ 10

18:11

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

Ugrađeni DTD

XML i DTD su u istom dokumentu

hello.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE poruka [
  <!ELEMENT poruka (tema, potpis)>
  <!ELEMENT tema (#PCDATA)>
  <!ELEMENT potpis (#PCDATA)>
]>
<poruka>
  <tema>Opis DTD formata!</tema>
  <potpis>Perica P!</potpis>
</poruka>
```

Kombinacija spoljašnji-ugrađeni DTD

DTD je jednim delom ugrađen u XML dokument, a drugim delom je u posebnoj fajlu

```
hello.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE poruka SYSTEM "message.dtd" [
  <!ELEMENT potpis (#PCDATA)>
  <!-- preklapa potpis u message.dtd -->
]>
<poruka>
  <tema>Opis DTD formata!</tema>
  <potpis>Perica P!</potpis>
</poruka>

message.dtd
<!ELEMENT poruka(tema,farewell)>
<!ELEMENT tema (#PCDATA)>
<!ELEMENT potpis EMPTY>
```

▶ 13

18:11

Blokovi koji se definišu preko DTDa

- ▶ Elementi
- ▶ Atributi
- ▶ Entiteti
 - ▶ Neki karakteri imaju specijalno značenje. (<,>,&...). Entiteti se ekspanuju kada se neki dokument parsira.
- ▶ PCDATA (engl. *Parsed Character Data*)
 - ▶ Tekst koji će biti procesiran od parsera. Tekst će biti analiziran kao markap celine.
- ▶ CDATA
 - ▶ *Character Data*.
 - ▶ Označava tekst koji neće biti parsiran. Tagovi u okviru ovog teksta neće biti tretirani kao markap.

▶ 14

18:11

Deklarisanje elemenata

► **<!ELEMENT ime (sadrzaj)>**

ime – ime elementa
sadrzaj – podaci koje element može da sadrži

Primer:

DTD

```
<!ELEMENT pozdrav (#PCDATA)>
```

Validni XML kod

```
<pozdrav>Hello World!</greeting>

<pozdrav>
  <![CDATA[G'day!]]>
</pozdrav>
```

► 15

18:11

Vrste/tipovi sadržaja elementa

- ❖ prazan element - **EMPTY**
- ❖ bilo koji element - **ANY**
- ❖ **tačno definisane elemente**
- ❖ **tekst (#PCDATA)**
- ❖ **mešavinu**

EMPTY

Ne sme da sadrži nikakve podatke. **Može da ima samo attribute.**

Primer:

DTD deklaracija: `<!ELEMENT NewLine EMPTY>`

Ispravan XML kod: `< NewLine ></ NewLine > ili < NewLine />`

► 16

18:11

ANY

Svi elementi koji su korektno definisani mogu da budu sadržani u elementu čiji je sadržaj podataka **ANY**. Elementi koji se koriste moraju biti deklarirani u DTD-u.

Primer: DTD

```
<!ELEMENT grupa ANY>
<!ELEMENT student (#PCDATA)>
<!ELEMENT nastavnik (#PCDATA)>
```

XML kod:

```
<grupa/> ili <grupa></grupa>
<grupa>pera, luka, mika</grupa>
<grupa>
  <student>joca</student>
  <nastavnik>milan</nastavnik>
</grupa>
```

► 17

18:11

Definisanje strukture

Definiše niz elemenata deklariranih na sledeći način:

```
<!ELEMENT ime (struktura)>
```

Element je definisan imenom (**ime**) i strukturom elemenata koji su njegovi podelementi tj. članovi.

Tipovi članstva u strukturi elementa:

Sekvenca	<!ELEMENT ime (a,b)>
Izbor	<!ELEMENT ime (a b)>
Jedan	<!ELEMENT ime (a)>
Jedan ili više	<!ELEMENT ime (a) +>
Nula ili više	<!ELEMENT ime (a) *>
Nula ili jedan	<!ELEMENT ime (a) ?>

a i b mogu imati složeniju strukturu, kao na primer a = (c,d)

► 18

18:11

▶ Sekvenca (a,b,c)

- ▶ Navedeni elementi moraju se pojaviti **u zatom redosledu**.

▶ Broj potomaka

- ▶ ? Dozvoljeno **nula ili jedan** element
- ▶ * Dozvoljeno **nula ili više** elemenata
- ▶ + Dozvoljeno **jedan ili više** elemenata

▶ Izbor |

- ▶ Spisak elemenata razdvojenih vertikalnom crtom. Mora sadržati jedan od ponuđenih elemenata. Ne može

▶ 19 sadržati više od jednog ponuđenog.

18:11

▶ Primer:

- ▶ Treba saopštiti da element *krug* sadrži element *centar* i element *prečnik* ili *poluprečnik*, ali ne oba.

<!ELEMENT krug (centar, (prečnik | poluprečnik)) (#PCDATA)>

▶ 20

18:11

Primer: DTD deklaracija:

```
<!ELEMENT person ((fname,lname)|((lname,fname)))>
<!ELEMENT fname (#PCDATA)>
<!ELEMENT lname (#PCDATA)>
```

Ispravan XML kod:

```
<person>
  <lname>Smith</lname>
  <fname>John</fname>
</person>
ili
<person>
  <fname>John</fname>
  <lname>Smith</lname>
</person>
```

► 21

18:11

DTD:

```
<!ELEMENT order (order-item+,delivery-address,order-date?)>
```

XML kod 2 deo:

```
<order>
  <order-item>item3</order-item>
  <order-item>item4</order-item>
  <delivery-address>123 State Street</delivery-address>
</order>
ili
<order>
  <order-item>item5</order-item>
  <order-item>item6</order-item>
  <delivery-address>123 State Street</delivery-address>
  <order-date>July 5, 2001</order-date>
</order>
```

► 22

18:11

Primer

DTD deklaracija:

```
<!ELEMENT phonebook (page)+>
<!ELEMENT page (heading, (entry|advert)+)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT entry (#PCDATA)>
<!ELEMENT advert (#PCDATA)>
```

Ispravan XML:

```
<phonebook>
  <page>
    <heading>The whole town</heading>
    <entry>John Smith, 555-1212</entry>
    <advert>Fred's Fish n' Chips - 123-4567</advert>
  </page>
</phonebook>
```

Neispravan XML:

```
<phonebook><page><entry/><entry/></page></phonebook>
iii
<phonebook><page/></phonebook>
```

Mixed

Opisuje elemente koji sadrže tekstualne podatke i/ili druge elemente.

Primer DTD deklaracije:

```
<!ELEMENT product (#PCDATA)> (karakter)
<!ELEMENT review (#PCDATA | product)*> (karakter + dete element)
<!ELEMENT review (#PCDATA | name | proffesion | note)*>
```

Ispravan XML kod:

```
<review>review text goes here</review>
<review>
  This is a review of some
  <product>car</product>
  that goes on for pages of regular text.
</review>
```

Definisanje atributa

ATTLIST se koristi za deklarisanje atributa elementa.

Dve moguće sintakse:

```
<!ATTLIST element atribut tip opis>
```

```
<!ATTLIST element
    atribut tip opis
    ...
    atribut tip opis >
```

tip je definisani tip atributa.

opis je opis sadržaja / vrednosti atributa.

Primer:

```
▶ <!ATTLIST img source CDATA #REQUIRED>
```

1. Element **img** ima atribut **source**.
2. Tip podataka atributa su znakovni podaci (CDATA).
3. Svi primerci elementa *img* moraju imati neku vrednost za atribut *source*.

- ▶ Jedna deklaracija ATTLIST može deklarirati više atributa istog elementa primenom druge sintakse. Na primer:

```
<!ATTLIST img source      CDATA #REQUIRED
              width       CDATA #REQUIRED
              height      CDATA #REQUIRED
              alt          CDATA #IMPLIED >
```

- ❖ Atribut alt je neobavezan i može se izostaviti u nekim elementima.
- ❖ Ova deklaracija ima isti efekat i značenje kao 4 zasebne deklaracije ATTLIST, po jedna za svaki atribut.

Tipovi atributa: lista

CDATA	Vrednost je „character data“
(en1 en2 ..)	Vrednost mora biti jedna iz liste
ID	Vrednost je jedinstvena - id
IDREF	Vrednost je id nekog drugog elementa
IDREFS	Vrednost je lista drugih ids
NMTOKEN	Vrednost je validno XML ime
NMTOKENS	Vrednost je lista validnih XML imena
ENTITY	Vrednost je entitet
ENTITIES	Vrednost je lista entiteta
NOTATION	Vrednost je naziv notacij
xml:	Vrednost je neka predefinisana xml vrednost

Tip atributa: CDATA

- ▶ Znači da kao vrednost može sadržati proizvoljan tekstualni niz prihvatljiv u dobro formiranom XMLu.
- ▶ Najopštiji tip atributa.

▶ 29

18:11

Tip atributa: NMTOKEN (XML imenski token)

- ▶ Sme da sadrži iste znakove kao i XML ime, tj. alfanumeričke i/ili ideografske znakove i znakove interpunkcije `_`, `-`, `.` i `:`
- ▶ Ne sme sadržati beline.
- ▶ Sme početi svim znakovima (za razliku od xml imena). Na primer `12` i `.chcrs` su validni tokeni.

▶ 30

18:11

Tip atributa: NMTOKENS

- ▶ Sadrži jednu ili više XML imenskih tokena razdvojenih belinom.

▶ 31

18:11

Nabrajanje kao način definisanja tipa atributa

- ▶ `<!ATTLIST datum mesec (januar | februar | mart | april | maj | jun | jul | avgust | septembar | oktobar | novembar | decembar) >`
- ▶ Željene vrednosti su ujedno XML imenski tokeni. Ovo ne bi bilo uspešno ako bi želeli da imena imaju praznine ili neki drugi znak interpunkcije osim donje crte, crtice, dvotačke i tačke.

▶ 32

18:11

Tip atributa: ID

- ▶ Atribut tipa ID mora sadržati XML ime (ne imenski token, već ime) jedinstveno unutar XML dokumenta.
- ▶ Nijedan drugi atribut tipa ID u dokumentu ne može imati istu vrednost.
- ▶ Atribut tipa ID dodeljuje elementu jedinstveno ime.
- ▶ Primer:
- ▶ `<!ATTLIST radnik jmbg ID #REQUIRED >`

▶ 33

18:11

Tip atributa: IDREF

- ▶ Upućuje na atribut tipa ID nekog elementa u dokumentu.
- ▶ Mora biti XML ime.
- ▶ UPUĆUJE NA ATRIBUT TIPRA ID NEKOG ELEMENTA U

```
<projekat id="p1">
  <naziv>akreditacija</naziv>
  <clan osoba="p2" />
  <clan osoba="a123451234" />
</projekat>
<projekat id="p2">
  <naziv>samovrednovanje</naziv>
  <clan osoba="a121212121" />
  <clan osoba="a343434343" />
</projekat>
<radnik jmbg="a343434343">
  <ime>Pera Peric</ime>
</radnik>
<radnik jmbg="a123456789">
  <ime>Pera Peric</ime>
</radnik>
```

```
<!ATTLIST radnik jmbg ID #REQUIRED>
<!ATTLIST projekat id ID #REQUIRED>
<!ATTLIST clan osoba IDREF #REQUIRED>
```

- ▶ Ove deklaracije prisiljavaju atribut osoba da sadrži vrednost koja odgovara nekom atributu u dokumentu koji je tipa ID. Formalno bi to mogao da bude i ID projekta, a ne samo osobe?!

34

18:11

Tip atributa: ENTITY, ENTITIES

- ▶ ENTITY sadrži ime neraščlanjenog (koji se ne parsira) entiteta deklarisanog na drugom mestu DTDa. Na primer, element *film* može imati atribut ENTITY koji identifikuje MPEG ili QuickTime datoteku koju treba reprodukovati.
- ▶ ENTITIES sadrži imena jednog ili više neraščlanjenih entiteta, razdvojenih belinama i deklarisanih na drugom mestu

```
<!ATTLIST film izvor ENTITY #REQUIRED>
<film izvor="supermen 3"/>
```

```
<!ATTLIST projekcija slajdovi ENTITIES #REQUIRED>
<projekcija slajdovi="slajd1 slajd2 slajd3"/>
```

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE opis [
  <!ELEMENT opis (prikaz)*>
  <!ELEMENT prikaz EMPTY>
  <!ATTLIST prikaz slika ENTITY #REQUIRED>
  <!ENTITY ppp SYSTEM
    "http://www.viser.edu.rs/prikaz/slika/ppp.gif">
]>
< opis >
  <prikaz slika="ppp"/>
< opis >
```

Dodatni opis vrednosti atributa

Deklaracija	Opis
#REQUIRED	Atribut mora postojati.
#IMPLIED	Atribut nije obavezan i nema podrazumevane vrednosti.
value	Ako vrednost atributa ne postoji, ovaj parametar predstavlja podrazumevanu vrednost.
#FIXED value	Ako atribut postoji, njegova vrednost mora da odgovara parametru attribute-value.

▶ 37

18:11

Primeri:

DTD:

```
<!ELEMENT kosulja (#PCDATA)>
<!ATTLIST kosulja tip CDATA #REQUIRED>
<!ATTLIST kosulja kragna CDATA #IMPLIED>
<!ATTLIST kosulja velicina (small|medium|large) "large">
<!ATTLIST kosulja proizvođjac CDATA #FIXED "Levi">
```

Validan XML:

```
<kosulja tip="standard">cotton</kosulja>
<kosulja tip="standard" velicina="large">wool</kosulja>
<kosulja tip="standard" proizvođjac="Levi">denim</kosulja>
<kosulja tip="standard kratka" kragna="button-down"></kosulja>
```

Nevalidan XML:

```
<kosulja></shirt>
<kosulja tip="standard" velicina="medium large">cardigan</kosulja>
<kosulja tip="standard" proizvođjac="Gap">designer</kosulja>
```

▶ 38

18:11

Primeri:

DTD:

```
<!ELEMENT book (#PCDATA)>
<!ATTLIST book
    id          ID          #IMPLIED
    isbn         CDATA       #REQUIRED
    booktype    (Harcover|Paperback) "Paperback"
    storeloc     CDATA       "5th Avenue"
    year         CDATA       #FIXED      "2000"
    comment     CDATA       #IMPLIED>
```

Validan XML:

```
<book isbn="1-35267-742-4" storeloc="Times Square">
  XML Pocket Reference
</book>
```

Specijalni karakteri - ENTITY

- ▶ To su promenljive korišćene za definisanje skraćenica ka standardnom tekstu ili specijalnim karakterima.

- ▶ **Interna deklaracija**

- ▶ `<!ENTITY entity-name "entity-value">`

- ▶ **Primeri:**

```
<!ENTITY writer "Donald Duck.">
<!ENTITY copyright "Copyright W3Schools.">
```

```
<author>&writer;&copyright;</author>
```

&lt;	<
&gt;	>
&amp;	&
&apos;	'
&quot;	"

- ▶ **Važno: Entity ima 3 dela: ampersend (&), entity ime, i tačka-zarez (;).**

- ▶ **Eksterna deklaracija**

- ▶ `<!ENTITY entity-name SYSTEM "URI/URL">`

- ▶ **DTD Example:**

```
<!ENTITY writer SYSTEM "http://www.w3schools.com/entities.dtd">
<!ENTITY copyright SYSTEM "http://www.w3schools.com/entities.dtd">
```

XML example:

```
<author>&writer;&copyright;</author>
```

Spoljašnji neraščlanjeni entiteti

- ▶ Postoje mnogi formati koji nisu XML. Istovremeno oni su potrebni kao podaci koji mogu ulaziti u XML datoteke. Takvi su jpeg, mpeg, gif, quickTime,
- ▶ Mehanizam koji se predlaže je zasnovan na spoljnim neraščlanjenim entitetima.
- ▶ `<!ENTITY test SYSTEM "http://www.turing.org.uk/busgroup.jpg" NDATA jpeg>`

Nedostaci DTD-a

Glavni nedostaci DTD-a su:

- ❖ odsustvo tipizacije podataka (#PCDATA može biti bilo koji string)
- ❖ sintaksa DTD nije usklađena sa sintaksom XML-a
- ❖ postoje ograničenja koja se ne mogu lako izraziti DTD-om (na pr. element x se može pojaviti od 4 do 17 puta)

Mnoga ograničenja DTD-a uspešno prevazilazi XML Schema (XML šema).

Primer:

```
▶ XmlReaderSettings xs = new XmlReaderSettings();  
▶ xs.ProhibitDtd = false;  
▶ xs.ValidationEventHandler += new ValidationEventHandler(xs_ValidationEventHandler);  
▶ xs.ValidationType = ValidationType.DTD;  
  
▶ XmlReader xr = XmlReader.Create(@"C:\Users\innet\Desktop\proba.xml", xs);  
  
▶ while (xr.Read());  
  
▶ Console.WriteLine("fajl je ok");
```