XML Validation

Document Type Definition (DTD)

XML – Validation

- Validation is a process by which an XML document is validated.
- An XML document with correct syntax is known as valid XML document.

Let's see few important rules to check for syntax errors.

- All XML documents must have a root element.
- XML is a case sensitive language so you should be careful with the case while opening and closing tags.
- All XML tags must have a closing tag.
- XML attribute name should not be quoted while its value must be quoted.
- There are two ways to check whether the XML document is valid.

1 XML DTD (Document Type Definition)

2 XML Schema

Document Type Definition (DTD)

- DTD stands for Document Type Definition. It is used to define document structure with a list of legal elements and attributes.
- Document Type Definition (DTD) is a certain piece of code, which can defines structure of XML document.
- Each DTD contains a list of elements, which specifies the rules for structuring a given XML document.
- Each DTD specifies the relationship between root element, child elements and sub child elements.
- An XML document is considered "**well formed**" and "**valid**" if it is successfully validated against DTD.

Document Type Definition (DTD) Contd..

- DTD's are optional in XML, but recommended for clarity purpose.

- DTD's can be declared internally (With in XML doc) and as an external file (with in a separate file with "**.dtd**" extension).

Basic Building Blocks:

There are 4 building blocks to build a XML document with DTD.

those are

1.Tags

2.Elements

3.Attributes

4.Entities

Document Type Definition (DTD) Contd..

1. Tags:

– The XML allows the user to create own tags.Useally the tag <tagname> is an Opening tag, </tagname> refers to its equivalent closing tag.

Example:

<book>Let us C</book>

<author> Yashwant Kanetkar </ author >

– In XML the tags are case sensitive . And closing tag is mandatory.

– All the elements in XML should be properly nested and they should not overlap.

Document Type Definition (DTD) Contd..

2. Elements in DTD:

– The DTD document is composed with various elements. These elements are used to represent the tags in XML document.

Declaration:

<**!ELEMENT** name-of-element(context)>

– To declare an empty element

Syntax:

<**!ELEMENT** name-of-element(**EMPTY**)>

– To declare an element, this carries data

Syntax:

<**!ELEMENT** name-of-element(**#PCDATA**)>

(or)

<**!ELEMENT** name-of-element(**#CDATA**)>

Document Type Definition (DTD) Contd..

– To declare an element, this contains child elements

Syntax:

<**!ELEMENT** name-of-element(child-names)>

Examples:

<!ELEMENT student(id,name,age)>

<!ELEMENT name(#PCDATA)>

<!ELEMENT age(EMPTY)>

3.Attributes:

- These are used to provide the additional information along with elements.
- In DTD, the attributes are declared by using "**ATTLIST**" **Keyword.**

Syntax:

<! ATTLIST name-of-element name-of-attribute

attribute-type [default-value]>

Document Type Definition (DTD) Contd..

- In the above syntax, the field ‘attribute-type’ can specifies the following pre-defined values.
- **ID:**It is a value which remains unique.
- **CDATA:** The value supplied here is nothing but character data.
- **ENTITY:** The value supplied in this case is nothing but an entity.

- In the same way, the filed ‘default-value’ can specifies the following pre-defined values.
  - ✓ **#REQUIRED**: It means the value for the attribute is required.
  - ✓ **#IMPLIED**: It means the attribute is not required.
  - ✓ **#FIXED**: Here a fixed value is supplied.
  - ✓ **Default-value**: It is a default value of given attribute.

Example:

<! ATTLIST student address CDATA #REQUIRED>

Document Type Definition (DTD) Contd..

4. Entities:

- Some markup elements can contain complex data; these types of elements are called as Entities.
- These are used to create small piece of data which you want you use repeatedly throughout your schema.

Syntax:

<! ENTITY name-of-entity “value”>

Example:

<! ENTITY Book “Web Technologies”>

Used as

<author> the &Book author is Uttam K Roy</author>

Document Type Definition (DTD) Contd..

- DTD's can be declared internally (With in XML doc) and as an external file (with in a separate file with **".dtd**" extension)

Example with external DTD:

- Create a DTD for a remainder; it has following **remainder** as root element and child elements-**heading, to, from, message**.

"remainder.dtd"

```
<!ELEMENT remainder(heading,to,from,message)>
<!ELEMENT heading(#PCDATA)>
<!ELEMENT to(#PCDATA)>
<!ELEMENT from(#PCDATA)>
<!ELEMENT message(#PCDATA)>
```

Document Type Definition (DTD) Contd..

“remainder.xml”

```
<?xml version="1.0"?>
<!DOCTYPE note SYSTEM "remainder.dtd">
<remainder>
<heading>Final Remainder</heading>
<to>Mr.Madhu</to>
<from> Ravi Gupta</from>
<message> Date of joining is 5th Feb</message>
</remainder>
```

Output:



Document Type Definition (DTD) Contd..

Example with internal DTD:

“remainderdemo.xml”

```
<?xml version="1.0"?>
<!DOCTYPE note [
<!ELEMENT remainder (heading,to,from,message)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT message (#PCDATA)>
]>
<remainder>
<heading>Final Remainder</heading>
<to>Mr.Madhu</to>
<from> Ravi Gupta</from>
<message> Date of joining is 5th Feb</message>
</remainder>
```

Document Type Definition (DTD) Contd..

Example:

- Create a DTD for a **catalog** of four stroke engine **motorbikes** where each motor bike has the following child elements: **make, model, year, color, engine, chassis number and accessories**. The engine element has child elements those are – **engine number, number of cylinders, type of fuel**. The accessories elements has the attributes like **disk break, auto start and radio**, each of which required and has the possible values **'YES'** and '**NO'** . Entities must be declared for the names of the popular motorbike makes.

Document Type Definition (DTD) Contd..

“bikecatalog.dtd”

```
<!ELEMENT catalog (motorbike)*>
<!ELEMENT motorbike (make, model,year,color,engine, chasis_num, accessories)>
<!ELEMENT make (#PCDATA)>
<!ELEMENT model (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT color (#PCDATA)>
<!ELEMENT engine (engine_num,cylinder_num,fuel_type)>
<!ELEMENT engine_num (#PCDATA)>
<!ELEMENT cylinder_num (#PCDATA)>
<!ELEMENT fuel_type (#PCDATA)>
<!ELEMENT chasis_num (#PCDATA)>
<!ELEMENT accessories (#PCDATA)>
<!ATTLIST accessories diskbrake (Yes|No) #REQUIRED autostart (Yes|No)
                                                    #REQUIRED radio (Yes|No) #REQUIRED>
```

Document Type Definition (DTD) Contd..

“bikecatalog.xml”

```
<?xml version="1.0"?>
<!DOCTYPE catalog SYSTEM "bikecatalog.dtd">
<catalog>
<motorbike>
<make>Hero</make>
<model>Glammur 120cc</model>
<year>2014</year>
<color>Red-Black</color>
<engine>
<engine_num>564789</engine_num>
<cylider_num>32</cylider_num>
<fuel_type>PreMimum</fuel_type>
</engine>
<chasis_num>546</chasis_num>
<accessories discbrake="Yes" autostart="Yes" radio="No"/>
</motorbike>
</catalog>
```

