Session-2

XML Validation

A well formed XML document can be validated against **DTD** or **Schema**.

A well-formed XML document is an XML document with correct syntax. It is very necessary to know about valid XML document before knowing XML validation.

XML Document can be validated in

Well Formed XML Documents

An XML document with correct syntax is called "Well Formed".

The syntax rules were described in the previous chapters:

- XML documents must have a root element
- XML elements must have a closing tag
- XML tags are case sensitive
- XML elements must be properly nested
- XML attribute values must be quoted

- An XML document with correct syntax is called "Well Formed".

- An XML document validated against a DTD is both "Well Formed" and "Valid".

DTD:

DTD stands for Document Type Definition.

A DTD defines the structure and the legal elements and attributes of an XML document.

The DTD can be declared in two ways as follows:

1.Internal (or) Embedded DTD

2.External DTD

DTD declarations either internal XML document or make external DTD file, after linked to a XML document.

Internal DTD : If the DTD is declared inside the XML file, it must be wrapped inside the <!DOCTYPE> definition:

Example:

note.xml

```
<?xml version="1.0"?>
<!DOCTYPE message [
<!ELEMENT message (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
<message>
<to>krishna</to>
<from>venkatesh</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend</body>
</message>
```

External DTD: If the DTD is declared in an external file, the <!DOCTYPE> definition must contain a reference to the DTD file:

And here is the file "note.dtd", which contains the DTD:

note.dtd

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

note.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE note SYSTEM "Note.dtd">
<message>
<to>Krishna</to>
<from>Venkatesh</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</ message>
```

XML Schema:

An XML Schema describes the structure of an XML document.

The XML Schema language is also referred to as XML Schema Definition (XSD).

Example:

note.xml

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="note">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="to" type="xs:string"/>
      <xs:element name="from" type="xs:string"/>
      <xs:element name="heading" type="xs:string"/>
      <xs:element name="body" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```

Data Types:

XML Schema has a lot of built-in data types. The most common types are:

- xs:string

- xs:decimal
- xs:integer
- xs:boolean
- xs:date
- xs:time

Example

Here are some XML elements:

```
<lastname>Refsnes</lastname>
<age>36</age>
<dateborn>1970-03-27</dateborn>
```

And here are the corresponding simple element definitions:

```
<xs:element name="lastname" type="xs:string"/>
<xs:element name="age" type="xs:integer"/>
<xs:element name="dateborn" type="xs:date"/>
```

XSD Element Types:

In XSD elements can be of type

1. **simpleType** : An element of type **simpleType** contains only text.
2. **complexType :** An element of type **complexType** is parent to all the elements and attributes contained within it.
3. **anyType:** An `any` element in an XSD specifies that any well-formed XML is allowed in its place in XML instance.

simpleType:

The simpleType element defines a simple type and specifies the constraints and information about the values of attributes or text-only elements.

Element Information

- **Parent elements:** attribute, element, list, restriction, schema, union

Syntax

```
<simpleType
```

```
id=ID
name=NCName
any attributes
>

(annotation?,(restriction|list|union))

</simpleType>
```

The complexType element defines a complex type. A complex type element is an XML element that contains other elements and/or attributes.

Element Information

- **Parent elements:** element, redefine, schema

Syntax

```
<complexType
id=ID
name=NCName
abstract=true|false
mixed=true|false
block=(#all|list of (extension|restriction))
final=(#all|list of (extension|restriction))
any attributes
>

(annotation?,(simpleContent|complexContent|((group|all|
choice|sequence)?,((attribute|attributeGroup)*,anyAttribute?))))

</complexType>
```

The XSD path location can be declared in xml file as follows:

1) xsi:schemaLocation

2) xsi:noNamespaceSchemaLocation

1.xsi:schemaLocation

The **xsi:schemaLocation** attribute locates schemas for elements and attributes that are in a specified namespace.

Its value is a namespace URI followed by a relative or absolute URL where the schema for that namespace can be found.

xsi:noNamespaceSchemaLocation

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

<message xmlns="https://www.epskills.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://www.epskills.com note.xsd">

<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</message>
```

2.xsi:noNamespaceSchemaLocation:

The **xsi:noNamespaceSchemaLocation** attribute locates the schema for elements that are not in any namespace. (Attributes that are not in any namespace are assumed to be declared in the same schema as their parent element.) Its value is a relative or absolute URL where the schema document can be found. It is most commonly attached to the root element but can appear further down the tree.

Example:

Let's have a look at this XML document called "shiporder.xml":

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

    orderid="889923"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xsi:noNamespaceSchemaLocation="shiporder.xsd">

  <orderperson>John Smith</orderperson>
  <shipto>
    <name>Ola Nordmann</name>
    <address>Langgt 23</address>
    <city>4000 Stavanger</city>
    <country>Norway</country>
  </shipto>
  <item>
    <title>Empire Burlesque</title>
    <note>Special Edition</note>
    <quantity>1</quantity>
    <price>10.90</price>
  </item>
  <item>
    <title>Hide your heart</title>
    <quantity>1</quantity>
    <price>9.90</price>
  </item>
</shiporder>
```

The XML document above consists of a

root element, →"shiporder",

that contains a required attribute called "orderid".

The "shiporder"→ element contains three different child elements:

"orderperson", "shipto" and "item". The "item" element appears twice, and it contains a "title", an optional "note" element, a "quantity", and a "price" element.

The line above:

xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance→ tells the XML parser that this document should be validated against a schema.

The line: **xsi:noNamespaceSchemaLocation="shiporder.xsd"** →specifies WHERE the schema resides (here it is in the same folder as "shiporder.xml").