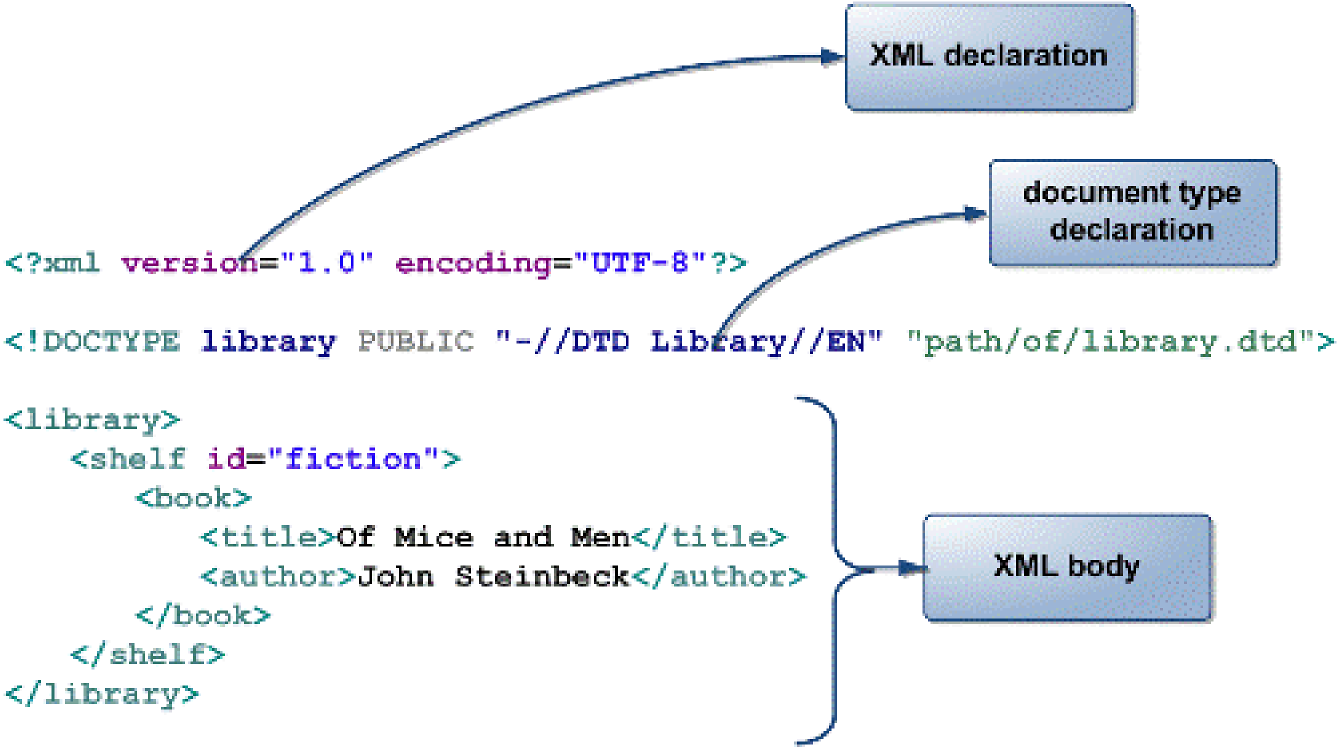
12th February

XSD Interview Questions

What is XSD? What are the benefits of having XSDs? What are the advantages of XSD?

An XML Schema describes the structure of an XML instance document by defining what each element must or may contain.XML Schema is expressed in the form of a separate XML file. XML Schema provides much more control on element and attribute datatypes. Some datatypes are predefined and new ones can be created.



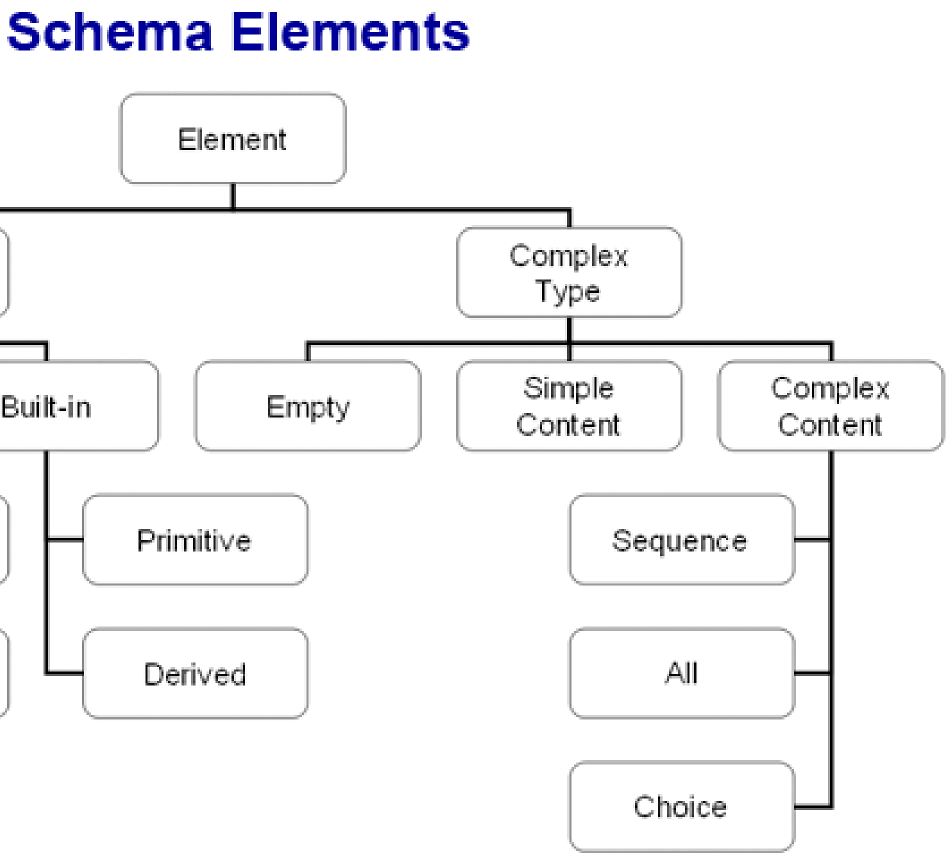
```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE library PUBLIC "-//DTD Library//EN" "path/of/library.dtd">

<library>
   <shelf id="fiction">
      <book>
         <title>Of Mice and Men</title>
         <author>John Steinbeck</author>
      </book>
   </shelf>
</library>
```

[http://www.developersbook.com/xml/images/XML-Structure.gif]

```
<xsd:schema  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
     <xsd:element  name="test">
       <xsd:complexType>
```



[http://www.developersbook.com/xml/images/schema.gif]

What is the difference between complexType and simpleType attribute in XSD?
<xsd:complexType name="FirstElement" />
<xsd:simpleType name="SecondElement"/>

A simple element is an XML element that can contain only text.
- A simple element cannot have attributes
- A simple element cannot contain other elements
- A simple element cannot be empty
- However, the text can be of many different types, and may have various restrictions applied to it

In the SimpleType, you can have the list as follows :

```
<xs:element name="stringvalues" type="valuelist">
<xs:simpleType name="valuelist">
  <xs:list itemType="xs:string"/>
</xs:simpleType>
```

Output

```
 <stringvalues>I love XML Schema</stringvalues>
```

A complex element is an XML element that contains other elements and/or attributes.
There are four kinds of complex elements:

- empty elements

```
<product prodid="1345" />
```

- elements that contain only other elements

```
<xs:element name="product">
  <xs:complexType>
    <xs:complexContent>
      <xs:restriction base="xs:integer">
        <xs:attribute name="prodid" type="xs:positiveInteger"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

- elements that contain only text

```
<xs:element name="shoesize">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:integer">
        <xs:attribute name="country" type="xs:string" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

- elements that contain both other elements and text

```
<letter>
  Dear Mr.<name>John Smith</name>.
  Your order <orderid>1032</orderid>
  will be shipped on <shipdate>2001-07-13</shipdate>.
</letter>
```

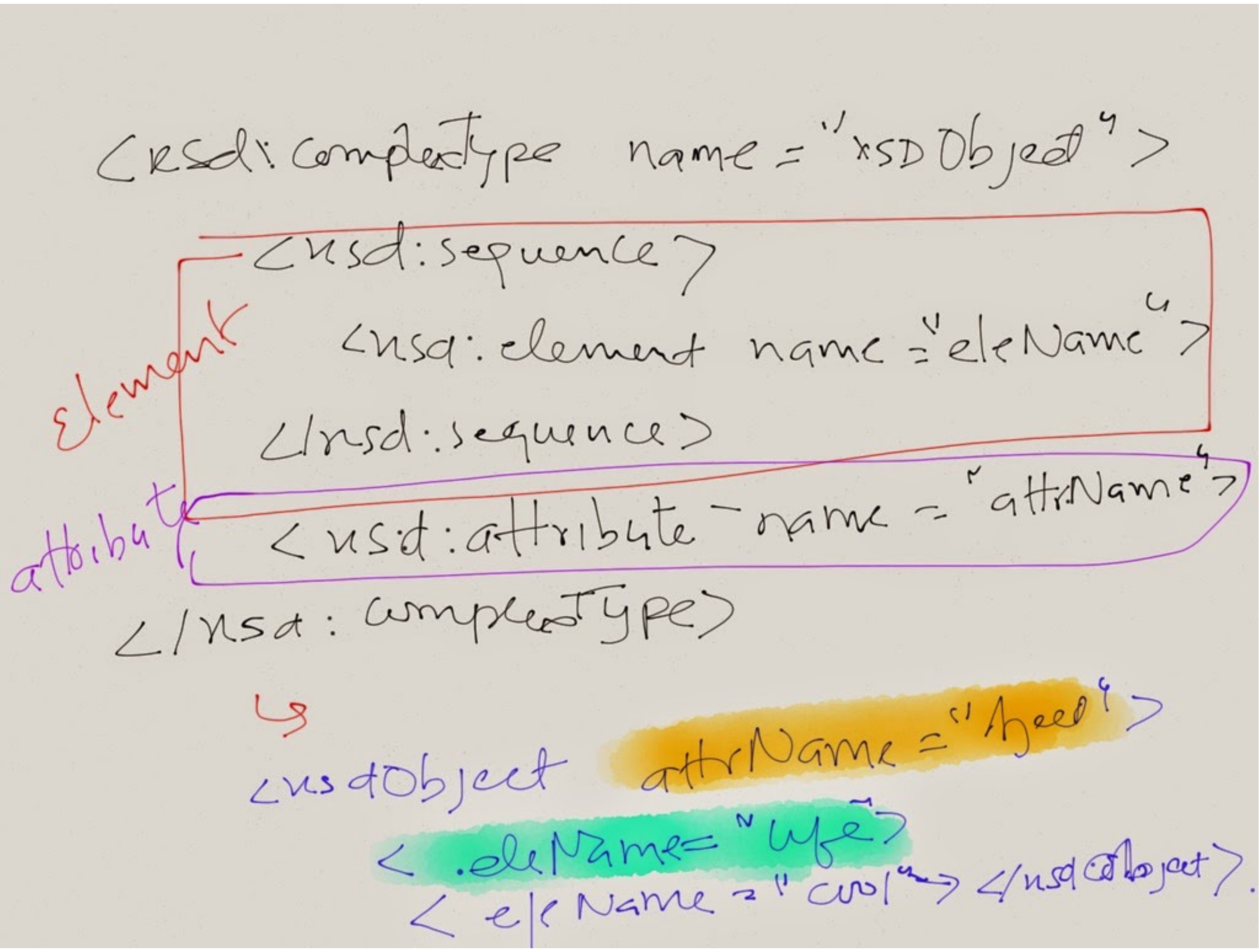
The following schema declares the "letter" element:

```
<xs:element name="letter">
  <xs:complexType mixed="true">
    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="orderid" type="xs:positiveInteger"/>
      <xs:element name="shipdate" type="xs:date"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

What is the difference between attribute and element in XSD?

<xsd:attribute name="FirstAttribute" />
<xsd:element name="FirstElement" />

Element is the root element which contains the attribute. Attribute are the element which defines the property and generally holds the value. Element can be of complexType or SimpleType. Each ComplexType or SimpleType can have different attribute based on the requirement.



```
<xsd:complextype name="xsD Object">
  <xsd:sequence>
     <xsd:element name="eleName">
  </xsd:sequence>
  <xsd:attribute name="attrName">
</xsd:complexType>
↳
<xsdObject attrName="Ajeet">
  <.eleName="wfe">
   <eleName="cool"> </xsdObject>.
```

element
attribute

[http://1.bp.blogspot.com/-AqxBo-xbcsA/U3ufpi-ay-I/AAAAAAAAYUo/PgaD4JMcVoE/s1600/xsd.jpeg]

What are the different data types in XSD? How do you define it?
There are different datatypes: int, string, date etc. You can have custom datatype as in the Java classes. You have to refer them using the <xsd:attribute name="shoetype" type="xsd:string"/>

How do you enforce a sequence order for filling the elements in XSDs?
By Sequence order attribute.

How do you reference elements of different model with each other?
By defining one attribute in one element and defining it outside. We can refer the same in that element.

```
<xs:element name="shoesize" type="shoetype"/>


<xs:complexType name="shoetype">
  <xs:simpleContent>
    <xs:extension base="xs:integer">
      <xs:attribute name="country" type="xs:string" />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

What are namespaces? Why are they important?

A simple element is an XML element that can contain only text.

- Namespaces are a simple and straightforward way to distinguish names used in XML documents, no matter where they come from.
- XML namespaces are used for providing uniquely named elements and attributes in an XML instance
- They allow developers to qualify uniquely the element names and relationships and make these names recognizable, to avoid name collisions on elements that have the same name but are defined in different vocabularies.
- They allow tags from multiple namespaces to be mixed, which is essential if data is coming from multiple sources.

Example: a bookstore may define the <TITLE> tag to mean the title of a book, contained only within the <BOOK> element. A directory of people, however, might define <TITLE> to indicate a person's position, for instance: <TITLE>President</TITLE>. Namespaces help define this distinction clearly.

Note: a) Every namespace has a unique name which is a string. To maintain the uniqueness among namespaces a IRL is most preferred approach, since URLs are unique.

b) Except for no-namespace Schemas, every XML Schema uses at least two namespaces:

1.the target namespace.

2. The XMLSchema namespace (http://w3.org/2001/XMLSchema)

What are the ways to use namespaces?

There are two ways to use namespaces:

- Declare a default namespace
- Associate a prefix with a namespace, then use the prefix in the XML to refer to the namespace

What are differences between DTDs and Schema?

Schema	DTD
Schema document is an XML document i.e., the structure of an XML document is specified by another XML document.	DTDs follow SGML syntax.
Schema supports variety of dataTypes similar to programming language.	In DTD everything is treated as text.
In Schema, It is possible to inherit and create relationship among elements.	This is not possible in DTD without invalidating existing documents.
In Schema, It is possible to group elements and attributes so that they can be treated as single logical unit.	Grouping of elements and attributes is not possible in DTD.
In Schemas, it is possible to specify an upper limit for the number of occurrences of an element	It is not possible to specify an upper limit of an element in DTDs

Posted 12th February by Big Data Enthusiast

0 Add a comment

Enter your comment...

Comment as: Google Accoui ▼

Publish Preview