XML Schema

- **External XSD:** Locate the xsd at the root element

  <note

  xmlns="https://www.w3schools.com"

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

  xsi:schemaLocation="https://www.w3schools.com/xml note.xsd">

- **Restrictions:** Restrictions are used to define acceptable values for XML elements or attributes. Restrictions on XML elements are called facets.

  Sample XML schema for Restrictions for age element:

```
<xs:element name="age">
    <xs:simpleType>
        <xs:restriction base="xs:integer">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="120"/>
    </xs:restriction>
    </xs:simpleType>
    </xs:element>
```

  - **Example 2: For car element:** Possible values = Audi, Golf, BMW

```
<xs:element name="car">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="Audi"/>
            <xs:enumeration value="Golf"/>
            <xs:enumeration value="BMW"/>
         </xs:restriction>
    </xs:simpleType>
  </xs:element>
```

  - **Example 3 for pattern:**

```
<xs:element name="letter">
    <xs:simpleType>
        <xs:restriction base="xs:string">
                <xs:pattern value="[a-z]"/>
        </xs:restriction>
    </xs:simpleType>
```

</xs:element>

- **Restrictions for Data types:** syntax <xs:name>

1. enumeration
2. fractionDigits
3. maxExclusive
4. maxInclusive
5. maxLength
6. minExclusive
7. minInclusive
8. minLength
9. pattern
10. totalDigits
11. whiteSpace

❖ **Attributes:** All attributes are declared as simple types.

❖ **Syntax for defining attribute:**

<xs:attribute name="xxx" type="yyy" use=" " default|fixed=value>

❖ **Different ways to define values for an attributes:**

1. *fixed = value* → A fixed value is also *automatically assigned to the attribute, and you cannot specify another value*.
2. *default = value* → A default value is automatically assigned to the attribute when no other value is specified.
3. use = "required"

XSL (eXtensible Stylesheet Language)

(XSLT: – XSL Transformations)

❖ **Goal:** transform XML document into other formats

❖ ***Need for XSL***: In HTML browser knows the functionalities every tag. So it will be rendered clearly but in *XML won't support predefined tags and the purpose of the tags is not defined*. So, need a some method to render XML document to display the XML document in understandable format.

❖ **Prerequisite:**

1. Identify the suitable node in DOM-XML source tree. (selective navigation)
2. Need XSLT processor for transformation

❖ **Input:** Validated XML Document

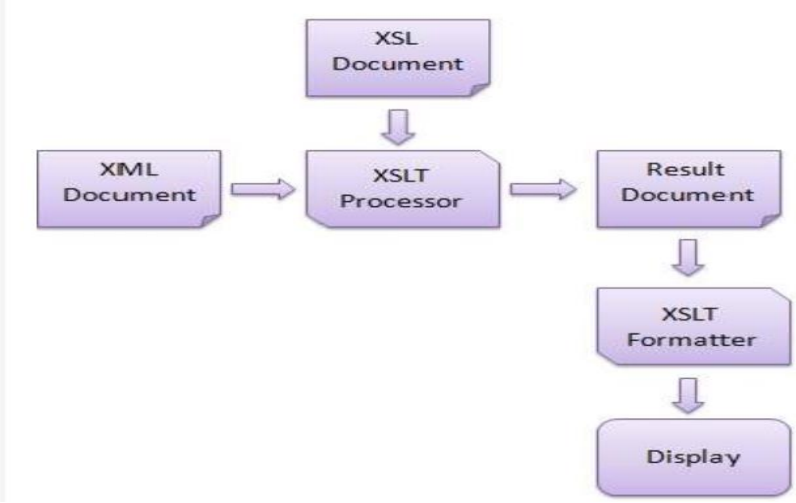
❖ **Output:** New Formatted XML document

❖ **Four Parts of XSL:**

1. XSLT → language for XSL Transformation
2. XPath → language for navigating in XML documents
3. XSL – FO → language for formatting XML document (Generating PDF)
4. XQuery → a language for querying XML documents (extracting elements and attributes in XML documents)

❖ All the browsers should support both XSLT and XPath.

❖ **Working Principle of XSLT:**



❖ **XSL Style Sheet Elements::**

1. <xsl:stylesheet>
2. <xs:template match=”/”>
3. <xsl:value-of select=”tag_name”>
4. <xsl:for-each select=”path_from_root_element_tag_name”>
5. <xsl:sort select=”tag_name”> [must be defined inside of for-each]
6. <xsl:if test=”expression”>
7. <xsl:choose>
   <xsl:when test=”expression”> … </xsl:when>
   <xsl:otherwise> ….. </xsl:otherwise>
   </xsl:choose>
8.

❖ **Filtering the Output:**

1. Equal (=)
2. Not Equal (!=)
3. Less than (<)
4. Greater than (>)

❖ **Example Program:**

Testxsl.xsl

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"   xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
 <html>
 <body>
```

```
  <table  border="1">
    <tr>
          <td> Name</td>
          <td> Players</td>
          <td> Like</td>
    </tr>

    <xsl:for-each select="sports/game">
    <xsl:sort select="like"></xsl:sort>
    <tr>
          <td> <xsl:value-of  select="name"></xsl:value-of> </td>
          <td>  <xsl:value-of  select="players"></xsl:value-of></td>
          <td>  <xsl:value-of  select="like"></xsl:value-of></td>
    </tr>
    </xsl:for-each>
  </table>

 </body>
 </html>
</xsl:template>
</xsl:stylesheet>
```

Test.xml:

```
<?xml-stylesheet type="text/xsl" href="testxsl.xsl"?>
<sports>
    <game>
          <name> cricket</name>
           <players> 11 </players>
          <like> India </like>
    </game>
    <game>
          <name> Kabadi </name>
           <players> 7 </players>
          <like> India </like>
```

```
        </game>
        <game>
                <name> Volleyball</name>
                <players> 6 </players>
        <like> USA </like>
        </game>
    </sports>
```

- **Running XSL**: Open the browser and load *xml file.
-
- S

enumeration
fractionDigits
maxExclusive
maxInclusive
maxLength
minExclusive
minInclusive
minLength
pattern
totalDigits
whiteSpace