

XML EXERCISE



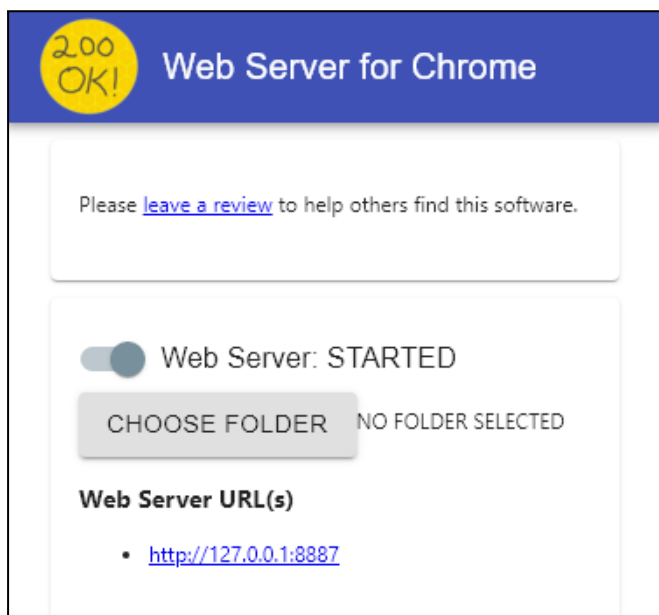
Web Server for Chrome

<https://chrome.google.com/webstore/detail/web-server-for-chrome/ofhbbkphhbklhfoeikjpcbhelmocgigb?hl=en>

Add To Chrome

When you run Chrome web browser for the first time, browse this url :

chrome://apps/

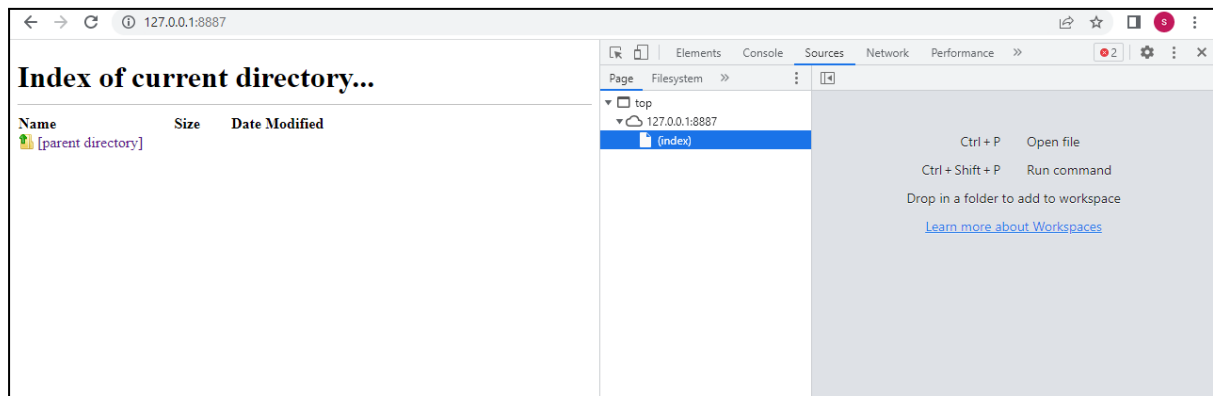


<http://127.0.0.1:8887/>



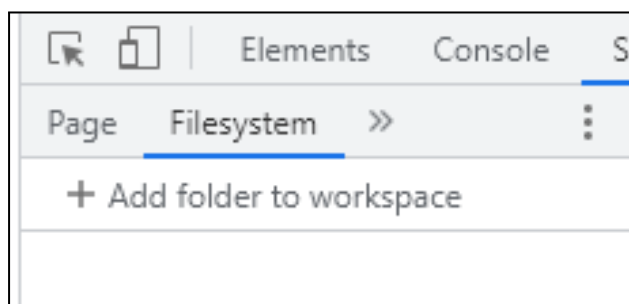
Press [F12] to open Developer Panel

Click Sources tab.

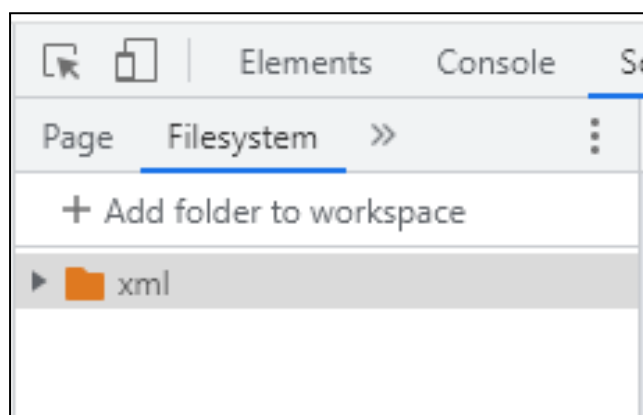


Click Filesystem.

Click the button “+ Add folder to workspace”.



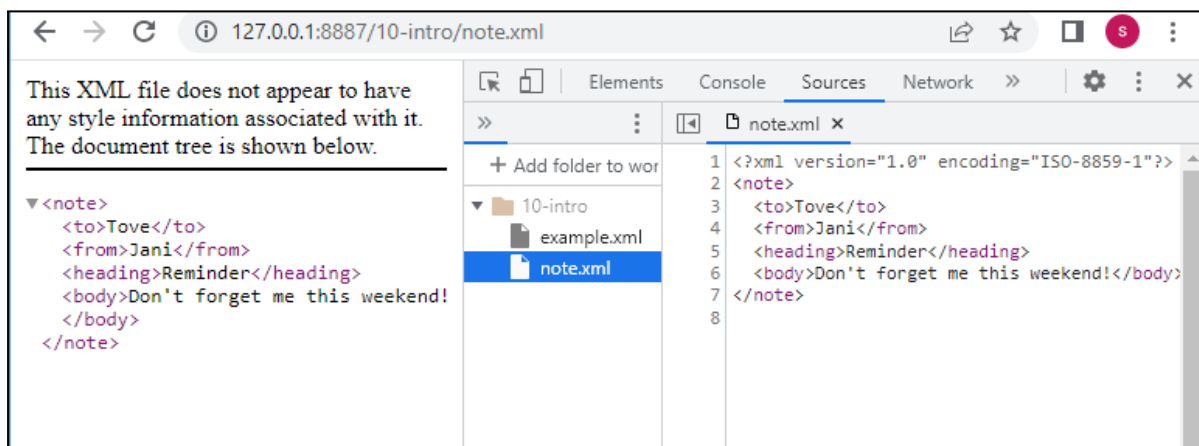
Select the folder to be used for this exercise, e.g. “xml”.



File name: note.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Press [CTRL+S] to save the file.



File name: bookstore.xml

```
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

An XML document keeps data in two ways; (1) as element e.g. Title is Harry Porter , (2) as attribute, e.g. category is Children.

File name: email.xml
<pre><?xml version="1.0" encoding="UTF-8"?> <emails> <email> <to>Vimal</to> <from>Sonoo</from> <heading>Hello</heading> <body>Hello brother, how are you!</body> </email> <email> <to>Peter</to> <from>Jack</from> <heading>Birth day wish</heading> <body>Happy birth day Tom!</body> </email> <email> <to>James</to> <from>Jaclin</from> <heading>Morning walk</heading> <body>Please start morning walk to stay fit!</body> </email> <email> <to>Kartik</to> <from>Kumar</from> <heading>Health Tips</heading> <body>Smoking is injurious to health!</body> </email> </emails></pre>

Line no.1 is called PROLOG line. It tells computers that this is an XML document.

XML DEFINITION USING DTD

File name: employee.xml
<pre><?xml version="1.0"?> <!DOCTYPE employee SYSTEM "employee.dtd"> <employee></pre>

```
<firstname>vimal</firstname>
<lastname>jaiswal</lastname>
<email>vimal@javatpoint.com</email>
</employee>
```

File name: employee.dtd

```
<!ELEMENT employee (firstname,lastname,email)>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT email (#PCDATA)>
```

DTD= Data Type Definition. Defines data structure in XML document.

XML DEFINITION USING XSD

File name: staff.xml

```
<?xml version="1.0"?>
<staff
xmlns="http://www.javatpoint.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.javatpoint.com staff.xsd">

  <firstname>vimal</firstname>
  <lastname>jaiswal</lastname>
  <email>vimal@javatpoint.com</email>
</staff>
```

File name: staff.xsd

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

  <xs:element name="staff">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="firstname" type="xs:string"/>

```

```
<xs:element name="lastname" type="xs:string"/>
<xs:element name="email" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>

</xs:schema>
```

EXTENSIBLE STYLE SHEET (XSLT)

Create new folder 11-XSLT

File name: employee.xml

```
<?xml version = "1.0"?>
<?xml-stylesheet type = "text/xsl" href = "employee.xsl"?>
<class>
  <employee id = "001">
    <firstname>Aryan</firstname>
    <lastname>Gupta</lastname>
    <nickname>Raju</nickname>
    <salary>30000</salary>
  </employee>
  <employee id = "024">
    <firstname>Sara</firstname>
    <lastname>Khan</lastname>
    <nickname>Zoya</nickname>
    <salary>25000</salary>
  </employee>
  <employee id = "056">
    <firstname>Peter</firstname>
    <lastname>Symon</lastname>
    <nickname>John</nickname>
    <salary>10000</salary>
  </employee>
</class>
```

File name: employee.xsl

```

<?xml version = "1.0" encoding = "UTF-8"?>
<!-- xsl stylesheet declaration with xsl namespace:
Namespace tells the xslt processor about which
element is to be processed and which is used for output purpose only
-->
<xsl:stylesheet version = "1.0"
xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">
<!-- xsl template declaration:
template tells the xslt processor about the section of xml
document which is to be formatted. It takes an XPath expression.
In our case, it is matching document root element and will
tell processor to process the entire document with this template.
-->
<xsl:template match = "/">
  <!-- HTML tags
  Used for formatting purpose. Processor will skip them and browser
  will simply render them.
  -->
  <html>
    <body>
      <h2>Employee</h2>
      <table border = "1">
        <tr bgcolor = "#9acd32">
          <th>ID</th>
          <th>First Name</th>
          <th>Last Name</th>
          <th>Nick Name</th>
          <th>Salary</th>
        </tr>
        <!-- for-each processing instruction
        Looks for each element matching the XPath expression
        -->
        <xsl:for-each select="class/employee">
          <tr>
            <td>
              <!-- value-of processing instruction
              process the value of the element matching the XPath expression
              -->
              <xsl:value-of select = "@id"/>
            </td>
            <td><xsl:value-of select = "firstname"/></td>
            <td><xsl:value-of select = "lastname"/></td>
            <td><xsl:value-of select = "nickname"/></td>
            <td><xsl:value-of select = "salary"/></td>
          </tr>
        </xsl:for-each>
      </table>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>

```

Employee

ID	First Name	Last Name	Nick Name	Salary
001	Aryan	Gupta	Raju	30000
024	Sara	Khan	Zoya	25000
056	Peter	Symon	John	10000

```

1 <?xml version = "1.0" encoding =
2 <!-- xsl stylesheet declaration v
3 Namespace tells the xlst process
4 element is to be processed and wh
5 -->
6 <xsl:stylesheet version = "1.0"
7 xmlns:xsl = "http://www.w3.org/19
8 <!-- xsl template declaration:
9 template tells the xlst processor
10 document which is to be formatt
11 In our case, it is matching docum
12 tell processor to process the ent
13 -->

```

`<xsl:template match = "/">` - start reading from root

`<xsl:for-each select="class/employee">` - read each data, repeat for each data (for loop)

`class/employee` - data starts from root class

`<xsl:value-of select = "@id"/>` - select data item from attribute `@id`

`<xsl:value-of select = "firstname"/>` - select data item from element `firstname`

SORT DATA

File name: employee1.xml

```

<?xml version = "1.0"?>
<?xml-stylesheet type = "text/xsl" href = "employee1.xsl"?>
<class>
  <employee id = "001">
    <firstname>Aryan</firstname>
    <lastname>Gupta</lastname>
    <nickname>Raju</nickname>
    <salary>30000</salary>
  </employee>
  <employee id = "024">
    <firstname>Sara</firstname>
    <lastname>Khan</lastname>
    <nickname>Zoya</nickname>
    <salary>25000</salary>
  </employee>
  <employee id = "056">
    <firstname>Peter</firstname>

```

```
<lastname>Symon</lastname>
<nickname>John</nickname>
<salary>10000</salary>
</employee>
</class>
```

File name: employee1.xsl

```
<?xml version = "1.0" encoding = "UTF-8"?>
<xsl:stylesheet version = "1.0"
xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">
  <xsl:template match = "/">
    <html>
      <body>
        <h2>Employee</h2>
        <table border = "1">
          <tr bgcolor = "pink">
            <th>ID</th>
            <th>First Name</th>
            <th>Last Name</th>
            <th>Nick Name</th>
            <th>Salary</th>
          </tr>
          <xsl:for-each select = "class/employee">
            <xsl:sort select = "lastname"/>
            <tr>
              <td><xsl:value-of select = "@id"/></td>
              <td><xsl:value-of select = "firstname"/></td>
              <td><xsl:value-of select = "lastname"/></td>
              <td><xsl:value-of select = "nickname"/></td>
              <td><xsl:value-of select = "salary"/></td>
            </tr>
          </xsl:for-each>
        </table>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

`<xsl:sort select = "lastname"/>` - sort data by lastname

Try changing lastname for Gupta (change to Zupta) and refresh the browser.

FILTER DATA

File name: employee2.xml

```
<?xml version = "1.0"?>
<?xml-stylesheet type = "text/xsl" href = "employee2.xsl"?>
<class>
  <employee id = "001">
    <firstname>Aryan</firstname>
    <lastname>Gupta</lastname>
    <nickname>Raju</nickname>
    <salary>30000</salary>
  </employee>
  <employee id = "024">
    <firstname>Sara</firstname>
    <lastname>Khan</lastname>
    <nickname>Zoya</nickname>
    <salary>25000</salary>
  </employee>
  <employee id = "056">
    <firstname>Peter</firstname>
    <lastname>Symon</lastname>
    <nickname>John</nickname>
    <salary>10000</salary>
  </employee>
</class>
```

File name: employee2.xsl

```
<?xml version = "1.0" encoding = "UTF-8"?>
<xsl:stylesheet version = "1.0"
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">
  <xsl:template match = "/">
    <html>
      <body>
        <h2>Employee </h2>
        <table border = "1">
          <tr bgcolor = "pink">
            <th>ID</th>
            <th>First Name</th>
            <th>Last Name</th>
            <th>Nick Name</th>
            <th>Salary</th>
          </tr>
```

```

<xsl:for-each select = "class/employee">
  <xsl:if test = "salary > 30000">
    <tr>>
      <td><xsl:value-of select = "@id"/></td>
      <td><xsl:value-of select = "firstname"/></td>
      <td><xsl:value-of select = "lastname"/></td>
      <td><xsl:value-of select = "nickname"/></td>>
      <td><xsl:value-of select = "salary"/></td>
    </tr>
  </xsl:if>
</xsl:for-each>

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

```

`<xsl:if test = "salary > 30000">` - select if `salary > 30000`

Now, the browser does not show any data because everyone receives 30000 or less.

Change 30000 to 25000, then we will see 2 records.

CHOOSE DATA

File name: employee3.xml

```

<?xml version = "1.0"?>
<?xml-stylesheet type = "text/xsl" href = "employee3.xml"?>
<class>
  <employee id = "001">
    <firstname>Aryan</firstname>
    <lastname>Gupta</lastname>
    <nickname>Raju</nickname>
    <salary>30000</salary>
  </employee>
  <employee id = "024">
    <firstname>Sara</firstname>
    <lastname>Khan</lastname>
    <nickname>Zoya</nickname>
  </employee>
</class>

```

```

    <salary>25000</salary>
  </employee>
  <employee id = "056">
    <firstname>Peter</firstname>
    <lastname>Symon</lastname>
    <nickname>John</nickname>
    <salary>10000</salary>
  </employee>
</class>

```

File name: employee3.xsl

```

<?xml version = "1.0" encoding = "UTF-8"?>
<xsl:stylesheet version = "1.0"
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">
  <xsl:template match = "/">
    <html>
      <body>
        <h2>Employee</h2>
        <table border = "1">
          <tr bgcolor = "pink">
            <th>ID</th>
            <th>First Name</th>
            <th>Last Name</th>
            <th>Nick Name</th>
            <th>Salary</th>
            <th>Grade</th>
          </tr>

          <xsl:for-each select = "class/employee">

            <tr>
              <td><xsl:value-of select = "@id"/></td>
              <td><xsl:value-of select = "firstname"/></td>
              <td><xsl:value-of select = "lastname"/></td>
              <td><xsl:value-of select = "nickname"/></td>
              <td><xsl:value-of select = "salary"/></td>

              <td>
                <xsl:choose>
                  <xsl:when test = "salary > 25000">
                    High
                  </xsl:when>

                  <xsl:when test = "salary > 20000">
                    Medium
                  </xsl:when>

                  <xsl:otherwise>

```

```

        Low
      </xsl:otherwise>
    </xsl:choose>
  </td>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

```

      <xsl:choose>
        <xsl:when test = "salary > 25000">
          High
        </xsl:when>
      ...
      <xsl:otherwise>
        Low
      </xsl:otherwise>
    </xsl:choose>

```

- Display a label of either High, Medium or Low based on conditions.

SEARCH DATA (BY KEY)

File name: employee4.xml

```

<?xml version = "1.0"?>
<?xml-stylesheet type = "text/xsl" href = "employee4.xsl"?>
<class>
  <employee id = "001">
    <firstname>Aryan</firstname>
    <lastname>Gupta</lastname>
    <nickname>Raju</nickname>
    <salary>60000</salary>
  </employee>
  <employee id = "024">
    <firstname>Sonam</firstname>
    <lastname>Gupta</lastname>
    <nickname>Vinni</nickname>
    <salary>45000</salary>
  </employee>
  <employee id = "056">

```

```
<firstname>Peter</firstname>
<lastname>Symon</lastname>
<nickname>John</nickname>
<salary>20000</salary>
</employee>
</class>
```

File name: employee4.xsl

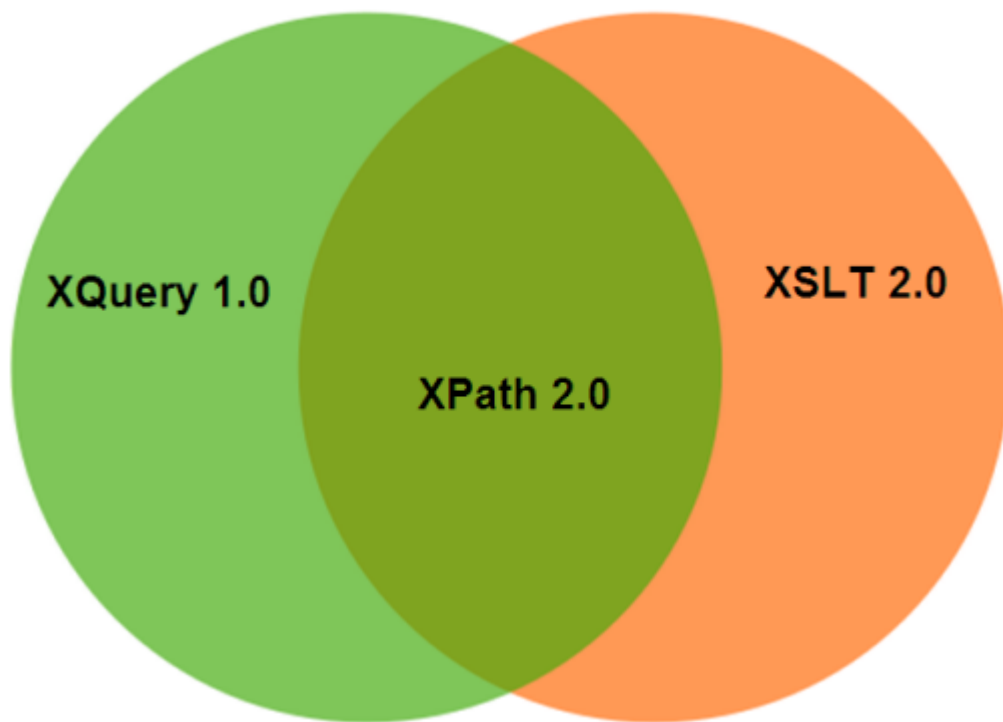
```
<xsl:stylesheet version = "1.0"
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">
  <xsl:key name = "firstname-search" match = "employee" use = "firstname"/>
  <xsl:template match = "/">
    <html>
      <body>
        <h2> Employee</h2>
        <table border = "1">
          <tr bgcolor = "pink">
            <th> ID</th>
            <th>> First Name</th>
            <th> Last Name</th>
            <th> Nick Name</th>
            <th> Salary</th>
          </tr>

          <xsl:for-each select = "key('firstname-search', 'Sonam')">

            <tr>
              <td> <xsl:value-of select = "@id"/> </td>
              <td> <xsl:value-of select = "firstname"/> </td>
              <td> <xsl:value-of select = "lastname"/> </td>
              <td> <xsl:value-of select = "nickname"/> </td>
              <td> <xsl:value-of select = "salary"/> </td>
            </tr>

          </xsl:for-each>
        </table>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

<xsl:for-each select = "key('firstname-search', 'Sonam')"> - search by key firstname = Sonam



XPATH

Create new folder 12-xpath

File name: employee1.xml

```
<?xml version = "1.0"?>
<?xml-stylesheet type = "text/xsl" href = "employee1.xsl"?>
<class>
  <employee id = "001">
    <firstname>Aryan</firstname>
    <lastname>Gupta</lastname>
    <nickname>Raju</nickname>
    <salary>30000</salary>
  </employee>
  <employee id = "024">
    <firstname>Sara</firstname>
    <lastname>Khan</lastname>
    <nickname>Zoya</nickname>
    <salary>25000</salary>
  </employee>
  <employee id = "056">
```

```
<firstname>Peter</firstname>
<lastname>Symon</lastname>
<nickname>John</nickname>
<salary>10000</salary>
</employee>
</class>
```

File name: employee1.xsl

```
<?xml version = "1.0" encoding = "UTF-8"?>
<xsl:stylesheet version = "1.0"
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">
  <xsl:template match = "/">
    <html>
      <body>
        <h2> Employees</h2>
        <table border = "1">
          <tr bgcolor = "pink">
            <th> ID</th>
            <th> First Name</th>
            <th> Last Name</th>
            <th> Nick Name</th>
            <th> Salary</th>
          </tr>
          <xsl:for-each select = "class/employee">
            <tr>
              <td> <xsl:value-of select = "@id"/> </td>
              <td> <xsl:value-of select = "firstname"/> </td>
              <td> <xsl:value-of select = "lastname"/> </td>
              <td> <xsl:value-of select = "nickname"/> </td>
              <td> <xsl:value-of select = "salary"/> </td>
            </tr>
          </xsl:for-each>
        </table>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

ABSOLUTE PATH

File name: employee2.xml

```
<?xml version = "1.0"?>
<?xml-stylesheet type = "text/xsl" href = "employee2.xsl"?>
<class>
  <employee id = "001">
    <firstname>Abhiram</firstname>
    <lastname>Kushwaha</lastname>
    <nickname>Manoj</nickname>
    <salary>15000</salary>
  </employee>
  <employee id = "002">
    <firstname>Akash</firstname>
    <lastname>Singh</lastname>
    <nickname>Bunty</nickname>
    <salary>25000</salary>
  </employee>
  <employee id = "003">
    <firstname>Brijesh</firstname>
    <lastname>Kaushik</lastname>
    <nickname>Ballu</nickname>
    <salary>20000</salary>
  </employee>
  <employee id = "004">
    <firstname>Zoya</firstname>
    <lastname>Mansoori</lastname>
    <nickname>Sonam</nickname>
    <salary>30000</salary>
  </employee>
</class>
```

File name: employee2.xsl

```
<?xml version = "1.0" encoding = "UTF-8"?>
<xsl:stylesheet version = "1.0"
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">
  <xsl:template match = "/" >
    <html>
      <body>
        <h3>Details of each Employee. </h3>
        <table border = "1">
          <tr bgcolor = "pink">
            <th>ID</th>
```

```

        <th>First Name</th>
        <th>Last Name</th>
        <th>Nick Name</th>
        <th>Salary</th>
    </tr>
    <tr>
        <td><xsl:value-of select = "/class/employee[1]/@id"/></td>
        <td><xsl:value-of select = "/class/employee[1]/firstname"/></td>
        <td><xsl:value-of select = "/class/employee[1]/lastname"/></td>
        <td><xsl:value-of select = "/class/employee[1]/nickname"/></td>
        <td><xsl:value-of select = "/class/employee[1]/salary"/></td>
    </tr>
    <tr>
        <td><xsl:value-of select = "/class/employee[2]/@id"/></td>
        <td><xsl:value-of select = "/class/employee[2]/firstname"/></td>
        <td><xsl:value-of select = "/class/employee[2]/lastname"/></td>
        <td><xsl:value-of select = "/class/employee[2]/nickname"/></td>
        <td><xsl:value-of select = "/class/employee[2]/salary"/></td>
    </tr>
    <tr>
        <td><xsl:value-of select = "/class/employee[3]/@id"/></td>
        <td><xsl:value-of select = "/class/employee[3]/firstname"/></td>
        <td><xsl:value-of select = "/class/employee[3]/lastname"/></td>
        <td><xsl:value-of select = "/class/employee[3]/nickname"/></td>
        <td><xsl:value-of select = "/class/employee[3]/salary"/></td>
    </tr>
    <tr>
        <td><xsl:value-of select = "/class/employee[4]/@id"/></td>
        <td><xsl:value-of select = "/class/employee[4]/firstname"/></td>
        <td><xsl:value-of select = "/class/employee[4]/lastname"/></td>
        <td><xsl:value-of select = "/class/employee[4]/nickname"/></td>
        <td><xsl:value-of select = "/class/employee[4]/salary"/></td>
    </tr>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

RELATIVE PATH

File name: employee3.xml

```

<?xml version = "1.0"?>
<?xml-stylesheet type = "text/xsl" href = "employee3.xsl"?>

```

```

<class>
  <employee id = "001">
    <firstname>Abhiram</firstname>
    <lastname>Kushwaha</lastname>
    <nickname>Manoj</nickname>
    <salary>15000</salary>
  </employee>
  <employee id = "002">
    <firstname>Akash</firstname>
    <lastname>Singh</lastname>
    <nickname>Bunty</nickname>
    <salary>25000</salary>
  </employee>
  <employee id = "003">
    <firstname>Brijesh</firstname>
    <lastname>Kaushik</lastname>
    <nickname>Ballu</nickname>
    <salary>20000</salary>
  </employee>
  <employee id = "004">
    <firstname>Zoya</firstname>
    <lastname>Mansoori</lastname>
    <nickname>Sonam</nickname>
    <salary>30000</salary>
  </employee>
</class>

```

File name: employee3.xsl

```

<?xml version = "1.0" encoding = "UTF-8"?>
<xsl:stylesheet version = "1.0"
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">
  <xsl:template match = "/">
    <html>
      <body>
        <h3>Details of each Employee. </h3>
        <table border = "1">
          <tr bgcolor = "pink">
            <th>ID </th>
            <th>First Name </th>
            <th>Last Name </th>
            <th>Nick Name </th>
            <th>Salary </th>
          </tr>
          <xsl:for-each select = "/class/employee">
            <tr>

```

```

        <td><xsl:value-of select = "@id"/> </td>
        <td><xsl:value-of select = "firstname"/> </td>
        <td><xsl:value-of select = "lastname"/> </td>
        <td><xsl:value-of select = "nickname"/> </td>
        <td><xsl:value-of select = "salary"/> </td>
    </tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

WILDCARD PATH

File name: employee4.xml

```

<?xml version = "1.0"?>
<?xml-stylesheet type = "text/xsl" href = "employee4.xsl"?>
<class>
    <employee id = "001">
        <firstname>Abhiram</firstname>
        <lastname>Kushwaha</lastname>
        <nickname>Manoj</nickname>
        <salary>15000</salary>
    </employee>
    <employee id = "002">
        <firstname>Akash</firstname>
        <lastname>Singh</lastname>
        <nickname>Bunty</nickname>
        <salary>25000</salary>
    </employee>
    <employee id = "003">
        <firstname>Brijesh</firstname>
        <lastname>Kaushik</lastname>
        <nickname>Ballu</nickname>
        <salary>20000</salary>
    </employee>
    <employee id = "004">
        <firstname>Zoya</firstname>
        <lastname>Mansoori</lastname>
        <nickname>Sonam</nickname>
        <salary>30000</salary>
    </employee>
</class>

```

File name: employee4.xsl

```
<?xml version = "1.0" encoding = "UTF-8"?>
<xsl:stylesheet version = "1.0"
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">
  <xsl:template match = "/">
    <html>
      <body>
        <h2>Employee </h2>
        <xsl:apply-templates select = "class/*" />
      </body>
    </html>
  </xsl:template>
  <xsl:template match = "class/*">
    <xsl:apply-templates select = "@id" />
    <xsl:apply-templates select = "firstname" />
    <xsl:apply-templates select = "lastname" />
    <xsl:apply-templates select = "nickname" />
    <xsl:apply-templates select = "salary" />
    <br />
  </xsl:template>
  <xsl:template match = "@id">
    <span style = "font-size = 22px;">
      <xsl:value-of select = "." />
    </span>
    <br />
  </xsl:template>
  <xsl:template match = "firstname">
    First Name: <span style = "color:blue;">
      <xsl:value-of select = "." />
    </span>
    <br />
  </xsl:template>
  <xsl:template match = "lastname">
    Last Name: <span style = "color:green;">
      <xsl:value-of select = "." />
    </span>
    <br />
  </xsl:template>
  <xsl:template match = "nickname">
    Nick Name: <span style = "color:red;">
      <xsl:value-of select = "." />
    </span>
    <br />
  </xsl:template>
  <xsl:template match = "salary">
    Salary: <span style = "color:brown;">
      <xsl:value-of select = "." />
    </span>
    <br />
  </xsl:template>
</xsl:stylesheet>
```

PREDICATE

File name: employee5.xml

```
<?xml version = "1.0"?>
<?xml-stylesheet type = "text/xsl" href = "employee5.xsl"?>
<class>
  <employee id = "001">
    <firstname>Abhiram</firstname>
    <lastname>Kushwaha</lastname>
    <nickname>Manoj</nickname>
    <salary>15000</salary>
  </employee>
  <employee id = "002">
    <firstname>Akash</firstname>
    <lastname>Singh</lastname>
    <nickname>Bunty</nickname>
    <salary>25000</salary>
  </employee>
  <employee id = "003">
    <firstname>Brijesh</firstname>
    <lastname>Kaushik</lastname>
    <nickname>Ballu</nickname>
    <salary>20000</salary>
  </employee>
  <employee id = "004">
    <firstname>Zoya</firstname>
    <lastname>Mansoori</lastname>
    <nickname>Sonam</nickname>
    <salary>30000</salary>
  </employee>
</class>
```

File name: employee5.xsl

```
<?xml version = "1.0" encoding = "UTF-8"?>
<xsl:stylesheet version = "1.0"
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">
  <xsl:template match = "/">
    <html>
      <body>
```

```

<h2>Employee</h2>
<table border = "1">
  <tr bgcolor = "pink">
    <th>ID</th>
    <th>First Name</th>
    <th>Last Name</th>
    <th>Nick Name</th>
    <th>Salary</th>
  </tr>
  <xsl:for-each select = "/class/employee[1]">
    <tr>
      <td><xsl:value-of select = "@id"/></td>
      <td><xsl:value-of select = "firstname"/></td>
      <td><xsl:value-of select = "lastname"/></td>
      <td><xsl:value-of select = "nickname"/></td>
      <td><xsl:value-of select = "salary"/></td>
    </tr>
  </xsl:for-each>
  <xsl:for-each select = "/class/employee[last()]">
    <tr>
      <td><xsl:value-of select = "@id"/></td>
      <td><xsl:value-of select = "firstname"/></td>
      <td><xsl:value-of select = "lastname"/></td>
      <td><xsl:value-of select = "nickname"/></td>
      <td><xsl:value-of select = "salary"/></td>
    </tr>
  </xsl:for-each>
  <xsl:for-each select = "/class/employee[@id = 002]">
    <tr>
      <td><xsl:value-of select = "@id"/></td>
      <td><xsl:value-of select = "firstname"/></td>
      <td><xsl:value-of select = "lastname"/></td>
      <td><xsl:value-of select = "nickname"/></td>
      <td><xsl:value-of select = "salary"/></td>
    </tr>
  </xsl:for-each>
  <xsl:for-each select = "/class/employee[salary > 10000]">
    <tr>
      <td><xsl:value-of select = "@id"/></td>
      <td><xsl:value-of select = "firstname"/></td>
      <td><xsl:value-of select = "lastname"/></td>
      <td><xsl:value-of select = "nickname"/></td>
      <td><xsl:value-of select = "salary"/></td>
    </tr>
  </xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Predicate specifies XPath expressions written in square bracket.

It is used to restrict the selected nodes in a node set for some conditions.

XQUERY

<https://www.videlibri.de/cgi-bin/xidelcgi>

https://www.w3schools.com/xml/xquery_intro.asp