

employees.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE employees SYSTEM "employees.dtd">
<employees xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="employees.xsd">
  <employee id="1">
    <name>John Doe</name>
    <position>Software Engineer</position>
    <department>IT</department>
  </employee>
  <employee id="2">
    <name>Jane Smith</name>
    <position>HR Manager</position>
    <department>Human Resources</department>
  </employee>
  <employee id="3">
    <name>Emily Johnson</name>
    <position>Marketing Specialist</position>
    <department>Marketing</department>
  </employee>
  <employee id="4">
    <name>Michael Brown</name>
    <position>Product Manager</position>
    <department>Product</department>
  </employee>
  <employee id="5">
    <name>Sarah Davis</name>
    <position>Data Analyst</position>
    <department>Data</department>
  </employee>
</employees>
```

employees.dtd

```
<!ELEMENT employees (employee*)>
<!ELEMENT employee (name, position, department)>
<!ATTLIST employee id ID #REQUIRED>
<!ELEMENT name (#PCDATA)>
<!ELEMENT position (#PCDATA)>
<!ELEMENT department (#PCDATA)>
```

employees.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="employees">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="employee" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="name" type="xs:string"/>
              <xs:element name="position" type="xs:string"/>
              <xs:element name="department" type="xs:string"/>
            </xs:sequence>
            <xs:attribute name="id" type="xs:integer" use="required"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

```

        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:schema>

```

employees.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>
            <head>
                <style>
                    table {
                        width: 100%;
                        border-collapse: collapse;
                    }
                    th, td {
                        border: 1px solid black;
                        padding: 15px;
                        text-align: left;
                    }
                    tr:nth-child(even) {
                        background-color: #f2f2f2;
                    }
                    th {
                        background-color: #4CAF50;
                        color: white;
                    }
                </style>
            </head>
            <body>
                <table>
                    <tr>
                        <th>ID</th>
                        <th>Name</th>
                        <th>Position</th>
                        <th>Department</th>
                    </tr>
                    <xsl:for-each select="employees/employee">
                        <tr>
                            <td><xsl:value-of select="@id"/></td>
                            <td><xsl:value-of select="name"/></td>
                            <td><xsl:value-of select="position"/></td>
                            <td><xsl:value-of select="department"/></td>
                        </tr>
                    </xsl:for-each>
                </table>
            </body>
        </html>
    </xsl:template>
</xsl:stylesheet>

```

output.html

```
<html>
  <head>
    <META http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <style>
      table {
        width: 100%;
        border-collapse: collapse;
      }
      th, td {
        border: 1px solid black;
        padding: 15px;
        text-align: left;
      }
      tr:nth-child(even) {
        background-color: #f2f2f2;
      }
      th {
        background-color: #4CAF50;
        color: white;
      }
    </style>
  </head>
  <body>
    <table>
      <tr>
        <th>ID</th><th>Name</th><th>Position</th><th>Department</th>
      </tr>
      <tr>
        <td>1</td><td>John Doe</td><td>Software Engineer</td><td>IT</td>
      </tr>
      <tr>
        <td>2</td><td>Jane Smith</td><td>HR Manager</td><td>Human Resources</td>
      </tr>
      <tr>
        <td>3</td><td>Emily Johnson</td><td>Marketing
          Specialist</td><td>Marketing</td>
      </tr>
      <tr>
        <td>4</td><td>Michael Brown</td><td>Product Manager</td><td>Product</td>
      </tr>
      <tr>
        <td>5</td><td>Sarah Davis</td><td>Data Analyst</td><td>Data</td>
      </tr>
    </table>
  </body>
</html>
```

XSLTTransform.java

```
import javax.xml.transform.*;
import javax.xml.transform.stream.*;
import java.io.File;

public class XSLTTransform {
    public static void main(String[] args) {
        try {
            TransformerFactory factory = TransformerFactory.newInstance();
            Source xsl = new StreamSource(new File("employees.xsl"));
            Transformer transformer = factory.newTransformer(xsl);

            Source xml = new StreamSource(new File("employees.xml"));
            transformer.transform(xml, new StreamResult(new File("output.html")));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

ID	Name	Position	Department
1	John Doe	Software Engineer	IT
2	Jane Smith	HR Manager	Human Resources
3	Emily Johnson	Marketing Specialist	Marketing
4	Michael Brown	Product Manager	Product
5	Sarah Davis	Data Analyst	Data