XML, XSLT – Transforming XML

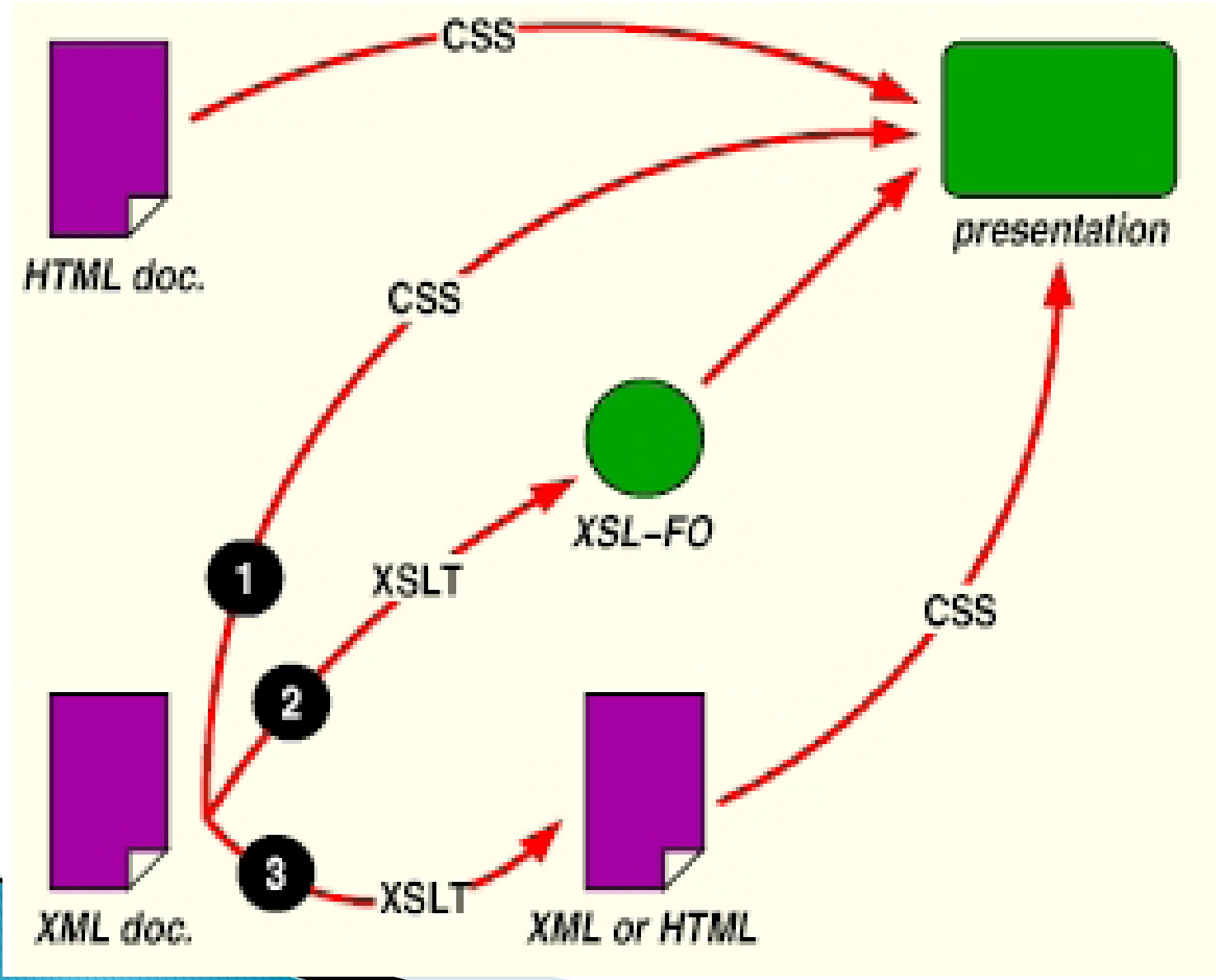
Introduction

- CSS → Used as a style sheet language for HTML.
- XSL → it is equivalent and used for XML.
- XSL covers a family of different languages.
- XSLT → XSL Transformations
  - It concerns the transforming of XML documents from one syntax to another.
- XSL:Fo → XSL Formatting Objects
  - It transform XML to several formats such as PDF, PostScript and text files.
- Xpath → XSLT used to access or refer to, parts of XML documents.

XML Transformed

- XSLT is used to change, and rearrange elements, add and sort to build a new document.
- It transforms a source XML tree into a resulting XML tree.
- XSLT uses XPath to specify and address parts of the source document's XML tree.

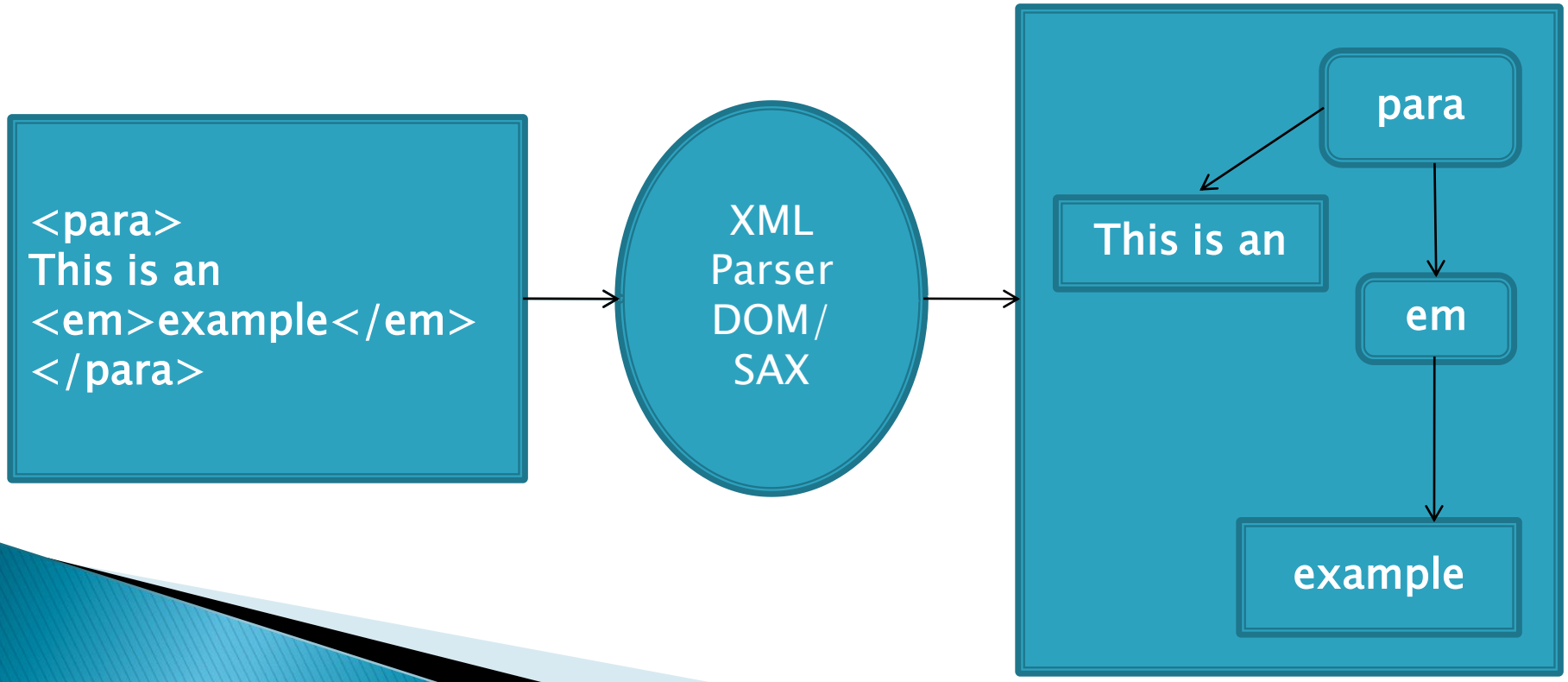
Publishing XML data



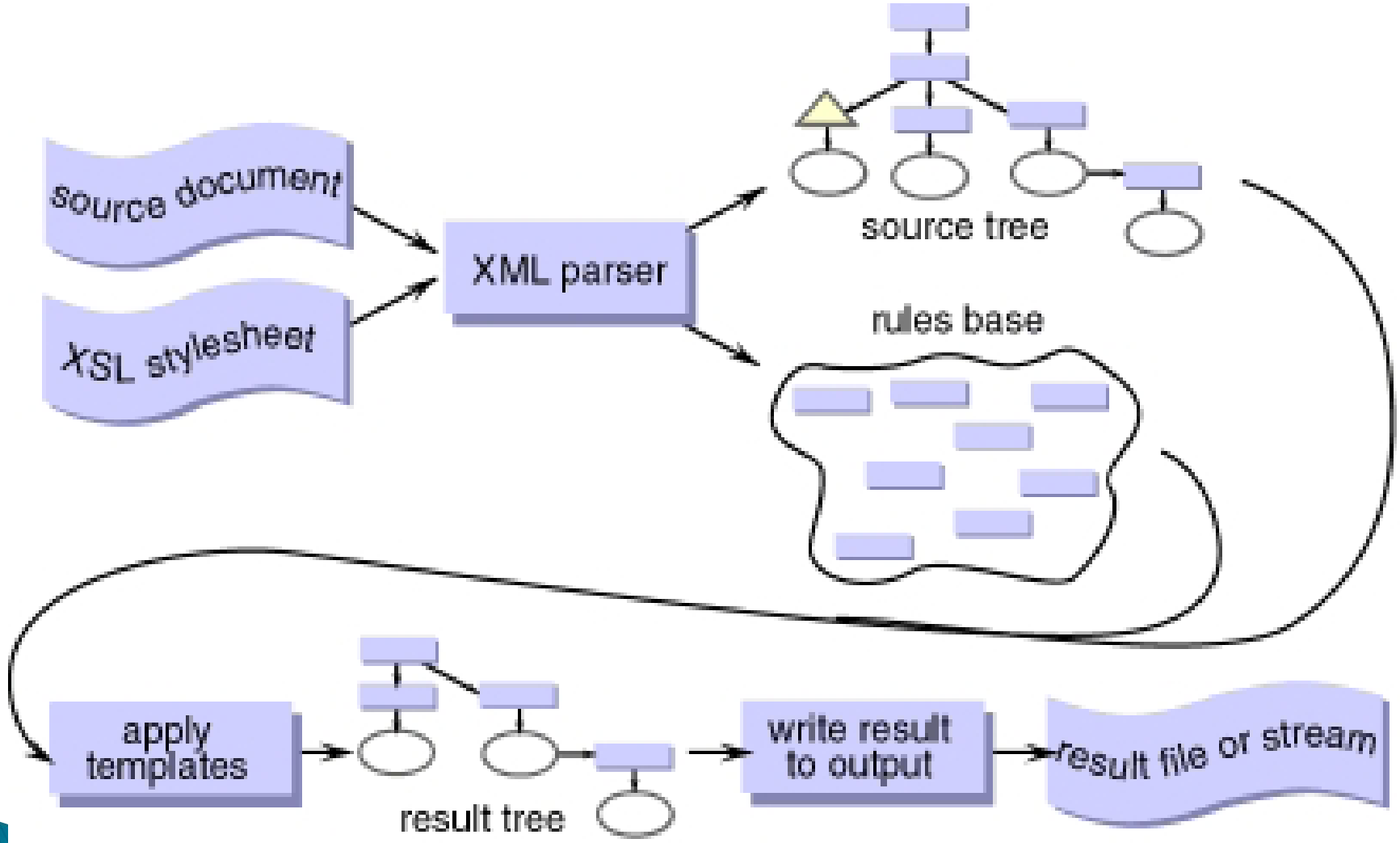
How XML data can be transformed using XSLT?

- A conversion of XML data into a tree structure, e.g. using an XML parser conformant to

1. Document Object Model (DOM) http://www.w3.org/DOM/
2. Simple Api for XML (SAX)
   http://www.megginson.com/SAX/sax.html



XML data transformed using XSLT



```
<?xml version="1.0" encoding="ISO-8859-1"
  ?>
<library>
  <book>
      <title>A search for the life</title>
      <author>Philip k. Mark</author>
      <publisher>Wiley - India</publisher>
      <price>350</price>
      <year>1982</year>
      <isbn>1-85759-452-6</isbn>
  </book>
</library>
```

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xsl:stylesheet version="1.0"
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
   <body>
   <h2>My Library </h2>
   <table border="1">
   <tr bgcolor="#9acd32">
         <th align="left">Title</th>
          <th align="left">Author</th>
   </tr>
   <xsl:for-each select="library/book">
   <tr>
   <td><xsl:value-of select="title"/></td>
   <td><xsl:value-of select="author"/></td>
   </tr>
    </xsl:for-each>
  </table>
 </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<?xml-stylesheet type="text/xsl"
  href="libstyle.xsl"?>
<library>
  <book>
      <title>A search for the life</title>
      <author>Philip k. Mark</author>
      <publisher>Wiley - India</publisher>
      <price>350</price>
      <year>1982</year>
      <isbn>1-85759-452-6</isbn>
  </book>
</library>
```

Style Sheet

- Style sheet provides the template that transforms the document from one structure to another.
- <xsl:template> starts the definition of the actual template.
- match="/" attribute is used to applying the template to the root of the source XML document.
- Here HTML is used for formatting and a table is applied.

Style Sheet

- <xsl:for-each select="library/book"> → each book element is visited in turn and the code inside the *for-each* structure repeated in turn.
- Each record is accessed by using:
  - <xsl:value-of select="title" />
  - <xsl:value-of select="author" />

Linking with .xml file

- Style sheet is linked into the XML by adding the below statement.
- <?xml-stylesheet type="text/xsl" href="libstyle.xsl"?>

XSL Elements

- value-of
  - Allows information to be extracted and added to the resulting document.
- for-each
  - Used to iterate and select through every XML element.
- sort
  - Information can be sorted for output, simply by adding an <xsl:sort> element inside a for-each.

XSL Elements

- if
  - <xsl:if> element allows the specifying of a query to extract information you may want.

- <xsl:if test="year > 2001">

- It simply checks whether the year is greater than 2001 and will only output those items that qualify.

XSL Elements

- Choices
  - When a multiple condition is required, <xsl:choose> is used with the <xsl:when> and <xsl:otherwise> elements.
- Basic Pattern

```
<xsl:choose>
    <xsl:when test “test expression”>
        output if true
    </xsl:when>
    <xsl:otherwise> output in other cases
    </xsl:otherwise>
</xsl:choose>
```

Applying Templates

- <xsl:apply-templates> → It is used to activates the other templates in the document.
- This calls the others in sequence to give desired output.

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transfor
  m">
<xsl:template match="/">
<html>
  <body>
      <h2> Book Library</h2>
      <xsl:apply-templates/>
  </body>
</html>
</xsl:template>
```

…. Continue on next slide

```
<xsl:template match="book">
  <p>
      <xsl:apply-templates select="title"/>
      <xsl:apply-templates select="author"/>
      <xsl:apply-templates select="price"/>
  </p>
  </xsl:template>
  <xsl:template match="title">
  Title: <span style="color:red">
  <xsl:value-of select="."/></span>
  <br />
  </xsl:template>
```

```
<xsl:template match="author">
Artist: <span style="color:blue">
<xsl:value-of select="."/></span>
<br />
</xsl:template>
<xsl:template match="price">
Price: <span style="color:green">
<xsl:value-of select="."/></span>
<br />
</xsl:template>
</xsl:stylesheet>
```