```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="html"/>
    <xsl:template match="/">
        <html>
          <head>
            <title>Wonders of the World</title>
          </head>
          <body>
            <p>
              <img src="images/herodotus.jpeg" width="120" height="171"/>
            </p>
            <p> The famous Greek historian Herodotus wrote of seven great archite
                achievements. Although his writings did not survive, he planted s
                for what has become the list of the <strong>Seven Wonders of the
                Ancient World</strong>
            </p>
          </body>
        </html>
    </xsl:template>
</xsl:stylesheet>
```

# XSLT

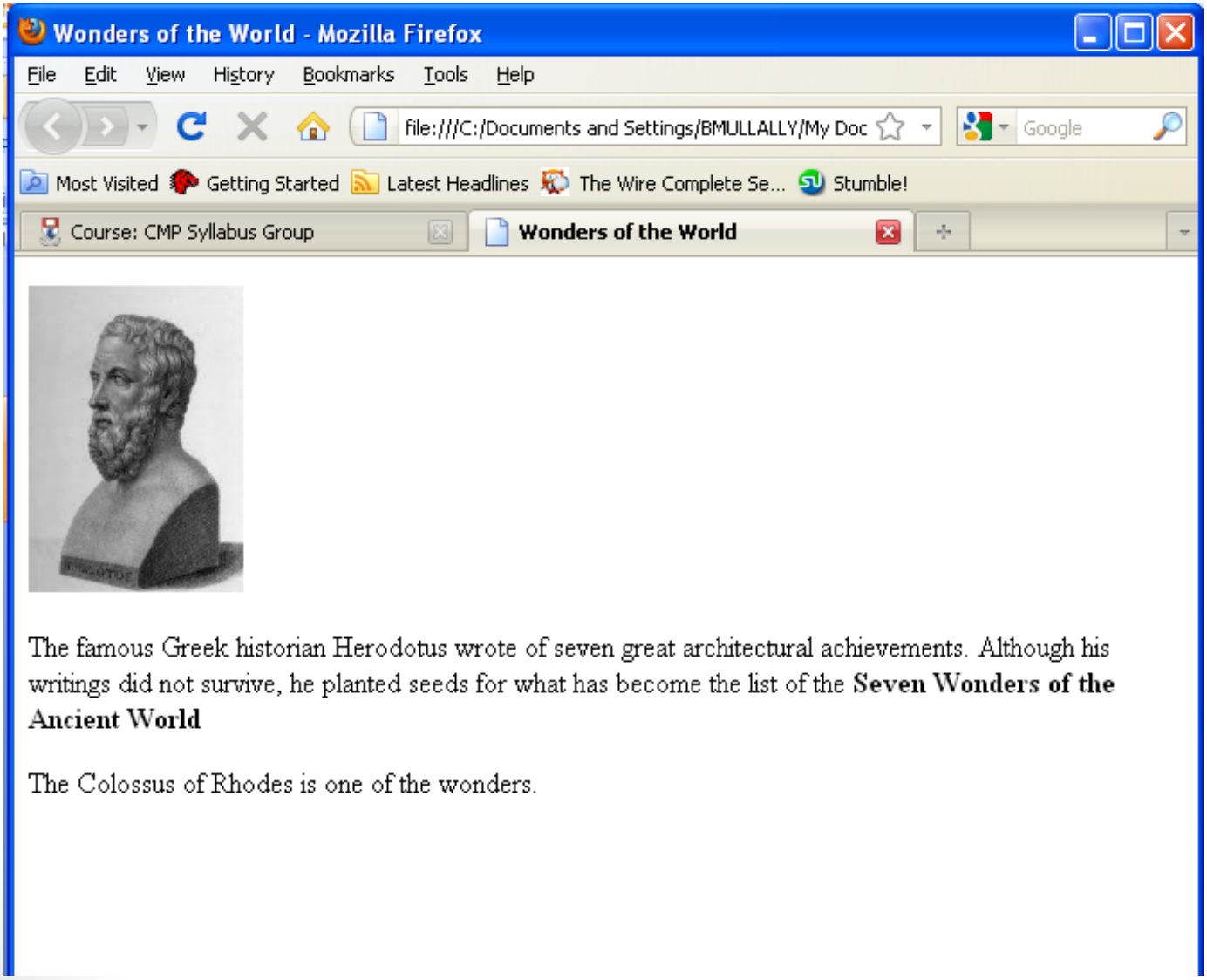
**Outputting Values:**

In order to actually output the content of an XML node we will use
the <xsl:value-of> element.

```
<p>The <xsl:value-of select="ancient_wonders/wonder/name"></xsl:value-of>
is one of the wonders.</p>
```

When the XSLT processor applies the root template it first outputs all the HTML header code. Then when it gets to the xsl:value-of element, it only outputs the value of the first node it finds, which is Colossus of Rhodes.

# XSLT


Wonders of the World - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///C:/Documents and Settings/BMULLALLY/My Doc
Google
Most Visited Getting Started Latest Headlines The Wire Complete Se... Stumble!
Course: CMP Syllabus Group
Wonders of the World
The famous Greek historian Herodotus wrote of seven great architectural achievements. Although his writings did not survive, he planted seeds for what has become the list of the Seven Wonders of the Ancient World
The Colossus of Rhodes is one of the wonders.

# XSLT

**Outputting Values:**

If the select expression (select=“”) matches more than one node in the XML document, only the first nodes value is output.
If I wanted to return the name nodes where the language attribute was equal to “Greek”, I would write:

<xsl:value-of select=“name[@language=‘Greek’]”/>

# XSLT

**Outputting Values:**

If the select expression matches a node, the string value of that node is output. If the node has child elements, the output includes the text contained in those child elements as well.

```
<p>The <xsl:value-of select="ancient_wonders/wonder"></xsl:value-of>
is one of the wonders.
</p>
```

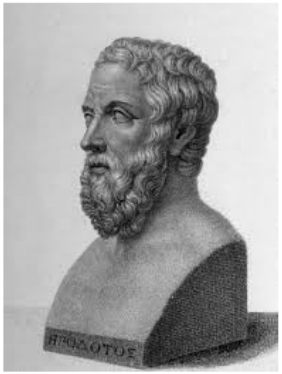
# XSLT

```
<?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>
              <title>Wonders of the world</title>

        </head>
        <body>
          <p>
            <img src="images/herodotus.jpg" />
          </p>
          <p>
          The famous Greek historian Herodotus wrote of seven great architectural achievements.
          Although his writings did not survive, he planted seeds for what has become the list of the Seven Wonders
          </p>
          <p> The <xsl:value-of select="ancient_wonders/wonder"></xsl:value-of> is one of the seven wonders.</p>
          </body>
          </html>

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

# XSLT

lisa.html | lisa.html | census.dtd | person.xml | Wonders of the world | tester.xsl | wonders_basic.xml

file:///C:/Users/bmullally/Desktop/WebDevWorkspace/xslt_basics/WebContent/wonders_basic.xml




The famous Greek historian Herodotus wrote of seven great architectural achievements. Although his writings did not survive, he planted seeds for what has become the list of the Seven Wonders of the Acient World

The Colossus of Rhodes Κολοσσός της Ρόδου Rhodes, Greece 107 282 226 earthquake In 294 BC, the people of the island of Rhodes began building a colossal statue of the sun god Helios. They believed that it was because of his blessings that they were able to withstand a long siege on the island and emerge victorious. The Colossus was built with bronze, reinforced with iron, and weighted with stones. While it is often depicted straddling Mandrákion harbor, this is now considered technically impossible; and therefore, it likely stood beside the harbor.The statue was toppled by an earthquake in 226 BC. It snapped at the knees and fell over on to the land. The Oracle of Delphi suggested that it fell because the people of Rhodes had offended Helios, and they decided not to rebuild it.The statue remained on the ground until 654 AD., and even broken, it was so impressive that many traveled to see it. is one of the seven wonders.

# XSLT

**Outputting Values:**

If the select expression matches a node set that is empty, there is nothing to output.

# XSLT

**Looping over nodes**
As you saw in the previous slides the <xsl:value-of> element will only act on one node, even if there are many nodes that it matches.

The <xsl:for-each> element allows you to act on all nodes matched. It processes all the nodes matched by its select attributes, one after the other.

Within the root template you include the following:

# XSLT

```
<table border="1">
  <tr>
    <th>Wonder Name</th>
    <th>Location</th>
    <th>Height</th>
  </tr>
  <xsl:for-each select="ancient_wonders/wonder">
  <tr>
    <td><xsl:value-of select="name[@language='English']"></xsl:value-of>(
        <xsl:value-of select="name[@language!='English']"></xsl:value-of>)</td>
    <td><xsl:value-of select="location"></xsl:value-of></td>
    <td><xsl:value-of select="height"></xsl:value-of></td>
  </tr>
  </xsl:for-each>
</table>
```

# XSLT

The famous Greek historian Herodotus wrote of seven great architectural achievements. Although his v
list of the **Seven Wonders of the Ancient World**

| Wonder Name | Location | Height |
|---|---|---|
| Colossus of Rhodes( Κολοσσός της Ρόδου) | Rhodes, Greece | 107 |
| Great Pyramid of Giza( ) | Giza, Egypt | 455 |
| Hanging Gardens of Babylon( ) | Al Hillah, Iraq | 0 |
| Statue of Zeus at Olympia( Δίας μυθολογία) | Olympia, Greece | 39 |
| Temple of Artemis at Ephesus( Ἀρτεμίσιον) | Ephesus, Turkey | 60 |
| Mausoleum at Halicarnassus( Μαυσωλεῖον Ἁλικαρνασσεύς) | Bodrum, Turkey | 135 |
| Lighthouse of Alexandria( ὁ Φάρος τῆς Ἀλεξανδρείας) | Alexandria, Egypt | 384 |

# XSLT

The xsl:for-each instruction creates a new table row for each wonder of the world.

Once it has processed all of the nodes in the selected set (in this case ancient_wonders/wonder), it continues with the rest of the template.

In general, place the <xsl:for-each> right before the rules that should be repeated for each node to be found.

When accessing nodes to add to a table you put the xsl:for-each before and after the opening and closing tags for the table row.

- In the select condition of the xsl:for-each element you can require a specific attribute match by using the

  [@attribute='expression'].

- This is the same syntax as seen in the <xsl:value-of> elements.

# XSLT

**Processing Nodes Conditionally**

It is not uncommon to want to process a node or a set of nodes only if a certain condition is met. The condition is written as an expression. For example, you might want to perform a certain action if a particular node set is not empty, or if the string value of a node is equal to a particular word.

**<xsl:if test="expression"></xsl:if>**

# XSLT

```
<xsl:for-each select="ancient_wonders/wonder">
<tr>
  <td><xsl:value-of select="name[@language='English']"></xsl:value-of>
      <xsl:if test="name[@language!='English']">(
        <xsl:value-of select="name[@language!='English']"></xsl:value-of>)
      </xsl:if>
 </td>
  <td><xsl:value-of select="location"></xsl:value-of></td>
  <td><xsl:value-of select="height"></xsl:value-of></td>
</tr>
</xsl:for-each>
```

The <xsl:if test> condition tests to see if the current node is a name with a language attribute that is not English. If so, then it outputs the value of the name node. If not, then nothing is done. This prevents displaying a set of empty brackets when no other language name exists.

# XSLT

The famous Greek historian Herodotus wrote of seven great architectural achievements. Althou
list of the **Seven Wonders of the Ancient World**

| Wonder Name | Location | Height |
|---|---|---|
| Colossus of Rhodes( Κολοσσός της Ρόδου) | Rhodes, Greece | 107 |
| Great Pyramid of Giza | Giza, Egypt | 455 |
| Hanging Gardens of Babylon | Al Hillah, Iraq | 0 |
| Statue of Zeus at Olympia( Δίας μυθολογία) | Olympia, Greece | 39 |
| Temple of Artemis at Ephesus( Ἀρτεμίσιον) | Ephesus, Turkey | 60 |
| Mausoleum at Halicarnassus( Μαυσωλεῖον Ἁλικαρνασσεύς) | Bodrum, Turkey | 135 |
| Lighthouse of Alexandria( ὁ Φάρος τῆς Ἀλεξανδρείας) | Alexandria, Egypt | 384 |

# XSLT

When referring to a node set in the expression, the test returns true if the node set is not empty; that is, if it contains at least one node.

If you want to be able to specify an alternate result when an expression is false, then you must use <xsl:choose> .

You can test for all sorts of conditions, we will look at constructing more elaborate test expressions later.

# XSLT

**Adding conditional  choices:**

The <xsl:if> instruction only allows one condition and one resulting action. You can use <xsl:choose> when you want to test for several different conditions, and react accordingly to each one. The simplest example of this is when you want to do one action when the condition is true, and another action when it's false.

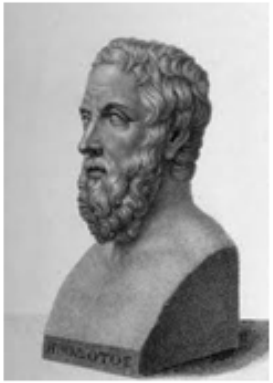
```
<td><xsl:choose>
    <xsl:when test="height !=0">
      <xsl:value-of select="height"></xsl:value-of>
    </xsl:when>
    <xsl:otherwise>unknown</xsl:otherwise>
    </xsl:choose>
</td>
```

# XSLT

In the case of multiple conditions, once a condition is found to be true, all the remaining conditions are ignored. The action contained in this first true condition is the only one performed.

By using the <xsl:choose>, instead of showing a height of zero for the Hanging Gardens of Babylon, the word unknown is displayed.

# XSLT



The famous Greek historian Herodotus wrote of seven great architectural achievements. Although h
what has become the list of the **Seven Wonders of the Ancient World**

| Wonder Name | Location | Height |
|---|---|---|
| Colossus of Rhodes( Κολοσσός της Ρόδου) | Rhodes, Greece | 107 |
| Great Pyramid of Giza | Giza, Egypt | 455 |
| Hanging Gardens of Babylon | Al Hillah, Iraq | unknown |
| Statue of Zeus at Olympia( Δίας μυθολογία) | Olympia, Greece | 39 |
| Temple of Artemis at Ephesus( Ἀρτεμίσιον) | Ephesus, Turkey | 60 |
| Mausoleum at Halicarnassus( Μαυσωλεῖον Ἁλικαρνασσεύς) | Bodrum, Turkey | 135 |
| Lighthouse of Alexandria( ὁ Φάρος τῆς Ἀλεξανδρείας) | Alexandria, Egypt | 384 |

```
<table border="1">
  <tr>
    <th>Wonder Name</th>
    <th>Location</th>
    <th>Height</th>
  </tr>
  <xsl:for-each select="ancient_wonders/wonder">
  <tr>
    <td><xsl:value-of select="name[@language='English']"></xsl:value-of>
        <xsl:if test="name[@language!='English']">(
          <xsl:value-of select="name[@language!='English']"></xsl:value-of>)
        </xsl:if>
   </td>
    <td><xsl:value-of select="location"></xsl:value-of></td>
    <td><xsl:choose>
        <xsl:when test="height !=0">
          <xsl:value-of select="height"></xsl:value-of>
        </xsl:when>
        <xsl:otherwise>unknown</xsl:otherwise>
        </xsl:choose>
    </td>
  </tr>
  </xsl:for-each>
</table>
```

```
<table border="1">
  <tr>
    <th>Wonder Name</th>
    <th>Location</th>
    <th>Height</th>
  </tr>
  <xsl:for-each select="ancient_wonders/wonder">
  <tr>
    <td><xsl:value-of select="name[@language='English']"></xsl:value-of>
        <xsl:if test="name[@language!='English']">(
          <xsl:value-of select="name[@language!='English']"></xsl:value-of>)
        </xsl:if>
    </td>
    <td><xsl:value-of select="location"></xsl:value-of></td>
    <td><xsl:choose>
        <xsl:when test="height !=0">
          <xsl:value-of select="height"></xsl:value-of>
        </xsl:when>
        <xsl:otherwise>unknown</xsl:otherwise>
        </xsl:choose>
    </td>
  </tr>
  </xsl:for-each>
</table>
```

Name of the wonder is outputed in English and in an alternative language if it exists.

Condition to ouput height if it is greater than zero, otherwise output "unknown".

Loop to move through each wonder node.