XSLT

What is XSL?

- XSL - EXtensible Stylesheet Language
- It is W3 standard

CSS = Style Sheets for HTML

CSS

- HTML uses predefined tags, and the meaning of each tag is well understood.
- The <table> tag in HTML defines a table - and a browser knows how to display it.
- Adding styles to HTML elements are simple. Telling a browser to display an element in a special font or color, is easy with CSS.

XSL = Style Sheets for XML

- XML does not use predefined tags and meaning of each tag is not well understood.
- A <table> tag
  - could mean an HTML table, a piece of furniture, or something else
  - browser does not know how to display it.
- XSL describes how the XML document should be displayed!

XSL - More Than a Style Sheet Language

- XSL consists of three parts
  - XSLT - a language for transforming XML documents
  - XPath - a language for navigating in XML documents
  - XSL-FO - a language for formatting XML documents

...Contd

XSLT

A language for Transforming XML document

XPath

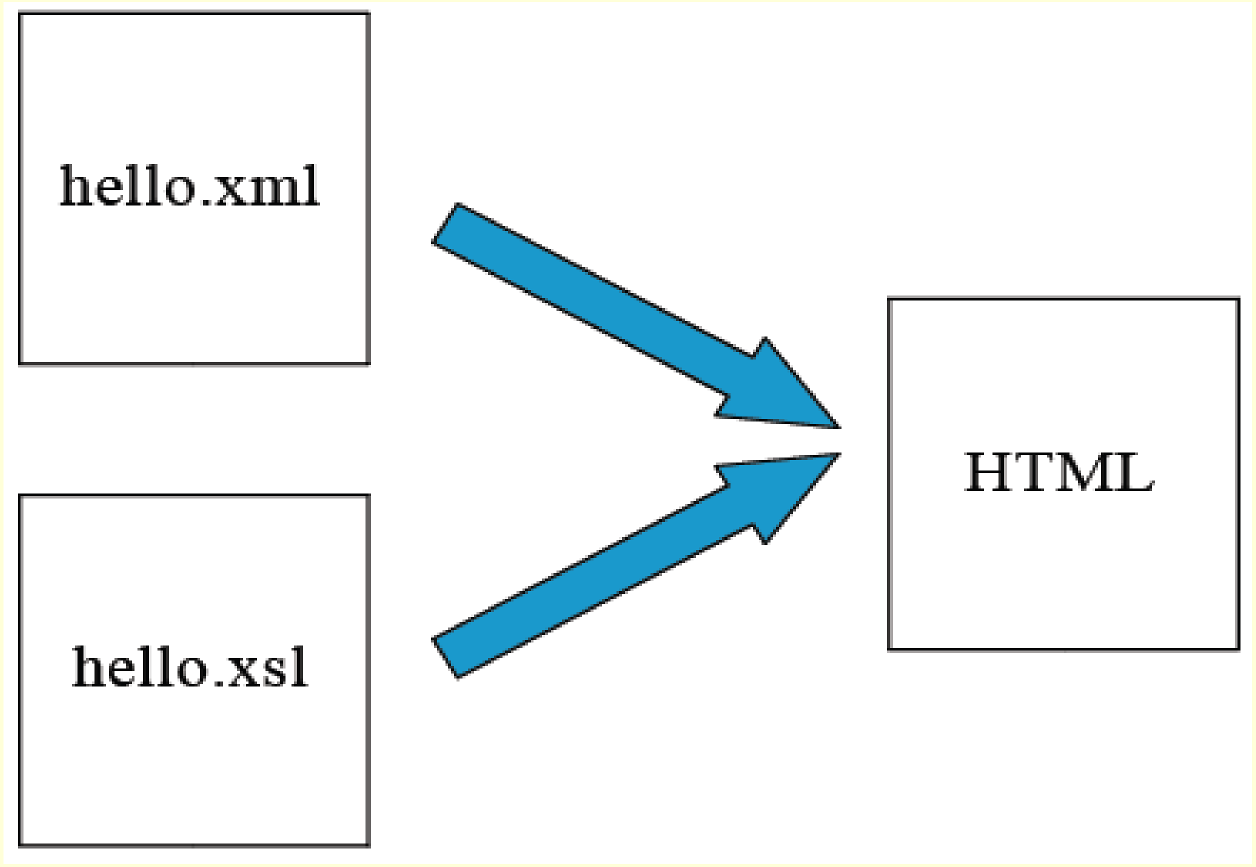
Expression language used by XSLT to locate

elements and attributes in an XML doc.

XSL-FO (Formatting Objects)

Specifies formatting properties for rendering the

XML Transformation Example



Elements in XSLT - Examples

1. <xsl:template match="/">

</xsl:template>

2. <xsl:value-of select="catalog/cd/title”</xsl:value>

3. <xsl:for-each select="catalog/cd">

</xsl:for-each>

4. <xsl:sort select="artist"/>

...Contd

```
 5. <xsl:if test="price &gt; 10">
  <xsl:value-of select="title"/>
 </xsl:if>
6. <xsl:choose>
  <xsl:when test="expression">
    ... some output ...
  </xsl:when>
  <xsl:otherwise>
    ... some output ....
  </xsl:otherwise>
```

XSLT Stylesheet elements

Matching and selection templates

- xsl:template
- xsl:applytemplates
- xsl:valueof

Branching elements

- xsl:foreach
- xsl:if
- xsl:choose

Sorting element

- xsl:sort

Sample cd.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href=“cd.xsl"?>
<catalog>
  <cd>
    <title>Empire Burlesque</title>
    <artist>Bob Dylan</artist>
    <country>USA</country>
    <company>Columbia</company>
    <price>10.90</price>
    <year>1985</year>
  </cd>
.
.
</catalog>
```

Book.xml

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Edited by XMLSpy® -->
<store>
<book title="Computer Networks" lang="tamil">
<author>Andrew</author>
<author>Tanenbaum</author>
<price>130</price>
</book>
<book title="Web Technology" lang="en">
<author>Jeffrey</author>
<author>Jackson</author>
<price>250</price>
</book>
</store>
```

Xsl:template

An XSL style sheet consists of one or more set of rules that are called templates.

Each template contains rules to apply when a specified node is matched.

The `<xsl:template>` element is used to build templates.

The `match` attribute is used to associate a template with an XML element.

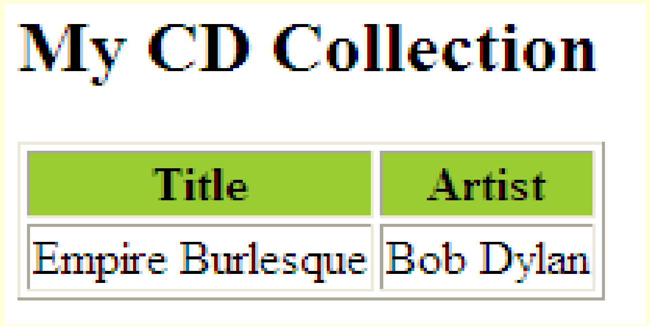
The value of the match attribute is an XPath expression.

Sample cd.xsl

```
<?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 CD Collection</h2>
    <table border="1">
      <tr bgcolor="#9acd32">
        <th>Title</th>
        <th>Artist</th>
      </tr>
```

…Contd

```
xsl:for-each select="catalog/cd">
        <tr>
          <td><xsl:value-of select="title"/></td>
          <td><xsl:value-of select="artist"/></td>
        </tr>
        </xsl:for-each>
      </table>
      </body>
      </html>
    </xsl:template>

    </xsl:stylesheet>
```



My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan

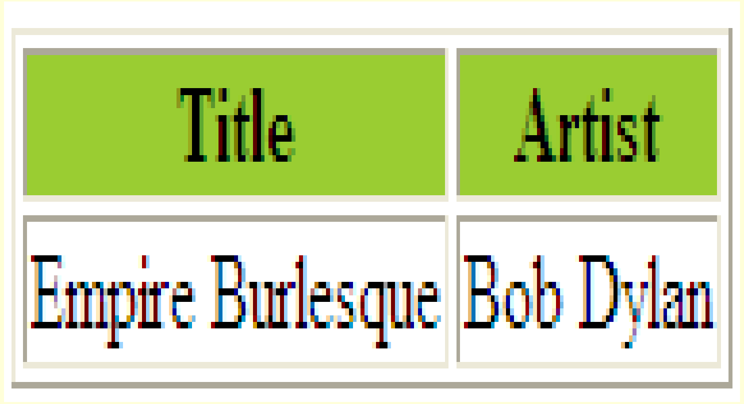
<xsl:for-each>

<xsl:for-each> – allows you to do looping in XSLT.

```
<xsl:template match="/">
      <html>
      <body>
      <h2>My CD Collection</h2>
      <table border="1">
<tr bgcolor="#9acd32"> <th>Title</th><th>Artist</th></tr>
        <xsl:for-each select="catalog/cd">
        <tr>
          <td><xsl:value-of select="title"/></td>
          <td><xsl:value-of select="artist"/></td>
        </tr>
        </xsl:for-each>
      </table>
      </body>
      </html>
    </xsl:template>
    </xsl:stylesheet>
```

<xsl:for-each> Predicate

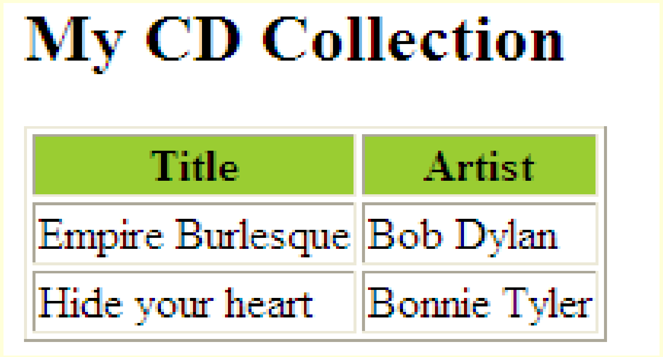
```
<xsl:for-each select="catalog/cd[artist='Bob Dylan']">
      <tr>
        <td><xsl:value-of select="title"/></td>
        <td><xsl:value-of select="artist"/></td>
      </tr>
</xsl:for-each>
```



Title	Artist
Empire Burlesque	Bob Dylan

<xsl:sort>

```
<xsl:template match="/">
   <html>
   <body>
   <h2>My CD Collection</h2>
   <table border="1">
<tr bgcolor="#9acd32"> <th>Title</th> <th>Artist</th> </tr>
    <xsl:for-each select="catalog/cd">
     <xsl:sort select="artist"/>
     <tr>
      <td><xsl:value-of select="title"/></td>
      <td><xsl:value-of select="artist"/></td>
     </tr>
    </xsl:for-each>
   </table>
   </body>
   </html>
  </xsl:template>
  </xsl:stylesheet>
```

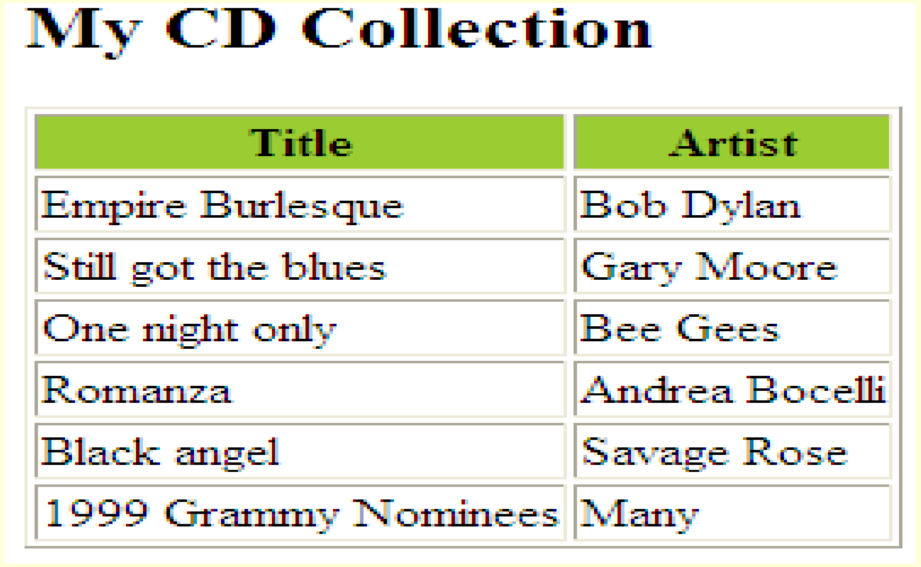


My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler

<xsl:if>

```
<xsl:for-each select="catalog/cd">
         <xsl:if test="price &gt; 10">
           <tr>
             <td><xsl:value-of select="title"/></td>
             <td><xsl:value-of select="artist"/></td>
           </tr>
         </xsl:if>
       </xsl:for-each>
```



My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan
Still got the blues	Gary Moore
One night only	Bee Gees
Romanza	Andrea Bocelli
Black angel	Savage Rose
1999 Grammy Nominees	Many

The <xsl:choose> Element

<xsl:choose> – used in conjunction with <xsl:when> and <xsl:otherwise> to express multiple conditional tests.

```
<xsl:choose>
<xsl:when test="expression">
... some output ...
</xsl:when>
<xsl:otherwise>
... some output ....
</xsl:otherwise>
</xsl:choose>
```

<xsl:choose>

```
<xsl:for-each select="catalog/cd">
        <tr>
          <td><xsl:value-of select="title"/></td>
          <xsl:choose>
            <xsl:when test="price &gt; 10">
              <td bgcolor="#ff00ff">
              <xsl:value-of select="artist"/></td>
            </xsl:when>
            <xsl:otherwise>
              <td><xsl:value-of select="artist"/></td>
            </xsl:otherwise>
          </xsl:choose>
        </tr>
        </xsl:for-each>
```

<xsl:applytemplates>

<xsl:applytemplates>

– applies a template to the current element or to the current element's child nodes.

The select attribute to <xsl:apply-templates> element, will process only the child element that matches the value of the attribute.

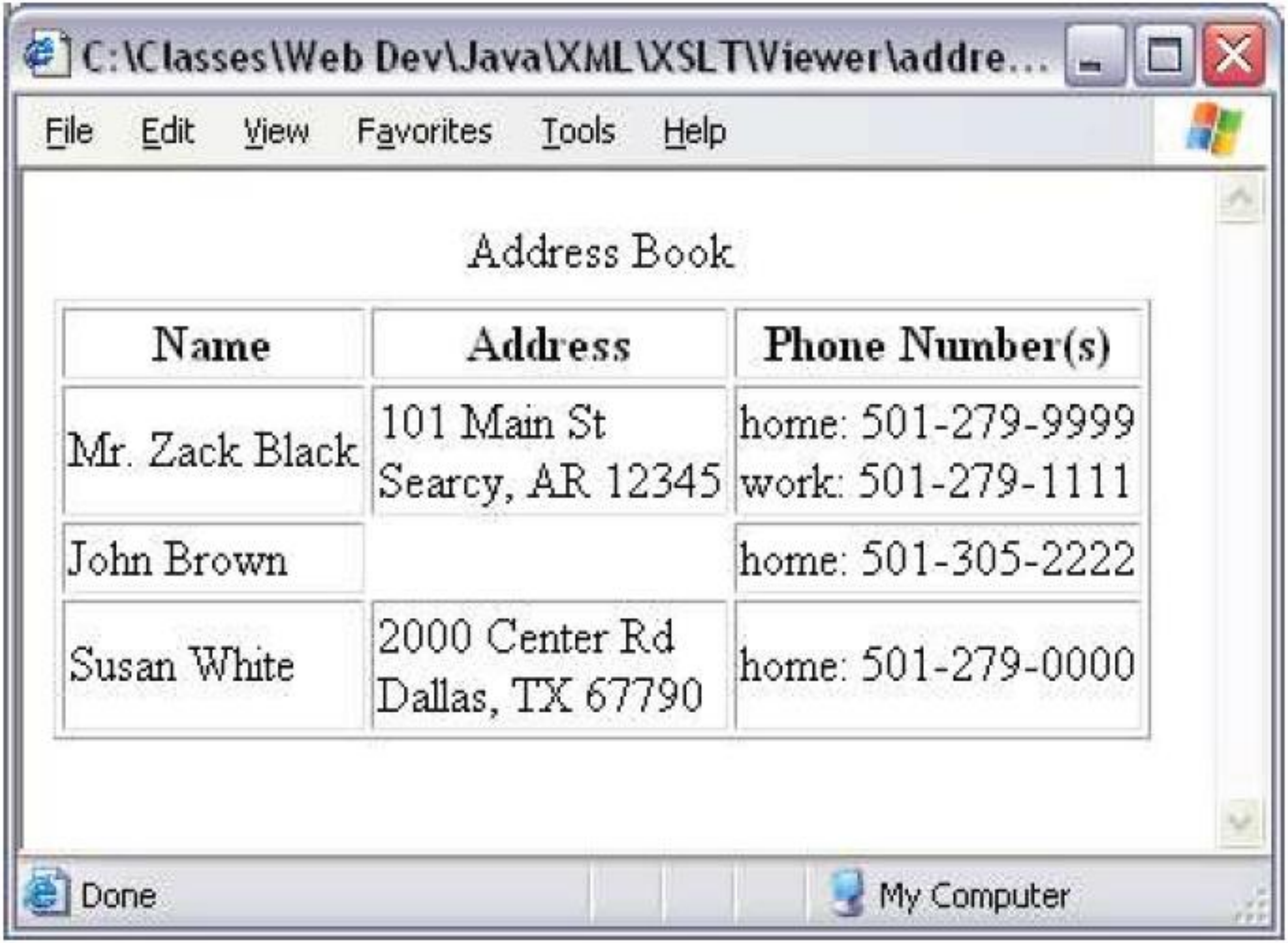
Use the select attribute to specify the order in which the child nodes are processed.

Address Book Example

```
<addressbook>
 <entry>
  <name prefix="Mr.">
      Zack Black</name>
  <address>
   <street>101 Main St</street>
   <city>Searcy</city>
   <state>AR</state>
   <zip>12345</zip>
  </address>
  <phone location="home"
      number="501-279-9999" />
  <phone location="work"
      number="501-279-1111" />
 </entry>

 <entry>
  <name>John Brown</name>
  <phone number="501-305-2222" />
 </entry>

 <entry>
  <name>Susan White</name>
  <address>
   <street>2000 Center Rd</street>
   <city>Dallas</city>
   <state>TX</state>
   <zip>67790</zip>
  </address>
  <phone location="home“
       number="501-279-0000" />
 </entry>
</addressbook>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<xsl:stylesheet
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
 <xsl:output method="html"/>

 <xsl:template match="/addressbook">
   <html><body>

   <table border="1">
     <caption>Address Book</caption>
     <tr>
      <th>Name</th><th>Address</th><th>Phone Number(s)</th>
     </tr>
     <xsl:apply-templates select="entry"/>
   </table>

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

```
<xsl:template match="entry">
   <tr>
     <td><xsl:apply-templates select="name"/></td>
     <td><xsl:apply-templates select="address"/></td>
     <td><xsl:apply-templates select="phone"/></td>
   </tr>
  </xsl:template>

  <xsl:template match="name">
    <xsl:value-of select="@prefix"/> <xsl:text> </xsl:text>
    <xsl:value-of select="."/>
  </xsl:template>

  <xsl:template match="address">
    <xsl:value-of select="street"/><br/>
    <xsl:value-of select="city"/>, <xsl:value-of select="state"/>
       <xsl:text> </xsl:text>
    <xsl:value-of select="zip"/>
  </xsl:template>

  <xsl:template match="phone">
    <xsl:value-of select="@location"/>: <xsl:value-of select="@number"/>
    <br/>
  </xsl:template>
</xsl:stylesheet>
```


C:\Classes\Web Dev\Java\XML\XSLT\Viewer\addre...
File Edit View Favorites Tools Help
Done
My Computer

Address Book

Name	Address	Phone Number(s)
Mr. Zack Black	101 Main St Searcy, AR 12345	home: 501-279-9999 work: 501-279-1111
John Brown		home: 501-305-2222
Susan White	2000 Center Rd Dallas, TX 67790	home: 501-279-0000