**Unit – 3**

**XLink , XPath, XPointer, XSLT & CSS**

**1. XLink**

XLink (the XML Linking language) defines methods for creating links within XML documents.

## What is XLink?

|  |  |
| --- | --- |
| XPath | * XLink is used to create hyperlinks within XML documents * Any element in an XML document can behave as a link * XLink supports simple links (like HTML) and extended links (for linking multiple resources together) * With XLink, the links can be defined outside the linked files * XLink is a W3C Recommendation |

## XLink Syntax

In HTML, the <a> element defines a hyperlink. However, this is not how it works in XML. In XML documents, you can use whatever element names you want - therefore it is impossible for browsers to predict what link elements will be called in XML documents.

Below is a simple example of how to use XLink to create links in an XML document:

<?xml version="1.0" encoding="UTF-8"?>  
  
<homepages xmlns:xlink="http://www.w3.org/1999/xlink">  
  <homepage xlink:type="simple" xlink:href="http://www.w3schools.com">Visit W3Schools</homepage>  
  <homepage xlink:type="simple" xlink:href="http://www.w3.org">Visit W3C</homepage>  
</homepages>

To get access to the XLink features we must declare the XLink namespace. The XLink namespace is: "http://www.w3.org/1999/xlink".

The xlink:type and the xlink:href attributes in the <homepage> elements come from the XLink namespace.

The xlink:type="simple" creates a simple "HTML-like" link (means "click here to go there").

The xlink:href attribute specifies the URL to link to.

## XLink Browser Support

There is no browser support for XLink in XML documents. However, all major browsers support [simple XLinks in SVG](http://www.w3schools.com/svg/svg_text.asp).

## XLink Example

The following XML document contains XLink features:

<?xml version="1.0" encoding="UTF-8"?>  
  
<bookstore xmlns:xlink="http://www.w3.org/1999/xlink">  
  
<book title="Harry Potter">  
  <description xlink:type="simple" xlink:href="/images/HPotter.gif"  
  xlink:show="new">  
  As his fifth year at Hogwarts School of Witchcraft and  
  Wizardry approaches, 15-year-old Harry Potter is.......  
  </description>  
</book>  
  
<book title="XQuery Kick Start">  
  <description  
  xlink:type="simple"  
  xlink:href="/images/XQuery.gif"  
  xlink:show="new">  
  XQuery Kick Start delivers a concise introduction  
  to the XQuery standard.......  
  </description>  
</book>  
  
</bookstore>

**Example explained:**

* The XLink namespace is declared at the top of the document (xmlns:xlink="http://www.w3.org/1999/xlink")
* The xlink:type="simple" creates a simple "HTML-like" link
* The xlink:href attribute specifies the URL to link to (in this case - an image)
* The xlink:show="new" specifies that the link should open in a new window

## XLink - Going Further

In the example above we have demonstrated simple XLinks. XLink is getting more interesting when accessing remote locations as resources, instead of standalone pages. If we set the value of the **xlink:show** attribute to "embed", the linked resource should be processed inline within the page. When you consider that this could be another XML document you could, for example, build a hierarchy of XML documents.

You can also specify WHEN the resource should appear, with the xlink:actuate attribute.

## XLink Attribute Reference

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | Description |
| xlink:actuate | onLoad onRequest other none | Defines when the linked resource is read and shown:  –The optional xlink:actuate has four possible value   **OnLoad:** The link should be followed as soon as the application sees it.  **OnRequest**: The link should be followed when the user asks to follow it.  **Other**: Other markup in the document, not specified by xlink, determines when to follow the link.  **None**: No details are available |
| xlink:href | *URL* | Specifies the URL to link to |
| xlink:show | embed new replace other none | –It has five possible values:  **New** : shows the content in the new window  **Replace**: Shows the ending resource in the current window, replacing the current document  **Embed**: Embeds the ending resources in the current document at the location of the link element.  **Other**: Do something other than what new, replace, embed do.  **None**: Specifies no behavior  Default is "replace" |
| xlink:type | simple extended | Specifies the type of link  **Simple** : unidirectional link between two documents  **Extended**: multiple links between two or more than two documents |

**2. XPointer**

XPointer (the XML Pointer language) allows hyperlinks to point to specific parts (fragments) of XML documents.

## What is XPointer?

|  |
| --- |
| * XPointer allows the links to point to specific parts of an XML document * XPointer uses XPath expressions to navigate in the XML document * XPointer is a W3C Recommendation |

## XPointer Browser Support

There is no browser support for XPointer. But XPointer is used in other XML languages.

## XPointer Example

In this example, we will use XPointer in conjunction with XLink to point to a specific part of another document.

We will start by looking at the target XML document (the document we are linking to):

<?xml version="1.0" encoding="UTF-8"?>  
  
<dogbreeds>  
  
<dog breed="Rottweiler" id="Rottweiler">  
  <picture url="http://dog.com/rottweiler.gif" />  
  <history>The Rottweiler's ancestors were probably Roman  
  drover dogs.....</history>  
  <temperament>Confident, bold, alert and imposing, the Rottweiler  
  is a popular choice for its ability to protect....</temperament>  
</dog>  
  
<dog breed="FCRetriever" id="FCRetriever">  
  <picture url="http://dog.com/fcretriever.gif" />  
  <history>One of the earliest uses of retrieving dogs was to  
  help fishermen retrieve fish from the water....</history>  
  <temperament>The flat-coated retriever is a sweet, exuberant,  
  lively dog that loves to play and retrieve....</temperament>  
</dog>  
  
</dogbreeds>

Note that the XML document above uses id attributes on each element!

So, instead of linking to the entire document (as with XLink), XPointer allows you to link to specific parts of the document. To link to a specific part of a page, add a number sign (#) and an XPointer expression after the URL in the xlink:href attribute, like this: **xlink:href="http://dog.com/dogbreeds.xml#xpointer(id('Rottweiler'))".** The expression refers to the element in the target document, with the id value of "Rottweiler".

**XPointer** also allows a shorthand method for linking to an element with an **id**. You can use the value of the id directly, like this: **xlink:href="http://dog.com/dogbreeds.xml#Rottweiler".**

The following XML document contains links to more information of the dog breed for each of my dogs:

<?xml version="1.0" encoding="UTF-8"?>  
  
<mydogs xmlns:xlink="http://www.w3.org/1999/xlink">  
  
<mydog>  
  <description>  
  Anton is my favorite dog. He has won a lot of.....  
  </description>  
  <fact xlink:type="simple" xlink:href="http://dog.com/dogbreeds.xml#Rottweiler">  
  Fact about Rottweiler   
  </fact>  
</mydog>  
  
<mydog>  
  <description>  
  Pluto is the sweetest dog on earth......  
  </description>  
  <fact xlink:type="simple" xlink:href="http://dog.com/dogbreeds.xml#FCRetriever">  
  Fact about flat-coated Retriever   
  </fact>  
</mydog>  
  
</mydogs>

**3. XPath**

XPath (the XML Path language) is a language for finding information in an XML document.

XPath Path Expressions

XPath uses path expressions to select nodes or node-sets in an XML document. These path expressions look very much like the expressions you see when you work with a traditional computer file system.

Today XPath expressions can also be used in JavaScript, Java, XML Schema, PHP, Python, C and C++, and lots of other languages.

Selecting Nodes

XPath uses path expressions to select nodes in an XML document. The node is selected by following a path or steps. The most useful path expressions are listed below:

|  |  |
| --- | --- |
| **Expression** | **Description** |
| *nodename* | Selects all nodes with the name "*nodename*" |
| / | Selects from the root node |
| // | Selects nodes in the document from the current node that match the selection no matter where they are |
| . | Selects the current node |
| .. | Selects the parent of the current node |
| @ | Selects attributes |

In the table below we have listed some path expressions and the result of the expressions:

|  |  |
| --- | --- |
| **Path Expression** | **Result** |
| bookstore | Selects all nodes with the name "bookstore" |
| /bookstore | Selects the root element bookstore  **Note:** If the path starts with a slash ( / ) it always represents an absolute path to an element! |
| bookstore/book | Selects all book elements that are children of bookstore |
| //book | Selects all book elements no matter where they are in the document |
| bookstore//book | Selects all book elements that are descendant of the bookstore element, no matter where they are under the bookstore element |
| //@lang | Selects all attributes that are named lang |

Predicates

Predicates are used to find a specific node or a node that contains a specific value.

Predicates are always embedded in square brackets.

In the table below we have listed some path expressions with predicates and the result of the expressions:

|  |  |
| --- | --- |
| **Path Expression** | **Result** |
| /bookstore/book[1] | Selects the first book element that is the child of the bookstore element.  **Note:** In IE 5,6,7,8,9 first node is[0], but according to W3C, it is [1]. To solve this problem in IE, set the SelectionLanguage to XPath:  *In JavaScript: xml*.setProperty("SelectionLanguage","XPath"); |
| /bookstore/book[last()] | Selects the last book element that is the child of the bookstore element |
| /bookstore/book[last()-1] | Selects the last but one book element that is the child of the bookstore element |
| /bookstore/book[position()<3] | Selects the first two book elements that are children of the bookstore element |
| //title[@lang] | Selects all the title elements that have an attribute named lang |
| //title[@lang='en'] | Selects all the title elements that have a "lang" attribute with a value of "en" |
| /bookstore/book[price>35.00] | Selects all the book elements of the bookstore element that have a price element with a value greater than 35.00 |
| /bookstore/book[price>35.00]/title | Selects all the title elements of the book elements of the bookstore element that have a price element with a value greater than 35.00 |

Selecting Unknown Nodes

XPath wildcards can be used to select unknown XML nodes.

|  |  |
| --- | --- |
| **Wildcard** | **Description** |
| \* | Matches any element node |
| @\* | Matches any attribute node |
| node() | Matches any node of any kind |

In the table below we have listed some path expressions and the result of the expressions:

|  |  |
| --- | --- |
| **Path Expression** | **Result** |
| /bookstore/\* | Selects all the child element nodes of the bookstore element |
| //\* | Selects all elements in the document |
| //title[@\*] | Selects all title elements which have at least one attribute of any kind |

Selecting Several Paths

By using the | operator in an XPath expression you can select several paths.

In the table below we have listed some path expressions and the result of the expressions:

|  |  |
| --- | --- |
| **Path Expression** | **Result** |
| //book/title | //book/price | Selects all the title AND price elements of all book elements |
| //title | //price | Selects all the title AND price elements in the document |
| /bookstore/book/title | //price | Selects all the title elements of the book element of the bookstore element AND all the price elements in the document |

## XPath Axes

An axis defines a node-set relative to the current node.

|  |  |
| --- | --- |
| **AxisName** | **Result** |
| Ancestor | Selects all ancestors (parent, grandparent, etc.) of the current node |
| ancestor-or-self | Selects all ancestors (parent, grandparent, etc.) of the current node and the current node itself |
| Attribute | Selects all attributes of the current node |
| Child | Selects all children of the current node |
| Descendant | Selects all descendants (children, grandchildren, etc.) of the current node |
| descendant-or-self | Selects all descendants (children, grandchildren, etc.) of the current node and the current node itself |
| Following | Selects everything in the document after the closing tag of the current node |
| following-sibling | Selects all siblings after the current node |
| namespace | Selects all namespace nodes of the current node |
| parent | Selects the parent of the current node |
| preceding | Selects all nodes that appear before the current node in the document, except ancestors, attribute nodes and namespace nodes |
| preceding-sibling | Selects all siblings before the current node |
| self | Selects the current node |

## Location Path Expression

A location path can be absolute or relative.

An absolute location path starts with a slash ( / ) and a relative location path does not. In both cases the location path consists of one or more steps, each separated by a slash:

An absolute location path:  
  
/step/step/...  
  
A relative location path:  
  
step/step/...

Each step is evaluated against the nodes in the current node-set.

A step consists of:

* **an axis (defines the tree-relationship between the selected nodes and the current node)**
* **a node-test (identifies a node within an axis)**
* **zero or more predicates (to further refine the selected node-set)**

The syntax for a location step is:

axisname::nodetest[predicate]

### Examples

|  |  |
| --- | --- |
| **Example** | **Result** |
| child::book | Selects all book nodes that are children of the current node |
| attribute::lang | Selects the lang attribute of the current node |
| child::\* | Selects all element children of the current node |
| attribute::\* | Selects all attributes of the current node |
| child::text() | Selects all text node children of the current node |
| child::node() | Selects all children of the current node |
| descendant::book | Selects all book descendants of the current node |
| ancestor::book | Selects all book ancestors of the current node |
| ancestor-or-self::book | Selects all book ancestors of the current node - and the current as well if it is a book node |
| child::\*/child::price | Selects all price grandchildren of the current node |

XPath Operators

Below is a list of the operators that can be used in XPath expressions:

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| | | Computes two node-sets | //book | //cd |
| + | Addition | 6 + 4 |
| - | Subtraction | 6 - 4 |
| \* | Multiplication | 6 \* 4 |
| Div | Division | 8 div 4 |
| = | Equal | price=9.80 |
| != | Not equal | price!=9.80 |
| < | Less than | price<9.80 |
| <= | Less than or equal to | price<=9.80 |
| > | Greater than | price>9.80 |
| >= | Greater than or equal to | price>=9.80 |
| Or | or | price=9.80 or price=9.70 |
| and | and | price>9.00 and price<9.90 |
| mod | Modulus (division remainder) | 5 mod 2 |

**Example XML Document**

<?xml version="1.0" encoding="UTF-8"?>  
<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">XQuery Kick Start</title>  
  <author>James McGovern</author>  
  <author>Per Bothner</author>  
  <author>Kurt Cagle</author>  
  <author>James Linn</author>  
  <author>Vaidyanathan Nagarajan</author>  
  <year>2003</year>  
  <price>49.99</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>

|  |  |
| --- | --- |
| **XPath Expression** | **Result** |
| /bookstore/book[1] | Selects the first book element that is the child of the bookstore element |
| /bookstore/book[last()] | Selects the last book element that is the child of the bookstore element |
| /bookstore/book[last()-1] | Selects the last but one book element that is the child of the bookstore element |
| /bookstore/book[position()<3] | Selects the first two book elements that are children of the bookstore element |
| //title[@lang] | Selects all the title elements that have an attribute named lang |
| //title[@lang='en'] | Selects all the title elements that have a "lang" attribute with a value of "en" |
| /bookstore/book[price>35.00] | Selects all the book elements of the bookstore element that have a price element with a value greater than 35.00 |
| /bookstore/book[price>35.00]/title | Selects all the title elements of the book elements of the bookstore element that have a price element with a value greater than 35.00 |

**4. XSLT (Extensible Style Sheet Language Transformation)**

*The following xml document is used for both XSLT and CSS example*

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

**<?xml-stylesheet type="text/xsl" href="foreach.xsl"?>**

<!-- <?xml-stylesheet type="text/xsl" href="filterforeach.xsl"?> -->

<!-- <?xml-stylesheet type="text/xsl" href="sort.xsl"?> -->

<!-- <?xml-stylesheet type="text/xsl" href="if.xsl"?> -->

<!-- <?xml-stylesheet type="text/xsl" href="choose.xsl"?> -->

<!-- <?xml-stylesheet type="text/xsl" href="applytemplate.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>

<cd>

<title>Hide your heart</title>

<artist>Bonnie Tyler</artist>

<country>UK</country>

<company>CBS Records</company>

<price>9.90</price>

<year>1988</year>

</cd>

<cd>

<title>Greatest Hits</title>

<artist>Dolly Parton</artist>

<country>USA</country>

<company>RCA</company>

<price>9.90</price>

<year>1982</year>

</cd>

<cd>

<title>Still got the blues</title>

<artist>Gary Moore</artist>

<country>UK</country>

<company>Virgin records</company>

<price>10.20</price>

<year>1990</year>

</cd>

<cd>

<title>Eros</title>

<artist>Eros Ramazzotti</artist>

<country>EU</country>

<company>BMG</company>

<price>9.90</price>

<year>1997</year>

</cd>

<cd>

<title>One day</title>

<artist>Bee Gees</artist>

<country>UK</country>

<company>Polydor</company>

<price>10.90</price>

<year>1998</year>

</cd>

<cd>

<title>Sylvias Mother</title>

<artist>Dr.Hook</artist>

<country>UK</country>

<company>CBS</company>

<price>8.10</price>

<year>1973</year>

</cd>

<cd>

<title>Maggie May</title>

<artist>Rod Stewart</artist>

<country>UK</country>

<company>Pickwick</company>

<price>8.50</price>

<year>1990</year>

</cd>

<cd>

<title>Romanza</title>

<artist>Andrea Bocelli</artist>

<country>EU</country>

<company>Polydor</company>

<price>10.80</price>

<year>1996</year>

</cd>

<cd>

<title>When a man loves a woman</title>

<artist>Percy Sledge</artist>

<country>USA</country>

<company>Atlantic</company>

<price>8.70</price>

<year>1987</year>

</cd>

<cd>

<title>Black angel</title>

<artist>Savage Rose</artist>

<country>EU</country>

<company>Mega</company>

<price>10.90</price>

<year>1995</year>

</cd>

<cd>

<title>1999 Grammy Nominees</title>

<artist>Many</artist>

<country>USA</country>

<company>Grammy</company>

<price>10.20</price>

<year>1999</year>

</cd>

<cd>

<title>For the good times</title>

<artist>Kenny Rogers</artist>

<country>UK</country>

<company>Mucik Master</company>

<price>8.70</price>

<year>1995</year>

</cd>

<cd>

<title>Big Willie style</title>

<artist>Will Smith</artist>

<country>USA</country>

<company>Columbia</company>

<price>9.90</price>

<year>1997</year>

</cd>

<cd>

<title>Tupelo Honey</title>

<artist>Van Morrison</artist>

<country>UK</country>

<company>Polydor</company>

<price>8.20</price>

<year>1971</year>

</cd>

<cd>

<title>Soulsville</title>

<artist>Jorn Hoel</artist>

<country>Norway</country>

<company>WEA</company>

<price>7.90</price>

<year>1996</year>

</cd>

<cd>

<title>The very best of</title>

<artist>Cat Stevens</artist>

<country>UK</country>

<company>Island</company>

<price>8.90</price>

<year>1990</year>

</cd>

<cd>

<title>Stop</title>

<artist>Sam Brown</artist>

<country>UK</country>

<company>A and M</company>

<price>8.90</price>

<year>1988</year>

</cd>

<cd>

<title>Bridge of Spies</title>

<artist>T`Pau</artist>

<country>UK</country>

<company>Siren</company>

<price>7.90</price>

<year>1987</year>

</cd>

<cd>

<title>Private Dancer</title>

<artist>Tina Turner</artist>

<country>UK</country>

<company>Capitol</company>

<price>8.90</price>

<year>1983</year>

</cd>

<cd>

<title>Midt om natten</title>

<artist>Kim Larsen</artist>

<country>EU</country>

<company>Medley</company>

<price>7.80</price>

<year>1983</year>

</cd>

<cd>

<title>Pavarotti Gala Concert</title>

<artist>Luciano Pavarotti</artist>

<country>UK</country>

<company>DECCA</company>

<price>9.90</price>

<year>1991</year>

</cd>

<cd>

<title>The dock of the bay</title>

<artist>Otis Redding</artist>

<country>USA</country>

<company>Atlantic</company>

<price>7.90</price>

<year>1987</year>

</cd>

<cd>

<title>Picture book</title>

<artist>Simply Red</artist>

<country>EU</country>

<company>Elektra</company>

<price>7.20</price>

<year>1985</year>

</cd>

<cd>

<title>Red</title>

<artist>The Communards</artist>

<country>UK</country>

<company>London</company>

<price>7.80</price>

<year>1987</year>

</cd>

<cd>

<title>Unchain my heart</title>

<artist>Joe Cocker</artist>

<country>USA</country>

<company>EMI</company>

<price>8.20</price>

<year>1987</year>

</cd>

</catalog>

XSLT files should be created as a separate file (with an extention .xsl) and then applied on an xml file

# Structure of XSLT

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

**First statement ----🡪 <xsl:stylesheet version="1.0"**

**xmlns:xsl="http://www.w3.org/1999/XSL/Transform">**

**Second statement--🡪 <xsl:template match="/">**

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

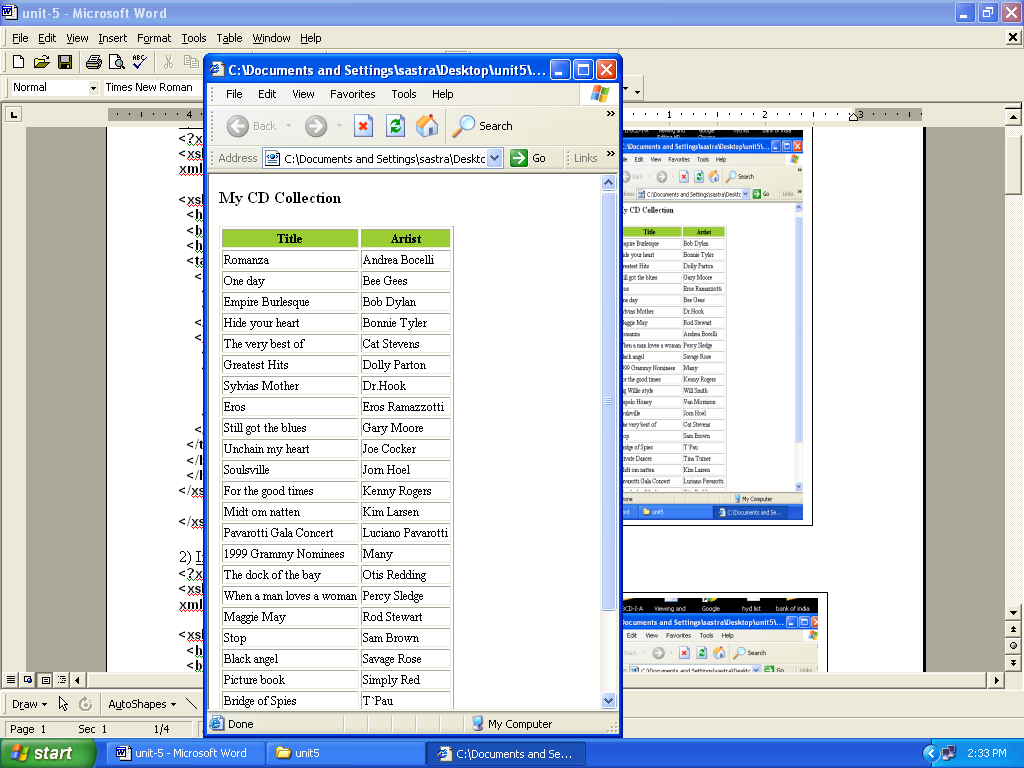
**3rd <xsl:sort**

**3rd <xsl:if**

**3rd <xsl:choice**

**3rd <xsl:apply-templates**

1) Sort.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>**

**<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>**

2) If.xsl ( conditional access in xml)

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

**<xsl:stylesheet version="1.0"**

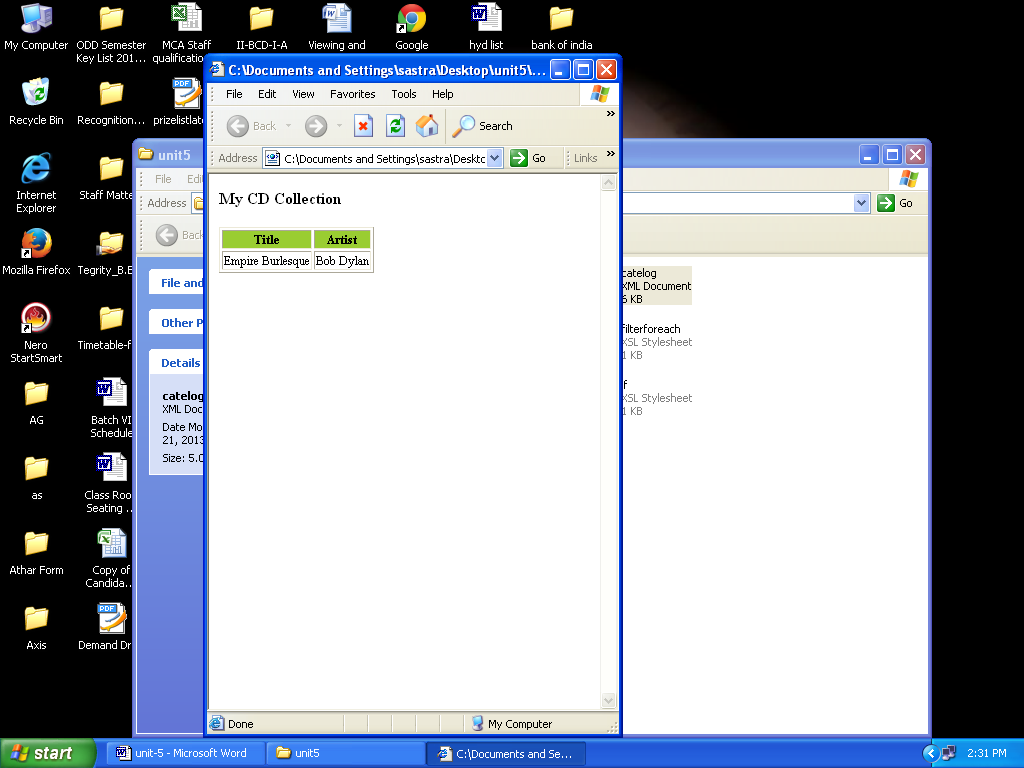
**xmlns:xsl="http://www.w3.org/1999/XSL/Transform">**

**<xsl:template match="/">**

**<html>**

**<body>**

**<h2>My CD Collection</h2>**



**<table border="1">**

**<tr bgcolor="#9acd32">**

**<th>Title</th>**

**<th>Artist</th>**

**</tr>**

**<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>**

**</table>**

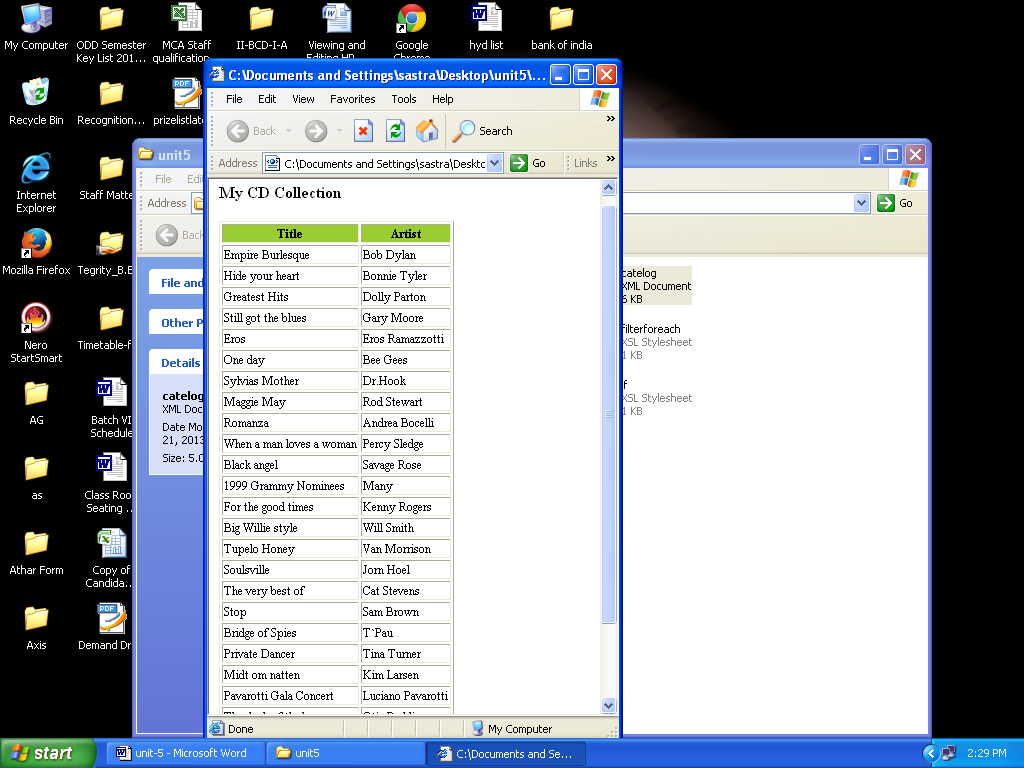
**</body>**

**</html>**

**</xsl:template>**

**</xsl:stylesheet>**

1. foreach.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>**

**<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>**

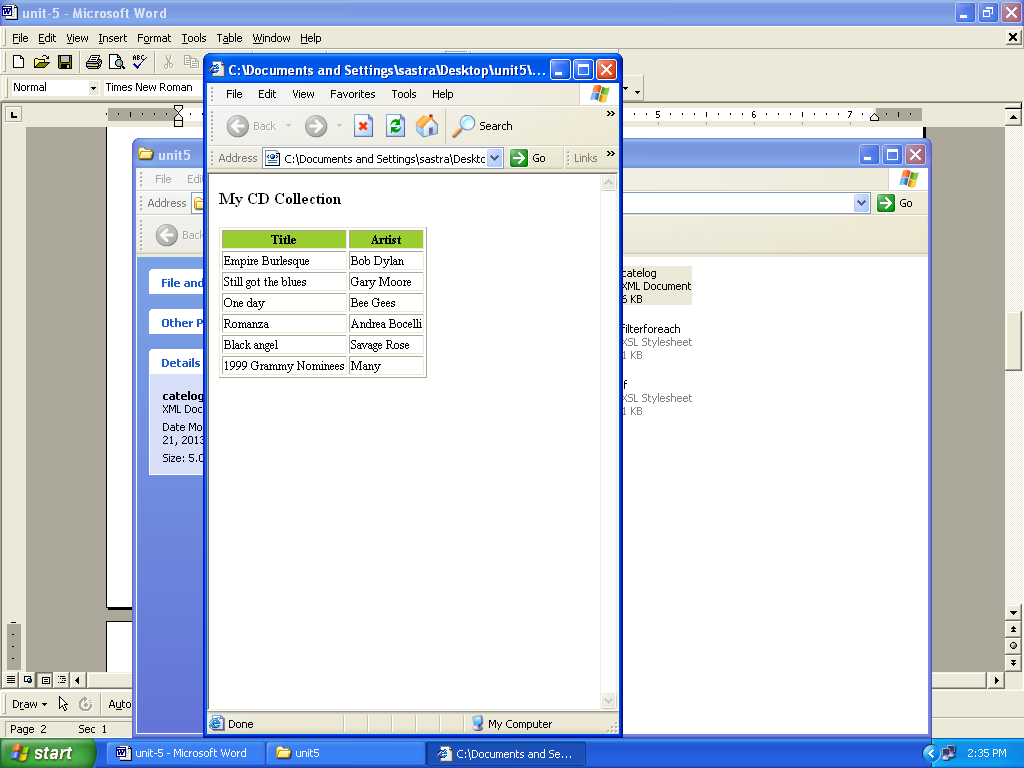
4) filterforeach.xsl

**<!--Legal filter operators are:**

**= (equal)**

**!= (not equal)**

**&lt; less than**



**&gt; greater than-->**

**<?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>**

**<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>**

**</table>**

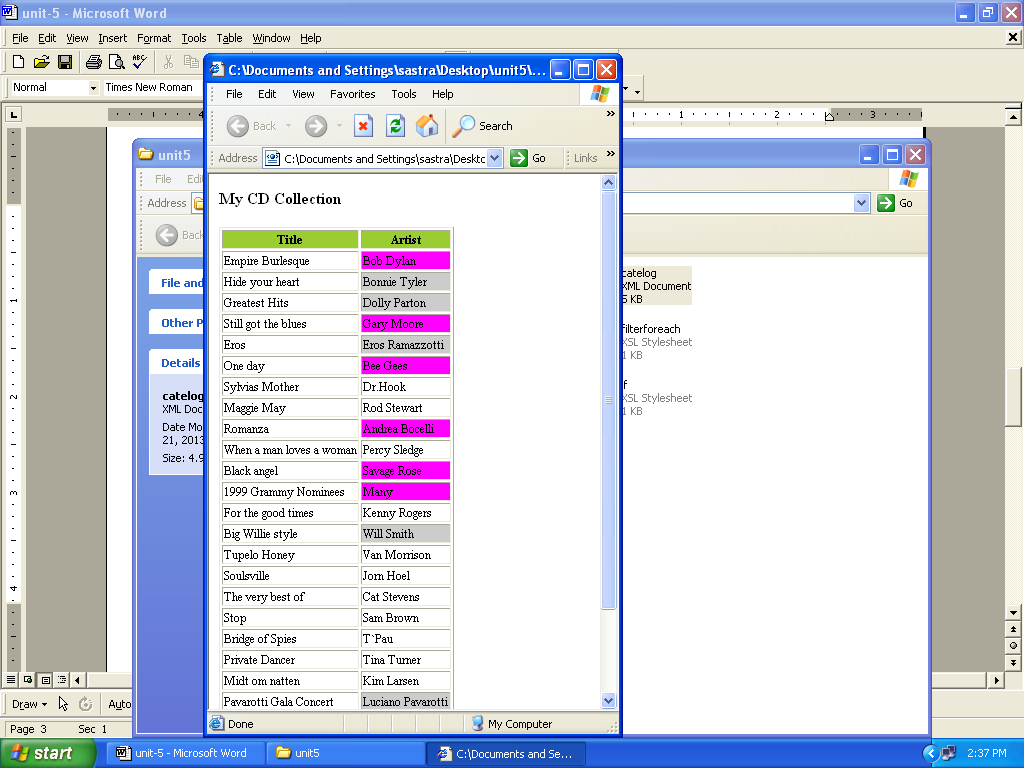
**</body>**

**</html>**

**</xsl:template>**

**</xsl:stylesheet>**

5) Choose.xsl ( equivalent to switch in programming language)



**<?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>**

**<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:when test="price &gt; 9">**

**<td bgcolor="#cccccc">**

**<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>**

**</table>**

**</body>**

**</html>**

**</xsl:template>**

**</xsl:stylesheet>**

6) Apply-templates.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>**

**<xsl:apply-templates/>**

**</body>**

**</html>**

**</xsl:template>**

**<xsl:template match="cd">**

**<p>**

**<xsl:apply-templates select="title"/>**

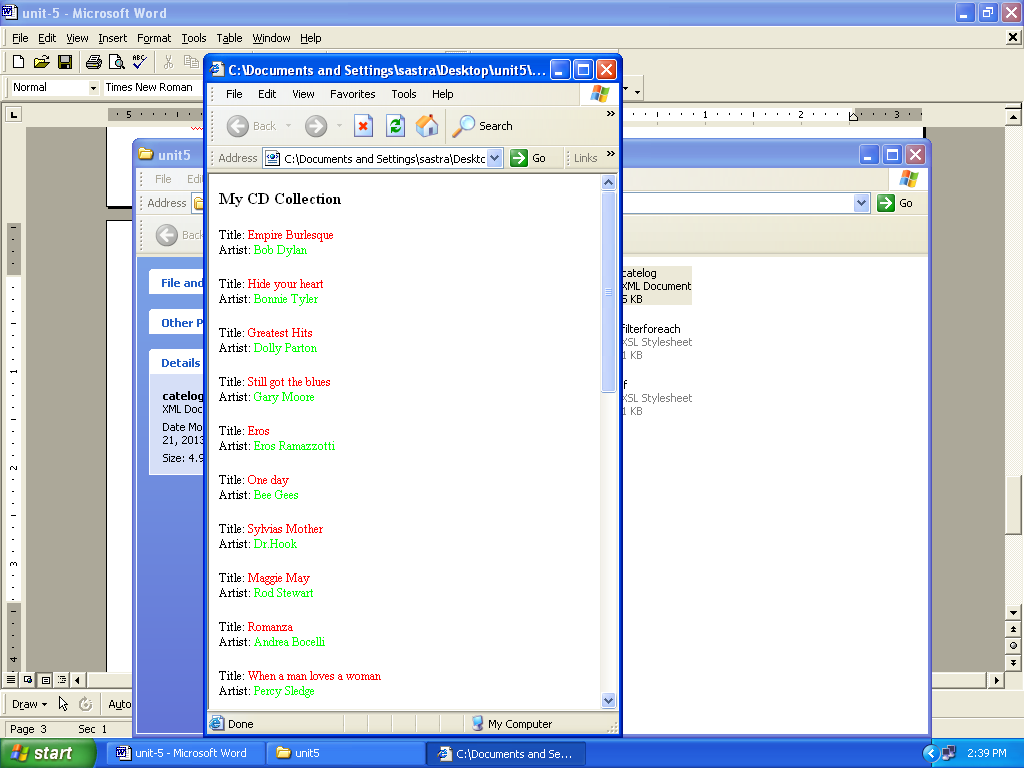
**<xsl:apply-templates select="artist"/>**

**</p>**

**</xsl:template>**

**<xsl:template match="title">**

**Title: <span style="color:#ff0000">**



**<xsl:value-of select="."/></span>**

**<br />**

**</xsl:template>**

**<xsl:template match="artist">**

**Artist: <span style="color:#00ff00">**

**<xsl:value-of select="."/></span>**

**<br />**

**</xsl:template>**

**</xsl:stylesheet>**

**5. CSS (Cascading Style Sheet)**

**Cascading Style Sheet**

*Formatting and page layout css – refer castro book*

**Structure of CSS**

**Catalog.xml**

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

**<xml-stylesheet type=”text/css” href=”e:\catalog.css”>**

<CATALOG>

<CD>

. . . . . . .

. . . . . . .

</CATALOG>

**Catalog.css**

CATALOG

{

background-color: #ffffff;

width: 100%;

}

CD

{

display: block;

margin-bottom: 30pt;

margin-left: 0;

}

TITLE

{

color: #FF0000;

font-size: 20pt;

}

ARTIST

{

color: #0000FF;

font-size: 20pt;

}

COUNTRY,PRICE,YEAR,COMPANY

{

display: block;

color: #000000;

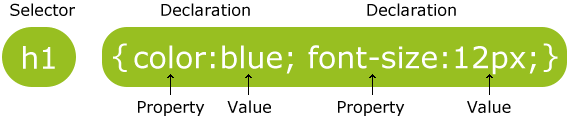
margin-left: 20pt;

}

**Cascading Style Sheet**

CSS Rule:

A CSS rule has two main parts: a selector, and one or more declarations:



The selector is normally the HTML element you want to style.

Each declaration consists of a property and a value.

The property is the style attribute you want to change. Each property has a value.

## CSS Example

A CSS declaration always ends with a semicolon, and declaration groups are surrounded by curly brackets:

p {color:red;text-align:center;}

To make the CSS more readable, you can put one declaration on each line, like this:

## Example

p  
{  
color:red;  
text-align:center;  
}

## CSS Comments

Comments are used to explain your code, and may help you when you edit the source code at a later date. Comments are ignored by browsers.

A CSS comment begins with "/\*", and ends with "\*/", like this:

/\*This is a comment\*/  
p  
{  
text-align:center;  
/\*This is another comment\*/  
color:black;  
font-family:arial;  
}

## The id and class Selectors

In addition to setting a style for a HTML element, CSS allows you to specify your own selectors called "id" and "class".

## The id Selector

The id selector is used to specify a style for a single, unique element.

The id selector uses the id attribute of the HTML element, and is defined with a "#".

The style rule below will be applied to the element with id="para1":

## Example

#para1  
{  
text-align:center;  
color:red;  
}

## The class Selector

The class selector is used to specify a style for a group of elements. Unlike the id selector, the class selector is most often used on several elements.

This allows you to set a particular style for many HTML elements with the same class.

The class selector uses the HTML class attribute, and is defined with a "."

In the example below, all HTML elements with class="center" will be center-aligned:

## Example

.center {text-align:center;}

You can also specify that only specific HTML elements should be affected by a class.

In the example below, all p elements with class="center" will be center-aligned:

## Example

p.center {text-align:center;}

## Three Ways to Insert CSS

There are three ways of inserting a style sheet:

* External style sheet
* Internal style sheet
* Inline style

## External Style Sheet

An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing one file. Each page must link to the style sheet using the <link> tag. The <link> tag goes inside the head section:

<head>  
<link rel="stylesheet" type="text/css" href="mystyle.css">  
</head>

An external style sheet can be written in any text editor. The file should not contain any html tags. Your style sheet should be saved with a **.css extension**. An example of a style sheet file is shown below:

hr {color:sienna;}  
p {margin-left:20px;}  
body {background-image:url("images/back40.gif");}

## Internal Style Sheet

An internal style sheet should be used when a single document has a unique style. You define internal styles in the head section of an HTML page, by using the <style> tag, like this:

<head>  
<style>  
hr {color:red;}  
p {margin-left:20px;}  
body {background-image:url("images/back40.gif");}  
</style>  
</head>

## Inline Styles

An inline style loses many of the advantages of style sheets by mixing content with presentation. Use this method sparingly!

To use inline styles you use the style attribute in the relevant tag. The style attribute can contain any CSS property. The example shows how to change the color and the left margin of a paragraph:

<p style="color:red;margin-left:20px">This is a paragraph.</p>

## Multiple Style Sheets

If some properties have been set for the same selector in different style sheets, the values will be inherited from the more specific style sheet.

For example, an external style sheet has these properties for the h3 selector:

h3  
{  
color:red;  
text-align:left;  
font-size:8pt;  
}

And an internal style sheet has these properties for the h3 selector:

h3  
{  
text-align:right;  
font-size:20pt;  
}

If the page with the internal style sheet also links to the external style sheet the properties for h3 will be:

color:red;  
text-align:right;  
font-size:20pt;

The color is inherited from the external style sheet and the text-alignment and the font-size is replaced by the internal style sheet.

## Background Color

The background-color property specifies the background color of an element.

The background color of a page is defined in the body selector:

## Example

body {background-color:#b0c4de;}

With CSS, a color is most often specified by:

* a HEX value - like "#ff0000"
* an RGB value - like "rgb(255,0,0)"
* a color name - like "red"

In the example below, the h1, p, and div elements have different background colors:

## Example

h1 {background-color:#6495ed;}  
p {background-color:#e0ffff;}  
div {background-color:#b0c4de;}

## Background Image

The background-image property specifies an image to use as the background of an element.

By default, the image is repeated so it covers the entire element.

The background image for a page can be set like this:

## Example

body {background-image:url('paper.gif');}

Below is an example of a bad combination of text and background image. The text is almost not readable:

## Example

body {background-image:url('bgdesert.jpg');}

|  |  |
| --- | --- |
| **Property** | **Description** |
| [background](http://www.w3schools.com/cssref/css3_pr_background.asp) | Sets all the background properties in one declaration |
| [background-attachment](http://www.w3schools.com/cssref/pr_background-attachment.asp) | Sets whether a background image is fixed or scrolls with the rest of the page |
| [background-color](http://www.w3schools.com/cssref/pr_background-color.asp) | Sets the background color of an element |
| [background-image](http://www.w3schools.com/cssref/pr_background-image.asp) | Sets the background image for an element |
| [background-position](http://www.w3schools.com/cssref/pr_background-position.asp) | Sets the starting position of a background image |
| [background-repeat](http://www.w3schools.com/cssref/pr_background-repeat.asp) | Sets how a background image will be repeated |

## Text Color

The color property is used to set the color of the text.

With CSS, a color is most often specified by:

* a HEX value - like "#ff0000"
* an RGB value - like "rgb(255,0,0)"
* a color name - like "red"

he default color for a page is defined in the body selector.

## Example

body {color:blue;}  
h1 {color:#00ff00;}  
h2 {color:rgb(255,0,0);}

## Text Alignment

The text-align property is used to set the horizontal alignment of a text.

Text can be centered, or aligned to the left or right, or justified.

When text-align is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers).

## Example

h1 {text-align:center;}  
p.date {text-align:right;}  
p.main {text-align:justify;}

## Text Decoration

The text-decoration property is used to set or remove decorations from text.

The text-decoration property is mostly used to remove underlines from links for design purposes:

## Example

a {text-decoration:none;}

It can also be used to decorate text:

## Example

h1 {text-decoration:overline;}  
h2 {text-decoration:line-through;}  
h3 {text-decoration:underline;}

## Text Transformation

The text-transform property is used to specify uppercase and lowercase letters in a text.

It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word.

## Example

p.uppercase {text-transform:uppercase;}  
p.lowercase {text-transform:lowercase;}  
p.capitalize {text-transform:capitalize;}

## Text Indentation

The text-indent property is used to specify the indentation of the first line of a text.

## Example

p {text-indent:50px;}

|  |  |
| --- | --- |
| **Property** | **Description** |
| [color](http://www.w3schools.com/cssref/pr_text_color.asp) | Sets the color of text |
| [direction](http://www.w3schools.com/cssref/pr_text_direction.asp) | Specifies the text direction/writing direction |
| [letter-spacing](http://www.w3schools.com/cssref/pr_text_letter-spacing.asp) | Increases or decreases the space between characters in a text |
| [line-height](http://www.w3schools.com/cssref/pr_dim_line-height.asp) | Sets the line height |
| [text-align](http://www.w3schools.com/cssref/pr_text_text-align.asp) | Specifies the horizontal alignment of text |
| [text-decoration](http://www.w3schools.com/cssref/pr_text_text-decoration.asp) | Specifies the decoration added to text |
| [text-indent](http://www.w3schools.com/cssref/pr_text_text-indent.asp) | Specifies the indentation of the first line in a text-block |
| [text-shadow](http://www.w3schools.com/cssref/css3_pr_text-shadow.asp) | Specifies the shadow effect added to text |
| [text-transform](http://www.w3schools.com/cssref/pr_text_text-transform.asp) | Controls the capitalization of text |
| unicode-bidi |  |
| [vertical-align](http://www.w3schools.com/cssref/pr_pos_vertical-align.asp) | Sets the vertical alignment of an element |
| [white-space](http://www.w3schools.com/cssref/pr_text_white-space.asp) | Specifies how white-space inside an element is handled |
| [word-spacing](http://www.w3schools.com/cssref/pr_text_word-spacing.asp) | Increases or decreases the space between words in a text |

## All CSS Font Properties

|  |  |
| --- | --- |
| **Property** | **Description** |
| [font](http://www.w3schools.com/cssref/pr_font_font.asp) | Sets all the font properties in one declaration |
| [font-family](http://www.w3schools.com/cssref/pr_font_font-family.asp) | Specifies the font family for text |
| [font-size](http://www.w3schools.com/cssref/pr_font_font-size.asp) | Specifies the font size of text |
| [font-style](http://www.w3schools.com/cssref/pr_font_font-style.asp) | Specifies the font style for text |
| [font-variant](http://www.w3schools.com/cssref/pr_font_font-variant.asp) | Specifies whether or not a text should be displayed in a small-caps font |
| [font-weight](http://www.w3schools.com/cssref/pr_font_weight.asp) | Specifies the weight of a font |

**Usage:**

p{font-family:"Times New Roman", Times, serif;}

p.normal {font-style:normal;}  
p.italic {font-style:italic;}  
p.oblique {font-style:oblique;}

h1 {font-size:40px;}

## Styling Links

Links can be styled with any CSS property (e.g. color, font-family, background, etc.).

In addition, links can be styled differently depending on what **state** they are in.

The four links states are:

* a:link - a normal, unvisited link
* a:visited - a link the user has visited
* a:hover - a link when the user mouses over it
* a:active - a link the moment it is clicked

## Example

a:link {color:#FF0000;}      /\* unvisited link \*/  
a:visited {color:#00FF00;}  /\* visited link \*/  
a:hover {color:#FF00FF;}  /\* mouse over link \*/  
a:active {color:#0000FF;}  /\* selected link \*/

## List

In HTML, there are two types of lists:

* unordered lists - the list items are marked with bullets
* ordered lists - the list items are marked with numbers or letters

With CSS, lists can be styled further, and images can be used as the list item marker.

## Different List Item Markers

The type of list item marker is specified with the list-style-type property:

## Example

ul.a {list-style-type: circle;}  
ul.b {list-style-type: square;}  
  
ol.c {list-style-type: upper-roman;}  
ol.d {list-style-type: lower-alpha;}

## All CSS List Properties

|  |  |
| --- | --- |
| **Property** | **Description** |
| [list-style](http://www.w3schools.com/cssref/pr_list-style.asp) | Sets all the properties for a list in one declaration |
| [list-style-image](http://www.w3schools.com/cssref/pr_list-style-image.asp) | Specifies an image as the list-item marker |
| [list-style-position](http://www.w3schools.com/cssref/pr_list-style-position.asp) | Specifies if the list-item markers should appear inside or outside the content flow |
| [list-style-type](http://www.w3schools.com/cssref/pr_list-style-type.asp) | Specifies the type of list-item marker |

## Table Borders

To specify table borders in CSS, use the border property.

The example below specifies a black border for table, th, and td elements:

## Example

table, th, td  
{  
border: 1px solid black;  
}

## Table Width and Height

Width and height of a table is defined by the width and height properties.

The example below sets the width of the table to 100%, and the height of the th elements to 50px:

## Example

table   
{ width:100%; }

th  
{height:50px;}

## Example

table, td, th  
{ border:1px solid green;}  
th  
{  
background-color:green;  
color:white;  
}

## All CSS Border Properties

|  |  |
| --- | --- |
| **Property** | **Description** |
| [border](http://www.w3schools.com/cssref/pr_border.asp) | Sets all the border properties in one declaration |
| [border-bottom](http://www.w3schools.com/cssref/pr_border-bottom.asp) | Sets all the bottom border properties in one declaration |
| [border-bottom-color](http://www.w3schools.com/cssref/pr_border-bottom_color.asp) | Sets the color of the bottom border |
| [border-bottom-style](http://www.w3schools.com/cssref/pr_border-bottom_style.asp) | Sets the style of the bottom border |
| [border-bottom-width](http://www.w3schools.com/cssref/pr_border-bottom_width.asp) | Sets the width of the bottom border |
| [border-color](http://www.w3schools.com/cssref/pr_border-color.asp) | Sets the color of the four borders |
| [border-left](http://www.w3schools.com/cssref/pr_border-left.asp) | Sets all the left border properties in one declaration |
| [border-left-color](http://www.w3schools.com/cssref/pr_border-left_color.asp) | Sets the color of the left border |
| [border-left-style](http://www.w3schools.com/cssref/pr_border-left_style.asp) | Sets the style of the left border |
| [border-left-width](http://www.w3schools.com/cssref/pr_border-left_width.asp) | Sets the width of the left border |
| [border-right](http://www.w3schools.com/cssref/pr_border-right.asp) | Sets all the right border properties in one declaration |
| [border-right-color](http://www.w3schools.com/cssref/pr_border-right_color.asp) | Sets the color of the right border |
| [border-right-style](http://www.w3schools.com/cssref/pr_border-right_style.asp) | Sets the style of the right border |
| [border-right-width](http://www.w3schools.com/cssref/pr_border-right_width.asp) | Sets the width of the right border |
| [border-style](http://www.w3schools.com/cssref/pr_border-style.asp) | Sets the style of the four borders |
| [border-top](http://www.w3schools.com/cssref/pr_border-top.asp) | Sets all the top border properties in one declaration |
| [border-top-color](http://www.w3schools.com/cssref/pr_border-top_color.asp) | Sets the color of the top border |
| [border-top-style](http://www.w3schools.com/cssref/pr_border-top_style.asp) | Sets the style of the top border |
| [border-top-width](http://www.w3schools.com/cssref/pr_border-top_width.asp) | Sets the width of the top border |
| [border-width](http://www.w3schools.com/cssref/pr_border-width.asp) | Sets the width of the four borders |

**Usage:**

p  
{  
border-top-style:dotted;  
border-right-style:solid;  
border-bottom-style:dotted;  
border-left-style:solid;  
}

## All CSS Margin Properties

|  |  |
| --- | --- |
| **Property** | **Description** |
| [margin](http://www.w3schools.com/cssref/pr_margin.asp) | A shorthand property for setting the margin properties in one declaration |
| [margin-bottom](http://www.w3schools.com/cssref/pr_margin-bottom.asp) | Sets the bottom margin of an element |
| [margin-left](http://www.w3schools.com/cssref/pr_margin-left.asp) | Sets the left margin of an element |
| [margin-right](http://www.w3schools.com/cssref/pr_margin-right.asp) | Sets the right margin of an element |
| [margin-top](http://www.w3schools.com/cssref/pr_margin-top.asp) | Sets the top margin of an element |

## Example

margin-top:100px;  
margin-bottom:100px;  
margin-right:50px;  
margin-left:50px;

**The following is an example (catalog.css) file created for catalog.xml**

**Filename:** catalog.css

**CATALOG {**

**background-color: #00ffff;**

**}**

**CD {**

**display: block;**

**margin-bottom: 30pt;**

**margin-left: 0;**

**}**

**TITLE {**

**display: block;**

**color: #ff0000;**

**font-size: 20pt;**

**text-transform:lowercase;**

**text-indent:50px;**

**}**

**ARTIST {**

**display: block;**

**color: #0000ff;**

**font-size: 20pt;**

**}**

**COUNTRY, PRICE, YEAR, COMPANY {**

**display: block;**

**color: #000000;**

**margin-left: 20pt;**

**}**