

XSL}

Extensible Style sheet Language

XSL

- { What is XSL
- { Anatomy of a XSLT Document
- { Connects XML to XSL
- { Extracting Content
- { More about XSLT

What is XSL



XSL

Extensible Style sheet Language. Despite its name it doesn't style xml, but it does format the document.

XSL... So What does it do?

XSL will show selected content in a XHTML context versus displaying XML as raw data.



Anatomy of XSLT Document

Anatomy of XSL

XSL is really just an xml file saved out as a .xsl referencing a different set of rules.

Open Dreamwaver.
File -- > new --> XSLT(Entire page)

Anatomy of XSL

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE xsl:stylesheet [

    <!ENTITY nbsp    " ">
    <!ENTITY copy    "©">
    <!ENTITY reg      "®">
    <!ENTITY trade    "™">
    <!ENTITY mdash    "—">
    <!ENTITY ldquo    "“">
    <!ENTITY rdquo    "”">
    <!ENTITY pound    "£">
    <!ENTITY yen      "¥">
    <!ENTITY euro     "€">

]>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="UTF-8" doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN" doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"/>

<xsl:template match="/">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
<title>Untitled Document</title>
</head>

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

Anatomy of XSL

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE xsl:stylesheet [
```

```
    <!ENTITY nbsp    " ">
    <!ENTITY copy    "©">
    <!ENTITY reg      "®">
    <!ENTITY trade    "™">
    <!ENTITY mdash    "—">
    <!ENTITY ldquo    "“">
    <!ENTITY rdquo    "”">
    <!ENTITY pound    "£">
    <!ENTITY yen      "¥">
    <!ENTITY euro     "€">
```

```
]>
```

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

```
<xsl:output method="html" encoding="UTF-8" doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN" doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"/>
```

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

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
```

```
<title>Untitled Document</title>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

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

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

Anatomy of XSL

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE xsl:stylesheet [

    <!ENTITY nbsp      "&#160;">
    <!ENTITY copy      "&#169;">
    <!ENTITY reg        "&#174;">
    <!ENTITY trade      "&#8482;">
    <!ENTITY mdash      "&#8212;">
    <!ENTITY ldquo      "&#8220;">
    <!ENTITY rdquo      "&#8221;">
    <!ENTITY pound       "&#163;">
    <!ENTITY yen         "&#165;">
    <!ENTITY euro        "&#8364;">

]>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/
Transform">
<xsl:output method="html" encoding="UTF-8" doctype-public="-//W3C//DTD XHTML
1.0 Transitional//EN" doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd"/>
```

**This is preparing the
xsl documnt to
follow certian rules.**

Anatomy of XSL

```
<xsl:template match="/">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html">
<title>Untitled Document</title>
</head>

<body>
</body>
</html>
```

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

**This is closes
the document.**

Anatomy of XSL

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE xsl:stylesheet [

    <!ENTITY nbsp    "&#160;";>
    <!ENTITY copy    "&#169;";>
    <!ENTITY reg      "&#174;";>
    <!ENTITY trade    "&#8482;";>
    <!ENTITY mdash    "&#8212;";>
    <!ENTITY ldquo    "&#8220;";>
    <!ENTITY rdquo    "&#8221;";>
    <!ENTITY pound    "&#163;";>
    <!ENTITY yen      "&#165;";>
    <!ENTITY euro     "&#8364;";>

]>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="UTF-8" doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN" doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"/>

<xsl:template match="/">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
<title>Untitled Document</title>
</head>

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

Anatomy of XSL

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

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html;  
charset=UTF-8" />
```

```
<title>Untitled Document</title>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

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

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

Look familiar?

Anatomy of XSL

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

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html;  
charset=UTF-8" />
```

```
<title>Untitled Document</title>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

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

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

This is XHTML.

Anatomy of XSL

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

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html;  
charset=UTF-8" />
```

```
<title>Untitled Document</title>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

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

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

**All of our
content will go
inside the
body element.**

All standard XHTML rules apply.



Connect XML
to XSL

Connect XML to XSL

The XML document must reference the XSL document in order for the formatting to take place.

Connect XML to XSL

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="songsStyle.xsl" ?>
<songs>
  <song>
    <name>Smells Like Teen Spirit</name>
    <album>Nevermind</album>
    <artist>Nirvana</artist>
    <year>1991</year>
  </song>
</songs>
```

Connect XML to XSL

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="songsStyle.xsl" ?>
<songs>
  <song>
    <name>Smells Like Teen Spirit</name>
    <album>Nevermind</album>
    <artist>Nirvana</artist>
    <year>1991</year>
  </song>
</songs>
```

Connect XML to XSL

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="songsStyle.xsl" ?>
<songs>
  <song>
    <name>Smells Like Teen Spirit</name>
    <album>Nevermind</album>
    <artist>Nirvana</artist>
    <year>1991</year>
  </song>
</songs>
```

**The href points to
the location of
the .xsl.**

Extracting Content



View the XML

- { Open XML file in a browser.
- { *You should see a blank screen.*

We are going add simple XHTML.

Basic Tags

```
<xsl:template match="/">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8" />
<title>Untitled Document</title>
</head>

<body>
  <h1>My list</h1>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Basic Tags

```
<xsl:template match="/">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8" />
<title>Untitled Document</title>
</head>

<body>
  <h1>My list</h1>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

**Note the
<h1> element**

View the XML

- { Open XML file in a browser.
- { *You should see My List in Big and Bold*

**We are extracting *ALL* of
the content from the xml document.**

Basic Tags

```
<xsl:template match="/">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8" />
<title>Untitled Document</title>
</head>

<body>
  <h1>My list</h1>
  <xsl:apply-templates select="songs" />
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Basic Tags

```
<xsl:template match="/">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Untitled Document</title>
</head>

<body>
  <h1>My list</h1>
  <xsl:apply-templates select="songs" />
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

**Note the
<xsl> specific
element. It's used to
show all data.**

Basic Tags

```
<xsl:template match="/">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html" charset=UTF-8"/>
<title>Untitled Document</title>
</head>

<body>
  <h1>My list</h1>
  <xsl:apply-templates select="songs" />
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

“songs” refers to the root element in the xml document.

XML Document

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="songsStyle.xsl" ?>
<songs>
  <song>
    <name>Smells Like Teen Spirit</name>
    <album>Nevermind</album>
    <artist>Nirvana</artist>
    <year>1991</year>
  </song>
</songs>
```

XML Document

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="songsStyle.xsl" ?>
<songs>
  <song>
    <name>Smells Like Teen Spirit</name>
    <album>Nevermind</album>
    <artist>Nirvana</artist>
    <year>1991</year>
  </song>
</songs>
```

“songs” is the root element.

View the XML

- { Open XML file in a browser.
- { *You should see My List in Big and Bold*
- { *You should see all of the content from the xml document*

We are extracting *SPECIFIC* content from the xml document.

Basic Tags

```
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8" />
<title>Untitled Document</title>
</head>

<body>
  <h1>My list</h1>
  <xsl:for-each select="songs/song">
    <h2><xsl:value-of select="artist" /></h2>
    <h6><xsl:value-of select="name" /></h6>
    <xsl:value-of select="year" />
  </xsl:for-each>
</body>
</html>
</xsl:template>
```

Basic Tags

```
<meta http-equiv="Content-  
charset=UTF-8" />  
<title>Untitled Document<  
</head>
```

```
<body>  
  <h1>My list</h1>  
  <xsl:for-each select="songs/song">  
    <h2><xsl:value-of select="artist" /></h2>  
    <h6><xsl:value-of select="name" /></h6>  
    <xsl:value-of select="year" />  
  </xsl:for-each>  
</body>  
</html>  
</xsl:template>
```

**Note the
<xsl> specific
elements. They are
used to show
specific data.**

Basic Tags

```
<meta http-equiv="Content-  
charset=UTF-8" />  
<title>Untitled Document<  
</head>
```

```
<body>  
  <h1>My list</h1>  
  <xsl:for-each select="songs/song">  
    <h2><xsl:value-of select="artist" /></h2>  
    <h6><xsl:value-of select="name" /></h6>  
    <xsl:value-of select="year" />  
  </xsl:for-each>  
</body>  
</html>  
</xsl:template>
```

**It's similar to the
concept of
running a query.**

Basic Tags

```
<meta http-equiv="Content-  
charset=UTF-8" />  
<title>Untitled Document</  
</head>
```

```
<body>
```

```
  <h1>My list</h1>
```

```
  <xsl:for-each select="songs/song">
```

```
    <h2><xsl:value-of select="artist" /></h2>
```

```
    <h6><xsl:value-of select="name" /></h6>
```

```
    <xsl:value-of select="year" />
```

```
  </xsl:for-each>
```

```
</body>
```

```
</html>
```

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

**for-each loops
through all of
the song elements
inside of the root
songs element.**

Basic Tags

```
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8" />
<title>Untitled Document</title>
</head>

<body>
  <h1>My list</h1>
  <xsl:for-each select="songs/song">
    <h2><xsl:value-of select="artist" /></h2>
    <h6><xsl:value-of select="name" /></h6>
    <xsl:value-of select="year" />
  </xsl:for-each>
</body>
</html>
</xsl:template>
```

XML Document

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="songsStyle.xsl" ?>
<songs>
  <song>
    <name>Smells Like Teen Spirit</name>
    <album>Nevermind</album>
    <artist>Nirvana</artist>
    <year>1991</year>
  </song>
</songs>
```

XML Document

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="songsStyle.xsl" ?>
<songs>
  <song>
    <name>Smells Like Teen Spirit</name>
    <album>Nevermind</album>
    <artist>Nirvana</artist>
    <year>1991</year>
  </song>
</songs>
```

**song is inside
of songs.**

Basic Tags

```
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8" />
<title>Untitled Document</title>
</head>

<body>
  <h1>My list</h1>
  <xsl:for-each select="songs/song">
    <h2><xsl:value-of select="artist" /></h2>
    <h6><xsl:value-of select="name" /></h6>
    <xsl:value-of select="year" />
  </xsl:for-each>
</body>
</html>
</xsl:template>
```

Basic Tags

```
<meta http-equiv="Content-  
charset=UTF-8" />  
<title>Untitled Document<  
</head>
```

```
<body>  
  <h1>My list</h1>  
  <xsl:for-each select="songs/song">  
    <h2><xsl:value-of select="artist" /></h2>  
    <h6><xsl:value-of select="name" /></h6>  
    <xsl:value-of select="year" />  
  </xsl:for-each>  
</body>  
</html>  
</xsl:template>
```

**value-of displays the
content for the
selected element.**

Basic Tags

```
<meta http-equiv="Content-  
charset=UTF-8" />  
<title>Untitled Document<  
</head>
```

```
<body>  
  <h1>My list</h1>  
  <xsl:for-each select="songs/song">  
    <h2><xsl:value-of select="artist" /></h2>  
    <h6><xsl:value-of select="name" /></h6>  
    <xsl:value-of select="year" />  
  </xsl:for-each>  
</body>  
</html>  
</xsl:template>
```

**In this case artist is
the selected
element.**

XML Document

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="songsStyle.xsl" ?>
<songs>
  <song>
    <name>Smells Like Teen Spirit</name>
    <album>Nevermind</album>
    <artist>Nirvana</artist>
    <year>1991</year>
  </song>
</songs>
```

XML Document

```
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet type="text/xsl" href="style.xsl" ?>
<songs>
  <song>
    <name>Smells Like Teen Spirit</name>
    <album>Nevermind</album>
    <artist>Nirvana</artist>
    <year>1991</year>
  </song>
</songs>
```

**This is the reference
in the XSL.**

View the XML

- { Open XML file in a browser.
- { *You should see My List in Big and Bold*
- { *You should see just the content that you have specified*



More on XSL

More on XSL

<http://www.w3schools.com/xsl/>

XSL: Summary

- { What is XSL
- { Anatomy of a XSLT Document
- { Connects XML to XSL
- { Extracting Content
- { More about XSLT