

How to turn XML into HTML

Table of Contents

- Why XML? What is XSLT? 1
 - XML..... 1
 - XSLT 1
 - Displaying XML with XSLT 2
- Simple XSLT Exercise..... 2
 - 1. Start with a Raw XML Document..... 2
 - 2. Create an XSL Style Sheet (the XSLT code) 3
 - 3. Link the XSL Style Sheet to the XML Document 3
- Next Steps - XSLT on the Server 4

Why XML? What is XSLT?

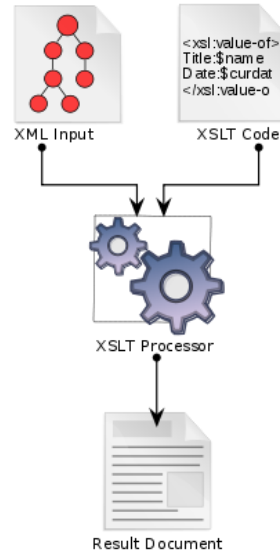
XML is often used to separate data from presentation (HTML/CSS). XML does not carry any information about how to be displayed. The same XML data can be used in many different presentation scenarios. Because of this, with XML, there is a full separation between data and presentation.

XML

Raw XML files can be viewed in all major browsers but XML files will not be displayed as HTML pages.

XSLT

You need XSLT (eXtensible Stylesheet Language Transformations) because XML cannot be displayed as HTML pages. XSLT will allow you to transform an XML document (the input) into HTML and XHTML (the output). XSLT does this by transforming or mapping each XML element into an (X)HTML element.



With XSLT, in the form of *.xsl files, you can add/remove elements and attributes to or from the (X)HTML output file. You can also rearrange and sort elements, perform tests and make decisions about which elements to hide and display, and a lot more.

Displaying XML with XSLT

XSLT is more sophisticated than CSS. With XSLT you can add/remove elements and attributes to or from the output file. You can also rearrange and sort elements, perform tests and make decisions about which elements to hide and display, and a lot more.

Simple XSLT Exercise

In the transformation process, XSLT uses XPath to define parts of the source document that should match one or more predefined templates. When a match is found, XSLT will transform the matching part of the source document into the result document.

1. Start with a Raw XML Document

This is the *Hello World* of XSLT. A simple example named *doc.xml* that runs in all processors.

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <doctitle>A document title</doctitle>
  <para>My very first transformed paragraph</para>
</doc>
```

Viewing XML Files in IE, Chrome, Firefox, Safari, and Opera:

Open the XML file - The XML document will be displayed with color-coded root and child elements

(except in Safari). Often, there is a plus (+) or minus sign (-) to the left of the elements that can be clicked to expand or collapse the element structure.

Tip: To view the raw XML source, right-click in XML file and select "View Source"!

2. Create an XSL Style Sheet (the XSLT code)

Then you create an XSL Style Sheet ("hello.xsl") with a transformation template

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:template match="/doc">
    <html>
      <head>
        <title>Test Document</title>
      </head>
      <body>
        <xsl:apply-templates/>
        <i>Some additional content from the stylesheet</i>
      </body>
    </html>
  </xsl:template>

  <xsl:template match="doctitle">
    <h1><xsl:apply-templates/></h1>
  </xsl:template>

  <xsl:template match="para">
    <p><xsl:apply-templates/></p>
  </xsl:template>

</xsl:stylesheet>
```

With one exception this looks like any other XML document. The namespace `xsl:version="1.0"` `xmlns:xsl="http://www.w3.org/1999/XSL/Transform"` tacked onto the outer element of the document lets the processor know that this is a stylesheet document. The impact of this is that the processor looks for elements in that namespace, i.e. which look like `<xsl:....>` and does something special with them. Since this document has three rules, one for the doc, the doctitle, and the para element respectively, those basic rules are used. A rule for processors is that anything which is not in that special namespace is passed straight through to the output, which in this case are all the HTML tags and their text content!

3. Link the XSL Style Sheet to the XML Document

Add the XSL style sheet reference (bolded line below) to your XML document ("doc.xml")

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="hello.xsl"?>
```

```
<doc>
  <head>A document title</head>
  <para>My very first transformed paragraph</para>
</doc>
```

If you have an XSLT compliant browser, the XML content will nicely **transform** into XHTML when opening the file.

Next Steps - XSLT on the Server

In the previous section you saw how XSLT can be used to transform a document from XML to XHTML in the browser.

To make XML data available in a business environment and compatible for all kinds of browsers, however, you will need to transform the XML document on the server and send the output back to the browser as XHTML.

That's one advantage of XSLT. One of the design goals for XSLT was to make it possible to transform data from one format to another on a server, returning readable data to all kinds of browsers.

If you run an XML document, i.e. the sample document *doc.xml* from above, through an XSLT processor you will be able to create HTML output using a simple stylesheet. For example, using [James Clarks XT](http://www.jclark.com/xml/xt-old.html) (<http://www.jclark.com/xml/xt-old.html>) processor from the command line, using the stylesheet *hello.xsl* and the XML document *doc.xml* from above, and you want the output to be *output.html*, then the command

```
xt doc.xml hello.xsl output.html
```

produces output which has both literal stuff from the stylesheet and input from the source document.

Processing PDQ XML Data

The document *PDQ-Data_high-level-process-flow.jpg* gives an overview of the necessary steps if your goal is to present the PDQ XML data on your own website. The information above gives details on how to handle the steps (c) and (d):

- a) You access the XML data from the SFTP server using your FTP client
- b) You extract and store the files from the downloaded *.tar.gz files
- c) You prepare the XSL style sheet(s) allowing you to process the XML file into HTML documents
- d) You process the files using your XSLT processor
- e) You present the HTML documents on your website.

Resources:

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