



FREE eBook

LEARNING

xslt

Free unaffiliated eBook created from
Stack Overflow contributors.

#xslt

Table of Contents

About	1
Chapter 1: Getting started with xslt	2
Remarks.....	2
Versions.....	2
Examples.....	2
Installation or Setup.....	2
Simple XSLT example.....	3
Chapter 2: Variables in xslt	7
Examples.....	7
Xslt Variables.....	7
Credits	8

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [xslt](#)

It is an unofficial and free xslt ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official xslt.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with xslt

Remarks

XSLT (XSL Transformations, Extensible Stylesheet Language Transformations) is a programming language that is based on XML and whose purpose is to manipulate and transform XML. It is a functional programming language and a W3C standard. Using XSLT you can transform one or more XML documents into a result that could have variety of output formats (XML, HTML, text).

To understand XSLT -- it is best to have a good working knowledge of:

1. XML
2. XPath

Without these two underlying concepts you would have difficulty learning XSLT. Firstly, because an XSLT document is an XML document. And secondly, XSLT documents make use of XPath to query the XML document being transformed.

Recommended reading: <https://en.wikipedia.org/wiki/XSLT>

Versions

Version	Specification	Status	Release Date
XSLT 1.0	https://www.w3.org/TR/xslt	Recommendation	1998-08-18
XSLT 2.0	https://www.w3.org/TR/xslt20	Recommendation	2007-01-23
XSLT 3.0	https://www.w3.org/TR/xslt-30	Candidate Recommendation	2015-11-19

Examples

Installation or Setup

XSLT is a special-purpose programming language; it is widely used for transforming XML documents either into a different XML format, into HTML, or into text-based formats.

There are two main versions of XSLT in use: XSLT 1.0 and XSLT 2.0. XSLT 1.0 is more widely implemented but has many restrictions and limitations compared with XSLT 2.0; you will need to decide which version to use. If an XSLT 2.0 processor is available for your chosen environment, then this is almost always the better choice.

(XSLT 1.0 came out in November 1999, and many implementations appeared within a year or two, both from mainstream vendors such as Microsoft, IBM, Sun, and Oracle, and from individual enthusiasts working in their spare time. XSLT 2.0 came out in January 2007, and many of the

original 1.0 processors were never upgraded, because their developers had lost interest. The most widely used XSLT 2.0 processor is Saxon, but it does not have the field to itself: other free-standing products include RaptorXML (from Altova), XmlPrime (from CBCL) and Exselt, and 2.0 processors are also available embedded in IBM's WebLogic, in MarkLogic's XML database server, and in Intel's XML Accelerator.)

The specification of XSLT 3.0 is technically complete (Proposed Recommendation in April 2017) but is best regarded as bleeding-edge for the time being: only consider it if you have a pressing need for its new features, such as streaming, packages, JSON support, or try/catch. There are three known implementations: Saxon, Exselt, and RaptorXML.

To get started with XSLT you have several options:

- Use an online XSLT tool. There are several available (search for "online XSLT tool"), a very reliable online IDE is <http://xsltransform.net/>. This is a good way of getting an initial feel for the language, but you won't want to carry on this way once you are doing real development.
- Use the XSLT engine built into every browser. As with online tools, these have the advantage that you don't need to install anything; but the browsers only support XSLT 1.0, they only support XML-to-HTML conversion, and they have very weak debugging support. A very recent -- but still experimental -- development is Saxon-JS, which allows execution of XSLT 2.0 (and parts of XSLT 3.0) in the browser.
- Install an XSLT processor (such as Saxon or xsltproc). Most of these products can be invoked using the operating system command line, or using an API for particular programming languages such as Java, C, C#, or Python.
- Install an XML development environment such as Altova XML Spy, oXygen from SyncroSoft, or Stylus Studio. Although this is a more expensive option, it gives much richer development support and debugging capability.

Whichever option you choose, you should first decide whether you want to use XSLT 1.0 or XSLT 2.0.

Simple XSLT example

Here's a simple example that uses XSLT to convert data in an XML file into a table in an HTML file. You can use it to experiment with simple XSLT transforms.

Prerequisite: Install a Java Runtime Environment and add the location of the JRE to your PATH variable. (On Windows, most installers will add Java to your path for you.) If this works, you should be able to open a command-line window and run the command `java -version` and get a printout of info about your JRE.

1. Download the Saxon-HE XSLT processor for Java here: saxon.sourceforge.net and unzip it to anywhere on your computer.
2. In a text editor, create a file named `pets.xml` with the following code:

```
<pets>
```

```

<petType name="Dogs">
  <pet id="123" name="Sparky" vaccineStatus="vaccinated" healthStatus="healthy"/>
  <pet id="234" name="Sadie" vaccineStatus="unvaccinated" healthStatus="sick"/>
  <pet id="345" name="Herman" vaccineStatus="unvaccinated" healthStatus="unknown"/>
</petType>
<petType name="Cats">
  <pet id="456" name="Cleo" vaccineStatus="vaccinated" healthStatus="healthy"/>
  <pet id="567" name="Janet" vaccineStatus="unvaccinated" healthStatus="healthy"/>
  <pet id="678" name="Furry" vaccineStatus="vaccinated" healthStatus="sick"/>
</petType>
</pets>

```

3. In a text editor, create a file named `petTransform.xsl` with the following code:

```

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="2.0">

  <!-- handle the root XML element -->
  <xsl:template match="/">
    <html><head>
      <title>Pets that are available for adoption</title>
    </head>
    <body>
      <xsl:apply-templates/>
    </body>
  </html>
</xsl:template>

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

  <xsl:template match="petType">
    <h2><xsl:value-of select="@name"/></h2>
    <table id="{@name}">
      <tr>
        <th colname="id">ID</th>
        <th colname="name">Name</th>
        <th colname="vaccinated">Vaccine status</th>
        <th colname="health">Health status</th>
      </tr>
      <tbody>
        <!-- add a row for each pet in this category -->
        <xsl:for-each select="pet">
          <tr>
            <td colname="id"><xsl:value-of select="@id"/></td>
            <td colname="name"><xsl:value-of select="@name"/></td>
            <td colname="vaccinated"><xsl:value-of select="@vaccineStatus"/></td>
            <td colname="health"><xsl:value-of select="@healthStatus"/></td>
          </tr>
        </xsl:for-each>
      </tbody>
    </table>
  </xsl:template>

  <!-- ignore the content of other tags because we processed them elsewhere -->
  <xsl:template match="*">
    <!-- do nothing -->
  </xsl:template>

```

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

4. Open a command-line window and go to the folder with the XML and XSLT files.
5. Run the following command, where `path_to_saxon.jar` is the full path to the file `saxon9he.jar`:

```
java -jar "path_to_saxon.jar" -o  
petOutput.html -s:pets.xml -xsl:pettransform.xslt
```

For example:

```
java -jar "C:\Program Files\SaxonHE9-7-0-7J\saxon9he.jar" -o  
petOutput.html -s:pets.xml -xsl:pettransform.xslt
```

Make sure to run this command on a single line.

6. Open the output file `petOutput.html` in a text editor. It should look like this:

```
<html>  
  <head>  
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
    <title>Pets that are available for adoption</title>  
  </head>  
  <body>  
  
    <h2>Dogs</h2>  
    <table id="Dogs">  
      <tr>  
        <th colname="id">ID</th>  
        <th colname="name">Name</th>  
        <th colname="vaccinated">Vaccine status</th>  
        <th colname="health">Health status</th>  
      </tr>  
      <tbody>  
        <tr>  
          <td colname="id">123</td>  
          <td colname="name">Sparky</td>  
          <td colname="vaccinated">vaccinated</td>  
          <td colname="health">healthy</td>  
        </tr>  
        <tr>  
          <td colname="id">234</td>  
          <td colname="name">Sadie</td>  
          <td colname="vaccinated">unvaccinated</td>  
          <td colname="health">sick</td>  
        </tr>  
        <tr>  
          <td colname="id">345</td>  
          <td colname="name">Herman</td>  
          <td colname="vaccinated">unvaccinated</td>  
          <td colname="health">unknown</td>  
        </tr>  
      </tbody>  
    </table>  
  
    <h2>Cats</h2>  
    <table id="Cats">  
      <tr>
```

```
<th colname="id">ID</th>
<th colname="name">Name</th>
<th colname="vaccinated">Vaccine status</th>
<th colname="health">Health status</th>
</tr>
<tbody>
  <tr>
    <td colname="id">456</td>
    <td colname="name">Cleo</td>
    <td colname="vaccinated">vaccinated</td>
    <td colname="health">healthy</td>
  </tr>
  <tr>
    <td colname="id">567</td>
    <td colname="name">Janet</td>
    <td colname="vaccinated">unvaccinated</td>
    <td colname="health">healthy</td>
  </tr>
  <tr>
    <td colname="id">678</td>
    <td colname="name">Furry</td>
    <td colname="vaccinated">vaccinated</td>
    <td colname="health">sick</td>
  </tr>
</tbody>
</table>

</body>
</html>
```

7. Open the output file `petOutput.html` in a web browser. It should show the data in a simple table.

Read Getting started with xslt online: <https://riptutorial.com/xslt/topic/1129/getting-started-with-xslt>

Chapter 2: Variables in xslt

Examples

Xslt Variables

Global Variable: This variable is available everywhere in the xsl stylesheet. This variable should only be the child of <xsl:stylesheet> element.

Local variable: This variable is only available where it is declared.

See Below Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  exclude-result-prefixes="xs"
  version="2.0">
  <xsl:output omit-xml-declaration="yes"/>

  <xsl:variable name="a" select="5"/>      <!-- Global Variable -->

  <xsl:template match="/">
    <xsl:variable name="b" select="2"/> <!--Local Variable -->
    <xsl:value-of select="$a+$b"/>      <!--Addition of 'a' and 'b' -->
  </xsl:template>
</xsl:stylesheet>
```

The Output of the above code would be: **7**

There are two ways to define a value to the variable like:

By xpath expression in the **@select attribute of <xsl:variable> element** like:

```
<xsl:variable name="apple" select="'RED'"/>
```

OR

By the **content of the <xsl:variable> element** like:

```
<xsl:variable name="apple">RED</xsl:variable>
```

To call a declared variable use **\$** Sign with variable name, like in above code '\$a'

Read Variables in xslt online: <https://riptutorial.com/xslt/topic/10060/variables-in-xslt>

Credits

S. No	Chapters	Contributors
1	Getting started with xslt	Community , Daniel Haley , Dimitre Novatchev , Eero Helenius , JLRishe , Lukasz , Mads Hansen , Mathias Müller , Michael Kay , Nate , Tim McMackin
2	Variables in xslt	palo , Tim C