

XLST 2.0 - Abridged Quick Reference Sheet

Stylesheet structure

```
<xsl:transform
  xsl:import*, xsl:variable|xsl:param)*
  (declaration|xsl:variable|xsl:param)*
</xsl:transform>
```

```
<xsl:import
  href = anyURI />
```

Template Rules and Functions

```
<xsl:template
  match = pattern >
  xsl:param*, sequence-constructor*
</xsl:template>
```

```
<xsl:apply-templates
  select = expression >
  (xsl:sort|xsl:with-param)*
</xsl:apply-templates>
```

```
<xsl:call-template
  name = QName >
  xsl:with-param*
</xsl:call-template>
```

Node Creation

```
<xsl:element
  name = {string} >
  sequence-constructor
</xsl:element>
```

```
<xsl:attribute
  name = {string}
  select = expression >
  sequence-constructor
</xsl:attribute>
```

```
<xsl:text>
  character data
</xsl:text>
```

```
<xsl:value-of
  select = expression >
  sequence-constructor
</xsl:value-of>
```

Variables and parameters

```
<xsl:variable
  name = QName
  select = expression >
  sequence-constructor
</xsl:variable>
```

```
<xsl:param
  name = QName
  select = expression >
  sequence-constructor
</xsl:param>
```

Miscellaneous

```
<xsl:message
  sequence-constructor
</xsl:message>
```

Attributes for any element

```
default-collation = uri-list
exclude-result-prefixes =
  (NCName)*|"#all"
extension-element-prefixes = prefix-list
use-when = expression
xpath-default-namespace = anyURI
version = decimal
```

All declarations

```
xsl:attribute-set, xsl:character-map,
xsl:decimal-format, xsl:function,
xsl:import-schema, xsl:include, xsl:key,
xsl:namespace-alias, xsl:output,
xsl:preserve-space, xsl:strip-space,
xsl:template, xsl:variable
```

All instructions

```
xsl:analyze-string, xsl:apply-imports,
xsl:apply-templates, xsl:attribute,
xsl:call-template, xsl:choose,
xsl:comment, xsl:copy, xsl:copy-of,
xsl:document, xsl:element, xsl:fallback,
xsl:for-each, xsl:for-each-group, xsl:if,
xsl:message, xsl:namespace,
xsl:next-match, xsl:number,
xsl:perform-sort,
xsl:processing-instruction,
xsl:result-document, xsl:sequence,
xsl:text, xsl:value-of, xsl:variable
```

Conventions used in this sheet

bold	required attribute	<i>italic</i>	element content
{...}	attribute value template	,	sequence
?	optional		alternative
*	repetition 0 or more	+	repetition 1 or more

```
<xsl:with-param
  name = QName
  select = expression >
  sequence-constructor
</xsl:with-param>
```

Source copy

```
<xsl:copy
  sequence-constructor
</xsl:copy>
```

```
<xsl:copy-of
  select = expression />
```

Condition and iteration

```
<xsl:if
  test = expression >
  sequence-constructor
</xsl:if>
```

```
<xsl:choose>
  xsl:when+, xsl:otherwise?
</xsl:choose>
```

```
<xsl:when
  test = expression >
  sequence-constructor
</xsl:when>
```

```
<xsl:otherwise>
  sequence-constructor
</xsl:otherwise>
```

```
<xsl:for-each
  select = expression >
  xsl:sort*, sequence-constructor*
</xsl:for-each>
```

```
<xsl:for-each-group
  select = expression
  group-by = expression >
  xsl:sort*, sequence-constructor*
</xsl:for-each-group>
```

Sorting, searching and numbering

```
<xsl:sort
  sequence-constructor
</xsl:sort>
```