## CSIT884: Web Development

# eXtensible Stylesheet Language Transformations XSLT

School of Computing and Information Technology University of Wollongong

### **XSLT**

Extensible Stylesheet Language Transformations (XSLT) is an XML-based language used for the transformation of XML documents.

- file extension is .xsl
- used to transform XML file into other file formats, such as HTML
- describes how the XML elements should be displayed.

#### The content of the stylesheet XSL file looks like the following:

```
<?xml version="1.0"?>
<<u>xsl:stylesheet</u>
version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns="http://www.w3.org/1999/xhtml">
... xslt code here ...
</xsl:stylesheet>
```

This is an XML data representing exam information for a particular subject:

```
<exam subject="CSIT884">
    <title>Web Development</title>
        <venue>20.G01</venue>
        <date>15/06/2021</date>
        <time>1PM-4PM</time>
        <note>Closed book exam, calculator allowed</note>
</exam>
```

We would like to transform this XML content into the following HTML display:

### **CSIT884: Web Development**

20.G01 15/06/2021, 1PM-4PM

#### exam.xml

#### exam-with-style.xml

These two XML files have almost the same content.

The 2nd one has a reference to a **stylesheet**. Let's look at them on a browser.

exam.xml

exam-with-style.xml

### **CSIT884: Web Development**

20.G01 15/06/2021, 1PM-4PM

Closed book exam, calculator allowed

On web browser, the two XML files display differently.

This is because the second XML uses a **stylesheet**.

```
<?xml-stylesheet type="text/xsl" href="style-exam.xsl"?>
```

Let's have a look at the stylesheet file

Let's have a look at the stylesheet file: style-exam.xsl

```
<?xml version="1.0"?>
<xsl:stylesheet</pre>
 version="1.0"
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
 xmlns="http://www.w3.org/1999/xhtml">
  <xsl:output method="xml" indent="yes" encoding="UTF-8"/>
  <xsl:template match="/exam">
    < ht.ml>
      <head>
        <title>XSLT example</title>
      </head>
      <body>
                                      What do you see in this stylesheet file:
      </body>
                                      style-exam.xsl
    </html>
  </xsl:template>
</xsl:stylesheet>
```

```
Stylesheet file: style-exam.xsl
```

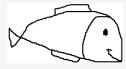
```
<?xml version="1.0"?>
<xsl:stylesheet</pre>
 version="1.0"
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
 xmlns="http://www.w3.org/1999/xhtml">
  <xsl:output method="xml" indent="yes" encoding="UTF-8"/>
  <xsl:template match="/exam"</pre>
    <html>
                                                        this is the root element
                                                        of the XML file
      <head>
         <title>XSLT example</title>
      </head>
      <body>
                              <?xml version="1.0" ?>
                              <exam subject="CSIT884">
      </body>
                                <title>Web Development</title>
                                <venue>20.G01</venue>
    </html>
                                <date>15/06/2021</date>
  </xsl:template>
                                <time>1PM-4PM</time>
                                <note>Closed book exam, calculator allowed</note>
</xsl:stylesheet>
                              </exam>
```

#### Stylesheet file: style-exam.xsl

```
<?xml version="1.0"?>
<xsl:stylesheet</pre>
  version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns="http://www.w3.org/1999/xhtml">
  <xsl:output method="xml" indent="yes" encoding="UTF-8"/>
  <xsl:template match="/exam">
    <html>
                                                       this looks like HTML code
      <head>
                                                        that's why the XML file is
         <title>XSLT example</title>
                                                       displayed on the browser
      </head>
                                                               as HTMI
      <body>
      </body>
                                                     CSIT884: Web Development
     </html>
                                                               20.G01
  </xsl:template>
                                                        15/06/2021, 1PM-4PM
                                                          Closed book exam, calculator allowed
</xsl:stylesheet>
```

#### Stylesheet file: style-exam.xsl

```
<?xml version="1.0"?>
<xsl:stylesheet</pre>
 version="1.0"
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
 xmlns="http://www.w3.org/1999/xhtml">
  <xsl:output method="xml" indent="yes" encoding="UTF-8"/>
  <xsl:template match="/exam">
    <html>
      <head>
        <title>XSLT example</title>
      </head>
      <body>
      </body>
     </html>
  </xsl:template>
</xsl:stylesheet>
```



Let's have a look at this code in more detail

#### **CSIT884: Web Development**

20.G01 15/06/2021, 1PM-4PM

Stylesheet file: style-exam.xsl

```
<body>
  <div align="center" style="background-color:#f0371930">
    <h1>
      <xsl:value-of select="@subject" />
      <xsl:text>: </xsl:text>
      <xsl:value-of select="title" />
    </h1>
    <font size="6" color="green">
      <xsl:value-of select="venue" />
      <br />
      <xsl:value-of select="date" />
      <xsl:text>, </xsl:text>
      <xsl:value-of select="time" />
    </font>
    <br />
    <xsl:value-of select="note" />
  </div>
</body>
```

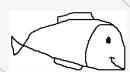


**CSIT884: Web Development** 

20.G01 15/06/2021, 1PM-4PM

```
Stylesheet file: style-exam.xsl
```

```
<body>
  <div align="center" style="background-color:#f0371930">
    <h1>
      <xsl:value-of select="@subject" />
      <xsl:text>: </xsl:text>
      <xsl:value-of select="title" />
    </h1>
    <font size="6" color="green">
      <xsl:value-of select="venue" />
      <br />
      <xsl:value-of select="date" />
      <xsl:text>, </xsl:text>
      <xsl:value-of select="time"</pre>
    </font>
    <br />
    <xsl:value-of select="note" />
  </div>
</body>
```



#### **CSIT884: Web Development**

20.G01 15/06/2021, 1PM-4PM

Stylesheet file: style-exam.xsl

#### **CSIT884: Web Development**

20.G01 15/06/2021, 1PM-4PM

#### Stylesheet file: style-exam.xsl

```
Get value from element
```

#### **CSIT884: Web Development**

20.G01 15/06/2021, 1PM-4PM

Stylesheet file: style-exam.xsl

#### **CSIT884: Web Development**

20.G01 15/06/2021, 1PM-4PM

```
Stylesheet file: style-exam.xsl
```

```
<body>
  <div align="center" style="background-color:#f0371930">
    <h1>
      <xsl:value-of select="@subject" /> CSIT884
      <xsl:text>: </xsl:text>
      <xsl:value-of select="title" /> Web Development
    </h1>
    <font size="6" color="green">
      <xsl:value-of select="venue" /> 20.G01
      <br />
     <xsl:value-of select="date" /> 15/06/2021
      <xsl:text>, </xsl:text>
      <xsl:value-of select="time" /> 1PM-4PM
    </font>
    <br />
    <xsl:value-of select="note" /> Closed book...
                                              CSIT884: Web Development
 </div>
</body>
```

20.G01

15/06/2021, 1PM-4PM

transaction0.xml uses stylesheet transaction-style0.xsl

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="transaction-style0.xsl"?>
<dailyTransaction date="24/02/2021">
 <person staffDbId="103" operation="update">
   <firstName>John</firstName>
   <lastName>Smith
   <mobile>0211223344
 </person>
 <person staffDbId="-1" operation="add">
   <firstName>Mary</firstName>
   <lastName>Jane
   <mobile>0244556677</mobile>
 </person>
</dailyTransaction>
```

Have a look at the XML file transaction 0.xml in the browser

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" href="transaction-style0.xsl"?>
<dailyTransaction date="24/02/2021">
  <person staffDbId="103" operation="update">
                                                     24/02/2021
    <firstName>John</firstName>
    <lastName>Smith
                                                     John
    <mobile>0211223344</mobile>
                                                     Smith
  </person>
                                                     0211223344
  <person staffDbId="-1" operation="add">
                                                     103
    <firstName>Mary</firstName>
                                                     update
    <lastName>Jane
    <mobile>0244556677</mobile>
                                                     Mary
  </person>
                                                     Jane
                                                     0244556677
</dailyTransaction>
                                                     -1
                                                     add
```

Let's look at the xml stylesheet: transaction-style0.xsl

```
<xsl:template match="/dailyTransaction">
   < ht.ml>
     <head>
       <title>XSLT example</title>
     </head>
                                      <dailyTransaction date="24/02/2021">
     <body>
                                        <person staffDbId="103" operation="update">
        . . .
                                         <firstName>John</firstName>
                                         <lastName>Smith
     </body>
                                         <mobile>0211223344</mobile>
   </html>
                                        </person>
</xsl:template>
                                        <person staffDbId="-1" operation="add">
                                         <firstName>Mary</firstName>
                                         <lastName>Jane
                                         <mobile>0244556677</mobile>
                                        </person>
                                      </dailyTransaction>
```

Let's look at the xml stylesheet: transaction-style0.xsl

```
<body>
  <xsl:value-of select="@date" /> <br /><br />
  <xsl:for-each select="person">
    <xsl:value-of select="firstName" />
    <br />
    <xsl:value-of select="lastName" />
    <br />
    <xsl:value-of select="mobile" />
    <br />
    <xsl:value-of select="@staffDbId" />
    <br />
    <xsl:value-of select="@operation" />
    <br />
    \langle br / \rangle
  </xsl:for-each>
</body>
```

#### 24/02/2021

John Smith 0211223344 103 update

Mary Jane 0244556677 -1 add

Let's look at the xml stylesheet: transaction-style0.xsl

```
<body>
  <xsl:value-of select="@date" /> <br /><br />
  <xsl:for-each_select="person">
    <xsl:value-of select="firstName" />
    <br />
    <xsl:value-of select="lastName" />
    <br />
    <xsl:value-of select="mobile" />
    <br />
    <xsl:value-of select="@staffDbId" />
    <br />
    <xsl:value-of select="@operation" />
    <br />
    \langle br / \rangle
  </xsl:for-each>
</body>
```

**FOR LOOP** 

24/02/2021

John Smith 0211223344 103 update

Mary Jane 0244556677 -1 add

Let's look at the xml stylesheet: transaction-style0.xsl

```
<body>
  <xsl:value-of select="@date" /> <br /><br />
  <xsl:for-each select="person">
    <xsl:value-of select="firstName" />
    <br />
    <xsl:value-of _select="lastName" />
    <br />
    <xsl:value-of select="mobile" />
    <br />
    <xsl:value-of select="@staffDbId" />
    <br />
    <xsl:value-of select="@operation" />
    <br />
    <br />
  </xsl:for-each>
</body>
```

24/02/2021

John
Smith
0211223344
103
update

Mary
Jane
0244556677
-1

add

Let's look at the xml stylesheet: transaction-style0.xsl

```
<body>
  <xsl:value-of select="@date" /> <br /><br />
  <xsl:for-each select="person">
    <xsl:value-of select="firstName" />
    <br />
    <xsl:value-of select="lastName" />
    <br />
    <xsl:value-of select="mobile" />
    <br />
    <xsl:value-of select="@staffDbId" />
   <br />
    <xsl:value-of select="@operation" />
   <br />
    <br />
  </xsl:for-each>
</body>
```

John Smith 0211223344 103 update Mary

24/02/2021

Jane 0244556677 -1 add

transaction1.xml uses stylesheet transaction-style1.xsl

```
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet type="text/xsl" href="transaction-style1.xsl"?>
<dailyTransaction date="24/02/2021">
  <person staffDbId="103" operation="update">
   <firstName>John</firstName>
   <lastName>Smith
   <mobile>0211223344
  </person>
  <person staffDbId="-1" operation="add">
   <firstName>Mary</firstName>
   <lastName>Jane
   <mobile>0244556677</mobile>
  </person>
</dailyTransaction>
```

Have a look at the XML file in the browser

- John Smith, 0211223344, 103, update
- Mary Jane, 0244556677, -1, add
- Matt Brown, 0455221212, 104, remove
- Jack Patel, 0244448888, 105, update
- Frank Jones, 0466889595, -1, add
- James Williams, 0222335566, 106, remove

Let's look at the xml stylesheet: transaction-style1.xsl

```
<h1>Daily transaction <xsl:value-of select="@date" /> </h1>
<111>
  <xsl:for-each select="person">
    <1i>>
      <xsl:value-of select="firstName" />
      <xsl:text> </xsl:text>
      <<u>xsl:value-of</u> select="lastName" />
      <xsl:text>, </xsl:text>
      <xsl:value-of select="mobile" />
      <xsl:text>, </xsl:text>
      <<u>xsl:value-of select="@staffDbId"</u> />
      <xsl:text>, </xsl:text>
      <xsl:value-of select="@operation" />
    </xsl:for-each>
```

- John Smith, 0211223344, 103, update
- Mary Jane, 0244556677, -1, add
- Matt Brown, 0455221212, 104, remove
- Jack Patel, 0244448888, 105, update
- Frank Jones, 0466889595, -1, add
- James Williams, 0222335566, 106, remove

Version 2 is exactly like version 1 except that we only display staffDbId if it is a positive number.

- John Smith, 0211223344, 103, update
- Mary Jane, 0244556677, add
- Matt Brown, 0455221212, 104, remove
- Jack Patel, 0244448888, 105, update
- Frank Jones, 0466889595, add
- James Williams, 0222335566, 106, remove



IF statement will be applied when a specified condition is true.

Use CHOOSE-WHEN-OTHERWISE statement (<xsl:choose> element with one <xsl:when> and one <xsl:otherwise> children) to express multiple conditional tests.

- John Smith, 0211223344, 103, update
- Mary Jane, 0244556677, add
- Matt Brown, 0455221212, 104, remove
- Jack Patel, 0244448888, 105, update
- Frank Jones, 0466889595, add
- James Williams, 0222335566, 106, remove

### **Conditional statement examples**

```
Mark = 49:
                            <xsl:if test="@staffDbId &qt; 0">
mark = 49
                               <xsl:text>, </xsl:text>
                               <xsl:value-of select="@staffDbId" />
Mark not equal to 49:
                            </xsl:if>
mark != 49
Student type = 'U':
type = 'U'
Student type not equal to 'U':
type != 'U'
Mark > 35:
mark &gt; 35
Mark >= 85:
                               Mark NOT equal to 49:
mark &gt;= 85
                               not(mark = 49)
Mark < 35:
                                Student type = 'U' or 'P':
mark &lt; 35
                                (type = 'U') or (type = 'P')
Mark <= 85:
mark &lt;= 85
                                Mark >= 75 and mark < 85:
                                (mark &qt;= 75) and (mark < 85)
```

Now look at transaction3.xml, it uses an xml stylesheet: transaction-style3.xsl

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="transaction-style3.xsl"?>
<dailyTransaction date="24/02/2015">
 <person staffDbId="103" operation="update">
   <firstName>John</firstName>
   <lastName>Smith
   <mobile>0211223344
 </person>
 <person staffDbId="-1" operation="add">
   <firstName>Mary</firstName>
   <lastName>Jane
   <mobile>0244556677</mobile>
 </person>
</dailyTransaction>
```

Now look at transaction3.xml, it uses an xml stylesheet: transaction-style3.xsl

```
<body>
 <h1>Daily transaction <xsl:value-of select="@date" /> </h1>
 <t.r>
   Name
    mobile
                             Header row
    Staff ID
    Operation
                                             Each "person"
   node is presented
                                             as a row in a table.
   <xsl:for-each select="person">
      <xsl:value-of select="firstName" />
             <xsl:text> </xsl:text>
             <xsl:value-of select="lastName" /> 
         <xsl:value-of select="mobile" /> 
         <xsl:value-of select="@staffDbId" /> 
         <xsl:value-of select="@operation" /> 
      </xsl:for-each>
 body>
```

View the source of the xml stylesheet: transaction-style3.xsl we can see that it displays a table.

Name	mobile	Staff ID	Operation
John Smith	0211223344	103	update
Mary Jane	0244556677	-1	add
Matt Brown	0455221212	104	remove
Jack Patel	0244448888	105	update
Frank Jones	0466889595	-1	add
James Williams	0222335566	106	remove

### Daily transaction 24/02/2021

Name	mobile	Staff ID	Operation
Matt Brown	0455221212	104	remove
Mary Jane	0244556677	-1	add
Frank Jones	0466889595	-1	add
Jack Patel	0244448888	105	update
John Smith	0211223344	103	update
James Williams	0222335566	106	remove

Sorted by last name.

### Daily transaction 24/02/2021

Name	mobile	Staff ID	Operation
Mary Jane	0244556677	-1	add
Frank Jones	0466889595	-1	add
Matt Brown	0455221212	104	remove
James Williams	0222335566	106	remove
John Smith	0211223344	103	update
Jack Patel	0244448888	105	update

Sorted by operation attribute

### Daily transaction 24/02/2021

Name	mobile	Staff ID	Operation
Mary Jane	0244556677	-1	add
Frank Jones	0466889595	-1	add
John Smith	0211223344	103	update
Jack Patel	0244448888	105	update
James Williams	0222335566	106	remove
Matt Brown	0455221212	3	remove

Name	mobile	Staff ID	Operation
Mary Jane	0244556677	-1	add
Frank Jones	0466889595	-1	add
Matt Brown	0455221212	3	remove
John Smith	0211223344	103	update
Jack Patel	0244448888	105	update
James Williams	0222335566	106	remove

Now look at transaction4.xml

Name	mobile	Staff ID	Operation
John Smith	0211223344	103	update
Mary Jane	0244556677	-1	add
Matt Brown	0455221212	3	remove
Jack Patel	0244448888	105	update
Frank Jones	0466889595	-1	add
James Williams	0222335566	106	remove

Now look at transaction-style4.xsl

```
<xsl:choose>
 <xsl:when test="@operation = 'remove'">
   <xsl:value-of select="@operation" />
   </xsl:when>
 <xsl:when test="@operation = 'add'">
   <xsl:value-of select="@operation" />
   </xsl:when>
 <xsl:otherwise>
   <xsl:value-of select="@operation" />
   </xsl:otherwise>
</xsl:choose>
```

Now look at transaction-style5.xsl

If the nested element "mobile" is empty, the cell in the table will be in red colour.

# Daily transaction 24/02/2021

Name	mobile	Staff ID	Operation
John Smith	0211223344	103	update
Mary Jane		-1	add
Matt Brown	0455221212	3	remove
Jack Patel	0244448888	105	update
Frank Jones		-1	add
James Williams	0222335566	106	remove

Now look at transaction6.xml

### Daily transaction 24/02/2021

#### operation add

Name	mobile	Staff ID	Operation
Mary Jane	0244556677	-1	add
Frank Jones	0466889595	-1	add

### Daily transaction 24/02/2021

#### operation remove

Name	mobile	Staff ID	Operation
Matt Brown	0455221212	3	remove
James Williams	0222335566	106	remove

### Daily transaction 24/02/2021

#### operation update

Name	mobile	Staff ID	Operation
Jack Patel	0244448888	105	update
John Smith	0211223344	103	update

Now look at transaction-style6.xsl

```
<xsl:for-each select="person[@operation='add']">
 <xsl:sort select="lastName"/>
 <t.d>
     <xsl:value-of select="firstName" />
     <xsl:text> </xsl:text>
     <xsl:value-of select="lastName" />
   <t.d>
     <xsl:value-of select="mobile" />
   <t.d>
     <xsl:value-of select="@staffDbId" />
   <xsl:value-of select="@operation" />
   </xsl:for-each>
```

Now look at transaction7.xml

# Daily transaction 24/02/2021

Name	Mobile	Staff Id	Operation
Matt Brown	0413273345	104	×
Mary Jane	0244556677	-1	+
Frank Jones	0244556677	-1	+
Jack Patel	0211223344	105	
John Smith	0211223344	103	<b>S</b>
James Williams	0413273345	106	×

Now look at transaction-style7.xsl

```
<t.d>
 <xsl:value-of select="@staffDbId" />
<imq>
   <xsl:attribute name="src">
     <xsl:text>images/</xsl:text>
     <xsl:value-of select="@operation"/>
     <xsl:text>.pnq</xsl:text>
   </xsl:attribute>
   <xsl:attribute name="width">
     <xsl:text>30px</xsl:text>
   </xsl:attribute>
                                            OUTPUT:
 </imq>
                           <img src="images/add.png" width="30px"/>
```

Now look at transaction8.xml

# Daily transaction 24/02/2021

Name	Mobile	Staff Id	Operation
Matt Brown	0413273345	104	remove
Mary Jane	0244556677	-1	add
Frank Jones	0244556677	-1	add
Jack Patel	0211223344	105	update
John Smith	0211223344	103	update
James Williams	0413273345	106	remove

Now look at transaction-style8.xsl

```
<xsl:choose>
    <xsl:when test="@staffDbId &lt; 0">
      <span <u>style="color:red"</u>>
        <xsl:value-of select="@staffDbId"</pre>
      </span>
    </xsl:when>
    <xsl:otherwise>
     <span style="color:green">
        <xsl:value-of select="@staffDbId"</pre>
      </span>
    </xsl:otherwise>
  </xsl:choose>
```

# **Summary**

#### **Get attribute value:**

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

#### **Get element value:**

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

### Literal text:

<xsl:text>some text here ...</xsl:text>

# **Summary**

### FOR loop:

### FOR loop with sort:

```
<xsl:for-each select="element-name">
    <xsl:sort select="field-to-be-sorted"/>
    ...
</xsl:for-each>
```

### FOR loop with filter:

```
<xsl:for-each select="element-name[filter-condition]">
    ...
</xsl:for-each>
```



### IF statement: (note, there is no IF-ELSE)

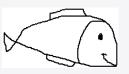
```
<xsl:if test="the-if-condition">
    ...
</xsl:if>
```



#### **CHOOSE-WHEN-OTHERWISE statement:**

```
<xsl:choose>
  <xsl:when test="condition1">
  </xsl:when>
  <xsl:when test="condition2">
  </xsl:when>
  <xsl:otherwise>
  </xsl:otherwise>
</xsl:choose>
```

# **Summary**



```
Mark = 49:
mark = 49
Mark not equal to 49:
mark != 49
Student type = 'U':
type = 'U'
Student type not equal to 'U':
type != 'U'
Mark > 35:
mark &qt; 35
Mark >= 85:
mark > = 85
Mark < 35:
mark < 35
Mark <= 85:
mark < = 85
```

Mark NOT equal to 49:
not (mark = 49)

Student type = 'U' or 'P':
(type = 'U') or (type = 'P')

Mark >= 75 and mark < 85:
(mark &gt; = 75) and (mark &lt; 85)</pre>