

---

# XML Technologies

---

XSLT

---

# An Empty XSLT Script

---

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

  <xsl:output method="xml" indent="yes"/>

</xsl:stylesheet>
```

---

# XSLT Templates

---

## □ Unnamed

```
<xsl:template match="xpath_expression(s)">  
    ...  
</xsl:template>
```

## □ Named

```
<xsl:template name="template_name">  
    ...  
</xsl:template>
```

---

# Implicit XSLT Templates

---

## ❑ Matching root node

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

## ❑ Matching element nodes

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

---

# Implicit XSLT Templates

---

## □ Matching text nodes

```
<xsl:template match="text()">
  <xsl:value-of select="." />
</xsl:template>
```

## □ Matching attribute nodes

```
<xsl:template match="@*">
  <xsl:value-of select="." />
</xsl:template>
```

---

|                                                                                                                                                                                  |                                                                                                                                          |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <code>&lt;xsl:apply-templates<br/>  select="xpath" /&gt;</code>                                                                                                                  | Application of templates on nodes selected using <b>xpath</b> (if <b>select</b> is not specified, we apply the templates on child nodes) |
| <code>&lt;xsl:value-of select="xpath"/&gt;</code>                                                                                                                                | Output of values specified using <b>xpath</b>                                                                                            |
| <code>&lt;xsl:text&gt;...&lt;/xsl:text&gt;</code>                                                                                                                                | Construction of a text node                                                                                                              |
| <code>&lt;xsl:attribute name="str"&gt;<br/>  ... &lt;/xsl:attribute&gt;</code>                                                                                                   | Construction of an attribute                                                                                                             |
| <code>&lt;xsl:element name="str"&gt;<br/>  ... &lt;/xsl:element&gt;</code>                                                                                                       | Construction of an element                                                                                                               |
| <code>&lt;element-name atr="str"&gt;<br/>  ... &lt;/element-name&gt;</code>                                                                                                      | Construction of an element <b>element-name</b> , having attribute <b>atr</b> with value <b>str</b>                                       |
| <code>&lt;xsl:for-each select="xpath"&gt;<br/>  ... &lt;/xsl:for-each&gt;</code>                                                                                                 | Iteration                                                                                                                                |
| <code>&lt;xsl:if test="xpath"&gt; ... &lt;/xsl:if&gt;</code>                                                                                                                     | Condition (without else branch)                                                                                                          |
| <code>&lt;xsl:choose&gt;<br/>  &lt;xsl:when test="xpath"&gt;...&lt;/xsl:when&gt;<br/>  ...<br/>  &lt;xsl:otherwise&gt; ... &lt;/xsl:otherwise&gt;<br/>&lt;/xsl:choose&gt;</code> | Generalization of condition (with else branch)                                                                                           |
| <code>&lt;xsl:variable name="var"<br/>  select="xpath" /&gt;</code>                                                                                                              | Variable, accessible via <b>\$var</b> .                                                                                                  |
| <code>&lt;xsl:param name="var"<br/>  select="xpath" /&gt;</code>                                                                                                                 | Parameter                                                                                                                                |

# Modes of Templates

---

## □ Template definition

```
<xsl:template match="aaa" mode="overview">
...</xsl:template>

<xsl:template match="aaa" mode="detail">
...</xsl:template>
```

## □ Template calling

```
<xsl:template match="/">
  <xsl:apply-templates select="aaa" mode="overview" />
</xsl:template>
```

```
<xsl:template match="xyz">
  <xsl:apply-templates select="aaa" mode="detail" />
</xsl:template>
```

# Named Templates with Parameters

---

## □ Template definition

```
<xsl:template name="template">
  <xsl:param name="parameter" />
  ...
  <!-- example of usage of parameter -->
  <xsl:value-of select="$parameter" />
</xsl:template>
```

## □ Template calling

```
<xsl:call-template name="template">
  <xsl:with-param name="parameter"
    select="xpath expression" />
</xsl:call-template>
```

# Common Errors

---

- ❑ Don't forget that we have implicit templates

```
<xsl:stylesheet>
  <xsl:template match="aaa">
    <xsl:value-of select="@attr" />
  </xsl:template>
</xsl:stylesheet>
```

- ❑ It does not print only the values of attribute `attr` of element `aaa!!!`

```
<xsl:template match="node()" />
```

---

# Common Errors

---

- ❑ Don't forget how `<xsl:apply-templates />` works
  - It moves the processing to child nodes, but not attributes
    - ❑ XPath axis `child::`
- ❑ To move to attributes we must use:

```
<xsl:apply-templates select="@*" />
```

---