Further Features

- xsl:for-each

- xsl:sort

- xsl:if

- xsl:choose

- xsl:variable

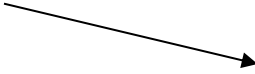# Further Features - xsl:for-each

```
<courses>
    <course>
        <title>SSD</title>
        <day> Thursday </day>
        <time> 09:15 </time>
    </course>
    <course>
        <title>Databases</title>
        <day> Tuesday </day>
        <time> 09:15 </time>
    </course>
</courses>
```

The structure of the course element is not known

We only know that the first child of each course element is title

```
<courses>
    <SSD>
        <day> Thursday </day>
        <time> 09:15 </time>
    </SSD>
    <Databases>
        <day> Tuesday </day>
        <time> 09:15 </time>
    </Databases>
</courses>
```

Further Features - xsl:for-each

```
<xsl:template match="courses">

    <courses>

        <xsl:apply-templates select="course"/>

    </courses>

</xsl:template>


<xsl:template match="course">

    <xsl:element name="{title/text()}">

        <xsl:for-each select="*[position() &gt; 1]">

            <xsl:copy-of select="."/>

        </xsl:for-each>

    </xsl:element>

</xsl:template>
```
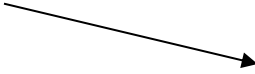
creates a deep copy
of the selected node

Further Features - xsl:sort

```
<courses>
    <course>
        <title>SSD</title>
        <day> Thursday </day>
        <time> 09:15 </time>
    </course>
    <course>
        <title>Databases</title>
        <day> Tuesday </day>
        <time> 09:15 </time>
    </course>
</courses>
```

Sort by name

```
<courses>
    <Databases>
        <day> Tuesday </day>
        <time> 09:15 </time>
    </Databases>
    <SSD>
        <day> Thursday </day>
        <time> 09:15 </time>
    </SSD>
</courses>
```

Further Features - xsl:sort

```
<xsl:template match="courses">
    <courses>
        <xsl:for-each select="course">
            <xsl:sort select="title" order="ascending"/>
            <xsl:element name="{title/text()}">
                <xsl:for-each select="*[position() &gt; 1]">
                    <xsl:copy-of select="."/>
                </xsl:for-each>
            </xsl:element>
        </xsl:for-each>
    </courses>
</xsl:template>
```

Further Features - xsl:if

```
<courses>
    <course>
        <title>SSD</title>
        <day> Thursday </day>
        <time> 09:15 </time>
    </course>
    <course>
        <day> Tuesday </day>
        <title>Databases</title>
        <time> 09:15 </time>
    </course>
</courses>
```
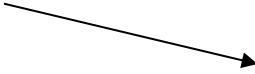
The structure of the course element is not known

The first child of each course element is not necessarily title

```
<courses>
    <Databases>
        <day> Tuesday </day>
        <time> 09:15 </time>
    </Databases>
    <SSD>
        <day> Thursday </day>
        <time> 09:15 </time>
    </SSD>
</courses>
```
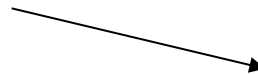
Further Features - xsl:if

```xml
<xsl:template match="courses">
    <courses>
        <xsl:for-each select="course">
            <xsl:sort select="title" order="ascending"/>
            <xsl:element name="{title/text()}">
                <xsl:for-each select="*">
                    <xsl:if test="local-name() != 'title' ">
                        <xsl:copy-of select="."/>
                    </xsl:if>
                </xsl:for-each>
            </xsl:element>
        </xsl:for-each>
    </courses>
</xsl:template>
```

Further Features - xsl:choose

```
<courses>
    <course>
        <title>SSD</title>
        <day> Thursday </day>
        <time> 09:15 </time>
    </course>
    <course>
        <title>Databases</title>
        <day> Tuesday </day>
        <time> 09:15 </time>
    </course>
</courses>
```

```
<courses>
    <Databases>
        Tuesday, 09:15.
    </Databases>
    <SSD>
        Thursday, 09:15.
    </SSD>
</courses>
```

Further Features - xsl:choose

```
<xsl:template match="courses">
    <courses>
        <xsl:for-each select="course">
            <xsl:sort select="title" order="ascending"/>
            <xsl:element name="{title/text()}">
                <xsl:for-each select="*">
                    <xsl:if test="local-name() != 'title' ">
                        <xsl:value-of select="."/>
                        <xsl:choose>
                            <xsl:when test="position() = last()">.</xsl:when>
                            <xsl:otherwise>,</xsl:otherwise>
                        </xsl:choose>
                    </xsl:if>
                </xsl:for-each>
            </xsl:element>
        </xsl:for-each>
    </courses>
</xsl:template>
```

Further Features - xsl:variable

```
<university>
    <courses>
        <course taughtBy="L1">
            <title>SSD</title>
            <day> Thursday </day>
            <time> 09:15 </time>
        </course>
        <course taughtBy="L2">
            <title>Databases</title>
            <day> Tuesday </day>
            <time> 09:15 </time>
        </course>
    </courses>
    <lecturers>
        <lecturer id="L1">Mantas Šimkus</lecturer>
        <lecturer id="L2">Reinhard Pichler</lecturer>
    </lecturers>
</university>
```

```
<courses>
    SSD taught by Mantas Šimkus
    Databases taught by Reinhard Pichler
</courses>
```

Further Features - xsl:variable

```
<xsl:template match="university">

    <xsl:apply-templates select="courses"/>

</xsl:template>


<xsl:template match="courses">

    <courses>

        <xsl:apply-templates select="course"/>

    </courses>

</xsl:template>


<xsl:template match="course">

    <xsl:value-of select="title"/> taught by <xsl:value-of select="//lecturer[@id = @taughtBy]"/>

</xsl:template>
```

Result:

```
<courses>
    SSD taught by
    Databases taught by
</courses>
```

?

Further Features - xsl:variable

```
<xsl:template match="university">

      <xsl:apply-templates select="courses"/>

</xsl:template>


<xsl:template match="courses">

      <courses>

            <xsl:apply-templates select="course"/>

      </courses>

</xsl:template>


<xsl:template match="course">

      <xsl:variable name="var" select="@taughtBy"/>

      <xsl:value-of select="title"/> taught by <xsl:value-of select="//lecturer[@id = $var]"/>

</xsl:template>
```