

WPUB - XSLT /2

6. cvičenie

25.3. 2021

Eduard Kuric

Šablóna

Šablóna definuje, ako sa časti (XML fragmenty) transformovaného dokumentu vyhovujúce **výrazu** spracujú/transformujú.

```
<xsl:template match="XPATH pattern">
```

```
...
```

```
</xsl:template>
```

Pozn. ďalej **výraz** = XPATH výraz

XSLT procesor

- Načíta vstupný XML dokument, vytvorí „stromovú“ reprezentáciu
- Postupne prechádza stromom, od koreňa, spôsobom do hĺbky (v poradí, akom sú elementy v dokumente)
- V momente, keď nájde pre element šablónu, začne generovať výstup zodpovedajúci danej šablóne
 - Potomkovia predmetného elementu nie sú ďalej spracovávaní.
 - Ak chceme, aby boli, musíme použiť vo vnútri šablóny `apply-templates`.
 - Tým zabezpečíme, že sa budú hľadať aj pre potomkov zodpovedajúce šablóny.

`<xsl:value-of select="výraz">`

- Ak chceme použiť v šablóne iba obsah elementu alebo jeho potomkov, bez aplikovania šablóny

<xsl:apply-templates /

- Ak použijeme vo vnútri šablóny:
- <xsl:apply-templates/> začnú sa hľadať šablóny pre všetky deti aktuálneho elementu
- <xsl:apply-templates select="výraz"/> hľadanie šablóny pre inú časť dokumentu, môžeme sa odvolať na úplne iné miesto, mimo aktuálneho elementu (pozor na zacyklenie)

Atribút mode

- Niektoré elementy chceme spracovať opakovane, ale iným spôsobom

```
<xsl:apply-templates select="//item"/>  
<xsl:apply-templates select="//item"  
mode="detailed"/>
```

```
<xsl:template match="item">  
</xsl:template>
```

```
<xsl:template match="item" mode="detailed">  
</xsl:template>
```

Generovanie výstupu /2

```
<xsl:text>Text</xsl:text>
```

```
<xsl:element name="{vyraz  obsah elementu  
</xsl:element>
```

```
<xsl:element name="{concat('s', 'p', 'a', 'n')}">  
  <xsl:attribute name="class">bold</xsl:attribute>  
  obsah elementu  
</xsl:element>
```

```
// <span class="bold">obsah elementu</span>
```

Attribute set

```
<xsl:attribute-set name="heading">
  <xsl:attribute name="align">center</xsl:attribute>
  <xsl:attribute name="class">level1</xsl:attribute>
  <xsl:attribute name="style">color: black</xsl:attribute>
</xsl:attribute-set>
```

- `<h2 xsl:use-attribute-sets="heading">Úvod</h1>`
- `// <h2 align="center" class= "level1" style="color: black">Introduction</h1>`

<xsl:comment>

<xsl:comment>Komentár</xsl:comment>

<!--Komentár-->

```
<xsl:if>
```

```
<xsl:if test="výraz">
```

```
...
```

```
</xsl:if>
```

choose

```
<xsl:choose>  
  <xsl:when test="výraz">  
  </xsl:when>  
  <xsl:when test="výraz">  
  </xsl:when>  
  ...  
  <xsl:otherwise>  
  </xsl:otherwise>  
</xsl:choose>
```

```
<xsl:for-each>
```

```
  <xsl:for-each select="order/item">
```

```
    <xsl:call-template name="showTitle">
```

```
      <xsl:with-param name="title" select =  
"title" />
```

```
    </xsl:call-template>
```

```
</xsl:for-each>
```

`<xsl:param>`

Lokálna – vo vnútri šablóny

Globálna – na top úrovni

```
<xsl:param name="color" select= "výraz" />
```

...

Musí byť použitý pred akýmkoľvek iným elementom v šablóne.

Hodnotu vieme zmeniť cez `xsl:with-param`

`<xsl:variable>`

Lokálna – vo vnútri šablóny

Globálna – na top úrovni

`<xsl:variable name="color" select= "výraz" />`

Po priradení nie je možné hodnotu zmeniť