

Transformace XML dokumentů pomocí XSLT

Podklady pro cvičení k předmětu Technologie XML

Irena Mlýnková
Martin Nečaský

XSLT skript

□ Prázdný XSLT skript:

```
<?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 šablony

□ Nepojmenovaná šablona:

```
<xsl:template match="xpath expression">  
    ...  
</xsl:template>
```

□ Pojmenovaná šablona:

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

Implicitní XSLT šablony

□ Matchuje kořenový uzel:

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

□ Matchuje elementy:

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

Implicitní XSLT šablony

□ Matchuje textové uzly:

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

<code><apply-templates select="xpath" /></code>	Aplikace možných šablon na uzly vybrané pomocí xpath (pokud není select , jde se do dětí).
<code><value-of select="xpath"/></code>	Výstup hodnot vybraných pomocí xpath .
<code><text>...</text></code>	Konstr. text. uzlu s hodnotou ... (text, value-of , ...).
<code><attribute name="str"> ... </attribute></code>	Konstr. atributu s názvem daným pomocí str a hodnotou
<code><element-name> ... </element-name></code>	Konstr. elementu s názvem element-name a hodnotou ... (text, podelementy, XSLT konstrukce).
<code><for-each select="xpath"> ... </for-each></code>	Iterace přes sekvenci uzlů vybranou pomocí xpath . Pro každý uzel se vykoná ..., přičemž uzel je v ... přístupný přes xpath výraz ..
<code><if test="xpath"> ... </if></code>	Podmíněný XSLT příkaz.
<code><choose> <when test="xpath">...</when> ... <otherwise> ... </otherwise> </choose></code>	Zobecnění podmíněného XSLT příkazu na více větví.
<code><variable name="var" select="xpath" /></code>	Nastavení proměnné. Hodnota je přístupná přes \$var .
<code><parameter name="var" select="xpath" /></code>	Nastavení proměnné. Chová se jako parametr (hodnota může být změněna voláním šablony, viz dál)

XSLT šablony s parametry

□ Šablona s parametrem:

```
<xsl:template name="template">
  <xsl:param name="parameter" />
  ...
  <!-- příklad využití hodnoty parametru -->
  <xsl:value-of select="$parameter" />
</xsl:template>
```

□ Volání šablony s parametrem:

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

Módy XSLT šablon

□ Šablona může mít mód:

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

□ Volání šablony s módem:

- pokud lze aplikovat více šablon, vybere se ta s volaným módem

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


Výstup do více souborů (XSLT 2)

- ❑ Pomocí konstrukce **output** definujeme výstupní formáty (pro každý výstup jedna pojmenovaná konstrukce **output**):

```
<xsl:output name="name1" method="xml" />
<xsl:output name="name2" method="html" />
```

- ❑ Pomocí konstrukce **result-document** směřujeme data do konkrétního dokumentu s daným formátem:

```
<xsl:template match="/">
  <xsl:result-document href="doc1.html" format="name2">
    <html>
      <body><xsl:apply-templates /></body>
    </html>
  </xsl:result-document>
  <xsl:for-each select="//aaa">
    <xsl:result-document href="doc{./@code}.html" format="name1">
      <xsl:apply-templates select="." />
    </xsl:result-document>
  </xsl:for-each>
</xsl:template>
```

Časté chyby v XSLT skriptech

- ❑ Pozor na implicitní šablony! Např. následující vypíše také všechny textové uzly "okolo" elementů **aaa**

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

- ❑ Řešení:

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

Časté chyby v XSLT skriptech

- ❑ Pozor na chování `<xsl:apply-templates />`.
 - Posouvá pouze do dětí, tj. ignoruje atributy
- ❑ Pro posun do atributů musíme použít:

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