



XSL (eXtensible Stylesheet Language)
XSLT (eXtensible Stylesheet Language
Transformation)
XPath



- **Jazyk** pro transformaci XML dokumentů do jiných XML dokumentů nebo do jiných typů dokumentů (HTML, XHTML, VRML, PDF)
- Vytvořen pomocí jazyka **XML**



- **XSLT** – jazyk pro transformaci XML dokumentů
- **XPath** – jazyk pro navigaci v XML dokumentech
- **XSL-FO** – jazyk pro formátování XML dokumentů



- XSLT = XSL Transformace
- XSLT využívá XPath pro navigaci v XML dokumentu
- XSLT je Web Standard (W3C)



- Mozilla Firefox od verze 1.0.2
- Mozilla s podporou – není implicitní
- Netscape od verze 8
- Opera od verze 9
- IE od verze 6

XSL – kořenový prvek



```
<xsl:stylesheet version="1.0"  
xmlns:xsl="http://www.w3.org/1999/XSL  
L/Transform">
```

nebo

```
<xsl:transform version="1.0"  
xmlns:xsl="http://www.w3.org/1999/XSL  
L/Transform">
```

XSL – XML pro transformaci



```
<?xml version="1.0"?>
<catalog>
  <cd>
    <title>Empire Burlesque</title>
    <artist>Bob Dylan</artist>
    <country>USA</country>
    <company>Columbia</company>
    <price>10.90</price>
    <year>1985</year>
  </cd>
  ...
</catalog>
```

XSL – XSL pro transformaci



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
    <h2>My CD Collection</h2>
    <table border="1">
      <tr><th>Title</th><th>Artist</th></tr>
      <xsl:for-each select="catalog/cd">
        <tr><td><xsl:value-of select="title"/></td>
          <td><xsl:value-of select="artist"/></td></tr>
      </xsl:for-each>
    </table>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```


XSL – Spojení XML a XSL



```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

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

```
<catalog>
```

```
...
```



- Šablona je spjata s prvkem dokumentu
- Atribut match určuje element s využitím jazyka XPath
- / znamená celý dokument – kořenový element

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

```
...
```

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



- Hodnoty elementů se načítají s využitím značky value-of
- Atribut select identifikuje element s využitím XPath
- Cesta je relativní vůči nadřazenému elementu v našem případě je to kořenový element

```
<xsl:value-of  
  select="catalog/cd/title" />
```



- Průchod všemi určenými značkami
- Atribut `select` identifikuje značky s využitím XPath
- Cesta je relativní vůči nadřazenému elementu v našem případě je to kořenový element

```
<xsl:for-each select="catalog/cd">
```

```
...
```

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



- S využitím XPath je možné filtrovat značky

```
<xsl:for-each  
  select="catalog/cd[artist='Bob  
  Dylan']">
```

...

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



- Element sort zajistí setřídění výstupu for-each operace
- S využitím XPath je možné definovat způsob třídění

```
<xsl:for-each select="catalog/cd">  
    <xsl:sort select="artist"/>  
    ...  
</xsl:for-each>
```



- Element if zajistí zpracování jen vybraných záznamů

```
<xsl:for-each select="catalog/cd">  
  <xsl:if select="price > 10">  
    ...  
  </xsl:if>  
  <xsl:if select="price < 10">  
    ...  
  </xsl:if>  
</xsl:for-each>
```



- Element choose s elementy when a otherwise

```
<xsl:choose>
```

```
  <xsl:when test="price > 10">
```

```
    <td bgcolor="#ff00ff">
```

```
      <xsl:value-of
```

```
select="artist"/></td>
```

```
  </xsl:when>
```

```
  <xsl:otherwise>
```

```
    <td><xsl:value-of
```

```
select="artist"/></td>
```

```
  </xsl:otherwise>
```

```
</xsl:choose>
```


XSL – apply-templates



```
<xsl:template match="/">
<h2>My CD Collection</h2>
<xsl:apply-templates/>
</xsl:template>
<xsl:template match="cd">
<p>
<xsl:apply-templates select="title"/>
<xsl:apply-templates select="artist"/>
</p>
</xsl:template>
<xsl:template match="title">
Title: <span style="color:#ff0000">
<xsl:value-of select="."/;></span>
<br />
</xsl:template>
...
</xsl:stylesheet>
```

XSL – výběr 2



- Element choose s elementy when a otherwise

```
<xsl:choose>
  <xsl:when test="price > 10">
    <td bgcolor="#ff00ff">
      <xsl:value-of
select="artist"/></td>
    </xsl:when>
    <xsl:otherwise>
      <td><xsl:value-of
select="artist"/></td>
    </xsl:otherwise>
  </xsl:choose>
```

XPath – atributy



- Získání hodnoty atributu

```
<xsl:for-each  
  select="METAIS/DATASET/OBJECT_STANDARD/OBJECT_STANDARD_NAZEV_ALT">  
  <xsl:value-of  
    select="@NAZEV_ALT" />  
</xsl:for-each>
```

```
<xsl:value-of  
  select="METAIS/DATASET/@PSCHEMA_U" /  
>
```

XPath – testování atributu



- Testování hodnoty atributu

```
<xsl:for-each
  select="METAIS/DATASET/DATASET_K_PS
  CHEMA">
  <xsl:choose>
    <xsl:when
      test="@K_PSCHEMA[.='spatialG0']">Zá
      kladní schéma G0</xsl:when>
  <xsl:when
      test="@K_PSCHEMA[.='spatialG1']">Úp
      lný rovinný graf G1</xsl:when>
  </xsl:choose>
</xsl:for-each>
```



- Testování hodnoty atributu

```
<xsl:when
```

```
test="@K_PSCHEMA[.='spatialG2' or
```

```
.='3']">Liniová síť s rovinným
```

```
grafem G2</xsl:when>
```



- Testování hodnoty atributu

```
<xsl:when
```

```
test="@K_PSCHEMA[.='spatialG2' or
```

```
.='3']">Liniová síť s rovinným
```

```
grafem G2</xsl:when>
```



- Testování hodnoty atributu

```
<xsl:when  
  test="METAIS/DATASET/@MERITKO_OD[.!  
  =' ']">od 1:<xsl:value-of  
  select="METAIS/DATASET/@MERITKO_OD"  
/> do 1:<xsl:value-of  
  select="METAIS/DATASET/@MERITKO_DO"  
/>
```



- Testování počtu

```
<xsl:when  
  test="/METAIS ['count (/ORG) ' != '0'] ">  
<xsl:for-each select="METAIS/ORG">  
  ...  
</xsl:when>
```


XSL – vnořené testování



```
<xsl:choose>
<xsl:when test="@MERITKO_DO[.!=' ']">
<xsl:choose>
<xsl:when test="@MERITKO_OD[.!=' ']">od 1:<xsl:value-of
  select="@MERITKO_OD"/> do 1:<xsl:value-of
  select="@MERITKO_DO"/>
</xsl:when>
<xsl:otherwise>od 1:? do 1:<xsl:value-of
  select="@MERITKO_DO"/>
</xsl:otherwise>
</xsl:choose>
</xsl:when>
<xsl:otherwise>
<xsl:choose>

<xsl:when test="@MERITKO_OD[.!=' ']">od 1:<xsl:value-of
  select="@MERITKO_OD"/> do 1:??</xsl:when>
</xsl:choose>
</xsl:otherwise>
</xsl:choose>
```

XSL – nový řádek



- Přidání znaku nového řádku

```
<xsl:text>&#xa;</xsl:text>
```

XSL – kopírování elementů



- Pokud chcete pouze zkopírovat element do výstupu můžete použít dvě značky
- `xsl:copy`
- `xsl:copy-of`



- Zkopíruje pouze element, bez dětí a atributů

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

```
  <xsl:copy>
```

```
    <xsl:apply-templates/>
```

```
  </xsl:copy>
```

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



- Zkopíruje element, děti i atributy

```
<xsl:template match="message">  
  <xsl:copy-of select="body" />  
</xsl:template>
```

XSL – variable



- Je možné využívat proměnné

```
<xsl:variable name="header">
  <tr>
    <th>Element</th>
    <th>Description</th>
  </tr>
</xsl:variable>
<xsl:copy-of select="$header" />
```

XSL – vytvoření elementu



- Je možné vytvářet nové XML elementy

```
<xsl:template match="/">  
  <xsl:for-each select="catalog/cd">  
    <xsl:element name="singer">  
      <xsl:value-of select="artist"  
/>  
    </xsl:element>  
  </xsl:for-each>  
</xsl:template>
```

XSL – vytvoření atributu



- Je možné vytvářet nové atributy

```
<picture>
```

```
  <xsl:attribute name="source" />
```

```
</picture>
```

```
<picture>
```

```
  <xsl:attribute name="source">
```

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

```
  </xsl:attribute>
```

```
</picture>
```


XSL – vytvoření sady atributů



- Je možné vytvářet sady atributů

```
<xsl:attribute-set name="font">  
  <xsl:attribute  
name="fname">Arial</xsl:attribute>  
  <xsl:attribute  
name="size">14px</xsl:attribute>  
  <xsl:attribute  
name="color">red</xsl:attribute>  
</xsl:attribute-set>
```



- Je možné vypisovat zprávy do systémového výstupu

```
<xsl:if test="artist=' '>  
    <xsl:message terminate="yes">  
        Error: Artist is an empty  
        string!  
    </xsl:message>  
</xsl:if>
```



- <http://www.w3schools.com/xpath/>
- <http://www.w3schools.com/xsl/>
- Papcun Petr. Šablona ISVS.
- Růžička Jan. Šablona GeoNetwork Strings.