



XSLT

Csubák Dániel

Doktorandusz

ELTE-IK, Információs Rendszerek Tanszék

BEVEZETÉS

- XML dokumentumok XHTML, illetve másik XML dokumentumokká történő átalakítására szolgál
- Az XSL része
 - EXtensible Stylesheet Language
 - HTML – CSS ~ XML – XSL
- XSLT = XSL Transformations
- XPath-t is használ



BEVEZETÉS

- Működési elv
 - XPath használatával felismeri az XML megadott részeit (amik megfelelnek az XPath-nak)
 - Találat esetén a megfelelő részekre alkalmazza a transzformációs műveleteket



PÉLDA: XML ÁTALAKÍTÁSA XHTML-LÉ

- Az XSLT is egy XML-szerű fájl fastruktúrával
- A gyökérelem
 - <xsl:stylesheet>, vagy <xsl:transform>
 - xmlns:xsl=<http://www.w3.org/1999/XSL/Transform>
hivatalos W3C névtér használata az XSLT-hez

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

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



PÉLDA: XML ÁTALAKÍTÁSA XHTML-LÉ

- Induljunk ki a cdcatalog.xml-ből
 - <http://www.w3schools.com/xsl/cdcatalog.xml>
- Készítsünk hozzá XSL stíluslapot
 - <http://www.w3schools.com/xsl/cdcatalog.xsl>
- Linkeljük össze a két fájlt
 - Ehhez XML stíluslap referenciát kell hozzáadni az XML dokumentumunkhoz
 - `<?xml-stylesheet type="text/xsl" href="cdcatalog.xsl"?>`
- Eredmény
 - http://www.w3schools.com/xsl/cdcatalog_with_xsl.xml



XSLT <XSL:TEMPLATE> ELEM

- Az XSL stíluslapok egy vagy több template-ből állnak, amik szabályhalmazokat tartalmaznak
- A szabályokat akkor alkalmazzuk, amikor a „match” rész megtalál egy XML csúcsot
 - Match ~ XPath



XSLT <XSL:TEMPLATE> ELEM

```
<?xml version="1.0" encoding="UTF-8"?>  
<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 bgcolor="#9acd32">  
          <th>Title</th>  
          <th>Artist</th>  
        </tr>  
        <tr>  
          <td>.</td>  
          <td>.</td>  
        </tr>  
      </table>  
    </body>  
  </html>  
</xsl:template>
```

```
</xsl:stylesheet>
```

[Online példa \(W3C\)](#)



XSLT <XSL:TEMPLATE> ELEM

- Az előző példa nem ad vissza tartalmat az XML-ből a kimenetbe.
- Erre fogjuk használni az **<xsl:value-of>** elemet.



XSLT <XSL:VALUE-OF> ELEM

- Arra használjuk, hogy egy kiválasztott XML node adatait el tudjuk érni, ki tudjuk bontani további felhasználásra (pl. output stream-be írásra)

```
<nz>my CD COLLECTION</nz>
<table border="1">
  <tr bgcolor="#9acd32">
    <th>Title</th>
    <th>Artist</th>
  </tr>
  <tr>
    <td><xsl:value-of select="catalog/cd/title"/></td>
    <td><xsl:value-of select="catalog/cd/artist"/></td>
  </tr>
</table>
</body>
```



XSLT <XSL:VALUE-OF> ELEM

- A „Select” szolgál a kiválasztásra
 - XPath-t használ
- Csak egy elemet ad vissza
 - Ahhoz, hogy visszacapjunk az összeset, végig kell iterálni az elemeken, erre használható az **<xsl:for-each>** elem



XSLT <XSL:FOR-EACH> ELEM

- Ciklust lehet vele létrehozni

```
<nz>my CD collection</nz>
<table border="1">
  <tr bgcolor="#9acd32">
    <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>
</html>
```



XSLT <XSL:FOR-EACH> ELEM

- Szűréseket is végezhetünk
 - Például:

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

- Ez hasonló az XPath szűréseihez, de a legális szűrési operátorok az alábbiak:
 - = (egyenlő)
 - != (nem egyenlő)
 - < (kisebb mint)
 - > (nagyobb mint)

[Online példa \(W3C\)](#)



XSLT <XSL:SORT> ELEM

- Kimenet rendezése
- Írjuk egy <xsl:for-each> elemen belülre
- A benne lévő select határozza meg, hogy mi szerint rendezünk

```
</tr>  
<xsl:for-each select="catalog/cd">  
  <xsl:sort select="artist"/>  
  <tr>  
    <td><xsl:value-of select="title"/></td>  
    <td><xsl:value-of select="artist"/></td>  
  </tr>  
</xsl:for-each>  
</table>
```

[Online példa \(W3C\)](#)



XSLT <XSL:IF> ELEM

- Amennyiben a feltétel teljesül, csináljunk valamit
- Szintaxis

```
<xsl:if test="expression">  
  ...some output if the expression is true...  
</xsl:if>
```

```
</tr>  
<xsl:for-each select="catalog/cd">  
  <xsl:if test="price > 10">  
    <tr>  
      <td><xsl:value-of select="title"/></td>  
      <td><xsl:value-of select="artist"/></td>  
      <td><xsl:value-of select="price"/></td>  
    </tr>  
  </xsl:if>  
</xsl:for-each>  
</table>
```

[Online példa \(W3C\)](#)



XSLT <XSL:CHOOSE> ELEM

- Az <xsl:when> és az <xsl:otherwise> elemekkel együtt többszörös feltételeket fogalmazhatunk meg (több irányú elágazás ~ case)
- Szintaxis

```
<xsl:choose>  
  <xsl:when test="expression">  
    ... some output ...  
  </xsl:when>  
  <xsl:otherwise>  
    ... some output ....  
  </xsl:otherwise>  
</xsl:choose>
```



XSLT <XSL:CHOOSE> ELEM

```
</tr>
<xsl:for-each select="catalog/cd">
  <tr>
    <td><xsl:value-of select="title"/></td>
    <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>
    </tr>
  </xsl:for-each>
</table>
```

[Online példa \(W3C\)](#)



XSLT <XSL:CHOOSE> ELEM

```
</tr>
<xsl:for-each select="catalog/cd">
  <tr>
    <td><xsl:value-of select="title"/></td>
    <xsl:choose>
      <xsl:when test="price > 10">
        <td bgcolor="#ff00ff">
          <xsl:value-of select="artist"/></td>
        </xsl:when>
      <xsl:when test="price > 9">
        <td bgcolor="#cccccc">
          <xsl:value-of select="artist"/></td>
        </xsl:when>
      <xsl:otherwise>
        <td><xsl:value-of select="artist"/></td>
      </xsl:otherwise>
    </xsl:choose>
  </tr>
</xsl:for-each>
</table>
```

[Online példa \(W3C\)](#)



XSLT <XSL:APPLY-TEMPLATES> ELEM

- Lényegében ezzel lehet templateket egybefűzni

```
<xsl:template match="/">
  <html>
  <body>
  <h2>My CD Collection</h2>
  <xsl:apply-templates/>
  </body>
  </html>
</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:template match="artist">
  Artist: <span style="color:#00ff00">
  <xsl:value-of select="."/></span>
  <br />
</xsl:template>
```



FELADATOK

- Feladatok
- XML
 - cdcatalog.xml
- Online XSLT
 - http://www.w3schools.com/xsl/tryxslt.asp?xmlfile=cdcatalog&xsltfile=cdcatalog_ex1
 - <http://xslttest.appspot.com/>



REFERENCIÁK

- XSLT

- <http://www.w3schools.com/xsl/default.asp>

