

eXtensible Stylesheet Language (XSL)

- XSL descreve como fazer o “display” de um documento XML.
- XSL é uma espécie de style sheets para XML ...
- ... do mesmo modo que CSS são style sheets para HTML.

XSL engloba 3 partes

1. XSLT: linguagem para transformar documentos XML.
2. XPath: linguagem para definir partes de um documento XML.
3. XSL-FO: linguagem para formatar documentos XML.

XSLT = XSL Transformations

- XSLT é usado para transformar um documento XML noutro documento XML
- Pode acrescentar ou remover novos elementos ao output.
- Pode ordenar, fazer testes condicionais, etc.

XSLT utiliza XPath

- XSLT utiliza o XPath para definir “matching patterns” para as transformações.
- Quando ocorre um “match”, o XSLT aplica uma transformação.
- As partes do documento que não fazem “match” mantêm-se inalteradas.

Browsers

- Nem todos os browsers suportam XSLT.
- Internet Explorer 6 e Netscape 7 suportam.
- Netscape 6 suporta a maioria das coisas.

Declaração do Style Sheet

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

Ou:

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

Exemplo

- Queremos transformar o seguinte documento XML ("cdcatalog.xml") para HTML.

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

Criação de um XSL Style Sheet

- Criamos um XSL style sheet (“cdcatalog.xsl”).
Exemplo:

```
<?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 bgcolor="lightgreen">  
        <th align="left">Title</th>  
        <th align="left">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>
```

... e colocamos um link no documento
XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl"
                href="cdcatalog.xsl"?>
<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>
```

XSLT utiliza regras (templates)

- O elemento `<xsl:template>` contém regras que aplica quando há um “match” com um determinado nó especificado.
- O atributo `match` associa o template com um elemento XML.
- `match="/" /` define o documento completo.

Elementos XSLT

- `<xsl:template>`
- `<xsl:value-of>`
- `<xsl:for-each>`
- `<xsl:sort>`
- `<xsl:if>`
- `<xsl:choose>`
- `<xsl:apply-templates>`

value-of

- O elemento `<xsl:value-of>` extrai o valor do nó selecionado.
- O valor do atributo `select` é uma expressão em XPath.
- Semelhante a um sistema de navegação de ficheiros.

for-each

- O elemento `<xsl:for-each>` permite fazer ciclos.
- Pode ser usado para selecionar todos os elementos XML de um determinado tipo.

Filtros

- Podemos filtrar o output de um ficheiro XML acrescentando um critério ao atributo `select` do elemento `<xsl:for-each>`.
- Exemplos de filtros:

`=` (igual)

`!=` (diferente)

`<` (menor que)

`>` (maior que)

sort

- O elemento `<xsl:sort>` permite fazer ordenações.
- O atributo `select` indica qual o elemento pelo qual se ordena.

...

```
<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>
```

...

if

- O elemento `<xsl:if>` permite fazer output condicional.
- O valor do atributo `test` contém a expressão booleana.
- Ex: Seleciona os CDs cujo preço é maior que 10.

...

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

...

choose

- O elemento `<xsl:choose>` é usado com `<xsl:when>` e com `<xsl:otherwise>` para expressar testes condicionais múltiplos.

...

```
<xsl:choose>
  <xsl:when test="price > 10">
    ... some code ...
  </xsl:when>
  <xsl:otherwise>
    ... some code ....
  </xsl:otherwise>
</xsl:choose>
```

...

apply-templates

- O elemento `<xsl:apply-templates>` aplica a regra ao nó corrente ou aos filhos do nó corrente.
- Se especificarmos o atributo `select`, a regra só é aplicada aos filhos que fazem “match” com o valor do atributo.