



# Linguagem para Web I

## XML

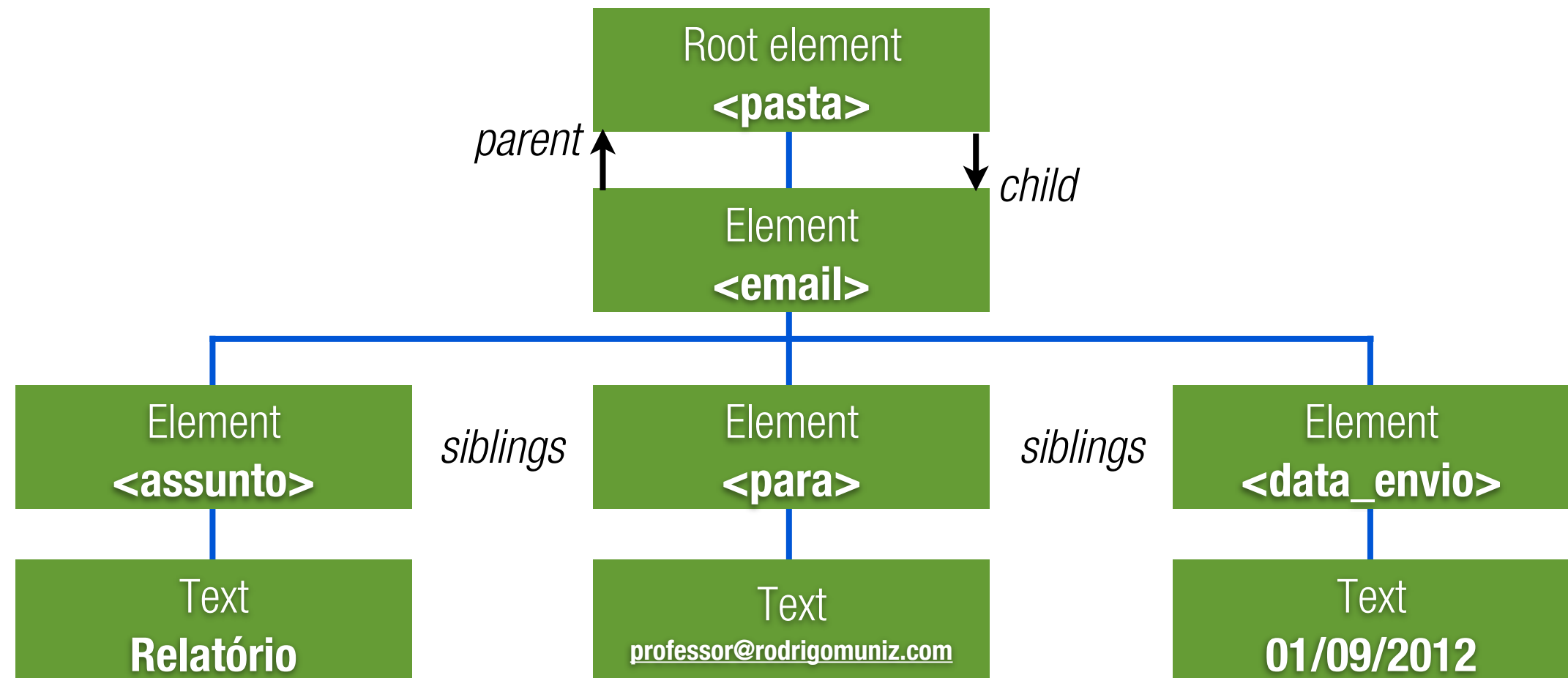
parte 2

Versão de 03.09.2012



# Árvore XML (parentesco dos nós)

```
<?xml version="1.0" encoding="utf-8"?>
<pasta>
  <email>
    <assunto>Relatório</assunto>
    <para>professor@rodrigomuniz.com</para>
    <data_envio>01/09/2012</data_envio>
  </email>
</pasta>
```





# CDATA

Já imaginou como **armazenar código** de marcação dentro de um XML?

**CDATA** serve justamente para marcar dados em um XML para que sejam considerados apenas como caracteres e não como marcação.



# CDATA

## Exemplo de CDATA em um XML

```
<?xml version="1.0" encoding="utf-8"?>
<meus_codigos>
  <fragmento>
    <![CDATA[
      <artista>Michael Jackson</artista>
    ]]>
  </fragmento>
  <fragmento>
    <![CDATA[
      <artista>The Beatles</artista>
      <faixa>Help!</faixa>
    ]]>
  </fragmento>
</meus_codigos>
```



# CDATA

Para o software interpretador é o mesmo que:

```
<?xml version="1.0" encoding="utf-8"?>
<meus_codigos>
  <fragmento>

    &lt;artista&gt;Michael Jackson&lt;/artista&gt;

  </fragmento>
  <fragmento>

    &lt;artista&gt;The Beatles&lt;/artista&gt;
    &lt;faixa&gt;Help!&lt;/faixa&gt;

  </fragmento>
</meus_codigos>
```



# Atributos XML

Consiste no par **nome/valor** que fica dentro da tag de abertura ou de uma tag vazia. O valor deve estar entre aspas duplas ou simples.

No exemplo a seguir temos os atributos **numero**, **cpf** e **moeda**:

```
<?xml version="1.0" encoding="utf-8"?>
<compras>
  <cupom_fiscal numero="1">
    <cliente cpf="001.002.003-04">Michael Jackson</cliente>
    <produto>Roda Gigante</produto>
    <valor moeda="U$" />
  </cupom_fiscal>
</compras>
```



# XSL

## Extensible Stylesheet

Folha de Estilo XML



# XSL

A W3C começou a desenvolver o XSL pela necessidade que havia de uma **folha de estilo** baseada em XML.

**XSL está para o XML assim com o CSS está para o HTML.** É uma folha de estilo para **apresentação** dos dados marcados.

A diferença é que sabemos que no HTML uma tag <table> serve para dados de uma tabela, ou seja, o browser sabe como apresentar isso. No XML a mesma tag <table> pode ser tanto uma tabela como uma marcação para o móvel (mesa), ou mesmo outra coisa totalmente diferente. Pois não há padrão pré-definido de tags, portanto o browser não sabe como exibi-los.





# XSL

É uma família de 3 recomendações para definir a **apresentação e transformação** de documentos XML.

- ✓ **XSL Transformations (XSLT)**

Uma linguagem para transformar o XML

- ✓ **XML Path Language (XPath)**

Uma linguagem usada para acessar ou referenciar (navegar) partes de um documentos XML

- ✓ **XSL Formatting Objects (XSL-FO)**

Vocabulário para especificar a formatação do XML



# XSL para que?

- ✓ **Transformar** XML em xHTML
- ✓ **Filtrar e ordenar** dados do XML
- ✓ **Definir partes** de um documento XML
- ✓ **Formatar** dados XML baseados nos valores dos dados  
Ex.: mostrar dados negativos em vermelho
- ✓ **Exportar** dados XML para mídias diferentes  
Ex.: impressão em papel, impressão em braille, leitor de telas



# XSLT (XSL Transformations)

Possibilita transformar documentos XML em xHTML ou em outros XML. É a parte mais importante do XSL.

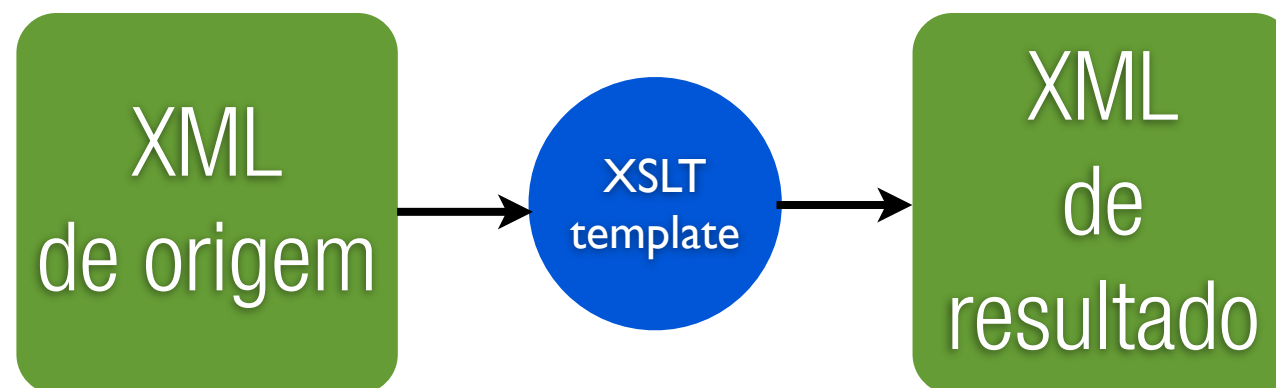
Transforma uma **árvore XML de origem** em uma **árvore XML de resultado**

- ✓ Podemos adicionar ou remover elementos no arquivo de saída
- ✓ Rearranjar e ordenar elementos



# XSLT (XSL Transformations)

XSLT usa XPath para definir partes do documento de origem que corresponde a um ou mais modelos (templates) pré-definidos. Quando um correspondente é encontrado, XSLT transforma essa parte do documento de origem em um documento de resultado.



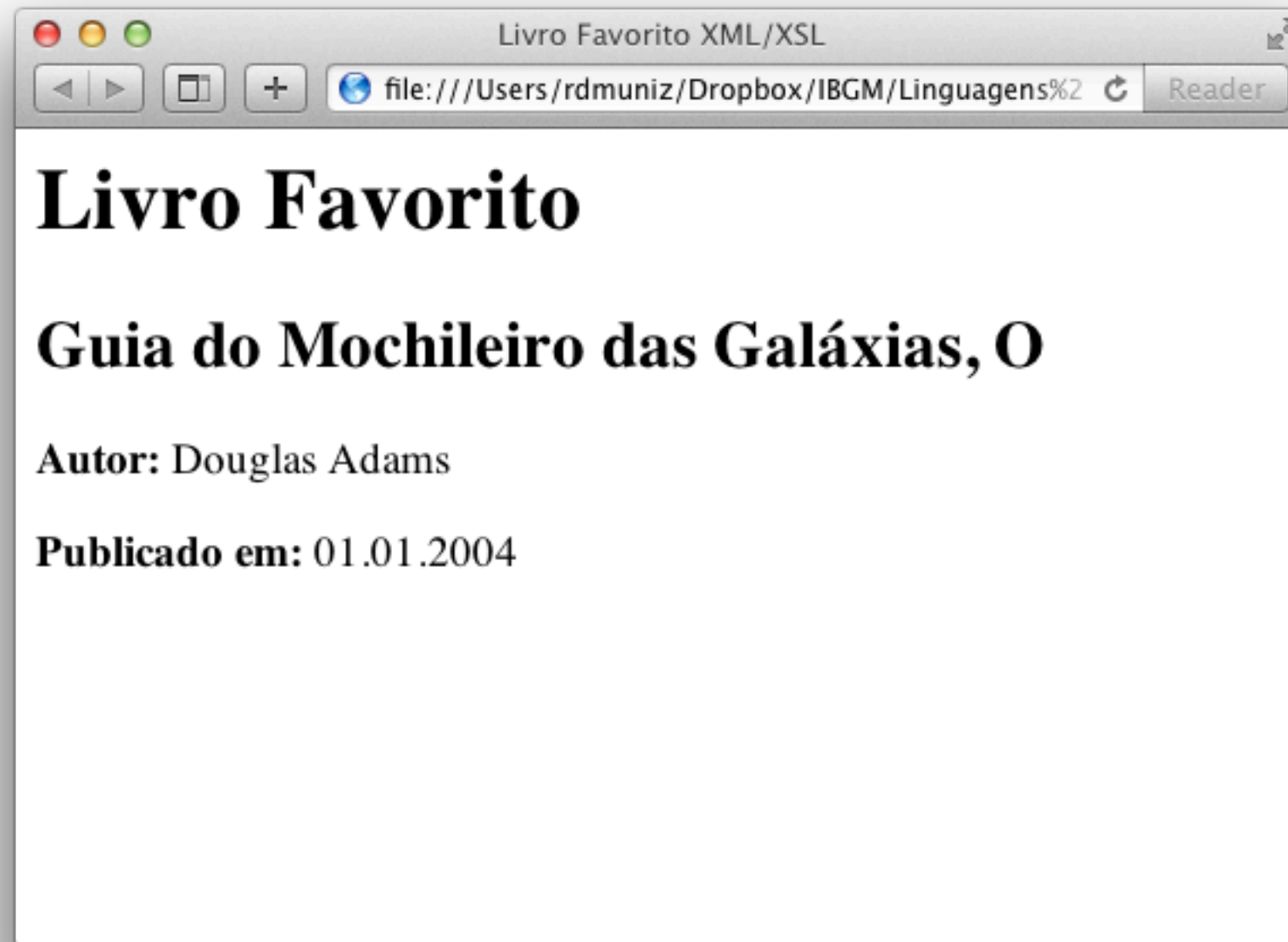


## XSLT (XSL Transformations)

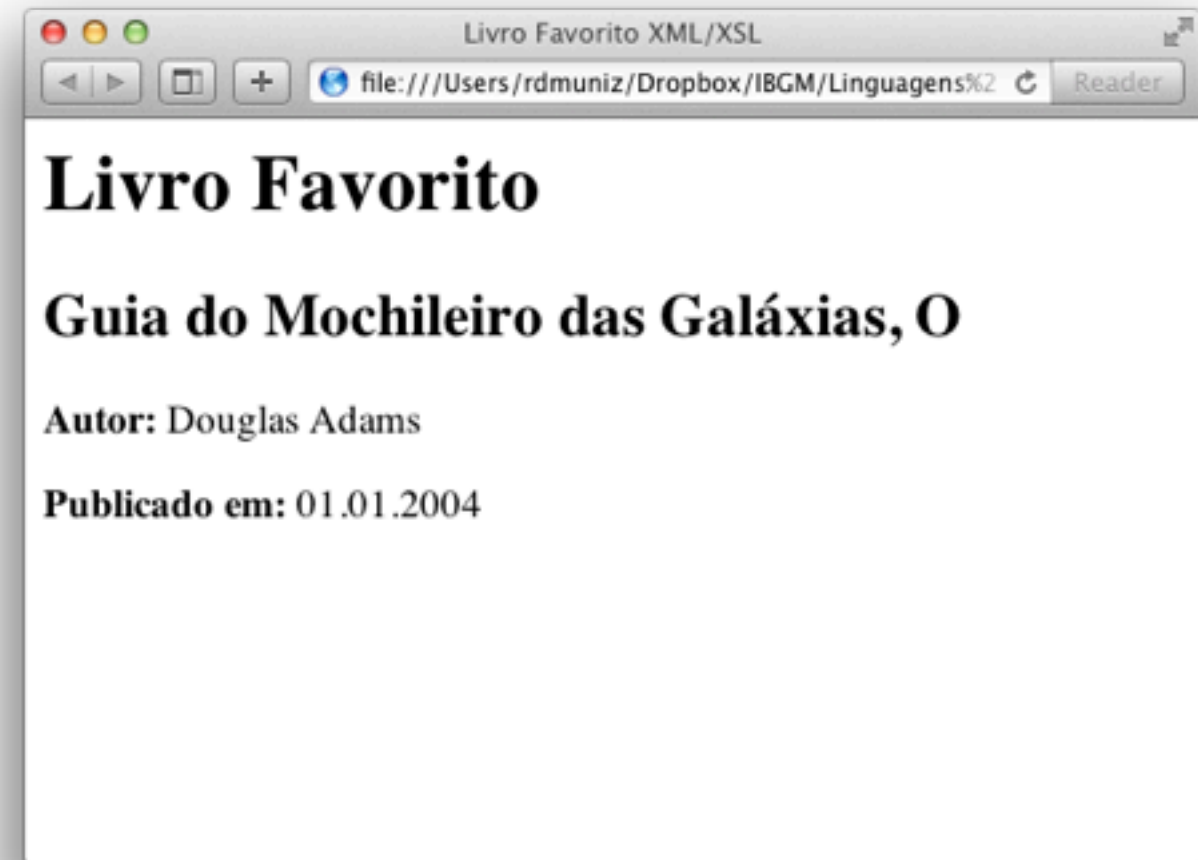
Considere o documento de origem a seguir:

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="livro.xsl"?>
<livro>
  <titulo>Guia do Mochileiro das Galáxias, O</titulo>
  <autor>Douglas Adams</autor>
  <datapub>
    <dia>01</dia>
    <mes>01</mes>
    <ano>2004</ano>
  </datapub>
</livro>
```

```
<?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>
<head><title>Livro Favorito XML/XSL</title></head>
  <body>
    <h1>Livro Favorito</h1>
    <h2><xsl:value-of select="livro/titulo"/></h2>
    <p><strong>Autor:</strong> <xsl:value-of select="livro/autor"/></p>
    <p><strong>Publicado em:</strong> <xsl:value-of select="livro/
datapub/dia"/>.<xsl:value-of select="livro/datapub/mes"/>.<xsl:value-of
select="livro/datapub/ano"/></p>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```



```
<?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>
<head><title>Livro Favorito XML/XSL</title></
head>
  <body>
    <h1>Livro Favorito</h1>
    <h2><xsl:value-of select="livro/titulo"/></h2>
    <p><strong>Autor:</strong> <xsl:value-of
select="livro/autor"/></p>
    <p><strong>Publicado em:</strong>
<xsl:value-of select="livro/datapub/dia"/
>.<xsl:value-of select="livro/datapub/mes"/
>.<xsl:value-of select="livro/datapub/ano"/></p>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```





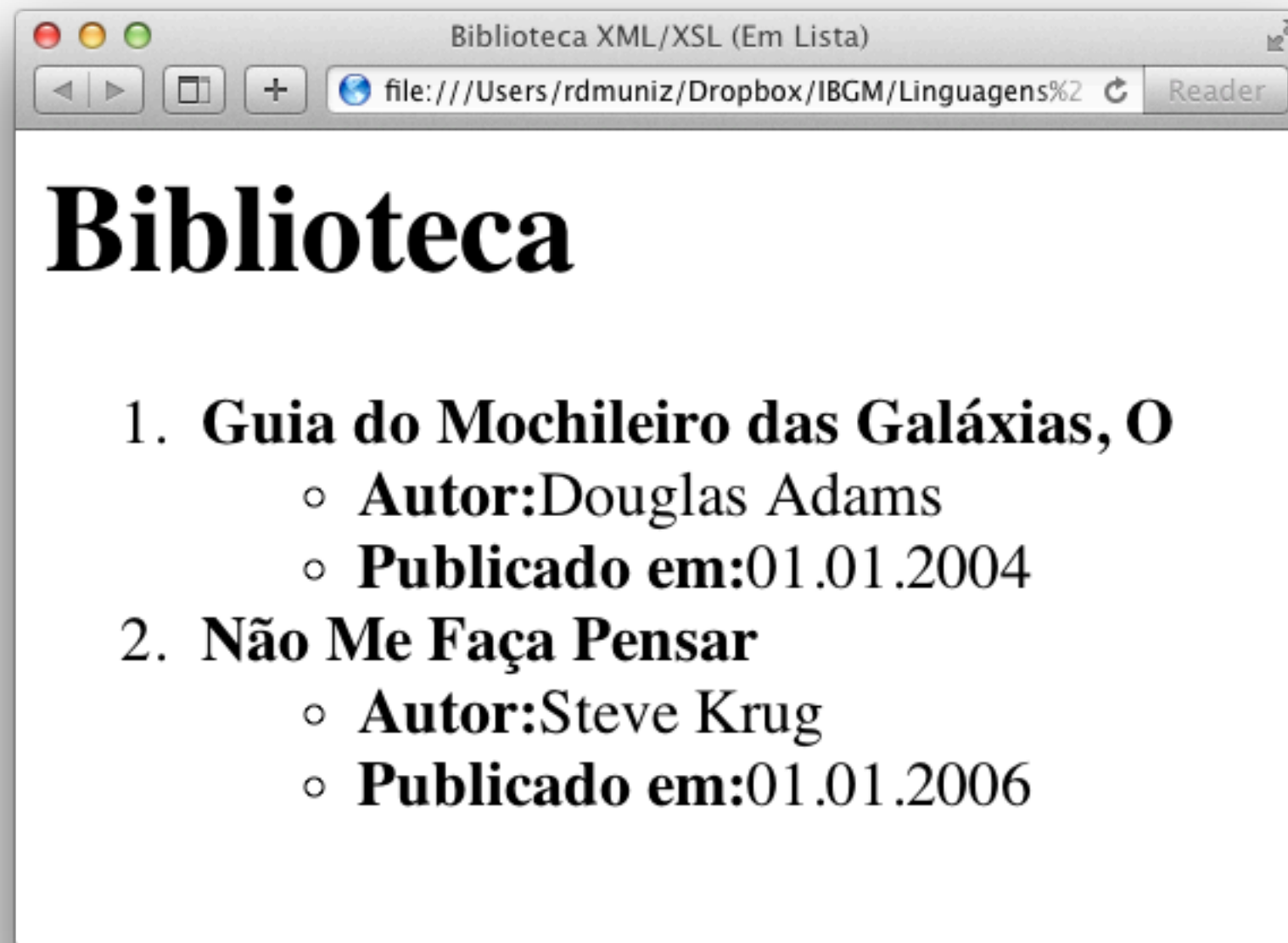


# XSLT (XSL Transformations)

Considere o documento de origem a seguir:

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="biblioteca.xsl"?>
<biblioteca>
  <livro paginas="192">
    <titulo>Guia do Mochileiro das Galáxias, O</titulo>
    <autor>Douglas Adams</autor>
    <datapub>
      <dia>01</dia>
      <mes>01</mes>
      <ano>2004</ano>
    </datapub>
  </livro>
  <livro paginas="144">
    <titulo>Não Me Faça Pensar</titulo>
    <autor>Steve Krug</autor>
    <datapub>
      <dia>01</dia>
      <mes>01</mes>
      <ano>2006</ano>
    </datapub>
  </livro>
</biblioteca>
```

```
<?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>
    <head><title>Biblioteca XML/XSL (Em Lista)</title></head>
      <body>
        <h1>Biblioteca</h1>
        <ol>
          <xsl:for-each select="biblioteca/livro">
            <li>
              <strong><xsl:value-of select="titulo"/></strong>
              <ul>
                <li><strong>Autor:</strong> <xsl:value-of select="autor"/></li>
                <li><strong>Publicado em:</strong> <xsl:value-of
select="datapub/dia"/>.<xsl:value-of select="datapub/mes"/>.<xsl:value-of
select="datapub/ano"/></li>
              </ul>
            </li>
          </xsl:for-each>
        </ol>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```





# XSLT - Elementos

<xsl:template>

<xsl:value-of>

<xsl:for-each>

<xsl:sort>

<xsl:if>

<xsl:choose>

<xsl:apply-templates>



# <xsl:template>

Com o atributo **match** é usado para associar um template com um elemento (nó) XML. O valor de **match** é uma expressão XPath.

Ex.: `match="/"` define todo o documento XML.

```
<?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>
<head><title>Livro Favorito XML/XSL</title></head>
  <body>
    <h1>Livro Favorito</h1>
    <h2><xsl:value-of select="livro/titulo"/></h2>
    <p><strong>Autor:</strong> <xsl:value-of select="livro/autor"/></p>
    <p><strong>Publicado em:</strong> <xsl:value-of select="livro/
datapub/dia"/>.<xsl:value-of select="livro/datapub/mes"/>.<xsl:value-of
select="livro/datapub/ano"/></p>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```



# <xsl:value-of>

Com o atributo **select** é usado para extrair o valor de um elemento XML e adicionar no documento de resultado.

```
<?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>
    <head><title>Livro Favorito XML/XSL</title></head>
    <body>
      <h1>Livro Favorito</h1>
      <h2><xsl:value-of select="livro/titulo"/></h2>
      <p><strong>Autor:</strong> <xsl:value-of select="livro/autor"/></p>
      <p><strong>Publicado em:</strong> <xsl:value-of select="livro/
datapub/dia"/>.<xsl:value-of select="livro/datapub/mes"/>.<xsl:value-of
select="livro/datapub/ano"/></p>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```



# Extrair o valor de atributos XML com XSL

Para acessar o valor de um atributo de um nó XML, use a mesma sintaxe do `<xsl:value-of>` porém usando um `@` na frente do valor passado no atributo **select**.

```
<?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>
    <head><title>Livro Favorito XML/XSL</title></head>
    <body>
      <h1>Livro Favorito</h1>
      <h2><xsl:value-of select="livro/titulo"/></h2>
      <p><strong>Autor:</strong> <xsl:value-of select="livro/autor"/></p>
      <p><strong>Páginas:</strong> <xsl:value-of select="livro/@paginas"/></p>
      <p><strong>Publicado em:</strong> <xsl:value-of select="livro/datapub/dia"/>.<xsl:value-of select="livro/datapub/mes"/>.<xsl:value-of select="livro/datapub/ano"/></p>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```



# <xsl:for-each>

É usado para selecionar **cada elemento** de um nó XML específico.

```
<?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>
    <head><title>Biblioteca XML/XSL (Em Lista)</title></head>
    <body>
      <h1>Biblioteca</h1>
      <ol>
        <xsl:for-each select="biblioteca/livro">
          <li>
            <strong><xsl:value-of select="titulo"/></strong>
            <ul>
              <li><strong>Autor:</strong> <xsl:value-of select="autor"/></li>
              <li><strong>Publicado em:</strong> <xsl:value-of select="datapub/dia"/>
                <xsl:value-of select="datapub/mes"/>.<xsl:value-of select="datapub/ano"/></li>
            </ul>
          </li>
        </xsl:for-each>
      </ol>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```





# <xsl:sort>

Usado para ordenar o resultado baseado em um elemento indicado no atributo **select**.

```
<?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>
      <head><title>Biblioteca XML/XSL (Em Lista)</title></head>
      <body>
        <h1>Biblioteca</h1>
        <ol>
          <xsl:for-each select="biblioteca/livro">
            <xsl:sort select="autor" />
            <li>
              <strong><xsl:value-of select="titulo"/></strong>
              <ul>
                <li><strong>Autor:</strong> <xsl:value-of select="autor"/></li>
                <li><strong>Publicado em:</strong> <xsl:value-of select="datapub/dia"/>
                  <xsl:value-of select="datapub/mes"/>.<xsl:value-of select="datapub/ano"/></li>
              </ul>
            </li>
          </xsl:for-each>
        </ol>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```



# <xsl:if>

Usado para colocar uma condicional teste contra o conteúdo do XML de origem.

```
<?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>
    <head><title>Biblioteca XML/XSL (Em Lista)</title></head>
    <body>
      <h1>Biblioteca</h1>
      <ol>
        <xsl:for-each select="biblioteca/livro">
          <xsl:if test="datapub/ano > 2005">
            <li>
              <strong><xsl:value-of select="titulo"/></strong>
              <ul>
                <li><strong>Autor: </strong> <xsl:value-of select="autor"/></li>
                <li><strong>Publicado em: </strong> <xsl:value-of select="datapub/dia"/>
                  <xsl:value-of select="datapub/mes"/>.<xsl:value-of select="datapub/ano"/></li>
              </ul>
            </li>
          </xsl:if>
        </xsl:for-each>
      </ol>
    </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```



# <xsl:choose>

Usado em conjunto com **<xsl:when>** e **<xsl:otherwise>** para expressar múltiplas condicionais testes.

```
<?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>
    <head><title>Biblioteca XML/XSL (Em Lista)</title></head>
    <body>
      <h1>Biblioteca</h1>
      <ol>
        <xsl:for-each select="biblioteca/livro">
          <xsl:choose>
            <xsl:when test="datapub/ano > 2005">
              <li>
                <strong><xsl:value-of select="titulo"/></strong>
                <ul>
                  <li><strong>Autor: </strong> <xsl:value-of select="autor"/></li>
                  <li><strong>Publicado em: </strong> <xsl:value-of select="datapub/dia"/>.<xsl:value-of
select="datapub/mes"/>.<xsl:value-of select="datapub/ano"/></li>
                </ul>
              </li>
            </xsl:when>
            <xsl:otherwise>
              <li>
                <strong><xsl:value-of select="titulo"/></strong>
              </li>
            </xsl:otherwise>
          </xsl:choose>
        </xsl:for-each>
      </ol>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```



# <xsl:apply-templates>

Usado para aplicar template em um elementos atual ou nos filhos deste elemento.

```
<?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>
    <head><title>Biblioteca XML/XSL</title></head>
    <body>
      <h1>Biblioteca</h1>
      <xsl:apply-templates />
    </body>
  </html>
</xsl:template>

  <xsl:template match="biblioteca/livro">
    <xsl:apply-templates select="titulo" />
    <xsl:apply-templates select="autor" />
  </xsl:template>

  <xsl:template match="titulo">
    <h2><xsl:value-of select="." /></h2>
  </xsl:template>

  <xsl:template match="autor">
    <p>Autor: <xsl:value-of select="." /></p>
  </xsl:template>
</xsl:stylesheet>
```



# XSLT - Revisando

## **<xsl:template>**

Com o atributo **match** é usado para associar um template com um elemento (nó) XML. O valor de **match** é uma expressão XPath.

Ex.: `match="/"` define todo o documento XML.

## **<xsl:value-of>**

Com o atributo **select** é usado para extrair o valor de um elemento XML e adicionar no documento de resultado.

## **<xsl:for-each>**

É usado para selecionar **cada elemento** de um nó XML específico.



# XSLT - Revisando

## **<xsl:sort>**

Usado para ordenar o resultado baseado em um elemento indicado no atributo **select**.

## **<xsl:if>**

Usado para colocar uma condicional teste contra o conteúdo do XML de origem.

## **<xsl:choose>**

Usado em conjunto com **<xsl:when>** e **<xsl:otherwise>** para expressar múltiplas condicionais testes.

## **<xsl:apply-templates>**

Usado para aplicar template em um elementos atual ou nos filhos deste elemento.



## Exercício para casa

Transforme seus XML's dos papéis dos exercícios anteriores em arquivos .xml digitais e crie uma folha de estilo XSL (.xsl) para cada um deles. Sinta-se livre para escolher a melhor forma de exibir seus dados (parágrafos, listas, tabelas). Mas seja organizado e aplique cada elemento que vimos nos slides dessa aula (template, value-of, for-each, sort, if, choose e apply:templates).

### Formato:

- Dupla
- Testar no Mozilla Firefox 15.0
- Arquivo **ZIP** contendo os arquivos .xml e .xsl
- Não esqueça o nome dos membros do grupo e turma no corpo do email

### Envio:

- Email com arquivo **ZIP** enviado como **anexo**
- Para: professor@rodrigomuniz.com
- **Assunto do email:**  
*[Web LPW] Exercício XSL*

Experimente usar o editor de XML/XSL online:  
<http://www.purplegene.com/static/transform.html>

### PRAZO

<http://rodrigomuniz.github.com/lpw2012.2/#calendario>  
Não esqueça seu nome e turma no email.



# Referências

**Extensible Markup Language (XML).** Disponível em <http://www.w3.org/XML/Schema> em 2012.

**XSLT Tutorial.** Disponível em <http://www.w3schools.com/xsl/default.asp> em 2012.