

10

XML

Fundamentos de XSL-FO

Helder da Rocha

(helder@argonavis.com.br)

XSL Formatting Objects

- *Aplicação XML para descrever o layout preciso de texto e imagens em uma página*
- *Elementos representam*
 - *Regras para formatação de várias páginas*
 - *Layout de páginas individuais, margens, rodapés*
 - *Hifenação, alinhamento, fontes, cores, leading*
 - *Imagens, gráficos, tabelas, listas, links*
- *Isoladamente, XSL-FO não é folha de estilo*
 - *É formato final, com estrutura que visa a apresentação (como XHTML) – linguagem de descrição de página (como PDF ou PostScript)*
 - *Típicamente, se cria um XSLT que gera FO a partir de uma fonte XML*



Hello World

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
```

```
<fo:layout-master-set>
```

```
<fo:simple-page-master master-name="p1">
```

```
<fo:region-body/>
```

```
</fo:simple-page-master>
```

```
</fo:layout-master-set>
```

```
<fo:page-sequence master-name="p1">
```

```
<fo:flow flow-name="xsl-region-body">
```

```
<fo:block color="blue" font-size="20pt">Hello!</fo:block>
```

```
<fo:flow>
```

```
</fo:page-sequence>
```

```
</fo:root>
```

Este é o "<head>"
do XSL-FO

Ligação entre as
regras de layout e
o conteúdo afetado

Este é o "<body>"
do XSL-FO

- *Raiz do documento XSL-FO*

- *Define o namespace*

- <http://www.w3.org/1999/XSL/Format>

- *Uso típico*

```
<fo:root
  xmlns:fo="http://www.w3.org/1999/XSL/Format">
  ...
</fo:root>
```

- *Normalmente, documentos FO são gerados com XSLT*

```
<xsl:template match="/">
  <fo:root
    xmlns:fo="http://www.w3.org/1999/XSL/Format">
    <xsl:apply-templates />
  </fo:root>
</xsl:template>
```



Estrutura do documento

- O elemento **<root>** deve conter:
- 1) Um **<layout-master-set>**
 - contém "mestres de layout": templates reutilizáveis para layout de páginas
 - cada mestre possui um identificador que é usado para associá-lo a um bloco de conteúdo
 - o tipo de mestre mais simples é **<simple-page-master>**
- 2) Pelo menos um **<page-sequence>**
 - onde fica o conteúdo
 - define uma seqüência de páginas em um documento
 - cada **<page-sequence>** deve estar associado a um mestre de layout existente
 - contém objetos **<flow>** e/ou **<static-content>**



- **Cabeçalho mínimo**

```
<fo:layout-master-set>  
  <fo:simple-page-master master-name="p1">  
    <fo:region-body/>  
  </fo:simple-page-master>  
</fo:layout-master-set>
```

- **<simple-page-master>** define margens, altura e largura da página e contém pelo menos uma região

- *master-name* do elemento é ID para que elementos de conteúdo **<page-sequence>** possam referenciá-lo

```
<fo:simple-page-master master-name="p1"  
  margin-top="1.5cm"   margin-bottom="2cm"  
  margin-left="2.5cm"  margin-right="1.5cm"  
  page-width="21cm"    page-height="29.7cm">  
  ... regiões afetadas ...  
</fo:simple-page-master>
```

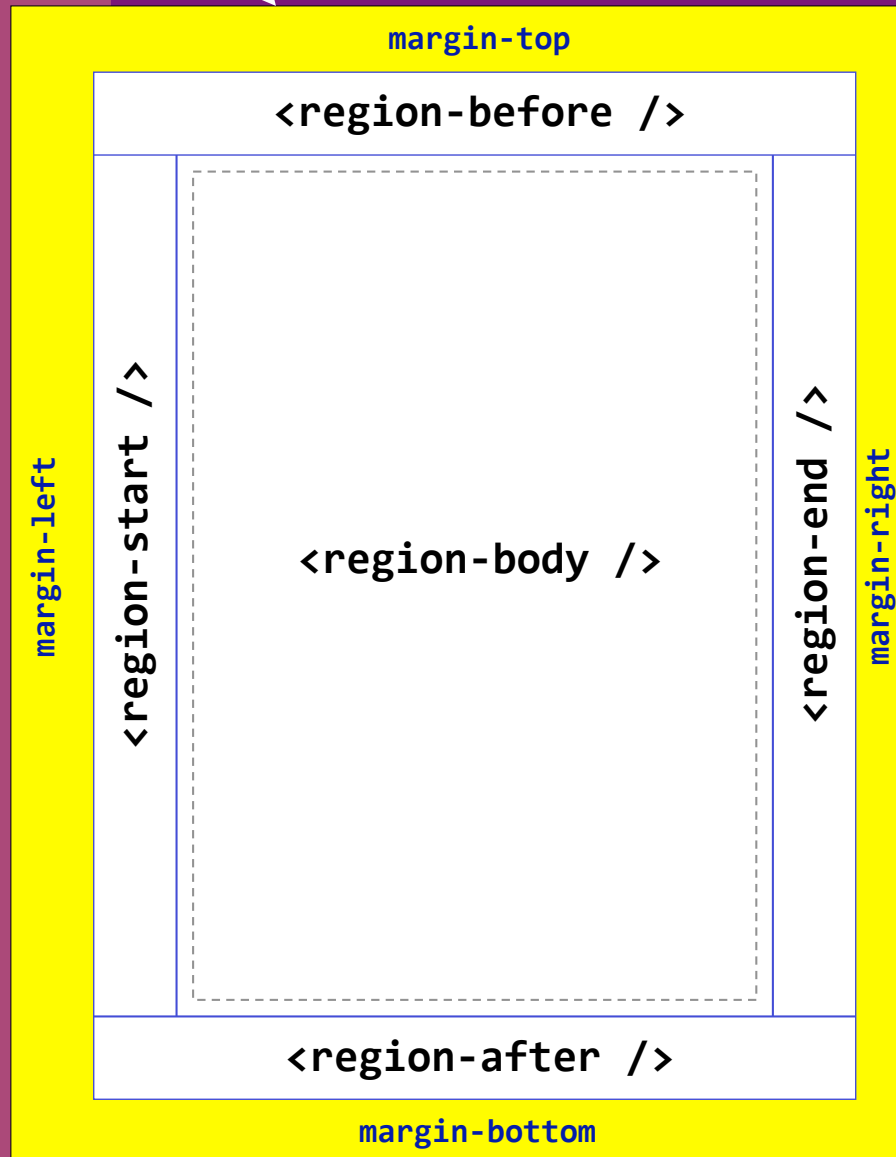


- **<region-body />**
 - *obrigatória*
 - *área de conteúdo principal da página*
 - *define um ID implícito (fixo): **xsl-region-body**, para uso por elementos de conteúdo de fluxo ou estático.*
- **<region-start />** e **<region-end />**
 - *Em documentos de linguagem ocidental, **start** é a margem esquerda e **end** é margem direita.*
 - *IDs: **xsl-region-start** e **xsl-region-end***
- **<region-before />** e **<region-after />**
 - ***before** é a margem superior, **after** é a margem inferior.*
 - *IDs: **xsl-region-before** e **xsl-region-after**.*



Margens e regiões

`<simple-page-master>`



- Margens do `<simple-page-master>` limitam a área de impressão
 - `<region-body>` ocupa todo o espaço restante
- Outras regiões opcionais podem ser definidas sobre `<region-body>`
 - `<region-before extent="2cm"/>`
 - `<region-end extent="1cm"/>`
- Margens de `<region-body>` devem ser iguais ou maiores que os **extent** das regiões marginais
 - `<region-body margin-left="2.7cm" margin-right="1.2cm" margin-bottom="2.1cm" margin-top="2.2cm" />`



- *Layout geralmente permanece fixo em folha XSLT+FO*

```
<xsl:template match="/">
  <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
    <fo:layout-master-set>
      <fo:simple-page-master master-name="p1">
        <fo:region-before extent="2cm"/>
        <fo:region-end extent="1cm"/>
        <fo:region-after extent="1.9cm"/>
        <fo:region-start extent="2.5cm"/>
        <fo:region-body margin-left="2.7cm"
                          margin-right="1.2cm"
                          margin-bottom="2.1cm"
                          margin-top="2.2cm" />
      </fo:simple-page-master>
    </fo:layout-master-set>
    <fo:page-sequence master-name="p1">
      <xsl:apply-templates />
    </fo:page-sequence>
  </fo:root>
</xsl:template>
```



- O conteúdo do documento pode estar
 - Em seções **<flow>**, que criam nova página quando o texto preenche o espaço disponível na atual
 - Em seções **<static-content>**, que são **repetidos** em todas as páginas da seqüência.
- **Regiões**
 - Cada bloco de conteúdo deve estar associado a uma região existente através do atributo **flow-name**:
`<fo:static-content flow-name="xsl-region-start"> ...`
`<fo:flow flow-name="xsl-region-body"> ...`
- **Blocos**
 - Dentro de **<flow>** ou **<static-content>**, conteúdo deve estar em blocos: **<block>**, **<list-block>**, **<table>**, etc.



Blocos de conteúdo

```
<fo:page-sequence master-name="p1" font-size="8pt">
```

```
  <fo:static-content flow-name="xsl-region-after">
```

```
    <fo:block>
```

```
      <fo:leader leader-pattern="rule"
        leader-length="16cm" />
```

Este bloco de conteúdo estático está associado com region-after

```
    </fo:block>
```

```
      <fo:block text-align="end">página
```

```
        <fo:page-number/>
```

```
      </fo:block>
```

```
    </fo:block>
```

```
  </fo:static-content>
```

Este bloco de conteúdo de fluxo está associado com region-body

```
  <fo:flow flow-name="xsl-region-body">
```

```
    <fo:block font-family="serif" font-size="11pt">
```

```
      Texto que pode atravessar páginas.
```

```
    </fo:block>
```

```
  </fo:flow>
```

```
</fo:page-sequence>
```



- Os elementos **<flow>** e **<static-content>** devem conter elementos de bloco
 - **<block>** Bloco de propósito geral
 - **<list-block>** Bloco que contém itens de lista
 - **<table>** Bloco que contém tabela
- Blocos podem ter conteúdo misto, contendo texto e elementos
 - **<inline>** Trecho de propósito geral
 - **<external-graphic>** Imagem ou gráfico externo
 - **<page-number>** Número de página
 - **<basic-link>** Referência de hipertexto
 - **<instream-foreign-object>** XML embutido (ex: SVG)
 - **<leader>** Linhas e outros decoradores



Propriedades

- Propriedades de estilo podem ser aplicadas em **<flow>**, **<static-content>** e em quaisquer elementos descendentes
- Propriedades são atributos XML
- Maior parte das propriedades são iguais e têm mesma sintaxe que as propriedades do CSS

```
<fo:block font-family="tahoma, sans-serif"
          font-size="12pt"
          color="rgb(255, 45, 45)"
          margin-top="1cm"
          font-weight="bold"
          text-align="center">Uma <fo:inline
color="blue">linha</fo:inline> de texto </fo:block>
```

- Relação com CSS
 - Algumas **propriedades CSS** são **elementos** em XSL-FO
 - Alguns **seletores CSS** são **atributos** em XSL-FO



<table>

```
<fo:table>
  <fo:table-header>
    <fo:table-cell>
      <fo:block font-family="tahoma, sans"
                font-weight="bold">Coluna 1</fo:block>
    </fo:table-cell>
    <fo:table-cell>
      <fo:block font-family="tahoma, sans"
                font-weight="bold">Coluna 2</fo:block>
    </fo:table-cell>
  </fo:table-header>

  <fo:table-body>
    <fo:table-row>
      <fo:table-cell>
        <fo:block font-family="monospace">célula 1.1</fo:block>
      </fo:table-cell>
      <fo:table-cell>
        <fo:block font-family="monospace">célula 1.2</fo:block>
      </fo:table-cell>
    </fo:table-row>
    ...
    <fo:table-row> ... </fo:table-row>
  </fo:table-body>
</fo:table>
```

<list-block>

```
<fo:list-block>
  <fo:list-item>
    <fo:list-item-label>
      <fo:block font-size="20pt" color="red">
        &#x2022;
      </fo:block>
    </fo:list-item-label>

    <fo:list-item-body start-indent="70pt">
      <fo:block>
        Item um
      </fo:block>
    </fo:list-item-body>
  </fo:list-item>
  <fo:list-item> ... </fo:list-item>
  ...
</fo:list-block>
```



<external-graphic>

- *Permite incluir uma imagem na página*

- *Objeto inline*

- <fo:block>

- <fo:external-graphic src="moe.gif"/>

- </fo:block>

- <fo:block>

- <fo:external-graphic

- src="http://www.x.com/moe.gif"/>

- </fo:block>



- *Vínculo de hipertexto*
- *Vincula uma referência de ID a um ID*
 - *Geralmente, quando o documento é gerado via XSLT, o ID é obtido do elemento ao qual se quer vincular ou via alguma variável*
 - *Exemplo (geração de link usando XSLT):*

```
<fo:basic-link internal-destination="{@id}"
                text-decoration="underline"
                color="blue">
    <xsl:value-of select="@codigo"/>
</fo:basic-link>
```



<instream-foreign-object>

- *Permite embutir outra linguagem XML na página*

```
<fo:instream-foreign-object>
```

```
<svg:svg
```

```
  xmlns:svg="http://www.w3.org/2000/svg"
```

```
  width="300" height="300"
```

```
  viewBox="0 0 300 300">
```

```
  <svg:circle style="fill: green"
```

```
    r="50" cx="150" cy="150" />
```

```
</svg:svg>
```

```
</fo:instream-foreign-object>
```



- *XSL-FO é uma linguagem de descrição de página*
- *Páginas podem ser geradas por um programa que construa o documento na memória (como faz o browser com HTML e CSS) ou que gere um arquivo*
 - *PostScript, PDF: Apache FOP, RenderX XEP, REXP*
 - *RTF (MS-Word): JFOR, RenderX*
 - *Visualização em browser: IBM XFC*
- *Ferramentas para desenvolver FO*
 - *HP FOA*
- *Exemplos usados neste curso (use scripts .bat em c:\xml ou .sh em /cursoxml00/software/scripts)*
- *1) Para rodar o FOP (gerar PDF a partir de FO)*
 - *fop arquivo.fo arquivo.pdf*
- *2) Para rodar o JFOR (gerar RTF a partir de FO)*
 - *jfor arquivo.fo arquivo.doc*

