

## XSL – eXtensible Stylesheet Language

Nós já vimos que as tags usadas em um documento XML podem ser criadas pelo usuário, então como o Navegador irá interpretar essas tags , já que não existem tags padrão em XML ?

Para exibir documentos XML será necessário um mecanismo que descreva como o documento será exibido. Esse mecanismo chama-se XSL – **Extensible Stylesheet Language** – e, é usado para transformar XML em HTML.

A XSL pode ser encarada como um linguagem que pode transformar XML em HTML, pode filtrar e ordenar dados em documentos XML e que pode formatar dados XML.

Ela pode ser usada para definir em como um arquivo XML será exibido pela transformação em um arquivo reconhecido pelo navegador do usuário.

Geralmente este serviço é feito pela transformação de cada elemento XML em um elemento HTML, sendo que a XSL pode incluir novos elementos, remover elementos, rearranjar e ordenar elementos e ainda testar e tomar decisões.

## Resumindo...

- XSL são as folhas de estilo do XML
- XSL descreve como o documento XML deve ser apresentado
- Recomendação W3C
- Os arquivos XSL são na verdade arquivos XML que utilizam o namespace específico do XSL, com o prefixo "**xsl**"
- O elemento raiz do XSL é representado por `<xsl:stylesheet>`
- É necessário incluir uma referência ao arquivo XSL no seu documento XML

***<?xml-stylesheet type="text/xsl" href="arquivo.xsl"?>***

## Elementos em XSL

### 1) Elemento `<xsl:template match="XPath">` (tem tag de fechamento)

- É um elemento que define um “molde”, ou “padrão” em conjunto com o atributo ***match***
  - O valor do atributo `match` é uma expressão XPath válida
  - Uma expressão XPath funciona como navegar em um *filesystem*, no qual uma barra / seleciona os subdiretórios
- No exemplo, `<xsl:template match="/">` associa o molde ao elemento raiz do documento XML
- O conteúdo dentro da tag `<xsl:template>` é código em XHTML que define como será exibido o conteúdo do arquivo XML no browser

## 2) Elemento `<xsl:value-of select="XPath">` (o fechamento é feito na própria tag)

- É utilizado para extrair o valor de um nó específico dentro do arquivo XML
- O valor do atributo *select* deve ser uma expressão XPath válida
- No nosso exemplo o elemento `<xsl:value-of select="dia"/>` extrai o valor do nó *dia*

## 3) Elemento `<xsl:for-each select="XPath">` (tem tag de fechamento)

- Utilizado para criar loops
- Permite extrair todos os elementos quando utilizado em conjunto com o *xsl:value-of*
- O valor do atributo *select* deve ser uma expressão XPath válida

`<xsl:for-each select="combustível/abastecimento">`

•  
•  
•

`</xsl:for-each>`

O elemento `<xsl:for-each>` pode ser utilizado para filtrar o resultado com o uso dos operadores lógicos:

|      |           |
|------|-----------|
| =    | Igual     |
| !=   | Não igual |
| &lt; | Menor que |
| &gt; | Maior que |

## Exemplo:

```
<xsl:for-each select="combustível/abastecimento[dia='22']">
```

## 4) Elemento `<xsl:sort select="XPath">` (não tem tag de fechamento)

- Serve para ordenar os resultados
- É colocado dentro de um elemento `<xsl:for-each>`

### Exemplo:

```
<xsl:sort select="valor" />
```



## 5) Elemento `<xsl:IF test="expressão de teste">` (tem tag de fechamento)

- Faz testes condicionais
- Deve ser colocado dentro de um elemento `<xsl:for-each>`

**Exemplo:** Para apresentar os registros que o elemento `<valor>` seja maior que 100.

```
<xsl:if test="valor > 100">
```

```
  .  
  .  
  .
```

```
</xsl:if>
```

## 6) Elementos `<xsl:choose>`, `<xsl:when>`, `<xsl:otherwise>` (tem tag de fechamento)

- Utilizados para fazer múltiplos testes condicionais

```
<xsl:choose>
  <xsl:when test="expression">
    ... algum código ...
  </xsl:when>
  <xsl:otherwise>
    ... algum código ....
  </xsl:otherwise>
</xsl:choose>
```