



XML & Cia: Introdução (CSS e XSL)

Renata Pontin de Mattos Fortes

Instituto de Ciências Matemáticas e de Computação

Universidade de São Paulo

{renata}@icmc.usp.br

Agradecimento especial



Prof.Dr.Maria da Graça Pimentel

e Prof. Luciano T. E. Pansanato

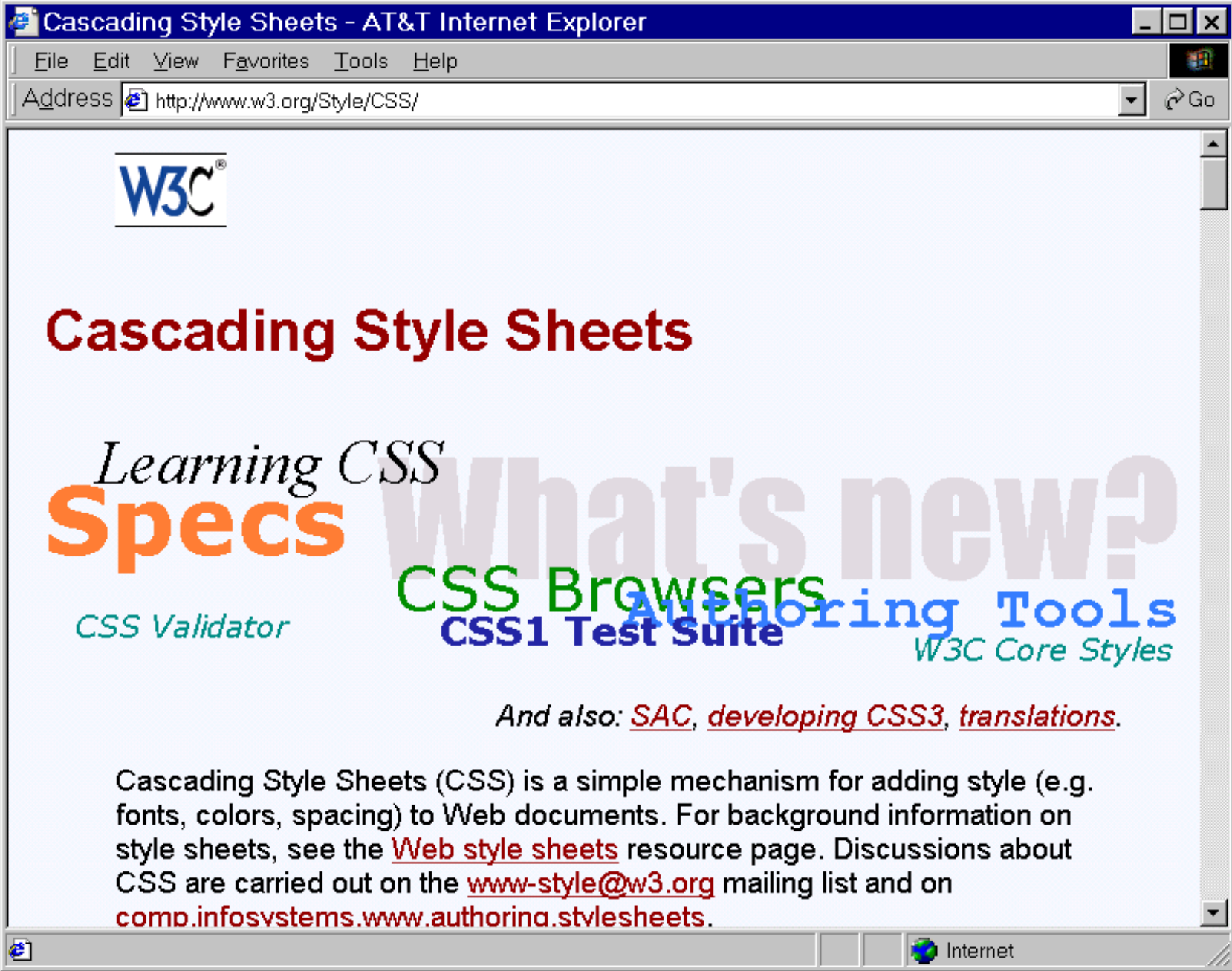
pelo material cedido

Linguagens de *Folhas de estilo*

- Permitem que descrições de estilo de apresentação de documentos sejam separadas da representação do conteúdo dos documentos
- Para um mesmo conteúdo podem ser especificadas diferentes apresentações
 - um documento pode, portanto, ser aproveitado em diferentes aplicações que exigem diferentes visões do mesmo
- **O princípio das folha de estilo está em:**
 - prover uma sintaxe para identificar partes específicas do conteúdo de documentos e
 - um conjunto de ações que devem ser realizadas sobre as partes identificadas.

CSS1: Cascading Style Sheets nível 1

- se aplica tanto a documentos XML como a documentos HTML
- cada regra especifica os nomes dos elementos a que se aplica e o estilo a aplicar
- associação de uma folha de estilo a um documento XML- instrução de processamento que referencia a folha de estilo a ser aplicada
 - **`<?xml-stylesheet type="text/css" href="aula.css"?>`**



Aplicativos Locais Sistema ?

Cascading Style Sheets - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

http://www.w3.org/Style/CSS/ rich in style

Most Visited Getting Started Latest Headlines icmc [DokRat] USP :: Sistemas USP:: https://192.168.231... Gmail: Google Tidia-Ae CoteiaWiki

RichInStyle.com RichInStyle.com CSS2 tu... Tidia-Ae : My Workspac... Tidia-Ae : SCC5811_Ver2... Cascading Style Sheets

W3C®

(Languages: AF AR AZ BE BG BN CA CS DA DE EL EN ES ET FA FI FR HE HI HR HU ID IS IT JA KA KK KO KY LT LV MK MO MT NB NL PL PT RO RU SK SQ SR-CYRL SV TG TH TL TR TT UK UZ VI ZH-HANS ZH-HANT)

Cascading Style Sheets

home page


What is CSS?

Cascading Style Sheets (CSS) is a simple mechanism for adding style (e.g., fonts, colors, spacing) to Web documents.

These pages contain information on how to **learn and use CSS** and on available **software**. They also contain news from the **CSS working group**.

News

Joining the discussion



A part of the CSS WG in November 2010. (Photo: Bert Bos / Anthony Grasso)

More »

Concluído

Cascading Style Sheet... re_aula3a_xml_css_x... [aulas2011_scc5811 ... re_aula4_prog_DOM.o...

zotero

Aplicativos Locais Sistema ?

How to add style to XML - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

http://www.w3.org/Style/styling-XML

Most Visited Getting Started Latest Headlines icmc [DokRat] USP:: Sistemas USP:: https://192.168.231... Gmail: Google Tidia-Ae CoteiaWiki

RichInStyle.com RichInStyle.com CS... Tidia-Ae : My Works... Tidia-Ae : SCC5811_... How to add style to... w3c stylebot css - P...

W3C®

[Activities](#) • [Tech. reports](#) • [Translations](#) • [Software](#) • [Site index](#) • [Search](#) •

(Languages: [be](#) [cs](#) [de](#) [en](#) [es](#) [hi](#) [id](#) [ja](#) [ky](#) [mk](#) [pl](#) [ru-1](#) [ru-2](#) [sr-cryl](#) [sr-latn](#) [th](#) [tl](#) [tr](#) [uk](#) [zh-hans](#) [zh-hant](#))

HOW TO ADD STYLE TO XML

The [CSS2 Recommendation](#) contains a brief tutorial on using CSS with XML (see [§ 2.2](#)). At the time CSS2 was written the official [specification for style in XML](#) wasn't ready. Here is what that tutorial should have contained. *Note that the examples use [CSS](#), but in most cases the style rules can also be written in [XSL](#).*

EXTERNAL STYLE SHEETS

Local links

- Style
- Which should I use?
- external
- CSS
- embedded
- XSL

Concluído

How to add style to X... re_aula3a_xml_css_x... aulas2011_scc5811 -... re_aula4_prog_DOM.o... [Área de Trabalho - Na... zotero

Aplicativos Locais Sistema ?

re_aulaHow to add style to XML - Mozilla Firefox press

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

http://www.w3.org/Style/styling-XML w3c stylebot css

Most Visited Getting Started Latest Headlines icmc [DokRat] USP: Sistemas USP: https://192.168.231... Gmail: Google Tidia-Ae CoteiaWiki

RichInStyle.com

Código-fonte: http://www.w3.org/Style/styling-XML - Mozilla Firefox

Arquivo Editar Exibir Ajuda

```
<!doctype html public "-//W3C//DTD HTML 4.01//EN"
'http://www.w3.org/TR/html4/strict.dtd'>

<html lang=en>
<title>How to add style to XML</title>

<style type="text/css">
pre { font-family: monospace; margin-left: 1em; color: maroon }
.note { color: #505050 }
.note { margin: 1em }
.demo { padding: 0.3em; border: solid thin; text-align: center }
.demo { float: right; width: 5em }
.aside { font-size: smaller }
.aside ol { margin-left: 0; list-style-position: inside }
</style>
<meta name="GENERATOR" content="amaya V2.4">
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<link rel="alternate stylesheet" title="Blue shadows" href="main.css">
<link rel="alternate stylesheet" title="Blue shadows" href="map-ns.css">
<link rel="alternate stylesheet" title="Blue shadows" href="map.css" media="all">
<!-- map.css is better than map-ns, but must be hidden from NS 4.x -->
<link rel="alternate stylesheet" title="Blue shadows" href="banner-k.css">
<link rel="alternate stylesheet" title="Blue shadows" href="shadows.css" media="all">

<link rel="alternate stylesheet" title="Without navigation bar" href="main.css">
<link rel="alternate stylesheet" title="Without navigation bar" href="map-ns.css">
```

W3C®

(Languages: b

HOW TO

The CSS2 Rec
the official spec
examples use

EXTERNAL STYLE SHEETS

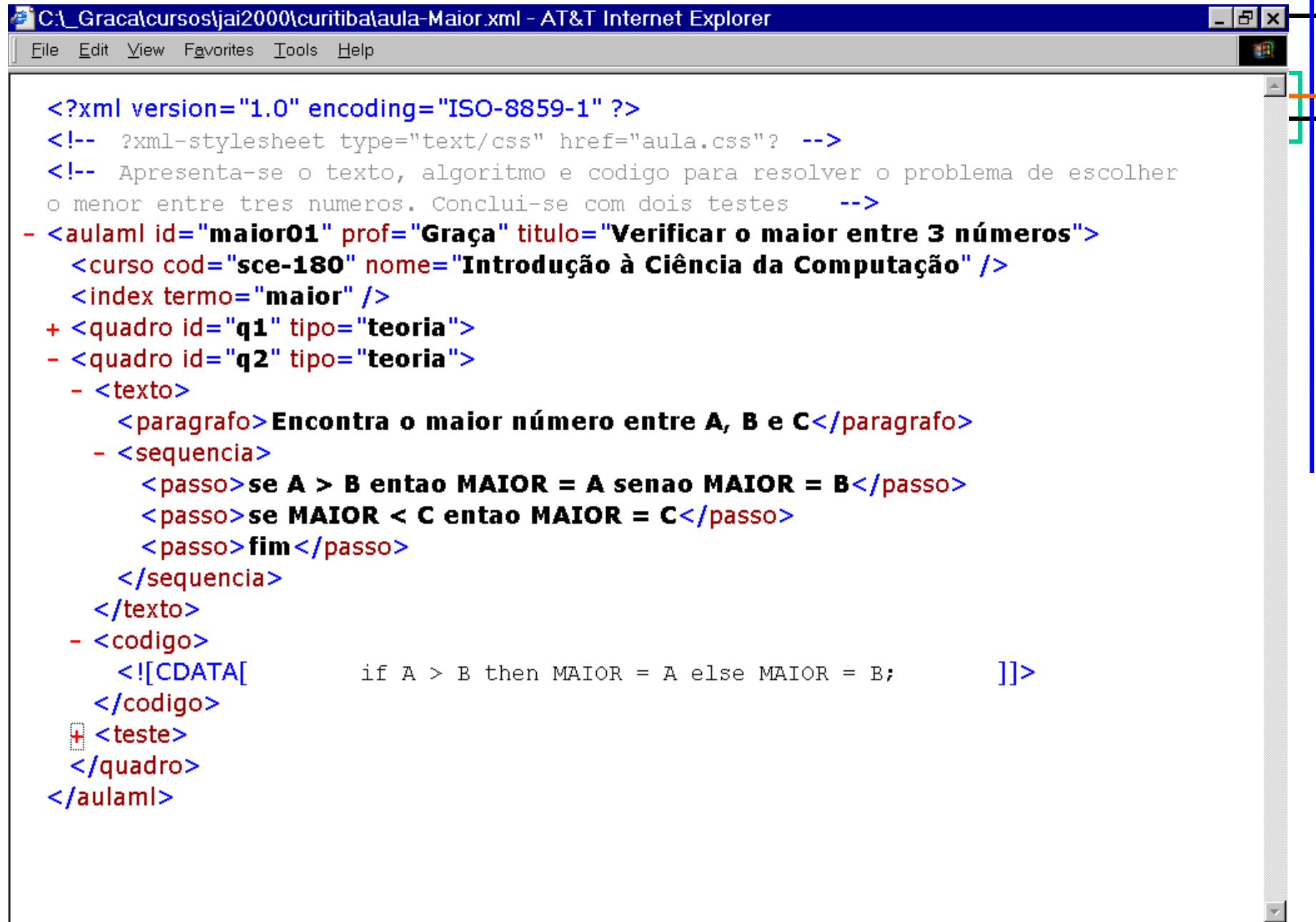
Concluído

How to add style to ... re_aula3a_xml_css ... aulas2011_scc5811... re_aula4_prog_DOM... [Área de Trabalho - ... Código-fonte: http://... zotero

Nosso exemplo

Figura 1: aula.xml

Figura 1 - Exemplo de um documento XML



Para encontrar o maior entre três números A, B e C o seguinte procedimento pode ser utilizado.

Compare A e B e armazene o maior valor em "maior"

Compare "maior" com C e armazene o maior valor em "maior"

Fim.

Encontra o maior número entre A, B e C

se $A > B$ então $MAIOR = A$ senão $MAIOR = B$

se $MAIOR < C$ então $MAIOR = C$

fim

if $A > B$ then $MAIOR = A$ else $MAIOR = B$;

O algoritmo apresentado é recursivo. Qual das afirmativas está correta? ☐ algoritmo não funciona se os números forem reais ☐ algoritmo funciona mesmo que todos os números sejam iguais ☐ algoritmo funciona apenas se os números forem todos diferentes

aula.css

Elemento a que se aplica

Regra

```
paragrafo { display: block; font-size:
  16pt; font-weight: bold; color:blue;
margin-top: 10px }
```

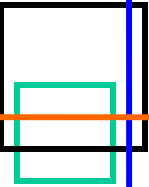
Estilo a aplicar

```
item { display: block; font-style:
  italic; margin-top: 5px }
```

```
passo { display: block; margin-
  bottom: 10px; margin-top: 5px }
```

```
codigo { display: block; margin-bottom:
  10px; font-family: Helvetica, sans-
  serif }
```

seleção dos elementos pode ser otimizada...



```
paragrafo, item, passo, codigo  
  { display: block }  
multiplo,  passo{ font-style:  
  italic; margin-top: 5px}
```

a seleção pode ser mais refinada...



```
item.difícil {color:blue}
```

```
item.difícil.segundo{font-  
weight: bold}
```

Associação ao documento

- por instrução de processamento

```
<?xml-stylesheet type="text/css"  
href="aula.css"?>
```

- uma stylesheet importa outras com @import

```
@import url  
    (http://www.w3.org/basicstyles.css)  
@import url (/styles/agenda.css)
```


No caso de um *browser*...



- O usuário pode especificar uma folha de estilo utilizando recursos do *browser*
- O *browser*, por sua vez, provê estilos padrão para a maioria das propriedades definidas para HTML...
- Cada browser é responsável por atender às especificações...

!important



- A declaração `!important` possibilita incrementar a prioridade de uma regra

paragrafo (`color:blue !important`
`font-family:serif`)

- O `!important` neste caso aplica-se à propriedade `cor`.

“Cascade”

- É possível associar mais que uma folha de estilo a um mesmo documento.
- O browser, por exemplo, pode ter uma folha de estilo default que é adicionada à folha especificada pelo autor do documento.
- Nesse caso podem ocorrer conflitos
- Passa a ser importante determinar em que ordem as regras se aplicam.
- Esse processo é chamado “cascade” e é responsável pelo nome “cascading” atribuído à linguagem.

Ordem de cascadeamento...

- As regras **mais** específicas tem maior prioridade.

<CONCERTO>

<MUSICA ID="x09" CLASS="ROCK">

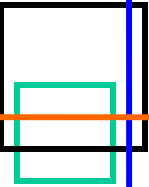
American woman

</MUSICA>

</CONCERTO>

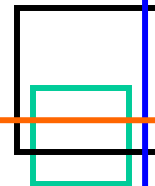
- Regra que seleciona MUSICA pelo ID tem preferência sobre a que seleciona MUSICA pela sua CLASSE, que sobrepõe regra que seleciona MUSICA por ser elemento contido em CONCERTO. Se nenhuma dessas situações acontece, então aplica-se a regra associada à seleção genérica MUSICA. Se nenhuma regra é encontrada para um determinado elemento, deve ser utilizada a regra herdada de seu pai. Se não houver valor herdado, então aplica-se o valor default.
- Se existir mais de uma regra em um mesmo nível de especificidade, a ordem de cascadeamento deve ser: . . .

Ordem de cascadeamento...



- 1- Declarações do autor sinalizadas como importante
- 2- Declarações do leitor sinalizadas como importante
- 3- Declarações do autor não importantes
- 4- Declarações do leitor não importantes
- 5- A última regra da stylesheet definida para o elemento

Valores p/ propriedade display



- **block**: default. Os itens devem aparecer em espaços próprios separados de alguma forma dos outros elementos
- **inline**: os itens aparecem em seqüência, na mesma linha (a menos que especificado explicitamente em contrário) (default p/ CSS2)
- **list-item**: itens aparecem em forma de lista
- **none**: itens não aparecem; são invisíveis.

Valores p/ demais propriedades



- Fonte
- cor
- background
- texto
- box
- etc.

Limitações da CSS1



- Atribui estilos somente a conteúdo que já aparece no documento. Não pode adicionar conteúdo, mesmo que simples pontuações! (XSL pode)
- Não é capaz de transformar o conteúdo, como aplicar ordenação, etc. (XSL é)
- Não tem suporte para tabelas.
- Não consegue manipular texto em diferentes ordens, como acontece em Árabe, Chinês, Hebreu, etc.

CSS2 - (1998) O que é . . .

- Extensão de CSS1
- Um superconjunto de CSS1
- Poucas redefinições, como o default para a propriedade **display**
- Capaz de formatar p/ papel e p/ Web
- Muitas das regras ainda não implementadas por browsers tradicionais como Mozilla e IE.

CSS2 - características



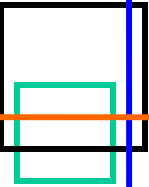
- Permite selecionar e formatar mais precisamente o elemento no documento
- Seleção do primeiro filho de um elemento
- Ajuste do elemento quando recebe foco
- Controle automático da disposição de elementos ao redor de elementos selecionados
- Controle de mudança de páginas p/ slide-show.
- Suporte a tabelas
- Indentação automática de listas

CSS2 - características



- Suporte a linguagens não ocidentais
- Suporte a estilos orais: especifica não como um documento deve ser renderizado mas sim como deve ser lido em voz alta.
- Dois pseudo elementos **:after** e **:before** que permitem a inserção de conteúdo

<http://www.richinstyle.com/>



- É um site com tutoriais, lista de *bugs* e mais!

Aplicativos Locais Sistema RichInStyle.com - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

w3c stylebot css

Most Visited Getting Started Latest Headlines icmc [DokRat] USP :: Sistemas USP:: https://192.168.231... Gmail: Google Tidia-Ae CoteiaWiki

RichInStyle.com RichInStyle.com CS... Tidia-Ae : My Works... Tidia-Ae : SCC5811_... How to add style to... w3c stylebot css - P...

RichInStyle.com

What's hot at RichInStyle.com

- Test suite** - new version, generated from a database of 2000 tests with results submission and automatic reporting on bugs, etc.
- Netscape 6** - standards review.
- Web-safe colorizer** - select one of the web-safe colors from a grid to view it as a background, and then see how each of the colors looks as a foreground, all at once.
- CSS2 guide** - learn all of CSS2 from the only guide on the net.
- HTML 4 guide** - learn all of HTML from the best guide on the net.
- CSS masterclass** - how to create perfect style sheets (not for CSS newbies)
- MySite** - a free script to give your users the freedom to select between style schemes or define their own, with their choice of font sizes, background colors, background images, etc., so they see your site the way *they* want.
- AlwaysWork style sheet** - free AlwaysWork style sheets - designed to improve the appearance of pages in all CSS browsers while causing problems in none.

Guides

- CSS1 - learn CSS1 now!
- CSS2 - the only guide to CSS2 on the internet
- HTML 4 - learn HTML

Bugs

- Bug table - all the CSS properties in each browser at a glance
- Internet Explorer 3 (195 bugs)
- Internet Explorer 4 (203 bugs)
- Internet Explorer 5/5.5 (162/157 bugs)

Concluído

RichInStyle.com - Mozi... re_aula3a_xml_css_x... aulas2011_scc5811 -... re_aula4_prog_DOM.o... [Área de Trabalho - Na...

RichInStyle.com - CSS support table - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

http://www.richinstyle.com/bugs/table.html w3c stylebot css

Most Visited Getting Started Latest Headlines icmc [DokRat] USP:: Sistemas USP:: https://192.168.231... Gmail: Google Tidia-Ae CoteiaWiki

RichInStyle.com - C... RichInStyle.com CS... Tidia-Ae : My Works... Tidia-Ae : SCC5811... How to add style to... w3c stylebot css - P...

Note that Mozilla 5 refers to the 12th July build. IE 5.5 is included in the IE 5 column, separated by a '/'.

Attachment of style to pages

Feature	NN4	Moz5	IE3	IE4	IE5	OP3.6
Linked style sheets	B	Y	B	Y	Y	Y
Alternate style sheets	N	N	N	N	N	N
STYLE	Y	Y	B	Y	Y	Y
@import (without media)	N	Y	D (avoidable)	B	Y	Y
@import (with media)	N	Y	D (avoidable)	N	N	B
Media type - screen	D (applied to print as well)	Y	N/A	Y	Y	Y
Media type - all	N	Y	N/A	Y	Y	Y
Media type - print	N	Y	N/A	Y	Y	N
Media attribute on LINK	B	Y	D (applied regardless of medium)	Y	Y	Y
@media	N	Y	D	D	Y	N
Inline style	B (on IMG)	G	G	G	G	G

Application of styles to pages

Concluído

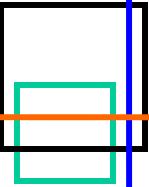
RichInStyle.com - CSS... re_aula3a_xml_css_x... re_aula4_prog_DOM.o... [em2008_Hipermedia... re_aula3b_xml_css_x...

... Dimensão de apresentação

A decorative graphic in the top right corner consisting of a black square, a green square slightly offset to the bottom-right, an orange horizontal line passing through the green square, and a blue vertical line passing through the right side of the black square.

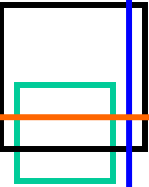
XSL: ***Extensible Style Language***

XSL



- constitui-se basicamente de duas linguagens descendentes de XML
 - uma linguagem de **transformação**, XSLT
 - e uma linguagem de **formatação**, XSLF
- A linguagem de transformação provê elementos que definem regras de transformação de um documento XML em outro documento XML
- A linguagem XSLF provê um conjunto de elementos e atributos *ideal* para caracterizar propriedades de apresentação.

XSLT



- A habilidade de XSLT em transformar um dado de uma representação XML para outra é importante em aplicações como comércio eletrônico e intercâmbio eletrônico de dados, em que é comum a necessidade de **troca de informações** envolvendo banco de dados distintos e plataformas heterogêneas.
- Afinal, não são todos os dados que devem ser renderizados em uma tela de monitor ou impressos no papel. Uma representação XML comum dos dados intercambiados pode ser bastante útil nesses casos.

XSLT trata um documento XML como uma árvore

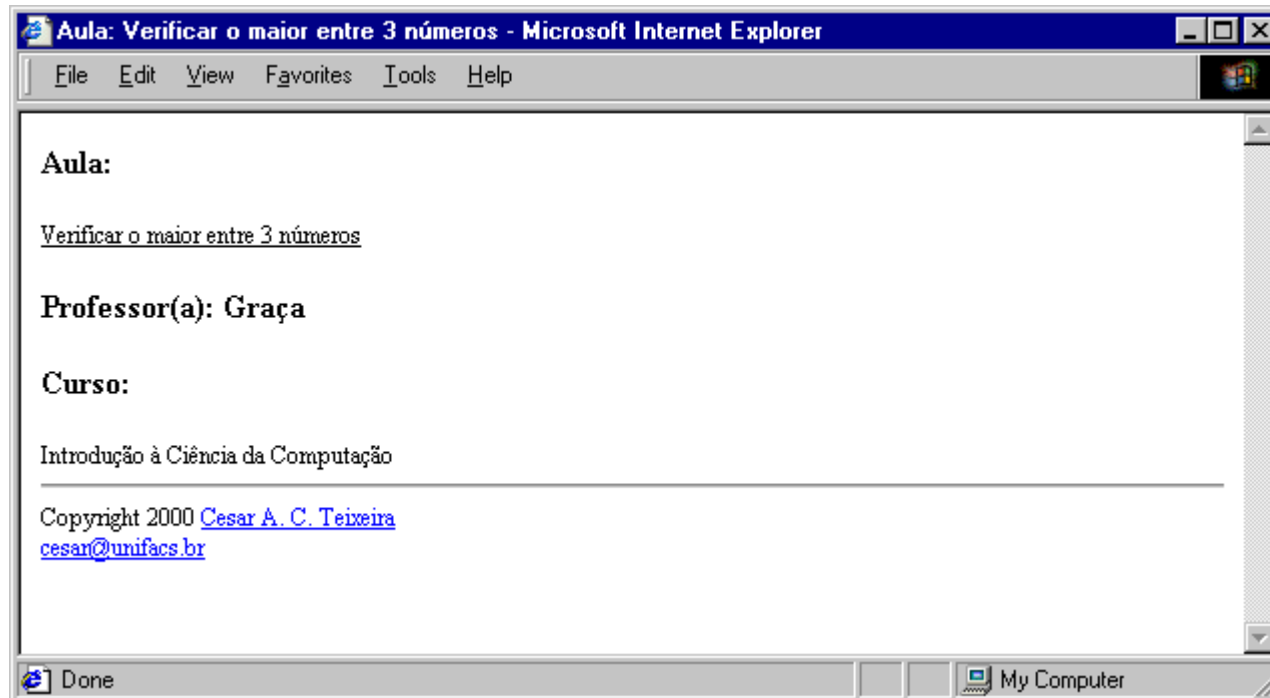


- Os nós da árvore são os elementos, seus conteúdos, atributos, *namespaces*, instruções de processamento e comentários.
- A raiz da árvore do documento deve ser distinta do elemento raiz do documento
 - já que instruções de processamento, por exemplo, estariam no mesmo nível que o elemento raiz.
- Assim XSLT permite, por exemplo, transformar uma árvore em outra.

Um documento XSL contém uma lista de regras



- Uma regra de “template” define um padrão
 - especifica as árvores a que se aplica
 - especifica o que deve ser produzido como resultado para quando o padrão for encontrado
 - o resultado geralmente inclui alguma marcação, dado novo ou dado copiado da árvore original
- Elementos de XSL possuem o prefixo xsl:
- Elementos sem esse prefixo fazem parte da árvore resultado.



exemplo: aula.xsl 1/3

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
    <html>
      <head>
        <title>
          Aula:
          <xsl:for-each select="aulaml">
            <xsl:value-of select="@titulo"/>
          </xsl:for-each>
        </title>
      </head>
```


exemplo: aula.xsl 2/3

```
<body>
  <xsl:for-each select="aulaml">
    <h2>Aula:</h2>
    <u><xsl:value-of select="@titulo"/></u>
    <p></p>
    <h2>Professor(a):
      <xsl:value-of select="@prof"/>
    </h2>
    <p/>
    <xsl:for-each select="curso">
      <h2>Curso:</h2>
      <xsl:value-of select="@nome"/>
    </xsl:for-each>
  </xsl:for-each>
```

exemplo: aula.xsl 3/3

```
<hr></hr>
  Copyright 2000
  < a href ="http://www.unifacs.br/cesar">
  Cesar A. C. Teixeira
  </a>
  <br />
  <a href="mailto:cesar@unifacs.br">
  cesar@unifacs.br
  </a>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

XSL



- Consiste de três partes:
 - XSLT (**XSL Transformation**) – uma linguagem para transformar documentos XML.
 - **XPath** – uma linguagem para definir partes de um documento XML.
 - **XSL Formatting Objects** – um vocabulário para formatar documentos XML.

Esclarecendo...

- Pense em XML com sendo uma linguagem que:
 - Pode transformar XML em XHTML.
 - Pode filtrar e classificar dados XML.
 - Define partes de um documento XML.
 - Pode formatar dados XML com base nos valores dos dados, como mostrar valores negativos em vermelho.
 - Pode exibir dados XML em diferentes dispositivos, como tela, papel ou voz.

XSLT e XPath



- XSLT será utilizada para definir as transformações XML.
- XPath será utilizada para definir padrões compatíveis para as transformações.

XSLT

- É utilizada para transformar documentos XML em outros documentos XML , ou em outro tipo de documento que é reconhecido por um *browser*.
- Pode acrescentar ou remover elementos ao arquivo de saída, re-arranjar os elementos, e tomar decisões sobre quais elementos apresentar ou não.

XSLT – Como funciona?



- Utiliza a XPath para definir as partes do documento fonte que “casam” com um ou mais *templates* pré-definidos.
- XSLT transforma a parte que “casou” do documento fonte em um documento resultante.

Browsers e XSL



- Internet Explorer (IE):
 - As versões IE 5 e IE 5.5 não são 100% compatíveis com a recomendação oficial W3C da XSL.
 - O IE 6 é completamente compatível com a recomendação oficial.
- Netscape:
 - O Netscape 6 não oferece suporte completo para a recomendação oficial W3C da XSL.
- Os demais??? ... verificar!

XSL-Transformation (XSLT)

Maneiras corretas de se declarar uma folha de estilo XSL:

```
<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">
```



Transformando um documento XML em XHTML

O documento XML... (cdcatalog.xml)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<catalog>
  <cd>
    <title>How The West Was Won</title>
    <artist> Led Zeppelin </artist>
    <company> Warner </company>
    <price> 61,90 </price>
    <year> 2003 </year>
  </cd>
</catalog>
```

A folha de estilo XSL...

(cdcatalog.xsl)

```
<?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="#9acd32">
        <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>
```

Ligando a folha de estilo XSL ao documento XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<?xml-stylesheet type="text/xsl"  
href="cdcatalog.xsl"?>  
<catalog>  
  <cd>  
    <title>How The West Was Won</title>  
    <artist> Led Zeppelin </artist>  
    <company> Warner </company>  
    <price> 61,90 </price>  
    <year> 2003 </year>  
  </cd>  
</catalog>
```

Explicando...

```
<?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="#9acd32">  
          <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>
```

<xsl:template> (1/2)



- Uma folha de estilo XSL consiste de um conjunto de regras chamadas **templates**.
- O elemento **<xsl: template>** contém regras a serem aplicadas quando um nó específico “casa”.
- O atributo **match** é utilizado para associar o template com um elemento XML.

<xsl:template> (2/2)

match = "/" associa um template à raiz do documento XML fonte

```
<?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 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>
```


<xsl:value-of>

- O elemento **<xsl:value-of>** pode ser usado para selecionar o valor de um elemento XML e o apresentar na saída da transformação.
- Exemplo:
<xsl: value-of select="catalog/cd/title"/>
<xsl: value-of select="catalog/cd/artist"/>

<xsl:for-each> (1/2)

- O elemento **<xsl:for-each>** permite realizar **loops** na XSL.
- Pode ser utilizado para selecionar todo elemento XML de um conjunto de nós específicos.
- Exemplo:

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

<xsl:for-each> (2/2)

- Pode-se filtrar a saída acrescentando um critério ao atributo selecionado no elemento <xsl:for-each>:

<xsl:for-each select="catalog/cd[artist='Led Zeppelin']">

- Operadores de filtro:

= (igual)

!= (diferente)

< (menor que)

> (maior que)

<xsl:sort>

- O elemento **<xsl:sort>** é utilizado para ordenar a saída.
- Exemplo:
<xsl:for-each select="catalog/cd">
<xsl:sort select="artist"/>
<xsl:value-of select="title"/>
<xsl:value-of select="artist"/>
</xsl:for-each>

<xsl:if>

- O elemento **<xsl:if>** contém um template que será aplicado somente se uma condição específica for verdadeira.
- Exemplo:

```
<xsl:for-each select="catalog/cd">  
  <xsl:if test="price > 30"/>  
    <xsl:value-of select="title"/>  
    <xsl:value-of select="artist"/>  
</xsl:if>  
</xsl:for-each>
```

<xsl:choose> (1/2)



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

<xsl:choose> (2/2)

- Exemplo:

```
<xsl:for-each select="catalog/cd">  
  <xsl:value-of select="title"/>  
  <xsl:choose>  
    <xsl:when test="price > 30">  
      <xsl:value-of select="artist"/>  
    </xsl:when>  
    <xsl:otherwise>  
      <xsl:value-of select="company"/>  
    </xsl:otherwise>  
  </xsl:choose>  
</xsl:for-each>
```

<xsl:apply-templates>



- O elemento **<xsl:apply-templates>** aplica uma regra de template ao elemento atual ou aos seus nós filhos.
- Exemplo:
Próximo slide


```
<?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>
<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>

</xsl:stylesheet>
```

