



SCC0661 – MULTIMÍDIA E HIPERMÍDIA

Aula 10 – Linguagens de Marcação

Prof. Marcelo G. Manzato (mmanzato@icmc.usp.br)

Baseado no material originalmente produzido por Profa. Dr. Renata Pontin
e Luciano Pansanato

LINGUAGENS DE MARCAÇÃO

- “Uma linguagem de marcação (*markup language*) é um tipo de codificação de texto que **representa** texto, assim como detalhes sobre a **estrutura** e a **aparência** do texto.”
 - É distinguível sintaticamente dos dados
- Exemplos de linguagens de marcação
 - SGML
 - Standard Generalized Markup Language
 - ISO 8879:1986
 - HTML
 - Hypertext Markup Language
 - RFC 1866 (HTML 2.0)
 - LaTeX, #twitter, etc.



LINGUAGENS DE MARCAÇÃO

- Três tipos gerais de marcação:
 - **Marcação para apresentação**: sistemas tradicionais de processamento de texto, usando a tecnologia WYSIWYG. Fica escondido de usuários e autores/editores
 - MS Word, OpenOffice, etc.
 - **Marcação de procedimento**: acoplado ao texto, fornecendo instruções para seu processamento
 - Groff, LaTeX, PostScript, etc.
 - **Marcação para descrição**: Usado para rotular partes do documento, ao invés de simplesmente prover instruções de como processá-lo. Desacopla a estrutura do documento de qualquer tipo de tratamento ou renderização particular
 - HTML, XML, etc.



LINGUAGENS DE MARCAÇÃO DESCRITIVAS

Tags (etiquetas)

- *descriptive markups*
- Marcação que delimita um elemento (SGML)

Elemento

- Um componente da estrutura de um documento

<message> Hello World! </message>



start tag
marca o
início



dados



end tag
marca o
final



LINGUAGENS DE MARCAÇÃO DESCRITIVAS

○ SGML – ISO 8879 (1986)

- Inspirada em Scribe (Brian Reid PhD Thesis, 1980).
 - Primeira linguagem a fornecer uma separação clara entre conteúdo e apresentação.
- É uma meta-linguagem → usada para descrever outras linguagens.
- Objetivo:
 - Permitir a formatação de informação de modo que sistemas de divulgação/publicação possam partilhar essas informações com facilidade
- Flexível e Poderosa | Chata e Difícil
 - Pensava-se em marcação manual




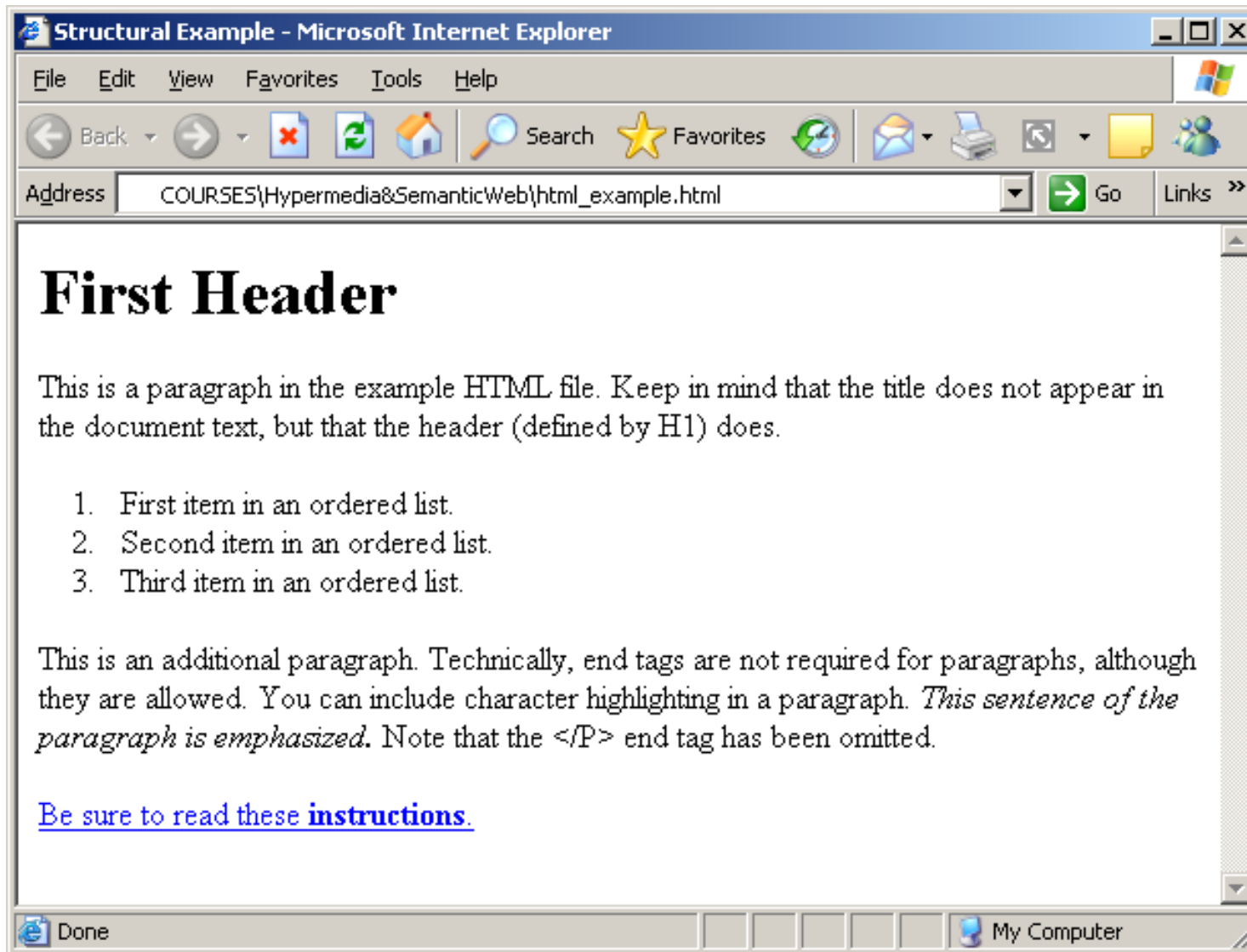
SGML

- Linguagem muito abrangente e complexa
 - Dificulta a criação de aplicações para o ambiente Web
- Vantagens
 - Flexibilidade dos elementos necessários em cada aplicação
 - Definição de novas aplicações quando necessárias
- Desvantagem
 - Custo do processamento no ambiente Web
- HTML é uma aplicação da SGML



```
<HTML>
<!-- Here's a good place to put a comment. -->
<HEAD>
<TITLE>Structural Example</TITLE>
</HEAD>
<BODY>
<H1>First Header</H1>
<P>This is a paragraph in the example HTML file.
Keep in mind that the title does not appear in the
document text, but that the header (defined by H1)
does.</P>
<OL> <LI>First item in an ordered list.
      <LI>Second item in an ordered list.
      <LI>Third item in an ordered list.
</OL>
<P>This is an additional paragraph. Technically,
end tags are not required for paragraphs, although
they are allowed. You can include character
highlighting in a paragraph. <EM>This sentence of the
paragraph is emphasized.</EM> Note that the <P> end
tag has been omitted.
<P> <A HREF="instructions.html"> Be sure to read
these <b>instructions</b>.</A>
</BODY>
</HTML>
```





HTML

- Define uma classe muito simples de documentos para serem apresentados em *browsers*
 - Cabeçalhos, parágrafos, listas, tabelas e imagens
- Limitações
 - Não é extensível
 - Um documento não pode ser reutilizado
 - Pouca ou quase nenhuma semântica pode ser extraída de um documento
 - Dificuldade de processamento por outras aplicações
 - Não separa conteúdo de apresentação



XML

- *Extensible Markup Language*
- Um **subconjunto** da SGML
 - Omite diversas partes mais complexas e menos usadas da SGML
- Compatibilidade com SGML
- Vantagens
 - Facilidade em definir tipos de documentos
 - Facilidade para escrever programas para manipular documentos
 - Separa conteúdo de apresentação



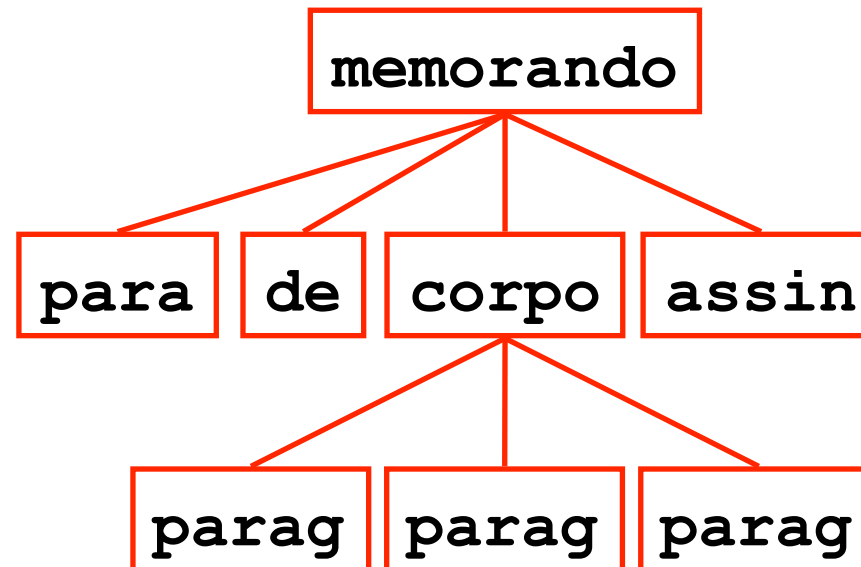
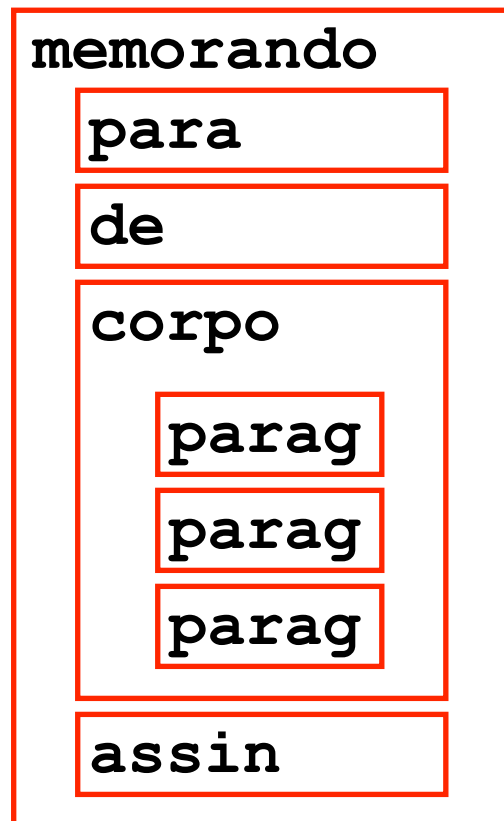
XML

- XML faz referência a:
 - Um padrão aberto (... uma *W3C recommendation*) que fornece um formato e uma linguagem de modelagem de dados
 - O uso de dados formatados em XML em uma aplicação
 - Uma meta-linguagem para criar linguagens de marcação
 - Um conjunto de recomendações e especificações associadas
 - ligação, estilo, transformação, consulta, APIs, etc.



DOCUMENTOS XML

- Documento: composto por elementos
- Elemento: consiste em marcações (tags) e conteúdo
- Conteúdo pode ser texto ou outros elementos



DOCUMENTOS XML

- Elementos podem ter **atributos**
 - Localizados dentro das *start tags*
 - Informação sobre os dados (metadados)
 - Acrescentam informação ao elemento

```
<numeroTelefone tipo="fixo" uso="residencial" plano="basico"  
    instalacao="10-03-2005">
```

16/3371-6519

```
</numeroTelefone>
```



DOCUMENTOS XML

- XML é um **formato de dados**
 - Exemplo: registro de funcionário

```
<registro-funcionario tipo="ativo" nfunc="15">
  <nome>
    <primeiro>Luciano</primeiro>
    <ultimo>Pansanato</ultimo>
  </nome>
  <afiliacao>
    <titulo>Analista de sistemas</titulo>
    <empresa>ICMC-USP</empresa>
    <localizacao>
      <cidade>São Carlos</cidade>
      <estado>SP</estado>
      <cep>13560-970</cep>
    </localizacao>
    <email>ltep@icmc.usp.br</email-name>
  </afiliacao>
  <peso unidade="Kg">70</peso>
  <altura unidade="m">1,76</altura>
</registro-funcionario>
```

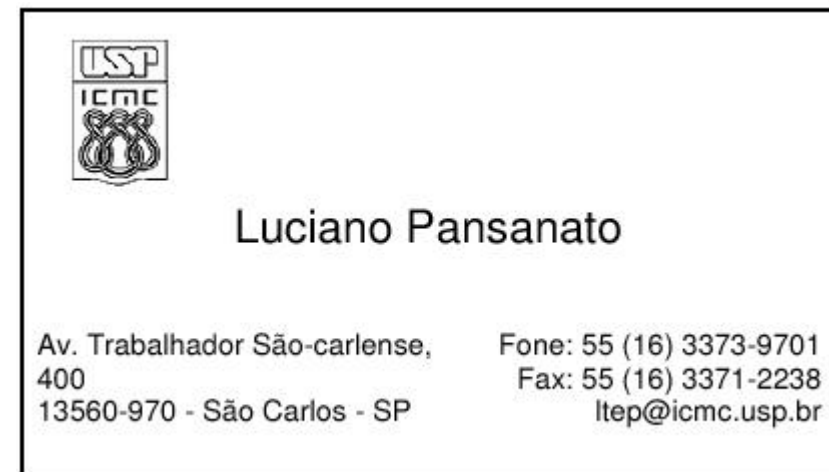


DOCUMENTOS XML

- HTML



- Aplicação impressa



DOCUMENTOS XML

- Base de dados

Key: 00095ABC

NFUNC: 009

001:Luciano


002:Pansanato

008:1,76

014:70

**020:Analista de
sistemas**

- Dados iguais,
aplicações diferentes

FUNCIONÁRIO NOVO!	
	Luciano Pansanato foi admitido recentemente no ICMC-USP como Analista de sistemas
Bem-vindo à nossa equipe, Luciano!	



XML

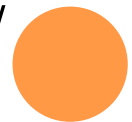
○ Finalidade

- Codificar dados somente uma vez
- Gerar diversos produtos a partir dessa marcação
- Permitir busca complexa (semanticamente)
- Reusar dados (no todo ou em parte) diversas vezes
- Realizar intercâmbio de dados
- Permitir comunicação entre sistemas e aplicações



XML

- XML não faz nada!
 - É apenas um formato de dados
- Software pode fazer muito com marcações
- Exemplos
 - Definir o tipo de informação que será apresentada em determinado momento.
 - Definir semanticamente um conteúdo específico.
 - Buscar por informações específicas no documento.
 - Definir como determinada informação será apresentada/formatada.



XML

- XML também é uma meta-linguagem
- Uma “linguagem” XML: conjunto de tags
 - Domínios: jornal, livro texto
 - Aplicações: comércio eletrônico, gerência de conteúdo
 - Coleções de informação: biografias, leis, dicionários
- Exemplos
 - CML
 - MathML
 - SVG
 - SMIL, NCL, XMT
 - ...



COMPONENTES LÓGICOS DE XML

- Uma aplicação XML é composta de:
 - Um ou mais documento XML (**dados**)
 - DTD ou XML Schema (**modelo**)
 - Especificações de saída (**comportamento**)
 - Transformações (**resultado**)



DOCUMENTO XML

- Composto por **tags** (marcações) e **texto** (conteúdo)
- Dois tipos de documentos
 - Bem-formado (*well-formed*): regras de sintaxe
 - Válido (*valid*): em conformidade com o modelo
- Criação de um documento XML
 - Autoria: usando um editor XML
 - Programação, a partir de:
 - uma base de dados
 - um outro documento XML (transformação)
 - um outro formato, por exemplo, MS-Word (conversão)



SINTAXE XML

- Um documento XML é composto por tags (marcações) e texto (conteúdo)
 - As tags identificam o elemento e seu conteúdo
 - Atributos de elemento
 - Aninhamento de elementos
 - Comentários
 - Seção CDATA
 - Instrução de processamento
 - Declaração XML



SINTAXE XML: ELEMENTO

- Cada elemento tem um nome, e pode ter um conjunto de atributos
- Elemento vazio
- Um documento XML deve ter um elemento raiz que contém todos os outros elementos

```
<director>Peter Jackson</director>
```

```
<br/>
```

```
<version status="draft" target="private"/>
```



SINTAXE XML: ATRIBUTO

- Um par (nome, valor) presente na start tag do elemento, logo após o seu nome
- Os valores dos atributos devem estar entre aspas
- Um atributo não pode aparecer mais de uma vez no mesmo elemento

```
<publication volume="2" number="4"  
  pages="29-66">World Wide Web Journal</publication>
```



SINTAXE XML: ANINHAMENTO

- Elementos podem conter outros elementos
- Elementos internos **não** podem sobrepor elementos externos
- Errado:

```
<livro><titulo>Dom Casmurro  
<autor>Machado de Assis</titulo></autor></livro>
```

- Correto:

```
<livro><titulo>Dom Casmurro</titulo>  
<autor>Machado de Assis</autor></livro>
```



SINTAXE XML: COMENTÁRIOS

- Não fazem parte do conteúdo do documento

```
<!-- comentário... -->
```



SINTAXE XML: CDATA

- Permite a inclusão de trechos que devem ser interpretados como caracteres e não como elementos de marcação
- Seções CDATA não podem ser aninhadas

```
<script>
<![CDATA[
function matchwo(a,b){
    if (a < b && a < 0) {
        return 1;
    } else {
        return 0;
    }
}
]]>
</script>
```



SINTAXE XML: INSTRUÇÃO DE PROCESSAMENTO

- Os parsers XML apenas repassam essas informações para a aplicação
- Um documento XML deve iniciar com uma declaração XML

```
<?xml version="1.0"?>
```

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

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



XML

```
<?xml version="1.0" encoding="utf-8"?>
<CATALOGO>
  <FILME>
    <TITULO tipo="DVD">As Panteras</TITULO>
    <GENERO>Aventura</GENERO>
    <DISTRIBUIDORA>CIC</DISTRIBUIDORA>
    <ANO>2000</ANO>
    <PRECO>64.28</PRECO>
  </FILME>
  <FILME>
    <TITULO tipo="VHS">Dinossauros</TITULO>
    <GENERO>Infantil</GENERO>
    <DISTRIBUIDORA>Walt Disney</DISTRIBUIDORA>
    <ANO>2001</ANO>
    <PRECO>36.90</PRECO>
  </FILME>
</CATALOGO>
```



DOCUMENTO XML BEM-FORMADO

- Algumas regras:

- No texto, não utilizar caracteres especiais como: <, >, &, ' e “
 - Usar (respectivamente): < > & ' e "
- Tags devem ser corretamente aninhadas e fechadas
- Tags são *case-sensitive*
- Nomes das tags não devem conter caracteres especiais:
 - !"#\$%&'()*+,-/;<=>?@[\\]^`{|}~
- Nomes das tags não devem começar com dígito, hífen, ponto, e não devem ter espaço em branco
- Existe um único elemento raiz que contém todos os outros elementos



MODELO DE DOCUMENTO

- Validação do Documento XML
- DTD (*Document Type Definition*)
 - Mecanismo de modelagem da especificação XML
 - Modela um tipo/classe de documento
 - Conjunto de regras que descrevem como os documentos desse tipo/classe podem ser formados
 - Especificam regras



MODELO DE DOCUMENTO

- Por que usar um modelo de documento?
 - DTD atua como um contrato entre produtor e consumidor
 - Ambos podem validar se receberam/enviaram o que era esperado
 - Especificação formal de tipos de informação permite processamento consistente
 - Interoperabilidade de documentos
 - Aplicações não precisam compartilhar software ou programas para processamento




EXEMPLO

- Documento XML

```
<film code="A010" type="DVD">  
  <title>The Lord of the Rings - The Two Towers</title>  
  <director>Peter Jackson</director>  
  <actor>Elijah Wood</actor>  
  <actor>Ian McKellen</actor>  
</film>
```

- DTD

```
<!ELEMENT film (title, director, actor+, year?)>  
<!ATTLIST film code ID #REQUIRED>  
<!ATTLIST film type (DVD|VHS) #REQUIRED>  
<!ELEMENT title (#PCDATA)>  
<!ELEMENT director (#PCDATA)>  
<!ELEMENT actor (#PCDATA)>  
<!ELEMENT year (#PCDATA)>
```



DTD (DOCUMENT TYPE DEFINITION)

- Corresponde à declaração que define a gramática do documento
- Palavras reservadas
 - As palavras reservadas devem aparecer sempre em letra maiúscula
 - ELEMENT, ATTLIST, CDATA, PCDATA, NOTATION, ENTITY
- Nomes de elementos, atributos, entidades e notações
 - Um nome começa com uma letra, underscore (“_”) ou dois-pontos (“:”), seguido de letras, dígitos ou dos sinais “_ - . :”, mas não pode começar com as letras “xml”



VALIDAÇÃO XML

- Indica que o documento deve seguir um DTD
- Pode-se ter DTDs internos ou externos

```
<?xml version="1.0"
      encoding="ISO-8859-1"?>
<!DOCTYPE HELLO [
  <!ELEMENT hello (#PCDATA)>
]>
<hello>
  Olá XML!
</hello>
```

```
<?xml version="1.0"
      encoding="ISO-8859-1"?>
<!DOCTYPE HELLO
  SYSTEM "hello.dtd">
<hello>
  Olá XML!
</hello>
```



DTD: DECLARAÇÃO DE ELEMENTOS

- Identifica o nome do elemento e a natureza de seu conteúdo

- Operadores

,	seqüência (and)
	escolha (ou)
?	opcional
*	zero ou mais filhos
+	um ou mais filhos
()	agrupamento

```
<!ELEMENT thesis (abstract, chapter+)>  
<!ELEMENT abstract (#PCDATA)>  
<!ELEMENT chapter (#PCDATA)>  
<!ELEMENT copyright EMPTY>
```



DTD: DECLARAÇÃO DE ATRIBUTOS

- Nome do atributo, tipo de dados e valor padrão
- Tipo de dados

- Valor

#REQUIRED	obrigatório
#IMPLIED	opcional
#FIXED "value"	fixo
"default value"	padrão

CDATA	Texto (Character Data)
ID	Um nome exclusivo no documento
IDREF	Uma referência para outro elemento
IDREFS	Diversas referências separadas por espaços
ENTITY	Uma entidade declarada
ENTITIES	Diversas entidades separadas por espaços
NMTOKEN	Um XML name token
NMTOKENS	Diversos XML name tokens
NOTATION	Um nome de notação declarado

<!ATTLIST thesis author CDATA #REQUIRED>

OUTRO EXEMPLO

- Crie um DTD para o seguinte XML:

```
<?xml version="1.0"?>
<person>
  <name first="Alan" last="Turing"/>
  <profession value="computer scientist"/>
  <profession value="mathematician"/>
  <profession value="cryptographer"/>
</person>
```



DTD: LIMITAÇÕES

- A verificação de que os valores contidos no documento correspondem ao esperado é responsabilidade da aplicação
- Por exemplo, não existem mecanismos para especificar que:
 - Um elemento <data> deve conter apenas valores válidos para dia-mês-ano
 - Valores de <custo> devem conter apenas valores positivos em alguma moeda corrente
 - Um elemento <titulo> deve conter entre 1 e 255 caracteres
- Outras limitações:
 - Utiliza uma sintaxe especial que não é parecida com a sintaxe XML usual
 - Extensibilidade e escalabilidade
 - Compatibilidade com outros namespaces



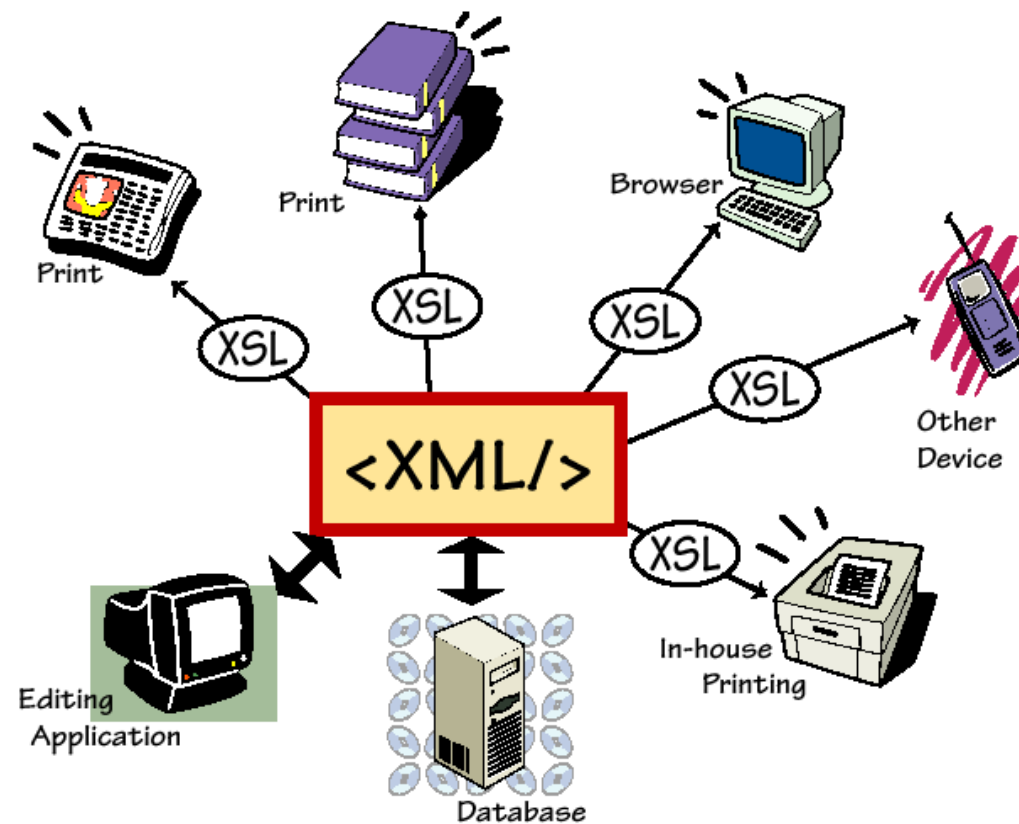
ESPECIFICAÇÕES DE SAÍDA: STYLE SHEET

- Separação de conteúdo do formato (de apresentação) e comportamento
- Define uma aparência ou comportamento
 - Para cada elemento
 - Em cada um de seus contextos do documento
- Uma stylesheet, diversos documentos
 - Manter consistência do formato ao longo dos documentos
 - É fácil desenvolver, manter e aplicar
- Um documento, diversas stylesheets
 - Criar documentos para tipos de mídia diferentes: impressa, online, etc.
 - É fácil produzir documentos derivados



ESPECIFICAÇÕES DE SAÍDA: STYLE SHEET

Transformações



RESUMO

- Markup Languages
- SGML
- HTML
- XML
- Sintaxe XML
- DTD



LINGUAGENS DE MARCAÇÃO:

REFERÊNCIAS

- W3C (2008) Extensible Markup Language (XML) 1.0 (Fifth Edition). W3C Recommendation. <http://www.w3.org/TR/xml/>
- Extensible Markup Language (XML). <http://www.w3.org/XML/>
- Cover Pages: Extensible Markup Language (XML). <http://xml.coverpages.org/xml.html>
- The XML FAQ v4.0 (2005-01-01) <http://www.ucc.ie/xml/>
- Harold, E. R. (2001) XML Bible. Second Edition, Hungry Minds.

