

XML

# O que é a XML ?

- XML significa eXtensible Markup Language.
- É uma linguagem com a finalidade de descrever informações. O mesmo ocorre com a HTML.
- XML representa um aperfeiçoamento da abordagem da HTML.
- O propósito original de sua existência é a WWW.

# Sobre a HTML

- HTML é uma linguagem de Marcação
  - Hyper-text Markup Language

# Um documento HTML X Documento XML

<! - - HTML Snippet - - >

<b>Invoice</b>

<p>From: Joe Bloggs

<p>To: M. Another

<p>Amount: \$100.00

<p>Tax: 21%

<p>Total Due: \$121.00

<! - - XML Snippet - - >

<Invoice>

<From>Joe Bloggs</From>

<To>M. Another</To>

<Date year = '1999' month = '2' day = '1' />

<Amount currency = 'Dollars'>100.00</Amount>

<TaxRate>21</TaxRate>

<TotalDue currency = 'Dollars'>121.00</TotalDue>

</Invoice>

# Comparando os documentos

- Agora, coloque-se em frente a seu computador ...
- Qual deles captura as informações mais úteis ?
- Qual deles possui uso mais potencial ?
- Qual dos dois documentos é mais fácil de se processar ?

## Essência da XML

- A distinção ilustrada nestes dois documentos representa a verdadeira essência da XML.
- XML tem tudo a ver em *preservar* informações úteis.
- Informações que os computadores podem usar para serem mais “inteligentes” no que puderem fazer com seus dados.

## Idéia básica de XML ...

- Clareza de expressão
- XML significa falar de forma mais clara, ou seja, denominar a informação pela forma em que ela é conhecida.
- XML não é um programa de computador, e dessa forma não *faz* nada por conta própria. O mesmo se pode dizer de HTML.

# O que é XML ?

- XML é uma linguagem que oferece uma abordagem padrão para descrição, captura, processamento e publicação de informações.
- XML é uma linguagem de marcação com mudanças muito importantes.
- A maioria das linguagens de marcação, são linguagens, com um certo conjunto fixo de elementos, e esse conjunto é baseado no modelo da linguagem.



# O que é a XML ?

- **HTML**, por exemplo, possui um conjunto fixo de *tags* com o qual podemos elaborar documentos.
- **XML**, por outro lado, **não define** qualquer conjunto de *tags* em particular.

## Um exemplo XML

<Spice>

<Name>Sichuan Peppercorns</Name>

<CountryOfOrigin Country = "China"/>

<Description>Pungent, distinctive, Excellent with slow cooked,  
earthy dishes.</Description>

<Example>Sichuan Braised Chicken</Example>

</Spice>

## XML é eXtensível ...

- Com a XML, temos a liberdade de **definir e usar qualquer nome** que desejarmos para implementar *tags* em seus dados.
- Porque podemos usá-la para **criarmos nossas próprias marcações** (*tags*).
- Ao contrário da maioria das linguagens de marcação, a XML possui uma **estrutura flexível**, com a qual se pode criar linguagens de marcação personalizadas.

# Tipos de Elementos

- Qualquer linguagem baseada em XML é composta de um conjunto de tipos de elementos que recebem alguns nomes e significados.
- Os exemplos incluem os tipos de elementos, Invoice (significado = fatura) , TaxRate (significado = taxa) e Spice (significado = tempero) .

# Metalinguagem

- Ao invés de um conjunto fixo de *tags*, XML oferece uma **estrutura padrão** que lhe possibilita criar sua **própria estrutura**, ou usar **aquelas definidas por terceiros** que melhor se encaixe às suas necessidades.
- XML proporciona uma **estrutura padrão** que lhe possibilita criar sua própria estrutura para seus documentos, por que ela é uma **metalinguagem**.
- Ou seja, **uma linguagem que fala de outra linguagem**.

# Linguagens baseadas em XML de padrão industrial

- XML é aplicada em áreas como:
  - ✓ Tecnologia Push (CDF – Channel Definition Format)
  - ✓ Comércio Eletrônico (OTP – Open Trading Protocol)
  - ✓ Matemática (MML – Mathematical Markup Language)

# Definições de Tipos de Documentos

- Um conjunto de tipos de elementos serve para definir tipos de documentos e são conhecidos como Document Type Definitions (DTD).
- Exemplos de DTD: CDF, OTP e MML

# Definir sua própria linguagem

- A principal vantagem de se poder definir sua própria linguagem de marcação é que ela lhe oferece a liberdade de capturar e publicar informações úteis sobre como seus dados ficarão e sua estruturas, ao invés de ter que se acostumar com um formato criado por outras pessoas.



# Documento XML

- Em XML, criamos o conteúdo do documento concentrando-se no significado real das informações e no modo como são estruturados.
- São adiados itens que tem a ver com apresentação, deixando-os que sejam tratados quando alguém precisar *olhar* o documento.

# SGML

- Desde 1986, existe, o padrão internacional denominado **Standard Generalized Markup Language** (SGML) – ISO 8879, para fazer o que a XML faz.
- Padrão muito poderoso e geral, mas com crescente complexidade.
- XML é um subconjunto da SGML que pretende tornar a SGML “leve” o suficiente para uso na Web.
- Todos os documentos XML são documentos SGML válidos.

# Estrutura de um Documento XML

- XML inclui um mecanismo para definição de regras que controlam como os documentos são estruturados.
- São os chamados **Document Type Definitions – DTD** (Definições do Tipo de Documento)
- Em um DTD, além da definição do tipo de documento, podemos fazer com que os documentos em XML sejam automaticamente verificados.

# DTDs

- Os efeitos de como as informações devem aparecer em um documento XML, são alcançados no DTD, listando-se os tipos de elementos que se deseja usar e indicando a ordem estrutural no qual os elementos podem ocorrer.

## Verificação do Documento

- Por ter uma estrutura simples e bem-definida, XML permite que validemos o documento antes de iniciarmos seu processamento.
- Um programa utilitário denominado **XML Parser** é então capaz de verificar se o documento corresponde ou não às regras prescritas.
- Se não quisermos que a estrutura seja verificada ?

# XML e a Aparência no Browser

- Padrão subsidiário da XML

XSL (XML Style Language)

- Conceito de **folha de estilo** em um processador de textos ou **folha de estilo do padrão HTML** (CSS – Cascading Style Sheet)

# Conceito de Folha de Estilo

- A idéia básica é capturar detalhes sobre como ficarão os vários elementos em um documento e então armazená-los em um **documento em separado**, ao invés de emaranhá-los com o seu conteúdo.

# Documento Separado

- Esta separação permitirá que a apresentação seja alterada apenas com a alteração da folha de estilo.
- XSL é a linguagem de folha de estilo proposta para XML.



# XSL

- Possui mais recursos do que a **CSS**, a qual é ainda largamente compatível com **XSL**.
- **XSL** é um subconjunto simplificado da linguagem de estilo padrão **DSSSL** (ISO/IEC 10179).

E quanto ao hipertexto ?

- XML não predefine qualquer elemento. Então como fazer para especificar vínculos de hipertexto ?
- XML delega a tarefa de capturar as informações de hipertexto, para um padrão subsidiário conhecido como XLL (eXtensible Link Language).

# XLL

- XLL é derivado do padrão existente para expressar vínculos de hipertexto em documentos SGML, conhecido como HyTime, um outro padrão ISO/IEC 10744, criado durante os anos da TEI baseada na SGML.

# Unicode - ISO/IEC 10646

- O Unicode, com seus caracteres de bytes múltiplos, suporta uma grande variedade de alfabetos e idiomas.
- Suportado em linguagens de programação como Java e sistemas operacionais como Windows e AIX.

# Baseada em Padrões ISO/IEC

- XML é derivada da SGML.
- XSL é derivada da DSSSL.
- XLL é derivada da HyTime.
- Unicode.

# Diferenças entre XML e HTML

- XML é a **estrutura** para se fazer linguagens de marcação, enquanto HTML é uma **linguagem** de marcação.
- XML pode ser vista com uma **generalização do paradigma de publicação de informações**, cuja pioneira é a HTML e a Web.

# Sobre XML

- O menor denominador comum no mundo dos documentos é o *texto simples*.
- XML foi criada com base no *texto simples*, permitindo que os níveis de informações úteis sejam capturados juntamente com dados básicos, fazendo assim de forma completamente independente do aplicativo e do fornecedor.

# Sobre XML

- XML é uma tecnologia voltada ao documento.
- A abordagem da XML é representada por conteúdo, estrutura e apresentação, atravessando exatamente o mundo da Tecnologia da Informação.



# Sobre XML

- XML é o que se chama de tecnologia de infraestrutura.
- Sobre a estrutura simples e bem-formada de XML, outras aplicações são construídas.
- O desenvolvimento de XML consiste na **criação de vocabulários** de domínios de aplicação específicos que serão processados por programas.

# Vocabulários em XML

- Foram criados vocabulários para aplicações como WAP, intercâmbio financeiro aberto, bancos de dados, química, comércio eletrônico, entre outros.
- Esses vocabulários são definidos através dos DTDs.

# Conceitos Básicos em XML

- XML é uma linguagem usada para descrever e manipular documentos estruturados.
- A manipulação de documentos XML é feita por meio desta estrutura.
- Um documento XML é estruturado em forma de árvore.

# Conceitos Básicos em XML

- Possui sempre um elemento-raiz, de onde outros elementos vão se ramificar.
- Essa estrutura em árvore estabelece como documentos XML vão ser definidos e vão ser tratados.
- O tratamento envolve principalmente encontrar um determinado elemento, ou grupo de elementos para serem processados.

# Exemplo

- <?xml version="1.0"?>  
 <Capitulo>  
 <Titulo>  
 Capitulo Um  
 </Titulo>  
 <Tabela>  
 <Titulo>  
 Tabela Um  
 </Titulo>  
 </Tabela>  
 <secao><Titulo>Secao Um</secao><Titulo>  
 <Tabela><Titulo>Tabela Dois</Titulo></Tabela>  
 </secao>  
 </Capitulo>

# Hierarquia e Sequência

- Em XML duas **relações entre os elementos** são importantes:

**Hierarquia e Sequência**

# Hierarquia

- Estabelece relações:

**pai/filho** e **ancestral/descendente**

- <Capitulo> é o elemento-raiz.
- Os “Titulo” **filhos** de <Capitulo> são:  
    “Capitulo Um”
- Os “Titulos” que são **descendentes** de <Capitulo> são: “Capitulo Um”, “Tabela Um”, “Secao Um”, e “Tabela Dois”.

# Sequência

- A sequência em que as informações aparecem também é importante:
- <Instrucoes>
  - <Titulo>Lavar os cabelos</titulo>
  - <Sequencia>
    - <Passo>Molhar os cabelos</Passo>
    - <Passo>Fazer Espuma</Passo>
    - <Passo>Enxaguar</Passo>
    - <Passo>Repetir</Passo>
  - </Sequencia>
- </Instrucoes>



# Dados e Marcações

- Um documento XML consiste de uma combinação de **dados caracteres** e **marcações (*tags*)**.
- **Marcações** podem ser:
  - caracteres de início de tags
  - fim de tags
  - elementos vazios
  - referências à entidades
  - referências à caracteres
  - comentários

# Dados e Marcações

- delimitadores de seções CDATA

(valores a serem desconsiderados quanto à sintaxe XML)

- declaração de tipo de documento
- instruções de processamento
- declarações XML
- declarações textuais
- quaisquer espaço em branco fora do elemento-raiz e fora de qualquer outra *tag*.

# Elementos de Informação

- Os elementos de informação podem representar estruturas de dados arbitrárias.
- Elementos de informação podem conter valores codificados como caracteres, e nesse caso, dizemos que são elementos com apenas *texto*.

# Elementos de Informação

- Um elemento representa um **tipo estruturado**, se ele não contém **dados textuais**, e sim um ou mais **elementos descendentes**.
- Também podem existir **elementos descendentes com texto** (**conteúdo misto**).
- Documentos que não têm conteúdo misto são chamados **normalizados**.

# Elemento com Tipo Estruturado

- <?xml version="1.0"?>  
 <Capitulo>  
 <Titulo>  
 Capitulo Um  
 </Titulo>  
 <Tabela>  
 <Titulo>  
 Tabela Um  
 </Titulo>  
 </Tabela>  
 <secao><Titulo>Secao Um</secao><Titulo>  
 <Tabela><Titulo>Tabela Dois</Titulo></Tabela>  
 </secao>  
 </Capitulo>

# XML com Conteúdo Misto - 1

<Pedidos>

<NroPedido NumPedido="12435">

<Cliente NumClient="543">

<CustName>IndustriasABC</CustName>

<Rua>Avenida Central,123</Rua>

<Cidade>São Paulo</Cidade>

<Estado>SP</Estado>

<CEP>60609</CEP>

</Cliente>

</DataPedido>981215</DataPedido>

<Linha NroLinha="1">

<Peca NroPeca="123">

# XML com Conteúdo Misto - 2

<Descricao>

<P><B>Alicate:</B><BR/>

Aco Inox, construido em uma so peca. Garantia 10 anos.</P>

</Descricao>

<Preco>9.95</Preco>

</Peca>

<Quantidade>10</Quantidade>

</Linha>

<Linha NroLinha="2">

<Peca NroPeca="456">

<Descricao>

# XML com Conteúdo Misto - 3

<P><B>Separador de gizmos:</B><BR/>

Aluminio,um ano de garantia.</P>

</Descricao>

<Preco>13.27>/Preco>

</Peca>

<Quantidade>5</Quantidade>

</Linha>

</NroPedido>

</Pedidos>



# Atributos em XML

- Cada item de elemento de informação pode ter uma ou mais propriedades, chamadas de **atributos**.
- Os **atributos** formam uma lista não-ordenada de itens de informação.

# Atributos em XML

- <IMG SRC="logotipo.gif" WIDTH="32" HEIGHT="64" ALT="Logotipo">
- Cada atributo tem um **nome único**, e cada um possui uma **propriedade**, consistindo de zero ou mais caracteres.
- Em XML, atributos podem ser usados para **descrever um elemento**, ou para fornecer informações adicionais sobre um elemento.

# Sintaxe de XML

- Para podermos escrever, ler e entender como XML estrutura documentos e também, como e onde podemos usá-la;
- Documento **XML bem-formado**:  
se possui o seguinte formato, nesta ordem:  
**Prólogo, Elemento-Raiz, Miscelânea**

# Prólogo

- Consiste obrigatoriamente na declaração de que este é um documento XML, e a versão da recomendação W3C que ele segue.
- Todas as declarações XML iniciam com os cinco caracteres:  
`<?xml version="1.0"? encoding="ISO-8859-1" standalone="yes"?>`

# Tipo de Codificação

- encoding

Estabelece o conjunto de caracteres utilizado no documento. Valores para esta declaração podem ser encontrados em <http://charts.unicode.org/>

Exemplos:

UTF-8, US-ASCII, ISO-8859-1

# DTD Externo

- standalone

Indica se o documento XML necessita ou não de um DTD externo:

no - o documento depende de um DTD externo;

yes - o documento não depende de um DTD externo.

# Elemento-Raiz

- Um documento XML bem-formado contém um e apenas um elemento-raiz.

<x>	<!--Incorreto!-->
<a> ... </a>	<a> ... </a>
<b> ... </b>	<b> ... </b>
</x>	

# Um Documento XML

- <?xml version="1.0" encoding="ISO-8859-1"?>  
 <peessoa>  
 <id>770412-4956</id>  
 <nome>  
 <prenome>João</prenome>  
 <sobrenome>Silva</sobrenome>  
 </nome>



# Continuação

- <endereço>  
    <rua>Central,205</rua>  
    <cep>01000</cep>  
    <cidade>São Paulo</cidade>  
    </endereço>

## Continuação

- <telefone>  
    <res>011-5556677</res>  
  </telefone>  
</pessoa>
- O primeiro elemento do documento é o elemento-raiz <pessoa>.
- As próximas linhas definem os elementos descendentes da raiz (**id**, **nome**, **endereço** e **telefone**)

# Tags

- São definidas pelo criador do documento, e devem seguir algumas regras não presentes em HTML.
- Se um documento não for **bem-formado** ele será obrigatoriamente rejeitado pelo programa que está lendo.

# ISO-8859-1

- Neste padrão estão presentes os caracteres acentuados como os usados na língua portuguesa. Assim, podemos usar caracteres acentuados não apenas nos *tags*, como também no corpo do documento.

# Documento XML bem-formado

- Sendo XML uma linguagem altamente estruturada, é imperativo que **todos os documentos XML estejam em conformidade com a recomendação XML 1.0.**
- Dizemos que um documento é **bem-formado** quando ele atende a todas as exigências **estruturais** desta recomendação.

# Documento XML

- `<?xml version="1.0" encoding="ISO8859-1" standalone="yes"?>`

`<Recibo>`

`<IDRecibo>1000</IDRecibo>`

`<NomeDoCliente>Jane Smith</NomeDoCliente>`

`<Items>`

`<Item>`

`<Codigo>134</Codigo>`

`<Descricao>Meia</Descricao>`

`<Preço>9.95</Preço>`

`</Item>`

# Continuação

<Item>

<Código>153</Código>

<Descrição>Wolkswagen Passat</Descrição>

<Preço>10000,00</Preço>

</Item>

<Item>

<Código>171</Código>

<Descrição>Cafezinho</Descrição>

<Preço>0,30</Preço>

</Item>

</Items>

</Recibo>

# Verificação de Bem-Formado

- Existem vários lugares na Web que podem ser usados para verificar se um documento XML é bem-formado.
- <http://www.ltg.ed.uk/~richard/xml-check-html>



## Cont ...

- <http://www.scripting.com/frontier5/xml/code/xmlValidator.html>
- <http://www.wml.com/xml/pub/tools/ruwf/check.html>
- <http://www.stg.brown.edu/service/xmlvalid/>
- <http://koala.inria.fr:8080/>

# Regras para Definição de Nomes

- **Nomes** (de **elementos** e **atributos**) devem iniciar com **uma letra** (A .. Z, excluindo quaisquer caracteres acentuados ou cedilha), seguida de **outras letras**, ou **dígitos**, ou **caracteres de pontuação**:
  - “.” (ponto decimal),
  - “,” (vírgula),
  - “-” (hífen),
  - “\_” (sublinhado),
  - “:” (dois pontos).

# Exemplos de Elementos Válidos

- <NOME>
- <Livro>
- <volume>
- <controle\_de\_trafego>

# Elementos Inválidos

- <livro%8>
- <controle de trafego>  
(caracteres brancos)
- <2linhas>

# Tags Maiúsculas e Minúsculas

- Uma *tag* `<CARTA>` é diferente da tag `<carta>`, que é diferente de `<Carta>`.
- *Tags* de abertura e de fechamento, devem ser escritas de maneira idêntica.

`<Mensagem>Esta frase é incorreta</message>`

`<Mensagem>Esta frase é correta</Mensagem>`

# Abertura e Fechamento

- Em XML todos os elementos devem ter uma tag de fechamento, o que não acontece em HTML.
- Exemplos:
  - `<p>Este é um parágrafo</p>`
  - `<p>Este é outro parágrafo</p>`

# Elementos XML Vazios

- Elementos vazios devem conter uma barra (/) de fechamento antes do final da *tag*.
- Exemplo:  
`<p>Este é um parágrafo seguido de uma linha em branco</p><br/>`

# Aninhamento de Elementos XML

- Todos os elementos XML devem ser aninhados apropriadamente.
- Em XML todos os elementos devem estar aninhados como segue:

***<b><i>Este texto está em negrito e itálico</i></b>***



# Elemento-Raiz

- Todos os documentos XML devem ter uma única tag-raiz que define o elemento-raiz.
- Todos os outros elementos, sem exceção, devem estar aninhados dentro desse elemento e podem conter, por sua vez, sub-elementos aninhados.

## Cont ...

- <raiz>  
    <filho>  
        <subfilho>  
            ... ..  
        </subfilho>  
    </filho>  
</raiz>

# Caracteres Reservados

- Vários caracteres fazem parte da estrutura de XML, e portanto, não podem ser usados nos dados do documento XML.
- É necessário substituí-los por **sequências especiais de caracteres** chamadas de **entidades**.

# Caracteres e Entidades

- <    &lt;
- &    &amp;
- >    &gt;
- “    &quot;
- ‘    &apos;

Cont ...

- Exemplo:

Para escrevermos,

“Moinho Brasil & Cia. Ltda”

<Empresa>

<Nome>

&quot;Moinho Brasil&Amp;Cia.Ltda&quot;

</Nome>

</Empresa>

# Atributos

- Um **atributo** é um par (**nome**,**valor**) associado a um elemento.
- Os atributos são **declarados dentro da *tag* de abertura** de um elemento.

# Valores de Atributos

- Valores de atributos devem estar entre aspas ( " " ) ou apóstrofos ( ' ' ).
- Exemplos:

```
<IMG SRC="logotipo.gif" WIDTH="32"  
HEIGHT="64" ALT="Logotipo">
```

```
<IMG SRC='logotipo.gif' WIDTH='32' HEIGHT='64'  
ALT='Logotipo'>
```

```
<peessoa NASCIMENTO="01/06/1980" >
```

# Elementos X Atributos

- Usando um atributo pra definir sexo:

```
< Pessoa sexo="feminino">
```

```
  < nome>Ana</ nome>
```

```
  < sobrenome>Silva</ sobrenome>
```

```
</ Pessoa>
```



## Cont ...

- <essoa>  
 <sexo>feminino</sexo>  
 <nome>Ana</nome>  
 <sobrenome>Silva</sobrenome>  
</essoa>

Faz mais sentido ?

# Limitações de Atributos

- Atributos não podem conter múltiplos valores. Os atributos são pares nome-valor. Um atributo pode ter apenas um valor.

- <EMPRESA>

<NOME FONE1="011-444-9999"

FONE2="011-554-9888">Refinaria

Brasil</NOME>

</EMPRESA>

# Usando Elementos

- <EMPRESA>

<NOME Refinaria Brasil</NOME>

<FONES>

<FONE1>011-444-9999</FONE1>

<FONE2>011-554-9888</FONE2>

</FONES>

</EMPRESA>

# Cont ...

- Atributos não descrevem estruturas.

Certos dados são mais bem descritos usando-se uma estrutura que um valor simples.

# Cont ...

- <peessoa NASCIMENTO='01/06/1980'>  
 <Nome>Armando</Nome>  
 ...  
 ...  
</peessoa>

## Cont ...

- Se usarmos um elemento temos a flexibilidade de descrever:
- `< Pessoa>`
  - `<nascimento>01/06/1980</nascimento>`
  - `<nome>Armando</nome>`
  - ...
  - ...
  - `</Pessoa>`

# Cont ...

- Ou expandindo para:

< Pessoa >

< nascimento >

< data >

< dia >01< /dia >

< mes >06< /mes >

< ano >1980< /ano >

< /data >

< /nascimento >

< nome >Armando< /nome >

...

< /Pessoa >

# Quando usar Atributos

- Não existem regras específicas que determinem quando usar atributos.
- Em geral, os dados devem ser colocados em elementos.
- Informações *a respeito* dos dados (metadados) devem ser colocados em atributos.



Cont ...

- Se estiver em dúvida coloque a informação nos elementos.
- Atributos são adequados para se colocar itens como números de identificação, endereços Web, referências e outras informações que não são diretamente relevantes ao leitor do documento XML.

# Seções CDATA

- São usadas para **inserir blocos de texto**, que podem conter caracteres conflitantes com os usados em marcações.
- **Seções CDATA** iniciam com a cadeia de caracteres “<![CDATA[ ” e terminam com a cadeia de caracteres “ ]]>”.

## Cont ...

- Exemplo:

```
<![CDATA[ <saudacao>Ola a  
    todos</saudacao> ]]>
```

```
<![CDATA[ trecho em JavaScript ]]>
```

# Espaço de Nomes

- *Namespaces* é uma recomendação aprovada pelo W3C pela qual desenvolvedores podem **qualificar nomes de elementos de maneira única na Internet** como um todo, evitando conflitos entre elementos que tenham o mesmo valor.
- Esse risco existe porque XML permite a criação de tags. Diferentes pessoas em lugares diferentes podem criar **os mesmos nomes de tags**.

## Cont ...

- É uma tecnologia que permite declarar que estamos utilizando nomes pertencentes a certo vocabulário, e assim podemos especificá-los de maneira não ambígua no documento.

# Por que usar NameSpaces

- Um dos atrativos de XML é a capacidade de se criar tags que tenham algum significado.

- <livro>

<titulo>A SuiTable Boy</titulo>

<preco moeda="BrReais">22.95</preco>

</livro>

## Cont ...

- Da mesma maneira podemos representar os dados de um autor:

<AUTOR>

<TITULO>Sr</TITULO>

<NOME>William Sheakespeare</NOME>

</AUTOR>

Cont ...

- **Namespaces** resolvem esse problema por meio da associação de um **vocabulário** com uma *tag*.

- Exemplo:

<Livros:TITULO>Hamlet</Livros:TITULO>

<Autor:TITULO>Sr.</Autor:TITULO>



## Cont ...

- O nome que precede uma tag é um **prefixo que se refere a um namespace**, também conhecido como **URI** (Universal Resource Identifier).
- Um URI garante que haverá **unicidade quando dois ou mais documentos XML forem combinados** não existindo conflito.

# Declarando Espaços de Nomes

- Declaração Default

```
<LIVRO xmlns="urn:Biblioteca.org:InformacaoDeLivros" >  
  <TITULO>Hamlet</TITULO>  
  <PRECO moeda="BR Real">22.95</PRECO>  
</LIVRO>
```

Define um nome de espaço para todos os elementos em um dado contexto.

# Declarando Espaços de Nomes

- Declaração Explícita

```
<lv:LIVRO xmlns:lv="urn:Biblioteca.org:InformacaoDeLivros"
  xmlns:dinheiro="urn:Financas:Dinheiro" >
  <lv:TITULO>Hamlet</lv:TITULO>
  <dinheiro:PRECO dinheiro:moeda="BR
  Real">22.95</dinheiro:PRECO>
</lv:LIVRO>
```

# Document Type Definition

- XML é uma linguagem extensível.

Flexibilidade,

Conflitos,

- Ordem em que os elementos ou atributos aparecem nos documentos, se existem elementos opcionais, e quais tipos desses elementos.

# DTD

- Para ajudar a resolver esse problema, criou-se o DTD – Document Type Definition.
- Um DTD define quais as classes possíveis para uma classe de documentos.

# DTD

- Tendo-se um DTD, um documento XML pode ser **validado**, ou seja, podemos confirmar se **o documento segue certa definição**, e a aplicação que vai processar esse documento não precisa se preocupar com erros de estrutura.
- Dessa forma, asseguramos que diferentes pessoas e programas podem tratar informações uns dos outros.

# DTD

- São usados para definir os chamados vocabulários.
- Cada vocabulário estabelece quais elementos são aplicáveis em um determinado domínio de aplicação.
- Cada ramo de atividade que envolve transferência de informação tem um grande potencial para o uso de DTDs.

# Declaração de Tipo de Documento

- Uma declaração de tipo de documento define qual o DTD usado por um documento XML.
- Essa declaração aparece no prólogo de um documento, antes de um elemento-raiz.
- A declaração pode efetuar uma ou as duas das seguintes funções:



## Cont ...

- Conter declarações de markup no prólogo do documento. Assim é definido um DTD interno ao documento.
- Referenciar declarações de markup externas, o que define um DTD externo.

Cont ...

- Quando nos referimos ao DTD de um documento, estamos nos referindo ao externo e ao interno, tomados em conjunto.
- O DTD externo descreve estruturas gerais para uma classe de documentos.
- O DTD interno descreve estruturas específicas para um dado documento.

# Exercícios

- Descreva o que é um documento XML?
- Qual a diferença entre XML e HTML?
- Como podemos mapear uma entidade para um documento XML?
- Mapeie a seguinte entidade para um documento XML bem formado:  
Pessoa (Nome, Endereço, Telefone, DataNascimento)
- O que são XSL e DTD? Para que servem?