

XML

Extensible Markup Language

Professor Vinícius Costa

XML

- Linguagem de marcação extensível
- Desenvolvida para armazenar e transportar dados
- Desenvolvida para ser legível tanto para humanos quanto por máquinas

O que é XML

- É o padrão para linguagem de marcação extensível
- É uma linguagem de marcação semelhante a HTML
- Desenvolvida para armazenar e transportar dados independente de hardware ou software
- Desenvolvida para ser auto descritiva
- É uma recomendação da W3C

O que XML faz?

- XML não faz nada
- XML é apenas informação envolvida por tags
- É preciso escrever algum software para manipular o XML (enviar, receber, armazenar, mostrar, etc)

Exemplo de XML

- `<nota>`
 - `<para>José</para>`
 - `<de>Maria</de>`
 - `<cabecalho>Lembrete</cabecalho>`
 - `<corpo>Lembre-se de comprar o leite!</corpo>`
 - `</nota>`
- Conteúdo do código XML é auto descritivo
 - Uma nota
 - Tem como destino o José
 - Envida por Maria
 - Tem um cabeçalho
 - E um corpo da mensagem

Diferença entre HTML e XML

- São linguagens criadas para propósitos diferentes
 - XML foi desenvolvida para transportar dados com foco no que é o dado
 - HTML foi desenvolvida para apresentar dados com foco em como os dados são apresentados
 - As tags XML não são predefinidas como acontece no HTML

XML não usa tags predefinidas

- XML não tem tags predefinidas
- Tags como no exemplo anterior (<para>, <de>) não são definidas no padrão XML, estas tags foram inventadas pelo autor do documento XML
- As tags HTML são predefinidas pela linguagem (<p>, <h1>, etc)
- O autor de um documento XML deve definir tanto as tags como a estrutura do documento

XML é Extensível

- A maioria das aplicações XML trabalharão com um conjunto esperado de tags
- Mesmo que alguma tag seja inserida ou removida a aplicação poderá continuar funcionando
- Suponha que no exemplo anterior uma tag `<data>` seja inserida e a tag `<cabecalho>` seja removida

Exemplo - código

- `<nota>`
 `<para>José</para>`
 `<de>Maria</de>`
 `<cabecalho>Lembrete</cabecalho>`
 `<corpo>Lembre-se de comprar o leite!</corpo>`
 `</nota>`
- `<nota>`
 `<para>José</para>`
 `<de>Maria</de>`
 `<data>01/01/2000</data>`
 `<corpo>Lembre-se de comprar o leite!</corpo>`
 `</nota>`

Exemplo – possível visualização

Nota

Para: José

De: Maria

Lembrete

Lembre-se de comprar o leite!

Nota

Para: José

De: Maria

01/01/2000

Lembre-se de comprar o leite!

XML simplifica as coisas

- Simplifica o compartilhamento dos dados
- Simplifica o transporte dos dados
- Simplifica a mudança de plataforma
- Simplifica a disponibilidade dos dados

XML simplifica as coisas

- Muitos sistemas computacionais contêm dados em formatos incompatíveis;
- Troca de dados entre sistemas incompatíveis ou atualização do sistema pode consumir muito tempo do desenvolvedores
- XML armazena dados em formato texto puro facilitando o armazenamento, transporte e compartilhamento de dados e garantindo independência de hardware e software
- XML pode estar disponível para diferentes leitores como pessoas, computadores, feeds de notícias, etc.

XML é uma recomendação W3C

- XML é uma recomendação W3C desde 1998

Como usar XML?

- É utilizada em muitos aspectos do desenvolvimento web
- É frequentemente utilizada para separar dados da apresentação

Como usar XML?

- XML separa dados da apresentação
 - XML não guarda qualquer informação sobre como será apresentado
 - Um único documento XML pode ser apresentado em diferentes cenários
 - Existe em XML uma completa separação dos dados e apresentação
- XML é frequentemente um complemento do HTML
 - XML pode ser utilizado junto com HTML
 - XML armazena ou transporta os dados
 - HTML formata e mostra os mesmos dados

Como usar XML?

- Utilizando XML quando os dados mudam você não precisa modificar o HTML
- Os dados podem ser armazenados em arquivos XML separados
- Utilizando JavaScript é possível ler arquivos XML e atualizar dados de qualquer página HTML

Transação de Dados

- Existe vários formatos para transação de dados entre as aplicações
- Exemplo:
 - XMLNews é uma especificação para troca de notícias e outras informações.

Exemplo de documento XMLNews

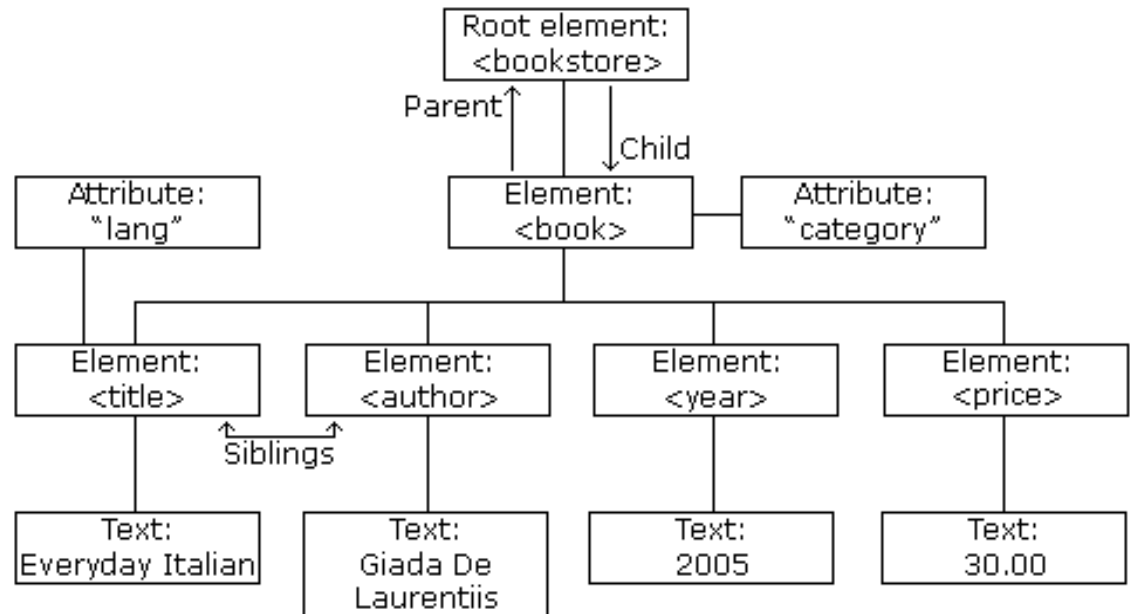
```
<?xml version="1.0" encoding="UTF-8"?>
  <nitf>
    <head>
      <title>Colombia Earthquake</title>
    </head>
    <body>
      <headline>
        <hl1>143 Dead in Colombia Earthquake</hl1>
      </headline>
      <byline>
        <bytag>By Jared Kotler, Associated Press Writer</bytag>
      </byline>
      <dateline>
        <location>Bogota, Colombia</location>
        <date>Monday January 25 1999 7:28 ET</date>
      </dateline>
    </body>
  </nitf>
```

Árvore XML

- Documentos XML formam uma árvore que inicia na raiz e ramifica para as folhas

Árvore XML

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
</bookstore>
```



Estrutura da Árvore XML

- Documentos XML são formados por **element trees**
- Uma árvore XML inicia no **root element** e ramifica para os **child elements**
- Todos os elementos podem ter sub elementos (elementos filhos)

Exemplo de Árvore XML

<raiz>

 <filho>

 <subfilho>

 ...

 </subfilho>

 </filho>

</raiz>

Árvore XML

- Os termos parent (pai), child (filho) e sibling (irmão) são usados para descrever o relacionamento entre os elementos
- Todos os elementos podem ter conteúdo do tipo texto e atributos

Sintaxe do Exemplo

- O prólogo define a versão e codificação de caracteres
 - `<?xml version="1.0" encoding="UTF-8"?>`
- Em seguida vem o elemento raiz
 - `<bookstore>`
- Na sequencia um `<book>` é iniciado
 - `<book category="cooking">`
- O `<book>` tem quatro elementos filhos: title, author, year e price
- Então o livro é finalizado
 - `</book>`
- E por fim o elemento raiz é finalizado
 - `</bookstore>`

Sintaxe do XML

- A sintaxe do XML é simples e lógica tanto para ler quanto para usar
- Todo documento XML deve ter um elemento raiz
 - O elemento raiz é o elemento pai de todos os elementos

Sintaxe do XML

- Prólogo
 - `<?xml version="1.0" encoding="UTF-8"?>`
- É opcional
- Se existir deve ser estar na primeira linha do documento
- Se utilizar uma codificação de caracteres diferente de UTF-8 é aconselhado indicar a codificação no prólogo
- UTF-8 é a codificação padrão no XML

Sintaxe do XML

- Todo elemento XML deve ser fechado
- Omitir a tag de fechamento no XML é ilegal
- Todo elemento aberto obrigatoriamente deve ser fechado
- Tags vazias podem usar a forma abreviada para fechamento
- Exemplo
 - `<p>Ola mundo</p>`
 - `<input type=text />`
- O prólogo do XML não tem tag de fechamento pois não é código XML

Sintaxe do XML

- Tags XML são case sensitive, isto é, maiúsculo é diferente de minúsculo.
 - `<livro> ... </livro>` Correto
 - `<nota> ... </Nota>` Errado

Sintaxe do XML

- As tags devem estar aninhadas
 - `<i>Ola mundo</i>` Correto
 - `<i>Ola mundo</i>` Errado (funciona no HTML)

Sintaxe do XML

- Valores dos atributos devem estar entre aspas
 - `<peessoa nome="Jose"> ... </peessoa>`

Sintaxe do XML

- Entidade
 - Alguns caracteres tem significado especial no XML
 - Um sinal de < dentro de um elemento XML irá gerar um erro pois o parser (interpretador) interpretará o sinal de "maior que" como o início de um novo elemento.
 - Exemplo errado:
 - <expressao> x < 10 </expressao>
 - Para não gerar erro utilize **entity reference**
 - Exemplo
 - <expressao> x < 10</expressao>

Sintaxe do XML

- Algumas entidades predefinidas no XML

&lt;	<	less than
&gt;	>	greater than
&amp;	&	ampersand
&apos;	'	apostrophe
&quot;	"	quotation mark

Sintaxe do XML

- Comentários
 - <!-- tudo que vier aqui é comentário -->
- Espaços em branco são preservados
 - No HTML vários espaços em branco são transformados em apenas 1 espaço em branco, isto não acontece no XML

Sintaxe do XML

- Nova linha no XML é LF
 - Nova linha em aplicações Windows é composta por carriage return e line feed (CR + LF)
 - Unix e Mac OS usam LF
 - Mac antigos usam CR
 - XML usa LF para novas linhas

Sintaxe XML

- Documento XML que seguem as regras de sintaxe informada nos slides anteriores são chamados de documentos XML bem formatados

Elementos XML

- Um documento XML contém elementos XML
- Um elemento XML é tudo desde a tag inicial até a tag final, incluindo as tags.
- Um elemento pode conter
 - Texto
 - Atributos
 - Outros elementos
 - Uma mistura dos itens acima

Elementos XML

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
</bookstore>
```

- <title>, <author>, <year> e <price> tem conteúdo do tipo texto
- <bookstore> e <book> tem elementos como conteúdo
- <book> tem o atributo category

Elementos XML

- Elementos XML vazios
- Elementos sem conteúdo são elementos vazios
- Exemplo
 - `<input type=text></input>`
 - `<input type=text />`

Regras para nomes de elementos XML

- O nome dos elementos
 - São case sensitive
 - Deve começar com letras ou sublinhado
 - Não pode começar com xml (ou XML, ou Xml, etc)
 - Pode conter letras, números, hífen, sublinhado e pontos
 - Não pode conter espaços

Boas práticas para nomes de elementos XML

- Utilize nomes intuitivos: <pessoa>, <nome>, <idade>, etc
- Utilize nomes curtos: <primeiro_nome> ao invés de <o_primeiro_nome_é>
- Evite sinal de menos (-). Alguns softwares podem entender isto como subtração.
- Evite ponto final (.). Alguns softwares podem entender isto como propriedade de algum objeto
- Evite dois pontos (:). São utilizados com namespaces
- Evite utilizar palavras com acento ou cedilha (á, é, ç, etc). Pode surgir problemas com softwares que não suportam estes caracteres

Atributos XML

- Os atributos devem obrigatoriamente ter um valor atribuído
- Atributos contém dados relacionados ao elemento
- Os valores dos atributos devem estar entre aspas ou apóstrofos

Exemplo de atributos XML

- `<peessoa profissao="Professor">`
- `<peessoa profissao='Professor'>`
- Um atributo pode conter aspas ou apostrofos
- `<peessoa nome="Vinícius 'professor' Costa">`
- Ou utilize entidades
- `<peessoa nome="Vinícius
"professor" Costa">`

Exemplo de atributos XML

```
< Pessoa idade="30">  
  < nome>Jose</ nome>  
  < sobrenome>Carlos</ sobrenome>  
</ Pessoa>
```

```
< Pessoa>  
  < idade>30</ idade>  
  < nome>Jose</ nome>  
  < sobrenome>Carlos</ sobrenome>  
</ Pessoa>
```

Exemplo de atributos XML

```
< Pessoa endereco="310 sul, alameda 1, lote 100, Palmas">  
  < nome>Jose</ nome>  
  < sobrenome>Carlos</ sobrenome>  
</ Pessoa>
```

```
< Pessoa>  
  < endereco>  
    310 sul, alameda 1, lote 100, Palmas  
  </ endereco>  
  < nome>Jose</ nome>  
  < sobrenome>Carlos</ sobrenome>  
</ Pessoa>
```

Exemplo de atributos XML

```
< Pessoa >  
  < endereco >  
    < logradouro >  
      310 sul, alameda 1  
    < /logradouro >  
    < lote >  
      100  
    < /lote >  
    < cidade >  
      Palmas  
    < /cidade >  
  < /endereco >  
  < nome >Jose< /nome >  
  < sobrenome >Carlos< /sobrenome >  
< / Pessoa >
```

Atributos ou Elementos

- Atributos não podem conter múltiplos valores
- Atributos não podem conter uma estrutura em árvore
- Atributos são difíceis de serem estendidos para atualizações futuras no código

Atributos ou Elementos

- Atributos são uma ótima ferramenta para identificar um elemento
- Atributos são aconselhados para serem utilizados como metadados (dados sobre dados)
- Por exemplo, o "codigo" não faz parte de uma pessoa mas é um dado que auxilia a identificar a pessoa
- `< Pessoa codigo="01">`
- `<p id="primeiro">`

Namespace

- XML Namespace serve para evitar conflito de nomes
- As tags são definidas pelo desenvolvedor
- Em alguns casos é preciso utilizar vários documentos XML desenvolvido por pessoas diferentes
- Por exemplo, suponha uma aplicação que tenha uma tabela HTML (table) e uma mesa (table em inglês)

Sem o uso de Namespace – conflito de nomes

```
<table>
  <tr>
    <td>Apples</td>
    <td>Bananas</td>
  </tr>
</table>
```

```
<table>
  <name>African Coffee
  Table</name>
  <width>80</width>
  <length>120</length>
</table>
```

Namespace

- Para resolver o conflito de nomes utiliza-se namespace
- O namespace consiste em utilizar um prefixo
- Exemplo
 - `<h:table>`
 - `<f:table>`

Namespace

```
<h:table>
```

```
  <h:tr>
```

```
    <h:td>Apples</h:td>
```

```
    <h:td>Bananas</h:td>
```

```
  </h:tr>
```

```
</h:table>
```

```
<f:table>
```

```
  <f:name>African Coffee Table</f:name>
```

```
  <f:width>80</f:width>
```

```
  <f:length>120</f:length>
```

```
</f:table>
```

Namespace - Atributo xmlns

- Para que um namespace funcione é preciso defini-lo
- Para definir um namespace
 - utilize o atributo **xmlns** na tag de início do elemento
 - Declare no elemento raiz do documento XML
- Sintaxe:
xmlns:prefixo="URI"

Exemplo do uso do atributo xmlns

```
<root>
```

```
  <h:table xmlns:h="http://www.w3.org/TR/html4/">
```

```
    <h:tr>
```

```
      <h:td>Apples</h:td>
```

```
      <h:td>Bananas</h:td>
```

```
    </h:tr>
```

```
  </h:table>
```

```
  <f:table xmlns:f="https://www.w3schools.com/furniture">
```

```
    <f:name>African Coffee Table</f:name>
```

```
    <f:width>80</f:width>
```

```
    <f:length>120</f:length>
```

```
  </f:table>
```

```
</root>
```

Exemplo do uso do atributo xmlns

- Quando um namespace é definido para um elemento todos os seus elementos filhos com o mesmo prefixo serão associados ao namespace

Exemplo do uso do atributo xmlns

```
<root xmlns:h="http://www.w3.org/TR/html4/"
xmlns:f="https://www.w3schools.com/furniture">
  <h:table>
    <h:tr>
      <h:td>Apples</h:td>
      <h:td>Bananas</h:td>
    </h:tr>
  </h:table>

  <f:table>
    <f:name>African Coffee Table</f:name>
    <f:width>80</f:width>
    <f:length>120</f:length>
  </f:table>
</root>
```

Namespace

- A URI não é observada pelo parser
- O intuito da URI é dar um nome único ao namespace
- A URI é frequentemente utilizada como um indicador de página web contendo informações sobre o namespace

Default Namespace

- Namespace padrão serve para evitar de utilizar o prefixo em todos os elementos filhos
- Sintaxe
 - `xmlns="namespaceURI"`

Default Namespace

```
<table xmlns="http://www.w3.org/TR/html4/">  
  <tr>  
    <td>Apples</td>  
    <td>Bananas</td>  
  </tr>  
</table>
```

```
<table xmlns="https://www.w3schools.com/furniture">  
  <name>African Coffee Table</name>  
  <width>80</width>  
  <length>120</length>  
</table>
```

Exercício

Nome	E-mail	Data de Nascimento	Telefones
José Silva	josesilva@gmail.com	10/12/2000	98123 4321 3223 3434
Mônica Aparecida	moaparecida@hotmail.com	03/04/1998	3225 5123 9999 2222 9834 9451
Joaquim Gomes	jgomes@yahoo.com.br	25/08/1980	3323 2242 98413 3242
Ana Soares	anasoares@outlook.com	29/02/1990	3234 2345 99841 9896
Paulo Gustavo	paulogustavo@gmail.com	21/09/2002	3276 6578 97898 1390
Carlos Alfredo	Calfredo@uol.com	07/03/1987	3134 7900 99701 9718
Maria Sônia	marsonia@outlook.com	15/10/1972	3511 9712 98123 9866

Bibliografia

- <https://www.w3schools.com/xml/>