

XML

- Um documento XML possui
 - Unidade lógica - os elementos
 - Usuário "*inventa*" as marcas através de DTDs
 - Unidade física - as entidades
 - Armazenamento separado dos dados
- Como toda linguagem de marcação:
 - XML não FAZ NADA!!!!
 - É necessário um *software* para manipular os dados XML

XML

- Processador XML
 - Gerenciador de Entidades
 - Software responsável pela localização das entidades utilizadas em um documento XML
 - O *parser*
 - Software responsável por fazer a validação de um documento XML.
 - Parser sem validação
 - **Documento bem formado.** O software valida a sintaxe do documento XML
 - Parser com validação
 - **Documento válido.** O software valida o documento XML contra a DTD construída

Estruturação

- É possível definir regras que expressam como os elementos podem ser combinados (estrutura hierárquica)
 - DTD (*Document Type Definition*)
 - XMLSchema

Instruções especiais

- Um documento pode conter instruções especiais para o processador XML.
- Não é o texto útil do documento nem as *tags*
 - Declarações:
 - **Declarações de elementos**
 - **Declarações de comentários**
 - **Declarações de seção *character data***
 - Instruções:
 - **Instruções de processamento**
 - **Instrução especial XML**

Comentário

- Identificado pela símbolos reservados "
-- "

- Comentário bem formado
`<!-- Este é um comentário -->`

- Comentário com erro

`<!-- Este é um comentário -- ILEGAL- - -->`

Seção *character data*

- Em um segmento de texto *character data* os caracteres de marcação não possuem relevância
 - Possibilitam o uso de *caracteres de marcação especiais* no texto
 - Delimitadas por "`<![CDATA["` e "`"]>`"
 - Exemplo:

`<![CDATA[Um exemplo do elemento: <titulo>]]>`

Resultado depois do documento ser processado:

`"Um exemplo do elemento: <titulo>"`

Instruções de processamento

- Contém informação requerida por uma aplicação específica
- Não é especificada por uma declaração de marcação
- Delimitada pelos caracteres " <? " e " ?> "
- Exemplo

<para>

seria bom finalizar esta pagina

<?ACME- paginator DO: new- page?>

<pagebreak/> aqui.

</ para>

Instrução especial XML

- É uma instrução de processamento especial

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

- Parâmetros

- `version` indica a versão do padrão XML usado no documento em questão (atualmente, existe somente uma versão) - **obrigatório**
- `encoding` indica o código de caracteres utilizado no documento - **opcional**
 - cada codificação pode representar um número "x" de caracteres
 - Por *default*, a codificação "UTF-8" é utilizada. Outros valores possíveis para o atributo *encoding* são:
 - UTF-16, ISO-10646, UCS-2, UCS-4, US_ASCII, entre outros.
 - Para uso de caracteres latin: ISO-8859-1

Instrução especial XML

- É uma instrução de processamento especial

```
<?xml version="1.0" encoding="UTF-8"
standalone="yes"?>
```
- Parâmetros
 - standalone - **opcional**
 - = "yes", indica que não existem declarações externas que afetam a interpretação (*default*)
 - = "no", indica que um conjunto de declarações definido externamente contém informação que afeta a interpretação do conteúdo do documento. O valor "no" deve ser usado se qualquer elemento, atributo ou entidade externa for definida em uma DTD externa

Instrução especial XML

- Instrução de processamento especial

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

Convenção: usar esta linha no início de todo documento XML.

Caso a instrução for usada, o atributo version="1.0" é obrigatório

Definição da estrutura

- DTD - *Document Type Definition*
 - *Gramática Regular*
 - Conjunto de regras que definem a estrutura do documento
 - Hierarquia e granularidade
 - Elementos que podem ser usados
 - Onde podem ser aplicados (uns em relação aos outros)
- *XMLSchema*
 - Esquema escrito em XML

Exemplos de DTD

<ul style="list-style-type: none">- registros médicos- enciclopédias- catálogos de produto- jornais- propostas- dicionários- relatórios técnicos- contratos	<ul style="list-style-type: none">- políticas e procedimentos- artigos científicos- livros- ordens de compra- recibos / notas- revistas- normas / regulamentos- manuais de usuário- normas de referência
--	--

Estrutura da DTD

- Composta por declarações. Utiliza a sintaxe

<! >

- Definição de elemento

<!ELEMENT . . . >

- Definição de atributos

<!ATTLIST . . . >

```
<!ELEMENT email (de, para, assunto, corpo)>
<!ATTLIST email data CDATA #REQUIRED>
<!ELEMENT de (#PCDATA)>
<!ELEMENT para (#PCDATA)>
<!ELEMENT assunto (#PCDATA)>
<!ELEMENT corpo (#PCDATA)>
<!ATTLIST corpo fonte CDATA #REQUIRED>

<email data='20.12.2005'>
  <de>dorneles@upf.br</de>
  <para>undisclosed-recipients@...</para>
  <assunto>reuniao</assunto>
  <corpo fonte='Texto plano'>
    Não esqueçam da reuniao de amanha
  </corpo>
</email>
```

Definição de Elementos

- Definir nome e especificar seu conteúdo

- Contendo outros elementos dentro:

```
<!ELEMENT capitulo (titulo, secao)>
```

- Contendo texto:

```
<!ELEMENT titulo (#PCDATA)>  
#PCDATA ( Parsable Character Data)
```

- Elemento Vazio:

```
<!ELEMENT figura EMPTY>
```

Elemento vazio

- Elemento cujo conteúdo é dado pelo nome do próprio elemento, ou por atributos definidos a ele

```
<peessoa>  
  <nome>Aninha</nome>  
  <sexo><f/></sexo>  
</peessoa>
```

Deve haver uma
/ antes de fechar o
elemento

ou

```
<figura arquivo="logo" formato="GIF"/>
```


Conteúdo Misto

- Quando um elemento possui conteúdo misturado:

```
<resumo>
```

```
  Este trabalho apresenta uma alternativa  
  para o
```

```
  <palavraChave>armazenamento</palavraChave>  
  de documentos XML (e linguagens análogas e  
  <ingles>mark-up</ingles>) em ...
```

```
</resumo>
```

Conteúdo Misto

- Quando um elemento possui conteúdo misturado:

```
<!ELEMENT resumo  
  (#PCDATA|palavraChave|ingles) *>
```

Conteúdo Misto

- Existe uma regra importante que deve ser seguida quando existe escolha entre texto e elementos filho.
 - **#PCDATA** deve ser obrigatoriamente o primeiro *token* no grupo, e o grupo deve ser de **escolha**, **opcional** e **com repetição**.
 - Ex.:

```
<!ELEMENT paragrafo (#PCDATA | Subpar | Superpar) *>
```

Definição de Atributos

- Definir o elemento
- Definir, para um elemento, os seus atributos
 - Parâmetros:
 - nome do elemento
 - nome de atributo
 - tipo de atributo
 - palavra reservada (#REQUIRED ou #IMPLIED)

```
<!ELEMENT email (de, para, assunto, corpo)>  
<!ATTLIST email data CDATA #REQUIRED  
             hora CDATA #REQUIRED>
```

Exemplos

- Valor default:

DTD:

```
<!ELEMENT quadrado EMPTY>
```

```
<!ATTLIST quadrado lado CDATA "0">
```

XML:

```
<quadrado lado="100" />
```

Exemplos

DTD:

```
<!ATTLIST pagamento forma  
(cheque|dinheiro) "dinheiro">
```

XML:

```
<pagamento forma="cheque" />
```

ou

```
< pagamento forma = " dinheiro " />
```

Exemplo 1 - Livros

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

<!DOCTYPE LIVROS [
  <!ELEMENT LIVROS ANY>
  <!ELEMENT LIVRO (TITULO, AUTOR)>
  <!ELEMENT TITULO (#PCDATA)>
  <!ELEMENT AUTOR (#PCDATA)>
]>

<LIVROS>
  <LIVRO>
    <TITULO>XML Como Programar</TITULO>
    <AUTOR>Deitel e Deitel</AUTOR>
  </LIVRO>
</LIVROS>
```

Exemplo 2 - Veículos

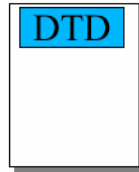
```
<?xml version="1.0"?>

<!DOCTYPE VEICULO [
  <!ELEMENT VEICULO ANY>
  <!ELEMENT PARTES (MOTOR, ACESSORIOS)>
  <!ELEMENT MOTOR (MECANICA, ELETRICA)>
  <!ELEMENT MECANICA (PARAFUSO)>
  <!ELEMENT ELETRICA (#PCDATA)>
  <!ELEMENT PARAFUSO (#PCDATA)>
  <!ELEMENT ACESSORIOS (#PCDATA)>
]>

<VEICULO>
  <PARTES>
    <MOTOR>
      <MECANICA>
        <PARAFUSO>Ferro</PARAFUSO>
      </MECANICA>
      <ELETRICA>Fio de Cobre</ELETRICA>
    </MOTOR>
    <ACESSORIOS>Vidro Eletrico, MP3 Player</ACESSORIOS>
  </PARTES>
</VEICULO>
```

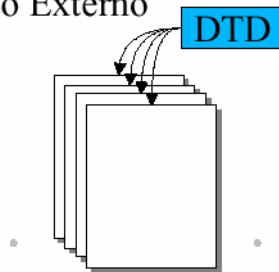
Declaração da DTD no XML

Subconjunto Interno



DTD interna. Definida dentro do próprio documento XML

Subconjunto Externo



DTD externa. Definida fora do documento XML, como um arquivo fisicamente separado (.dtd)

Subconjunto interno – no arquivo .xml

arquivo.xml

```
<?xml version="1.0"?>
<!DOCTYPE artigo [
  <!ELEMENT artigo (titulo, autor)>
  <!ELEMENT titulo (#PCDATA)>
  <!ELEMENT autor (nome)>
  <!ELEMENT nome (#PCDATA)>
]>
<artigo>
  <titulo> ... .. </titulo>
  <autor>
    <nome> ... .. </nome>
  </autor>
</artigo>
```

Subconjunto externo – arquivos físicos separados

artigo.xml

```
<?xml version="1.0"?>
<!DOCTYPE artigo SYSTEM "artigo.dtd">
<artigo>
  <titulo> ... .. </titulo>
  <autor>
    <nome> ... .. </nome>
  </autor>
</artigo>
```

artigo.dtd

```
<!ELEMENT artigo (titulo, autor)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT autor (nome)>
<!ELEMENT nome (#PCDATA)>
```

DTD Externo

```
<!DOCTYPE VEICULO [
  <!ELEMENT VEICULO ANY>
  <!ELEMENT PARTES (MOTOR, ACESSORIOS)>
  <!ELEMENT MOTOR (MECANICA, ELETRICA)>
  <!ELEMENT MECANICA (PARAFUSO)>
  <!ELEMENT ELETRICA (#PCDATA)>
  <!ELEMENT PARAFUSO (#PCDATA)>
  <!ELEMENT ACESSORIOS (#PCDATA)>
]>
```

Arquivo "dtdaula2.dtd"

Exemplo 2 - Veículos

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

<!DOCTYPE VEICULO SYSTEM "dtdaula2.dtd">

<VEICULO>
  <PARTES>
    <MOTOR>
      <MECANICA>
        <PARAFUSO>Ferro</PARAFUSO>
      </MECANICA>
      <ELETRICA>Fio de Cobre</ELETRICA>
    </MOTOR>
    <ACESSORIOS>Vidro Eletrico, MP3 Player</ACESSORIOS>
  </PARTES>
</VEICULO>
```

Exercício

- Usando o documentos XML criado para log de acesso defina a sua DTD.
 - Defina como DTD externa

Conectores - seqüência e escolha

- Conector de seqüência.

- `<!ELEMENT artigo (titulo, autor, secao)>`

- Conector de escolha.

- `<!ELEMENT endereco (instituicao | cidade)>`

Ou exclusivo, significa **um** **ou** **outro**.

Ou instituicao **OU** cidade

Conectores - seqüência e escolha

- Ao misturar os conectores de escolha, deve-se usar "**()**". Por exemplo:

`<!ELEMENT palavra (a, (b|c))>`

- Para não confundir o *parser*

Controle de quantidade

- Elemento opcional sem repetição: "?"

```
<!ELEMENT autor (primnome, segnome?, tercnome)>
```

- Elemento requerido e com repetição "+"

```
<!ELEMENT capitulo (titulo, paragrafo+)>
```

- Elemento opcional e com repetição "*"

```
<!ELEMENT autor (nome, email*)>
```

Estudo de Caso: Uma Livraria

- Exemplo prático de uma livraria
- Os livros poderão ter ISBN, titulo, autor, editora, ano, numero de paginas e sinopse
- Um ano pode existir ou não, mas se existir, apenas um
- Pode possuir um ou mais autores
- Um autor se divide em nome e sobrenome

```

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE livraria [
  <!ELEMENT livraria (titulo,autor+,editora,ano?,paginas,sinopse)>
  <!ELEMENT autor (nome,sobrenome)>
  <!ELEMENT titulo (#PCDATA)>
  <!ELEMENT nome (#PCDATA)>
  <!ELEMENT sobrenome (#PCDATA)>
  <!ELEMENT editora (#PCDATA)>
  <!ELEMENT ano (#PCDATA)>
  <!ELEMENT paginas (#PCDATA)>
  <!ELEMENT sinopse (#PCDATA)>
]>
<livraria>
  <livro isbn="8536301473">
    <titulo>XML: Como Programar </titulo>
    <autor>
      <nome>Deitel</nome>
      <sobrenome>Harvey</sobrenome>
      <nome>Paul</nome>
      <sobrenome>Deitel</sobrenome>
    </autor>
    <editora>Bookman</editora>
    <ano>2003</ano>
    <paginas>974</paginas>
    <sinopse>O novo livro dos maiores autores de livros didáticos de linguagem de programação examina cuidadosamente o desenvolvimento de sistemas baseados em XML, incluindo programação de aplicativos de múltiplas camadas, aplicativos cliente-servidor, aplicativos orientados a banco de dados e para a Internet e a World Wide Web. Em XML: Como Programar, os Deitels e seus colegas, Tem R. Nieto, Ted Lin e Praveen Sadhu, discutem tópicos indispensáveis para quem precisa construir completos sistemas funcionais baseados em XML.</sinopse>
  </livro>
  <livro isbn="8524105909">
    <titulo>Projeto de Banco de Dados</titulo>
    <autor>
      <nome>Carlos</nome>
      <sobrenome>Heuser</sobrenome>
    </autor>
    <editora>Sagra Luzzatto</editora>
    <paginas>202</paginas>
    <sinopse>Além de tratar o tema do projeto de banco de dados, na forma clássica, que aparece na maioria dos textos de banco de dados, o livro aborda o problema de engenharia reversa de banco de dados, apresentando uma técnica para projeto de banco de dados a partir de sistemas legados. Não se destina somente ao uso acadêmico, mas também aos profissionais de Informática. São feitas muitas relações com problemas práticos e o estilo é tal que permite fácil compreensão por profissionais. Já foi utilizado em cursos de curta-duração para treinamento de profissionais em empresas, além de ser adotado em diversas Universidades. A obra traz uma extensa lista de exercícios e muitos deles estão resolvidos. Este é o quarto volume da Série Livros Didáticos.</sinopse>
  </livro>
</livraria>

```

Exercício

- Usando o documento XML criado para a receita de bolo, defina a sua DTD.
 - Observe o uso dos conectores
 - Observe o controle de quantidade
 - Defina o fato da receita ser doce ou salgada através de um elemento vazio
 - Defina a cobertura como elemento misto
 - Defina como DTD externa

Validação de documentos

- Um documento construído de acordo com as regras gerais de XML é
 - bem-formado (*well-formed*)
- Um documento construído de acordo com as regras de uma DTD é
 - válido (*valid*)

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> podem conter apenas valores positivos em alguma moeda corrente

Limitações de XML

- XML suporta a especificação de documentos estruturados
- Não trata os aspectos de apresentação e reutilização de seu conteúdo
- Não trata o suporte a estruturas hipertexto