

eXtensible Markup Language (XML)

- XML é uma linguagem de anotação (markup).
- HTML utiliza tags para formatação (ex: “bold”).
- XML utiliza tags para descrever informação.
- Em XML, os tags não são pré-definidos. Temos de definir os nossos tags.
- XML utiliza um “Document Type Definition” (DTD) ou um XML Schema para descrever a informação.

XML versus HTML

- XML não é um substituto para o HTML.
- XML foi desenhado para descrever informação.
- HTML foi desenhado “adhoc” (serve para descrever mas também para visualizar informação).
- XML é independente da plataforma.
- HTML tem a vantagem de ser muito simples, facilitando o crescimento rápido da web.

Documentos XML bem formados

- Documento deve começar com uma declaração entre `<? ... ?>`.

- Exemplo:

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

– “Standalone” = “sem DTD”

- Documento tem um só elemento raiz.
- Os outros elementos estão “well nested”.

Exemplo de documento bem formado

```
<?xml version="1.0" standalone="yes"?>
<RECEITAS>
  <TITULO>Livro de receitas de F. Lobo</TITULO>
  <RECEITA>
    <NOME>Bolo de chocolate</NOME>
    <INGREDIENTE>500g de farinha</INGREDIENTE>
    <INGREDIENTE>200g de açucar</INGREDIENTE>
    <INGREDIENTE>300g de manteiga</INGREDIENTE>
    <INGREDIENTE>1 tablete de chocolate</INGREDIENTE>
  </RECEITA>
</RECEITAS>
```

Document Type Definitions (DTDs)

- Trata-se de uma gramática para descrever os tags XML.

```
<!DOCTYPE <tag raiz> [  
  <!ELEMENT <nome> ( <componentes> )  
  <mais elementos>  
>
```

Elementos em DTDs

- A descrição de um elemento consiste num nome (tag), e na descrição dos seus elementos constituintes.
 - inclui ordem e multiplicidade.
- As folhas são elementos de texto e têm #PCDATA (Parsed Character Data)

Exemplo de DTD para receitas

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE RECEITAS [
  <!ELEMENT RECEITAS      (TITULO,RECEITA*)>
  <!ELEMENT TITULO        (#PCDATA)>
  <!ELEMENT RECEITA       (NOME,INGREDIENTE*)>
  <!ELEMENT NOME          (#PCDATA)>
  <!ELEMENT INGREDIENTE  (#PCDATA)>
]>
```

Descrição de elementos

- “Subtags” devem aparecer pela ordem especificada.
- No final do tag coloca-se um símbolo que indica a multiplicidade.
 - * = zero ou mais.
 - + = um ou mais.
 - ? = zero ou um.
- O símbolo | tem o significado de “ou” (permite especificar uma sequência alternativa de tags).

Exemplo

- Um nome contém um título opcional (ex: “Prof.”), um nome próprio, e um apelido (tudo por esta ordem), ou então, contém um endereço IP.

```
<!ELEMENT NOME (  
    (TITULO?, NOMEPROPRIO, APELIDO) | IPADDR  
)>
```

Utilização de DTD's

1. Colocar STANDALONE = "no" .

2. e depois:
 - (a) Incluir o DTD no preâmbulo do documento XML, ou

 - (b) Colocar DOCTYPE <elemento raiz> SYSTEM "path para o ficheiro que contem o DTD" .

Exemplo (a)

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE RECEITAS [
  <!ELEMENT RECEITAS      (TITULO,RECEITA*)>
  <!ELEMENT TITULO        (#PCDATA)>
  <!ELEMENT RECEITA      (NOME,INGREDIENTE*)>
  <!ELEMENT NOME          (#PCDATA)>
  <!ELEMENT INGREDIENTE  (#PCDATA)>
]>
<RECEITAS>
  <TITULO>Livro de receitas de F. Lobo</TITULO>
  <RECEITA>
    <NOME>Bolo de chocolate</NOME>
    <INGREDIENTE>500g de farinha</INGREDIENTE>
    <INGREDIENTE>200g de açucar</INGREDIENTE>
    <INGREDIENTE>300g de manteiga</INGREDIENTE>
    <INGREDIENTE>1 tablete de chocolate</INGREDIENTE>
  </RECEITA>
</RECEITAS>
```

Exemplo (b)

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE RECEITAS SYSTEM "receitas.dtd">
<RECEITAS>
  <TITULO>Livro de receitas de F. Lobo</TITULO>
  <RECEITA>
    <NOME>Bolo de chocolate</NOME>
    <INGREDIENTE>500g de farinha</INGREDIENTE>
    <INGREDIENTE>200g de açucar</INGREDIENTE>
    <INGREDIENTE>300g de manteiga</INGREDIENTE>
    <INGREDIENTE>1 tablete de chocolate</INGREDIENTE>
  </RECEITA>
</RECEITAS>
```

XML

- XML é usado para trocar informação.
- XML é usado para criar novas linguagens (ex: WAP/WML).

Relações entre elementos XML

- Um documento XML pode ser representado por uma árvore.
- Elementos têm relações entre si (pai, filho, irmão, ...).

Exemplo de documento XML

```
<book>
<title>My First XML</title>
<prod id="33-657" media="paper"></prod>

<chapter>Introduction to XML
<para>What is HTML</para>
<para>What is XML</para>
</chapter>

<chapter>XML Syntax
<para>Elements must have a closing tag</para>
<para>Elements must be properly nested</para>
</chapter>

</book>
```

Relações entre elementos XML

- book é o elemento raiz.
- title, prod e chapter são filhos de book.
- book é o elemento pai de title, prod e chapter.
- title, prod e chapter são irmãos.

Elementos têm conteúdo

- O conteúdo de um elemento XML é tudo o que está compreendido entre os tags de abertura e fecho, incluindo os próprios tags de abertura e fecho.
- Os elementos podem ter diferentes tipos de conteúdo.
- Um elemento pode ter como conteúdo,
 - outros elementos
 - conteúdo misto
 - conteúdo simples
 - conteúdo vazio

No exemplo ...

- `book` tem como conteúdo outros elementos.
- `chapter` tem conteúdo misto porque contém texto juntamente com outros elementos.
- `para` tem conteúdo simples (ou de texto).
- `prod` tem conteúdo vazio.

Elementos podem ter atributos

- prod tem 2 atributos: id e media.
- O valor dos atributos têm de vir entre aspas (ou plicas).

```
id tem o valor "33-657"  
media tem o valor "paper"
```

- O mesmo acontece em HTML:

```

```

Atributos versus Elementos

- Informação pode ser guardada como elemento ou como atributo.

```
<person sex="female">  
  <firstname>Anna</firstname>  
  <lastname>Smith</lastname>  
</person>
```

ou:

```
<person>  
  <sex>female</sex>  
  <firstname>Anna</firstname>  
  <lastname>Smith</lastname>  
</person>
```

Limitações sobre atributos

- Atributos não podem conter vários valores (mas os elementos podem)
- Atributos não descrevem estrutura.
- Atributos são mais difíceis de manipular através de programas.

Atributo ou elemento?

- Trata-se de um problema de análise e sem resposta exacta.
- Princípios que poderão ajudar a decidir:
 - Se a informação é estrutural deve usar-se um elemento.
 - Se a informação requer um processamento especial deve também usar-se um elemento.
 - Se a informação apenas qualifica o conteúdo deve usar-se um atributo.

Atributo ou elemento?

- Trata-se de um problema de análise e sem resposta exacta.
- Princípios que poderão ajudar a decidir:
 - Se a informação é estrutural deve usar-se um elemento.
 - Se a informação requer um processamento especial deve também usar-se um elemento.
 - Se a informação apenas qualifica o conteúdo deve usar-se um atributo.

Validação de documentos XML

- Um documento válido deve respeitar um DTD ou um XML Schema.
- Um DTD define um padrão de estrutura para um documento XML.
- Um XML Schema é uma alternativa ao DTD e é totalmente definido em XML (ao contrário de um DTD).

Visualização de ficheiros XML

- Os ficheiros XML podem ser visualizados no IE 5.0 (ou superior) e no Netscape 6
- ... mas tem de se fazer algo mais para que o resultado seja uma página web bonita.
- Pode-se visualizar XML usando CSS (pouco comum).
- Pode-se visualizar XML usando XSL (mais comum).

Namespaces

- Pode haver conflito de nomes quando se junta 2 ou mais ficheiros XML.
- Os namespaces são utilizados para evitar conflito de nomes de elementos.

Exemplo

Ficheiro 1:

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

Ficheiro 2:

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

Namespaces (cont.)

- Um namespace é uma colecção de nomes identificados por uma referência URI.
- É usada nos documentos XML como prefixo dos nomes de elementos e atributos.
- Coloca-se o atributo `namespace` no início do tag de um elemento. Sintaxe:

```
xmlns:namespace-prefix="namespace"
```

- Exemplo:

```
xmlns:f="http://www.w3schools.com/furniture"
```

Eliminar conflito com namespaces

Ficheiro 1:

```
<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>
```

Ficheiro 2:

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

Namespaces (cont.)

- Quando um namespace é definido no início de um tag de um elemento, todos os elemento filho com o mesmo prefixo ficam associados ao namespace.
- Pode usar-se um namespace por defeito para os elemento filho. Exemplo:

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

XML Parsing

- O conteúdo de um documento XML é sujeito a um parser.
- Quando se faz o parsing de um elemento XML, o texto que está compreendido entre os tags também é sujeito a parsing.
- Isto acontece porque pode haver elementos dentro de elementos.

Caracteres especiais

- Tal como em HTML, alguns caracteres especiais têm de ser transformados em referências a entidades.

<	---	<
>	---	>
&	---	&
'	---	'
"	---	"

XML CDATA

- Tudo o que estiver dentro de uma secção CDATA (Character Data) é ignorado pelo parser.
- Util para texto com muitos caracteres especiais. Ex: Prima a tecla <<ENTER>>.

Prima a tecla <<ENTER>>.

```
<![CDATA[Prima a tecla <<ENTER>>.]>
```