

Introdução à XML

A XML foi projetada para descrever dados e para enfatizar o que os dados são.

A HTML foi projetada para exibir dados e para enfatizar como os dados aparecem.

O Que Você Já Deve Saber

Antes de continuar, você deve ter algum conhecimento básico sobre o seguinte:

- WWW, HTML e os fundamentos da construção de páginas Web
- Linguagens de scripting Web como JavaScript ou VBScript

Se você quiser estudar estes assuntos primeiro, antes de começar a ler sobre XML, você pode encontrar os tutoriais necessários na [Página da W3Schools](#).

O Que É XML?

- XML significar **EX**tensible **M**arkup **L**anguage (Linguagem de Marcação Extensível)
 - XML é uma **linguagem de marcação** bem como a HTML
 - XML foi projetada para **descrever dados**
 - As tags XML não são pré-definidas. Você deve **definir suas próprias tags**
 - A XML usa uma **Document Type Definition** (DTD) ou um **XML Schema** para descrever os dados
 - A XML com uma DTD ou um XML Schema é designada para ser **auto-descritiva**
-

As principais diferenças entre XML e HTML

A XML foi projetada para transportar dados.

A XML não é um substituto para a HTML.

A XML e a HTML foram projetadas com objetivos diferentes:

A XML foi projetada para descrever dados e enfatizar o que os dados são.

A HTML foi projetada para exibir os dados e enfatizar como os dados aparecem.

A HTML está relacionada com a exibição de informação, enquanto a XML está relacionada com a descrição de informação.

A XML nada FAZ

A XML não foi projetada para FAZER algo.

Talvez seja um pouco difícil de entender, mas a XML nada faz. A XML foi criada para estruturar, armazenar e enviar informação.

O seguinte exemplo é um bilhete para José de Maria, armazenada como XML:

```
<bilhete>
<para>José</para>
<de>Maria</de>
<título>Lembrete</título>
<corpo>Não me esqueça neste fim-de-semana!</corpo>
</bilhete>
```

O bilhete tem um título e um corpo de mensagem. Ele também tem informação sobre o remetente e o receptor. Mas ainda assim, este documento XML nada faz. Ele é só informação pura envolta em tags XML. Alguém deve escrever uma peça de software para enviá-la, recebê-la ou exibi-la.

A XML é livre e extensível

As tags XML não são pré-definidas. Você deve "inventar" suas próprias tags.

As tags usadas para marcar documentos HTML e a estrutura de documentos HTML são predefinidas. O autor de documentos HTML somente pode usar as tags que estão definidas no padrão HTML (como <p>, <h1>, etc.).

A XML permite ao autor definir suas próprias tags e sua própria estrutura de documento.

As tags no exemplo acima (como <para> e <de>) não estão definidas em qualquer padrão XML.

A XML é um complemento para a HTML

A XML não é um substituto para a HTML.

É importante entender que a XML não é um substituto para a HTML. No desenvolvimento futuro da Web é provável que a XML seja usada para descrever os dados, enquanto que a HTML seja usada para formatar e exibir os mesmos dados.

Minha melhor descrição da XML é esta: **A XML é uma ferramenta multi-plataforma (cross-platform) independente de software e hardware para transmitir informação.**

A XML no desenvolvimento futuro da Web

A XML estará em todos os lugares.

Temos participado do desenvolvimento da XML desde a sua criação. É impressionante ver o quão rapidamente o padrão XML foi desenvolvido e quão rapidamente um grande número de distribuições de software adotou o padrão.

Acreditamos enfaticamente que a XML será tão importante para o futuro da Web como a HTML tem sido para o estabelecimento da Web e que a XML será a ferramenta mais comum para toda a manipulação de dados e transmissão de dados.

Como a XML Pode Ser Usada?

É importante entender que a XML foi projetada para armazenar, transportar, e trocar dados. A XML não foi projetada para exibir dados.

A XML Pode Separar os Dados da HTML

Com a XML, os seus dados são armazenados fora da sua HTML.

Quando a HTML é usada para exibir dados, os dados são armazenados dentro da sua HTML. Com a XML, os dados podem ser armazenados em arquivos XML separados. Deste modo você pode concentrar-se no uso da HTML para o layout e exibição dos dados, e ter certeza de que as mudanças nos dados básicos não irão requerer quaisquer mudanças na sua HTML.

Os dados em XML também podem ser armazenados dentro de páginas HTML como "Data Islands" [ilhas de dados]. Você pode ainda concentrar-se no uso da HTML somente para formatar e exibir os dados.

A XML é usada para Trocar Dados

Com a XML, os dados podem ser trocados entre sistemas incompatíveis.

No mundo real, os sistemas de computadores e bancos de dados contém dados em formatos incompatíveis. Um dos desafios que mais consomem tempo para os desenvolvedores tem sido a troca de dados entre tais sistemas na Internet.

Converter os dados para a XML pode reduzir enormemente esta complexidade e criar dados que podem ser lidos por tipos de aplicações muito diferentes.

XML e B2B

Com a XML, informações financeiras podem ser trocadas pela Internet.

Espere ver muito XML e B2B (Business To Business) num futuro próximo.

A XML será a principal linguagem para trocar informações financeiras entre negócios pela Internet. Muitas das aplicações de interesse no B2B estão em desenvolvimento.

A XML pode ser usada para Compartilhar Dados

Com a With XML, arquivos de texto puro podem ser usados para compartilhar dados.

Como os dados em XML são armazenados em formato de texto puro, a XML provê um meio independente de software e hardware para compartilhar dados.

Isto torna muito mais fácil criar dados com os quais diferentes aplicações podem trabalhar. Também torna mais fácil expandir ou atualizar um sistema para novos sistemas operacionais, servidores, aplicações, e novos navegadores.

A XML pode ser usada para Armazenar Dados

Com a XML, arquivos de texto puro podem ser usados para armazenar dados.

A XML pode também ser usada para armazenar dados em arquivos ou em bancos de dados. Aplicações podem ser escritas para armazenar e recuperar informação armazenadas, e aplicações genéricas podem ser usadas para exibir os dados.

A XML pode tornar os seus Dados mais Úteis

Com a XML, os seus dados estão disponíveis para mais usuários.

Já que a XML é independente de hardware, de software e de aplicações, você pode tornar os seus dados disponíveis para outros além de somente para os navegadores HTML padrões.

Outros clientes e aplicações podem acessar os seus arquivos XML como fontes de dados, como se estivessem acessando bancos de dados. Os seus dados podem estar disponíveis para todos os tipos de "máquinas de leitura" (agentes), e é mais fácil tornar os seus dados disponíveis para pessoas cegas, ou pessoas com outras deficiências.

A XML pode ser usada para Criar Linguagens novas

A XML é a mãe da WAP e da WML.

A Wireless Markup Language (WML) [Linguagem de Marcação (para aparelhos) Sem Fio], usada para marcar aplicações na Internet para aparelhos portáteis como telefones móveis [celulares], é escrita em XML.

Você pode ler mais sobre a WML no [tutorial de WML](#) (em inglês).

Se os Desenvolvedores tiverem Juízo

Se eles TIVEREM juízo, todas as aplicações futuras trocarão seus dados em XML.

O futuro pode nos dar processadores de texto, aplicações de planilhas de cálculo (spreadsheet) e bancos de dados que podem ler os dados uns dos outros num formato de texto puro, sem qualquer utilitário de conversão entre eles.

Só podemos rezar para que a Microsoft e todos os outros distribuidores de softwares concordem.

A Sintaxe da XML

As regras de sintaxe da XML são muito simples e muito restritas. As regras são muito fáceis de aprender, e muito fáceis de usar.

Por causa disto, criar software que possa ler e manipular XML é muito fácil de fazer.

Um exemplo de documento XML

Os documentos XML usam uma sintaxe simples e auto-descritiva.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bilhete>
<para>José</para>
<de>Maria</de>
<título>Lembrete</título>
<corpo>Não me esqueça neste fim-de-semana!</corpo>
</bilhete>
```

A primeira linha do documento - a declaração XML - define a versão XML e a codificação de caracteres usada no documento. Neste caso o documento está conforme a especificação 1.0 da XML e usa o conjunto de caracteres ISO-8859-1 (Latin-1/West European).

A próxima linha descreve o elemento raiz do documento (como se dissesse: "este documento é um bilhete"):

```
<bilhete>
```

As próximas 4 linhas descrevem 4 elementos filhos da raiz (para, de, título, e corpo)):

```
<para>José</para>
<de>Maria</de>
<título>Lembrete</título>
<corpo>Não me esqueça neste fim-de-semana!</corpo>
```

E finalmente a última linha define o final do elemento raiz:

```
</bilhete>
```

Você pode deduzir deste exemplo que o documento XML contém um Bilhete para José de Maria? Você não concorda que a XML é bastante auto-descritiva?

Todos os elementos XML devem ter uma tag de fechamento

Com a XML, é ilegal omitir a tag de fechamento.

Na HTML alguns elementos não têm que ter uma tag de fechamento. O seguinte código é legal na HTML:

```
<p>Este é um parágrafo
<p>Este é outro parágrafo
```


Na XML todos os elementos devem ter uma tag de fechamento, assim:

```
<p>Este é um parágrafo</p>
<p>Este é outro parágrafo</p>
```

Observação: Você pode ter observado nos exemplos anteriores que a declaração XML não tinha uma tag de fechamento. Isto não é um erro. A declaração não é parte do documento XML em si. Ela não é um elemento XML, e ela não deve ter uma tag de fechamento.

As tags XML são sensíveis à caixa (case sensitive)

Diferente da HTML, as tags XML são sensíveis à caixa [maiúsculas e minúsculas são diferentes].

Com a XML, a tag <Letra> é diferente da tag <letra>.

As tags de abertura e fechamento devem portanto estar escritas com a mesma caixa:

```
<Mensagem>Isto está incorreto</mensagem>
<mensagem>Isto está correto</mensagem>
```

Todos os elementos XML devem estar propriamente aninhados

Aninhamento impróprio de tags não faz sentido para a XML.

Na HTML alguns elementos podem estar imprópriamente aninhados uns dentro dos outros como estes:

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

Na XML todos os elementos devem estar propriamente aninhados uns dentro dos outros como estes:

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

Todos os documentos XML devem ter um elemento raiz

Todos os documentos XML devem conter um único par de tags para definir um elemento raiz.

Todos os outros elementos devem estar dentro deste elemento raiz.

Tods os elementos podem ter sub-elementos (elementos filho). Os sub-elementos devem estar corretamente aninhados dentro de seu elemento pai:

```
<raiz>
  <filho>
    <subfilho>.....</subfilho>
  </filho>
</raiz>
```

Valores de Atributos devem sempre estar Aspados

Com a XML, é ilegal omitir as marcas de aspas em torno de valores de atributos.

Os elementos XML podem ter atributos em pares name/value (nome/valor) assim como na HTML. Na XML o valor do atributo deve sempre estar aspadado. Estude os dois documentos XML abaixo. O primeiro está incorreto, o segundo está correto:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bilhete data=12/11/2002>
<para>José</para>
<de>Maria</de>
</bilhete>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bilhete data="12/11/2002">
<para>José</para>
<de>Maria</de>
</bilhete>
```

O erro no primeiro documento é que o atributo data no elemento bilhete não está aspadado.

Isto está correto: data="12/11/2002". Isto está incorreto: data=12/11/2002.

Com a XML, os espaços escritos são preservados

Com a XML, o espaço em branco no seu documento não é truncado.

Isto é diferente da HTML. Com a HTML, uma sentença como esta:

Ôi, meu nome é José,

será exibida assim:

Ôi, meu nome é José,

porque a HTML tira fora o espaço em branco.

Com a XML, CR / LF é convertida para LF

Com a XML, uma nova linha é sempre armazenada como LF.

Você sabe o que é uma máquina de escrever? Bem, uma máquina de escrever é um aparelho mecânico usado nos séculos anteriores para produzir documentos impressos. :-)

Após você ter datilografado uma linha de texto numa máquina de escrever, você tinha que manualmente retornar o carro de impressão para a posição na margem esquerda e manualmente alimentar o papel com uma linha.

Em aplicações para o Windows, uma nova linha é normalmente armazenada como um par de caracteres: retorno do carro (CR: carriage return) e alimentação de linha (LF: line feed). O par de caracteres guarda alguma semelhança com as ações da máquina de escrever de estabelecer uma nova linha. Nas aplicações para Unix, uma nova linha é normalmente armazenada como um caractere LF. Aplicações para Macintosh usam somente um caractere CR para armazenar uma nova linha.

Comentários em XML

A sintaxe para escrever comentários em XML é similar àquela da HTML.

```
<!-- Este é um comentário -->
```

Nada há de especial sobre a XML

Nada há de especial sobre a XML. Ela é apenas texto puro com a adição de algumas tags XML cercadas por parênteses angulares [angle brackets].

Programas que podem manejar texto puro podem também manejar XML. Num simples editor de texto, as tags XML serão visíveis e não irão ser manejadas especialmente.

Numa aplicação XML-cilente entretanto, as tags XML podem ser manejadas especialmente. As tags podem ou não estar visíveis, ou ter um significado funcional, dependendo da natureza da aplicação.

Elementos XML

Os Elementos XML são extensíveis e têm relacionamentos.

Os Elementos XML têm regras simples para nomes.

Os Elementos XML são Extensíveis

Os documentos XML documents podem ser estendidos para conter mais informação.

Veja o exemplo BILHETE XML seguinte:

```
<bilhete>
<para>José</para>
<de>Maria</de>
<corpo>Não me esqueça neste fim-se-semana!</corpo>
</bilhete>
```

Vamos imaginar que criamos uma aplicação que extrai os elementos <para>, <de>, e <corpo> do documento XML e produz esta saída:

BILHETE

Para: José
De: Maria

Não me esqueça neste fim-se-semana!

Imagine que o autor do documento XML adiciona alguma informação extra à ele:

```
<bilhete>
<data>2002-08-01</data>
<para>José</para>
<de>Maria</de>
<título>Lembrete</título>
<corpo>Não me esqueça neste fim-se-semana!</corpo>
</bilhete>
```

A aplicação deveria falhar ou entrar em colapso?

Não. A aplicação deveria ainda ser capaz de encontrar os elementos <para>, <de>, e <corpo> no documento XML e produzir a mesma saída.

Os documentos XML são Extensíveis.

Os Elementos XML têm Relacionamentos

Os Elementos são relacionados como pais e filhos.

Para entender a terminologia da XML, você deve saber como as relações entre os elementos XML são denominados, e como o conteúdo do elemento é descrito.

Imagine que isto é a descrição de um livro:

Minha Primeira XML

Introdução à XML

- O Que É a HTML
- O Que É a XML

A Sintaxe da XML

- Os elementos devem ter uma tag de fechamento
- Os elementos devem estar propriamente aninhados

Imagine que este documento XML descreve o livro:

```
<livro>
<título>Minha Primeira XML</título>
<prod id="33-657" media="papel"></prod>
<capítulo>Introdução à XML
<para>O Que É a HTML</para>
<para>O Que É a XML</para>
</capítulo>

<capítulo>A Sintaxe da XML
<para>Os elementos devem ter uma tag de fechamento</para>
<para>Os elementos devem estar propriamente aninhados</para>
</capítulo>

</livro>
```

Livro é o elemento **raiz**. Título, prod, e capítulo são **elementos filhos** de livro. Livro é o **elemento pai** de título, prod, and capítulo. Título, prod, e capítulo are **parentes (siblings)** (ou **elementos irmãos**) porque eles têm o mesmo pai.

Os Elementos têm Conteúdo

Os elementos podem ter diferentes tipos de conteúdo.

Um **elemento XML** é tudo desde (e incluindo) a tag de abertura do elemento até (e incluindo) a tag de fechamento do elemento.

Um elemento pode ter conteúdo de **elemento**, conteúdo **misto**, conteúdo **simples**, ou conteúdo **vazio**. Um elemento pode também ter **atributos**.

No exemplo acima, livro tem **conteúdo de elemento**, porque ele contém outros elementos. Capítulo tem **conteúdo misto** porque ele contém tanto texto quanto outros elementos. Para tem **conteúdo simples** (ou **conteúdo de texto**) porque ele contém somente texto. Prod tem **conteúdo vazio**, porque ele não carrega informação alguma.

No exemplo acima somente o elemento prod tem **atributos**. O **atributo** denominado id tem o **valor** "33-657". O **atributo** denominado media tem o **valor** "papel".

Nomeação de Elementos

Os elementos XML devem seguir estas regras de nomeação:

- Nomes podem conter letras, números, e outros caracteres
- Nomes não devem começar com um número ou caractere de pontuação
- Nomes não devem começar com as letras xml (ou XML ou Xml ..)
- Nomes não podem conter espaços

Tenha cuidado quando você "inventar" nomes de elementos e siga estas regras simples:

Qualquer nome pode ser usado, não há palavras reservadas, mas a idéia é tornar os nomes descritivos. Nomes com um sublinhado separador são bons.

Exemplos: <primeiro_nome>, <último_nome>.

Evite "-" e "." nos nomes. Por exemplo, se você nomeia algo como "primeiro-nome", pode haver uma bagunça se o seu software tentar subtrair nome de primeiro. Ou se você nomeia algo como "primeiro.nome", o seu software pode pensar que "nome" é uma propriedade do objeto "primeiro".

Os nomes de elementos podem ser tão longos quanto você queira, mas não exagere. Os nomes devem ser curtos e simples, como este: <título_livro> e não como este: <o_título_do_livro>.

Os documentos XML geralmente têm um banco de dados correspondente, no qual existem campos correspondentes aos elementos no documento XML. Uma boa prática é usar as regras de nomeação do seu banco de dados para os elementos nos documentos XML.

Letras não inglesas como éóá são perfeitamente legais em nomes de elementos XML, mas fique atento a problemas se a sua distribuição de software não os suportar.

Os ":" não devem ser usados em nomes de elementos porque são reservados para serem usados em algo chamado namespaces (veremos mais adiante).

Atributos XML

Os elementos XML podem ter atributos na tag de abertura, assim como na HTML.

Os atributos são usados para prover informação adicional sobre os elementos.

Atributos XML

Os elementos XML podem ter atributos.

Da HTML você lembrará disto: ``. O atributo SRC prove informação adicional sobre o elemento IMG.

Na HTML (e na XML) os atributos provêem informação adicional sobre os elementos:

```
  
<a href="demo.asp">
```

Os atributos geralmente provêem informação que não é parte dos dados. No exemplo abaixo, o tipo de arquivo é irrelevante ao dado, mas importante para o software que quer manipular o elemento:

```
<arquivo tipo="gif">computer.gif</arquivo>
```

Estilos de Aspas, "feminino" ou 'feminino'?

Os valores de atributo devem sempre ser colocados entre aspas, mas tanto aspas simples quanto duplas podem ser usadas. Para o sexo de uma pessoa, a tag pessoa pode ser escrita assim:

```
<pessoa sexo="feminino">
```

ou assim:

```
<pessoa sexo='feminino'>
```

Observação: Se o valor do atributo já contém aspas duplas é necessário usar aspas simples, como neste exemplo:

```
<bandido nome='George "Espingarda" Ziegler'>
```

Observação: Se o valor do atributo já contém aspas simples é necessário usar aspas duplas, como neste exemplo:

```
<bandido nome="George 'Espingarda' Ziegler">
```

Uso de Elementos vs. Atributos

Dados podem ser armazenados em elementos filhos ou em atributos.

Dê uma olhada nestes exemplos:

```
< Pessoa sexo="feminino">
  <primeironome>Anna</primeironome>
  <últimonome>Smith</últimonome>
</ Pessoa>
```

```
< Pessoa>
  < sexo>feminino</ sexo>
  <primeironome>Anna</primeironome>
  <últimonome>Smith</últimonome>
</ Pessoa>
```

No primeiro exemplo `sexo` é um atributo. No último, `sexo` é um elemento filho. Ambos os exemplos provêm a mesma informação. Não há regras sobre quando usar atributos e quando usar elementos filhos. Minha experiência é que os atributos são convenientes na HTML, mas na XML você deve tentar evita-los. Use elementos filhos se a informação parecer-se com dados.

Meu Jeito Favorito

Eu gosto de armazenar dados em elementos filhos.

Os três documentos seguintes contém exatamente as mesmas informações:

Um atributo `data` é usado no primeiro exemplo:

```
< bilhete data="12/11/2002">
  < para>José</ para>
  < de>Maria</ de>
  < título>Lembrete</ título>
  < corpo>Não me esqueça neste fim-de-semana!</ corpo>
</ bilhete>
```

Um elemento `data` é usado no segundo exemplo:

```
< bilhete>
  < data>12/11/2002</ data>
  < para>José</ para>
  < de>Maria</ de>
  < título>Lembrete</ título>
  < corpo>Não me esqueça neste fim-de-semana!</ corpo>
</ bilhete>
```

Um elemento `data` expandido é usado no terceiro: (ESTE É O MEU FAVORITO):

```
< bilhete>
  < data>
    < dia>12</ dia>
    < mês>11</ mês>
    < ano>2002</ ano>
  </ data>
  < para>José</ para>
  < de>Maria</ de>
  < título>Lembrete</ título>
  < corpo>Não me esqueça neste fim-de-semana!</ corpo>
</ bilhete>
```

Evitar o uso de atributos?

Você deve evitar o uso de atributos?

Aqui estão alguns dos problemas no uso de atributos:

- atributos não podem conter múltiplos valores (elementos filhos podem)
- atributos não são tão facilmente expansíveis (para futuras mudanças)
- atributos não podem descrever estruturas (elementos filhos podem)
- atributos são mais difíceis de manipular por código de programa
- valores de atributos values não são tão fáceis de testar em compáração com uma Document Type Definition (DTD) - a qual é usada para definir os elementos legais de um documento XML

Se você usa atributos como compartimentos para dados, você acaba com documentos que são difíceis de ler e manter. Tente usar **elementos** para descrever dados. Use atributos somente para prover informação que não seja relevante para os dados.

Não acabe assim (se você pensa que isto se parece com XML, você não entendeu o ponto):

```
<bilhete dia="12" mês="11" ano="2002"
para="José" de="Maria" título="Lembrete"
corpo="Não me esqueça neste fim-de-semana!">
</bilhete>
```

Uma Exceção para a minha Regra do Atributo

Regras sempre têm exceções.

A minha regra sobre os atributos tem uma exceção:

Às vezes eu destino referências de ID aos elementos. Estas referências de ID podem ser usadas para acessar elementos XML de maneira muito semelhante aos atributos NAME ou ID na HTML. Este exemplo demonstra isso:

```
<mensagens>
  <bilhete id="p501">
    <para>José</para>
    <de>Maria</de>
    <título>Lembrete</título>
    <corpo>Não me esqueça neste fim-de-semana!</corpo>
  </bilhete>

  <bilhete id="p502">
    <para>Maria</para>
    <de>José</de>
    <título>Re: Lembrete</título>
    <corpo>Não esquecerei!</corpo>
  </bilhete>
</mensagens>
```

O ID nestes exemplos é apenas um contador, ou um identificador único, para identificar os diferentes bilhetes no arquivo XML, e não são parte dos dados dos bilhetes.

O que eu estou tentando dizer aqui é que metadados (dados sobre dados) devem ser armazenados como atributos, e que os dados propriamente ditos devem ser armazenados como elementos.

Validação XML

XML com sintaxe correta é XML Bem Formada.

XML validada em comparação com uma DTD é XML Válida.

Documentos XML "Bem Formados"

Um documento XML "Bem Formado" tem a sintaxe XML correta.

Um documento XML "Bem Formado" é um documento que obedece as regras de sintaxe da XML que foram descritas nos capítulos anteriores:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bilhete>
<para>José</para>
<de>Maria</de>
<título>Lembrete</título>
<corpo>Não me esqueça neste fim-de-semana!</corpo>
</bilhete>
```

Documentos XML "Válidos"

Um documento XML "Válido" também obedece a uma DTD.

Um documento XML "Válido" é um documento XML "Bem Formado", e que também obedece as regras de uma Document Type Definition (DTD):

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE bilhete SYSTEM "BilhetesInternos.dtd">
<bilhete>
<para>José</para>
<de>Maria</de>
<título>Lembrete</título>
<corpo>Não me esqueça neste fim-de-semana!</corpo>
</bilhete>
```

A DTD da XML

Uma DTD define os elementos legais de um documento XML.

O propósito de uma DTD é definir os blocos de construção legais de um documento XML. Ela define a estrutura do documento com uma lista de elementos legais. Você pode ler mais sobre DTD e como validar seus documentos XML no [Tutorial de DTD](#) (em inglês).

O Schema (esquema) da XML

O Schema XML é uma alternativa baseada na XML para a DTD.

O W3C suporta uma alternativa para a DTD chamada XML Schema. Você pode ler mais sobre o Schema XML no [Tutorial de Schema](#) (em inglês).

Erros Irão Parar Você

Os errors em um documento XML irão parar o programa XML.

A Especificação da XML do W3C estabelece que um programa não deve continuar a processar um documento XML se ele encontra um erro de validação. A razão é que os softwares em XML devem ser fáceis de escrever, e que todos os documentos XML devem ser compatíveis.

Com a HTML era possível criar documentos com muitos erros (como quando você esquece uma tag de fechamento). Uma das razões principais pelas quais os navegadores HTML são tão grandes e incompatíveis, é que eles têm seus próprios meios de descobrir como um documento deve parecer quando eles encontram um erro de HTML.

Com a XML isto não será possível.

Um Validador XML Geral

Para ajuda-lo a validar seus arquivos XML, criamos este vínculo de modo que você possa [validar qualquer arquivo XML](#) (em inglês).

A XML no Netscape e no Explorer

Em nosso tutorial XML enfocamos o Internet Explorer 5.0 e superiores.

A XML neste sítio Web

Muitas aplicações suportam a XML. Estamos enfocando o Internet Explorer 5.0 e superiores.

Alguns reclamaram disso, mas pensamos que este é o meio mais prático de demonstrar exemplos reais de XML para você na Internet.

A XML no Netscape 6

O Netscape 6 suporta a XML.

Para ver a fonte XML no Netscape 6: clique com o botão direito na página e selecione "View Page Source (Ver Página Fonte)".

A XML no Internet Explorer 5.0

O Internet Explorer 5.0 suporta a XML 1.0 padrão.

O Internet Explorer 5.0 suporta a maioria dos padrões internacionais da XML 1.0 e da XML DOM (Document Object Model [Modelo do Objeto Documento]). Estes padrões são estabelecidos pelo World Wide Web Consortium (W3C).

O Internet Explorer 5.0 tem o seguinte suporte à XML:

- Ver documentos de XML
- Suporte pleno aos padrões da DTD do W3C
- XML embutida na HTML como Data Islands
- Vincular dados de XML com elementos de HTML
- Transformar e exibir XML com XSL
- Exibir XML com CSS
- Acesso ao DOM da XML

O Internet Explorer 5.0 também tem suporte para Behaviors (Comportamentos):

- Behaviors é uma tecnologia só da Microsoft
- Behaviors podem separar os scripts da página em HTML.
- Behaviors podem armazenar dados XML no disco do cliente.

Exemplos de todas estas características são dados nos próximos capítulos deste sítio web.

Você pode ler mais sobre o Internet Explorer, e baixar a última versão na [Informação sobre Navegadores da W3Schools](#) (em inglês).

Ver Arquivos em XML

Arquivos XML puros podem ser vistos no IE 5.0 (e superiores) e no Netscape 6, mas para exibi-los como uma página web, você tem que adicionar alguma informação de exibição.

Ver Arquivos XML

Para ver um documento XML no IE 5.0 (e superiores) você pode clicar num vínculo, digitar uma URL na barra de endereço, ou clicar duplamente no nome de um arquivo XML numa pasta de arquivos. Se você abrir um documento XML no IE, ele exibirá o documento com os elementos raiz e filhos em cores codificadas. Um sinal de mais (+) ou sinal dum e menos (-) à esquerda dos elementos pode ser clicado para expandir ou recolher a estrutura do elemento. Se você quer ver a fonte da XML pura, você deve selecionar "Exibir Fonte" no menu do navegador.

Para ver um documento XML no Netscape 6 você terá que abrir o arquivo XML e então clicar com o botão direito no arquivo XML e selecionar "Exibir Página Fonte". Se você abrir um documento XML no Netscape 6, ele exibirá o documento com os elementos raiz e filhos em cores codificadas.

Veja este arquivo XML: [bilhete.xml](#)

```
▼<bilhete>
  <para>José</para>
  <de>Maria</de>
  <título>Lembrete</título>
  <corpo>Não me esqueça neste fim-de-semana!</corpo>
</bilhete>
```

Observação: Não espere que os arquivos XML estejam formatados com documentos HTML!

Ver um arquivo XML inválido

Se um arquivo XML errado for aberto, o navegador irá relatar o erro.

Veja este arquivo XML: [bilhete_errado.xml](#)

This page contains the following errors:

error on line 5 at column 17: Opening and ending tag mismatch: de line 0 and Dde

Below is a rendering of the page up to the first error.

José

Outros Exemplos

Ver alguns documentos em XML pode ajuda-lo a pegar o jeito da XML.

Coletamos os seguintes arquivos XML para você:

[Um catálogo de CDs em XML](#)

Esta é minha coleção de CDs favoritos, armazenada como dados XML (títulos velhos e chatos eu penso).

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

<CATALOG>

<CD>

<TITLE>Empire Burlesque</TITLE>
<ARTIST>Bob Dylan</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Columbia</COMPANY>
<PRICE>10.90</PRICE>
<YEAR>1985</YEAR>

</CD>

<CD>

<TITLE>Hide your heart</TITLE>
<ARTIST>Bonnie Tyler</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>CBS Records</COMPANY>
<PRICE>9.90</PRICE>
<YEAR>1988</YEAR>

</CD>

<CD>

<TITLE>Greatest Hits</TITLE>
<ARTIST>Dolly Parton</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>RCA</COMPANY>
<PRICE>9.90</PRICE>
<YEAR>1982</YEAR>

</CD>

<CD>

<TITLE>Still got the blues</TITLE>
<ARTIST>Gary Moore</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>Virgin records</COMPANY>
<PRICE>10.20</PRICE>
<YEAR>1990</YEAR>

</CD>

<CD>

<TITLE>Eros</TITLE>
<ARTIST>Eros Ramazzotti</ARTIST>
<COUNTRY>EU</COUNTRY>
<COMPANY>BMG</COMPANY>
<PRICE>9.90</PRICE>
<YEAR>1997</YEAR>

</CD>

<CD>

<TITLE>One night only</TITLE>
<ARTIST>Bee Gees</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>Polydor</COMPANY>
<PRICE>10.90</PRICE>
<YEAR>1998</YEAR>

</CD>

<CD>

<TITLE>Sylvias Mother</TITLE>
<ARTIST>Dr.Hook</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>CBS</COMPANY>
<PRICE>8.10</PRICE>
<YEAR>1973</YEAR>

</CD>

<CD>

<TITLE>Maggie May</TITLE>
<ARTIST>Rod Stewart</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>Pickwick</COMPANY>
<PRICE>8.50</PRICE>
<YEAR>1990</YEAR>

</CD>

<CD>

<TITLE>Romanza</TITLE>

<ARTIST>Andrea Bocelli</ARTIST>
<COUNTRY>EU</COUNTRY>
<COMPANY>Polydor</COMPANY>
<PRICE>10.80</PRICE>
<YEAR>1996</YEAR>
</CD>
<CD>
<TITLE>When a man loves a woman</TITLE>
<ARTIST>Percy Sledge</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Atlantic</COMPANY>
<PRICE>8.70</PRICE>
<YEAR>1987</YEAR>
</CD>
<CD>
<TITLE>Black angel</TITLE>
<ARTIST>Savage Rose</ARTIST>
<COUNTRY>EU</COUNTRY>
<COMPANY>Mega</COMPANY>
<PRICE>10.90</PRICE>
<YEAR>1995</YEAR>
</CD>
<CD>
<TITLE>1999 Grammy Nominees</TITLE>
<ARTIST>Many</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Grammy</COMPANY>
<PRICE>10.20</PRICE>
<YEAR>1999</YEAR>
</CD>
<CD>
<TITLE>For the good times</TITLE>
<ARTIST>Kenny Rogers</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>Mucik Master</COMPANY>
<PRICE>8.70</PRICE>
<YEAR>1995</YEAR>
</CD>
<CD>
<TITLE>Big Willie style</TITLE>
<ARTIST>Will Smith</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Columbia</COMPANY>
<PRICE>9.90</PRICE>
<YEAR>1997</YEAR>
</CD>
<CD>
<TITLE>Tupelo Honey</TITLE>
<ARTIST>Van Morrison</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>Polydor</COMPANY>
<PRICE>8.20</PRICE>
<YEAR>1971</YEAR>
</CD>
<CD>
<TITLE>Soulsville</TITLE>
<ARTIST>Jorn Hoel</ARTIST>
<COUNTRY>Norway</COUNTRY>
<COMPANY>WEA</COMPANY>
<PRICE>7.90</PRICE>
<YEAR>1996</YEAR>
</CD>
<CD>
<TITLE>The very best of</TITLE>
<ARTIST>Cat Stevens</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>Island</COMPANY>
<PRICE>8.90</PRICE>
<YEAR>1990</YEAR>
</CD>

<CD>
 <TITLE>Stop</TITLE>
 <ARTIST>Sam Brown</ARTIST>
 <COUNTRY>UK</COUNTRY>
 <COMPANY>A and M</COMPANY>
 <PRICE>8.90</PRICE>
 <YEAR>1988</YEAR>
</CD>
<CD>
 <TITLE>Bridge of Spies</TITLE>
 <ARTIST>T'Pau</ARTIST>
 <COUNTRY>UK</COUNTRY>
 <COMPANY>Siren</COMPANY>
 <PRICE>7.90</PRICE>
 <YEAR>1987</YEAR>
</CD>
<CD>
 <TITLE>Private Dancer</TITLE>
 <ARTIST>Tina Turner</ARTIST>
 <COUNTRY>UK</COUNTRY>
 <COMPANY>Capitol</COMPANY>
 <PRICE>8.90</PRICE>
 <YEAR>1983</YEAR>
</CD>
<CD>
 <TITLE>Midt om natten</TITLE>
 <ARTIST>Kim Larsen</ARTIST>
 <COUNTRY>EU</COUNTRY>
 <COMPANY>Medley</COMPANY>
 <PRICE>7.80</PRICE>
 <YEAR>1983</YEAR>
</CD>
<CD>
 <TITLE>Pavarotti Gala Concert</TITLE>
 <ARTIST>Luciano Pavarotti</ARTIST>
 <COUNTRY>UK</COUNTRY>
 <COMPANY>DECCA</COMPANY>
 <PRICE>9.90</PRICE>
 <YEAR>1991</YEAR>
</CD>
<CD>
 <TITLE>The dock of the bay</TITLE>
 <ARTIST>Otis Redding</ARTIST>
 <COUNTRY>USA</COUNTRY>
 <COMPANY>Atlantic</COMPANY>
 <PRICE>7.90</PRICE>
 <YEAR>1987</YEAR>
</CD>
<CD>
 <TITLE>Picture book</TITLE>
 <ARTIST>Simply Red</ARTIST>
 <COUNTRY>EU</COUNTRY>
 <COMPANY>Elektra</COMPANY>
 <PRICE>7.20</PRICE>
 <YEAR>1985</YEAR>
</CD>
<CD>
 <TITLE>Red</TITLE>
 <ARTIST>The Communards</ARTIST>
 <COUNTRY>UK</COUNTRY>
 <COMPANY>London</COMPANY>
 <PRICE>7.80</PRICE>
 <YEAR>1987</YEAR>
</CD>
<CD>
 <TITLE>Unchain my heart</TITLE>
 <ARTIST>Joe Cocker</ARTIST>
 <COUNTRY>USA</COUNTRY>
 <COMPANY>EMI</COMPANY>
 <PRICE>8.20</PRICE>

```
<YEAR>1987</YEAR>
</CD>
</CATALOG>
```

Um catálogo de plantas em XML

Este é o catálogo de plantas de uma loja de plantas, armazenado como dados em XML.

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

<CATALOG>
  <CD>
    <TITLE>Empire Burlesque</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Columbia</COMPANY>
    <PRICE>10.90</PRICE>
    <YEAR>1985</YEAR>
  </CD>
  <CD>
    <TITLE>Hide your heart</TITLE>
    <ARTIST>Bonnie Tyler</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>CBS Records</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1988</YEAR>
  </CD>
  <CD>
    <TITLE>Greatest Hits</TITLE>
    <ARTIST>Dolly Parton</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>RCA</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1982</YEAR>
  </CD>
  <CD>
    <TITLE>Still got the blues</TITLE>
    <ARTIST>Gary Moore</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>Virgin records</COMPANY>
    <PRICE>10.20</PRICE>
    <YEAR>1990</YEAR>
  </CD>
  <CD>
    <TITLE>Eros</TITLE>
    <ARTIST>Eros Ramazzotti</ARTIST>
    <COUNTRY>EU</COUNTRY>
    <COMPANY>BMG</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1997</YEAR>
  </CD>
  <CD>
    <TITLE>One night only</TITLE>
    <ARTIST>Bee Gees</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>Polydor</COMPANY>
    <PRICE>10.90</PRICE>
    <YEAR>1998</YEAR>
  </CD>
  <CD>
    <TITLE>Sylvias Mother</TITLE>
    <ARTIST>Dr.Hook</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>CBS</COMPANY>
    <PRICE>8.10</PRICE>
    <YEAR>1973</YEAR>
  </CD>
  <CD>
    <TITLE>Maggie May</TITLE>
```

<ARTIST>Rod Stewart</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>Pickwick</COMPANY>
<PRICE>8.50</PRICE>
<YEAR>1990</YEAR>
</CD>
<CD>
<TITLE>Romanza</TITLE>
<ARTIST>Andrea Bocelli</ARTIST>
<COUNTRY>EU</COUNTRY>
<COMPANY>Polydor</COMPANY>
<PRICE>10.80</PRICE>
<YEAR>1996</YEAR>
</CD>
<CD>
<TITLE>When a man loves a woman</TITLE>
<ARTIST>Percy Sledge</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Atlantic</COMPANY>
<PRICE>8.70</PRICE>
<YEAR>1987</YEAR>
</CD>
<CD>
<TITLE>Black angel</TITLE>
<ARTIST>Savage Rose</ARTIST>
<COUNTRY>EU</COUNTRY>
<COMPANY>Mega</COMPANY>
<PRICE>10.90</PRICE>
<YEAR>1995</YEAR>
</CD>
<CD>
<TITLE>1999 Grammy Nominees</TITLE>
<ARTIST>Many</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Grammy</COMPANY>
<PRICE>10.20</PRICE>
<YEAR>1999</YEAR>
</CD>
<CD>
<TITLE>For the good times</TITLE>
<ARTIST>Kenny Rogers</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>Mucik Master</COMPANY>
<PRICE>8.70</PRICE>
<YEAR>1995</YEAR>
</CD>
<CD>
<TITLE>Big Willie style</TITLE>
<ARTIST>Will Smith</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Columbia</COMPANY>
<PRICE>9.90</PRICE>
<YEAR>1997</YEAR>
</CD>
<CD>
<TITLE>Tupelo Honey</TITLE>
<ARTIST>Van Morrison</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>Polydor</COMPANY>
<PRICE>8.20</PRICE>
<YEAR>1971</YEAR>
</CD>
<CD>
<TITLE>Soulsville</TITLE>
<ARTIST>Jorn Hoel</ARTIST>
<COUNTRY>Norway</COUNTRY>
<COMPANY>WEA</COMPANY>
<PRICE>7.90</PRICE>
<YEAR>1996</YEAR>
</CD>


```
<CD>
  <TITLE>The very best of</TITLE>
  <ARTIST>Cat Stevens</ARTIST>
  <COUNTRY>UK</COUNTRY>
  <COMPANY>Island</COMPANY>
  <PRICE>8.90</PRICE>
  <YEAR>1990</YEAR>
</CD>
<CD>
  <TITLE>Stop</TITLE>
  <ARTIST>Sam Brown</ARTIST>
  <COUNTRY>UK</COUNTRY>
  <COMPANY>A and M</COMPANY>
  <PRICE>8.90</PRICE>
  <YEAR>1988</YEAR>
</CD>
<CD>
  <TITLE>Bridge of Spies</TITLE>
  <ARTIST>T'Pau</ARTIST>
  <COUNTRY>UK</COUNTRY>
  <COMPANY>Siren</COMPANY>
  <PRICE>7.90</PRICE>
  <YEAR>1987</YEAR>
</CD>
<CD>
  <TITLE>Private Dancer</TITLE>
  <ARTIST>Tina Turner</ARTIST>
  <COUNTRY>UK</COUNTRY>
  <COMPANY>Capitol</COMPANY>
  <PRICE>8.90</PRICE>
  <YEAR>1983</YEAR>
</CD>
<CD>
  <TITLE>Midt om natten</TITLE>
  <ARTIST>Kim Larsen</ARTIST>
  <COUNTRY>EU</COUNTRY>
  <COMPANY>Medley</COMPANY>
  <PRICE>7.80</PRICE>
  <YEAR>1983</YEAR>
</CD>
<CD>
  <TITLE>Pavarotti Gala Concert</TITLE>
  <ARTIST>Luciano Pavarotti</ARTIST>
  <COUNTRY>UK</COUNTRY>
  <COMPANY>DECCA</COMPANY>
  <PRICE>9.90</PRICE>
  <YEAR>1991</YEAR>
</CD>
<CD>
  <TITLE>The dock of the bay</TITLE>
  <ARTIST>Otis Redding</ARTIST>
  <COUNTRY>USA</COUNTRY>
  <COMPANY>Atlantic</COMPANY>
  <PRICE>7.90</PRICE>
  <YEAR>1987</YEAR>
</CD>
<CD>
  <TITLE>Picture book</TITLE>
  <ARTIST>Simply Red</ARTIST>
  <COUNTRY>EU</COUNTRY>
  <COMPANY>Elektra</COMPANY>
  <PRICE>7.20</PRICE>
  <YEAR>1985</YEAR>
</CD>
<CD>
  <TITLE>Red</TITLE>
  <ARTIST>The Communards</ARTIST>
  <COUNTRY>UK</COUNTRY>
  <COMPANY>London</COMPANY>
  <PRICE>7.80</PRICE>
```

```
<YEAR>1987</YEAR>
</CD>
<CD>
  <TITLE>Unchain my heart</TITLE>
  <ARTIST>Joe Cocker</ARTIST>
  <COUNTRY>USA</COUNTRY>
  <COMPANY>EMI</COMPANY>
  <PRICE>8.20</PRICE>
  <YEAR>1987</YEAR>
</CD>
</CATALOG>
```

Um Cardápio Simples

Este é um cardápio para o café da manhã de um restaurante, armazenado como dados em XML.

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

<breakfast_menu>
  <food>
    <name>Belgian Waffles</name>
    <price>$5.95</price>
    <description>
      two of our famous Belgian Waffles with plenty of real maple syrup
    </description>
    <calories>650</calories>
  </food>
  <food>
    <name>Strawberry Belgian Waffles</name>
    <price>$7.95</price>
    <description>
      light Belgian waffles covered with strawberries and whipped cream
    </description>
    <calories>900</calories>
  </food>
  <food>
    <name>Berry-Berry Belgian Waffles</name>
    <price>$8.95</price>
    <description>
      light Belgian waffles covered with an assortment of fresh berries and whipped cream
    </description>
    <calories>900</calories>
  </food>
  <food>
    <name>French Toast</name>
    <price>$4.50</price>
    <description>
      thick slices made from our homemade sourdough bread
    </description>
    <calories>600</calories>
  </food>
  <food>
    <name>Homestyle Breakfast</name>
    <price>$6.95</price>
    <description>
      two eggs, bacon or sausage, toast, and our ever-popular hash browns
    </description>
    <calories>950</calories>
  </food>
</breakfast_menu>
```

Por que a XML é exibida assim?

Os documentos em XML não carregam informação sobre como exibir os dados.

Como as tags em XML são "inventadas" pelo autor do documento em XML, os navegadores não sabem se um a tag como <table> descreve uma tabela em HTML ou uma mesa de jantar [em inglês table pode significar tanto tabela quanto mesa, NT].

Sem qualquer informação sobre como exibir os dados, muitos navegadores exibirão o documento XML simplesmente como ele é.

Nos próximos capítulos, veremos diferentes soluções para o problema da exibição, usando CSS, XSL, JavaScript, e XML Data Islands.



Exibir a XML com CSS

Com CSS (Cascading Style Sheets [Folhas de Estilo em Cascata]) você pode adicionar informação de exibição à um documento XML.

Exibir seus futuros arquivos em XML com CSS?

Usaremos CSS para formatar seus futuros arquivos XML?

Não, não pensamos assim! Mas não pudemos resistir de tentar:

Dê uma olhada neste arquivo XML puro: [O Catálogo de CDs](#)

Finalmente, veja: [O Catálogo de CDs formatado com o arquivo CSS](#)

Empire Burlesque Bob Dylan

USA
Columbia
10.90
1985

Hide your heart Bonnie Tyler

UK
CBS Records
9.90
1988

Greatest Hits Dolly Parton

USA
RCA
9.90
1982

Still got the blues Gary Moore

UK
Virgin records
10.20
1990

Eros Eros Ramazzotti

EU
BMG
9.90
1997

One night only Bee Gees

UK
Polydor
10.90
1998

Sylvias Mother Dr.Hook

UK
CBS
8.10
1973

Maggie May Rod Stewart

UK
Pickwick
8.50
1990

Romanza Andrea Bocelli

Abaixo está uma fração do arquivo XML, com uma referência à folha de estilo CSS adicional. A segunda linha, **<?xml-stylesheet type="text/css" href="cd_cat.css"?>**, vincula o arquivo XML com o arquivo CSS:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/css" href="cd_catalog.css"?>
<CATALOG>
  <CD>
    <TITLE>Empire Burlesque</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Columbia</COMPANY>
    <PRICE>10.90</PRICE>
    <YEAR>1985</YEAR>
  </CD>
  <CD>
    <TITLE>Hide your heart</TITLE>
    <ARTIST>Bonnie Tyler</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>CBS Records</COMPANY>
    <PRICE>9.90</PRICE>
```

```
<YEAR>1988</YEAR>
</CD>
.
.
.
.
</CATALOG>
```

Nós NÃO acreditamos que formatar XML com CSS seja o futuro da Web. Mesmo que pareça certo usar a CSS deste modo, nós SIM acreditamos que formatar com XSL será o novo padrão (tão logo os principais navegadores a suportem).

Criar suas futuras Homepages com XML?

Você escreverá suas futuras Homepages em XML?

Não, não pensamos que irá! Mas não pudemos resistir de tentar: [Uma homepage escrita em XML](#).

Uma Simple Página em XML

Bemvindo à uma das poucas páginas escritas em XML!

Esta página não contém muito texto.

Ela é apenas um exemplo de que é possível criar uma homepage escrita em XML!

Nós NÃO acreditamos que a XML será usada para criar futuras Homepages.

Nós SIM acreditamos entretanto, que a XHTML - HTML definida como XML fará o truque: Por gentileza, leia o [tutorial de XHTML](#).

Exibir a XML com XSL

Com XSL você pode adicionar informação de exibição ao seu documento XML.

Exibir a XML com XSL

A XSL é a linguagem de folha de estilo preferida da XML.

A XSL (eXtensible Stylesheet Language [Linguagem de Folha de Estilo Extendível]) é muito mais sofisticada do que a CSS. Uma maneira de usar a XSL é transformar a XML em HTML antes de exibi-la no navegador conforme demonstrado nestes exemplo:

arquivo menu_cat.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="menu.xsl"?>

<breakfast_menu>
  <food>
    <name>Belgian Waffles</name>
    <price>$5.95</price>
    <description>Two of our famous Belgian Waffles with plenty of real maple syrup</description>
    <calories>650</calories>
  </food>
  <food>
    <name>Strawberry Belgian Waffles</name>
    <price>$7.95</price>
    <description>Light Belgian waffles covered with strawberries and whipped cream</description>
    <calories>900</calories>
  </food>
  <food>
    <name>Berry-Berry Belgian Waffles</name>
    <price>$8.95</price>
    <description>Light Belgian waffles covered with an assortment of fresh berries and whipped
cream</description>
    <calories>900</calories>
  </food>
  <food>
    <name>French Toast</name>
    <price>$4.50</price>
    <description>Thick slices made from our homemade sourdough bread</description>
    <calories>600</calories>
  </food>
  <food>
    <name>Homestyle Breakfast</name>
    <price>$6.95</price>
    <description>Two eggs, bacon or sausage, toast, and our ever-popular hash browns</description>
    <calories>950</calories>
  </food>
</breakfast_menu>
```

arquivo menu.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<body style="font-family:Arial;font-size:12pt;background-color:#EEEEEE">
<xsl:for-each select="breakfast_menu/food">
  <div style="background-color:teal;color:white;padding:4px">
    <span style="font-weight:bold"><xsl:value-of select="name"/> - </span>
    <xsl:value-of select="price"/>
  </div>
  <div style="margin-left:20px;margin-bottom:1em;font-size:10pt">
    <p>
      <xsl:value-of select="description"/>
      <span style="font-style:italic"> (<xsl:value-of select="calories"/> calories per serving)</span>
    </p>
  </div>
</xsl:for-each>
</body>
</html>
```

Abaixo está uma fração do arquivo XML, com uma referência à XSL adicionada. A segunda linha, **<?xml-stylesheet type="text/xsl" href="simple.xml"?>**, vincula o arquivo XML ao arquivo XSL:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="menu.xml"?>
<breakfast_menu>
  <food>
    <name>Belgian Waffles</name>
    <price>$5.95</price>
    <description>
      two of our famous Belgian Waffles
    </description>
    <calories>650</calories>
  </food>
</breakfast_menu>
```

Belgian Waffles - \$5.95

Two of our famous Belgian Waffles with plenty of real maple syrup *(650 calories per serving)*

Strawberry Belgian Waffles - \$7.95

Light Belgian waffles covered with strawberries and whipped cream *(900 calories per serving)*

Berry-Berry Belgian Waffles - \$8.95

Light Belgian waffles covered with an assortment of fresh berries and whipped cream *(900 calories per serving)*

French Toast - \$4.50

Thick slices made from our homemade sourdough bread *(600 calories per serving)*

Homestyle Breakfast - \$6.95

Two eggs, bacon or sausage, toast, and our ever-popular hash browns *(950 calories per serving)*

Se você quer aprender mais sobre XSL, por gentileza veja o [tutorial de XSL](#) (em inglês).

A XML em Data Islands (Ilhas de Dados)

Com o Internet Explorer 5.0 e superiores, a XML pode ser embutida em páginas HTML em Data Islands.

A XML Embutida na HTML

A tag não oficial <xml> é usada para embutir dados em XML na HTML.

Os dados XML podem ser embutidos diretamente em uma página HTML como esta:

```
<xml id="bilhete">
<blhete>
  <para>José</para>
  <de>Maria</de>
  <título>Lembrete</título>
  <corpo>Não me esqueça neste fim-de-semana!</corpo>
</bilhete>
</xml>
```

Ou um arquivo XML separado pode ser embutido assim:

```
<xml id="bilhete" src="bilhete.xml">
</xml>
```

Observe que a tag <xml> é um elemento da HTML, não um elemento da XML.

Ligação de Dados

As Data Islands podem ser ligadas com elementos da HTML (como tabelas).

No exemplo abaixo, um arquivo XML externo com uma ID "cdcat" é carregado de um arquivo XML externo. Uma tabela em HTML é ligada à Data Island com um atributo de data source (fonte de dados), e finalmente os elementos tabledata são ligados aos dados XML com um atributo data field (campo de dados) dentro de um span [o termo "span" (vão), neste caso, é usado para delimitar a abrangência de uma ação, NT].

```
<html>
<body>

<xml id="cdcat" src="cd_cat.xml"></xml>

<table border="1" datasrc="#cdcat">
<tr>
<td><span datafld="ARTIST"></span></td>
<td><span datafld="TITLE"></span></td>
</tr>
</table>

</body>
</html>
```

Se você está rodando o IE 5.0 ou superiores, você pode [tentar por si próprio](#).

Com o IE 5.0 e superiores você pode também ver o [arquivo XML externo](#). Tente também [este exemplo](#), que demonstra as tags <thead>, <tbody>, and <tfoot>.

O Parser (Partidor) XML da Microsoft

Para ler e atualizar - criar e manipular - um documento XML, você precisa de um parser XML.

Usando o parser XML

O parser XML da Microsoft vem com o Microsoft Internet Explorer 5.0.

Uma vez que você tenha instalado o IE 5.0, o parser estará disponível para scripts, tanto dentro de documentos HTML quanto fora de arquivos ASP. O parser representa um modelo de linguagem neutra de programação que suporta:

- JavaScript, VBScript, Perl, VB, Java, C++ e mais
- W3C XML 1.0 e XML DOM
- DTD e validação

Se você está usando JavaScript no IE 5.0, você pode criar um objeto de documento XML com o seguinte código:

```
var xmlDoc=new ActiveXObject("Microsoft.XMLDOM")
```

Se você está usando VBScript, você pode criar um objeto de documento XML com o seguinte código:

```
set xmlDoc=CreateObject("Microsoft.XMLDOM")
```

Se você está usando VBScript em ASP, você pode usar o seguinte código:

```
set xmlDoc=Server.CreateObject("Microsoft.XMLDOM")
```

Carregando um arquivo XML no parser

Os arquivos XML podem ser carregados no parser usando código de script.

O seguinte código carrega um documento XML (bilhete.xml) no parser XML:

```
<script type="text/javascript">
var xmlDoc = new ActiveXObject("Microsoft.XMLDOM")
xmlDoc.async="false"
xmlDoc.load("bilhete.xml")
// ..... o processamento do documento vai aqui
</script>
```

A segunda linha no código acima cria uma instância do parser XML da Microsoft.

A terceira linha desliga a carga não sincronizada, para ter certeza que o parser não continuará a execução antes que o documento esteja completamente carregado.

A quarta linha diz ao parser para carregar o documento XML chamado bilhete.xml.

Carregando texto XML puro no parser

Texto em XML pode ser carregado de uma sequência (string) de texto.

O seguinte código carrega uma sequência de texto no parser XML:

```
<script type="text/javascript">

var text="<bilhete>"
text=text+"<para>José</para><de>Maria</de>"
text=text+"<título>Lembrete</título>"
text=text+"<corpo>Não me esqueça neste fim-de-semana!</corpo>"
text=text+"</bilhete>"

var xmlDoc = new ActiveXObject("Microsoft.XMLDOM")
xmlDoc.async="false"
xmlDoc.loadXML(text)
// ..... o processamento do documento vai aqui
</script>
```

Observe que o método "loadXML" (em vez do método "load") é usado para carregar uma sequência de texto.

Exibir XML com JavaScript

Para exibir XML você pode usare JavaScript.

O JavaScript (ou VBScript) pode ser usado para importar os dados de um arquivo XML e exibir os dados XML dentro de uma página HTML.

Para ver como a XML e a HTML complementam-se desta maneira, primeiro veja o documento **XML** ([bilhete.xml](#)), depois abra o documento **HTML** ([bilhete.htm](#)) que contém um JavaScript que lê o arquivo XML e exibe a informação dentro de spans pré-definidos na página HTML.

Para ver como funciona, [tente você mesmo](#).

Você pode ver muito mais deste tipo de JavaScript na [DOM School](#) (em inglês).

XML na Vida Real

Alguns exemplos da vida real de como a XML pode ser usada para transportar informação.

Exemplo: XML News (Notícias)

XMLNews é uma especificação para troca de notícias e outras informações.

Usando um tal padrão torna-se fácil tanto para os produtores de notícias e novos consumidores produzir, receber, e arquivar qualquer tipo de informações sobre notícias através de hardware, software, e linguagens de programação diferentes.

Um exemplo de documento XML News:

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

<nitf>

<head>
<title>Colombia Earthquake</title>
</head>

<body>

<body.head>
<headline>
<h1>143 Dead in Colombia Earthquake</h1>
</headline>
<byline>
<bytag>By Jared Kotler, Associated Press Writer</bytag>
</byline>
<dateline>
<location>Bogota, Colombia</location>
<story.date>Monday January 25 1999 7:28 ET</story.date>
</dateline>
</body.head>

</body>

</nitf>
```

Fonte:

<http://www.clem.ufba.br/tuts/xml/default.htm>