

APRENDENDO O HTML, FORMATAÇÃO PARA WEB.

▶ Por André Marinho C.

▶ **O QUE É HTML?**

HTML é a linguagem com que se escrevem as páginas web. As páginas web podem ser vistas pelo usuário mediante um tipo de aplicação chamada navegador (browser). Podemos dizer portanto, que o HTML é a linguagem usada pelos navegadores para mostrar as páginas webs ao usuário, sendo hoje em dia a interface mais extensa na rede.

Esta linguagem nos permite aglutinar textos, imagens e áudios, e combiná-los a nosso gosto. Ademais, e é aqui onde está a sua vantagem em relação aos livros e revistas, o HTML nos permite a introdução de referências a outras páginas por meio dos links hipertextos.

O HTML se criou a princípio com objetivos de divulgação. Porém, não se pensou que a web chegaria a ser uma área de ócio com caráter multimídia, de modo que, o HTML se criou sem dar respostas a todos os possíveis usos que lhe dariam posteriormente e ao todo coletivo de gente que o utilizariam no futuro. Entretanto, frente a este deficiente planejamento, com o tempo, foram se incorporando modificações, as quais são os padrões do HTML. Numerosos padrões já se apresentaram. O HTML 4.01 foi o último padrão feito até o mês de abril de 2004.

Esta evolução tão anárquica do HTML, trouxe toda uma série de inconvenientes e deficiências que tiveram que ser superadas com a introdução de outras tecnologias acessórias capazes de organizar, otimizar e automatizar o funcionamento das webs. Exemplos que podem soar como conhecidos são as CSS, JavaScript ou outros. Veremos mais adiante em que consistem algumas delas.

Alguns dos problemas que acompanham ao HTML é a diversidade de navegadores presentes no mercado, os quais não são capazes de interpretar o mesmo código de uma maneira unificada. Isto obriga ao webmaster a, uma vez criada sua página, comprovar que esta pode ser lida satisfatoriamente por todos os navegadores, ou pelo menos, pelos mais utilizados.

Além do navegador necessário para ver os resultados de nosso trabalho, necessitamos evidentemente outra ferramenta capaz de criar a página em si. Um arquivo HTML (uma página) não é mais do que um texto. É por isso que para programar em HTML necessitamos um editor de textos.

É recomendável usar o bloco de notas que vem no Windows, ou outro editor de textos simples. No entanto, há de ter cuidado com alguns editores mais complexos como Wordpad ou MicrosoftWord, pois colocam seu próprio código especial ao salvar as páginas e como o HTML é unicamente texto plano, poderemos ter problemas com isso.

Existem outros tipos de editores específicos para a criação de páginas web, os quais oferecem muitas facilidades que nos permitem aumentar nossa produtividade. Porém, é aconselhável à princípio utilizar uma ferramenta, o mais simples possível para poder prestar a máxima atenção ao nosso código e familiarizarmos o antes possível com ele. Logo mais adiante, ganharemos tempo ao utilizarmos os editores mais versáteis.

Par ter mais claro todo o tema de editores e os tipos que existem, visite os artigos:

- Editores de HTML
- Bloco de notas

• Também se pode acessar a descrições de editores mais complexos que o Bloco de notas, porém mais potentes como o Hometown.

É importante ter claro tudo isso visto que em função de seus objetivos, pode ser que mais do que aprender HTML, resulte mais interessante aprender o uso de uma aplicação para a criação de páginas. Sendo assim, uma página é um arquivo onde está contido o código HTML em forma de texto. Estes arquivos tem extensão .html ou .htm (é indiferente qual deles utilizar). De modo que quando programarmos em HTML, o faremos com um editor de textos e salvaremos nossos trabalhos com extensão .html, por exemplo, minhapagina.html.

Conselho: Utilize sempre a mesma extensão em seus arquivos HTML. Isso evitará que você se confunda ao escrever os nomes dos seus arquivos umas vezes com .html e outras com .htm. E se você trabalha com uma equipe em um projeto, é ainda mais importante que todos estejam de acordo com a mesma extensão.

▶ **SINTAXES DO HTML**

O HTML é uma linguagem que baseia sua sintaxe em um elemento de base que chamamos etiqueta (tag). A etiqueta apresenta frequentemente duas partes:

Uma abertura de forma geral <etiqueta>

Um fechamento do tipo </etiqueta>

Tudo que estiver incluído no interior desta etiqueta sofrerá as modificações que caracterizam a esta etiqueta. Como por exemplo:

As etiquetas e definem um texto em negrito. Se em nosso documento HTML escrevemos uma frase com o seguinte código:

```
<b>Isto está em negrito</b>
```

O resultado será:

Isto está em negrito

As etiquetas <p> e </p> definem um parágrafo. Se em nosso documento HTML escrevêssemos: <p>Olá, estamos no parágrafo 1</p>

```
<p>Agora mudamos de parágrafo</p>
```

O resultado seria:

Olá, estamos no parágrafo 1

Agora mudamos de parágrafo

▶ PARTES DE UM DOCUMENTO HTML

Ademais de tudo isso, um documento HTML deve estar delimitado pela etiqueta <html> e </html>.

Dentro

deste documento, podemos ainda distinguir duas partes principais:

O cabeçalho, delimitado por <head> e </head> onde colocaremos etiquetas de índole informativo como por exemplo o título de nossa página.

O corpo, delimitado pelas etiquetas <body> e </body>, que será onde colocaremos nosso texto e imagens delimitados por sua vez por outras etiquetas como as que vimos anteriormente.

O resultado é um documento com a seguinte estrutura:

```
<html>
```

```
<head>
```

Etiquetas e conteúdos do cabeçalho

Dados que não aparecem em nossa página mas que são importantes para catalogá-la: Título,

palavras-chaves,

etc

```
</head>
```

```
<body>
```

Etiquetas e conteúdo do corpo

Parte do documento que será mostrada pelo navegador: Textos e imagens

```
</body>
```

```
</html>
```

As maiúsculas ou minúsculas são indiferentes ao escrever as etiquetas

As etiquetas podem ser escritas com qualquer tipo de combinação entre maiúsculas e minúsculas. Ou seja, <HTML> ou <HtMl> são a mesma etiqueta. Entretanto, é aconselhável acostumar-se a escrevê-las em minúscula já que outras tecnologias que podem conviver com nosso HTML (XML por exemplo) não são tão permissivas e nunca está mal ter bons costumes desde o princípio para evitar falhos triviais no futuro.

▶ SUA PRIMEIRA PÁGINA

Podemos já com estes conhecimentos, e mais algum outro, criar nossa primeira página. Para isso, abra seu editor de textos e, copie e cole o seguinte texto em um novo documento:

```
<html>
```

```
<head>
```

```
<title>Cozinha Para Todos</title>
```

```
</head>
```

```
<body>
```

```
<p><b>Bem-vindo,</b></p>
```

```
<p>Você está na página<b>Comida para todos</b>.</p>
```

```
<p>Aqui você aprenderá receitas fáceis e deliciosas.</p>
```

```
</body>
```

```
</html>
```

Agora salve este arquivo com extensão .html ou .htm em seu disco rígido. Para isso, acessamos no menu "Arquivo" e selecionamos a opção "Salvar como". Na janela escolhemos o diretório onde desejamos salvá-lo e colocaremos um nome para ele, como por exemplo: minha_página.html.

FORMATO DE PARÁGRAFOS EM HTML

Nos capítulos anteriores apresentamos a título de exemplo, algumas etiquetas que permitem dar formato ao nosso texto. Neste capítulo veremos com mais detalhe, as etiquetas mais amplamente utilizadas e exemplificaremos algumas delas posteriormente.

Formatar um texto passa por tarefas tão evidentes como definir os parágrafos, justificá-los, introduzir marcadores, numerações, ou pôr em negrito, itálico, etc.

Vimos que para definir os parágrafos utilizamos a etiqueta <p> que introduz um salto e deixa uma linha em branco antes de continuar com o resto do documento.

Podemos também utilizar a etiqueta
, da qual não existe seu fechamento correspondente (</br>), para realizar um simples enter, com o que não deixamos uma linha em branco, e sim, somente mudamos de linha.

Exemplo prático:

Para praticar um pouco o que acabamos de ver, vamos propor um exercício que se pode resolver no seu computador. Simplesmente queremos construir uma página que tenha, por esta ordem:

2 parágrafos centralizados

3 parágrafos à direita

Um salto triplo de linha

1 parágrafo alinhado à esquerda

Não vamos escrever nesta ocasião o código fonte do exercício. Podemos vê-lo em funcionamento em nosso navegador e na janela podemos obter o código fonte selecionando no menu Exibir a opção Código fonte.

CABEÇALHOS

Existem outras etiquetas para definir parágrafos especiais, formatados como títulos. São os cabeçalhos ou Header em inglês. Como dissemos, são etiquetas que formatam o texto como um título, para o qual atribuem um tamanho maior de letra e colocam o texto em negrita.

Existem vários tipos de cabeçalhos que se diferenciam no tamanho da letra que utilizam. A etiqueta em concreto é a <h1>, para os cabeçalhos maiores, <h2> para os de segundo nível e assim, até <h6> que é o cabeçalho menor.

Os cabeçalhos implicam também uma separação em parágrafos, portanto, tudo o que for escrito dentro de <h1> e </h1> (ou qualquer outro cabeçalho) será colocado em um parágrafo independente.

Podemos ver como se apresentam alguns cabeçalhos a seguir:

```
<h1>Cabeçalho de nível 1</h1>
```

Será visto dessa maneira na página:

Cabeçalho de nível 1

Os cabeçalhos, como outras etiquetas de html, suportam o atributo align. Vejamos um exemplo de cabeçalho de nível 2 alinhado ao centro:

```
<h2 align="center">Cabeçalho de nível 2</h2>
```

Será visto dessa maneira na página:

Cabeçalho de nível 2

Outro exercício interessante é construir uma página web que contenha todos os cabeçalhos possíveis.

Pode-se ver a seguir:

```
<html>
<head>
<title>Todos os cabeçalhos</title>
</head>
<body>
<h1>Cabeçalho de nível 1</h1>
<h2>Cabeçalho de nível 2</h2>
<h3>Cabeçalho de nível 3</h3>
```

```
<h4>Cabeçalho de nível 4</h4>
<h5>Cabeçalho de nível 5</h5>
<h6>Cabeçalho de nível 6</h6>
</body>
</html>
```

▶ FORMATANDO O TEXTO

Além de tudo relativo à organização dos parágrafos, um dos aspectos primordiais da formatação de um texto, é o da própria letra. É muito comum e prático, apresentar o texto ressaltado em negrito, itálico e outros. Paralelamente, o uso de índices e subíndices é vital para a publicação de textos científicos. Tudo isso e muito mais, é possível por meio do HTML a partir de uma série de etiquetas entre as quais vamos destacar algumas.

Negrito

Podemos escrever o texto em negrito o incluindo dentro das etiquetas `<b>` e `</b>` (bold). Esta mesma tarefa é desempenhada por `<strong>` e `</strong>` sendo ambas equivalentes. Nós nos inclinamos pelas primeiras por uma simples razão de esforço.

Escrevendo um código deste tipo:

```
<b>Texto em negrito</b>
```

Obteremos este resultado: **Texto em negrito**

Itálico

Também neste caso existem duas possibilidades, uma curta: `<i>` e `</i>` (italic) e outra um pouco mais longa: `<em>` e `</em>`. Neste manual, e na maioria das páginas que se vê por aí, é normal encontrar com a primeira forma que é sem dúvida mais simples para escrever e para se lembrar.

Abaixo, um exemplo de texto em itálico:

```
<i>Texto em itálico</i>
```

Que dá o seguinte efeito:

Texto em itálico

Sublinhado

O HTML nos propõe também para o sublinhado as etiquetas: `<u>` e `</u>` (underline). Entretanto o uso do sublinhado deve ser usado com muita precaução visto que os links hipertextos vão, a não ser que se indique o contrário, sublinhados com o que podemos confundir o leitor e afastá-lo do verdadeiro interesse de nosso texto.

Subscrito e sobrescrito

Este tipo de formato é de extrema utilidade para textos científicos. As etiquetas empregadas são:

```
<sup> e </sup> para os sobrescritos
```

```
<sub> e </sub> para os subscritos
```

Temos aqui um exemplo:

```
A <sup>13</sup>CC<sub>3</sub>H<sub>4</sub>CINOS é um heterociclo alergeno enriquecido
```

O resultado seria:

A ¹³CC₃H₄CINOS é um heterociclo alergeno enriquecido.

Combinar etiquetas

Todas estas etiquetas, incluindo as que já vimos e as que ainda vamos ver, podem ser combinadas estando umas dentro das outras de forma que conseguimos resultados diferentes. Assim, podemos sem nenhum problema criar um texto em negrito e em itálico colocando uma etiqueta dentro da outra:

```
<b>Isto só está em negrito<i>e isto em negrito e itálico</i></b>
```

Isto daria:

Isto só está em negrito e isto em negrito e itálico

 COR, TAMANHO E TIPO DE LETRA

Apesar de que por razões de homogeneidade e simplicidade de código estes tipos de formatos são controlados atualmente por folhas de estilo cascata (das quais já teremos tempo de falar), existe uma forma clássica e direta de definir cor, tamanho e tipo de letra de um determinado texto. Isto se faz a partir da etiqueta e seu fechamento correspondente. Dentro desta etiqueta devemos especificar os atributos correspondentes a cada um destes parâmetros que desejamos definir. A seguir comentamos os atributos principais desta etiqueta:

Atributo face

Define o tipo de letra. Este atributo é interpretado por versões do Netscape a partir da 3 e de MSIE 3 ou superiores. Outros navegadores as ignoram completamente e mostram o texto com a fonte que utilizam. Deve-se ter cuidado com este atributo já que cada usuário, dependendo da plataforma que utilize, pode não dispor dos mesmos tipos de letra que nós com o que, se nós escolhemos um tipo do que não dispõe, o navegador se verá forçado a mostrar o texto com a fonte que utiliza por padrão (que costuma ser Times New Roman). Para evitar isso, dentro do atributo costuma-se selecionar vários tipos de letras separados por vírgulas. Neste caso, o navegador comprovará que dispõe do primeiro tipo numerado e se não for assim, passará para o segundo e assim sucessivamente até encontrar um tipo que possua ou então, até acabar a lista e colocar a fonte padrão. Vejamos um exemplo:

```
<font face="Comic SansMS,arial,verdana">Este texto tem outra tipografia</font>
```

Que se visualizaria assim em uma página web:

Este texto tem outra tipografia

Atributo size

Define o tamanho da letra. Este tamanho pode ser absoluto ou relativo.

Se falarmos em termos absolutos, existem 7 níveis de tamanhos distintos numerados de 1 a 7 por ordem crescente. Escolheremos portanto um valor size="1" para a menor letra ou size="7" para a maior.

```
<font size=4>Este texto é maior</font>
```

Que se visualizaria assim em uma página web:

Este texto é maior

Podemos também modificar o tamanho de nossa letra em relação ao do texto mostrado anteriormente definindo o número de níveis que queremos subir ou descer nesta escala de tamanhos por meio do signo + ou - . Desse modo, se definimos nosso atributo como size="+1" o que queremos dizer é que aumentamos um nível o tamanho da letra. Se estávamos escrevendo previamente em 3, passaremos automaticamente a 4.

Os tamanhos reais que vemos na tela dependerão da definição e do tamanho da fonte escolhido pelo usuário no navegador. Este tamanho de fonte pode ser definido no Explorer indo ao menu superior, Exibir/Tamanho da fonte. No Netscape escolheremos View/Text Size. Esta flexibilidade pode acabar sendo embaraçosa para nós em mais de uma ocasião, já que em muitos casos desejaremos que o tamanho do texto permaneça constante para que este caiba em um determinado espaço. Veremos em seu momento que esta pré-fixação do tamanho pode ser realizada pelas folhas de estilo em cascata.

Atributo cor

A cor do texto pode ser definida através do atributo cor. Cada cor é por sua vez definida por um número hexadecimal que está composto por três partes. Cada uma destas partes representa a contribuição do vermelho, verde e azul à cor em questão.

Por outro lado é possível definir de uma maneira imediata algumas das cores mais freqüentemente usadas para as que se criaram um nome mais memotécnico:

Nombre Color

Aqua
Black
Blue
Fuchsia
Gray
Green

Lime
Maroon
Navy
Olive
Purple
Red
Silver
Teal
White
Yellow

`<font color="red">Este texto está em vermelho</font>`

Que se visualizaria assim em uma página web:

Este texto está em vermelho

Com tudo isso, já somos capazes de criar um texto formatado de uma forma realmente elaborada. Colocamos então em prática tudo o que aprendemos nestes capítulos fazendo um exercício consistente em uma página que tenha as seguintes características:

- Um título com cabeçalho de nível 1, em itálico e na cor verde oliva.
- Um segundo título com cabeçalho de nível 2, também na cor verde oliva.
- Todo texto da página deverá apresentar-se com uma fonte diferente da fonte padrão. Por exemplo, "Comic Sans MS" e no caso de que esta não esteja no sistema, que se coloque a fonte "Arial".

Atributos para páginas

As páginas HTML podem ser construídas com uma variedade de atributos que lhe podem dar um aspecto à página muito personalizado. Podemos definir atributos como a cor de fundo, a cor do texto ou dos links. Estes atributos se definem na etiqueta `<body>` e, como dizíamos são gerais a toda a página. O melhor para explicar seu funcionamento é vê-los um a um.

Atributos para fundos

Bgcolor:

especificamos uma cor de fundo para a página. No capítulo anterior aprendemos a construir qualquer cor, com seu nome ou seu valor RGB. A cor de fundo que podemos atribuir com bgcolor é uma cor plana, ou seja, a mesma para toda a superfície do navegador.

Background:

serve para indicar a colocação de uma imagem como fundo da página. Nos capítulos mais adiante, veremos como se inserem imagens com HTML e os tipos de imagens que se podem utilizar.

EXEMPLO DE FUNDO

Vamos colocar esta imagem no fundo da página.

A imagem chama-se fundo.jpg e supomos que se encontra no mesmo diretório que a página. Neste caso se colocaria a seguinte etiqueta `<body>`

`<body background="fundo.jpg">`

Pode-se ver o efeito de colocar este fundo em uma página a parte.

Cor do texto

Text: Este atributo serve para atribuir a cor do texto da página. Por padrão é o negro.

Ademais da cor do texto, temos três atributos para atribuir a cor dos links da página. Já devemos saber que os links devem diferenciar-se do resto do texto da página para que os usuários possam identificá-los facilmente. Para isso, eles costumam aparecer sublinhados e com uma cor mais viva que o texto. Os três atributos são os seguintes:

Link: a cor dos links que não foram visitados.

Vlink: a cor dos links visitados. A letra "v" vem justamente da palavra visitado. É a cor que terão os links que já visitamos. Por padrão sua cor é roxa. Esta cor deverá ser um pouco menos viva que a cor dos links normais.

Alink: é a cor dos links ativos. Um link está ativo no preciso momento em que se clica. Às vezes é difícil

perceber quando um link está ativo porque no momento em que se ativa, é porque o estamos clicando e nesse caso, o navegador abandonará a página rapidamente e não poderemos ver o link ativo mais que um mínimo instante.

Exemplo de cor de texto

Vamos ver uma página em que a cor de fundo seja preta, e as cores dos textos e dos links sejam claros. Colocaremos a cor do texto branca e os links amarelos, mais ressaltados os que não tenham sido visitados e menos ressaltados os que já tenham sido. Para isso, escreveríamos a etiqueta body assim:

```
<body bgcolor="#000000" text="#ffffff" link="#ffff33" alink="#ffffcc" alink="#ffff00">
```

O efeito pode-se ver em uma página a parte.

MARGENS

Com outros atributos da etiqueta <body> podem-se atribuir espaços de margens nas páginas, o que é muito útil para eliminar as margens em branco que aparecem nos lados, em cima e embaixo da página. Estes atributos são diferentes para o Internet Explorer e para o Netscape Navigator, por isso, devemos utilizá-los todos se queremos que todos os navegadores os interpretem perfeitamente.

Leftmargin: para a indicar a margem nos lados da página. Válido para Internet Explorer.

Topmargin: para indicar a margem acima e abaixo da página. Para Internet Explorer.

Marginwidth: a contrapartida de leftmargin para Netscape. (margem nos lados)

Marginheight: igual ao topmargin, mas para Netscape. (margem acima e abaixo)

Um exemplo de página sem margem é a própria página de andrecavalcante.com que você está visitando atualmente. (pelo menos na hora de escrever este artigo). Além disso, vamos ver outra página sem margens, caso alguém necessite ver o exemplo nestas linhas:

```
<body topmargin=0 leftmargin=0 marginheight=0 marginwidth=0 bgcolor="ffffff">
<table width=100% bgcolor=ff6666><tr><td>
<h1>Olá amigos</h1>
<br>
<br>
Obrigado por me visitar!
</td></tr></table>
</body>
```

Esta página tem o fundo branco e dentro um painel com o fundo vermelho. Na página poderemos ver que o painel ocupa o espaço na página sem deixar lugar para nenhum tipo de margem. Pode-se ver o exemplo em uma página à parte.

LISTAS

São realmente notáveis as possibilidades que o HTML nos oferece em questão de tratamento de texto. Não se limitam ao que vimos até agora, pois vão ainda mais longe. Vários exemplos disso são as listas, que servem para numerar e definir elementos, os textos pré-formatados e os cabeçalhos ou os títulos. As listas são utilizadas para citar, numerar e definir objetos. Também são utilizadas correntemente para deslocar o começo da linha para a direita.

Podemos distinguir três tipos de listas:

- Listas desordenadas
- Listas ordenadas
- Listas de definição

Veremos agora detalhadamente uma por uma:

Listas desordenadas

São delimitadas pelas etiquetas e (unordered list). Cada um dos elementos da lista é citado por meio de uma etiqueta (sem fechamento, ainda que exista nenhum inconveniente em colocá-lo).

Então, fica assim:

```
<p>Países do mundo</p>
<ul>
<li>Brasil</li>
<li>Espanha</li>
<li>Austrália</li>
```


O resultado:

Países do mundo

- Brasil
- Espanha
- Austrália

Podemos definir o tipo de marcador empregado para cada elemento. Para isso devemos especificá-lo por meio do atributo type incluído dentro da etiqueta de abertura , se queremos que o estilo seja válido para toda a lista, ou dentro da etiqueta se queremos especificar um só elemento. A sintaxe é do seguinte tipo:

<ul type="tipo de marcador">

E o tipo de marcador pode ser um dos seguintes:

- Circle
- Disc
- Square

Nota: Em alguns navegadores não funciona a opção de mudar o tipo de marcador e por mais que nos empenhemos, sempre sairá a bolinha preta.

Em caso de que não funcione, sempre podemos construir a lista à mão com o marcador que quisermos, utilizando as tabelas do HTML. Veremos mais adiante como trabalhar com tabelas.

Vamos ver um exemplo de lista com um quadrado ao invés de uma bolinha e, no último elemento, colocaremos um círculo. Para isso, vamos colocar o atributo type na etiqueta , que irá afetar todos os elementos da lista.

```
<ul type="square">
<li>Elemento 1</li>
<li>Elemento 2</li>
<li>Elemento 3</li>
<li type="circle">Elemento 4</li>
</ul>
```

Que tem como resultado:

- _ Elemento 1
- _ Elemento 2
- _ Elemento 3
- o Elemento 4

LISTAS II

Continuamos estudando as listas do HTML, que nos possibilita criar estruturas atrativas para apresentar a informação.

Listas ordenadas

Neste caso, usaremos as etiquetas (ordered list) e seu fechamento. Cada elemento será igualmente precedido de sua etiqueta .

Como exemplo:

```
<p>Regras de comportamento no trabalho</p>
<ol>
<li>O chefe sempre tem a razão</li>
<li>Em caso de dúvida, aplicar a regra 1</li>
</ol>
```

O resultado é:

Regras de comportamento no trabalho

1. O chefe sempre tem a razão
2. Em caso de dúvida aplicar a regra 1

Do mesmo modo das listas desordenadas, as listas ordenadas oferecem a possibilidade de modificar o estilo. Concretamente, nos é possível especificar o tipo de numeração empregado escolhendo entre números (1,2,3...), letras (a,b,c...) e suas maiúsculas (A,B,C...) e números romanos em suas versões maiúsculas (I,II,III...) e minúsculas (i,ii,iii...).

Para realizar tal seleção temos de utilizar, como para o caso anterior, o atributo type, o qual será situado

dentro da etiqueta . Neste caso, os valores que o atributo pode tomar são:

- 1 Para ordenar por números
- a Por letras do alfabeto
- A Por letras maiúsculas do alfabeto
- i Ordenação por números romanos em minúsculas
- I Ordenação por números romanos em maiúsculas

```
<p>Ordenamos por números</p>
<ol type="1">
<li>Elemento 1</li>
<li>Elemento 2</li>
</ol>
<p>Ordenamos por letras</p>
<ol type="a">
<li>Elemento a</li>
<li>Elemento b</li>
</ol>
<p>Ordenamos por números romanos começando pelo 10</p>
<ol type="i" start="10">
<li>Elemento x</li>
<li>Elemento xi</li>
</ol>
```

O resultado:

- Ordenamos por números
- 1. Elemento 1
- 2. Elemento 2
- Ordenamos por letras
- a. Elemento a
- b. Elemento b
- Ordenamos por números romanos começando pelo 10
- i. Elemento x
- ii. Elemento xi

LISTAS III

Terminamos o assunto das listas estudando as listas de definição. Veremos também a combinação de listas.

Listas de definição

Cada elemento é apresentado junto com sua definição. A definição principal é <dl> e </dl> (definition list). As etiquetas do elemento e sua definição são <dt> (definition term) e <dd> (definition definition) respectivamente.

Aqui lhe propomos um código que poderá clarear este sistema:

```
<p>Dicionário da Língua Portuguesa</p>
<dl>
<dt>Açougue</dt>
<dd>Estabelecimento onde se vendem carnes frescas</dd>
<dt>Colheita</dt>
<dd>Ato de colher os produtos agrícolas</dd>
</dl>
```

O efeito produzido:

- Dicionário da Língua Portuguesa
- Açougue
- Estabelecimento onde se vendem carnes frescas
- Colheita
- Ato de colher os produtos agrícolas

Observe que em cada linha <dd> está deslocada da direção da esquerda. Este tipo de etiquetas é usado muitas vezes com o propósito de criar textos mais ou menos deslocados da direção da esquerda.

O código:

```
<dl>
<dd>Primeiro nível de deslocamento
<dl>
<dd>Segundo nível de deslocamento
<dl>
<dd>Terceiro nível de deslocamento
</dl>
</dl>
</dl>
```

O resultado:

Primeiro nível de deslocamento
Segundo nível de deslocamento
Terceiro nível de deslocamento

Combinando listas

Nada nos impede de utilizar todas estas etiquetas de forma combinada como vimos em outros casos. Dessa forma, podemos conseguir listas mistas como, por exemplo:

```
<p>Cidades do mundo</p>
<ul>
<li>Brasil
<ol>
<li>Rio de Janeiro
<li>Salvador
</ol>
<li>Espanha
<ol>
<li>Madrid
<li>Barcelona
</ol>
</ul>
```

Dessa forma criamos uma lista como esta:

Cidades do mundo

- Brasil
 1. Rio de Janeiro
 2. Salvador
- Espanha
 1. Madrid
 2. Barcelona

CARACTERES ESPECIAIS

Uma página web é vista por diferentes países, que usam conjuntos de caracteres distintos. A linguagem HTML nos oferece um mecanismo pelo qual podemos estar seguros que uma série de caracteres raros serão bem vistos em todos os computadores do mundo, independentemente de seu jogo de caracteres. Estes conjuntos são os caracteres especiais. Quando queremos pôr um desses caracteres numa página, devemos substituí-lo por seu código.

Por exemplo, a letra "á" (a minúscula acentuada) se escreve: "à" de modo que a palavra página seria escrita numa página HTML desse modo: pàgina .

Caracteres especiais básicos

Na realidade estes caracteres se usam em HTML para não confundir um início ou final de etiqueta, umas aspas ou um & com seu correspondente caracter.

â < â >
â â â â â â

Caracteres especiais do HTML 2.0

Á Á À À
É É È È
Í Í Ì Ì
Ó Ó Ò Ò
Ú Ú Ù Ù
á á à à
é é è è
í í ì ì
ó ó ò ò
ú ú ù ù
Ä Ä Â Â
Ë Ê Ê Ê
Ï Î Î Î
Ö Ö Ô Ö
Ü Ü Û Ü
ä ä â â
ë ë ê ê
ï î î î
ö ö ô ô
ü ü û û
Ã Ã å Å
Ñ Ñ Å Å
Õ Ö Ç Ç
ã ã ç ç
ñ ñ Ý Ý
õ õ ý ý
Ø Ø ÿ ÿ
ø ø Þ Þ
Ð Ð þ þ
ð ð Æ Æ
ß ß æ æ

Caracteres especiais do HTML 3.2

¼ ¼ ;
½ ½ ¡ ¡
¾ ¾ £ £
© © ¥ ¥
® ® § §
ª ¢ ¤ ¤
² ² ¦ ¯
³ ³ « «
¹ ¹ ¬ ¬
¯ ¯ ­ ¯
µ µ º º
¶ ¶ ´ ´
· · ¨ ¨
° ° ± ±
¸ ¸ » »
¿ ¿

Outros caracteres especiais

× × ¢ ¢
÷ ÷ € €
“ “ ™ ™
” ” ‰ ‰
Œ OE ƒ f
‡ ‡ † †

 LINKS EM HTML

Até aqui, vimos que uma página web é um arquivo HTML no qual podemos incluir, entre outras coisas, textos formatados ao nosso gosto e imagens que veremos mais adiante. Do mesmo modo, um site web poderá ser considerado como um conjunto de arquivos, principalmente páginas HTML e imagens, que constituem o conteúdo ao qual o navegador tem acesso.

Entretanto, não poderíamos falar de navegador ou de navegação se estes arquivos HTML não estivessem devidamente conectados entre eles e com o exterior de nosso site por meio de links hipertexto. Na verdade, o atrativo original do HTML reside no possível empenho dos conteúdos dos arquivos introduzindo referências sob a forma de links que permitem um acesso rápido à informação desejada. Serviria pouco se tivéssemos na rede páginas isoladas as quais as pessoas não pudessem acessar ou páginas onde não fosse possível ir para outras.

Um link pode ser facilmente detectado em uma página. Basta deslizar o ponteiro do mouse sobre as imagens ou o texto e ver como muda de sua forma original transformando-se por regra geral em uma mão com um dedo indicador. Adicionalmente, estes links costumam ir, no caso dos textos, coloridos e sublinhados para que o usuário não tenha dificuldade em os reconhecer. Se não especificamos o contrário (já teremos a ocasião de explicar como), estes links-texto estarão sublinhados e coloridos de azul. No caso das imagens que servem de link, veremos que estão delimitadas por uma marcação azul por padrão.

Para colocar um link, utilizaremos as etiquetas `<a>` e `</a>`. Dentro da etiqueta de abertura devemos especificar o destino do link. Este destino será introduzido sob a forma de atributo, no qual leva o nome `href`.

A sintaxe geral de um link é portanto da seguinte forma:

```
<a href="destino">conteúdo</a>
```

Sendo o conteúdo um texto ou uma imagem. É a parte da página que se colocará ativa e onde deveremos clicar para acessar ao link.

E o destino por sua vez, será uma página, um correio eletrônico ou um arquivo.

Em função do destino, os links são classicamente agrupados da seguinte forma:

- **Links internos:** os que se dirigem a outras partes dentro da mesma página.
- **Links locais:** os que se dirigem a outras páginas do mesmo site web.
- **Links remotos:** os que se dirigem à páginas de outros sites web.
- **Links com endereços de correio:** para criar uma mensagem de correio dirigido a um endereço.
- **Links com arquivos:** Para que os usuários possam fazer download de arquivos.

Links internos

São os links que apontam a um lugar diferente dentro da mesma página. Este tipo de link é essencialmente utilizado em páginas onde o acesso aos conteúdos pode ser prejudicado devido ao grande tamanho da mesma. Mediante estes links, podemos oferecer aos visitantes a possibilidade de acessar rapidamente ao início e ao final da página, ou também a diferentes parágrafos ou seções.

Para criar um link deste tipo é necessário, além do link origem propriamente dito, um segundo link que será colocado no destino. Vejamos mais claramente como funcionam estes links com um simples exemplo:

Suponhamos que queremos criar um link que aponte ao final da página. O primeiro a fazer será colocar nosso link origem. Colocaremos e escreveremos da seguinte forma:

```
<a href="#abaixo">Ir abaixo</a>
```

Link com o final deste documento para que prove seu funcionamento:

Ir abaixo

Como pode ser visto, o conteúdo do link é o texto "Ir abaixo" e o destino, abaixo, é um ponto da mesma página que ainda não foi definido. Atenção ao símbolo # ; é ele quem especifica ao navegador que o link aponta a uma seção particular.

Em segundo lugar, temos que gerar um link no destino. Este link levará o nome "abaixo" para poder distingui-lo dos outros possíveis links realizados dentro da mesma página. Neste caso, a etiqueta que escreveremos será esta:

```
<a name="abaixo"></a>
```

Na verdade, estes links, mesmo sendo úteis, não são os mais utilizados. A tendência geral é a de criar páginas (arquivos) independentes com tamanhos mais reduzidos linkados entre eles por links locais (que veremos em seguida). Desta forma, evitamos o excesso de tempo de carregamento de um arquivo e a introdução de excesso de informação que possa desviar a atenção do usuário.

Uma aplicação corrente destes links consiste em colocar um pequeno índice ao princípio de nosso documento onde introduzimos links origem às diferentes seções. Paralelamente, ao final de cada seção introduzimos um link que aponta ao índice de forma que possamos guiar o navegador na busca da informação útil para ele.

Links locais

Como dissemos, um site web está constituído de páginas interconexas. No capítulo anterior vimos como linkar diferentes seções dentro de uma mesma página. Resta-nos estudar a maneira de relacionar os distintos documentos HTML que compõem nosso site web.

Para criar este tipo de links, temos que criar uma etiqueta da seguinte forma:

```
<a href="arquivo.html">conteúdo</a>
```

Por regra geral, para uma melhor organização, os sites costumam estar ordenados por diretórios. Estes diretórios costumam conter diferentes seções da página, imagens, audios...É por isso que em muitos casos não nos será válido especificar o nome do arquivo, e sim, o diretório onde nosso arquivo.html está alojado.

Se você já tiver trabalhado com MS-DOS não terá nenhum problema para compreender o modo de funcionamento. Somente deverá ter cuidado em usar a barra "/" no lugar da contra-barras "\".

Para aqueles que não sabem como mostrar um caminho de um arquivo, aqui vai uma série de indicações que lhes vão ajudar a compreender a forma de expressá-los. Não é nada difícil e com um pouco de prática o fará praticamente sem pensar.

1. Há de situar mentalmente no diretório no qual se encontra a página com o link.
2. Se a página destino está em um diretório incluído dentro do diretório no qual nos encontramos, temos de marcar o caminho numerando cada um dos diretórios pelos quais passamos até chegar ao arquivo e separando-os pelo símbolo barra "/". No final, obviamente, escrevemos o arquivo.
3. Se a página destino encontra-se em um diretório que inclui o da página com o link, temos que escrever dois pontos seguidos e uma barra "../" tantas vezes quantas forem os níveis que subirmos na hierarquia de diretórios, até chegar no diretório onde está localizado o arquivo destino.
4. Se a página encontra-se em outro diretório não incluído nem incluído do arquivo origem, teremos que subir com a regra 3 por meio de ".." até encontrar o diretório que englobe o diretório que contém a página destino. A seguir faremos como a regra 2. Escreveremos todos os diretórios pelos quais passamos até chegar ao arquivo.

Exemplo:

Para esclarecer este ponto podemos fazer um exemplo a partir da estrutura de diretórios da imagem.

Para fazer um link desde index.html para yyy.html:

```
<a href="secao1/paginas/yyy.html">conteúdo</a>
```

Para fazer um link desde xxx.html para yyy.html:

```
<a href="../secao1/paginas/yyy.html">conteúdo</a>
```

Para fazer um link desde yyy.html para xxx.html:

```
&<a href="../../secao2/xxx.html">conteúdo</a>
```

Os links locais podem por sua vez já apontar mais precisamente a uma seção concreta, ao invés da página em geral. Este tipo de link costuma ser um híbrido de interno e local. A sintaxe é deste tipo:

```
<a href="arquivo.html#secao">conteúdo</a>
```

Como para os links internos, neste caso temos que marcar a seção com outro link do tipo:

```
<a name="secao"></a>
```

Como exemplo temos aqui um link que aponta ao capítulo anterior ao final da página.

Links externos, de correio e para arquivos

Para terminar o tema dos links veremos os três últimos tipos de links que havíamos assinalado.

Links remotos

São os links que se dirigem às páginas que se encontram fora do nosso site web, ou seja, qualquer outro documento que não faz parte de nosso site.

Este tipo de link é muito comum e não representa nenhuma dificuldade. Simplesmente colocamos no atributo HREF de nossa etiqueta <A> a URL ou endereço da página com a qual queremos linkar. Será algo parecido a isto:

```
<a href="http://www.yahoo.com.br">ir a yahoo.com.br</a>
```

Somente cabe destacar que todos os endereços web (URLs) começam por **http://**. Isto indica que o protocolo pelo qual se acessa é HTTP, o utilizado na web. Não devemos nos esquecer de colocá-lo porque senão os links serão tratados como links locais em nosso site.

Outra coisa interessante é que não temos necessariamente que linkar com uma página web com o protocolo HTTP. Também podemos acessar recursos através de outros protocolos como o FTP. Em tal caso, os endereços dos recursos não começarão com http:// e sim por ftp://.

Links a endereços de correio

Os links a endereços de correio são aqueles em que ao clicá-los nos abre uma nova mensagem de

correio eletrônico dirigido a um determinado endereço de mail. Estes links são muito habituais nas páginas web e é a maneira mais rápida de oferecer ao visitante uma via para o contato com o proprietário da página.

Para colocar um link dirigido a um endereço de correio colocamos **mailto:** no atributo href do link, seguido do endereço de correio ao qual se deve dirigir o link.

```
<a href="mailto: andre@andrecavalcante.com ">andre@andrecavalcante.com</a>
```

Links com arquivos

Este não é um tipo de link propriamente dito, mas o assinalamos aqui porque é um tipo de link muito habitual e que apresenta alguma complicação para o usuário novato.

O mecanismo é o mesmo que conhecemos nos links locais e nos remotos, com a única particularidade de que em vez de estar dirigidos para uma página web, está dirigido para um arquivo de outro tipo.

Se queremos linkar com um arquivo meu_arquivo.zip que se encontra no mesmo diretório que a página, escreveríamos um link assim:

```
<a href="meu_arquivo.zip">Baixar meu_arquivo.zip</a>
```

Se clicamos um link deste tipo nosso navegador baixará o arquivo, fazendo a pergunta típica se: "Deseja abrir o arquivo ou salvá-lo no computador?".

Se queremos linkar outro tipo de arquivos como um PDF ou mundo VRML (Realidade Virtual para Internet) continuamos fazendo da mesma maneira. O navegador, se reconhece o tipo de arquivo, é o responsável de abri-lo utilizando o conector adequado para isso. Assim, se por exemplo linkamos com um PDF colocará o programa Acrobat Reader em funcionamento para mostrar os conteúdos. Se linkamos com um mundo VRML colocará em funcionamento o plug-in que o usuário tenha instalado para ver os mundos virtuais (Cosmo, por exemplo).

Este seria um exemplo de link a um documento PDF.

```
<a href="meu_documento.pdf">Baixar o PDF</a>
```

Imagens em HTML

Sem dúvida, um dos aspectos mais vistosos e atrativos das páginas web é o grafismo. A introdução em nosso texto de imagens pode nos ajudar a explicar mais facilmente nossa informação e dar um ar muito mais estético. Porém, o abuso pode nos conduzir a uma sobrecarga que se traduz em uma distração para o navegante, quem terá mais dificuldade em encontrar a informação necessária, e um maior tempo para carregar a página o que pode ser de um efeito nefasto se nosso visitante não tem uma boa conexão ou se é um pouco impaciente.

Neste capítulos não explicaremos como criar, nem como tratar as imagens, unicamente diremos que para isso utilizam-se aplicações como Paint Shop Pro, Photoshop ou Corel Draw. Também não explicaremos as particularidades de cada tipo de arquivo GIF ou JPG e a forma de otimizar nossas imagens. A este assunto será dedicado em um futuro capítulo.

As imagens são armazenadas em forma de arquivos, principalmente GIF (para desenhos) ou JPG (para fotos). Estes arquivos podem ser criados por nós mesmos ou podem ser baixados gratuitamente em sites web especializados.

Sendo assim, nestes primeiros capítulos nos limitaremos a explicar como inserir e alinhar devidamente em nossa página uma imagem já criada.

A etiqueta que utilizaremos para inserir uma imagem é `<img>` (image). Esta etiqueta não possui seu fechamento correspondente e nela temos de especificar obrigatoriamente o paradeiro de nosso arquivo mediante o atributo `src` (source).

A sintaxe fica então da seguinte forma:

```

```

Para expressar o caminho, faremos da mesma forma que vimos para os links. As regras continuam sendo as mesmas, o único que muda é que, no lugar de uma página destino, o destino é um arquivo gráfico.

Além deste atributo, obviamente indispensável para a visualização da imagem, a etiqueta `<img>` nos propõe outra série de atributos de maior ou menor utilidade:

Atributo alt

Entre aspas deste atributo, colocaremos uma brevíssima descrição da imagem. Esta etiqueta não é indispensável, mas apresenta várias utilidades.

Primeiramente, durante o processo de carregamento da página, quando a imagem não tiver sido ainda

carregada, o navegador mostrará esta descrição, com a qual o navegante poderá ter uma idéia do que se trata neste caso.

Isto não é tão trivial se temos em conta que alguns usuários navegam pela rede com uma opção do navegador que desativa a amostra de imagens, com o que tais pessoas poderão sempre saber de que se trata o gráfico e eventualmente mudar o modo com imagens para visualizar.

Além disso, determinadas aplicações para incapacitados ou para telefones vocais que não mostram imagens oferecem a possibilidade de lê-las, o que nunca é demais pensar nestes coletivos.

Em geral, podemos considerar como aconselhável o uso deste atributo salvo para imagens de pouca importância e absolutamente indispensável se a imagem em questão serve de link.

Atributos height e width

Definem a altura e largura respectivamente da imagem em pixels.

Todos os arquivos gráficos possuem umas dimensões de largura e altura. Estas dimensões podem ser obtidas a partir do próprio designer gráfico ou também, clicando com o botão direito sobre a imagem vista pelo navegador para logo escolher propriedades sobre o menu que se desdobra.

O fato de explicitar em nosso código as dimensões de nossas imagens ajuda ao navegador a confeccionar a página da forma que nós desejamos inclusive antes das imagens serem baixadas.

Assim, se as dimensões das imagens tiverem sido proporcionadas, durante o processo de carregamento, o navegador reservará o espaço correspondente a cada imagem criando uma planificação correta. O usuário poderá começar a ler tranquilamente o texto sem que este se mova de um lado a outro cada vez que se carregue uma imagem.

Além desta utilidade, o alterar os valores destes atributos, é uma forma imediata de redimensionar nossa imagem. Este tipo de utilidade não é aconselhável visto que, se o que pretendemos é aumentar o tamanho, a perda da qualidade da imagem será muito sensível. Inversamente, se desejamos diminuir seu tamanho, estaremos usando um arquivo maior do que o necessário para a imagem que estamos mostrando, com o que aumentamos o tempo de descarregamento de nosso documento desnecessariamente.

É importante insistir neste ponto já que muitos estereótipos têm o péssimo costume de criar gráficos pequenos redimensionando a imagem por meio desses atributos a partir de arquivos de tamanho descomunal. Temos que pensar que o tamanho de uma imagem com umas dimensões da metade não se reduz à metade, e sim, que é aproximadamente 4 vezes inferior.

Atributo border

Define o tamanho em pixels do quadro que rodeia a imagem.

Dessa forma, podemos re-enquadrar nossa imagem se desejamos. É particularmente útil quando desejamos eliminar a borda que aparece quando a imagem serve de link. Em tal caso teremos que especificar border="0".

Atributos vspace e hspace

Serve para indicar o espaço livre, em pixels, que tem que ser colocado entre a imagem e outros elementos que a rodeiam, como texto, outras imagens, etc.

Atributo lowsrc

Com este atributo podemos indicar um arquivo de baixa resolução. Quando o navegador detecta que a imagem tem este atributo, primeiro descarrega e mostra a imagem de baixa resolução (que ocupa muito pouco e que se transfere muito rápido). Posteriormente, descarrega e mostra a imagem de resolução adequada (assinalada com o atributo src, que se supõe que ocupará mais e que será mais lenta de se transferir).

Está atributo está em desuso, mesmo supondo uma vantagem considerável para que o descarregamento inicial se realize mais rápido e que um visitante possa ver uma amostra da imagem enquanto se descarrega a imagem real.

Dica: Utilizar imagens como links

Isto quer dizer que uma imagem, assim como um texto, pode servir de link. Visto a estrutura dos links podemos muito facilmente adivinhar o tipo de código necessário.

```
<a href="arquivo.html"></a>
```

Exemplo prático

Será óbvio para os leitores, fazer agora uma página que contenha uma imagem várias vezes repetidas, mas com diferentes atributos.

- Uma das vezes que saia deve ser mostrada com seu tamanho original e com uma borda de 3 pixels.
- Em outra ocasião a imagem aparecerá sem borda, com sua mesma altura e com uma largura superior a original.
- Também mostraremos a imagem sem borda, com sua mesma largura e com uma altura superior a original.
- Por último, mostraremos a imagem com uma altura e largura maiores que as originais, mas proporcionalmente igual que antes.

Vamos utilizar esta imagem para fazer o exercício:

As dimensões originais da imagem são 28x21, o código fonte seria então da seguinte forma:

```

<br>
<br>

<br>
<br>

<br>
<br>

```

Alinhamento de imagens com HTML

Vimos em seu momento o atributo align que nos permitia alinhar o texto à direita, esquerda ou no centro de nossa página. Dissemos que este atributo não era exclusivo da etiqueta e sim, que podia ser encontrado em outro tipo de etiquetas.

Sendo assim, é uma dessas etiquetas que aceitam este atributo, mesmo sendo, neste caso, o funcionamento diferente.

Para alinhar uma imagem horizontalmente podemos fazer da mesma forma que o texto, ou seja, utilizando align dentro de uma etiqueta <p> ou <div>. Neste caso, o que incluiremos dentro desta etiqueta será a imagem no lugar do texto:

Este código mostrará a imagem no centro.

Entretanto, já dissemos que a etiqueta pode aceitar o atributo align. Neste caso, a utilidade que lhe damos é diferente da anterior.

O fato de utilizar o atributo align dentro da etiqueta nos permite, no caso de dar os valores left ou right, justificar a imagem do lado que desejamos uma vez que recheamos com texto o lado oposto. Dessa forma, incorporamos nossas imagens dentro do texto de uma maneira simples.

Aqui se pode ver o tipo de código a criar para obter tal efeito:

```
<p>
Texto tão extenso quanto quisermos que cubra a parte esquerda da
imagem. Continuo colocando texto para que se veja o efeito, Bla bla bla bla bla bla...
</p>
```

Ficaria assim:

Texto tão extenso quanto quisermos que cubra a parte esquerda da imagem. Continuo colocando texto para que se veja o efeito, Bla bla bla bla bla bla...

```
<p>
Texto tan extenso como queramos que cubrirá la parte derecha de la
imagen. Sigo poniendo texto para que se vea el efecto, Bla bla bla bla bla bla...
</p>
```

Ficaria assim:

Texto tan extenso como queramos que cubrirá la parte derecha de la imagen. Sigo poniendo texto para que se vea el efecto, Bla bla bla bla bla bla...

Se em algum momento desejarmos preencher esse espaço lateral, podemos passar a uma zona livre introduzindo uma quebra de linha
 dentro do qual acrescentaríamos um atributo: clear

Sendo então, etiquetas do tipo:

```
<br clear="left">
```

Pulará verticalmente até encontrar a lateral esquerdo livre.

<br clear="right">

Pulará verticalmente até encontrar a lateral direita livre.

<br clear="all">

Pulará verticalmente até encontrar ambas laterais livres.

Existe outro tipo de valores que pode adotar o atributo align dentro da etiqueta . Isto é relativo ao alinhamento vertical da imagem.

Supomos que escrevemos uma linha ao lado de nossa imagem. Esta linha pode ficar, por exemplo acima, abaixo ou no meio da imagem. Ainda assim, pode que uma mesma linha tenhamos várias imagens de alturas diferentes que podem ser alinhadas de distintas formas.

Estes valores adicionais são:

Top

Ajusta a imagem à parte mais alta da linha. Isto quer dizer que, se existe uma imagem mais alta, ambas imagens apresentarão a borda superior na mesma altura.

Bottom

Ajusta o baixo da imagem ao texto.

Absbottom

Colocará a borda inferior da imagem ao nível do elemento mais baixo da linha.

Middle

Faz coincidir a base da linha de texto com o meio vertical da imagem.

Absmiddle

Ajusta a imagem ao meio absoluto da linha.

Estas explicações, que podem ser um pouco complicadas, serão mais facilmente assimiladas com um pouco mais de prática.

Falta explicar como introduzir debaixo da imagem um pé de foto ou uma explicação. Para isso, teremos que ver antes de mais nada as tabelas, nos próximos capítulos...

Formatos gráficos para páginas web

O componente gráfico das páginas web tem muita importância, é que o que faz com que estas seja vistosas e o que nos permite aplicar nossa criatividade para fazer o design do site uma tarefa agradável. É também uma ferramenta para aproximar os sites ao mundo onde vivemos, entretanto, é também o causador de graves erros nas páginas e fazer destas, em alguns casos, um martírio para o visitante.

As noções básicas para o uso de arquivos gráficos são simples, conhecê-las, mesmo que seja ligeiramente, nos ajudará a criar sites agradáveis e rápidos. Não cometer erros no uso das imagens é fundamental, mesmo que não seja um designer e as imagens que utiliza são feias, utilize-as bem e assim, estará tornando mais agradável a visitas às suas páginas.

Tipos de arquivos

Em Internet se utilizam principalmente dois tipos de arquivos gráficos GIF e JPG, pensados especialmente para otimizar o tamanho que ocupam em disco, já que os arquivos pequenos se transmitem mais rapidamente pela Rede.

O formato de arquivo GIF se usa para as imagens que tenham desenhos, enquanto que o formato JPG se usa para as fotografias. Os dois comprimem as imagens para salvá-las. A forma de comprimir a imagem que utiliza cada formato é o que os faz ideais para um propósito ou outro.

Adicionalmente, pode-se usar um terceiro formato gráfico nas páginas web, o PNG. Este formato não tem tanta aceitação como o GIF ou o JPG, por várias razões, entre elas: o desconhecimento do formato por parte dos desenvolvedores, que as ferramentas habituais para tratar gráficos (como por exemplo, Photoshop) geralmente não suportam, e que os navegadores antigos também têm problemas para visualizá-las. Entretanto, o formato se comporta muito bem quanto a compreensão e a qualidade do gráfico conseguinte, pelo que seria útil se chega a estender seu uso.

GIF

Além de ser um arquivo ideal para as imagens que estão desenhadas, tem muitas outras características que são importantes e úteis.

Compresión: É muito boa para desenhos, como já foi dito. Inclusive pode ser interessante se a imagem é muito pequena, mesmo que seja uma foto.

Transparência: É uma utilidade para definir certas partes do desenho como transparentes. Desse modo, podemos colocar as imagens sobre distintos fundos sem que se veja o quadrado onde está es está inscrito o desenho, vendo em troca a silhueta do desenho em questão. Para criar um gif transparente devemos utilizar um programa de desenho gráfico, com o qual podemos indicar que cores do desenho queremos que sejam transparentes. Geralmente, definimos a transparência quando vamos salvar o tráfico.

Cores: Com este formato gráfico podemos utilizar conjuntos de 256 cores ou menos. Este é um detalhe muito importante, visto que quanto menos cores utilizarmos na imagem, em geral, menos ocupará o arquivo. Às vezes, mesmo utilizando menos cores em um gráfico, este não perde muita qualidade, chegando a ser inapreciável à vista. Em alguns programas podemos modificar a quantidade de cores ao salvar o arquivo, em outros, fazemos enquanto criamos o gráfico.

JPG

Vejamos agora quais são as características fundamentais do formato JPG:

Compressão: Tal como dissemos anteriormente, sua gama de compressão torna ideal este formato para salvar fotografias.

Além disso, com JPG podemos definir a qualidade da imagem, com qualidade baixa o arquivo ocupará menos, e vice-versa.

Transparência: Este formato não tem possibilidade de criar áreas transparentes. Se desejamos colocar uma imagem com uma área que pareça transparente procederemos assim: com nosso programa de desenho gráfico faremos com que o fundo da imagem seja o mesmo que o da página onde queremos colocá-la. Em muitos casos, os fundos da imagem e a página parecerão o mesmo.

Cores: JPG trabalha sempre com 16 milhões de cores, ideal para fotografias.

Otimizar arquivos

Para que as imagens ocupem o menos possível e se transfiram rapidamente pela Rede devemos aprender a otimizar os arquivos gráficos. Para isso, devemos fazer o seguinte:

Para os arquivos GIF: Reduziremos o número de cores de nossa aquarela. Isto se faz com nosso editor gráfico, em muitos casos poderemos fazer ao salvar o arquivo.

Para os arquivos JPG: Ajustaremos a qualidade do arquivo quando estivermos salvando. Este formato nos permite baixar muito a qualidade da imagem sem que esta perca muito em seu aspecto visual.

Parte desta

imagem é

transparente

Uma fotografia com formato JPG

Photoshop é uma ferramenta excelente para otimizar arquivos. Vendo várias cópias

É imprescindível dispor para otimizar a imagem de uma boa ferramenta que nos permita configurar estas características da imagem com liberdade e facilidade. Photoshop 5.5 ou 6 é um programa bastante recomendável, pois incorpora uma opção que se chama "Salvar para Web" com a qual podemos definir as cores do gif, a qualidade do JPG e outras opções em várias amostras. Assim, com todas as opções configuráveis, vendo os resultados do tamanho do arquivo, podemos otimizar a imagem de uma maneira precisa com os resultados que desejamos.

Também existem no mercado outros programas que nos permitem otimizar estas imagens de maneira surpreendente. Uma vez que criamos a imagem, a passamos por estes programas e nos comprimem ainda mais o arquivo, fazendo-lhe rápido de transferir e, portanto, mais otimizado para Internet. Ao ser estas utilidades tão especializadas, os resultados costumam ser melhores que com os programas de edição gráfica.

Exemplos de otimizadores gráficos:

- WebGraphics Optimizer
- ProJPG, GIF Imation

E com versões On-line:

- JPG - GIF Crunchers
- GIF Wizard

As cores e HTML

As cores possuem um papel muito importante na composição de webs. São indicadas em valores RGB, ou seja, que para conseguir uma cor qualquer misturaremos quantidades de Vermelho, Verde e Azul.

Os valores RGB são indicados em numeração hexadecimal, em base 16. (Os dígitos podem crescer até 16). Como não existem tantos dígitos numéricos se utilizam as letras da A à F.

0=0 4=4 8=8 C=12

1=1 5=5 9=9 D=13

2=2 6=6 A=10 E=14

3=3 7=7 B=11 F=15

Para conseguir uma cor, misturaremos valores desta maneira:

RRGGBB

Onde cada valor pode crescer desde 00 até FF.

Exemplo: Como se mudaria a fonte para escrever em vermelho:

```
<font color="#FF0000">Vermelho</font>
```

Ao Atributo **cor** lhe damos um valor RGB em formato hexadecimal. O caractere # se coloca ao princípio da expressão.

Outras cores:

Laranja #FF8000

Verde turquesa #339966

Azul escuro #000080

Cores compatíveis em todos os sistemas

Como as páginas web têm que ser vistas por todos os usuários, e os sistemas que utilizam para entrar são distintos, há que utilizar cores compatíveis com a paleta de todos eles.

A forma de conseguir isto é limitando nossas cores aos que se podem conseguir utilizando a seguinte norma:

Utilizaremos sempre

podemos escolher a mais adequada.

Tabela de cor

estes valores:

00

33

66

99

CC

FF

Exemplos: #3366FF #FF9900 #666666

Consegue-se as seguintes cores:

#000000 #000033 #000066 #000099 #0000CC #0000FF
#003300 #003333 #003366 #003399 #0033CC #0033FF
#006600 #006633 #006666 #006699 #0066CC #0066FF
#009900 #009933 #009966 #009999 #0099CC #0099FF
#00CC00 #00CC33 #00CC66 #00CC99 #00CCCC #00CCFF
#00FF00 #00FF33 #00FF66 #00FF99 #00FFCC #00FFFF
#330000 #330033 #330066 #330099 #3300CC #3300FF
#333300 #333333 #333366 #333399 #3333CC #3333FF
#336600 #336633 #336666 #336699 #3366CC #3366FF
#339900 #339933 #339966 #339999 #3399CC #3399FF
#33CC00 #33CC33 #33CC66 #33CC99 #33CCCC #33CCFF
#33FF00 #33FF33 #33FF66 #33FF99 #33FFCC #33FFFF
#660000 #660033 #660066 #660099 #6600CC #6600FF
#663300 #663333 #663366 #663399 #6633CC #6633FF
#666600 #666633 #666666 #666699 #6666CC #6666FF
#669900 #669933 #669966 #669999 #6699CC #6699FF
#66CC00 #66CC33 #66CC66 #66CC99 #66CCCC #66CCFF
#66FF00 #66FF33 #66FF66 #66FF99 #66FFCC #66FFFF
#990000 #990033 #990066 #990099 #9900CC #9900FF
#993300 #993333 #993366 #993399 #9933CC #9933FF
#996600 #996633 #996666 #996699 #9966CC #9966FF
#999900 #999933 #999966 #999999 #9999CC #9999FF
#99CC00 #99CC33 #99CC66 #99CC99 #99CCCC #99CCFF
#99FF00 #99FF33 #99FF66 #99FF99 #99FFCC #99FFFF
#CC0000 #CC0033 #CC0066 #CC0099 #CC00CC #CC00FF
#CC3300 #CC3333 #CC3366 #CC3399 #CC33CC #CC33FF
#CC6600 #CC6633 #CC6666 #CC6699 #CC66CC #CC66FF
#CC9900 #CC9933 #CC9966 #CC9999 #CC99CC #CC99FF
#CCCC00 #CCCC33 #CCCC66 #CCCC99 #CCCCCC #CCCCFF
#CCFF00 #CCFF33 #CCFF66 #CCFF99 #CCFFCC #CCFFFF
#FF0000 #FF0033 #FF0066 #FF0099 #FF00CC #FF00FF
#FF3300 #FF3333 #FF3366 #FF3399 #FF33CC #FF33FF
#FF6600 #FF6633 #FF6666 #FF6699 #FF66CC #FF66FF
#FF9900 #FF9933 #FF9966 #FF9999 #FF99CC #FF99FF

#FFCC00 #FFCC33 #FFCC66 #FFCC99 #FFCCCC #FFCCFF
#FFFFFF00 #FFFF33 #FFFF66 #FFFF99 #FFFFCC #FFFFFF

TABELAS EM HTML

Uma tabela em um conjunto de células organizadas dentro das quais podemos alojar distintos conteúdos. À princípio, poderia parecer que as tabelas são raramente úteis e que podem ser utilizadas simplesmente para listar dados como agendas, resultados e outros dados de uma forma organizada. Nada mais distante da realidade.

Hoje, grande parte dos desenhistas de páginas baseia seu planejamento neste tipo de engenhoca. De fato, uma tabela nos permite organizar e distribuir os espaços da melhor forma. Pode nos ajudar a gerar textos em colunas como os jornais, prefixar os tamanhos ocupados por distintas seções da página ou colocar de uma maneira simples uma legenda a uma imagem.

Pode ser que a princípio seja um pouco complicado trabalhar com estas estruturas mas, se desejamos criar uma página de qualidade, mais cedo ou mais tarde teremos que nos ver com elas e nos dar conta das possibilidades que nos oferecem.

Para começar, nada mais simples do que pelo princípio: as tabelas são definidas pelas etiquetas `<table>` e `</table>`

Dentro dessas duas etiquetas colocaremos todas as outras etiquetas, textos e imagens que darão forma e conteúdo à tabela.

As tabelas são descritas por linhas da esquerda para direita. Cada uma destas linhas é definida por outra etiqueta e seu fechamento: `<tr>` e `</tr>`

Ainda assim, dentro de cada linha, haverá diferentes células. Cada uma dessas células será definida por outro par de etiquetas: `<td>` e `</td>`. Dentro desta etiqueta será onde colocaremos nosso conteúdo.

Aqui temos um exemplo de estrutura de tabela:

```
<table>
<tr>
<td>Célula 1, linha 1</td>
<td>Célula 2, linha 2</td>
</tr>
<tr>
<td>Célula 1, linha 2</td>
<td>Célula 2, linha 2</td>
</tr>
</table>
```

O resultado:

Célula 1, linha 1 Célula 2, linha 1

Célula 1, linha 2 Célula 2, linha 2

NOTA: Até aqui vimos todas as etiquetas que necessitamos conhecer para criar as tabelas. Existem outras etiquetas, mas o que podemos conseguir com elas se pode conseguir também usando as que já vimos.

Por exemplo, assinalamos a etiqueta `<th>`, que serve para criar uma célula cujo conteúdo esteja formatado como um título ou cabeçalho da tabela. Na prática, o que faz é colocar em negrito e centralizado o conteúdo dessa célula, o que se pode conseguir aplicando as correspondentes etiquetas dentro da célula. Assim:

```
<td align="center"><b>Conteúdo da célula</b></td>
```

A partir desta ideia simples, as tabelas adquirem outra magnitude quando lhes incorporamos toda uma bateria de atributos aplicados sobre cada tipo de etiquetas que as compõem. Ao longo dos próximos capítulos nos aprofundaremos no estudo desses atributos de forma a proporcionar-lhe tudo que é útil e indispensável para um bom arranque em páginas.

Tabelas em HTML. Atributos para filas e células.

Vimos no capítulo anterior que as tabelas estão compostas de linhas que, por sua vez, contém células. As células são delimitadas pelas etiquetas `<td>` ou pelas etiquetas `<th>` (se queremos texto em negrito e centralizado) e constituem um entorno independente do resto do documento. Isto quer dizer que:

- Podemos usar praticamente qualquer tipo de etiqueta dentro da etiqueta `<td>` para, desta forma, dar forma a seu conteúdo.
- As etiquetas situadas no interior da célula não modificam o resto do documento.
- As etiquetas de fora da célula não são tidas em conta por esta.

Sendo assim, podemos especificar o formato de nossas células a partir de etiquetas introduzidas em seu interior ou mediante atributos colocados dentro da etiqueta de célula <td> ou também, em alguns casos, dentro da etiqueta <tr>, se desejamos que o atributo seja válido para toda a linha. A forma mais útil e atual de dar forma às células é a partir das folhas de estilo em cascata que já teremos a oportunidade de abordar mais adiante.

Vemos em seguida alguns atributos úteis para a construção de nossas tabelas. Começamos vendo alguns atributos que nos permite modificar uma célula em concreto ou toda uma linha.

align Justifica o texto da célula da mesma forma que se fosse o de um parágrafo.

valign Podemos escolher se queremos que o texto apareça acima (top), no centro (middle) ou abaixo (bottom) da célula.

bgcolor Dá cor à célula ou escolha de linha.

bordercolor Define a cor da borda.

Outros atributos que podem ser unicamente atribuídos a uma célula e não ao um conjunto de células de uma linha são:

background Permite-nos colocar um fundo para a célula a partir de um link a uma imagem.

height Define a altura da célula em pixels ou porcentagem.

width Define a largura da célula em pixels ou porcentagem

colspan Expande um célula horizontalmente.

rowspan Expande um célula verticalmente.

Nota: O atributo height não funciona em todos os navegadores, ademais, seu uso não está muito estendido. As células em geral, têm a altura que necessitam para caber todo o conteúdo que tenha inserido, ou seja, crescem o suficiente para que caiba o que colocamos dentro. O atributo width sim que funciona em todos os navegadores e deve ser utilizado constantemente. Se lhe atribuímos uma largura à célula, a largura será respeitada e se a tal célula tiver muito texto ou qualquer outro conteúdo, a célula crescerá um tanto para baixo quanto for o necessário para caber o que colocamos. Uma observação neste último parágrafo. Trata-se que se definimos uma célula de uma largura 100, por exemplo e colocamos na célula um conteúdo como uma imagem que meça mais de 100 pixels, a célula crescerá em horizontalmente tanto quanto for necessário para que a imagem caiba. Se o elemento, mesmo mais largo, fosse divisível (como um texto) a largura seria respeitada e o texto cresceria para baixo, ou o que é o mesmo, em altura, como assinalamos no parágrafo anterior.

Estes últimos quatro atributos descritos são de grande utilidade. Concretamente, height e width nos ajudam a definir as dimensões de nossas células de uma forma absoluta (em pixel ou em pontos de tela) ou de uma forma relativa, ou seja, por porcentagens referentes ao tamanho total da tabela.

Por exemplo:

<td width="80">

Dará uma largura de 80 pixels à célula. Entretanto,

<td width =80%>

Dará uma largura à célula do 80% da largura da tabela.

Há de ter em conta que, definidas as dimensões das células, o navegador vai fazer o que bondosamente puder para satisfazer ao programador. Isto quer dizer que pode que em algumas ocasiões o resultado que obtemos não seja o esperado. Concretamente, se o texto apresenta uma palavra excessivamente comprida pode que a largura da célula se veja aumentada para manter a palavra na mesma linha. Por outro lado, se o texto é muito grande, a célula aumentará sua para poder mostrar todo o seu conteúdo. Analogamente se, por exemplo, definimos duas larguras distintas às células de uma mesma coluna, o navegador não saber qual atender. É por isso que é conveniente ter bem claro desde o princípio como é a tabela que queremos desenhar. Não é demais saber se pré-desenharmos no papel se a complexidade é importante. O HTML em geral é fácil, mas as tabelas podem converte-se em um verdadeiro quebra-cabeça se não chegamos a compreendê-las devidamente.

Os atributos rowspan e colspan são também utilizados freqüentemente. Graças a isso é possível expandir células fundindo estas com suas vizinhas. O valor que podem tomar estas etiquetas é numérico. O número representa a quantidade de células fundidas.

Assim,

<td colspan="2">

Fundirá a célula em questão com sua vizinha direita.

Esta célula tem um colspan="2"

Célula normal Outra célula

Do mesmo modo,

<td rowspan="2">

Célula

Esta célula tem `rowspan="2"`, por isso Normal tem fundida a célula abaixo.

Outra célula normal

A célula expandirá para baixo fundindo-se com a célula inferior.

O resto dos atributos apresentados apresenta uma utilidade e um uso bastante óbvio. Por isso, os deixamos a sua própria investigação.

Tabelas em HTML. Atributos da tabela e conclusão.

Além dos atributos específicos de cada célula ou linha, as tabelas podem ser adicionalmente formatadas a partir dos atributos que nos oferece a própria etiqueta `<table>`. A seguir, mostramos aqueles que nos podem parecer à princípio mais importantes:

align Alinha horizontalmente a tabela em relação ao seu entorno.

background Permite-nos colocar um fundo para a tabela desde um link a uma imagem.

bgcolor Dá cor de fundo à tabela.

border Define o número de pixels da borda principal.

bordercolor Define a cor da borda.

cellpadding Define, em pixels, o espaço entre as bordas da célula e o conteúdo da mesma.

cellspacing Define o espaço entre as bordas (em pixels).

height Define a altura da tabela em pixels ou porcentagem.

width Define a largura da tabela em pixels ou porcentagem.

Os atributos que definem as dimensões, `height` e `width`, funcionam de uma maneira análoga ao das células, tal como vimos no capítulo anterior. Contrariamente, o atributo `align` não nos permite justificar o texto de cada uma das células que compõem a tabela, mas sim que permite, justificar a própria tabela em relação ao seu entorno.

Vamos colocar três exemplos de alinhamento de tabelas, centralizadas, alinhadas à direita e à esquerda.

Exemplo de tabela centralizada

Esta tabela está centralizada (`align="center"`). Tem somente uma célula.

Este seria um texto qualquer colocado ao lado de uma tabela centralizada.

Exemplo de tabela alinhada à direita

Para que seja visto o efeito de alinhamento da tabela devemos colocar um texto ao lado e o texto irá rodear a tabela, assim como ocorreria como as imagens alinhadas a um lado.

Esta tabela está alinhada à direita (`align="right"`). Tem somente uma célula.

Exemplo de tabela alinhada à esquerda

Para que seja visto o efeito de alinhamento da tabela, devemos colocar um texto ao lado e o texto irá rodear a tabela, assim como ocorreria com as imagens alinhadas a um lado.

Esta tabela está alinhada à esquerda (`align="left"`). Tem somente uma célula.

Os atributos `cellpadding` e `cellspacing` ajudarão a dar a nossa tabela um aspecto mais estético. À princípio pode nos parecer um pouco confuso seu uso mas com um pouco de prática já será suficiente para saber utilizá-los.

Na seguinte imagem podemos ver graficamente o significado destes atributos:

Você mesmo pode comprovar que os atributos definidos para uma célula têm prioridade em relação aos definidos para uma tabela. Podemos definir, por exemplo, uma tabela com a cor de fundo vermelha e uma

das células com a cor de fundo verde. E assim, toda a tabela será vista de cor vermelha menos a célula verde. Da mesma forma, podemos definir uma cor azul para as bordas da tabela e fazer com que uma célula particular seja mostrada com uma borda vermelha. (Apesar de que isto não funcionará em todos os navegadores devido ao fato de alguns não reconhecerem o atributo bordercolor.)

Tabela de cor

vermelha de fundo

O atributo bgcolor da tabela está em vermelho.

Célula normal Esta célula está em verde. Tem o atributo bgcolor na cor verde

Tabelas aninhadas

O uso de tabelas aninhadas também é muito útil. Da mesma forma que podíamos incluir listas dentro de outras listas, as tabelas podem ser incluídas dentro de outras. Assim, podemos incluir uma tabela dentro da célula de outra. A forma de funcionamento continua sendo o mesmo apesar de que a situação pode se complicar se o número de tabelas incluídas dentro de outras for elevado.

Vamos ver um código de um aninhamento de tabelas. Veremos primeiro o resultado e em seguida, o código, pois assim conseguiremos entendê-lo melhor.

Célula da tabela principal

Tabela aninhada, célula 1 Tabela aninhada, célula 2

Tabela aninhada, célula 3 Tabela aninhada, célula 4

Este seria o código:

```
<table cellspacing="10" cellpadding="10" border="3">
<tr>
<td align="center">
Célula da tabela principal
</td>
<td align="center">
<table cellspacing="2" cellpadding="2" border="1">
<tr>
<td>Tabela aninhada, célula 1</td>
<td>Tabela aninhada, célula 2</td>
</tr>
<tr>
<td>Tabela unida, célula 3</td>
<td>Tabela aninhada, célula 4</td>
</tr>
</table>
</td>
</tr>
</table>
```

Exemplos práticos

Estas são as informações que pretendíamos transmitir-lhes sobre as tabelas em HTML. Agora seria importante fazer algum exemplo de realização de uma tabela um pouco mais complexa. Por exemplo, a seguinte:

Animais em perigo de extinção

Nome Cabeças Previsão 2010 Previsão 2020

Baleia 6000 4000 1500

Urso Pardo 50

Lince 10

0

Tigre 300 210

Outro exemplo de tabela com a qual podemos praticar:

Climas de América do Sul

Venezuela Argentina

Norte da América de

Sul. Países como:

Colômbia

Equador

Perú

Sul da América do Sul.

Países como:

Chile

Uruguai

Paraguai

Floresta tropical, clima de savana,

clima marítimo com invernos

secos.

Climas marítimos com verões

secos, com invernos secos,

climas frios, clima de estepe,

clima desértico.



FORMULÁRIOS HTML

Até agora vimos a forma na qual o HTML providencia e mostra a informação, essencialmente mediante o texto, imagens e links. Falta ver de que forma podemos trocar informações com nosso visitante. Desde então, este novo aspecto é primordial para a grande quantidade de ações que se podem realizar mediante o Web: Comprar um artigo, preencher uma enquete, enviar um comentário ao autor...

Vimos anteriormente que poderíamos, mediante os links, entrar em contato diretamente com um correio eletrônico. Entretanto, esta opção pode ser em alguns casos pouco versátil se o que desejamos é que o navegante nos envie uma informação bem precisa. É através dele, o HTML que propõem outra solução muito mais ampla: Os formulários.

Os formulários são estas famosas caixas de texto e botões que podemos encontrar em muitas páginas web. São muito utilizados para realizar buscar ou também para introduzir dados pessoais, por exemplo, em sites de comércio eletrônico. Os dados que o usuário introduz nestes campos são enviados ao correio eletrônico do administrador do formulário ou também, em um programa que se encarrega de processá-lo automaticamente.

Usando HTML podemos unicamente enviar o formulário a um correio eletrônico. Se quisermos processar o formulário mediante um programa, a coisa pode ser um pouco mais complexa, já que teremos que empregar outras linguagens mais sofisticadas. Neste caso, a solução mais simples é utilizar os programas pré-desenhados que nos propõem um grande número de servidores de hospedagem e que nos permitem armazenar e processar os dados em forma de arquivo ou outros formatos. Se sua página es ta hospedada em um servidor que não lhe propõem estes tipos de vantagens, você sempre poderá recorrer a servidores de terceiros que oferecem este ou outros tipos de serviços gratuitos para webs. É claro que também existe outra alternativa que é a de aprender linguagens como ASP ou PHP que nos permitirá, entre outras coisas, o tratamento de formulários.

Os formulários são definidos por meio das etiquetas `<form>` e `</form>`. Entre estas duas etiquetas colocaremos todos os campos e botões que compõem o formulário. Dentro desta etiqueta `<form>` devemos especificar alguns atributos.:

Action

Define o tipo de ação a realizar com o formulário. Como já dissemos, existem duas possibilidades:

- O formulário é enviado a um endereço de correio eletrônico
- O formulário é enviado a um programa ou script que processa seu conteúdo.

No primeiro caso, o conteúdo do formulário é enviado ao endereço de correio eletrônico especificada por meio de uma sintaxe deste tipo:

```
<form action="mailto:endereço@correio.com"...>
```

Se o que queremos é que o formulário seja processado por um programa, temos de especificar o endereço do arquivo que contem o tal programa. A etiqueta ficaria neste caso da seguinte forma:

```
<form action="endereço do arquivo"...>
```

A forma na qual se expressa a localização do arquivo que contém o programa é a mesma que a vista para os links.

Method

Este atributo se encarrega de especificar a forma na qual o formulário é enviado. Os dois valores possíveis que este atributo pode tomar são post e get. A efeitos práticos e salvo se lhe disserem o contrário, daremos sempre o valor post.

Enctype

Utiliza-se para indicar a forma na qual viajará a informação que for mandada pelo formulário. No caso mais corrente, enviar o formulário por correio eletrônico, o valor deste atributo deve ser "text/plain". Assim, conseguimos que o conteúdo do formulário seja enviado como texto plano dentro do e-mail.

Se quisermos que o formulário se processe automaticamente por um programa, geralmente não

utilizaremos este atributo, de forma que tome seu valor padrão, ou seja, não incluiremos enctype dentro da etiqueta <form>.

Exemplo de etiqueta <form> completa

Assim, para o caso mais habitual - o envio do formulário por correio - a etiqueta de criação do formulário terá o seguinte aspecto:

```
<form action="mailto:endereco@correio.com (ou o nome do arquivo de processo)" method="post"
enctype="text/plain">
```

Entre esta etiqueta e seu fechamento, colocaremos o resto de etiquetas que darão forma ao nosso formulário, as quais serão vistas nos próximos capítulos.

Elementos de formulários. Campos de texto.

O HTML nos propõem uma grande diversidade de alternativas na hora de criar nossos formulários. Estas vão desde a clássica caixa de texto até a lista de opções passando pelas caixas de validação.

Veremos em que consiste cada uma destas modalidades e como podemos implementá-las em nosso formulário.

Texto curto

As caixas de texto são colocadas por meio da etiqueta <input>. Dentro desta etiqueta temos de especificar o valor de dois atributos: **type** e **name**.

A etiqueta é da seguinte forma:

```
<input type="text" name="nome">
```

Deste modo expressamos nosso desejo de criar uma caixa de texto cujo conteúdo será chamado nome (por exemplo). O aspecto deste tipo de caixas é conhecido, como pode ser visto aqui:

O nome do elemento do formulário é de grande importância para poder identificá-lo em nosso programa de processamento ou no e-mail recebido. Por outro lado, é importante indicar o atributo type, já que, como veremos, existem outras modalidades de formulário que usam esta mesma etiqueta.

O emprego destas caixas está fundamentalmente destinado à tomada de dados breves: palavras ou conjuntos de palavras de longitude relativamente curta. Veremos mais adiante que existe outra forma de tomar textos mais longos a partir de outra etiqueta.

Além destes dois atributos, essenciais para o correto funcionamento de nossa etiqueta, existem outra série de atributos que podem ser de utilidade, mas que não são imprescindíveis.

size

Define o tamanho da caixa em número de caracteres. Se ao escrever o usuário chega ao final da caixa, o texto irá desfilando à medida que se escreve fazendo desaparecer a parte de texto que fica à esquerda.

maxlength

Indica o tamanho máximo do texto que pode ser tomado pelo formulário. É importante não confundir-lo com o atributo size. Enquanto o primeiro define o tamanho aparente da caixa de texto, maxlength indica o tamanho máximo real do texto que pode ser escrito. Podemos ter uma caixa de texto com um tamanho aparente (size) que é menor do que o tamanho máximo (maxlength). O que ocorrerá neste caso é que, ao escrever, o texto irá desfilando dentro da caixa até que cheguemos ao seu tamanho máximo definido por maxlength, momento no qual será impossível continuar escrevendo.

value

Em alguns casos pode ser interessante atribuir um valor definido ao campo em questão. Isto pode ajudar ao usuário a preencher mais rapidamente o formulário ou a dar alguma idéia sobre a natureza de dados que se requerem. Este valor inicial do campo pode ser expresso mediante o atributo value. Vejamos seu efeito com um exemplo simples:

```
<input type="text" name="nome" value="Perico Palotes">
```

Gera um campo deste tipo:

Josefa Palotes

Nota: estamos obrigados a utilizar a etiqueta <form>

Para que fique mais claro este conteúdo, ressaltamos: Quando queremos utilizar em qualquer situação elementos de formulário devemos escrevê-los sempre entre as etiquetas <form> y </form>. Caso contrário, os elementos serão vistos perfeitamente no Explorer, mas não no Netscape.

Com outras palavras, no Netscape não se visualizam os elementos de formulário a não ser que estejam colocados entre as correspondentes etiquetas de início e fim de formulário.

É por isso que para mostrar um campo de texto não adianta colocar a etiqueta <input>, e sim, coloca-la dentro de um formulário. Assim:

```
<form>
```

```
<input type="text" name="nome" value="Josefa Palotes">
```

```
</form>
```

Veremos posteriormente que este atributo pode ser relevante em determinadas situações.

Texto oculto

Podemos esconder o texto escrito por meio de asteriscos de forma a fornecer uma certa confiabilidade.

Este tipos de campos são análogos aos de texto com somente uma diferença: deslocando o atributo `type="text"` por `type="password"`:

```
<input type="password" name="nome">
```

Neste caso, pode ser comprovado que ao escrever dentro do campo no lugar de texto serão vistos asteriscos.

Estes campos são ideais para a introdução de dados confidenciais, principalmente códigos de acesso.

Isto pode ser visto em funcionamento a seguir:

Texto longo

Se desejarmos colocar à disposição do usuário um campo de texto onde possa escrever comodamente sobre um espaço composto de várias linhas, temos de convocar uma nova etiqueta: `<textarea>` e seu fechamento correspondente.

Estes tipos de campos são práticos quando o conteúdo a enviar não é um nome, telefone ou qualquer outro dado breve, e sim, um comentário, opinião, etc.

Dentro da etiqueta `textarea` deveremos indicar, como para o caso visto anteriormente, o atributo `name` para associar o conteúdo a um nome que será semelhante a uma variável nos programas de processo.

Além disso, podemos definir as dimensões do campo a partir dos seguintes atributos:

rows

Define o número de linhas do campo de texto.

cols

Define o número de colunas do campo de texto.

A etiqueta fica portanto, desta forma:

```
<textarea name="comentário" rows="10" cols="40"></textarea>
```

O resultado é o seguinte:

Mesmo assim, é possível definir o conteúdo do campo. Para isso, não usaremos o atributo `value` e sim, que escreveremos dentro da etiqueta o conteúdo que lhe desejamos atribuir. Vejamos:

```
<textarea name="comentário" rows="10" cols="40">Escreva seu comentário....</textarea>
```

Outros elementos de formulários

Efetivamente, os textos são uma maneira muito prática de fazer chegar a informação do navegante. Porém, em muitos casos, os textos são dificilmente adaptáveis a programas que possam processá-los devidamente ou também pode ser que seu conteúdo não se ajuste ao tipo de informação que requeremos. É por isso que, em determinados casos, pode ser mais efetivo propor uma escolha ao navegante a partir da exposição de uma série de opções.

Este é o caso de, por exemplo, oferecer uma lista de países, o tipo de cartão de crédito para um pagamento, etc.

Estes tipos de opções podem ser expressadas de diferentes formas. Vejamos a seguir quais são:

Listas de opções

As listas de opções são esse tipo de menus desdobráveis que nos permite escolher uma (ou várias) das múltiplas opções que nos propõem. Para construí-las utilizaremos uma etiqueta com seu respectivo fechamento: `<select>`

Como para os casos já vistos, dentro desta etiqueta definiremos seu nome por meio do atributo `name`.

Cada opção será incluída em uma linha precedida da etiqueta `<option>`.

Podemos ver, a partir destas diretrizes, a forma mais típica e simples desta etiqueta:

```
<select name="estação">
<option>Primavera</option>
<option>Verão</option>
<option>Outono</option>
<option>Inverno</option>
</select>
```

O resultado é:

Primavera

Esta estrutura pode ser vista modificada principalmente a partir de outros dois atributos:

size

Indica o número de valores mostrados da lista. O resto pode ser visto por meio da barra lateral de deslocamento.

multiple

Permite a seleção de mais vários elementos da lista. A escolha de mais de um elemento se faz como com o explorador de Windows, a partir das teclas `ctrl` ou `shift`. Este atributo se expressa sem valor algum, ou seja, não se utiliza com o igual: simplesmente se coloca para conseguir o efeito, ou não se coloca se quisermos uma lista desdobrável comum.

Conselho: Se for possível, não utilize multiple

Não recomendamos especialmente a prática desta opção já que o manejo das teclas ctrl ou shift para escolher várias opções pode ser desconhecido para o navegador. Evidentemente, sempre cabe a possibilidade de explicar como funciona a pesar de ser uma complicação a mais para o visitante.

Vejamos qual é o efeito produzido por estes dois atributos mudando a linha:

```
<select name="estação">
```

por:

```
<select name="estação" size="3" multiple>
```

A lista ficará desta forma:

Primavera

Verão

Outono

A etiqueta **<option>** ainda pode ser precisada por meio de **outros atributos**

selected

Da mesma forma que multiple, este atributo não toma nenhum valor senão que simplesmente indica que a opção que apresenta está escolhida por padrão.

Assim, se mudamos a linha do código anterior:

```
<option>Outono</option>
```

por:

```
<option selected>Outono</option>
```

O resultado será:

Outono

value

Define o valor da opção que será enviada ao programa ou ao correio eletrônico se o usuário escolher essa opção. Este atributo pode ser muito útil se o formulário for enviado a um programa visto que a cada opção se pode associar um número ou letra, o qual torna-se mais facilmente manipulável que uma palavra ou texto. Poderíamos assim escrever linhas do tipo:

```
<option value="1">Primavera</option>
```

Deste modo, se o usuário escolher primavera, o que chegará ao programa (ou ao correio) é uma variável chamada estação que terá com valor 1. No correio eletrônico receberíamos:

estação=1

Botões de radio

Existe outra alternativa para expor uma escolha, neste caso, obrigamos ao internauta a escolher unicamente uma das opções que lhe propõem.

A etiqueta empregada neste caso é **<input>** na qual teremos a atributo type que temos de tomar o valor radio. Vejamos um exemplo:

```
<input type="radio" name="estação" value="1">Primavera
```

```
<br>
```

```
<input type="radio" name="estação" value="2">Verão
```

```
<br>
```

```
<input type="radio" name="estação" value="3">Outono
```

```
<br>
```

```
<input type="radio" name="estação" value="4">Inverno
```

Nota: Temos de observar que a etiqueta **<input type="radio">** somente coloca o campo para clicar na página. Os textos que aparecem ao lado, assim como as quebras de linha, colocamos com o correspondente texto no código da página e com as etiquetas HTML que necessitarmos.

O resultado é o seguinte:

Primavera

Verão

Outono

Inverno

Como podemos ver, a cada uma das opções se atribui uma etiqueta input dentro da qual atribuímos o mesmo nome (name) para todas as opções e um valor (value) distinto. Se o usuário escolher supostamente Outono, receberemos em nosso correio uma linha tal como esta:

estação=3

Cabe assinalar que é possível pré-selecionar por padrão uma das opções. Isto se pode conseguir por meio do atributo **checked**:

```
<input type="radio" name="estação" value="2" checked>Verão
```

Vejamos o efeito:

Primavera

Verão

Outono

Inverno

Caixas de validação

Estes tipos de elementos podem ser ativados ou desativados pelo visitante com um simples clique sobre a caixa em questão. A sintaxe utilizada é muita parecida com as vistas anteriormente:

```
<input type="checkbox" name="paella">Adoro uma feijoada
```

O efeito:

Adoro uma feijoada

A única diferença fundamental é o valor adotado pelo atributo type.

Da mesma forma que para os botões de radio, podemos ativar a caixa por meio do atributo **checked**.

O tipo de informação que chegará ao nosso correio (ou ao programa) será do tipo:

feijoada=on (ou off dependendo se tiver sido ativada ou não)

Botão Submit, Apagar Campos e outros em formulários HTML

Os formulários têm de dar lugar não somente à informação a tomar do usuário como também, a outra série de funções. Concretamente, permite-nos seu envio mediante seu botão. Também pode ser prático poder propor um botão de Apagar Campos ou também, acompanhá-lo de dados ocultos que possam ajudar-nos em seu processamento.

Neste capítulo, para terminar a saga de formulários, tornaremos conhecidos os meios de instalar todas estas funções.

botão Submit (ou de envio)

Para finalizar o processo de preenchimento do formulário e fazê-lo chegar a seu gestor, o navegante tem de validá-lo por meio de um botão previsto para tal efeito. A construção de tal botão não implica nenhuma dificuldade uma vez familiarizados com as etiquetas input já vistas:

```
<input type="submit" value="Enviar">
```

Com este código geramos um botão como este:

Enviar

Como pode ser visto, somente temos de especificar se trata de um botão de envio (type="submit") e temos de definir a mensagem do botão por meio do atributo value.

botão Apagar Campos

Este botão nos permitirá apagar o formulário por completo no caso de que o usuário deseje refazê-lo desde o princípio. Sua estrutura sintática é igual a anterior:

```
<input type="reset" value="Apagar Campos">
```

A diferença do botão de envio, indispensável em qualquer formulário, o botão de Apagar Campos é meramente optativo e não é utilizado freqüentemente. Há de ter cuidado de não colocá-lo muito perto do botão de envio e de distinguir claramente um do outro.

Dados ocultos

Em alguns casos, aparte dos próprios dados enviados pelo usuário, pode ser prático enviar dados definidos por nós mesmos que ajudem ao programa em seu processamento do formulário. Estes tipos de dados, que não se mostram na página, mas que podem ser detectados solicitando o código fonte, não são freqüentemente utilizados por páginas construídas em HTML, são mais usados por páginas que empregam tecnologias de servidor. Não se assustem, pois veremos mais adiante o que isto quer dizer. Queremos apenas dar constância de sua existência e de seu modo de criação. Como por exemplo:

```
<input type="hidden" name="site" value="www.andrecavalcante.com">
```

Esta etiqueta, incluída dentro de nosso formulário, enviará um dado adicional ao correio ou ao programa encarregado da gestão do formulário. Poderíamos a partir deste dado, tornar conhecido ao programa a origem do formulário ou algum tipo de ação a realizar (um re-endereçamento, por exemplo).

Botões normais

Dentro dos formulários também podemos colocar botões normais, clicáveis como qualquer outro botão.

Da mesma forma que ocorre com os campos hidden, estes botões por si só não têm muita utilidade, mas poderemos necessita-los para realizar ações no futuro. Sua sintaxe é a seguinte:

```
<input type="button" value="Texto escrito no botão">
```

Ficaria desta maneira:

O uso mais frequente de um botão é na programação no cliente. Utilizando linguagens como Javascript podemos definir ações a tomar quando um visitante clique o botão de uma página web.

Exemplo de formulário

Com este capítulo finalizamos nosso assunto sobre formulários. Passamos agora a exemplificar todo o aprendizado a partir da criação de um formulário que consulta o grau de satisfação dos usuários de uma linha de ônibus fictícia. O formulário está construído para que envie os dados por correio eletrônico a uma caixa de entrada determinada.

Vemos o formulário nesta página. Vocês tratem de construí-lo para ver se realmente entenderam bem os temas sobre formulários.

Nome

E-mail

@

Cidade

Sexo

Homem

Mulher

Frequência das viagens

Várias vezes por dia

Comentários sobre sua satisfação pessoal

Desejo receber notificação das novidades nas linhas de ônibus.

Enviar formulário

Apagar tudo

A seguir também mostraremos o código fonte deste formulário, que é importante que todos dêem uma olhada, mesmo que seja rapidamente.

```
<form action="mailto: andre@andrecavalcante.com " method="post" enctype="text/plain">
```

```
Nome <input type="text" name="nome" size="30" maxlength="100">
```

```
<br>
```

```
E-mail <input type="text" name="email" size="25" maxlength="100" value="@">
```

```
<br>
```

```
Cidade <input type="text" name="cidade" size="20" maxlength="60">
```

```
<br>
```

```
Sexo
```

```
<br>
```

```
<input type="radio" name="sexo" value="Masculino" checked> Homem
```

```
<br>
```

```
<input type="radio" name="sexo" value="Feminino"> Mulher
```

```
<br>
```

```
<br>
```

Frequência das viagens

```
<br>
```

```
<select name="utilização">
```

```
<option value="1">Várias vezes por dia
```

```
<option value="2">Uma vez por dia
```

```
<option value="3">Várias vezes por semana
```

```
<option value="4">várias vezes por mês
```

```
</select>
```

```
<br>
```

```
<br>
```

Comentários sobre sua satisfação pessoal

```
<br>
```

```
<textarea cols="30" rows="7" name="comentários"></textarea>
```

```
<br>
```

```
<br>
```

```
<input type="checkbox" name="receber_info" checked>
```

Desejo receber notificação das novidades nas linhas de ônibus.

```
<br>
```

```
<br>
```

```
<input type="submit" value="Enviar formulário">
```

```
<br>
```

```
<br>
```

```
<input type="Reset" value="Apagar tudo">
```

```
</form>
```

Para acabar, vamos ver o que receberiam por correio eletrônico na empresa de ônibus quando um usuário qualquer preenchesse este formulário e clicasse sobre o botão de envio.

nome=Frederico Silvestre

e-mail=fede@terramix.com

cidade=Rio de Janeiro

sexo=Masculino

utilização=2

comentários=Acho que não é uma boa linha. Colocar mais ônibus.

receber_info=on

Mapas de imagens com HTML

Nos capítulos anteriores nos aprofundamos no elemento básico de navegação do web: O link hipertexto. Vimos que estes links são palavras, textos ou imagens que, ao clicar sobre eles, nos enviam a outras páginas ou zonas.

Os mapas de imagem são uma nova exposição de navegação que incorpora uma série de links dentro de uma mesma imagem. Estes links são definidos por figuras geométricas e funcionam exatamente da mesma forma que os outros links. Pode-se ver o funcionamento de um neste link.

À princípio, estes mapas não eram diretamente reconhecidos pelos navegadores e recorriam à tecnologias de lado do servidor para serem visualizados. Hoje em dia podem ser implementados por meio de código HTML tal como veremos neste capítulo.

Podemos utilizar estes mapas, por exemplo, em portais onde tornamos conhecida cada uma das sessões do site por meio de uma imagem. Também pode ser muito prático, em mapas geográficos onde cada cidade, estado ou ponto qualquer representa um link a uma página.

Em qualquer caso, o uso destes mapas tem de ser sistematicamente acompanhado de um texto explicativo que dê a entender ao usuário sobre os distintos pontos da imagem. Frases como "Dê um clique sobre tal ícone para acesar a tal informação". São muito indicativas na hora de fazer intuitiva a navegação pelos mapas de imagens. Por outro lado, não é demais introduzir essa mesma explicação no atributo alt da imagem.

Sendo assim, um mapa de imagem está composto por duas partes:

- A imagem propriamente dita que estará situada como de costume dentro da etiqueta <body> de nosso documento HTML.
- Um código, situado no interior da etiqueta <map>, que delimitará por meio de linhas geométricas imaginárias cada uma das áreas dos links apresentados na imagem.

As linhas geométricas que delimitam os links, ou seja, as áreas dos links, devem ser definidas por meio de coordenadas. Cada imagem é definida por umas dimensões de largura (X) e altura (Y) e cada ponto da imagem pode ser definido portanto, dizendo a que altura (x) e largura (y) nos encontramos. Deste modo, a esquina superior esquerda corresponde à posição 0,0 e a esquina inferior direita corresponde às coordenadas X,Y. Se desejamos saber quais coordenadas correspondem a um ponto concreto de nossa imagem, o melhor é utilizar um programa de desenho gráfico como Photoshop ou Paint Shop Pro. A melhor forma de explicar o funcionamento deste tipo de mapa é a partir de um exemplo prático.

Suponhamos que temos uma imagem com um mapa como esta:

Clique nos círculos para acessar às seções!

Dentro dela queremos introduzir um link a cada um dos elementos que a compõem. Para isso, definiremos nossos links como zonas circulares de tamanho pequeno que serão distribuídas ao comprimento e largura da imagem.

Vejamos a seguir o código que utilizaremos:

```
<table border=0 width=450><tr><td align="center">
<map name="mapa1">
<area alt="Clique para ver a página de meus amigos" shape="CIRCLE" coords="44,36,29" href="#">
<area alt="Clique para ver minha noiva" shape="CIRCLE" coords="140,35,31" href="#">
<area alt="Clique para conhecer minha família" shape="circle" coords="239,37,30" href="#">
<area alt="Clique para conhecer meu trabalho" shape="CIRCLE" coords="336,36,31" href="#">
</map>

<br>
Clique nos círculos para acessar às seções!
</td></tr></table>
```

Nota: Os href das áreas vão a

Este é um exemplo parcial de utilização dos mapas, faltaria colocar os href com valores reais e não com a #. Cada um dos links das áreas -atributo href da etiqueta <area>- deveriam levar a uma página web. O exemplo ficaria completo se criássemos todas as páginas onde linkar as áreas e colocássemos os href dirigidos para tais páginas. Como não fizemos as páginas "destino" colocamos links que não levam a lugar nenhum, que, como se pode ver, indica-se com o caráter "#".

É possível observar, tal como explicamos antes, que nosso mapa consta de duas partes principais: a imagem e a etiqueta <map> que define as áreas de cada link.

Cada área se indica com uma etiqueta <area>, que tem os seguinte atributos:

alt

Para indicar um texto que será mostrado quando situarmos o mouse na área.

shape

Indica o tipo de área.

coords

As coordenadas que definem a área. Será um grupo de valores numéricos distintos dependendo do tipo de área (shape) que estivermos definindo.

href

Para indicar o destino do link correspondente à área.

Neste caso utilizamos áreas circulares (shape="CIRCLE"), que se definem indicando o centro do círculo -uma coordenada (X,Y) e o radio, que é um número inteiro que se corresponde com o número de pixels desde o centro até a borda do círculo.

Tipos de áreas: shape distintas.

Existem três tipos de áreas distintas, suficientes para fazer quase qualquer tipo de figura.

No desenho que acompanha estas linhas pode ser visto uma representação das áreas, que detalhamos a seguir.

shape="RECT"

Cria uma área retangular. Para defini-la utilizam-se as coordenadas dos pontos da esquina superior esquerda e da esquina inferior direita. Tal como estão nomeadas tais coordenadas em nosso desenho, a área teria a seguinte etiqueta:

```
<area shape="RECT" coords="X1,Y1,X2,Y2" href="#">
```

shape="CIRCLE"

Cria uma área circular, que se indica com a coordenada do centro do círculo e o radio. De acordo com nosso desenho, a etiqueta de uma área circular teria esta forma:

```
<area shape="CIRCLE" coords="X1,Y1,R" href="#">
```

shape="POLY"

Este tipo de área, poligonal, é a mais complexa de todas. Um polígono fica definido indicando todos seus pontos, mas atenção, pois temos que indicá-los em ordem, seguindo o caminho marcado pelo perímetro do polígono. Segundo nosso desenho e os nomes que demos aos pontos do polígono, a etiqueta <area> ficaria desta forma:

```
<area shape="POLY" coords=" X1,Y1, X2,Y2, X3,Y3, X4,Y4" href="#">
```

Frames em HTML

Uma das mais modernas características de HTML são os frames, que foram acrescentados tanto em Netscape Navigator como em Internet Explorer, a partir de suas versões 2.0. Os frames -que significam em português janelas- são uma forma de dividir a página em diferentes espaços independentes uns dos outros, de forma que em cada espaço se coloca uma página diferente que se codifica em um arquivo HTML diferente.

À princípio se criaram como etiquetas proprietárias do navegador Netscape e rapidamente a potência do recurso fez com que o uso de frames se estendesse por toda a web. Pouco tempo demoraria para Internet Explorer incluí-los, pois não podia deixar escapar uma novidade tão popular de seu competidor. Finalmente, como resposta à popularidade entre os desenvolvedores dos frames, o padrão HTML 4.0 incluiu estas etiquetas dentro das permitidas.

Os frames, como dizíamos, nos permitem dividir a janela do navegador em diferentes áreas. Cada uma destas áreas são independentes e devem ser codificadas com arquivos HTML também independentes. Como resultado, cada frame ou janela contém as propriedades específicas que indicamos no código HTML apresentado nesse espaço. Sendo assim, e dado que cada janela é independente, terão suas próprias barras de deslocamento, horizontais e verticais, separadamente.

Existem muitas páginas na web que contêm frames e certamente todos já tiveram a ocasião de conhecer algumas. Costuma-se utilizar para colocar em uma parte da janela uma barra de navegação, que geralmente encontra-se fixa e permite o acesso a qualquer zona da página web. Uma das principais vantagens da programação com frames vem derivada da independência dos distintos frames, pois podemos navegar pelos conteúdos de nosso site web com a barra de navegação sempre visível, e sem que se tenha que recarregar em cada uma das páginas que vamos visitando.

Um exemplo das áreas que se podem construir em uma construção de frames pode ser visto nas imagens a seguir.

Frames – Explicação básica

As páginas web que estão feitas com frames se compõem de uma declaração das janelas e tantas páginas em formato HTML corrente como distintas divisões tivermos definido. A declaração ou definição de frames é a única página que realmente devemos aprender, visto que as páginas que serão visualizadas em cada uma das janelas são arquivos HTML dos que viemos aprendendo anteriormente neste manual.

Tal definição está composta por etiquetas <FRAMESET> e <FRAME>, com as quais indicamos a disposição de todos os quadros. A etiqueta <FRAMESET> indica as divisões da janela do navegador e a etiqueta <FRAME> indica cada um dos quadros onde colocaremos uma página independente.

As partições que se podem fazer com um <FRAMESET> são em filas ou colunas. Por exemplo, poderíamos indicar que desejamos fazer uma divisão da página em duas filas, ou duas colunas, três filas, etc. Para indicar tanto a forma de dividir a janela -em filas ou colunas- como o número de partições que pretendemos fazer, temos de utilizar o atributo COLS ou ROWS. O primeiro serve para indicar uma partição em colunas e o segundo para uma partição em filas.

Nota: É importante indicar que não se pode fazer uma partição em filas e colunas de uma vez só, e sim, que devemos escolher em dividir a janela em uma das duas disposições. Mais adiante, indicaremos como dividir a janela tanto em filas como em colunas, que é feito com o acréscimo de frames.

No atributo COLS ou ROWS -somente podemos escolher um dos dois- colocamos entre aspas o número de divisões que desejamos realizar, indicando de antemão o tamanho que vai atribuir a cada uma. Um valor típico destes atributos seria o seguinte:

cols="20%,80%"

Indica que devem ser colocadas duas colunas, a da esquerda teria uns 20% do espaço total da janela e a da direita uns 80%.

rows="15%,60%,25%"

Assim, indicamos que desejamos três filas, a de cima com um 15% do espaço total, a do meio com um espaço correspondente ao 60% do total e a de abaixo com um 25%. Ao total somam o 100% do espaço da janela.

Além da porcentagem para indicar o espaço de cada um dos campos, também podemos indicá-lo em pixels. Desta maneira:

cols="200,600"

Para indicar que a coluna da esquerda deve ter 200 pixels de largura e a da direita 600. Isto está bem se nossa janela tem 800 pixels de largura, mas isto não tem porque ser assim em todos os monitores dos usuários, por isso é importante que este modo de expressar as janelas se indique da seguinte maneira.

cols="200,*"

Assim, indicamos que a primeira coluna tem de medir 200 pixels e que o resto do espaço disponível -que será maior ou menor dependendo da definição da tela do usuário- se atribuirá a segunda coluna.

Na prática podemos misturar todos estes métodos para definir as janelas da maneira de desejarmos, com porcentagem, com pixels ou com o asterístico (*). Não importa como se definem, a única recomendação é que um dos valores que indiquemos seja um asterisco, para que a área correspondente a tal asterisco seja mais ou menos grande dependendo do espaço que tenha a janela de nosso navegador. Outros métodos de definir filas e colunas, atendendo a este conselho, seriam os seguintes:

rows="100*,12%"

Definimos três filas, a primeira com 100 pixels de largura, a segunda com o espaço que sobre das outras duas, e a terceira com um 12% do espaço total.

cols="10%,50%,120,*"

Estamos indicando quatro colunas. A primeira de 10% do espaço da janela, a segunda com a metade justa da janela, a terceira com um espaço de 120 pixels e a última com a quantidade de espaço que sobre ao atribuir espaço às demais partições.

Uma vez indicado o número de filas ou colunas e o espaço reservado a cada uma com a etiqueta <FRAMESET>, devemos especificar com a etiqueta <FRAME> a procedência de cada um dos frames que dividimos a janela.

Para isso, dispomos do atributo SRC, que tem de ser definido para cada uma das filas ou colunas. Desta maneira.

<FRAME src="janela1.html">

Assim fica indicado que o frame que estamos definindo deve mostrar a página janela1.html em seu interior.

Frames – Criação de uma estrutura simples

Para ilustrar tudo o que viemos explicando podemos ver o exemplo sobre como se criaria a definição de frames da imagem que podemos ver a seguir.

```
<html>
<head>
<title>Definição de Frames</title>
</head>
<frameset rows="15%,*,75%">
<frame src="pagina1.html">
```

```
<frame src="pagina2.html">
<frame src="pagina3.html">
</frameset>
</html>
```

Pode-se ver esta partição de frames em uma página a parte.

Além disso, temos algumas considerações a fazer para terminar de compreender este exemplo:

- O título da definição de frames é o que herda toda a página web, por isso, não é boa idéia titular como "definição de frames" por exemplo, já que então toda nossa página se titularia assim e certamente não seria muito descritivo. Se estivéssemos fazendo uma página para o açougue Gonçalves seria melhor titular a definição de frames algo como "Açougue Gonçalves, as melhores carnes no Rio de Janeiro".
- A página que define os frames não tem body. O HTML pode totalizar um erro se o incluirmos.
- As páginas "pagina1.html", "pagina2.html" e "pagina3.html" devem ser escritas em arquivos independentes com o nome indicado. Neste exemplo, Tais páginas deveriam se encontrar no mesmo diretório que a declaração de frames. Se especificarmos uma rota para acessar ao arquivo podemos colocá-lo no diretório que desejarmos.
- As cores de cada um dos frames colocamos com o atributo bgcolor colocado na etiqueta <BODY> de cada uma das páginas que se mostram nas janelas.

Frames – Uma página em cada janela

As páginas que mostraremos em cada janela são documentos HTML iguais aos que viemos criando anteriormente. Podemos colocar qualquer elemento HTML dos estudados neste manual, como etiquetas de parágrafo, imagens, cores de fundo, etc.

Cada documento, como já indicamos, se escreve separado em seu próprio arquivo HTML. Para o exemplo do capítulo anterior podemos definir os arquivos HTML da seguinte maneira.

pagina1.html

É a página que contém o título da web. Simplesmente se trata de uma etiqueta <H1> de título. A página tem seu próprio título, com a etiqueta <TITLE>, que não poderá ser visualizada nenhum lugar, a não ser que se mostre esta página sem os frames, já que as páginas dentro das janelas herdam o título da definição dos frames.

```
<html>
<head>
<title>Título Açougue Gonçalves</title>
</head>
<body bgcolor="#DECC09">
<h1 align=center>Açougue Gonçalves</h1>
</body>
</html>
```

pagina2.html

É a página que se apresentará na área principal da definição de frames, ou seja, a página que tem mais espaço para ser visualizada e onde colocaremos os conteúdos da web. Neste caso, mostra uma mensagem de bem-vindo à web, que fará as vezes de portal.

```
<html>
<head>
<title>Portal do Açougue Gonçalves</title>
</head>
<body bgcolor="#CF391C" text="#ffffff">
<h1 align="center">Bem-vindo a nossa web</h1>
<br>
<br>
```

O açougue Gonçalves, com mais de 100 anos de experiência, é a melhor fonte de carnes de boi e de porco da região.

```
<br>
```

```
<br>
```

Tanto no inverno como no verão você pode encontrar nossas ofertas de temporada de primeira qualidade.

```
</body>
```

```
</html>
```

pagina3.html

Nesta página se mostrará a barra de navegação pelos conteúdos do site. Contém links que deveriam atualizar o conteúdo da área principal da declaração de frames, para mostrar os distintos conteúdos

do sitio, por exemplo, o portal, os produtos, a página de contato, etc.

```
<html>
<head>
<title>Barra de navegação de açougue Gonçalves</title>
</head>
<body bgcolor="#AC760E" link="ffffcc" vlink="ffffcc">
<div align="center">
<b>
<a href="pagina2.html">Portal</a> |
<a href="produtos.html">Produtos</a> |
<a href="contato.html">Contato</a>
</b>
</div>
</body>
</html>
```

Podemos ver como fica a página de frames com estes conteúdos, que simulam a página de um açougue.

Frames – Orientar os links

A única particularidade destacável no exemplo do capítulo anterior, no manejo de frames em geral, trata-se de que **cada um dos links que colocamos nas páginas atualizam o frame onde está colocado este link**. Por exemplo, se temos link na parte inferior da janela, no espaço correspondente a terceira janela, atualizarão os conteúdos do terceiro frame, que é onde estão situados os links.

Este efeito que comentamos pode-se observar no exemplo da página do açougue, tal como ficaria ao incluir os códigos das distintas páginas.

O lógico é que ao clicar sobre um link da barra de navegação atualizamos o frame principal, que é onde havíamos planejado colocar os conteúdos, no lugar do frame onde colocamos a barra de navegação, que deveria se manter fixa. Para conseguir este efeito devemos fazer duas coisas:

1. Dar um nome ao frame que desejamos atualizar

Tal nome se indica na etiqueta <FRAME> da definição de frames. Para isso utilizamos o atributo name, igualado ao nome que queremos dar a tal janela.

2. Orientar os links para esse frame

Para isso devemos colocar no atributo target dos links -etiqueta <A>- o nome do frame que desejamos atualizar ao clicar o link.

Depois de dar um nome ao frame principal, nossa declaração de frames ficaria da seguinte maneira.

```
<frameset rows="15%,*,75">
<frame src="pagina1.html">
<frame src="pagina2.html" name="principal">
<frame src="pagina3.html">
</frameset>
```

Ademais, deveríamos colocar o atributo target aos links, tal como se segue.

```
<a href="pagina2.html" target="principal">Portal</a> |
<a href="produtos.html" target="principal">Produtos</a> |
<a href="contato.html" target="principal">Contato</a>
```

Uma vez realizados estas duas mudanças podemos ver como os links da barra de navegação sim que atualizam a página que devem.

Valores para o atributo target

Como vimos, com o atributo target da etiqueta <A> podemos indicar o nome do frame que desejamos que atualize esse link. Entretanto, este não é o único valor que podemos aplicar ao atributo. Temos alguns valores adicionais que podemos atribuir a qualquer link em geral.

_blank

Para fazer com que esse link se abra em uma janela a parte. Nossos exemplos neste manual costumam abrir em uma janela a parte, colocando este valor no target dos links que levam aos exemplos.

_self

Atualiza o frame onde está situado o link. É o valor por padrão.

_parent

O link se atualiza sobre seu pai ou sobre a janela que estamos trabalhando, se não tiver um pai.

_top

A página carrega a tela completa, ou seja, eliminando todos os frames que pudesse haver. Este atributo é muito importante porque se colocamos em nossa página com frames um link a uma página externa, se

abriria em um dos frames e se manteriam visíveis outros frames da página, fazendo um efeito que costuma ser pouco agradável, porque parece que estão evitando que nos escapemos.

A sintaxe de um desses valores de atributos colocados em um link seria a seguinte:

```
<A href="http://www.andrecavalcante.com" target="_top">Acessar a andrecavalcante.com</A>
```

Frames – Aninhar frames

Para criar estruturas de janelas nas quais se misturam as filas e as colunas devemos aninhar etiquetas <FRAMESET>. Começando pela partição de frames mais geral, devemos colocar dentro as partições de frames menores. A maneira de indicar isto se pode ver facilmente com um exemplo.

Os passos para definir o aninhamento podem ser encontrados à direita.

Os distintos frames vêm numerados com a ordem na qual são escritas no código.

Na imagem pode ser visto o resultado final acompanhada da representação sobre a maneira de defini-los.

Em primeiro definimos uma estrutura de frames em duas colunas e dentro da primeira coluna colocamos outra partição de frames em duas filas. O código necessário é o seguinte.

```
<frameset cols="200,*">
<frameset rows="170,*">
<frame src="pagina1.html">
<frame src="pagina2.html">
</frameset>
<frame src="pagina3.html">
</frameset>
```

Podemos ver o exemplo em uma página a parte.

Nota: colocamos uma margem em cada uma das linhas desta definição de frames para conseguir um código mais compreensível visualmente. Estas margens não são em absoluto necessários, simplesmente nos servem para ver em que nível de aninhamento nos encontramos.

O exemplo anterior pode se complicar um pouco mais se incluirmos mais partições. Vamos ver algo um pouco mais complicado para praticar mais com os aninhamentos de frames.

Os passos para definir o aninhamento podem ser encontrados à direita.

Os distintos frames vêm numerados com a ordem na qual se escrevem no código.

Na imagem se observa que o primeiro frameset a definir se compõe de duas filas. Posteriormente, dentro da segunda fila do primeiro frameset, temos outra partição em duas colunas, dentro das que colocamos um terceiro nível de frameset com uma definição em filas nos dois casos. O código pode ser visto a seguir.

```
<frameset rows="60,*">
<frame src="pagina1.html">
<frameset cols="200,*">
<frameset rows="*,150">
<frame src="pagina2.html">
<frame src="pagina3.html">
</frameset>
<frameset rows="*,60">
<frame src="pagina4.html">
<frame src="pagina5.html">
</frameset>
</frameset>
</frameset>
</frameset>
```

Podemos ver o exemplo em uma página a parte.

Até aqui vimos a parte mais básica da criação de frames. Nos próximos capítulos poderemos aprender a configurar as janelas para variar sua aparência e, entre outras coisas, eliminar as barras que separam cada um dos distintos frames.

Frames – Atributos avançados

A parte da criação das janelas propriamente dita, existem muitos atributos com os que configurar sua aparência. Para isso, tanto a etiqueta <frameset> como <frame> admitem diversos atributos que permitem especificar a forma de elementos como as bordas dos frames, a margem, a existência ou não de barras de deslocamento, etc.

Atributos para a etiqueta <frameset>

Já conhecemos o atributo cols e rows, que servem para indicar se a distribuição em janelas se fará horizontalmente ou verticalmente. Somente se pode utilizar um deles e se iguala às dimensões de cada uma das divisões, separadas por vírgulas.

border="número de pixels"

Permite especificar de maneira global para todo o frameset o número de pixels que há de ter a borde dos frames.

bordercolor="#rrgbbb"

Com este atributo podemos modificar a cor da borda dos frames, também de maneira global a todo o frameset.

frameborder="yes|no|0"

Serve para mostrar ou não a borda do frame. Seus possíveis valores são "yes" (para que se vejam as bordas) e "não" ou "0" (para que não se vejam). Na prática elimina a borda, mas permanece uma linha de separação dos frames.

framespacing="número de pixels"

Para determinar a largura da linha de separação dos frames. Pode -se utilizar em Internet Explorer e junto com o atributo frameborder="0" serve para eliminar a borda das janelas.

Atributos para a etiqueta <frame>

Para esta etiqueta, assinalamos nos capítulos anteriores os atributos src, que servem para indicar o arquivo que contém a janela e name, para dar um nome à janela e logo direcionar os links para ele. Vejamos agora outros atributos disponíveis.

marginwidth="número de pixels"

Define o número de pixels que tem a margem do frame onde se indica. Esta margem se aplica à página que pretendemos ver nessa janela, de modo que se colocamos 0, os conteúdos da página nessa janela estarão completamente grudados na borda da margem e se indicamos um valor de 10, os conteúdos da página estariam separados da borda 10 pixels.

marginheight="número de pixels"

O mesmo que o atributo anterior, mas para a margem vertical.

scrolling="yes|no|auto"

Serve para indicar se queremos que haja barras de deslocamento nas diferentes janelas. Se indicamos "yes" sempre sairão as barras, se indicamos "no" não sairão nunca e se colocamos "auto" sairão somente se forem necessárias. Auto é o valor por padrão.

Conselho: Há que ter cuidado se eliminamos as bordas dos frames, visto que a a página web pode ter dimensões distintas dependendo da definição de tela do visitante. Se o espaço da janela se vê reduzido, poderia se ver reduzido o espaço para o frame e pode acontecer de não caber os elementos que antes sim cabiam e se eliminamos as barras de deslocamento pode ser que o visitante não possa ver todo o conteúdo da janela.

Este mesmo conselho pode ser aplicado ao redimensionamento de frames, que veremos no seguinte atributo. Se fizermos com que as janelas não sejam redimensionáveis provavelmente teríamos uma declaração de frames demasiado rígida, que pode ser mal vista em algum tipo de tela.

noresize

Este atributo não tem valores, simplesmente se coloca ou não se coloca. No caso de que esteja presente indica que o frame não se pode redimensionar. Como podemos ver, ao colocar o mouse sobre a borda das janelas sai um cursor que nos assinala que podemos mover tal borda e redimensionar assim os frames. Por padrão, se não colocamos nada, as janelas podem ser redimensionadas.

frameborder="yes|no|0"

Este atributo permite controlar a aparição das bordas dos frames. Com este atributo igualado a "0" ou "não" as bordeas se eliminam. Entretanto, ficam as feias margens na borda. Pelo que podemos comprovar funciona melhor no Netscape do que no Internet Explorer. De qualquer forma, temos um pouco mais adiante, uma nota para explicar os frames sem bordas.

Nota: os atributos de frames não funcionam sempre bem em todos os navegadores. É recomendável fazermos um teste sobre o que estamos desenhando em vários navegadores para comprovar que nosos frames se vêem bem em todas as plataformas.

bordercolor="#rrgbbb"

Permite especificar a cor da borda da janela.

Vantagens e inconvenientes do uso de frames

O desenho com frames é um assunto bastante controvertido, já que distintos desenhadores podem ter outras opiniões.

Referência: Se você deseja saber o que são os frames e como criá-los consulte os capítulos de Frames de nosso manual de HTML.

No meu caso, penso que é preferível não utilizá-los, apesar de que isso depende do tipo de web site que esteja construindo, pois em alguns casos sim que seria muito adequado seu uso.

Colocarei algumas vantagens e alguns inconvenientes deo uso de janelas (frames). Isto é ao meu ponto

de vista, outros podem ter outras opiniões.

Vantagens de usar frames

- A navegação da página será mais rápida. Apesar de que o primeiro carregamento da página seria igual, em sucessivas impressões páginas já teremos algumas janelas salvas, que não teriam que voltar a descarregar.
- Criar páginas do site seria mais rápido. Como não temos que incluir partes de código como a barra de navegação, título, etc. criar novas páginas seria um processo muito mais rápido.
- Partes da página (como a barra de navegação) se mantêm fixas e isso pode ser bom, para que o usuário não as perca nunca de vista.
- Estas mesmas partes visíveis constantemente, se contém links, podem servir muito bem para melhorar a navegação pelo site.
- Mantém uma navegabilidade do site onde se navega, pois os elementos fixos conservam a imagem sempre visível.

Inconvenientes de usar frames

- Tiram espaço da tela. O espaço ocupado pelos frames fixos se perde na hora de fazer páginas novas, porque já está utilizado. Em definições de tela pequena ou em dispositivos como Palms, este problema se torna mais patente.
- Forçam ao visitante a entrar pela declaração de frames. Se não o fazem assim, somente se veria uma página interior sem os requadros. Estes requadros poderiam ser insuficientes para uma boa navegação pelos conteúdos e poderiam não conservar uma boa imagem corporativa.
- A promoção da página seria, à princípio, mais limitada. Isto é devido a que somente se deveria promover o portal, pois se se promociam páginas interiores, poderia acontecer o caso em que os visitantes entrassem por elas no lugar do portal, criando o problema descrito no ponto anterior.
- Muita gente não gosta porque não se sentem livres na navegação, pois entendem que estas partes fixas estão limitando sua mobilidade pela web. Este efeito se torna mais patente se a página com frames tem links a outras páginas web fora do site e, ao clicar um link, se mostra a página nova com as janelas da página que tem frames.
- Alguns navegadores não os suportam. Isto não é muito habitual, mas se estamos fazendo uma página que queremos que seja totalmente acessível deveríamos considerá-lo importante.
- Os bookmarks ou favoritos não funcionam corretamente em muitos casos. Se quisermos incluir um favorito a uma página de um frame que não seja o portal podemos encontrar problemas.
- Pode acontecer que o botão detrás do navegador não se comporte como desejamos.
- Se você quer atualizar mais de um frame com o clique de um link deverá utilizar Javascript. Ademais os scripts podem se complicar bastante quando têm de comunicar vários frames entre si.

Conclusão

O trabalho com frames pode ser mais ou menos indicado dependendo das características da página a desenvolver, é sua tarefa saber se no seu caso deve utiliza-los ou não.

Todos los derechos de reproducción e difusión reservados

Voltar