Treinamento em HTML

Nível: Básico
Horas: 4

Índice

Rua Lauro Vannucci, 1020
Jardim Santa Cândida – Campinas – SP
CEP 13087-548
Tel (19) 3256-5433
Fax (19) 3256-5433
conceptia@conceptia.com.br

Introdução

Linguagem de marcação

Uma linguagem de marcação é utilizada para definir formatos e padrões dentro de um documento. Normalmente elas não possuem qualquer estrutura lógica de controle como as linguagens de programação tradicionais, como condicionais e repetição.

Essas linguagens se utilizam do conceito de marcador ou tag, que possuem um significado quando forem visualizados por algum sistema, para definir como essas informações deverão ser mostradas. Por exemplo, suponha que se queira deixar todas as palavras começadas pela letra “A” em negrito, para executar essa operação em HTML, seria necessário colocar entre as palavras começadas por esta letra, os símbolos “” e “”.

Existem diversos tipos e padrões para linguagens de marcação e a HTML é apenas mais uma delas, mas que se popularizou por causa do advento da Internet, já que a maioria dos documentos que trafegam na rede utilizam ela para exibir suas informações.

Linguagem Interpretada pelo próprio browser

No caso do HTML, ele possui um código com suas várias marcações, mas que se for aberto como um documento texto, não mostra nada além de texto com marcas especiais em determinados pontos. Para que o real efeito do HTML ocorra, é preciso de um outro software, que pegue o documento fonte e interprete o que significa cada uma das marcações existentes.

A função básica de um browser Web (como o Microsoft Internet Explorer e o Nestcape Navigator) é fazer exatamente isso, pegar documentos com marcações HTML e exibi-los aos usuários depois de terem interpretado suas tags, exibindo o real conteúdo de cada um documentos.

Vale destacar ainda que o HTML não é compilado, como algumas linguagens de programação o são, transitando na rede de uma forma que o seu código fonte possa ser lido e interpretado por qualquer pessoa. Ele também não tem nenhum processamento no lado do servidor, este apenas envia os arquivos para a máquina cliente, que fica responsável por fazer todo o processamento.

Modelo cliente / servidor

O modelo básico da internet é o chamado cliente / servidor, onde existe uma máquina central (no caso da internet, não existe um centro, pois ela cresceu a tal ponto que fica impossível gerenciá-la desta maneira) que tem armazenado os documentos que os usuários desejam ver, que são os chamados clientes.

Um servidor pode atender a requisição de diversos clientes ao mesmo tempo, cada um requisitando um documento (como uma página específica de um site) diferente ao mesmo tempo. Esse servidor normalmente possui um poder de processamento bem maior, para garantir que isso possa ser feito sem uma grande demora para os clientes.

As máquinas clientes após receberem os documentos do servidor pela Internet, ficam responsáveis por realizar alguma operação sobre esses arquivos, como por exemplo visualizar as informações de um determinado arquivo num browser ou em algum visualizador específico.

Casos especiais: Netscape | MS-Explorer

Os dois sistemas mais populares para visualização de páginas HTML, aquelas que são obtidas visitando sites na internet, são o Netscape Navigator e o Microsoft Internet Explorer. O primeiro roda na maioria das plataformas existentes, mas o segundo apenas em sistemas baseados no Microsoft Windows.

O Netscape Navigator já foi o líder deste segmento de mercado de visualização de páginas internet, até a aproximadamente seis anos atrás, quando a Microsoft lançou uma forte investida neste campo, disponibilizando seu browser direto em seu sistema operacional. Isso gerou diversos movimentos de acusação de concorrência desleal (deve-se levar em conta que este sistema operacional equipa a imensa maioria de computadores), porém o resultado foi a inversão da liderança deste produtos no mercado de browsers.

Tecnicamente, eles são muito parecidos, ambos devem visualizar o conteúdo de um documento HTML na tela do computador. Teoricamente, o resultado deveria ser igual se os documentos fossem iguais (essa é uma das bases das linguagens de marcação), mas devido a diferentes entendimentos de algumas marcas de documento, a implementação de um e de outro pode mudar, o que acaba gerando resultados diferentes para documentos iguais.

Conceitos Iniciais

Extensão de um novo documento: .html

A extensão padrão para os documentos HTML é o ".html", porém existe uma variação, o ".htm". De movo geral, se um documento com esta extensão for criado e aberto num browser, este tentará mostrar o seu conteúdo como HTML, interpretando tags e tudo o mais.

Vendo um documento HTML

Como já mencionado, a maneira de se ver o resultado de um documento HTML é abrindo-o num browser. Isso tanto pode ser feito abrindo o documento diretamente, como o colocando num servidor internet e acessando-o como mais uma página da internet. O resultado deve ser o mesmo.

Conceito de tags e de documento bem formado

Já foi falado rapidamente o significado de tags ou marcadores no início deste documento. Este item tem como objetivo aprofundar um pouco mais este conceito. Uma tag é uma marca que deve ser colocada num documento e que determinados programas interpretam ela de alguma forma, dando um significado específico, como alterando a formatação do documento naquele ponto.

Normalmente, o texto a ser modificado deve vir entre uma marca que indica o início e o fim da marcação. Essa tag deve ser alguma reconhecida pelo browser. Por exemplo:

```
<i> Texto Modificado </i>
```

No trecho acima, as palavras "`Texto`" e "`Modificado`" seriam mostradas em itálico, pois estão colocadas entre tags que possuem esse significado e alteram o texto desta forma quando interpretado por um browser. A imensa maioria de tags da definição do HTML possui um abre e um fecha correspondente, daí vem a definição de documento bem formado, ou seja, sempre que existir um abre, deve existir um fecha correspondente a este.

Se por algum motivo uma tag não for fechada, o documento continuará a ser exibido, eventualmente com algum erro na visualização, mas nenhum erro de execução será mostrado, diferente da maioria das linguagens de programação tradicionais, onde alguma saída de erro seria mostrada para o usuário.

Modificando parâmetros de tags

As tags também possuem modificadores que podem alterar parte do significado do conteúdo dela num determinado contexto. Por exemplo, o par de tags <body></body> define simplesmente o corpo de uma página HTML, porém no exemplo a seguir:

```
<body bgcolor = “blue”>  </body>
```

Ela continua definindo o corpo de um documento HTML, porém modifica a sua cor de fundo para azul (a não colocação deste parâmetro implica que a cor de fundo continua branca). Existe uma infinidade de parâmetros para cada tag, que apenas com uma referência do lado consegue saber todas as suas opções ou funcionalidades.

Rua Lauro Vannucci, 1020
Jardim Santa Cândida – Campinas – SP
CEP 13087-548

Tel (19) 3256-5433
Fax (19) 3256-5433
conceptia@conceptia.com.br

Base de um Documento

Definindo o documento principal: <html></html>

Antes de se escrever o conteúdo que um determinado documento deverá exibir na tela deve-se identificar claramente que o documento em questão deve ser entendido como um documento HTML e que tudo que estiver contido nele receberá um tratamento especial. Esse tipo de operação é feito utilizando as tags <html> e </html>, indicando que tudo dentro delas será interpretado como uma página Internet.

Definindo o cabeçalho do documento: <head></head>

Todo documento HTML possui um cabeçalho, onde pode ser escritas definições gerais e de padrões a serem seguidos em todas as partes do documento, bem como a escrita de determinados trechos de código de programação ou ligação com documentos externos. A definição do título da página, (detalhada abaixo) também deve ser escrita entre as tags <head> e </head>;

Dando um título para o documento: <title></title>

Por definição, todos os aplicativos que executam sobre a plataforma Microsoft Windows (e uma grande variedade de outras plataformas) que possuem interação por janelas, possuem uma barra superior que mostra um título para o documento sendo exibido ao mesmo tempo em que também fala qual o aplicativo que está sendo executado na janela corrente.

Nas páginas da Internet não é diferente a existência desses elementos. Por tanto, para colocarmos o título que desejarmos numa janela do browser utilizamos as tags <title></title>, que deve vir escrita entre as tags <head> e </head>. O título desejado deve vir colocado entre as tags. Podemos ter como exemplo:

```
...
<head>

<title> Título da minha Janela </title>

</head>
...
```

Nesse exemplo, será exibida uma janela com o título “Título da Janela”.

Rua Lauro Vannucci, 1020
Jardim Santa Cândida – Campinas – SP
CEP 13087-548
Tel (19) 3256-5433
Fax (19) 3256-5433
conceptia@conceptia.com.br

Definindo o corpo principal: <body></body>

A partir da definição destas tags, que indicam o início do corpo principal do documento, ou seja, o que será mostrado na tela do monitor em último caso, pode-se começar a escrever o conteúdo da página em si, colocando diversas tags de formatação, definindo campos de formulários, atribuindo textos, mostrando imagens e criando links entre diversos outros documentos.

Alterando a cor de fundo: atributo bgcolor

Como já mencionado anteriormente, a grande maioria de tags HTML possui parâmetros modificadores que de alguma forma alteram a forma como o conteúdo que virá entre as tags será exibido. Um dos parâmetros existentes para a tag <body></body> é a que altera a cor de fundo de toda a janela. Veja o exemplo:

```
...

<body bgcolor = "blue"> </body>

...
```

Neste caso, toda a janela do browser ficará com a cor azul, porém no lugar do azul, poderiam ser colocadas diversas outras cores, todas em inglês que mudariam também o fundo. Além do nome por extenso, também poderia ser colocado o valor correspondente da cor em hexadecimal, precedido do símbolo sustenido: "#".

Exibindo Informações

Escrevendo dados num documento

Esta seção trata principalmente da forma como os dados serão exibidos na tela, alterando sua formatação, alinhamento e disposição na tela, sempre utilizando tags HTML para permitir isso. Por padrão, pode ser escrito qualquer coisa no documento, mesmo fora das tags, e mesmo assim o browser irá exibir algum conteúdo, com a diferença de que não se tem nenhum controle sobre o efeito.

Vêm daí a importância de sempre utilizar tags para controlar a formatação. Não apenas abrindo uma tag e colocando o conteúdo, como também fechando a tag correspondente, para ao final gerar um documento HTML bem formado, ao mesmo tempo que tem-se um controle preciso do que um determinado browser irá exibir e não deixar que ele tome decisões que vá distorcer o conteúdo.

Criando cabeçalhos: <hn></hn>

A maneira mais simples de se definir um cabeçalho (ou título) num documento HTML, é através das tags <hn> e </hn>, onde o valor do "n" seria substituído por algum número entre 1 e 6, sendo que valores menores indicam que o tamanho da fonte exibida será maior e números menores indicam títulos de tamanho menores. Por exemplo:

```
<h1> Comparando Tamanhos </h1>
<h2> Comparando Tamanhos </h2>
<h3> Comparando Tamanhos </h3>
<h4> Comparando Tamanhos </h4>
<h5> Comparando Tamanhos </h5>
<h6> Comparando Tamanhos </h6>
```

Um parágrafo:

Um texto normal, ou melhor, que seja encarado como normal e que apareça em diversos pontos da página, deve vir entre as tags e , que denotam a existência de um parágrafo. Para visualização de algum resultado, não é necessário que exista uma tag de fechamento, porém isso pode fazer com que versões mais antigas de browsers vejam o resultado de forma diferente.

```
<p> Texto, Texto, Texto, Texto, Texto, Texto, Texto, Texto,
Texto, Texto, Texto, Texto, Texto, Texto, Texto, Texto,
Texto, Texto, Texto, Texto, Texto, Texto, Texto, Texto,
Texto, Texto, Texto, Texto, Texto, Texto, Texto </p>
```

Neste exemplo, será criado um parágrafo contendo o texto entre as tags. Caso sejam criados mais de um parágrafo, eles terão um pequeno espaçamento para indicarem que são elementos diferentes.

Alterando o alinhamento de um texto

Dentro das tags que trabalham com texto, pode vir um modificador, também chamado de parâmetro, que altera o alinhamento do texto que vier entre as tags (daí a importância de fechar as tags, pois mostra explicitamente até onde vai o alinhamento proposto para uma determinada tag). Existem quatro alinhamentos possíveis: esquerdo, centralizado, direito e justificado.

```
<p align='left'> Texto à esquerda </p>
<p align='center'> Texto à esquerda </p>
<p align='right'> Texto à esquerda </p>
<p align='justify'> Texto à esquerda </p>
```

O texto com alinhamento à esquerda é o valor padrão, fazendo com que não seja necessário colocar explicitamente esse item quando estiver sendo declarado. O texto com alinhamento justificado, faz com que o texto fique alinhado tanto à esquerda, quanto a direita, distribuindo as palavras de uma forma homogênea.

Mudando a cor do texto:

As tags e servem para modificar a fonte, o tamanho e as cores utilizadas num trecho de texto de forma global, não apenas uma ou outra palavra. Com isso, tem-se um controle preciso da formatação que está se tentando alcançar.

```
<font color='blue' size='20'> <p> Texto escrito </p> </font>
```

Tornando o texto negrito:

Para dar um destaque maior a algum trecho de texto, pode-se usar as tags e , que destacam o texto que estiver contido entre elas. Elas podem ser colocadas dentro de outras tags, como nas de parágrafo ou de título, desde que o ocorra o devido fechamento. Segue um exemplo:

```
<p> Texto normal, mas agora com <b> negrito </b> . </p>
```

O início da frase está normal, porém existe uma palavra destacada.

Tornando o texto itálico:

Da mesma forma que se pode destacar algum trecho de um texto utilizando negrito, como mostrado acima, também usa-se as tags e para mostrar um trecho de código em itálico, como alguma citação ou palavra estrangeira. O seu uso é idêntico ao caso da tag negrito.

```
<p> Agora em <i> itálico </i> ... </p>
```

Tornando o texto sublinhado: <u></u>

Por fim, o último dos modificadores padrão de textos é o de sublinhado, que deve ser utilizado da mesma forma que as tags exibidas anteriormente, de negrito e itálico. Ele deixa marcado com uma linha inferior, abaixo das palavras, todas as palavras entre elas.

```
<p> Parte <u> deste</u> texto <u> será sublinhado </u>. </p>
```

Vale destacar que pode ocorrer qualquer combinação destas tags mostradas até o momento para se chegar ao efeito desejado.

Quebrando linhas:

Está tag não possui fecha, ela é definida unicamente. Ela serve para inserir quebras de linhas no documento, fazendo com que o texto que vem abaixo seja ainda mais deslocado. É muito importante para arrumar a disposição dos elementos gráficos na tela. Pode-se colocados quantos desejar.

```
<h1> Linha 1 </h1> <br><br><br><br><br> <h1> Linha 2 </h1>
```

Neste exemplo, apesar de todas as frases estarem na mesma linha, foram inseridas cinco quebras de linha, o que irá fazer com que o segundo texto seja deslocado em cinco linhas para baixo.

Dividindo em seções: <hr>

Da mesma forma que a tag
, está também não possui uma tag para indicar o seu final, ela é definida unicamente. Ela serve para colocar linhas verticais, principalmente para implementar a divisão de seções dentro de um mesmo conteúdo. Por exemplo

```
<p> Primeira Seção </p> <hr> <p> Segunda Seção </p>
```

Ligações entre Páginas

Conceitos de links entre páginas

Um dos fundamentos da Internet é permitir que se possa ir de um lugar a outro da rede com apenas um clique. Esta é a função dos links, estabelecer conexões entre diversas páginas, sejam elas internas, dentro do mesmo site, seja referenciando sites externos, de outras empresas, mas que sejam relevantes para o conteúdo sendo discutido no momento.

Dessa forma, surgem dois conceitos, o de link absoluto e o de link relativo, fundamentais para a implementação de links dentro de uma página. Um link absoluto define todo o caminho onde uma segunda página está hospedada e que será aberta no browser. Um link relativo indica apenas parte do endereço, utilizando conceitos dos diretórios em uso para indicar outra página.

Em maiores detalhes, o diretório corrente onde uma página HTML está salva é chamado diretório raiz. Para se referenciar ao diretório (e eventuais arquivos) que está abaixo numa hierarquia, usa-se o ponto-ponto: "`..`". Para indicar um diretório acima na hierarquia, utiliza-se o nome do diretório, mas caso exista mais de um, deve-se separar os nomes por uma barra "`/`".

Esses conceitos são os mesmos que utilizados para a navegação em sistemas computacionais baseados em linha de comando, como o MS-DOS ou um terminal, no Linux. Prefira sempre utilizar caminhos relativos aos absolutos, pois isso garante uma facilidade maior caso seja necessário modificar o endereço em que a página está hospedada, pois todo os demais diretórios ficam intactos.

Criando um novo link: <a>

Agora, tecnicamente, um link é definido no HTML como um pedaço de texto (ou outro objeto qualquer, como uma imagem) que ao ser clicado leva a um outro lugar, ou seja, à outra página. Isso é feito utilizando as tags `<a>` e `</a>`, que deverão englobar o texto que será dado destaque e que será clicado pelo usuário final. Por exemplo, criando um link vazio:

```
<p> Texto: <a> Clique Aqui </a> </p>
```

Neste exemplo, nenhuma operação será feita, porém o link foi criado e pode ser clicado (aparecerá a mãozinha, inclusive) porém o usuário

permanecerá na mesma página. Para executar alguma operação que realmente leva o usuário a outra página, é preciso incluir um parâmetro que mostra a página de destino. Por exemplo:

```
<p>Texto:<a href='www.conceptia.com.br'>Clique Aqui</a></p>
```

Neste caso, o resultado será o redirecionamento para o link absoluto definido na tag href. Usando um link relativo:

```
<p>Texto:<a href='../arqs/artigo.html'>Clique Aqui</a></p>
```

Neste último exemplo, o arquivo que será aberto é procurado um nível abaixo da hierarquia corrente, depois dentro do diretório arqs e por fim, procura-se um arquivo de nome artigo.html. Caso ele não exista, um erro é retornado, senão o documento é exibido na tela.

Rua Lauro Vannucci, 1020
Jardim Santa Cândida – Campinas – SP
CEP 13087-548

Tel (19) 3256-5433
Fax (19) 3256-5433
conceptia@conceptia.com.br

Inserindo Imagens

Inserindo uma imagem: <img>

Uma imagem é encarada como um objeto que pode ser inserido em qualquer lugar da página, não precisando inclusive ter uma tag que a feche. No ponto em que se deseja colocar a imagem deve-se colocar a tag correspondente e indicar o local onde a imagem fonte está armazenada e o nome da imagem a ser inserida. Todos os comentários de links absolutos e relativos se aplicam aqui.

```
<img src='imgs/teste/cavalo.jpg'>
```

Para que uma imagem se torne um link, basta que uma tag <a> e englobe uma tag <img>.

Tabelas

Formatando dados usando tabelas: <table></table>

A melhor maneira, senão a única, de dispor elementos gráficos numa página HTML é através do uso de tabelas, pois garante que um determinado elemento ou texto ficará aonde deseja que ele fique, porém isso aumenta em muito a complexidade na elaboração de páginas. Atualmente, existem diversos editores que facilitam essa operação, pois desenham as tabelas e ocultam o código.

Para declarar uma nova tabela, deve ser feito uso das tags <table> e </table>, que indicam que uma nova tabela será criada, com suas várias linhas e colunas. Neste ponto, não existem dados ainda, então o resultado visível não é muito grande, mais adiante é que os dados serão inseridos dentro desta nova tabela.

Mostrando as bordas: atributo border

Quando se está trabalhando com tabelas, é fundamental saber onde as linhas e colunas estão aparecendo, mesmo que para a página como um todo seja mais interessante, por questões de design, que as bordas não apareçam. Enfim, ao menos em fase de desenvolvimento, as bordas devem aparecer. Isso é feito através do parâmetro border, dentro de table. Assim:

```
<table border='1'>

</table>
```

Neste caso, esta tabela terá bordas de tamanho 1.

Incluindo uma nova linha: <tr></tr>

Para incluir uma nova linha à uma tabela, deve-se usar as tags <tr> e </tr>. Tudo que estiver entre elas, será disposta na mesma linha, independente do resultado final de exibição. Elas devem vir dentro das tags <table> e <table>. Assim:

```
<table border='1'>
     <tr> Exemplo </tr>
     <tr> Exemplo </tr>
     <tr> Exemplo </tr>
</table>
```

Aqui, serão escritas três linhas nesta tabela.

Rua Lauro Vannucci, 1020
Jardim Santa Cândida – Campinas – SP
CEP 13087-548
Tel (19) 3256-5433
Fax (19) 3256-5433
conceptia@conceptia.com.br

Incluindo uma nova célula: <td></td>

Por fim, para declarar todas as possibilidades de uma tabela, deve ser possível declarar colunas dentro de uma tabela. Isso é feito utilizando as tags <td> e </td>, que por sua vez devem vir dentro das tags <tr> e </tr>. Os dados propriamente ditos são colocados dentro de <td> e </td>. Vale destacar, que desta forma, fica definido uma célula. Por exemplo:

```
<table border='1'>
     <tr> <td> C11 </td> <td> C12 </td> <td> C13 </td> </tr>
     <tr> <td> C21 </td> <td> C22 </td> <td> C23 </td> </tr>
     <tr> <td> C31 </td> <td> C32 </td> <td> C33 </td> </tr>
</table>
```

No exemplo acima, uma tabela, com três linhas e três colunas.

Mesclando linhas: atributo rowspan

Em algumas situações, algumas células não possuem dados e para não deixa-las simplesmente vazias, o que pode tornar mais confuso o entendimento do código, existe o conceito de span, que serve para mesclar (como os conceitos do Microsoft Excel), juntando o espaço de duas células e tornando-a uma só.

No caso do atributo rowspan, deve-se definir quantas linhas deseja-se incluir, daquela mesma coluna, na célula final resultante. Isso é feito apenas na primeira célula, sendo que as demais devem ser removidas. Pegando o exemplo anterior, vamos mesclar toda a primeira coluna:

```
<table border='1'>
     <tr> <td rowspan='2'> C11 </td> <td> C12 </td> </tr>
     <tr> <td> C22 </td> </tr>
     <tr> <td> C32 </td> </tr>
</table>
```

Mesclando colunas: atributo colspan

A mesma idéia aplicada no item acima, para a questão de mesclar linhas pode ser aplicada para mesclar colunas, onde diversas colunas podem ser unidas numa única, removendo o código das demais. Assim:

```
<table border='1'>
     <tr> <td colspan='2'> C11 </td> <td> C12 </td> </tr>
     <tr> <td> C21 </td> <td> C22 </td> </tr>
```

```
     <tr> <td> C31 </td> <td> C32 </td> </tr>
</table>
```

Alterando a cor de fundo de uma célula

Cada célula numa tabela pode ser tratada independentemente, modificando seus parâmetros, e fazendo com que apenas uma delas tenha sua exibição modificada. Para termos este efeito, é necessário incluir nas tags <td> os modificadores correspondentes para mudar o comportamento daquela célula em especial.

Dessa forma, para alterar a cor de fundo de uma célula, é preciso incluir o atributo bgcolor <??? Confirmar> o qual irá receber como valor modificador a nova cor que será colocada no fundo da célula. Esse nome de cor seguem os mesmo padrões de cores que devem ser utilizados em HTML, ou seja, devem ser em inglês ou o valor hexadecimal precedido de sustenido e entre aspas.

```
<table border='1'>
     <tr>
          <td bgcolor='0000FF'> azul </td>
          <td bgcolor='yellow'> amarelo </td>
     </tr>
     <tr>
          <td bgcolor='gray'> cinza </td>
          <td bgcolor='#FF0000'> vermelho </td>
     </tr>
</table>
```

Alterando o alinhamento de uma célula

Da mesma forma que se pode incluir modificadores para cores de células, independentes umas das outras, pode-se também alterar o alinhamento do conteúdo existente dentro de uma célula, seja texto, seja uma imagem, como exemplos. Alguns exemplos:

```
<table border='1'>
     <tr>
          <td align='left'> esquerda </td>
          <td align='center'> centro </td>
     </tr>
     <tr>
          <td align='right'> direito </td>
          <td align='justify'> justificado </td>
     </tr>
</table>
```

Listas

Definindo listas gerais: e

Uma outra forma de organizar as informações numa página HTML é através de listas, que são itens estruturados e separados por algum símbolo de marcação, como um círculo ou um quadrado, ou ainda numerados, para informar alguma ordem lógica entre as diversas informações que desejam ser exibidas.

As listas definidas com as tags <ol> possuem alguma ordenação lógica, ou seja, estão numeradas num intervalo de valores previamente conhecido. Já as listas definidas com <ul> não possuem uma ordem lógica, sendo utilizado algum caracter marcador para a disposição dos dados. Por exemplo:

```
<ol>
     <li> Primeiro </li>
     <li> Segundo </li>
     <li> Terceiro </li>
     <li> Quarto </li>
</ol>
```

Na lista acima, será exibido quatro itens, numerados de um até quatro, sendo colocado ao lado de cada marcador o valor que estiver entre as tags <li>, que é o marcador para definir um novo elemento a ser posto dentro de alguma lista, seja ela ordenada, como no exemplo acima, seja uma lista não ordenada. Outro exemplo:

```
<ul>
     <li> Um </li>
     <li> Dois </li>
     <li> Três </li>
     <li> Quatro </li>
</ul>
```

O resultado do exemplo acima será o mesmo que o obtido no primeiro exemplo, com exceção do formato dos marcadores, que ao invés de números, será exibido bolinhas para diferenciar os diferentes elementos que são mostrados.

Formulários

Trabalhando com dados em HTML

Para projetos maiores e que necessitam trocar ou obter informações do usuário, texto simples e estático, não é muito interessante, pois acaba sendo um caminho de mão única de comunicação entre o desenvolvedor e o usuário que está acessando os dados. Sendo assim, a linguagem HTML oferece suporte a um grupo de estruturas para fazer o caminho de volta.

Sempre que alguém preenche seus dados numa página de cadastro, ela está fazendo uso de formulários do HTML. Essas estruturas fornecem campos de texto por extenso, botões, caixas de seleção e de senha. Em sistemas mais complexos e que trabalham com bases de dados, esses conceitos são dos mais utilizados para a elaboração do sistema.

Nos itens a seguir, serão mostrados os principais elementos para construção de formulários em HTML e de que forma eles se comportam num sistema.

Definindo um novo formulário

Para deixar claro que daquele ponto em diante será utilizado um formulário e seus diversos elementos, deve-se antes declarar essa informação explicitamente, inclusive com alguns outros parâmetros que irão controlar o comportamento dos dados entre o computador cliente e o servidor, onde as informações poderão ser tratadas e, por exemplo, inseridas numa base.

```
< form name='pri_form' method='post' action='arquivo.php'>
…
</form>
```

Todos os demais elementos devem vir entre as tags `<form>` e `</form>`. O primeiro parâmetro, `name`, indica o nome do formulário, o que servirá para referenciá-lo em outros trechos do código. O parâmetro `method` indica de que forma os dados serão enviados para o servidor (post indica que os dados serão ocultos, a outra opção, get, os dados irão visíveis) e por fim, `action`, indica qual o arquivo que deve tratar os dados depois do envio.

Opções de entrada de dados: <input>

Uma das tags mais simples e ao mesmo tempo mais utilizadas, é a tag <input>. Ela não possui uma correspondente que a fecha e ela é utilizada para permitir ao usuário inserir informações alfanuméricas ou então mostrar um botão clicável. Existem alguns modificadores que podem ser utilizados que alteram a forma como os dados serão digitados pelos usuários.

A seguir são mostrados os vários tipos de estruturas que a tag input pode representar, alterando apenas o campo type, que indica qual o tipo de estrutura deve ser visualizada no formulário. Além do tipo, é importante colocar o parâmetro name ou id, para mostrar qual o identificador para aquela estrutura, para que possa ser referenciado no futuro, caso seja necessário.

Criando um novo campo: parâmetro text

Quando se deseja digitar alguma informação alfanumérica num formulário deve-se utilizar o parâmetro text na declaração do campo. Isso é feito da seguinte forma:

```
<input type='text' name='campo' maxlength='50' size='50'>
```

A palavra input indica uma nova entrada de dados. O item, type='text', indica um novo campo de entrada para texto. O item name='campo', declara um nome para o campo poder ser acessado por outras linguagens de programação. Os outros dois parâmetros são itens para visualização. O primeiro mostra o número máximo de caracteres que podem ser inseridos e o segundo o tamanho do campo que será mostrado na tela.

Outros parâmetros podem estar presentes. Eventualmente, podem existir informações padrões que vieram de outras fontes, como uma base de dados, e deseja-se que essa informação seja o padrão do campo, ou seja, que já venha digitado alguma coisa nele. Para isso, usa-se o parâmetro value, seguido do texto a ser exibido. Por fim, existe o item readonly, para não permitir modificações no conteúdo do campo.

Criando um botão: parâmetro button

Outro item muito interessante em formulário é a inserção de botões, que quando clicados, disparam alguma ação, como executar o código fonte de alguma outra linguagem de programação. É necessário o uso de outra linguagem de programação porque o HTML não fornece

suporte a um grande número de estruturas muito comuns em programação (condicionais, repetições, etc).

```
<input type='button' name='botao' value='Aperte'>
```

A declaração do botão acima não realizará qualquer operação em especial, apenas permitirá ao usuário ficar apertando um botão na tela, no entanto esse exemplo é bem explicativo. A declaração do botão segue o mesmo padrão de um campo texto, com a diferença do tipo. O parâmetro value também tem outra função, ele serve para indicar o título inserido no corpo do botão.

Para associar algum tipo de comportamento mais avançado ao botão, deve-se incluir um outro parâmetro, o mais comum é o `onClick` associado a alguma função, que também pode ser chamado de evento. Ele indica que sempre que o botão for clicado ele chamará a função (ou método) ao qual estiver associado, por exemplo da linguagem JavaScript. Outro exemplo:

```
<input type='button'
     name='botao_2'
     value='Aperte'
     onClick='ola_mundo();'>
```

A função ola_mundo() deve ser escrita em algum trecho do código e ser visível por esse trecho de código. No momento que o usuário apertar o botão, a função será chamada e código dela executada.

Criando um botão de submissão: parâmetro submit

A declaração de um botão de submissão é idêntica ao de um botão comum, inclusive com sua visualização sendo idêntica ao de um botão normal, com a diferença de que já existe uma operação pré-configurada, a de envio dos dados do formulário para o formulário onde as páginas HTML estão hospedadas, fazendo com que os dados sejam tratados. Por exemplo:

```
<input type='submit' name='envio' value='Enviar'>
```

Será criado um botão com o título Enviar que quando clicado pegará todos os dados preenchidos nos demais campos deste formulário e o enviará para o servidor.

Criando um botão de limpeza: parâmetro reset

Um botão de reset possui a mesma definição do submit, porém com uma ação associada diferente, e potencialmente destrutiva, ou seja, uma vez acionado o botão, todas as informações de todos os campos do formulário são apagadas, fazendo com que os campos retornem ao estado inicial. Essa operação é perigosa por eventualmente fazer com que o usuário digite todas as informações novamente. Por exemplo:

```
<input type='reset' name='apag' value='Apagar'>
```

Criando uma nova checkbox: parâmetro checkbox

Uma checkbox (ou caixa de marcas) permite ao usuário marcar, dentre um grupo de opções, aquelas que ele deseja, mais de uma se for o caso. Para indicar que todas as opções se referem ao mesmo grupo, os nomes de todos os campos devem ser o mesmo. Por exemplo:

```
Quais cores você gosta?
<input type='checkbox' name='cores'> Azul
<input type='checkbox' name='cores'> Vermelho
<input type='checkbox' name='cores'> Amarelo
```

Neste caso, o usuário poderia escolher qualquer combinação de cores, marcando uma, duas ou as três opções.

Criando um novo radiobutton: parâmetro radio

A forma de declarar um campo radiobutton (ou radio) é muito semelhante ao de checkbox, inclusive com a obrigatoriedade de o nome ser igual para todas as opções, sendo que a maior diferença está que nesta opção, a seleção é única, ou seja, só pode ser escolhida uma opção dentre as várias que são fornecidas para o usuário. Por exemplo:

```
Qual a sua faixa etária?
<input type='radio' name='idades'> Menor 18
<input type='radio' name='idades'> Entre (inclusive) 18 e 50
<input type='radio' name='idades'> Maior 50
```

Nestas opções, um mesmo usuário não pode escolher mais de uma opção, pois ele não pode se encaixar, simultaneamente em duas faixas etárias distintas, como é o caso.

Criando um campo de senha: parâmetro password

Campos password são úteis quando não se deseja que outros usuários vejam as informações que estão sendo digitadas num campo, por

exemplo numa senha ao se validar a entrada de um usuário de um sistema, ou mesmo quando o usuário estiver inserindo sua senha num cadastro qualquer. Sua declaração é muito parecida com a de um campo normal:

```
<input type='password' name='sna' size='10' maxlength='10'>
```

Aqui é criado um campo de tamanho dez caracteres e que só permite ao usuário digitar dez caracteres quando da inserção de dados. No lugar dos caracteres, que seriam o normal aparecer na tela, são exibidos asteriscos, ocultando assim a entrada dos usuários.

Caixas de seleção: <select></select>

Caixas de seleção são muito utilizadas para permitir ao usuário selecionar apenas uma opção dentre várias existentes, sem mostrar todos os itens ao mesmo tempo (como no caso da checkbox), obtendo-se um ganho de espaço considerável na tela, facilitando a formatação e diagramação da página. Sua declaração segue:

```
<select name='selecao'>
     <option value='1'> Item 1 </option>
     <option value='2'> Item 2 </option>
     <option value='3'> Item 3 </option>
</select>
```

Neste caso, foi criada uma caixa de seleção com três opções, sendo que cada opção possível é declarada entre as tags <option> e </option>. Deve ser dado um nome para a caixa de seleção para referência futura. Um parâmetro utilizado dentro da tag <option>, value, mostra qual o valor que será considerado quando a opção for selecionada.

Caixas de texto: <textarea></textarea>

Esse elemento é utilizado principalmente quando a quantidade de texto a ser digitado pelo usuário é muito grande. Ela permite uma melhor visualização de grandes quantidades de caracteres, mesmo que isso não seja a melhor solução de exibição de informações em páginas HTML. Sua declaração segue o seguinte modelo:

```
<textarea name='tarea' rows='10' cols='40'></textarea>
```

No exemplo acima, é declarada uma textarea de nome tarea, que tenha dez linhas e quarenta colunas, onde cada coluna é o

espaçamento de um caractere. Caso exista algum texto que se deseja colocar por padrão neste campo, ele deve ser posto entre as tags.

Janelas

Definição de janelas em HTML

Uma janela de um browser pode ser considerada inteira, com apenas um documento ocupando toda a área visível ou então pode ser dividida em diferentes partes, cada uma ocupada por um documento HTML diferente, que de alguma forma se relacionam, ou seja, ao se clicar num link numa janela a página será aberta em outra. Esse conceito é chamado de frames.

Como nem todos os browser suportam a definição de frames (todas as versões mais recentes suportam atualmente) nem sempre foi uma boa idéia implementar soluções utilizando esses conceitos. Vale destacar que com o uso de outras tecnologias (como PHP e ASP) o uso de frames pode dificultar o desenvolvimento dos sistemas, sendo mais simples utilizar uma única página.

Documento de definição de janelas: <frameset></frameset>

Quando se usa frames, deve existir um documento especial que define como ele serão montados, quais os arquivos serão chamados e de que forma eles se relacionam. Isso é feito através do uso das tags <frameset></frameset>, que devem ser utilizadas no lugar das tags <body></body> que existem num documento normal HTML.

Dentro destas tags é que virão algumas configurações de parâmetros e de frames que serão utilizadas para exibir o resultado. Vale destacar que quando se deseja visualizar o resultado, este documento de definição é o que deve ser aberto no browser e não qualquer outro que compunham o conteúdo que será visto na tela.

```
<frameset cols='25%, 75%'>
     <frame name='esq' src='menu.html'>
     <frame name='dir' src='conteudo.html'>
</frameset>
```

Nesta definição, é definido um frame com duas colunas sendo uma ocupando 25% da tela e a outra ocupando 75% da tela. A seguir, são definidos o conteúdo dos dois frames, sendo que no primeiro, chamado esq, deve ser aberto o arquivo menu.html e no menu direito será aberto o arquivo conteudo.html. Uma definição alternativa, é que ao invés de separar os frames por colunas, separar por linhas, neste caso, no lugar de cols deverá ser colocado a palavra rows.

Referência

www.w3c.org
World Wide Web Consortium. Site de referência mundial para desenvolvimento Internet, pois concentra todas as definições dos padrões utilizados. Possui um vasto material de consulta, do básico ao avançado, além de diversos artigos técnicos e novas tendências mundiais no que toca o avanço da Internet.