# World Wide Web

A Web é um conjunto de documentos acessíveis por meio da Internet. Esses documentos (ou páginas Web) contêm uma tecnologia chamada hipertexto. Onde você poderá percorrer partes do documento e ir para outros documentos através de pontes chamadas de links.

A quantidade de informação que está disponível na Internet é muito mais do que você poderia assimilar durante uma vida inteira, e esta informação se encontra das mais variadas formas, podendo se perder facilmente. Assim é muito importante o sistema de navegação, ou seja, a forma como as informações estão dispostas.

Poderia descrever a World Wide Web como um sistema de informação em hipertexto, gráfico, distribuído, independente de plataforma, dinâmico, interativo e global, utilizado na internet.

O hipertexto permite que você leia um texto e navegue por ele e por informações visuais de forma não-linear, com base nas informações que deseja obter. A Web fornece recursos de imagens e animação fantásticos.

É independente de plataforma pois você pode acessar as informações disponíveis na Web igualmente bem a partir de qualquer computador, sistema operacional e monitor de vídeo, inclusive de vários tipos de dispositivos móveis como celulares ou Tablet.

Como o volume de informações é muito grande ela é distribuída por diversos sites (um Site da Web é um local que divulga algum tipo de informação. Quando você exibe uma página Web, o seu navegador se conecta a esse site para obter essas informações). Estas informações contidas nestes sites são encontradas através de URLs (endereço na internet).

## Criando Páginas da WEB

Uma página Web é composta de textos e comandos especiais (tags) de HTML, um acrônimo de Hypertext Markup Language. Essa linguagem é bastante simples e tem como finalidade básica formatar o texto exibido e criar ligações entre as páginas da Web, criando assim documentos com o conceito de hipertexto.

Para que o conteúdo de um documento HTML possa ser formatado e exibido na Internet devemos utilizar um programa chamado browser, o navegador de internet. Ele lê o conteúdo do arquivo, interpreta os comandos e exibe sua página. Existem diversos tipos de browsers disponíveis no mercado, como por exemplo, o Internet Explorer, o Firefox, o Chrome entre outros.

O código pode ser escrito usando o mais simples editor de texto, como o bloco de notas do Windows. Porém a forma mais prática e produtiva é utilizar um editor HTML. Há vários no mercado como: HotDog Professional (www.sausage.com), o SiteAid (baixado do site www.siteaid.com), o FrontPage da Microsoft, o Dreamweaver, etc.

## Visualização das Paginas

Preciso estar conectado a internet para visualizar as páginas que eu criei?  
Não, o browser funciona independente de se estar conectado a Internet. A melhor forma de se trabalhar desenvolvendo uma página HTML é off-line, desconectado.

Primeiro você cria as páginas em seu computador e as testas por meio do browser. Depois, deve colocá-las em um servidor Web para que outras pessoas também possam visualizá-la.

# SERVIDOR WEB

Para exibir páginas na Web e navegar por elas, você precisará apenas de um navegador da Web. Para divulgar páginas na Web, você precisará, na maioria dos casos, de um servidor Web.

*Servidor Web* é o programa que é executado em um site da Web e que é responsável por atender ás solicitações de arquivos feitas pelo navegador da Web. Você precisará de um servidor Web para divulgar documentos na Web.

# HTML

A HTML é uma linguagem de marcação. Criar um documento em uma linguagem de marcação significa que você começa com o texto da sua página e inclui tags especiais no início e no final de determinadas palavras ou parágrafos.

As tags indicam as diversas partes da página e produzem diferentes efeitos no navegador, elas são os comandos da “linguagem” HTML.

A primeira página a ser exibida geralmente recebe o nome de index.htm ou .html, dependendo do provedor que hospeda a página, quando o site tem conteúdos feitos na linguagem PHP a página inicial pode se chamar index.php.

As tags normalmente são especificadas em pares, delimitando um texto que sofrerá algum tipo de formatação. As tags são identificadas por estarem entre os sinais < > e < / >.

Entre os sinais < > são especificados os comandos propriamente ditos. No caso de tags que necessitam envolver um texto, sua finalização deve ser feita usando-se a barra de divisão /, indicando que a tag está finalizando a marcação de um texto.

O formato genérico de uma tag é:

**<nome da tag>** Conteúdo que será visualizado**</nome da tag>**

A maioria das tags precisam ser iniciadas com <>, e finalizadas com </>, mas algumas tags não possuem finalização.

Assim como outras linguagens, a HTML possui uma estrutura básica para suas páginas. Para que um browser interprete corretamente a página, ela deve possuir alguns comandos básicos que sempre deverão estar presentes. Alguns browsers até dispensam seu uso, porém é melhor assumir como parte fundamental da página tais comandos.

Uma página HTML possui três partes básicas, a estrutura principal, o cabeçalho e o corpo da página. Toda página deve iniciar com o comando (tag), **<html>** e ser encerrado com o comando **</html>**.Esse par de comandos é essencial. A área de cabeçalho é opcional e é delimitada pelo par de comandos **<head>** e **</head>**. Estes comandos para cabeçalho são usados para especificar alguns poucos comandos da linguagem. Eles são opcionais, ou seja, um programa HTML pode funcionar sem eles. Mas é conveniente usá-los, pois o título da página é acrescentado através deles. E ainda temos as tags **<Title>** e **</Title>**, estes comandos delimitam o texto que irá aparecer na barra de título do browser.

A maioria dos comandos será especificada no corpo do programa que é delimitado pelas tags **<body>** e **</body>,** o qual é um comando obrigatório.

Formato básico de uma página HTML:

<HTML>

<HEAD>

<TITLE> Aqui vai o Titulo da Página </TITLE>

</HEAD>

<BODY>

Aqui vai o Corpo da página

</BODY>

</HTML>

Descrição das Tags:

**<HTML> </HTML>** Esta tag marca o início do programa HTML

**<HEAD> </HEAD>** Esta tag marca início e fim de cabeçalho

**<TITLE> </TITLE>** Esta tag delimita o texto que irá ser visualizado na barra de título do browser.

**<BODY> </BODY>** Esta tag delimita o corpo do programa, aonde a os comandos serão colocados

## Principais Elementos de uma Página

Uma página HTML é composta basicamente de títulos, textos, parágrafos, imagens e links, responsáveis pela chamada de outras páginas para a tela. Todos esses elementos são posicionados na página por meio de comandos específicos da linguagem.

### TÍTULO

É o texto que aparece na barra de título do browser.

### IMAGEM

São figuras, desenhos e fotos usados para ilustrar a página.

### TEXTO

É a informação mais comum dentro da página. Pode ser formatado através de vários comandos.

### LINK

É um trecho que aparece destacado do restante do texto, normalmente sublinhado e com outra cor. Ao clicar no link, o browser acessa outra região da página atual ou uma página localizada em qualquer lugar da Internet. Uma figura também pode ser usada como um link, ou seja, clicando na figura saltamos para outro local.

### CABEÇALHO

São linhas de texto com tamanhos especiais. Existem seis tamanhos prefixados de cabeçalho.

## TÍTULO

O título de uma página web indica qual o assunto abordado e irá aparecer na barra de título do browser. Este título é utilizado por programas de lista (hotlist) do seu navegador e também por outros programas que catalogam páginas da Internet. Para atribuir um título a página você deverá utilizar a tag:

<Title> </Title>

Esta tag sempre será incluída no cabeçalho (entre as tags <Head> </Head>).

Considerações sobre o título:

1. Você poderá ter apenas um título
2. Texto deverá ser simples (sem formatação) e não poderá conter outras tags
3. Escolha um texto curto e que descreva o conteúdo da página

## CABEÇALHOS

Os cabeçalhos são usados para dividir seções do texto, como capítulos de um livro. A HTML divide seis tamanhos de cabeçalhos, de H1 a H6, que aplicam um tamanho de fonte diferenciado no texto que envolve, uma linha antes e depois e, além disso, dão um efeito de negrito. O maior tamanho é o H1 e o menor é o H6.

<H1> Texto </H1>

Experimente o seguinte código:

<html>

<head><title>Cabeçalhos</title></head>

<body>

<h1>Cabeçalho Nível 1</h1>

<h2> Cabeçalho Nível 2</h2>

<h3> Cabeçalho Nível 3</h3>

<h4> Cabeçalho Nível 4</h4>

<h5> Cabeçalho Nível 5</h5>

<h6> Cabeçalho Nível 6</h6>

</body>

</html>

## PARÁGRAFOS

Para forçar o início de um novo parágrafo, ou seja, avançar uma linha em branco e iniciar o texto na Segunda linha após o final do parágrafo anterior, deve ser usado o comando <p>. Esse comando pode aparecer individualmente ou em par: <p> </p>.

Quando um comando <p> é inserido em um local do programa, o browser irá sempre avançar uma linha em branco, posiciona-se na linha seguinte ao comando <p>.

## QUEBRA DE LINHA

O comando <br> faz uma espécie de quebra de linha. Este comando insere uma linha em branco no seu local ou na linha seguinte á qual ele foi inserido, caso esteja no meio de uma linha e não em seu final.

A função deste comando é avançar para a linha imediatamente após aquela em que ele ocorre.

## TAGS DE FORMATAÇÃO

Assim como em um editor de texto, em HTML você poderá utilizar efeitos em seu texto. Como já percebeu os comandos, tags, trabalham em pares, devendo ser iniciado e depois finalizado.

Os principais comandos de estilo são:

**TAG SINTAXE FUNÇÃO**

|  |  |  |
| --- | --- | --- |
| **STRONG** | <strong>texto</strong> | Similar ao negrito |
| **TYPERWRITER** | <tt>texto</tt> | Deixa o texto com espaçamento regular |
| **BIG** | <big>texto</big> | Aumenta a fonte e aplica negrito |
| **SMALL** | <small>texto</small> | Reduz e altera a fonte |
| **SOBRESCRITO** | <sup>texto>/sup> | Eleva o texto e diminui seu corpo |
| **SUBESCRITO** | <sub>texto</sub> | Rebaixa o texto e diminui seu corpo |
| **BLINK** | <blink>texto</blink> | Faz com que o texto pisque |
| **NEGRITO** | <b>texto</b> | Aplica o estilo negrito |
| **ITÁLICO** | <I>texto</I> | Aplica o estilo itálico |
| **SUBLINHADO** | <u>texto</u> | Aplica um sublinhado (em alguns browsers esta tag não funciona) |

### TEXTO PRÉ-FORMATADO

O texto de um arquivo em HTML é formatado através das tags. Mas você possui a opção da tag <PRE> </PRE> para estabelecer uma formatação através de um editor de texto e que esta formatação permaneça na sua página html.

O texto pré-formatado é excelente para apresentar código-fonte com seus espaçamentos adequados. Este recurso pode ser utilizado para criar tabelas mas não é recomendável.

Uma observação importante é que, ao usar tags de estilo ou vínculo, não tags de elemento, neste espaço elas irão funcionar normalmente.

SINTAXE:

<PRE>

TEXTO

TEXTO

TEXTO

</PRE>

### CITAÇÕES E COMENTÁRIOS

A tag <BLOCKQUOTE> é utilizada na criação de citações longas que não devem ser aninhadas em parágrafos.. A tag <CITE> destaca citações curtas.

Um exemplo:

<BLOCKQUOTE>

"O mercado de webdesign é um dos segmentos de design com maior potencial de crescimento para os próximos anos. Estima-se um crescimento de no mínimo 50% ao ano. E, na corrida rumo a esse novo eldorado, as agências de propaganda mais uma vez largaram na frente dos escritórios de design, investindo primeiro em estruturas especializadas para atender á demanda do mercado (...)".

Maria Edicy Moreira.

</BLOCKQUOTE>

Caso queira fazer comentários a respeito do seu código HTML apenas como referência pessoal e não para ser exibido em suas páginas você deverá fazer o comentário da seguinte forma:

**<!--comentário-->**

## FONTES

Para fazer alterações no tamanho, tipo e cor da fonte de seu texto você utilizará a tag <font> </font> que é do tipo contêiner (precisa ser iniciada e finalizada). Sua sintaxe é a seguinte:

<font size=" " face=" " color=" " align=” “> Texto </font>

### SIZE

O tamanho Size especifica o tamanho da fonte utilizada. Os valores permitidos vão de 1, o menor, a 7, o maior. O padrão é 3. Se for especificado um número antecedido com o sinal de adição ou subtração o valor da fonte será acrescido ou subtraído nesse valor. Vamos a um exemplo: se eu estiver trabalhando com uma fonte 5 e for especificado <font size=+2, o texto passará a ser exibido em 7. Se não houver sinal a fonte será exibida naquele valor.

### FACE

Esse parâmetro permite que seja escolhida uma fonte diferente para o texto. Podem ser especificadas várias fontes, de maneira que, se o sistema não possuir uma determinada fonte ele irá utilizar outra.

**Face="name [,name2[,name3]]**

Se não forem encontradas as fontes especificadas, então, o texto será apresentado na fonte padrão.

**<font face="Agaramond, Arial" size=1>**

### COLOR

Este parâmetro especifica a cor do texto. Seu valor será especificado no formato hexadecimal, RGB ou através de um nome predefinido de cores.

Algumas cores:

|  |  |  |  |
| --- | --- | --- | --- |
| COR | CÓDIGO | COR | CÓDIGO |
| Aqua | #70DB93 | Blue | #0000FF |
| Black | #000000 | Cyan | #00FFFF |
| Fuchsia | #FF00FF | Gray | #C0C0C0 |
| Green | #00FF00 | Lime | #32CD32 |
| Brown | #800000 | Navy | #23238E |
| Olive | #808000 | Red | #FF0000 |
| Silver | #E6E8FA | Teal | #008080 |
| White | #FFFFFF | Yellow | #FFFF00 |

Vamos a um exemplo de código:

<html>

<head>

</title>Experiência com Fontes</title>

</head>

<body>

<h2>Primeiro exemplo com tipos de fontes</h2>

<font face="Agaramond"><p>

Este texto será exibido em Agaramond</font><br>

<font face="Tahoma"><p>

Este texto será exibido em Tahoma</font><br>

<font face="Arcane"><p>

Este texto será exibido em Aracane</font><br>

<p>

<h2>Exemplos com tamanhos diferentes de fontes</h2>

<font face="Courier" Size=1>

Este texto será exibido em Courier</font><br><p>

<font face="Courier" Size=2>

Este texto será exibido em Courier</font><br><p>

<font face="Courier" Size=3>

Este texto será exibido em Courier</font><br><p>

<font face="Courier" Size=4>

Este texto será exibido em Courier</font><br><p>

<font face="Courier" Size=5>

Este texto será exibido em Courier</font><br><p>

<font face="Courier" Size=6>

Este texto será exibido em Courier</font><br><p>

<font face="Courier" Size=7>

Este texto será exibido em Courier</font><br><p>

<p>

<h2>Agora iremos usar cores</h2>

<font Face="Verdana" size=5 color="#800000">

Este é o primeiro texto a ser modificado</font>

</body>

</html>

### ALINHAMENTO

Pense no que é alinhar um texto no Word. Pois então você sabe o que é alinhar um texto. É claro que você não irá alinhar em HTML da mesma forma, terá que utilizar tags e parâmetros para estas tags.

Para alinhar um texto você utilizará o atributo ALIGN com valores : LEFT (esquerda), RIGHT (direita), CENTER (centralizado) e JUSTIFY (justificado).

### PARÁGRAFOS

O comando <p> inicia um novo parágrafo onde o texto é automaticamente alinhado pela margem esquerda da tela. Caso você deseje que seu novo parágrafo seja iniciado com um alinhamento diferente você deverá especificar na hora que incluir esta tag o atributo align.

Vejamos um exemplo:

<html>

<head>

<title>Alinhamento</title>

</head>

<body>

<h1>MACROMEDIA FLASH</H1>

<P ALIGN=JUSTIFY>

A tecnologia Shockwave Flash, da Macromedia, já está presente na Web há vários anos. Cada vez mais utilizado, este programa está sendo empregado em diversos sites. Seu poder se deve a sua grande capacidade multimídia e á sua versatilidade. Sua finalidade é a criação de imagens vetoriais com animação, som e interatividade. Mas recentemente seu domínio esta sendo ameaçado pelo novo padrão HTML 5, que permite fazer tudo isto, sem a necessidade de instalação de um plugin.</p>

<p align=center>

Depois veremos mais sobre o Flash</p>

</body>

</html>

### ALINHAMENTO COM O COMANDO <DIV>

A utilização desta tag é uma forma prática e rápida de alinhar bloco de texto. A abreviatura DIV significa divisão. Ao usar esta tag num bloco de texto você estará atingindo todas as tags dentro daquele bloco. Suas vantagens sobre o Align é que:

* Precisará ser utilizada apenas uma vez, ao contrário do atributo Align, que tem de ser incluído em diversas tags.
* A tag <DIV> pode ser usada para alinhar qualquer elemento (cabeçalho, parágrafos, citações, imagens, tabelas, etc.). O atributo Align encontra-se disponível apenas em um número limitado de tags.

Você deverá colocar a tag <DIV> e acrescentar o atributo align na tag de abertura. O atributo align poderá ter os valores left, right e center.

Sintaxe:

<DIV ALIGN=" ">

TEXTO e OUTRAS TAGS

TEXTO e OUTRAS TAGS

</DIV>

### TAG CENTER

Além do atributo align=”center” temos a tag <center> </center> que funciona igualmente ao atributo.

Vamos treinar um pouco com este código-fonte:

**<html>**

**<head>**

**<title>**Utilizando a tag DIV para alinhar**</title>**

**</head>**

**<body>**

**<h2>**JavaScript na World Wide Web**</h2>**

**<p align=justify>**

A World Wide Web iniciou-se como um simples repositório de informações, mas cresceu muito além disso. A medida que a Web desenvolveu-se, as ferramentas também desenvolveram-se. Ferramentas simples de marcação como HTML foram unidas por linguagens de programação de verdade. Incluindo **<b>**JavaScript**</b>**, que na verdade não é uma linguagem de programação mas uma **<b>**linguagem Script hospedeira**</b>** de páginas em HTML.**<br>**

**<p>**

**<div align=center>**

**<h2>**Entendendo a utilidade do JavaScript**</h2>**

Como a HTML é uma linguagem apenas de marcação de texto simples, ela não pode responder para o usuário, tomar decisões, nem autorizar tarefas repetitivas. Tarefas interativas como essas exigem uma linguagem mais complexa.

**<h3>**Eu precisarei usar JavaScript em todas as minhas páginas**</h3>**

Apenas se quiser aprimorá-las e interagir com o usuário.

**</div>**

**</body>**

**</html>**

### CARACTERES ESPECIAIS

Há caracteres que não podem ser obtidos por meio do teclado. Eles podem ser inseridos em sua página através da especificação de um código especial que o browser interpreta e substitui por um caracter específico. Esses códigos podem ser obtidos utilizando-se uma nomeclatura definida pela ISO -LATIN - 1 ou a partir de uma entidade HTML.

A norma ISO utiliza um código composto pelo caractere & seguido do símbolo # e, então, de uma combinação numérica com ponto-e-vírgula. Já a entidade HTML usa o caractere & seguido de uma palavra que identifica o símbolo com ponto-e-vírgula.

Caso você queira escrever uma página HTML em outro idioma terá que usar bastante esses caracteres especiais.

Imagine que você queira fazer uma página HTML ensinando a outras pessoas a linguagem HTML. Como faria para explicar uma tag sem que o browser interpretasse como um comando. Terá de utilizar estes símbolos especiais.

Vamos a um exemplo:

**<html>**

**<head>**

**<title>**Tutorial em HTML**</title>**

**</head>**

**<body>**

**<h2>**O comando **&lt**BR**&gt</h2>**

A finalidade deste comando HTML é marcar uma seção do texto como parágrafo inserindo uma quebra de linha.**<br>**

**<p>**

Já o comando &ltP&gt é para indicar início de uma novo parágrafo.

**</body>**

**</html>**

Observe que para que o browser pudesse mostrar <br> e <p> sem interpretar como comandos usei caracteres especiais &lt para indicar sinal de < e &gt para indicar o sinal de >. Quando você utiliza a acentuação comum do teclado nas suas páginas podem até parecer que tudo irá correr tudo bem. Mas alguém que visualizar suas páginas em um computador que não tenha as mesmas configurações de idioma, alguns caracteres podem não sair da forma desejada.

Por exemplo:

A palavra café, se você utilizar esta palavra com a acentuação do seu teclado, com certeza na visualização do seu browser ficará perfeita, mas pode ser que em outros computadores com a configuração diferente não apareça da mesma forma. O ideal é colocá-la assim:

Caf&eacute; ou ca&#233;

E se desejasse colocar a palavra entre aspas:

&quot;Caf&eacute;&quot

**ALGUNS CARACTERES ESPECIAIS**

|  |  |  |  |
| --- | --- | --- | --- |
| **CARACTER ESPECIAL** | **SÍMBOLO** | **CARACTER ESPECIAL** | **SÍMBOLO** |
| < | &lt | à | &agrave; |
| > | &gt | â | &acirc; |
| { | &#123; | ã | &atilde; |
| } | &#125; | ü | &uuml; |
| : | &#58; | & | &#38; |
| ; | &#59; | ® | &#174; |
| = | &#61; | " | &#34; |
| á | &aacute; | " | &quot; |
| @ | &#64; | ª | &ordf; |
| $ | &#36; | º | &ordm; |
| % | &#37; | ? | &#63; |

## LISTAS

Vamos colocar assim, há basicamente dois tipos de listas: as não ordenadas, que contêm uma série de itens sem numerá-los, e as listas ordenadas, que atribuem um número para cada elemento da lista. Trabalhar com listas é simples.

Trabalharemos com os comandos <ul> para listas não ordenadas e <ol> para as listas ordenadas.

### LISTAS NÃO ORDENADAS

Para criar uma lista não ordenada, você deve usar um conjunto de comandos cuja sintaxe básica é mostrada a seguir. O comando que gera a lista não ordenada é o comando <UL> (unordered list), que deve envolver o primeiro e o último item da lista. Cada item da lista deve ser precedido do comando <LI> (line list).

**<UL>**

**<LI> Texto**

**<LI> Texto**

**</UL>**

### LISTAS ORDENADAS

Para criar uma lista básica não ordenada, você deve usar um conjunto de comandos cuja sintaxe básica é mostrada a seguir. O comando que gera a lista não ordenada (ordered list) é o comando <OL>, que deve envolver o primeiro e o último item da lista. Cada item da lista deve ser precedido do comando <LI> (Line item).

**<OL>**

**<LI> Texto**

**<LI> Texto**

**</OL>**

OBSERVAÇÃO:

Na lista não ordenada cada item receberá um marcador ("bolinha") e a numerada receberá automaticamente números.

Você poderá personalizar sua lista ordenada com o atributo TYPE e START. O atributo Type pode assumir cinco valores para definir o tipo de numeração a ser usado na lista.

* "1" : Especifica que os algarismos arábicos padrão devem ser usados para numerar a lista ( 1,2,3,4, ...).
* "a" : Especifica que as letras minúsculas devem ser usadas para numerar a lista (a, b, c, d, etc.)
* "A" : Especifica que as letras maiúsculas devem ser usadas para numerar a lista (A, B, C, D, etc.)
* "i" : Especifica que os algarismos romanos minúsculos devem ser usados para numerar a lista ( i, ii, iii, iv, etc.)
* "I" : Especifica que os algarismos romanos maiúsculos devem ser usados para numerar a lista ( I, II, III, IV, etc.).

Exemplo:

**<p>Ingredientes para bolo</p>**

**<ol>**

**<li>Farinha de Trigo**

**<li>Açucar**

**<li>Manteiga**

**<li>Ovos**

**<li>Leite**

**<li>Fermento**

**</ol>**

Agora um exemplo com o atributo Type:

**<p>Ingredientes para bolo</p>**

**<ol type="I">**

**<li>Farinha de trigo**

**<li>Açucar**

**<li>Manteiga**

**<li>Ovos**

**<li>Leite**

**<li>Fermento**

**</ol>**

Usando o atributo START você poderá determinar o número ou a letra que inicia sua lista. O ponto inicial é 1. Usando o Start, você pudará esse número. Um exemplo: **<ol Type=a start=3>**.

Usando o atributo VALUE em um item da lista poderemos alterar os valores da lista apartir de qualquer ponto.

Exemplo:

**<UL>**

**<LI>**

**<LI value=10>**

**<LI>**

**</UL>**

## IMAGENS

Para inserir uma imagem em uma página web utilizamos do comando, tag, <img src> e seus atributos.

**<img src="endereço da imagem no HD">**

exemplo: <img src="c:\site\imagem.gif">

Poderemos acrescentar a esta imagem os seguintes atributos:

**Width="valor em pixel"** largura da imagem

**Height="valor em pixel"** altura da imagem

**Align=(left, right ou center)** alinhamento da imagem

**Alt="texto"** texto que irá aparecer ao passar o mouse sobre a imagem ou texto que surgirá caso a imagem não possa ser visualizada.

**Border="valor em pixel"** especificação da largura da borda da imagem.

**Vspace="valor em pixel"** para especificar o espaço que deve ser deixado acima e abaixo da imagem.

**Hspace="valor em pixel"** especifica o espaço que deve ser deixado nas laterais da imagem.

O único obrigatório é o src.

### Colocando Uma Imagem Como Fundo Da Página

**<body background="endereço da imagem no HD">**

ex.: <body background="c:\site\imagem.jpg">

Procure usar imagens simples, com poucas variações de tons.

Para fixar a imagem de fundo (imagem não rola junto com a tela):

**<body background="c:\site\imagem.jpg" bgproperties="fixed">**

### Colocando Uma Cor Como Fundo Da Página:

**<body bgcolor="cor">**

Procure não usar cores chamativas. Para que a cor do fundo da sua página chame mais atenção que seu conteúdo.

## LINKS E ANCORAS

### LINKS

A principal atração da Internet é a criação de documentos com o conceito de hipertexto, ou seja, um documento que se vincula a outros documentos por meio de ligações especiais chamadas links ou hiperlinks. Com esse conceito, você pode criar documentos que façam referências e permitam ao usuário acessar tais referências não importando se elas estão em outra página Web, no seu micro ou em algum servidor de rede.

O uso de hipertexto também facilita a criação de documentos extensos cujas seções ou tópicos podem ser rapidamente acessados por meio dos links. Imagine que você crie uma página sobre determinado assunto em vários tópicos abordados. Assim como em um livro você cria todo o texto do documento e depois cria, no início, um sumário indicando os tópicos do documento. Se quiser acessar diretamente o texto de um tópico, basta dar um clique sobre o item do tópico no sumário e ele será imediatamente exibido.

Um link é reconhecido em uma página por estar em cor normalmente diferente do resto do texto e, ao passar com o mouse sobre esse texto o cursor muda para uma mão apontando para o link.

Para criar um link para uma página:

**<a href="endereço da página">texto que será o link</a>**

exemplo: <a href="c:\site\casa.html">Minha casa</a>

Usando uma imagem como link:

Exemplo: **<a href="c:\site\casa.html><img src="c:\site\imagem.jpg"></a>**

### Âncoras

Uma âncora é um ponto de referência ou endereço que será acessado por um link, é usada dentro do documento para marcar o início de uma seção do documento.

Suponha que seu texto seja muito grande, o que tornaria trabalhoso para quem estiver vendo a página conseguir se movimentar entre um tópico e outro. Então faça o seguinte: Nomeie um pedaço da sua página através da tag:

**<A NAME="NOME "> Texto </A>**

( Lembre –se que "NOME" é fictício, você escolhe o nome…)

Depois disso escreva o pedaço que você quer associar a esse Nome. Agora crie um link para chegar até esse pedaço, usando o tag:

**<A HREF="#NOME">Clique Aqui</A>**

Pronto! Ao clicar na mensagem "Clique Aqui" de sua página, o usuário irá até o pedaço que você nomeou.

ATENÇÃO : O sinal de cerquilha (#) é necessário para a âncora, pois avisa ao browser para procurar o link no documento atual!

### Linkando Arquivos de Outros Servidores

Agora a coisa fica um pouco mais séria. É aqui que está toda graça de Internet. Através de um esquema de endereçamento, a URL, a Internet consegue acessar um arquivo que está em qualquer micro deste planeta desde que esteja conectado adequadamente a Web.

Como já vimos, a URL serve para especificar a localização de páginas e arquivos em diretórios de servidores da Web. Vamos comparar com um esquema que você usa (as vezes até sem saber) para acessar algum arquivo em seu próprio computador .

Acessando um arquivo pelo DOS :

***C:\WINDOWS\SYSTEM\Lista.TXT***

Acessando uma página pela Web :

***HTTP : //WWW.ASPMASTERS.CJB.NET***

Uma URL é composta por duas partes principais. A primeira é o protocolo Internet do documento, a segunda parte é o endereço do servidor e da página.

A sintaxe disso é:

**PROTOCOLO://SERVIDOR.INSTITUIÇÃO/ARQUIVO**

Onde:

* Protocolo é o tipo de servidor que está sendo acessado;
* Servidor é o computador que contém a página;
* Instituição é o tipo de instituição (.com.br) a qual esse computador pertence (Comercial, Internet Publica, Militar, etc. );
* Arquivo é o caminho do arquivo.

Exemplo:

**HTTP://WWW.TIMASTER.COM.BR**

Esse endereço pode ser especificado diretamente na linha de URL do Browser, para acessar diretamente a página INDEX. HTM, ou então dentro de um link de um documento HTML, através do A HREF que cria uma ligação, um link para o URL .

**<A HREF = "HTTP://WWW.TIMASTER.COM.BR "> TIMASTER </A>**

## TABELAS

As tabelas são muito importantes para o *designer* de uma home-page. Com elas pode-se fazer alinhamentos que dificilmente seriam possíveis com simples comandos. A funcionalidade de uma tabela faz com que um determinado site tenha um aspecto mais profissional e o usuário possa navegar de forma mais eficiente, pois os objetos podem estar melhor posicionados na home-page.

Vamos discutir, nesta parte, quais são os comandos para se criar uma tabela e mostrar alguns exemplos. Claro que você deverá praticar e principalmente planejar a sua tabela, pois sem isso, fica muito difícil de controlar o código HTML da sua home-page.

O comando para se inserir uma tabela é **<TABLE>,** para iniciar uma linha devemos introduzir a tag **<TR>** e para uma célula (alguns preferem dizer coluna) **<TD>**. Todos estes comandos são encerrados como **</TABLE>**, **</TR>** e **</TD>** respectivamente.

Vejamos um exemplo. Vamos supor que queremos montar uma tabela 3X2, ou seja, de 3 colunas por 2 linhas, o código HTML para isso é :

<HTML>

<HEAD>

<TITLE>Criando Tabelas</TITLE>

</HEAD>

<BODY>

<CENTER><FONT COLOR=”BLUE” SIZE=”6”>TABELA</FONT></CENTER>

<BR>

<TABLE BORDER=”1”> <!-- Inicia a tabela e coloca uma borda de espessura igual a 1-->

<TR> <!-- Cria a primeira linha da tabela-->

<TD>PRIMEIRA COLUNA </TD> <!-- Aqui foi criada uma célula-->

<TD>SEGUNDA COLUNA </TD>

<TD>TERCEIRA COLUNA </TD>

</TR> <!-- Fecha a primeira linha da tabela-->

<TR> <!-- Abre a segunda linha da tabela-->

<TD> PRIMEIRA COLUNA</TD>

<TD>SEGUNDA COLUNA </TD>

<TD>TERCEIRA COLUNA </TD>

<TR> <!-- Encerra a Segunda linha da tabela-->

</TABLE> <!-- Encerra a tabela-->

</BODY>

</HTML>

Obs: Todas as tags que iniciam com <!-- são apenas comentários.

Salve este exemplo como TABELA.HTM e abra o arquivo em um Browser para ver como ficou.

Com isto você já consegue Ter uma ideia de como montar suas tabelas. O segredo para se conseguir fazer estas tabelas de forma eficiente é planejar e ter a ideia exata do conteúdo desejado.

Nas células da tabela você pode inserir desde simples frases até figuras inteiras. Parâmetros podem ser acrescidos às tabelas, tais como: aumentar ou diminuir a largura de uma célula, distanciar as células uma das outras, inserir cor de fundo dentro da célula, etc.

Vamos procurar trabalhar com vários exemplos para visualizarmos melhor o uso destes parâmetros.

### ALTERANDO A LARGURA DA CÉLULA

Para alterar o largura de uma célula da tabela basta acrescentar o parâmetro **WIDTH** dentro da tag **<TD>**. Exemplo:

<TABLE BORDER=”2”>

<TR>

<TD WIDTH=”100”> WIDTH=100</TD>

<TD ALIGN=”MIDDLE” WIDTH=”200”>WIDTH=200 (PIXELS)</TD>

</TR>

</TABLE>

O parâmetro **WIDTH** pode receber dois tipos de valores, em pixels como no exemplo acima ou em porcentagem em relação ao tamanho da tela, como mostrado no exemplo a seguir. Também dentro da tag **<TD>** temos outro parâmetro que é **VALIGN=”**MIDDLE”, este parâmetro tem por função alinhar o conteúdo da célula no meio da mesma, ou seja, dentro da célula as frases são alinhadas com relação ao seu centro.

Outro exemplo:

<TABLE BORDER=”2”>

<TR>

<TD WIDTH=”25%”> WIDTH=25%</TD>

<TD ALIGN=”MIDDLE” WIDTH=”75%”>WIDTH=75%</TD>

</TR>

### INSERINDO COR DE FUNDO E SEPARAÇÃO DE CÉLULAS

Outro atributo que podemos ter nas tabelas é mudar a sua cor de fundo, isto se torna particularmente útil quando se quer dar destaque a uma célula em especial.

Vejamos um exemplo:

<TABLE BORDER=”0”>

<TR>

<TD WIDTH=”150” BGCOLOR=”RED”>VERMELHO </TD>

<TD WIDTH=”150” BGCOLOR=”BLUE”>AZUL </TD>

<TD WIDTH=”150” BGCOLOR=”RED”>VEMELHO</TD>

</TR>

<TR>

<TD WIDTH=”150” BGCOLOR=”BLUE”>AZUL</TD>

<TD WIDTH=”150” BGCOLOR=”RED”>VEMELHO</TD>

<TD WIDTH=”150” BGCOLOR=”BLUE”> AZUL</TD>

</TR>

</TABLE>

Aqui, vimos que podemos acrescentar cores ao fundo de uma célula, outra coisa que também pode ser feita é mudar a fonte com os comandos de formatação que você já conhece, como **<FONT COLOR=>,** por exemplo.

Repare que no exemplo anterior nós colocamos **BORDER=”0”** e as células apareceram coloridas, mas com um certo espaçamento entre elas, para tirarmos este espaço devemos acrescentar dentro da tag **<TABLE>** o parâmetro **CELLSPACING=”0”**. Este parâmetro elimina por completo os espaços entre as células.

<TABLE BORDER=”0” CELLSPACING=”0”>

<TR>

<TD WIDTH=”150” BGCOLOR=”RED”>VERMELHO </TD>

<TD WIDTH=”150” BGCOLOR=”BLUE”>AZUL </TD>

<TD WIDTH=”150” BGCOLOR=”RED”>VEMELHO</TD>

</TR>

<TR>

<TD WIDTH=”150” BGCOLOR=”BLUE”>AZUL</TD>

<TD WIDTH=”150” BGCOLOR=”RED”>VEMELHO</TD>

<TD WIDTH=”150” BGCOLOR=”BLUE”> AZUL</TD>

</TR>

</TABLE>

Outro parâmetro que pode ser usado para separar espaços em uma célula é **CELLPADDING** que inclui ou exclui espaços **dentro da célula**, ou seja, se você tiver um elemento (frase, texto, figura etc.) dentro da célula você pode acrescentar espaços por igual nos quatros cantos da célula, fazendo com que seu texto ou figura não fique "grudado" nas paredes da célula. Aproveite os exemplos anteriores e experimente usar este atributo para ver como ele funciona.

Isto já dá uma ideia do funcionamento geral das tabelas na WEB. Muitos dos alinhamentos que encontramos na Internet são feito por intermédio de tabelas.

Você também pode usar todos os outros recursos já vistos de HTML para otimizar o uso de tabelas.

### UM POUCO MAIS DE TABELAS

Agora que você já está um pouco mais familiarizado com o uso de tabelas e com seus principais atributos, vamos ver como podemos aperfeiçoar o uso de tabelas.

Em seguida descreveremos uma série de outros atributos e comandos para tabelas:

**<TH> ... </TH>**

Esta tag insere um cabeçalho dentro da célula. Na verdade a única diferença entre a tag **<TH>** e a tag **<TD>** é que a **<TH>** exibe o conteúdo da célula em negrito.

**<CAPTION> ... </CAPTION>**

Exibe um texto centralizado em relação à tabela, como se fosse uma legenda.

Exemplo:

<TABLE BORDER=”1” CELLSPACING=”2” CELLPADDING=”6”> <!-- Esta linha inicia a tabela com borda de espessura 1, espaço entre as células 2 e espaço dentro da célula 6 -->

<CAPTION>LEGENDA</CAPTION> <! Introduz uma legenda na tabela>

<TR>

<TH>Conteúdo em negrito</TH> <! Insere o texto na forma de cabeçalho (negrito)>

</TR>

<TR>

<TH>Conteúdo em negrito</TH>

</TR>

</TABLE>

Repare neste exemplo, que a legenda já sai centralizada com relação a tabela e repare no espaço dentro da célula devido ao comando CELLPADDING.

### OS ATRIBUTOS COLSPAN E ROWSPAN

Estes atributos são muito importantes dentro da tag **<TABLE>**. São eles que nos possibilitam remodelar a disposição das células dentro da tabela. Vejamos um exemplo prático:

<TABLE BORDER=”2” CELLPADDING=”2”>

<TR>

<TD COLSPAN=”2”>Exemplo do uso do COLSPAN</TD>

<TR>

<TD>CÉLULA 1</TD> <TD>CÉLULA 2</TD>

</TR>

</TABLE>

Neste exemplo vemos que na tag **<TD>** foi introduzido o parâmetro **COLSPAN**. Este parâmetro tem a finalidade de dizer que sob a célula onde ele foi introduzido podemos ter um certo número de colunas. Logo, se você quiser que uma célula ocupe uma única linha, como no nosso exemplo, você deve inserir o parâmetro dentro da tag **<TD>** desta célula. A quantidade de células que virão em baixo da célula que recebeu o atributo **COLSPAN** deve ser de acordo com o número especificado no parâmetro.

Mas se quisermos definir uma única célula ocupando uma coluna inteira? Neste caso usamos o atributo **ROWSPAN**. Exemplo:

<TABLE BORDER=”2” CELLPADDING=”2”>

<TR>

<TD ROWSPAN=”2”>Exemplo do uso do ROWSPAN</TD>

<TD>CÉLULA 1</TD></TR><TR> <TD>CÉLULA 2</TD>

</TR>

</TABLE>

Para melhor utilizar estes parâmetros e ver como eles funcionam você deve fazer a sua própria tabela, pois somente desta forma você conseguirá fixar o uso destes parâmetros.

### USANDO TABELAS COMO MOLDURAS

Outro uso bastante interessante de tabelas é como moldura para alguma figura. Para isto basta você iniciar uma tabela e definir um valor para BORDER.

Exemplo:

<TABLE BORDER=”5”>

<TR>

<TD> <IMG SRC=”*NOME DA FIGURA”*> </TD>

</TR>

</TABLE>

Claro que você deve usar com muito bom senso este artifício da tag **<TABLE>**, pois nem sempre uma moldura em uma figura define um bom layout (aparência) da sua home-page.

### COLOCANDO UMA IMAGEM DE FUNDO NAS TABELAS

Um recurso parecido com a cor de fundo, colocar uma imagem como fundo de uma tabela é uma técnica muito usada. Através do parâmetro **BACKGROUND** dentro da tag **TABLE** , basta indicar qual o endereço e o nome da figura.

<TABLE BACKGROUND="POOL.GIF" BORDER=”5”>

<TR>

<TD><H1>Que tal um mergulho agora ? </H1></TD>

</TR>

</TABLE>

As tabelas são um forte recurso de modelagem das home-pages, e para que você consiga utilizar bem este recuso não deve se limitar somente aos exemplos desta apostila, procure em sites e até mesmos em livros por exemplos mais complexos, você pode treinar também desenvolvendo tabelas mais complexas.

## FRAMES

Um site com frames é dividido em no mínimo 3 páginas html para cada exibição, se dividirmos a tela em um cabeçalho com um menu superior por exemplo, teríamos a página que contem este menu, outra página com o conteúdo da opção selecionada, e uma terceira página que faz o controle do frame.

Fazer frames tem o mesmo procedimento que confecção de home-pages simples, a diferença é que se deve trocar a tag <BODY> pela tag <FRAMESET>

Um documento HTML simples seria assim:

<HTML>

<HEAD>

<TITLE>Página simples </TITLE>

</HEAD>

<BODY>

*Neste campo entram os comandos em geral*

</BODY>

</HTML>

Já um documento HTML que teria frames ficaria assim:

<HTML>

<HEAD>

<TITLE>Página com Frame </TITLE>

</HEAD>

<FRAMESET>

*NESTE CAMPO ENTRAM AS CHAMADAS PARA OS DOCUMENTOS DOS FRAMES*

</FRAMESET>

</HTML>

Vamos criar um exemplo:

<HTML>

<HEAD>

<TITLE> Exemplo de frame</TITLE>

</HEAD>

<FRAMESET COLS=”20% , 80%”> *<! Este comando inicia o frame e divide a tela do navegador em 2 partes, uma com 20% da tela e outra com 80%*>

<FRAME SRC=”FRAME1.HTML”> <! *Este comando chama as páginas HTML que devem ocupar as colunas divididas pelo frame*>

<FRAME SRC=”FRAME2.HTML”>

</FRAMESET>

</HTML>

Este é o código da página principal, ela vai gerar os frames e carregar as respectivas páginas. Agora vamos crias os códigos para serem inseridos nas duas colunas que criamos.

<HTML>

<HEAD>

<TITLE>Página simples </TITLE>

</HEAD>

<BODY>

AQUI ENTRA A PRIMEIRA PÁGINA NORMAL

</BODY>

</HTML>

Salve este exemplo como **FRAME1.HTML**. Por quê? Porque no código que geramos no exemplo de frames chamamos a página com o comando **<FRAME SRC=FRAME1.HTML>,** se você salvar o nome do código de forma diferente deverá alterar também o código da página de frame.

<HTML>

<HEAD>

<TITLE>Página simples </TITLE>

</HEAD>

<BODY>

AQUI ENTRA A SEGUNDA PÁGINA NORMAL

</BODY>

</HTML>

Este código deve ser salvo como **FRAME2.HTML**, pelo mesmo motivo do código anterior. Sempre lembrando que estas páginas devem estar na mesma pasta. Pronto, você conseguiu fazer a sua primeira página com frame.

Além de poder dividir o navegador em colunas com o parâmetro **COLS** dentro da tag **<FRAMESET>** , você também pode dividir em linhas usando o parâmetro ROWS, ou ainda dividindo em colunas e linhas ao mesmo tempo.

<HTML>

<HEAD>

<TITLE> Exemplo de frame</TITLE>

</HEAD>

<FRAMESET ROWS=”50% , 50%”> <! *Este comando inicia o frame e divide a tela do navegador em 2 partes, uma com 50% da tela e outra com 50%*>

<FRAME SRC=”FRAME1.HTML”> <! *Este comando chama as páginas HTML que devem ocupar as colunas divididas pelo frame*>

<FRAME SRC=FRAME2.HTM>

</FRAMESET>

</HTML>

Uma tag **<FRAMESET>** pode ser inserida dentro da outra gerando assim frames dentro de frames, como no código abaixo:

<HTML>

<HEAD>

<TITLE> Exemplo de frame</TITLE>

</HEAD>

<FRAMESET COLS=50%,50%>

<FRAMESET ROWS=50%,50%>

<FRAME SRC=”FRAME1.HTML”>

<FRAME SRC=”FRAME2.HTML”>

</FRAMESET>

<FRAMESET ROWS=50%,50%>

<FRAME SRC=”FRAME1.HTML”>

<FRAME SRC=”FRAME1.HTML”>

</FRAMESET>

</FRAMESET>

</HTML>

Vejamos alguns dos parâmetros que podem ser atribuídos para a tag **FRAMES:**

### Parâmetro SRC da tag <FRAME>

Este parâmetro é o responsável por chamar a página HTML que deve ser aberta (inserida) dentro de um frame. Em todos os exemplos até agora só usamos arquivos locais para abrir estas páginas em um frame, mas nada impede você de colocar no lugar uma URL completa. Por exemplo:

<FRAME SRC=”http://www.yahoo.com”>

Com este comando você será capaz de abrir um site qualquer dentro de seu frame .

### Parâmetro MARGINWIDTH

Define a margem entre as laterais do frame e seu conteúdo. Exemplo:

<FRAME SRC=”FRAME1.HTML” MARGINWIDTH=100>

Este comando cria uma margem de 100 pixels de cada lado do frame.

### Parâmetro MARGINHEIGHT

Da mesma forma que MARGINWIDTH, este parâmetro define uma distância de margem, que neste caso é a margem superior e inferior do frame. Exemplo:

<FRAME SRC=”FRAME1.HTML” MARGINHEIGHT=100>

Dá um espaço de 100 pixels entre as margens superior e inferior do frame.

### Parâmetro SCROLLING

Define se o frame terá ou não barras de rolagem. Por default quando o conteúdo de uma página excede o tamanho do frame as barras de rolagem são acrescentadas automaticamente. Os valores válidos para SCROLLING são YES, NO e AUTO. Exemplo:

<FRAME SRC=”FRAME1.HTML” SCROLLING=”YES”>

### Parâmetro NAME

Este é um dos principais parâmetros da tag **<FRAME>.** Ele serve, como era de se esperar, para definir o nome do frame. Ele deve ser usado em conjunto com o parâmetro TARGET.

### Parâmetro TARGET

Este é o mais importante de todos os parâmetros. Ele trabalha em conjunto com NAME e permite que coloquemos um link em um frame e o resultado, ou seja, a página linkada, aparece em outro frame.

Vejamos um exemplo de como podemos linkar um frame com outro. Vamos construir uma página que possua um menu de opções do lado direito, para isso podemos aproveitar os códigos de exemplo abordados anteriormente. Os códigos são:

*FRAME.HTML*

<HTML>

<HEAD>

<TITLE>FRAMES</TITLE>

</HEAD>

<FRAMESET COLS=25%,75%>

<FRAME SRC=”FRAME1.HTML” SCROLLING=”NO”>

<FRAME SRC=”FRAME2.HTML” NAME=”TESTE”>

</FRAMESET>

</HTML>

*FRAME1.HTML*

<HTML>

<HEAD>

<TITLE>FRAMES</TITLE>

</HEAD>

<BODY BGCOLOR=”YELLOW”>

<CENTER><H2>Escolha um Link</H2></CENTER>

<BR>

<A HREF=”http:\\www.amazon.com” TARGET=”TESTE”>Livraria Amazon</A>

<BR>

< A HREF=”http:\\www.yahoo.com” TARGET=”TESTE”>Pesquisador YAHOO</A>

<BR>

<A HREF=”FRAME2.HTML” TARGET=”TESTE”>Volta para Frame2</A>

</BODY>

</HTML>

*FRAME2.HTML*

<HTML>

<HEAD>

<TITLE>FRAMES</TITLE>

</HEAD>

<BODY>

<CENTER><H2>Os links ao lado deverão aparecer neste lado do Navegador</H2></CENTER>

<BR>

<CENTER>

<FONT SIZE=4>Este frame é o que foi nomeado como TESTE com o parâmetro NAME em Frame.HTML</FONT>

</CENTER>

</BODY>

</HTML>

## IFRAME

Parecido com o Frame, o IFrame carrega uma página html dentro de outra, o desenvolvedor determina uma janela dentro da página que ele esta e define qual pagina será carregada dentro desta janela.

<IFRAME name=”iframe0” src="iframe\_0.html" frameBorder=”0” width=”400” height=”150” scrolling=”auto”>

</IFRAME>

### NAME

É o nome da janela, ele será usado caso você queira criar links que abram dentro do iframe, é o valor do TARGET.

### SRC

É a página que será aberta dentro do iframe.

### FRAMEBORDER

Define a borda do frame.

### WIDTH e HEIGHT

Definem a largura e a altura do iframe, respectivamente.

### SCROLLING

Define se o frame terá barra de rolagem automática.

## FORMULÁRIOS

Formulários estão presentes na Internet para possibilitar cadastros, pesquisas, envio de comentários, aumentando o poder de interação com os visitantes dos sites. Um formulário HTML é uma pagina Web que contem, além de texto, elementos especiais chamados controles, representados por caixas de checagem, botões, caixas de seleção, campos de textos, etc.

Um formulário funciona assim: Os usuários preenchem os campos do formulário submetendo-o, em seguida, a algum agente de processamento. Neste momento, todas as informações fornecidas são enviadas a um programa escrito especialmente para processar esses dados de acordo com alguma necessidade especifica. Em alguns casos os dados são gravados em um Banco de Dados (BD), em outros casos uma nova página e construída, em outros ainda as informações são encaminhadas via e-mail.

A tag utilizada para criar formulários é **<FORM>** que deve ser finalizada com **</FORM>.** Os principais parâmetros desta tag são:

### NAME

Especifica o nome do formulário. Útil em casos de validação dos campos, por exemplo, quando se utiliza uma linguagem de scripts como JavaScript.

### METHOD

Método que define como os dados serão transmitidos para o programa que vai processa-los. Devem ter valores GET ou POST, sendo que a diferença entre estes dois valores esta no modo como os dados são enviados para o servidor. Normalmente o programa que será utilizado para processar o formulário já especifica o valor para o atributo METHOD.

### ACTION

Indica o endereço do programa ou página que receberá os dados do formulário.

Exemplo:

<FORM NAME=”maladireta” METHOD=”post” ACTION=”maladireta.asp”>

.

.

.

</FORM>

### Tag INPUT

Especifica um elemento (campo) de entrada de dados em um formulário. Exemplo:

<INPUT TYPE=”tipo” NAME=”nome” VALUE=”valor” SIZE=”tamanho em pixels” MAXLENGHT=”tamanho máximo em pixel”>

Alguns de seus parâmetros são:

**TYPE**: Define o tipo de elemento a ser inserido no formulário.

**NAME**: Nome do elemento, útil em casos de validação de campos, por exemplo, quando se usa uma linguagem de scripts como JavaScript.

**VALUE**: Valor que pode ser predefinido para o campo.

**MAXLENGHT**: Comprimento máximo do campo.

**CHECKED**: Em casos de caixas de checagem, predefine como checada.

#### CAIXA DE TEXTO

O valor TEXT no atributo TYPE de um INPUT indica que o campo será de texto, ou seja, um campo onde o usuário poderá entrar com dados.

Exemplo:

<p>Entre com seu nome:<INPUT TYPE=”TEXT” NAME=”Nome”></p>

#### VALUE

O atributo VALUE, neste caso, pode ser usado para determinar um valor prévio para o campo, que será exibido quando a pagina for carregada.

Exemplo:

<p>Entre com seu nome:<INPUT TYPE=”TEXT” NAME=”Nome” VALUE=”Maria”></p>

#### SIZE

O atributo SIZE configura o tamanho do campo e é baseado no numero de caracteres. Este valor não limita o tamanho do campo, somente define o tamanho em que será mostrado na pagina. Quem define o numero máximo de caracteres que podem ser digitados é o atributo MAXLENGHT, que é opcional.

Exemplo:

<p>Entre com seu nome:<INPUT TYPE=”TEXT” NAME=”Nome” VALUE=”Maria” SIZE=”20” MAXLENGHT=”30”></p>

#### FORMULÁRIO COM SENHA

Para o valor PASSWORD (senha) no atributo TYPE aplicam-se todos os atributos do tipo TEXT exceto que todas as letras digitadas aparem com um asterisco.

Exemplo:

<p>Digite sua senha:<INPUT TYPE=”PASSWORD” NAME=”Senha” SIZE=”8” MAXLENGHT=”8”></p>

#### CAIXA DE TEXTO COM VARIAS LINHAS

<TEXTAREA NAME=”nome” ROWS=”numero de linhas” COLS=”números de colunas”></TEXTAREA>

O marcador TEXTAREA permite definir um campo de texto com varias linhas.

**ROWS**: define o numero de linhas da caixa de texto.

**COLS**: define quantos caracteres (colunas) cada linha possui.

**NAME**: define o nome da caixa de texto.

Exemplo:

<p>Entre com seu comentário:</p>

<p><TEXTAREA NAME=”comentario” ROWS=”3” COLS=”20”></TEXTAREA> </p>

#### LISTA DE OPÇÕES

O marcador **SELECT** permite a criação de uma lista de opções a serem escolhidas pelo usuário.

**NAME**: nome da lista

**SIZE**: define o numero de opções a serem exibidas simultaneamente na tela. Se o valor for omitido ou igual a 1 e exibida uma opção por vez.

O marcador **OPTION** define cada opção a ser exibida pela lista. Este texto pode ter qualquer tamanho e deve vir entre os marcadores <OPTION></OPTION>.

O atributo **VALUE** ira determinar o valor de cada opção. Assim, se, por exemplo, o texto aparecer para o usuário for “Produtos destinados ao consumo imediato”, Não é isso que ficara armazenado quando o formulário for submetido e sim o VALUE associado a este texto, que pode ser uma abreviação como “PROD”.

Exemplo:

<SELECT NAME=”interesses” SIZE=”1”>

<OPTION VALUE=”vazio”>Escolha a área de interesse</OPTION>

<OPTION VALUE=”iniciant”>Curso de iniciantes</OPTION>

<OPTION VALUE=”formprof”>Formação profissional</OPTION>

<OPTION VALUE=”ctec”>Cursos Microsoft® CTEC</OPTION>

<OPTION VALUE=”todos”>Todos</OPTION>

</SELECT>

#### CAIXA DE LISTAGEM

Permite que na tela apareça uma lista com mais de uma opção, embora o usuário possa escolher somente uma. A criação de caixas de listagem é idêntica a criação de lista de opções, com exceção do atributo SIZE que deve ter um valor maior que 1, determinando o numero de opções a aparecerem na tela.

Exemplo:

<SELECT NAME=”interesses” SIZE=”5”>

<OPTION VALUE=”vazio”>Escolha a área de interesse</OPTION>

<OPTION VALUE=”iniciant”>Curso de iniciantes</OPTION>

<OPTION VALUE=”formprof”>Formação profissional</OPTION>

<OPTION VALUE=”ctec”>Cursos Microsoft® CTEC</OPTION>

<OPTION VALUE=”todos”>Todos</OPTION>

</SELECT>

#### CAIXA DE LISTAGEM COM MULTIPLA SELEÇÃO

Permite que na tela apareça uma lista com mais de uma opção, podendo o usuário escolher mais de uma. A criação de caixas de listagem com múltipla seleção é idêntica a criação de caixas de listagem sem múltiplas seleções com exceção do atributo MULTIPLE que deve ser acrescentado.

Exemplo:

<SELECT NAME=”interesses” SIZE=”5” MULTIPLE>

<OPTION VALUE=”vazio”>Escolha a área de interesse</OPTION>

<OPTION VALUE=”iniciant”>Curso de iniciantes</OPTION>

<OPTION VALUE=”formprof”>Formação profissional</OPTION>

<OPTION VALUE=”ctec”>Cursos Microsoft® CTEC</OPTION>

<OPTION VALUE=”todos”>Todos</OPTION>

</SELECT>

Como no Windows, nas caixas de listagem com múltipla seleção, o usuário pode escolher mais de uma opção utilizando as teclas Shift (quando as opções são continuas) ou Ctrl (quando as opções não são continuas).

#### CAIXAS DE CHECAGEM

O valor CHECKBOX no atributo TYPE define uma caixa de checagem. Na pratica, as caixas de checagem aparecem em grupos e os usuários podem escolher uma ou mais opções.

**NAME**: define o nome da caixa de checagem. No caso de varias caixas, cada uma deve ter um nome diferenciado.

**VALUE**: contem o valor do campo, que será passado ao programa interpretador do formulário.

**CHECKED**: atributo opcional que pode ser utilizado quando se deseja que a opção já apareça selecionada na pagina.

Exemplo:

<p>Escolha a área de interesse</p>

<INPUT TYPE=”CHECKBOX” NAME=”caixa1” VALUE=”iniciant”> Cursos para iniciantes<Br>

<INPUT TYPE=”CHECKBOX” NAME=”caixa2” VALUE=”formprof”> Cursos de formação profissional<Br>

<INPUT TYPE=”CHECKBOX” NAME=”caixa3” VALUE=”ctec”> Cursos da Microsoft® CTEC

#### BOTÃO LIMPA/CANCELA

O valor RESET no atributo TYPE define um botão que limpa todos os campos, devolvendo os mesmos valores de quando a pagina foi carregada. No atributo VALUE, pode-se definir o que será escrito no botão. Caso nenhum valor seja definido, aparecerá somente RESET.

Exemplo:

<p>Entre com seu nome: <INPUT TYPE=”TEXT”></p>

<INPUT TYPE=”RESET” VALUE=”Apagar”>

#### BOTÃO ENVIAR (SUBMIT)

O valor SUBMIT no atributo TYPE define um botão que aciona o envio das informações preenchidas no formulário ao programa interpretador. O atributo VALUE define o que será escrito no botão. Caso nenhum valor seja definido, aparecerá somente SUBMIT.

Exemplo:

<p>Entre com seu nome: <INPUT TYPE=”TEXT”></p>

<INPUT TYPE=”SUBMIT” VALUE=”Enviar”>

#### IMAGEM PARA O BOTÃO ENVIAR

Alternativa para o botão tipo SUBMIT, pode-se usar uma imagem que possua função de enviar o formulário para processamento. Ou seja, em vez de usar um botão comum usa-se uma imagem que pode ser um botão personalizado (colorido).

**NAME**: define o nome do controle.

**SRC**: define o endereço ou o nome do arquivo de imagem.

**ALT**: define o texto que será mostrado quando o mouse passar pela imagem.

Exemplo:

<p>Entre com seu nome: <INPUT TYPE=”TEXT”></p>

<INPUT TYPE=”IMAGE” SRC=”botenvia.gif” ALT=”Clique para enviar o formulário”>

#### CAMPOS ESCONDIDOS

O valor HIDDEN no atributo TYPE define dados que devem ser passados ao programa interpretador, embora não estejam visíveis na pagina.

**NAME**: identifica o dado

**VALUE**: define o valor que deve ser passado.

Geralmente os valores destes campos são atribuídos no momento em que esta sendo feita a consistência dos dados no formulário via uma linguagem de scripts, como JavaScript.

Exemplo:

<p>Entre com seu nome: <INPUT TYPE=”TEXT”></p>

<INPUT TYPE=”HIDDEN” NAME=”escolha” VALUE=”conteudo”>

<INPUT TYPE=”IMAGE” SRC=”botenvia.gif” ALT=”Clique para enviar o formulário”>

### PROCESSAMENTO DO FORMULÁRIO

De nada adianta construir um formulário e não ter como processa-lo. Existem varias tecnologias ligadas ao processamento de um formulário, das quais podemos citar: Active Server Pages ou ASP que é uma tecnologia da Microsoft ou PHP.

Neste caso o programador precisa inserir no código HTML da pagina, comandos que consigam acessar o servidor e executar as tarefas exigidas. A pagina que contem comandos ASP passa a ter extensão .asp e não mais .html (no caso do PHP a extensão passa a ser .PHP). Mas pelo fato dos scripts serem executados no servidor, o usuário não consegue ver nenhuma codificação em sua maquina. Quando utilizada esta tecnologia para processamento de um formulário, o atributo ACTION do marcador <FORM> indicara o arquivo .asp correspondente ao processamento do formulário.

Exemplo:

<FORM NAME=”maladireta” METHOD=”post” ACTION=”maladireta.asp”>