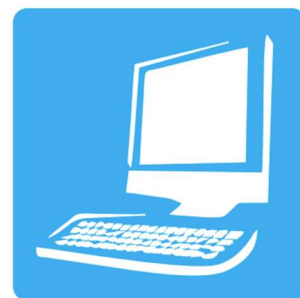


**ENSINO TÉCNICO
PROFISSIONALIZANTE**

CVT

CAMPINHO
CENTRO VOCACIONAL TECNOLÓGICO



Programador Web



GOVERNO DO ESTADO
RIO DE JANEIRO

FAETEC
Secretaria de
Ciência, Tecnologia
e Inovação



Como surgiu a WWW?

A web se originou no CERN (Conseil Européen pour la Recherche Nucleaire - Laboratório Europeu de Física de Partículas), em Genebra, Suíça. Os cientistas do CERN, como em várias outras instituições, necessitavam de uma forma de distribuir informações, como relatórios técnicos e resultados experimentais, para pesquisadores em várias localidades. A partir disso, foi criada a rede HEPNet (High-Energy Physics Network), usando-se o protocolo TCP/IP para conectar pesquisadores de Física da Europa e da América. O objetivo inicial, era de centralizar em uma única ferramenta, as várias tarefas necessárias para se obter as informações disponíveis na Internet.

Iniciado em 1989, através da proposição por Tim Bernes-Lee, cientista de Oxford, de um conjunto de protocolos baseado no TCP/IP para um novo sistema de informação da Internet: a Web. Através desses protocolos, um pesquisador utilizaria um programa, chamado de browser ou navegador, para visualizar a informação armazenada em outro computador. O projeto só se consolidou em 1991, explodindo a partir de 1993, com a introdução da versão final do software Mosaic, um browser, para a comunidade Internet.

Aos poucos a Web cresceu e seus protocolos foram sendo adotados por outras organizações. Para controlá-los foi criado o W3C ou WWW Consortium, o qual é dirigido pelo MIT, CERN e INRIA (Institut National de Recherche en Informatique et en Automatique).

O que são as URLs - Uniform Resource Locator

A abreviação URL corresponde à nomenclatura Uniform Resource Locator (localizador universal de recursos), utilizada pela Internet para indicar o endereço de um documento.

Saber especificar corretamente uma URL é imprescindível para utilizar adequadamente muitos comandos da linguagem HTML.

Uma URL deve conter três partes:

1. o protocolo do documento,
2. o endereço do servidor e
3. a localização do arquivo.

A sua sintaxe é: protocolo://servidor/nome do arquivo.

Onde:

protocolo - É o tipo de comunicação que deve ser usado. Ele é o conjunto de regras que especificam o formato, a sincronização, o sequenciamento e a verificação de erros na comunicação entre computadores.

servidor - É o nome do servidor que será acessado.

nome do arquivo - É o nome do documento que será acessado.

Alguns dos protocolos que podem ser usados:

HTTP - É um servidor da World Wide Web que contém documentos no formato 'HTTP' e que significa HyperText Transfer Protocol.

GOPHER - É um servidor composto por menus e diretórios com informações sobre arquivos e dados.

FTP - É uma abreviação de File Transfer Protocol.

Telnet - Este protocolo inicia uma sessão para se conectar remotamente a outro computador.

WAIS - Wide Area Indexed Server é um local onde documentos estão disponíveis em formato especial de pesquisa.

File - Indica que o documento está no computador ou na rede local.

INTRODUÇÃO AO HTML

O que é HTML (HiperText Markup Language)

É uma linguagem de autoria com hipertexto, utilizada para a criação de páginas para a World Wide Web.

HTML utiliza marcações específicas e distintas para dizer ao Web browser como exibir o documento.

Características da linguagem HTML

A linguagem HTML (Hypertext Markup Language) tem o objetivo de formatar textos através de marcações especiais denominadas tags, para que possam ser exibidos de forma conveniente pelos clientes Web, também denominados navegadores ou browsers. Além disso, esta linguagem possibilita a interligação entre páginas da Web, criando assim documentos com o conceito de hipertexto.

- *Documentos HTML são arquivos de texto escritos em ASCII.*

O ASCII (American Standard Code for Information Interchange) é o formato mais comum usado em arquivos texto em computadores e na Internet. Os documentos HTML devem ter tamanho reduzido, a fim de economizar tempo na transmissão através da Internet, evitando longos períodos de espera e congestionamento da rede;

- *Podem ser criados em qualquer editor de texto (vi, emacs, edit, notepad)*
- *O HTML não faz diferença entre letras maiúsculas e minúsculas em suas marcações, ou seja, não é "case sensitive".*

Ele não faz diferença entre maiúsculas e minúsculas (não é "case sensitive"). Então a notação **<title>** é equivalente a **<TITLE>** ou **<TiTlE>**.

Porém, as notações **< CENTER>**, **<CEN TER>**, **</ CENTER>** estão sintaticamente incorretas.

- *Nem todas as marcações (TAGs) e seus correspondentes recursos são suportados por qualquer navegador. Quando um browser não entende uma marcação, ele simplesmente a ignora. Não retornando um erro ao operador.*

Há diferenças entre os diversos navegadores, de forma que nem todas as marcações e seus correspondentes recursos são suportadas por todos eles. Quando um browser não entende uma determinada marcação, ele simplesmente a ignora. Ao criar um documento, é importante testá-lo com vários browsers.

- *Arquivos HTML podem possuir as extensões 'html' ou 'htm'. A primeira é normalmente utilizada em sistemas LINUX e a segunda em sistemas Windows. Os Browsers são capazes de exibir documentos com ambas as extensões.*

- *Os espaços em branco entre as tags, não as influenciam.*

Exemplos:

a) `<center>Esta turma é muito boa</center>`

b) `<center>

Esta turma é muito boa

</center>`

c) `<center>

Esta turma e muito boa

</center>`

Todos os exemplos anteriores, são equivalentes, apenas que, quanto mais elegante a página for escrita, melhor será a sua manutenção.

Então, com o HTML teremos:

- **Flexibilidade:** O usuário deve ter a possibilidade de customizar diversos elementos do documento, como as cores, tamanho das letras, etc.
- **Portabilidade:** Ela fornece uma maneira de descrever a informação independente da plataforma de visualização. Os elementos de HTML descrevem o que é a informação, não a forma específica de como ela deve mostrada.
- **Simplicidade:** A linguagem HTML é simples e pode ser aprendida sem maiores dificuldades pelo usuário leigo, não sendo necessárias, na maioria dos casos, noções anteriores de programação.

Você pode visualizar um documento que esteja criando, com o Chrome (ou qualquer outro browser Web). Basta abri-lo com o comando Open Local File disponível na opção File do menu, ou simplesmente arrastando o arquivo .HTML para o navegador.

Detalhe importante:

- *As tags podem ser influenciados por outras tags! Elas devem seguir a lógica do último comando que foi aberto, e que deve ser o primeiro a ser fechado (chama-se LIFO last in, first out).*

Exemplos:

`<center>Esta turma é muito boa</center>` (texto centralizado e em negrito)

`<center>Esta turma é muito boa</center>` (não está correto!)

Tags ou marcações HTML

As marcações do HTML - **tags** - consistem do sinal (<), (o símbolo de "menor que"), seguida pelo nome da marcação e fechada por (>) ("maior que").

De um modo geral, as **tags** aparecem em pares (chamadas tags compostas), por exemplo, <H1> Cabeçalho </H1>. O símbolo que termina uma determinada marcação é igual aquele que a inicia, antecedido por uma barra (/) e precedido pelo texto referente.

No exemplo, <H1> avisa ao cliente Web para iniciar a formatação do primeiro nível de cabeçalho e </H1> avisa que o cabeçalho acabou.

Há exceções a esse funcionamento em pares das marcações (tags isoladas). Por exemplo, a que indica um final de parágrafo: <P>. Não necessita de uma correspondente </P>.

A marcação que indica quebra de linha -
 - também não precisa de uma correspondente, e outras tais como <HR> e .

Exemplos de marcações compostas e isoladas:

- A notação: <TITLE> e a correspondente de fechamento </TITLE>, especificam o título de um documento.
- A notação: <H1> (e a correspondente </H1>), indica a aparência de um cabeçalho.
- A notação: <P> é a marcação para final de parágrafo.
- A notação: é uma *tag* com atributo, que indica a inserção neste local da imagem *quadro.gif*.

Além do comando propriamente dito, uma tag pode conter seus parâmetros.

Os parâmetros são recursos próprios de cada marcação, que permitem que ele seja executado de diversas formas.

Pode-se dizer que enquanto um comando diz ao Browser o que fazer, seus parâmetros dizem a ele como fazer, através de suas opções. Assim como cada comando possui parâmetros específicos, cada parâmetro possui opções específicas.

Veja o exemplo:

```
<FONT SIZE="4"COLOR="green"> Texto </FONT>
```

No exemplo acima, 'FONT' é o comando e, 'SIZE' e 'COLOR' são os parâmetros cujas opções escolhidas pelo programador foram, respectivamente, '4' e 'green'.

Note que, ao contrário da marcação de abertura, a marcação de fechamento de um comando não precisa mencionar seus parâmetros.

Criando Documentos HTML*A estrutura de um documento HTML*

Uma página de um documento HTML deve ter alguns elementos básicos, facilmente identificáveis pelo usuário:

a) Cabeçalho (HEAD)

Onde está a identificação da página que aparecerá na aba do navegador.

b) Corpo (BODY)

Onde está o conteúdo e seus elementos de navegação.

O chamado HTML "Mínimo"

Todo documento HTML deve ser iniciado com uma declaração HTML (`<html> </html>`), deve ter uma área de cabeçalho (`<head></head>`) com o nome para o documento (`<title> </title>`), um título principal e uma área definida como corpo (`<body></body>`) do conteúdo do documento.

A área do cabeçalho, embora muito conveniente, é opcional e delimitada pelas marcações `<HEAD>` e `</HEAD>`. Entre estas tags pode ser definido o título da página, através das marcações `<TITLE>` e `</TITLE>`. O título especificado será exibido na barra de título do Browser.

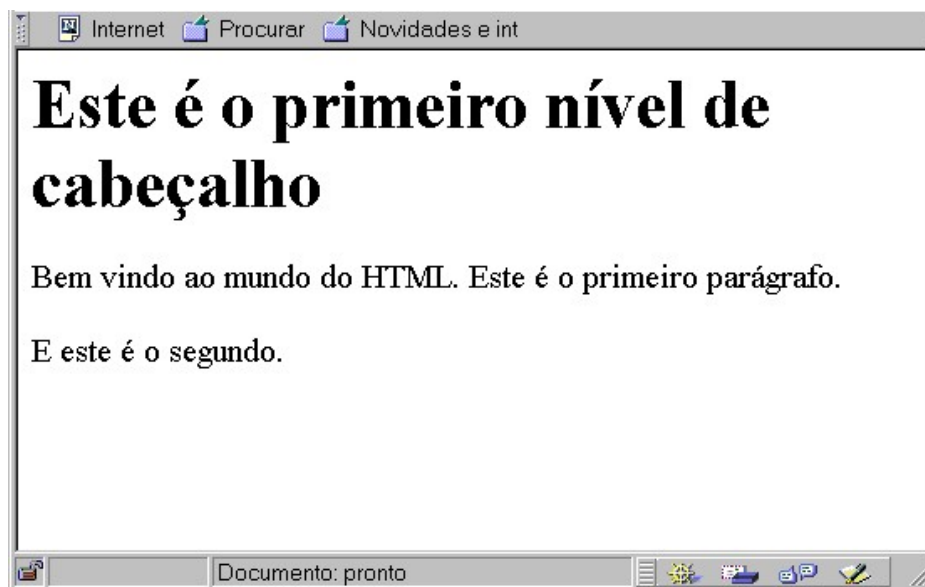
A maioria dos comandos HTML será colocada na área do corpo do programa, que é delimitada pelas marcações `<BODY>` e `</BODY>`.

Excetuando-se os "frames documents", que serão vistos mais adiante, esta é uma marcação obrigatória.

Veja no exemplo a seguir:

```
<HTML>
  <HEAD>
    <TITLE>HTML Mínimo </TITLE>
  </HEAD>
  <BODY>
    <H1>Este é o primeiro nível de cabeçalho</H1>
    Bem-vindo ao mundo do HTML.
    Este é o primeiro parágrafo. <P>
    E este é o segundo. <P>
  </BODY>
</HTML>
```

Esta página vai aparecer no navegador assim:



A estrutura das TAGs HEAD é a seguinte:

TÍTULOS - <TITLE></TITLE> - Ele que contém o título do documento, e não possui atributos. Servem também como uma forma de referenciar páginas em listas de links criados pelo usuário, tais como Bookmarks ou favoritos

Acostume-se que, todo documento em HTML deve possuir um título. De um modo geral o título aparece em um lugar separado no alto da tela do navegador, e é utilizado para identificar o documento (página) que está sendo apresentada. É interessante que o título possa sugerir claramente o conteúdo do documento.

Atenção; porque o conceito de título é diferente de cabeçalho. O título está mais para identificar o arquivo. Não é um elemento relevante na visualização do documento como acontece com o cabeçalho.

Códigos especiais de caracteres

Existem dois tipos de códigos especiais que permitem inserir qualquer caractere numa página HTML, mesmo que ele não esteja presente no teclado do computador.

Estes códigos são especificados pela norma ISO-Latin-1 ou a partir de uma identidade HTML.

A norma ISO utiliza um código composto pelo caractere '&' seguido do símbolo '#' e de uma combinação de números, sendo terminado com ';'. Já a identidade HTML usa o caractere '&' seguido de uma palavra que identifica o símbolo e de ';'. As identidades HTML correspondentes a alguns símbolos comuns, aparecerão em alguns exemplos desta apostila.

Criando, abrindo e manipulando documentos em HTML

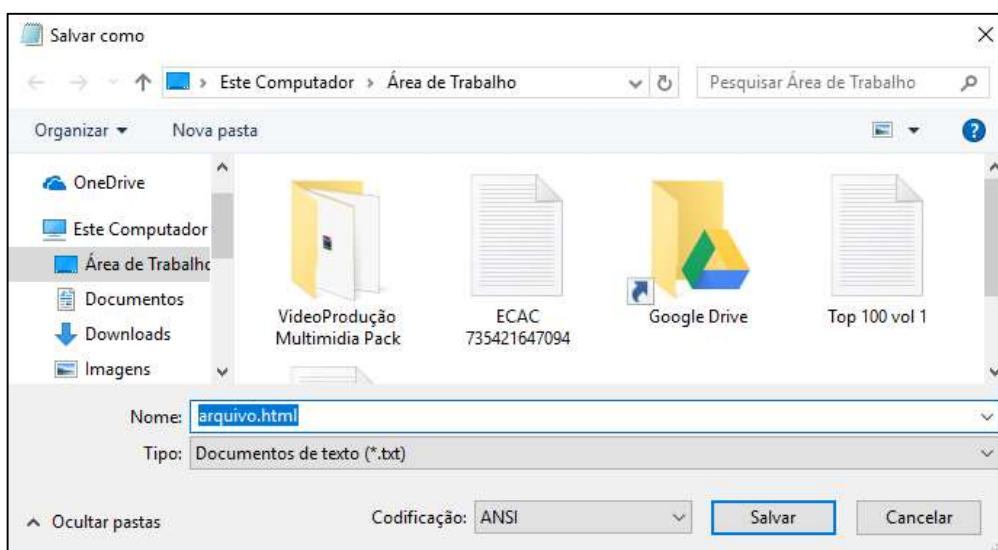
Documentos HTML podem ser criados em qualquer editor de texto visual (nos quais a visualização do arquivo não depende de compilação de código), tais como Bloco de Notas, VS, etc. Além dos editores específicos para várias plataformas, existem conversores de diversos formatos, por exemplo de *.doc para *.html.

Para criar uma página no Bloco de Notas, por exemplo, é preciso digitar o texto desejado juntamente com as marcações que o formatarão.

Um documento escrito desta forma é denominado código fonte.

Para praticar, abra o Bloco de Notas e digite o exemplo visto na seção anterior da apostila.

Na hora de salvá-lo, no campo 'Salvar como Tipo' da caixa de diálogo 'Salvar Como', escolha a opção 'Todos os Arquivos' e no campo 'Nome do Arquivo' digite o nome desejado juntamente com a extensão .html ou .htm:



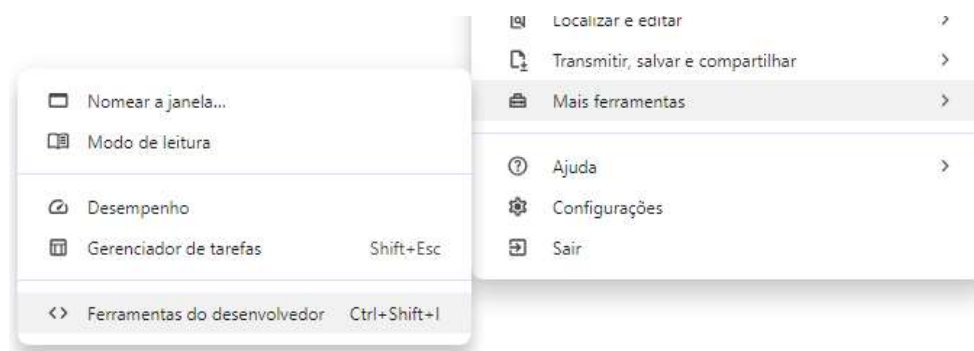
Fique atento! Se este procedimento não for adotado, o arquivo será gravado com a extensão padrão .txt e não poderá ser aberto por Browsers.

Depois de salvo o documento, é possível alterar seu código fonte clicando com o botão direito do mouse sobre o ícone da página, escolhendo a opção 'Abrir Com' e, em seguida, selecionando o programa editor de texto disponível. No caso do Windows, este programa será o Bloco de Notas.

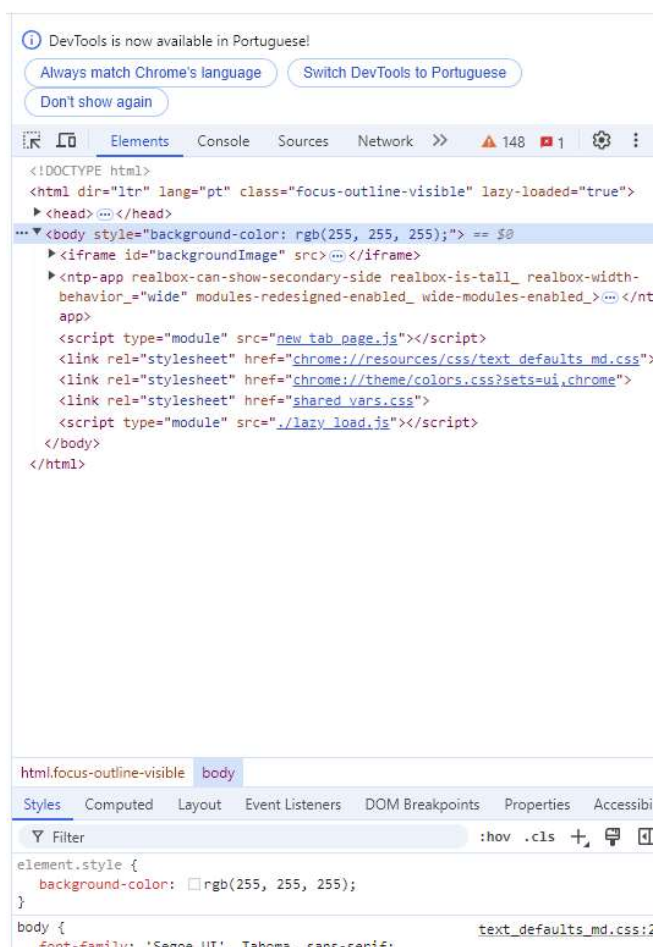
Para visualizar o arquivo criado no Google Chrome, por exemplo; no Gerenciador de Arquivos; clique duas vezes no ícone ao lado do nome do arquivo, que ele será aberto automaticamente no navegador padrão.

Uma dica interessante para quem está começando e para quem deseja cada vez mais conhecer os recursos da linguagem HTML é analisar códigos fonte de páginas da Web.

No Chrome, o usuário pode fazer isso clicando nos três pontinhos, no alto à direita do navegador, selecionando Mais ferramentas, <> Ferramentas do desenvolvedor.



Desta forma o código fonte da página será mostrado no lado direito do navegador:



Primeiros comandos em HTML

Parágrafos

A linguagem HTML não reconhece os caracteres de quebra de linha dos editores de texto, e mesmo que exista uma linha em branco no código fonte, os navegadores (browsers) só reconhecem o início de um novo parágrafo mediante à uma marcação apropriada.

A marcação `<P>` é utilizada para definir o início de um novo parágrafo, deixando uma linha em branco entre este e o texto anterior.

O parâmetro `ALIGN` define o alinhamento do texto do parágrafo e pode assumir os valores `left`, `right`, `center` e `justify`, correspondendo respectivamente aos alinhamentos à esquerda, à direita, ao centro e justificado. Se o parâmetro `ALIGN` não for especificado, por padrão o parágrafo será alinhado à esquerda.

Sintaxe básica: `<P>Texto</P>`

Sintaxe completa: `<P ALIGN="left | right | center | justify">Texto</P>`.

As barras verticais exibidas no exemplo de sintaxe acima, indicam apenas que as opções são exclusivas (únicas), não devendo ser usadas em seu código, e as aspas são opcionais neste caso.

A utilização de aspas na definição destes parâmetros só será obrigatória em duas situações: quando o valor do parâmetro for composto de duas ou mais palavras ou quando não for dado nenhum espaço entre os parâmetros na tag.

Este comando pode ou não ser terminado por `</P>`.

Quebras de linha

A marcação `
` faz uma quebra de linha sem acrescentar espaços. Não existe a marcação `</BR>`.

A diferença entre a aplicação de `<P>` e `
` pode ser visualizada claramente no exemplo abaixo.

Digite-o em seu computador e salve-o como `paragrafo.html`.

Veja o resultado.

```
<HTML>
  <HEAD>
    <TITLE> Parágrafos e quebras de linha </TITLE>
  </HEAD>
  <BODY>
    <P ALIGN=center>Utilizando P:</P>
```

```

<P> Vamos separar esta sentença com a marcação de parágrafo. </P>
Para verificar a diferença.
<P ALIGN=center>Utilizando BR:</P>
<P> Agora separamos as linhas utilizando <BR>
a marcação de quebra de linha. <BR>
Deu para notar?
</BODY>
</HTML>

```

Linhas horizontais

A tag <HR> (Horizontal Row) insere uma linha divisória horizontal na posição onde for colocado.

Não existe uma marcação </HR> para encerrá-la.

Os principais parâmetros deste comando são:

SIZE=número - Define a largura da linha e deve ser especificado em pixels.

WIDTH=número | número% - Especifica o comprimento da linha em pixels ou porcentagem em relação ao tamanho da janela.

ALIGN=left | center | right - Especifica o alinhamento horizontal da linha em relação às margens do navegador.

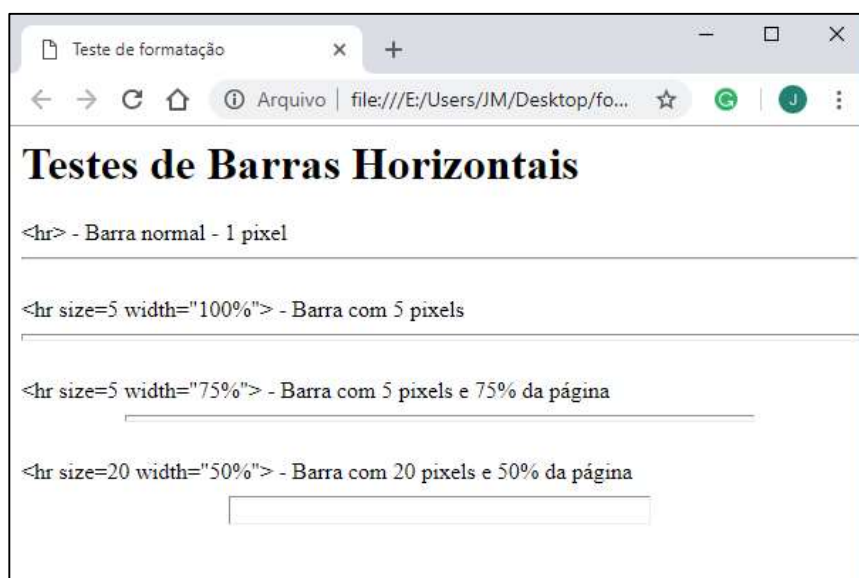
COLOR="cor" - Especifica a cor da linha.

NOSHADE - Indica que a linha não deve receber o efeito de sombreado.

Sintaxe básica: <HR>

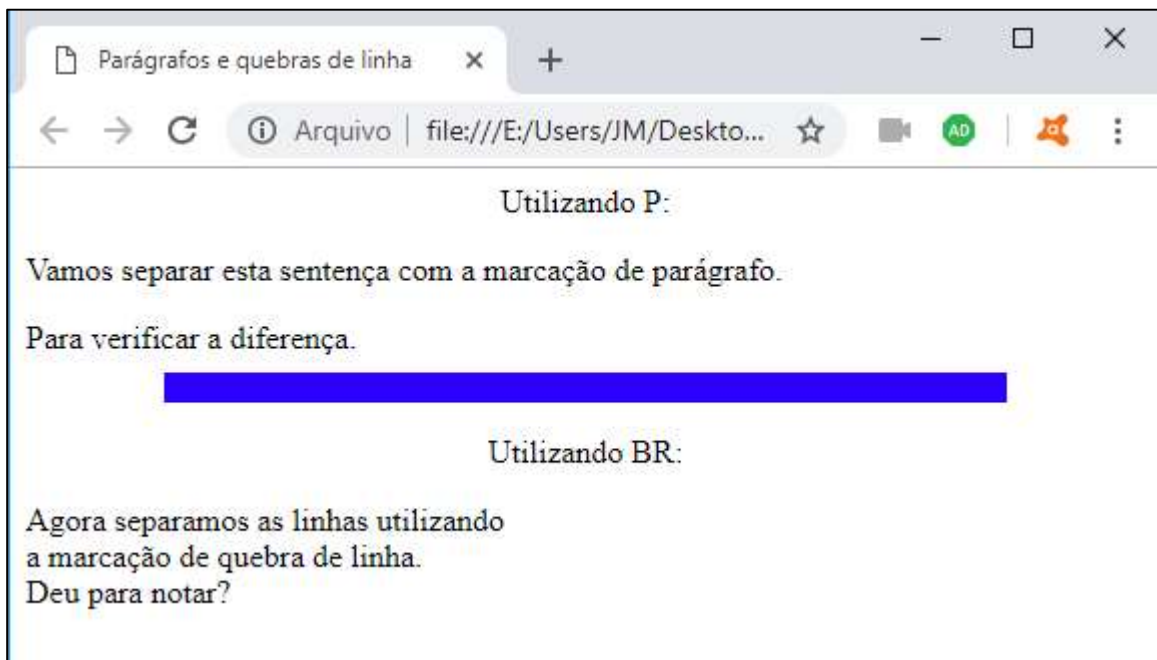
Sintaxe completa: <HR SIZE=número WIDTH=número | número% ALIGN="left | center | right" COLOR="cor" NOSHADE>

Crie o código para exemplificar a página abaixo:



Agora, modifique o exemplo paragrafo.html, conforme o indicado abaixo:

```
<HTML>
  <HEAD>
    <TITLE> Parágrafos e quebras de linha </TITLE>
  </HEAD>
  <BODY>
    <P ALIGN=center>Utilizando P:</P>
    <P> Vamos separar esta sentença com a marcação de parágrafo. </P>
    Para verificar a diferença.
    <HR ALIGN=center WIDTH=75% SIZE=15 COLOR="blue" NOSHADE>
    <P ALIGN=center>Utilizando BR:</P>
    <P> Agora separamos as linhas utilizando <BR>
    a marcação de quebra de linha. <BR>
    Deu para notar?
  </BODY>
</HTML>
```



Aplicando estilos de texto

Marcadores de estilo

Estes comandos permitem que o programador crie uma série de efeitos no texto, alterando a forma e o tamanho da fonte.

Todas as tags deste tipo devem ser especificadas em pares, marcando o início e o fim do texto a ser formatado.

Os principais comandos de estilo de texto são:

Negrito - Aplica o estilo negrito. Sintaxe: texto;

Itálico - Aplica o estilo itálico. Sintaxe: <I>texto</I>;

Sublinhado - Aplica o estilo sublinhado. Sintaxe: <U>texto</U>;

Strong - Similar ao negrito. Sintaxe: texto;

Typewriter - Deixa o texto com espaçamento regular. Sintaxe: <TT>texto</TT>;

Big - Aumenta a fonte e aplica o estilo negrito. Sintaxe: <BIG>texto</BIG>;

Small - Reduz a fonte. Sintaxe: <SMALL>texto</SMALL>;

Sobrescrito - Eleva o texto e diminui seu corpo. Sintaxe: ^{texto};

Subscrito - Rebaixa o texto e diminui seu corpo. Sintaxe: _{texto};

O exemplo a seguir mostra todos estes efeitos. Experimente!

Digite-o e salve-o como estilos.html.

```
<HTML>
  <HEAD>
    <TITLE> Estilos de texto </TITLE>
  </HEAD>
  <BODY>
    <B>Negrito</B>
    <BR><I>Itálico</I>
    <BR><U>Sublinhado</U>
    <BR><STRONG>Strong</STRONG>
    <BR><TT>Typewriter</TT>
    <BR><BIG>Big</BIG>
    <BR><SMALL>Small</SMALL>
    <BR>Texto <SUP>Sobrescrito</SUP>
    <BR>Texto <SUB>Subscrito</SUB>
  </BODY>
</HTML>
```

Cabeçalhos

Outra forma usada para mudar o tamanho da letra e ao mesmo tempo, aplicar o estilo negrito a um texto, é o uso das marcações do tipo <H>.

A linguagem HTML possui seis variações deste comando, de H1 a H6, que aplicam um tamanho de fonte diferenciado para o texto que venham a envolver.

Além de modificar o tamanho, os comandos H inserem linhas em branco acima e abaixo da linha de texto formatada.

O comando H1 possui o maior tamanho, enquanto que H6 possui o menor tamanho de fonte.

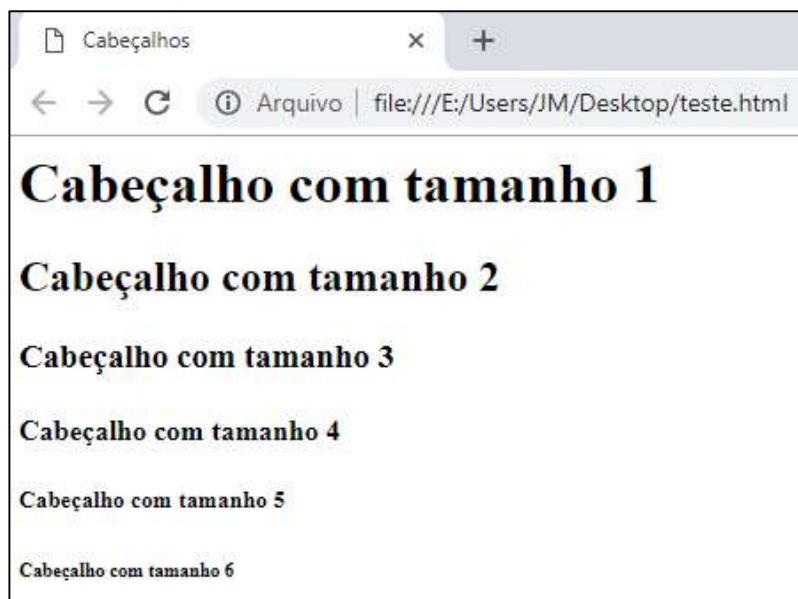
A sintaxe básica deste comando é: <Hn>Texto</Hn>.

Onde n é o número de 1 a 6 correspondente ao tamanho do cabeçalho.

O exemplo abaixo permite visualizar os tamanhos de cabeçalhos.

Digite-o salvando como cabecalho.html.

```
<HTML>
  <HEAD>
    <TITLE> Cabeçalhos </TITLE>
  </HEAD>
  <BODY>
    <H1> Cabeçalho com tamanho 1 </H1>
    <H2> Cabeçalho com tamanho 2 </H2>
    <H3> Cabeçalho com tamanho 3 </H3>
    <H4> Cabeçalho com tamanho 4 </H4>
    <H5> Cabeçalho com tamanho 5 </H5>
    <H6> Cabeçalho com tamanho 6 </H6>
  </BODY>
</HTML>
```



Exibição de textos pré-formatados

Através das tags <PRE> e </PRE> o programador pode incluir, em uma página, um texto que foi digitado num editor de textos qualquer, mantendo sua formatação original com marcas de tabulação, fim de linha gerado pela tecla Enter e outros formatos.

O principal parâmetro do comando PRE é o parâmetro WIDTH, que especifica, em caracteres, a largura da linha do texto.

Sintaxe básica: <PRE>Texto</PRE>.

Sintaxe completa: <PRE WIDTH=largura>Texto</PRE>.

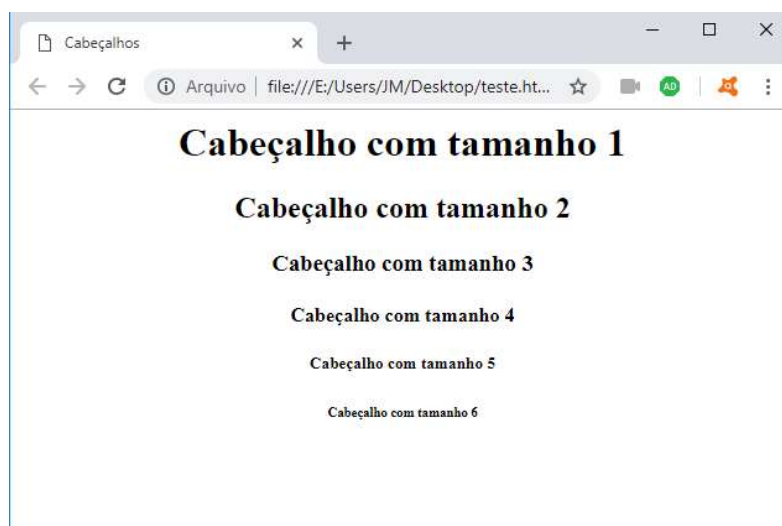
Centralização de textos e outros elementos

Para centralizar um cabeçalho, parágrafo ou figura dentro da largura da página, devem ser usadas as marcações <CENTER> e </CENTER> envolvendo o elemento a ser centralizado.

Sintaxe: <CENTER> [Elementos a serem centralizados] </CENTER>.

Modifique o exemplo 'cabecalho.html' conforme mostrado abaixo.

```
<HTML>
  <HEAD>
    <TITLE> Cabeçalhos </TITLE>
  </HEAD>
  <BODY>
    <CENTER>
      <H1> Cabeçalho com tamanho 1 </H1>
      <H2> Cabeçalho com tamanho 2 </H2>
      <H3> Cabeçalho com tamanho 3 </H3>
      <H4> Cabeçalho com tamanho 4 </H4>
      <H5> Cabeçalho com tamanho 5 </H5>
      <H6> Cabeçalho com tamanho 6 </H6>
    </CENTER>
  </BODY>
</HTML>
```



O marcador FONT

As tags e permitem ao programador modificar o tamanho, o tipo e a cor da fonte de um texto.

Seus parâmetros são:

SIZE=número - Especifica o tamanho da fonte a ser utilizada e pode variar de 1 a 7. Caso seja especificado um número antecedido por sinais de adição ou subtração, o tamanho padrão da fonte (tamanho 3) será, respectivamente, aumentado ou diminuído daquele valor. Assim size=2 e size=-1 têm o mesmo efeito prático.

FACE="nome" - Permite que seja escolhida uma fonte diferente para o texto. Podem ser especificados vários tipos de fonte, de modo que, se o sistema não possuir a primeira opção, a segunda é carregada automaticamente e assim por diante. Pode-se, portanto, escrever: FACE="fonte1,fonte2, fonte3". Neste caso, fonte1 é a fonte preferida pelo programador, fonte2 é sua segunda opção e fonte3 a última.

COLOR="cor" - Especifica a cor do texto. Seu valor pode ser especificado pelo nome da cor em inglês, no caso de cores mais comuns. Porém, no caso geral, a especificação de cores é feita através do padrão RGB (Red-Green-Blue).

Neste padrão a cor é descrita por um número de 6 dígitos, onde os dois primeiros correspondem à intensidade da cor vermelha, os dois do meio à da cor verde e os dois últimos à da cor azul. A cor resultante será uma combinação destas três. A base numérica utilizada para a representação das cores no padrão RGB é a hexadecimal, o que nos fornece 256 possibilidades de intensidade para cada cor básica (de 0 a 255).

Escreve-se: COLOR= "rrggb" ou COLOR= "nome". Os padrões RGB de algumas cores podem ser facilmente encontrados, pesquisando-se na internet.

Sintaxe básica: Texto.

Sintaxe completa: Texto.

Digite o exemplo abaixo e salve-o como cores.html.

```
<HTML>
  <HEAD><TITLE> Cores de texto </TITLE></HEAD>
  <BODY>
    <FONT SIZE=7 COLOR="0C01BD">F</FONT>
    <FONT SIZE=6 COLOR="655E7D">a</FONT>
    <FONT SIZE=5 COLOR="A39A4B">e</FONT>
    <FONT SIZE=4 COLOR="EEF303">t</FONT>
    <FONT SIZE=4 COLOR="EEF303">E</FONT>
    <FONT SIZE=5 COLOR="A39A4B">c</FONT>
    <FONT SIZE=6 COLOR="655E7D">R</FONT>
    <FONT SIZE=7 COLOR="0C01BD">J</FONT>
  </BODY>
</HTML>
```

Cores de texto e padrões de fundo

Através da manipulação dos parâmetros da marcação <BODY>, é possível definir cores para o fundo da página, para o texto e para os links de acordo com a preferência do programador.

Também é possível inserir imagens como plano de fundo, porém isto será tópico de um de nossos próximos capítulos.

Em todos os parâmetros do comando BODY, a cor deve ser definida da mesma forma que no parâmetro COLOR do comando FONT, ou seja, pelo seu nome ou padrão RGB.

Os principais parâmetros do comando BODY utilizados para alterar cores de texto e fundo de uma página são:

BGCOLOR="cor" - É a cor de fundo, onde o padrão é cinza ou branco, dependendo do Browser.

TEXT="cor" - É a cor do texto, sendo que o padrão é preto.

LINK="cor" - É a cor dos links, sendo que o padrão é azul.

VLINK="cor" - É a cor dos links já visitados, onde o padrão é vermelho-púrpuro.

Assim, poderíamos configurar as cores de uma página complementando a marcação <BODY> da seguinte forma: <BODY BGCOLOR="cor" TEXT="cor" LINK="cor" VLINK="cor">[Conteúdo da página] </BODY>

Deve-se ressaltar que o programador precisa ter cuidado ao escolher as cores para que não atrapalhem a leitura da informação contida na página.

Modificações nas cores dos links também podem vir a confundir o usuário, portanto, se quiser alterá-las, tome cuidado para não inverter o padrão ou escolher cores iguais às de texto ou fundo.

Modifique o exemplo 'cores.html' conforme mostrado abaixo:

```
<HTML>
<HEAD><TITLE> Cores de texto e fundo </TITLE></HEAD>
<BODY BGCOLOR="black">
  <FONT SIZE=7 COLOR="0C01BD">F</FONT>
  <FONT SIZE=6 COLOR="655E7D">a</FONT>
  <FONT SIZE=5 COLOR="A39A4B">e</FONT>
  <FONT SIZE=4 COLOR="EEF303">t</FONT>
  <FONT SIZE=4 COLOR="EEF303">E</FONT>
  <FONT SIZE=5 COLOR="A39A4B">c</FONT>
  <FONT SIZE=6 COLOR="655E7D">R</FONT>
  <FONT SIZE=7 COLOR="0C01BD">J</FONT>
  <P>
  <A HREF="">link</A> com cor modificada
</BODY>
</HTML>
```

Deve-se observar que a tag <A> utilizada no exemplo acima será definida mais adiante; portanto, deixe-a vazia (aspas-aspas).

Imagens

De modo geral, para que seja possível inserir uma imagem em uma página HTML, ela deve estar disponível preferencialmente nos formatos GIF ou JPG.

Estes formatos são aceitos por qualquer Browser, além de possibilitar a compactação dos arquivos de imagem, permitindo o carregamento mais rápido da página.

Certos navegadores conseguem exibir imagens de outros formatos, como BMP, JIFF, PNG, por exemplo. Porém, tendo em vista os aspectos de universalidade e eficiência, os formatos GIF e JPG continuam sendo os mais recomendados.

Você pode obter uma imagem através de páginas na Internet, CD/DVDs de imagens ou qualquer processo de digitalização. Neste último caso, é provável que se precise utilizar programas de tratamento de imagens, tais como o Paint ou o PhotoShop para realizar uma conversão de formatos, tamanhos, cortes e qualidades.

É importante ressaltar que algumas imagens encontradas em páginas da Web podem estar protegidas por direitos autorais. Portanto, antes de colocarem sua página uma figura interessante obtida em algum site, procure checar se ele faz alguma restrição quanto à utilização de suas imagens.

O marcador IMG

O marcador IMG insere uma imagem no corpo do documento. Seus principais parâmetros são:

SRC= “nome ou URL do arquivo” - Nome ou endereço (URL) da imagem a ser exibida. Este parâmetro é obrigatório.

ALT=“Texto” - Exibe o texto especificado quando o Browser não encontra a imagem ou quando o cursor passa por cima da figura. Nos ajuda muito, em caso de erros na página.

ALIGN= “Top | Middle | Bottom” - Especifica o alinhamento da imagem relativamente à linha de texto onde é exibida.

ALIGN= “Left | Right” - Especifica o alinhamento da imagem relativamente às bordas laterais da janela.

ISMAP - Determina que a figura é um mapa de imagem. Trataremos deste assunto com mais detalhes em uma de nossas próximas aulas.

WIDTH= número | número% - Especifica a largura de exibição da imagem em pixels ou em relação à largura da janela, independentemente de seu tamanho original. Quando se altera a dimensão horizontal da imagem, a dimensão vertical é modificada automaticamente para que a proporção original seja mantida.

HEIGHT= número | número% - Especifica a altura de exibição da imagem em pixels ou em relação à largura da janela, independentemente de seu tamanho original.

Quando se altera a dimensão vertical da imagem, a dimensão horizontal é modificada automaticamente para que a proporção original seja mantida.

Se quisermos forçar uma distorção na proporção original da imagem, temos que usar os parâmetros **WIDTH** e **HEIGHT** simultaneamente.

BORDER= número - Especifica, em pixels, a largura da borda da imagem. O valor zero remove a borda.

VSPACE=número - Determina, em pixels, o espaço que deve ser deixado em branco na parte de cima e na parte de baixo da imagem.

HSPACE=número - Determina, em pixels, o espaço que deve ser deixado em branco nas laterais da imagem.

Sintaxe básica: .

Sintaxe completa: .

Inserindo imagens do mesmo diretório

Se a imagem e o documento HTML que a exibirá estiverem no mesmo diretório, basta especificar o nome completo da imagem no parâmetro SRC do comando IMG.

Sintaxe básica: .

Digite o exemplo abaixo salvando-o como 'imagens1.html'. Para que ele funcione, é necessário que você escolha uma imagem qualquer e a salve com o nome de imagem1.gif no mesmo diretório do documento.

```
<HTML>
  <HEAD>
    <TITLE>Imagens 1 - Inserção de imagens</TITLE>
  </HEAD>
  <BODY>
    <CENTER>
      <H1>Inserindo imagens do mesmo diretório</H1>
      <BR>
      <IMG SRC="imagem1.gif">
    </CENTER>
  </BODY>
</HTML>
```

Inserindo imagens de diretórios diferentes

Se a imagem e o documento HTML que a exibirá estiverem em diretórios diferentes, o programador precisará especificar o caminho relativo para a imagem a partir do documento corrente, no parâmetro SRC do comando IMG.

O browser sempre assume o diretório atual como o inicial para procurar um arquivo especificado.

Para especificar uma imagem que esteja em um diretório abaixo do diretório atual, basta utilizar a sintaxe:

```
<IMG SRC="Diretório/nome do arquivo">.
```

Por outro lado, se o diretório onde se encontra a imagem estiver situado acima do diretório atual, a sintaxe utilizada será:

```
<IMG SRC="../nome do arquivo">.
```

Portanto, o símbolo ../ corresponde à passagem para um nível superior numa árvore de diretórios e o símbolo / corresponde a uma passagem para um nível inferior.

Utilizando estes símbolos, você poderá incluir em uma página um arquivo qualquer de imagem existente no computador.

Observe os exemplos de sintaxes:

 - Esta sintaxe insere no documento o arquivo da imagem 'imagem.jpg', que está salva no diretório 'Dir2', abaixo do diretório 'Dir1', que por sua vez encontra-se abaixo do diretório atual.

 - Esta sintaxe insere no documento o arquivo da imagem 'imagem.jpg', que está salva no terceiro diretório, acima do diretório corrente.

 - Esta sintaxe insere no documento o arquivo da imagem 'imagem.jpg', que está salva no diretório Dir3, abaixo do segundo diretório acima do diretório corrente.

Modifique o exemplo imagens1.html como mostrado a seguir.

Para que o código funcione, você deverá escolher uma imagem qualquer e salvá-la como imagem2.gif no diretório imediatamente acima do diretório atual. Salve também outra imagem como imagem3.html numa pasta do diretório atual chamada Subdir.

Lembre-se que o diretório atual será aquele que contém o documento imagens1.html.

```

<HTML>
  <HEAD>
    <TITLE> Imagens 1 - Inserção de imagens </TITLE>
  </HEAD>
  <BODY>
    <CENTER>
      <H1> Inserindo imagens do mesmo diretório </H1>
      <BR>
      <IMG SRC="imagem1.gif">
      <P>
      <H1> Inserindo imagens de outros diretórios </H1>
      <P>
      <H2> Diretório acima do diretório atual </H2>
      <BR>
      <IMG SRC=" ../imagem2.gif">
      <P>
      <H2> Diretório abaixo do diretório atual </H2>
      <BR>
      <IMG SRC="Subdir/imagem3.gif">
    </CENTER>
  </BODY>
</HTML>

```

Inserindo imagens de outros servidores

Para inserir em uma página HTML imagens localizadas em outro servidor (externo), é necessário apenas especificar corretamente sua URL no parâmetro SRC do comando IMG.

Sintaxe básica: .

Veja o exemplo:

```
<IMG SRC= "http://www.nomeservidor.com.br/imagem.gif">
```

A sintaxe do exemplo, insere no documento a imagem imagem.gif, localizada no servidor www.nomeservidor.com.br.

Nesse caso, o protocolo utilizado é o http, e sua declaração completa é necessária para a sintaxe correta da URL, como vimos no início da apostila.

Incrementalmente mais uma vez o exemplo 'imagens1.html', conforme mostrado abaixo:

```

<HTML>
  <HEAD><TITLE> Imagens 1 - Inserção de imagens </TITLE>
</HEAD>
  <BODY>
    <CENTER>
      <H1> Inserindo imagens do mesmo diretório
    </H1>
    <BR>

```

```
<IMG SRC="imagem1.gif">
<P>
<H1> Inserindo imagens de outros diretórios
</H1>
<P>
<H2> Diretório acima do diretório atual </H2>
<BR>
<IMG SRC="../imagem2.gif">
<P>
<H2> Diretório abaixo do diretório atual </H2>
<BR>
<IMG SRC="subdir/imagem3.gif">
<P>
<H1> Inserindo imagens de outros servidores </H1>
<BR>
<IMG
SRC="http://www.telecom.u_.br/pet/imgcomunicacao.jpg" >
</CENTER>
</BODY>
</HTML>
```

Alinhamento de imagens

Para alinhar uma imagem numa página é utilizado o parâmetro ALIGN.

As opções Top, Middle e Bottom do parâmetro ALIGN definem a posição da imagem com relação ao texto. Se for usada Top, a figura será alinhada com o texto pela sua parte superior, se for usada Middle, pelo centro e, se for usada Bottom, pela parte inferior.

Estas três opções mantêm a imagem como sendo um caractere de linha, ou seja, de cada lado da figura só poderá haver uma linha de texto.

Por outro lado, as opções Right e Left definem a posição da imagem na página. A opção Left determina que a imagem seja posicionada junto à borda esquerda da janela e a opção Right faz com que a imagem fique junto à borda direita.

Essas duas opções fazem com que a imagem não mais seja encarada como um caractere de linha, mas sim como um objeto a parte, e isso permite que o texto flua ao lado da figura.

Digite o exemplo abaixo e salve-o como imagens2.html na mesma pasta onde se encontra imagens1.html:

```
<HTML>
<HEAD>
<TITLE> Imagens 1 - Alinhamento de imagens </TITLE>
</HEAD>
<BODY>
<H1> Alinhamento de imagens </H1>
<P>
```

```
Alinhamento pelo <IMG SRC="imagem1.gif" ALIGN=top > topo
<P>
Alinhamento pelo <IMG SRC="imagem1.gif" ALIGN=bottom > fundo
<P>
Alinhamento pelo <IMG SRC="imagem1.gif" ALIGN=middle > meio
<P>
<IMG SRC="imagem1.gif" ALIGN=right> Note que, com este tipo de alinhamento, o
texto flui ao lado da figura naturalmente. Isto ocorre porque, quando alinhamos a figura
pelas laterais, ela passa a não ser encarada pelo Browser como um caractere de linha.
<P>
<IMG SRC="imagem1.gif" ALIGN=left> Note que, com este tipo de alinhamento, o
texto flui ao lado da figura naturalmente. Isto ocorre porque, quando alinhamos a figura
pelas laterais, ela passa a não ser encarada pelo Browser como um caractere de linha.
</BODY>
</HTML>
```

Backgrounds com imagens – Fundos de Página

Ao invés de definir uma cor para enfeitar o fundo de sua página, o programador pode optar por utilizar imagens.

Para fazer isto basta substituir o parâmetro BGCOLOR do comando BODY pelo parâmetro BACKGROUND.

Em BACKGROUND deverá ser especificado o nome ou a URL do arquivo de imagem desejado, lembrando que este deverá estar, preferencialmente, nos formatos GIF ou JPG.

Sintaxe: <BODY BACKGROUND= "nome ou URL do arquivo"> [Conteúdo da página]</BODY>

As técnicas estudadas anteriormente para inserir imagens que não estejam contidas no mesmo diretório do documento continuam válidas.

Observe os exemplos:

<BODY BACKGROUND= "../imagem.jpg"> [Conteúdo da página]</BODY>- Esta sintaxe insere, como plano de fundo do documento, a imagem 'imagem.jpg' localizada no diretório acima do diretório corrente.

<BODY BACKGROUND= _Dir1/imagem.jpg_> [Conteúdo da página]</BODY> - Esta sintaxe insere, como plano de fundo do documento, a imagem 'imagem.jpg' localizada no diretório Dir1 abaixo do diretório corrente.

<BODY BACKGROUND= "http://www.meuservidor.com.br/imagem.gif">[Conteúdo da página] </BODY> - Esta sintaxe insere, como plano de fundo do documento, a imagem imagem.jpg localizada no servidor www.meuservidor.com.br.

É necessário esclarecer que para preencher todo o fundo de uma página não é preciso inserir uma figura que possua as dimensões dela.

O que o Browser faz é carregar a imagem escolhida e então repeti-la quantas vezes for necessário a fim de preencher toda a tela. Chama-se ladrilhamento.

Devido ao fato de a imagem ser carregada uma única vez, é recomendável, para maior rapidez na abertura da página, que figuras pequenas sejam definidas como planos de fundo.

É claro que o aspecto estético também é determinante, mas em figuras do tipo textura, a adoção deste procedimento praticamente não será visível embora o tempo de carga da página seja sensivelmente reduzido.

Não é recomendável a utilização de figuras com muitos contrastes, pois isto pode dificultar a leitura das informações contidas na página.

Modifique o exemplo 'imagens2.html' para que fique como mostrado abaixo.

```
<HTML>
<HEAD>
<TITLE> Imagens 1 - Alinhamento de imagens </TITLE>
</HEAD>
<BODY BACKGROUND=“./imagem2.gif”>
  <H1> Alinhamento de imagens </H1>
  <P>
    Alinhamento pelo <IMG SRC=“imagem1.gif” ALIGN=top > topo
  <P>
    Alinhamento pelo <IMG SRC=“imagem1.gif” ALIGN=bottom > fundo
  <P>
    Alinhamento pelo <IMG SRC=“imagem1.gif” ALIGN=middle > meio
  <P>
    <IMG SRC=“imagem1.gif” ALIGN=right> Note que, com este tipo de alinhamento, o
    texto flui ao lado da figura naturalmente. Isso ocorre porque, quando alinhamos a figura
    pelas laterais, ela passa a não ser encarada pelo Browser como um caractere de linha.
  <P>
    <IMG SRC=“imagem1.gif” ALIGN=left> Note que, com este tipo de alinhamento, o
    texto flui ao lado da figura naturalmente. Isto ocorre porque, quando alinhamos a figura
    pelas laterais, ela passa a não ser encarada pelo Browser como um caractere de linha.
  </BODY>
</HTML>
```

Links

Como foi comentado no início das aulas, a principal atração da Internet é a criação de documentos com o conceito de hipertexto, ou seja, um documento que se liga a outros através de hyperlinks.

Com este recurso, o programador poderá permitir ao usuário o acesso imediato a qualquer endereço da Web a partir de sua página.

As palavras ou imagens de um documento que o interligam com outros são denominadas marcadores de links.

De um modo geral, os textos definidos como marcadores de links são exibidos pelos Browsers sublinhados e com uma cor padrão. Já as imagens que têm esta função costumam receber uma borda na cor padrão de link.

O marcador A (ANCHOR)

As tags <A> e podem marcar um texto ou uma imagem como hyperlink, assim como criar um endereço para uma parte específica de um documento que será referida por um link.

Os principais parâmetros do comando A são:

HREF= “nome ou URL do arquivo” - Especifica o endereço da URL ao qual o link está associado. Pode ser usado para referências dentro e fora do documento.

NAME=“nome” - Especifica o nome da seção de um documento que é referida por um link de hipertexto. Trataremos nas seções mais adiante.

TARGET= “janela” - Especifica o nome do frame ou janela onde será aberto o link. Quando formos tratar de frames, este parâmetro será mais bem explorado; por ora, é útil saber apenas que a sintaxe TARGET=“ blank” faz com que uma nova janela seja aberta para carregar o documento.

Caso o parâmetro TARGET não seja utilizado, o documento carregado será exibido na mesma janela onde está o link.

Nas subseções seguintes serão vistas as sintaxes para criar links em todas as situações possíveis.

Links para arquivos no mesmo diretório

Semelhante ao que acontece com o comando IMG, para fazer uma ligação entre dois documentos contidos no mesmo diretório, o programador necessita apenas especificar o nome completo do arquivo que será chamado no parâmetro HREF do comando A.

Este arquivo chamado poderá ser uma nova página, uma figura, ou até mesmo uma seção específica de um documento, como será visto à frente.

Sintaxe: Texto ou imagem

Na sintaxe apresentada, o elemento envolvido pelas marcações <A> e é definido como marcador do link para se chegar ao documento cujo nome foi especificado no parâmetro HREF, desde que este esteja no mesmo diretório do documento corrente.

Se o usuário clicar sobre o marcador, o documento especificado será exibido.

Como exemplos, podemos analisar as seguintes sintaxes:

** Praias ** - Faz com que a palavra 'Praias' se torne o marcador de um link para uma nova página, denominada praias.html no mesmo diretório do documento corrente.

** Praia ** - Faz com que a palavra 'Praia' se torne o marcador de um link para uma imagem, denominada praia.jpg no mesmo diretório do documento corrente.

** ** - Faz com que a imagem praia.jpg torne-se o marcador de um link para uma nova página, denominada praias.html no mesmo diretório do documento corrente.

Para visualizar o funcionamento de um link, crie os documentos abaixo.

Documento 1:

```
<HTML>
  <HEAD>
    <TITLE> Links 1 </TITLE>
  </HEAD>
  <BODY>
    <CENTER>
      <H1> Documento 1 </H1>
      <BR>
      Clique <A HREF="links2.html"> aqui </A> para acessar o documento 2.
    </CENTER>
  </BODY>
</HTML>
```

Salve-o com o nome de links1.html.

Documento 2:

```
<HTML>
  <HEAD>
    <TITLE> Links 2 </TITLE>
  </HEAD>
  <BODY>
    <CENTER>
      <H1> Documento 2 </H1>
      <P> Notou o funcionamento do link? </P>
    </CENTER>
  </BODY>
</HTML>
```

Salve-o com o nome de links2.html no mesmo diretório em que foi salvo o primeiro documento.

Abra o documento links1.html e teste o link criado.

Links para arquivos em diretórios diferentes

É possível fazer ligações com documentos que estejam em outros diretórios.

Para isso, basta especificar o caminho relativo, a partir do documento corrente, para o documento desejado, no parâmetro HREF do comando A.

O procedimento é análogo ao utilizado no comando IMG, visto na matéria anterior.

Como exemplos, podemos analisar as seguintes sintaxes:

** Praias ** - A palavra Praias aponta para o documento praias.html no diretório Dir1 abaixo do diretório corrente.

** Praias ** - A palavra Praias aponta para o documento praias.html no diretório imediatamente acima do diretório corrente.

** ** - A imagem praia.jpg aponta para o documento praias.html situada no diretório Dir4, abaixo do diretório Dir3, que por sua vez está abaixo do diretório que se encontra dois diretórios acima do diretório corrente.

Este tipo de especificação de endereço, onde o caminho para o documento desejado tem seu ponto de partida no diretório corrente, é conhecido como *link relativo*.

Podemos também especificar o endereço de um arquivo a partir do diretório raiz. Este tipo de especificação é denominado *link absoluto*.

Sintaxe de um link absoluto:

** Texto ou imagem**.

Por exemplo:

** Praias **.

A sintaxe acima indica que a palavra Praias aponta para o arquivo praias.htm situado no diretório Dir3, abaixo do diretório Dir2, abaixo do diretório Dir1 do disco C do computador.

Este tipo de endereçamento é o único possível quando se deseja acessar um arquivo contido numa unidade de disco diferente da do documento corrente.

Experimente mover o documento links2.html para o diretório acima do diretório atual e modificar o código do arquivo links1.html para:

```
<HTML>
  <HEAD>
    <TITLE> Links 1 </TITLE>
  </HEAD>
  <BODY>
    <CENTER>
      <H1> Documento 1 </H1>
      <BR>
      Clique <A HREF=" ../links2.html"> aqui </A> para acessar o documento 2.</CENTER>
    </BODY>
</HTML>
```

Verifique o funcionamento do link.

Experimente, agora, mover links2.html para uma pasta abaixo da que contém o arquivo 'links1.html'. Chame esta pasta de Subdir. Modifique novamente links1.html conforme mostrado abaixo:

```
<HTML>
  <HEAD>
    <TITLE> Links 1 </TITLE>
  </HEAD>
  <BODY>
    <CENTER>
      <H1> Documento 1 </H1>
      <BR>
      Clique <A HREF="Subdir/links2.html"> aqui </A> para acessar o documento 2.</CENTER>
    </BODY>
</HTML>
```

Verifique o funcionamento do link.

Links para arquivos em outro servidor

Para fazer ligações com documentos localizados em outro servidor é necessário apenas especificar corretamente sua URL no parâmetro HREF do comando A.

Este recurso nos permite ter acesso a documentos que estejam em qualquer computador do planeta e que esteja conectado adequadamente à Web.

Sintaxe: Texto ou imagem .

Observe o exemplo abaixo:

```
<A HREF= "http://www.servidor.com.br/praias.html"> Praias </A>.
```

No exemplo, a palavra Praias é o marcador de um link para se chegar ao arquivo praias.html situado no servidor www.servidor.com.br. O protocolo utilizado é o http e sua declaração é necessária para a sintaxe correta da URL, como vimos no primeiro capítulo.

Digite o exemplo abaixo, e salve-o como links3.html.

```
<HTML>
<HEAD>
<TITLE> Links 3 </TITLE>
</HEAD>
<BODY>
<CENTER>
<H1> Link para arquivo em outro servidor </H1>
<BR>
Clique <A HREF="http://www.yahoo.com.br/financas"> aqui </A> para acessar a página de
finanças do Yahoo na internet.
</CENTER>
</BODY>
</HTML>
```

Verifique o funcionamento do link e experimente variações.

Links para seções específicas em documentos

Muitas vezes, quando se tem uma página muito extensa, o usuário pode enfrentar dificuldades para localizar a informação que deseja.

Um procedimento muito comum nesses casos é endereçar partes importantes do texto do documento, como, por exemplo, títulos de seções, e criar um menu inicial no topo da página com links que levarão o usuário diretamente à parte que lhe interessa, evitando que ele tenha o trabalho de procurar.

Criando âncoras

Para endereçar uma seção é utilizado o parâmetro NAME do comando A. O endereço criado recebe o nome de âncora.

A sintaxe usada para criar uma âncora é: [Texto opcional] .

Referenciando âncoras

Uma vez criadas as âncoras, pode-se criar links para cada uma delas especificando seu nome no parâmetro HREF do comando A.

Se o programador tiver criado âncoras no início de cada seção do texto, ele será capaz de criar o menu inicial de que falamos, de forma a permitir ao usuário uma navegação mais dinâmica.

Sintaxe: Texto ou imagem .

O símbolo # avisa o Browser para que este procure o link no documento atual.

Utilizando-se os recursos de endereçamento descritos nas seções anteriores, o programador poderá referir, através de links, seções específicas em documentos no diretório corrente, em outros diretórios e até mesmo em outros servidores.

Observe os exemplos:

Seção 1 - O texto Seção 1 levará o usuário à seção Seção1 do documento atual. As marcações **** devem estar presentes no documento atual, endereçando a seção desejada.

Seção 1 - O texto Seção 1 levará o usuário à seção Seção1 do documento Doc1.html localizado no mesmo diretório do documento atual. As marcações **** devem estar presentes no documento Doc1.html, endereçando a seção desejada.

Seção 1 - O texto Seção 1 levará o usuário à seção Seção1 do documento Doc2.html localizado no diretório Dir1 abaixo do diretório atual. As marcações **** devem estar presentes no documento Doc2.html, endereçando a seção desejada.

Seção 1 - O texto Seção 1 levará o usuário à seção Seção1 do documento Doc3.html localizado no diretório imediatamente acima do diretório atual. As marcações **** devem estar presentes no documento Doc3.html, endereçando a seção desejada.

Digite o exemplo abaixo e salve-o como 'links4.html':

```
<HTML>
<HEAD>
<TITLE> Links 4 - Seções Específicas em Documentos </TITLE>
</HEAD>
<BODY>
<CENTER>
<H1> Link para uma seção específica em um documento </H1>
<BR>
Conheça <A
  HREF="https://extra.globo.com/tv-e-lazer/">
aqui </A> os famosos do Jornal Extra.
</CENTER>
</BODY>
</HTML>
```

Visualize o código fonte da página exibida pelo link do exemplo acima para ver as âncoras que ela possui. Tente modificar o exemplo links4.html para que o link leve o usuário a outra seção da mesma página.

■ Exemplo:

```

...
<h2>Preferências Pessoais</h2>
<ul>
  Hiperligação → <li><a href="#a1">Música</a></li>
                  <li><a href="#a2">Computação</a></li>
                  <li><a href="#a3">Automobilismo</a></li>
                  <li><a href="#a4">Natação</a></li>
                  <li><a href="#a5">Cinema</a></li>
</ul>
Local referenciado → <h2><a name="a1">Minhas músicas
                    preferidas</a></h2>
...

```

Enviando e-mail diretamente da página HTML

É possível acionar o programa padrão de e-mail diretamente de uma página HTML usando uma variação do comando A, que usa a opção MAILTO no lugar da URL.

Sintaxe: Texto ou imagem

Preenchendo o campo Assunto ou Subject

É possível preencher automaticamente o campo "Assunto" acrescentando a variável "?SUBJECT=" seguida do texto que se deseja exibir.

Sintaxe: Texto ou imagem .

Note que o texto do assunto não precisa estar entre aspas.

Enviando mensagens para mais de um destinatário

Para enviar mensagens para mais de um destinatário, basta separar seus endereços de e-mail por vírgulas.

Sintaxe: Texto ou imagem

Enviando uma cópia da mensagem

Para preencher o campo CC, basta adicionar "&cc=" e o endereço do destinatário.

Sintaxe: Texto ou imagem

Digite o exemplo abaixo e salve-o como links5.html.

<HTML>

<HEAD>

<TITLE> Links 5 - Enviando E-mail </TITLE>

</HEAD>

<BODY>

<CENTER>

<H1> Enviando e-mails diretamente
 da página </H1>

Envie um e-mail para o Professor.

</CENTER>

</BODY>

</HTML>

Listas

Uma forma muito conveniente de estruturar um documento HTML é a utilização de listas.

As listas podem funcionar como um sumário, um menu ou um resumo do conteúdo do documento e são excelentes para serem usadas como ponto de partida ao acesso de outras partes do documento.

Existem basicamente três tipos de listas:

- as não ordenadas, que contém uma série de itens sem numerá-los;
- as ordenadas, que atribuem um número para cada elemento da lista;
- e as listas de definições, que contém termos e suas respectivas definições.

Os comandos usados para criar cada um destes tipos de listas serão vistos nas próximas etapas.

O marcador UL

As marcações e são utilizadas para gerar listas não ordenadas (Unordered lists).

Estas tags devem envolver todo o conjunto de itens da lista, e cada item deve ser precedido da marcação (Line item).

O principal parâmetro do comando UL é o parâmetro TYPE, que permite que sejam especificadas variações para o marcador utilizado na lista.

Os valores possíveis de se atribuir ao parâmetro TYPE são:

TYPE=Square - define um quadrado preto como marcador.

TYPE=Circle - define um contorno circular como marcador.

TYPE=Disc - define um círculo preto como marcador.

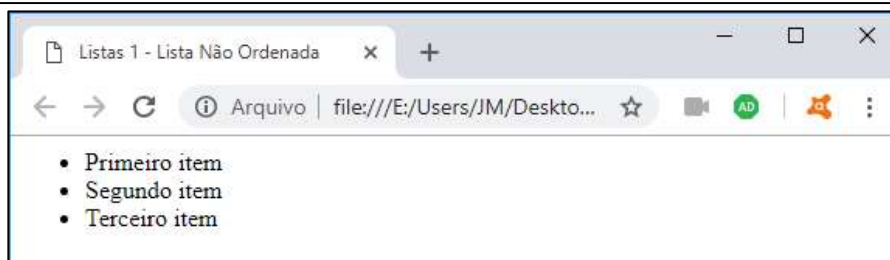
Sintaxe básica: [Itens da lista].

Sintaxe completa: <UL TYPE="tipo">[Itens da lista].

O exemplo abaixo ilustra a construção de uma lista não ordenada.

Digite-o salvando como 'listas1.html'.

```
<HTML>
<HEAD><TITLE> Listas 1 - Lista Não Ordenada </TITLE></HEAD>
<BODY>
<UL TYPE=DISC>
<LI> Primeiro item
<LI> Segundo item
<LI> Terceiro item
</UL>
</BODY>
</HTML>
```



O marcador OL

As marcações utilizadas para gerar listas ordenadas (Ordered lists) são `` e ``.

Estas tags devem envolver todo o conjunto de itens da lista, e cada item deve ser precedido da marcação ``.

Os principais parâmetros do comando OL são:

START="número" - permite escolher o primeiro número da lista. Se este parâmetro não for especificado, a numeração começará do início.

TYPE="tipo" - permite escolher o tipo de marcador da lista. Seus possíveis valores são:

TYPE=1 - Cria uma lista numérica normal. Não precisa ser especificado, pois é padrão.

TYPE=A - Cria uma lista alfabética com letras maiúsculas começando pelo A.

TYPE=a - Cria uma lista alfabética com letras minúsculas começando pelo a.

TYPE=I - Cria uma lista numérica com números romanos (maiúsculos) começando pelo I.

TYPE=i - Cria uma lista numérica com números romanos (minúsculos) começando pelo i.

É importante ressaltar que, independentemente do tipo de marcador utilizado, atribuiremos sempre um número ao parâmetro START. Se, por exemplo, o programador escrever `START=3` para uma lista alfabética com letras maiúsculas, o primeiro marcador da lista será a letra C, pois é a terceira letra do alfabeto. O mesmo acontece com listas alfabéticas com letras minúsculas e listas cujos marcadores são algarismos romanos.

Sintaxe básica: ` [Itens da lista] `.

Sintaxe completa: `<OL TYPE=_tipo_ START=_número_> [Itens da lista]`.

O exemplo abaixo ilustra a construção de uma lista ordenada.

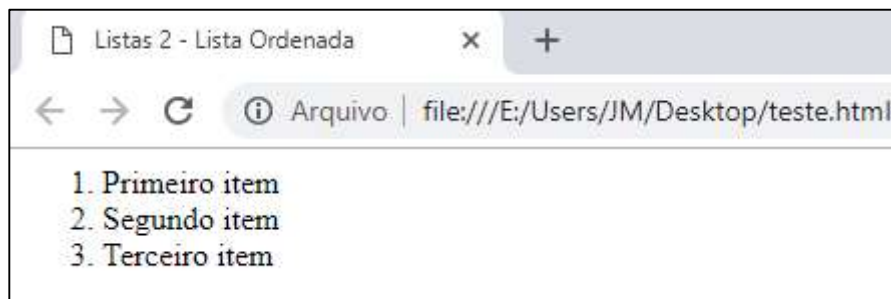
Digite-o salvando como listas2.html.

```
<HTML>
<HEAD>
<TITLE> Listas 2 - Lista Ordenada</TITLE>
</HEAD>
<BODY>
<OL>
<LI> Primeiro item
```

```

<LI> Segundo item
<LI> Terceiro item
</OL>
</BODY>
</HTML>

```



O marcador DL

Uma lista de definições consiste em alternar um termo e uma definição.

As tags <DL> e </DL> são utilizadas para gerar listas de definições.

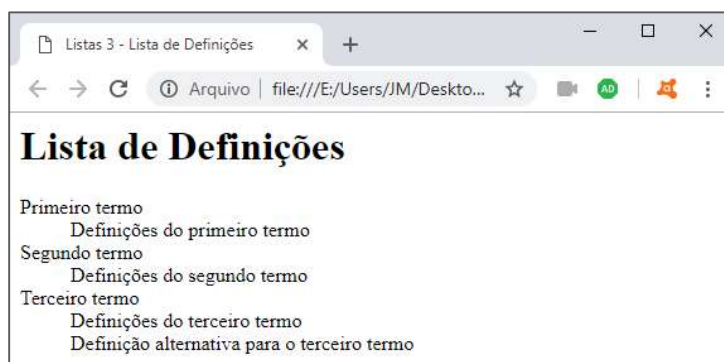
Estes comandos devem envolver todo o conjunto de termos e definições da lista. Os termos devem ser precedidos da marcação <DT> e as definições da marcação <DD>.

Digite o exemplo abaixo e salve-o como listas3.html.

```

<HTML>
<HEAD><TITLE> Listas 3 - Lista de Definições </TITLE></HEAD>
<BODY>
<H1>Lista de Definições</H1>
<DL>
<DT> Primeiro termo
<DD> Definições do primeiro termo
<DT> Segundo termo
<DD> Definições do segundo termo
<DT> Terceiro termo
<DD> Definições do terceiro termo
<DD> Definição alternativa para o terceiro termo
</DL>
</BODY>
</HTML>

```



Os browsers costumam exibir uma definição na linha de baixo e deslocada à direita, com relação a seu termo correspondente.

Aninhamento de listas

É possível aninhar diversas listas, ou seja, colocar uma lista dentro da outra para criar estruturas hierárquicas. Para tanto, na hora de escrever o código, o comando que cria a lista principal deve envolver o comando que cria a lista secundária.

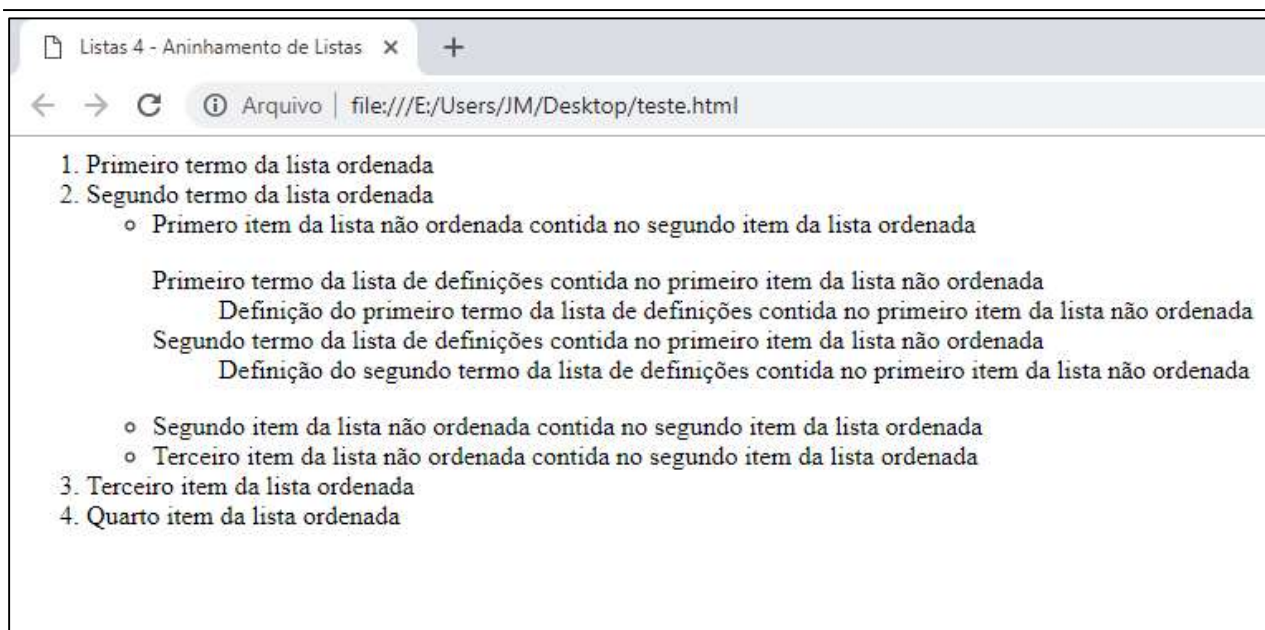
É permitido mesclar todos os tipos de listas.

O exemplo a seguir ilustra um aninhamento de listas de tipos diferentes.

Digite-o salvando como listas4.html.

```
<HTML>
<HEAD>
<TITLE> Listas 4 - Aninhamento de Listas </TITLE>
</HEAD>
<BODY>
<OL>
<LI> Primeiro termo da lista ordenada
<LI> Segundo termo da lista ordenada
<UL>
<LI> Primeiro item da lista não ordenada contida no segundo item da lista ordenada
<DL>
<DT> Primeiro termo da lista de definições contida no primeiro item da lista não ordenada
<DD> Definição do primeiro termo da lista de definições contida no primeiro item da lista não ordenada
<DT> Segundo termo da lista de definições contida no primeiro item da lista não ordenada
<DD> Definição do segundo termo da lista de definições contida no primeiro item da lista não ordenada
</DL>
<LI> Segundo item da lista não ordenada contida no segundo item da lista ordenada
<LI> Terceiro item da lista não ordenada contida no segundo item da lista ordenada
</UL>
<LI> Terceiro item da lista ordenada
<LI> Quarto item da lista ordenada
</OL>
</BODY>
</HTML>
```

Confira o resultado na página seguinte.



Listas com marcadores gráficos

Este não é um tipo padrão de lista da linguagem HTML, mas é um recurso bastante usado na confecção de páginas da Web.

O que desejamos agora é usar imagens como marcadores para os itens da lista, e para isso teremos que usar a marcação .

Escolha, primeiramente, uma imagem de seu agrado e salve como marcador. gif. Esta imagem será o marcador de nossa lista.

Digite, agora, o exemplo seguinte. Salve-o como 'listas5.html'.

```
<HTML>
<HEAD>
<TITLE> Listas 5 - Lista com Marcadores Gráficos </TITLE>
</HEAD>
<BODY>
<IMG SRC= "marcador.gif"> Primeiro item <BR>
<IMG SRC= "marcador.gif"> Segundo item <BR>
<IMG SRC= "marcador.gif"> Terceiro item
</BODY>
</HTML>
```

Note que, apesar de não possuir nenhum dos comandos padrão de lista estudados, a formatação do texto na tela será a de uma lista cujos marcadores são imagens.

Quando se deseja aninhar listas com marcadores gráficos, torna-se necessário utilizar as tags <DL> e <DD> para que as sub-listas sejam deslocadas à direita devidamente.

O exemplo seguinte ilustra este procedimento.

Digite-o salvando como listas6.html.

```
<HTML>
<HEAD>
<TITLE> Listas 6 - Lista com Marcadores Gráficos Aninhados
</TITLE>
</HEAD>
<BODY>
<IMG SRC= "marcador.gif"> Primeiro item <BR>
<IMG SRC= "marcador.gif"> Segundo item
<DL>
<DD><IMG SRC="marcador.gif" WIDTH="20" HEIGHT="20">
Primeiro sub-item do segundo item
<DD><IMG SRC="marcador.gif" WIDTH="20" HEIGHT="20">
Segundo sub-item do segundo item
<DL>
<DD><IMG SRC="marcador.gif" WIDTH="10" HEIGHT="10">
Primeiro sub-item do segundo sub-item
<DD><IMG SRC="marcador.gif" WIDTH="10" HEIGHT="10">
Segundo sub-item do segundo sub-item
</DL>
<DD><IMG SRC="marcador.gif" WIDTH="20" HEIGHT="20">
Terceiro sub-item do segundo item
</DL>
<IMG SRC= "marcador.gif"> Terceiro item <BR>
<IMG SRC= "marcador.gif"> Quarto item
</BODY>
</HTML>
```

Note que, no exemplo, os parâmetros WIDTH e HEIGHT do comando IMG foram utilizados para modificar os tamanhos dos marcadores de acordo com a hierarquia da lista.

Os tamanhos utilizados no exemplo podem não ficar bons dependendo da imagem escolhida, portanto, sinta-se livre para modificá-los.

É aconselhável usar sempre imagens cujo tamanho seja próximo ao do texto.

Tabelas

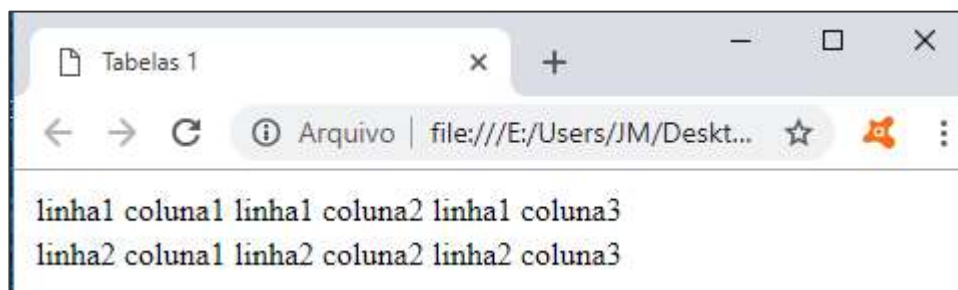
O uso de tabelas melhora muito a aparência de uma página, pois permite o alinhamento de textos e imagens, além de ordenar as informações de uma forma padronizada.

Uma tabela é constituída de linhas e colunas, e a interseção entre estes dois elementos é denominada célula.

As tabelas são criadas pelo comando TABLE, suas linhas são criadas pelo comando TR (Table Row) e as células de uma linha são criadas pelos comandos TD (Table Data) ou TH (Table Header). Todos esses comandos serão detalhados nas seções seguintes.

Para ter uma noção do funcionamento, digite o exemplo abaixo. Salve-o como tabs1.html.

```
<HTML>
<HEAD>
<TITLE> Tabelas 1 </TITLE>
</HEAD>
<BODY>
<TABLE>
<TR>
<TD> linha1 coluna1 </TD>
<TD> linha1 coluna2 </TD>
<TD> linha1 coluna3 </TD>
</TR>
<TR>
<TD> linha2 coluna1 </TD>
<TD> linha2 coluna2 </TD>
<TD> linha2 coluna3 </TD>
</TR>
</TABLE>
</BODY>
</HTML>
```



O marcador TABLE

As marcações <TABLE> e </TABLE> são usadas para criar tabelas.

Os principais parâmetros do comando TABLE são:

BORDER="número" - Especifica a largura da linha das bordas que separam as células da tabela. Se não for especificado este parâmetro, NÃO será exibida uma borda para a tabela.

CELLSPACING="número" - Especifica o espaçamento, em pixels, entre as células.

CELLPADDING="número" - Especifica o espaçamento, em pixels, entre o conteúdo e a borda da célula.

WIDTH="número | número%" - Especifica a largura da tabela em pixels ou em porcentagem relativa à largura da janela.

ALIGN="left | center | right" - Define o alinhamento da tabela na página.

BGCOLOR= "cor" - Especifica a cor de fundo de toda a tabela.

BORDERCOLOR="cor" - define a cor de borda da tabela. Só funciona se algum valor diferente de zero tiver sido atribuído ao parâmetro BORDER.

BACKGROUND= "nome ou URL do arquivo" - Define uma imagem como plano de fundo para TODA a tabela.

Sintaxe básica: <TABLE> [Comandos de criação de linhas e células]</TABLE>.

Sintaxe completa: <TABLE BORDER= "número" CELLSPACING="número" CELLPADDING="número" WIDTH="número | número%" ALIGN="tipo de alinhamento" BGCOLOR= "cor" BORDERCOLOR= "cor"> [Comandos de criação de linhas e células] </TABLE>.

ATRIBUTOS	COMENTÁRIOS	EXEMPLOS
BORDER	Quando especificado, a tabela será formatada com linhas de borda, de espessura igual ao valor em pixels especificado. Se o valor atribuído for igual a 0 (zero), funcionará exatamente como o caso padrão e não apresentará bordas sendo seu espaço liberado, podendo-se construir tabelas mais compactas.	<pre><TABLE BORDER=5> <TR> <TD>Primeiro</TD> <TD>Segundo</TD> </TR> </TABLE></pre>
ALIGN	Indica a posição que a tabela ocupará dentro do documento. Seus valores podem ser CENTER , LEFT ou RIGHT , informando se a tabela estará posicionada no centro, a esquerda ou a direita do documento, com o texto ocorrendo a sua volta.	<pre><TABLE ALIGN=LEFT> <TR> <TD>Primeiro</TD> <TD>Segundo</TD> </TR> </TABLE></pre>
WIDTH	Especifica a largura da tabela em pixels, ou o tamanho percentual de toda a tabela dentro da página.	<pre><TABLE WIDTH="50%"> <TR> <TD>Primeiro</TD> <TD>Segundo</TD> </TR> </TABLE></pre>
HEIGHT	Especifica a altura da tabela em pixels, ou o tamanho percentual de toda a tabela dentro da página.	<pre><TABLE HEIGHT="50%"> <TR> <TD>Primeiro</TD> <TD>Segundo</TD> </TR> </TABLE></pre>
CELLSPACING	Define o espaço em pixels entre cada célula na tabela.	<pre><TABLE CELLSPACING=5> <TR> <TD>Primeiro</TD> <TD>Segundo</TD> </TR> </TABLE></pre>
CELLPADDING	Define a distância em pixels, entre o conteúdo e a borda de cada célula.	<pre><TABLE CELLPPADDING=8> <TR> <TD>Primeiro</TD> <TD>Segundo</TD> </TR> </TABLE></pre>

BGCOLOR	Especifica uma cor de fundo para a tabela	<pre> <TABLE BGCOLOR="#eeffdd"> <TR> <TD>Primeiro</TD> <TD>Segundo</TD> </TR> </TABLE> </pre>
----------------	---	---

O marcador TR

A tag <TR> (Table Row) define uma linha da tabela. Para cada linha deve ser escrito um par de marcações, <TR> e </TR>, e entre elas devem ser especificados os comandos TD ou TH para criar as células de dados daquela linha.

As linhas de uma tabela não precisam ter necessariamente o mesmo número de células.

Os principais parâmetros do comando TR são:

ALIGN= "left | right | center | justify" - Especifica o alinhamento horizontal do conteúdo das células.

VALIGN="top | middle | bottom | baseline" - Especifica o alinhamento vertical do conteúdo das células.

BGCOLOR= "cor" - Especifica a cor de fundo para a linha.

BORDERCOLOR="cor" - Define a cor de borda das células da linha.

BACKGROUND="nome ou URL do arquivo" - Define uma imagem como plano de fundo para a linha.

Sintaxe básica: <TR> [Comandos de criação de células] </TR>.

Sintaxe completa: <TR ALIGN="tipo de alinhamento" VALIGN="tipo de alinhamento" BGCOLOR="cor" BORDERCOLOR="cor"> [Comandos de criação de células] </TR>.

OBS: Os parâmetros do comando TR se sobrepõem aos parâmetros de mesmo nome do comando TABLE.

O marcador TD

A tag <TD> (Table Data) define o conteúdo de uma célula da linha de uma tabela. Esse conteúdo deve ser escrito entre as marcações <TD> e </TD>.

Principais parâmetros do comando TD:

ALIGN= "left | right | center | justify" - Especifica o alinhamento horizontal do conteúdo da célula.

VALIGN="top | middle | bottom | baseline" - Especifica o alinhamento vertical do conteúdo da célula.

NOWRAP - Evita a quebra de linha do conteúdo da célula.

COLSPAN="número" - Este parâmetro permite aumentar a largura de uma célula fazendo com que ela ocupe colunas adjacentes. Nele deve ser especificado o número de colunas que a célula irá ocupar.

ROWSPAN="número" - Este parâmetro permite aumentar a altura de uma célula fazendo com que ela ocupe linhas adjacentes. Nele deve ser especificado o número de linhas que a célula irá ocupar.

WIDTH="número | número%" - Especifica a largura da célula em pixels ou em porcentagem relativa à largura da tabela.

BGCOLOR="cor" - Especifica a cor de fundo da célula.

BORDERCOLOR="cor" - Define a cor de borda da célula.

BACKGROUND="nome ou URL do arquivo" - Define uma imagem como plano de fundo para a célula.

Sintaxe básica: <TD> Caracteres </TD>.

Sintaxe completa: <TD ALIGN="tipo de alinhamento" VALIGN="tipo de alinhamento" NOWRAP COLSPAN=número ROWSPAN=número" WIDTH=número | número% BGCOLOR="cor" BORDERCOLOR="cor"> Caracteres </TD>.

Os parâmetros do comando TD se sobrepõem aos parâmetros de mesmo nome dos comandos TR e TABLE.

Títulos de colunas

Através do par de marcações <TH> e </TH> (Title Heading) o programador pode especificar títulos para as colunas. Estas tags funcionam da mesma forma que <TD> e </TD>, só que deixam, por padrão, o conteúdo da célula centralizado e em negrito. Os principais parâmetros são os mesmos do comando TD.

Os parâmetros do comando TH se sobrepõem aos parâmetros de mesmo nome dos comandos TR e TABLE.

Expansão de células em linhas e colunas - MESCLAGEM

Uma vez definida a estrutura básica de uma tabela, ou seja, o número de linhas e colunas, podemos 'apagar' a divisão entre células vizinhas de maneira a aumentar a largura ou a altura de uma célula, criando assim, tabelas mais bem definidas.

O parâmetro COLSPAN do comando TD permite aumentar o tamanho de uma célula fazendo com que ela ocupe colunas adjacentes, enquanto que o parâmetro ROWSPAN faz com que a célula aumente ocupando linhas adjacentes.

[illegible]

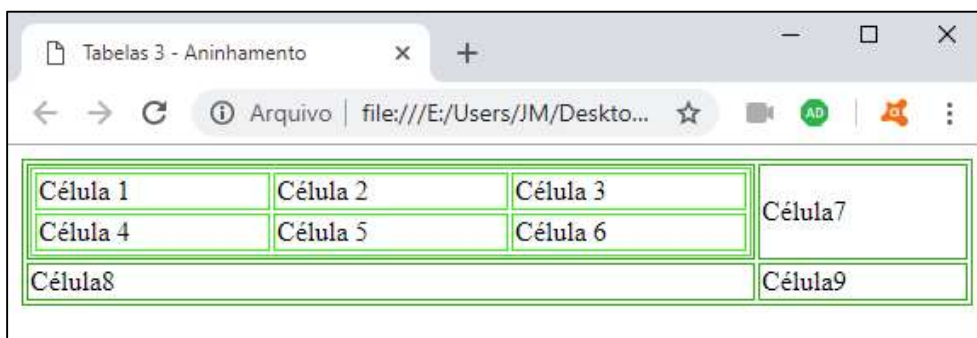
	Título 1		Título 2	
	Subtítulo 1	Subtítulo 2	Subtítulo 3	Subtítulo 4
Tópico 1	Dados	Dados	Dados não disponíveis	
Tópico 2	Dados	Dados		

Aninhamento de tabelas

Assim como o que acontece com as listas, é possível definir uma tabela dentro de outra. Para tanto, basta especificar entre marcações <TD> e </TD> a definição de uma outra tabela. Isto corresponde à inserção de uma tabela dentro de uma célula de outra tabela.

O exemplo abaixo ilustra este procedimento. Digite-o e salve como tabs3.html.

```
<HTML>
<HEAD>
<TITLE> Tabelas 3 - Aninhamento </TITLE>
</HEAD>
<BODY>
<TABLE BORDER=1 BORDERCOLOR= "black" WIDTH=100%>
<TR>
<TD>
<TABLE BORDER=1 BORDERCOLOR="red" WIDTH=100%>
<TR>
<TD> Célula 1 </TD>
<TD> Célula 2 </TD>
<TD> Célula 3 </TD>
</TR>
<TR>
<TD> Célula 4 </TD>
<TD> Célula 5 </TD>
<TD> Célula 6 </TD>
</TR>
</TABLE>
</TD>
<TD> Célula 7 </TD>
</TR>
<TR>
<TD> Célula 8 </TD>
<TD> Célula 9 </TD>
</TR>
</TABLE>
</BODY>
</HTML>
```



Legendas

Se o programador desejar criar um título ou legenda para a tabela, basta usar o comando CAPTION imediatamente após a abertura do comando TABLE.

O principal parâmetro do comando CAPTION é o parâmetro ALIGN.

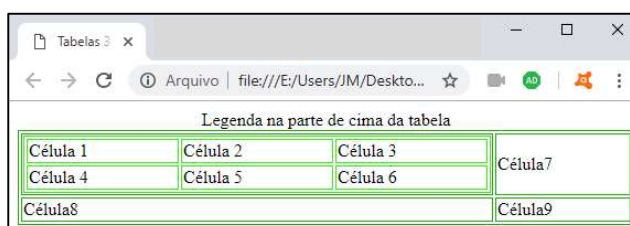
A sua sintaxe básica é: <CAPTION ALIGN= top | bottom> Legenda </CAPTION>.

Se atribuirmos ao parâmetro ALIGN o valor top, a legenda especificada será exibida na parte superior da tabela. Por outro lado, se a ele for atribuído o valor bottom, o browser exibirá a legenda especificada abaixo da tabela.

Em ambos os casos a legenda é centralizada de acordo com a tabela.

Modifique o exemplo 'tabs3.html', conforme indicado abaixo, para inserir uma legenda na parte superior da tabela.

```
<HTML>
<HEAD><TITLE> Tabelas 3 - Aninhamento </TITLE></HEAD>
<BODY>
<TABLE BORDER=1 BORDERCOLOR= "black" WIDTH=100%>
<CAPTION ALIGN= top> Legenda na parte de cima da tabela</CAPTION>
<TR>
<TD><TABLE BORDER=1 BORDERCOLOR="red" WIDTH=100%>
<TR>
<TD> Célula 1 </TD>
<TD> Célula 2 </TD>
<TD> Célula 3 </TD>
</TR>
<TR>
<TD> Célula 4 </TD>
<TD> Célula 5 </TD>
<TD> Célula 6 </TD>
</TR>
</TABLE></TD>
<TD> Célula 7</TD>
</TR>
<TR>
<TD> Célula 8</TD>
<TD> Célula 9</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```



Formulários

Os formulários eletrônicos são um dos mais fascinantes recursos da linguagem HTML pois através deles um usuário pode interagir com o servidor enviando-lhe dados para serem processados. Os comandos de criação de formulários são, portanto, os principais responsáveis pela troca de informações entre cliente e servidor.

Nesta etapa, você aprenderá a criar um formulário usando os recursos da linguagem HTML. Porém, para que um formulário funcione, é necessário que no lado do servidor exista um outro programa, escrito em outra linguagem de programação, que será o responsável pelo recebimento e processamento das informações enviadas. Este tipo de programa é denominado CGI e não será abordado nesta disciplina.

O marcador FORM

As marcações `<FORM>` e `</FORM>` são usadas para definir um formulário.

Este comando tem duas funções importantes: especificar o local do programa que controlará o formulário e definir a forma ou método como os dados serão enviados.

O comando FORM possui os seguintes parâmetros:

ACTION="URL" - Especifica o local (URL) do servidor e do programa CGI que vai processar os dados do formulário.

METHOD="Método de troca de dados" - Indica o método usado pelo servidor para receber os dados do formulário. As opções possíveis para este parâmetro são:

GET - As informações são enviadas como se fizessem parte da URL.

POST - As informações são enviadas num bloco de dados separado da URL. É o método mais utilizado.

Sintaxe: `<FORM ACTION="URL" METHOD="método"> [Campos do formulário] </FORM>`

Nos exemplos das seções seguintes ignoraremos os parâmetros do comando FORM, uma vez que não estaremos realizando envio de informação através dos formulários apresentados.

O marcador INPUT

O marcador INPUT define um campo de entrada de dados, onde o usuário digita as informações requeridas pelo formulário. Cada campo de um formulário atribui seu conteúdo a uma variável que possui nome e tipo específicos.

Os parâmetros do marcador INPUT são os seguintes:

TYPE="Tipo de variável" - Especifica o tipo de dado para a variável.

Suas opções possíveis são:

TEXT - Aceita dados do tipo caractere. É o tipo de variável mais usado para a digitação de um campo de texto com tamanho conhecido.

Por padrão, a caixa de digitação criada para este tipo de variável terá largura equivalente a 20 caracteres, não impõe limite para a quantidade de caracteres digitados.

PASSWORD - Aceita dados do tipo caractere. Funciona exatamente como uma variável do tipo TEXT, porém, exibe um asterisco no lugar de cada caractere digitado para impedir sua visualização na tela. É ideal para a digitação de senhas e palavras-chave.

RADIO - Aceita campos do tipo sim/não. Permite a exibição de várias opções, mas só aceita que uma delas seja escolhida. O campo de entrada de dados criado para este tipo de variável é um pequeno círculo, que pode ser marcado com um clique. Dois círculos associados à mesma variável não podem estar marcados simultaneamente, pois quando o usuário clicar sobre o segundo, o primeiro será automaticamente desmarcado.

CHECKBOX - Aceita campos do tipo sim/não. Permite a exibição de várias opções e aceita que o usuário escolha várias delas. O campo de entrada de dados criado para este tipo de variável é um pequeno quadrado, que pode ser marcado com um clique.

SUBMIT - Cria um botão que envia o conteúdo do formulário ao ser pressionado. Os dados são enviados para a URL especificada no parâmetro ACTION e de acordo com o método escolhido no parâmetro METHOD do comando FORM. Por padrão, a legenda do botão criado será 'Submit'

RESET - Cria um botão que, ao ser pressionado, possibilita ao usuário apagar e restabelecer o valor padrão para todos os campos do formulário de uma única vez. Por padrão, a legenda do botão criado será 'Reset'.

HIDDEN - Define um campo invisível, cujo conteúdo é enviado junto com os demais.

NAME="nome" - Nome da variável que receberá o conteúdo do campo. Este parâmetro é essencial na sintaxe de criação de qualquer campo de entrada de dados, seja qual for o tipo de variável escolhido.

VALUE="valor" - Permite definir um conteúdo prévio para campos dos tipos *TEXT* e *PASSWORD*, ou seja, um texto que aparecerá escrito na caixa de digitação quando o formulário for aberto. Para os campos do tipo *CHECKBOX* ou *RADIO* representa o valor que será atribuído à variável se a opção for marcada. Para os campos *SUBMIT* e *RESET* indica o texto que aparece no botão. Este parâmetro é obrigatório apenas para variáveis dos tipos *CHECKBOX* e *RADIO*.

CHECKED - Só se aplica a campos do tipo *CHECKBOX* e *RADIO*. Os campos cuja sintaxe de criação possui este parâmetro aparecerão marcados quando o formulário for aberto.

SIZE="tamanho" - Define a quantidade de caracteres exibidos em campos do tipo *TEXT* ou *PASSWORD*. Este número delimita apenas o tamanho do campo para digitação, e não a quantidade de caracteres que o campo pode ter.

MAXLENGTH="comprimento" - Define o número máximo de caracteres permitidos para variáveis do tipo *TEXT* ou *PASSWORD*.

ID="String de identificação" - Identifica o campo para permitir que seja criada uma tecla de atalho para ele.

Sintaxe básica para campos do tipo **TEXT**:

<INPUT TYPE="text" NAME="nome">.

Sintaxe completa: <INPUT TYPE="text" NAME="nome" SIZE="tamanho" MAXLENGTH="comprimento" VALUE="valor">.

Sintaxe básica para campos do tipo **PASSWORD**:

<INPUT TYPE="password" NAME="nome">.

Sintaxe completa: <INPUT TYPE="password" NAME="nome" SIZE="tamanho" MAXLENGTH="comprimento" VALUE="valor" >.

Sintaxe básica para campos do tipo **RADIO**:

<INPUT TYPE="radio" NAME="nome" VALUE="valor">.

Sintaxe completa: <INPUT TYPE="radio" NAME="nome" VALUE="valor" CHECKED>.

Sintaxe básica para campos do tipo CHECKBOX:

<INPUT TYPE="checkbox" NAME="nome" VALUE="valor">.

Sintaxe completa: <INPUT TYPE="radio" NAME="nome" VALUE="valor" CHECKED>.

Digite o exemplo abaixo e salve-o como 'forms1.html'.

```
<HTML>
<HEAD>
<TITLE> Formulários 1 </TITLE>
</HEAD>
<BODY>
<CENTER>
<H1>Ficha de Inscrição</H1>
</CENTER>
<FORM>
Nome: <INPUT TYPE=text NAME=nome>
<P>Sexo:
<INPUT TYPE=radio NAME=sexo VALUE=m>Masculino
<INPUT TYPE=radio NAME=sexo VALUE=f>Feminino
<P> Assinale abaixo os programas de computador que você domina:
<BR><INPUT TYPE=checkbox NAME=prog VALUE=word> Word
<BR><INPUT TYPE=checkbox NAME=prog VALUE=excel> Excel
<BR><INPUT TYPE=checkbox NAME=prog VALUE=access> Access
<BR><INPUT TYPE=checkbox NAME=prog VALUE=powerpoint>
Power Point
<P ALIGN=center> <INPUT TYPE=submit VALUE=Enviar> <INPUT TYPE=reset VALUE="Apagar
Campos"> </P>
</FORM>
</BODY>
</HTML>
```



The screenshot shows a web browser window with the title 'Formulário: x'. The address bar shows 'file:///E:/Users/JM/Desktop...'. The form content is as follows:

Ficha de Inscrição

Nome:

Sexo: ☐ Masculino ☐ Feminino

Assinale abaixo os programas de computador que você domina:

- ☐ Word
- ☐ Excel
- ☐ Access
- ☐ Power Point

Os marcadores SELECT e OPTION

As tags <SELECT> e </SELECT> definem e exibem uma lista de itens que podem ser selecionados pelo usuário. Visualmente elas criam uma lista de seleção onde uma caixa é exibida com o valor padrão, e o clique em uma seta faz surgir uma lista de opções. Um ou mais itens podem ser selecionados.

Os parâmetros do comando SELECT são:

NAME="nome" - Especifica o nome da variável que receberá o conteúdo do campo.

SIZE="tamanho" - Especifica a quantidade de itens que será exibida de uma vez pela lista. Este parâmetro é opcional.

MULTIPLE - Indica que podem ser escolhidos vários itens da lista. É opcional.

Sintaxe básica: <SELECT NAME="nome"> [Itens da lista] </SELECT>.

Sintaxe completa: <SELECT NAME="nome" SIZE="tamanho" MULTIPLE> [Itens da lista] </SELECT>.

Cada item da lista deve ser especificado pelo comando OPTION, cujos principais parâmetros são:

VALUE="conteúdo" - Define o conteúdo que a variável correspondente receberá caso a opção seja selecionada.

SELECTED - Indica que a opção deve ser previamente marcada como selecionada.

Sintaxe básica: <OPTION VALUE="valor">.

Sintaxe completa: <OPTION VALUE="valor" SELECTED>.

Complemente o exemplo 'forms1.html' conforme o indicado abaixo.

```
<HTML>
<HEAD>
<TITLE> Formulários 1 </TITLE>
</HEAD>
<BODY>
<CENTER>
<H1>Ficha de Inscrição</H1>
</CENTER>
<FORM>
Nome: <INPUT TYPE=text NAME=nome>
<P>Sexo:
<INPUT TYPE=radio NAME=sexo VALUE=m>Masculino
<INPUT TYPE=radio NAME=sexo VALUE=f>Feminino
<P> Assinale abaixo os programas de computador que você domina:
```

```
<BR><INPUT TYPE=checkbox NAME=prog VALUE=word> Word
<BR><INPUT TYPE=checkbox NAME=prog VALUE=excel> Excel
<BR><INPUT TYPE=checkbox NAME=prog VALUE=access> Access
<BR><INPUT TYPE=checkbox NAME=prog VALUE=powerpoint>
Power Point
<P> Cargo:</P>
<SELECT NAME=cargo>
<OPTION VALUE=analista>Analista de Sistemas
<OPTION VALUE=gerente>Gerente
<OPTION VALUE=programador>Programador C
<OPTION VALUE=Webmaster>Webmaster
</SELECT>
<P ALIGN=center>
<INPUT TYPE=submit VALUE=Enviar> <INPUT TYPE=reset VALUE="Apagar Campos"> </P>
</FORM>
</BODY>
</HTML>
```

The screenshot shows a web browser window with the title 'Formulários 1'. The address bar shows a local file path. The main content is a form titled 'Ficha de Inscrição'. It contains the following elements: a text input for 'Nome:', radio buttons for 'Sexo: Masculino' and 'Feminino', a section titled 'Assinale abaixo os programas de computador que você domina:' with checkboxes for 'Word', 'Excel', 'Access', and 'Power Point', a 'Cargo:' dropdown menu currently showing 'Analista de Sistemas', and two buttons at the bottom: 'Enviar' and 'Apagar'.

O marcador TEXTAREA

As marcações <TEXTAREA> e </TEXTAREA> definem uma caixa de digitação onde o usuário pode digitar livremente um texto. Esta caixa pode ser previamente preenchida com algum texto que deve ser especificado no local de 'texto padrão'.

Os parâmetros do comando TEXTAREA são:

NAME="nome" - Especifica o nome da variável que receberá o conteúdo do campo.

ROWS="número" - Especifica a altura, ou seja, a quantidade de linhas que a caixa deve ter.

COLS="número" - Especifica a largura, ou seja, a quantidade de colunas que a caixa deve ter.

Assim, sua sintaxe é: <TEXTAREA NAME="nome" ROWS="número" COLS="número"> Texto Padrão </TEXTAREA>.

Complemente mais uma vez o exemplo 'forms1.html' conforme o indicado.

```
<HTML>
<HEAD><TITLE> Formulários 1 </TITLE></HEAD>
<BODY>
<CENTER><H1>Ficha de Inscrição</H1></CENTER>
<FORM>
Nome: <INPUT TYPE=text NAME=nome>
<P>Sexo:
<INPUT TYPE=radio NAME=sexo VALUE=m>Masculino
<INPUT TYPE=radio NAME=sexo VALUE=f>Feminino
<P> Assinale abaixo os programas de computador que você domina:
<BR><INPUT TYPE=checkbox NAME=prog VALUE=word> Word
<BR><INPUT TYPE=checkbox NAME=prog VALUE=excel> Excel
<BR><INPUT TYPE=checkbox NAME=prog VALUE=access> Access
<BR><INPUT TYPE=checkbox NAME=prog VALUE=powerpoint>
Power Point
<SELECT NAME=cargo>
<OPTION VALUE=analista>Analista de Sistemas
<OPTION VALUE=gerente>Gerente
<OPTION VALUE=programador>Programador C
<OPTION VALUE=Webmaster>Webmaster
</SELECT>
<P> Por que deseja trabalhar em nossa empresa?
<BR><TEXTAREA ROWS=6 COLS=40 NAME=pq>
Mínimo de 3 linhas!</TEXTAREA><P ALIGN=center> <INPUT TYPE=submit VALUE=Enviar>
<INPUT TYPE=reset VALUE="Apagar Campos"> </P>
</FORM>
</BODY></HTML>
```

The screenshot shows a web browser window titled 'Formulários 1'. The address bar shows a local file path. The form content is as follows:

- Ficha de Inscrição** (Title)
- Nome:
- Sexo: ☐ Masculino ☐ Feminino
- Assinale abaixo os programas de computador que você domina:
 - ☐ Word
 - ☐ Excel
 - ☐ Access
 - ☐ Power Point
- (Dropdown menu)
- Por que deseja trabalhar em nossa empresa?

Mínimo de 3 linhas!
- Buttons:

Alinhamento de campos em formulários

Usando o marcador PRE

Um dos problemas que o programador pode enfrentar na criação de formulários é o fato de os campos do formulário não ficarem alinhados naturalmente.

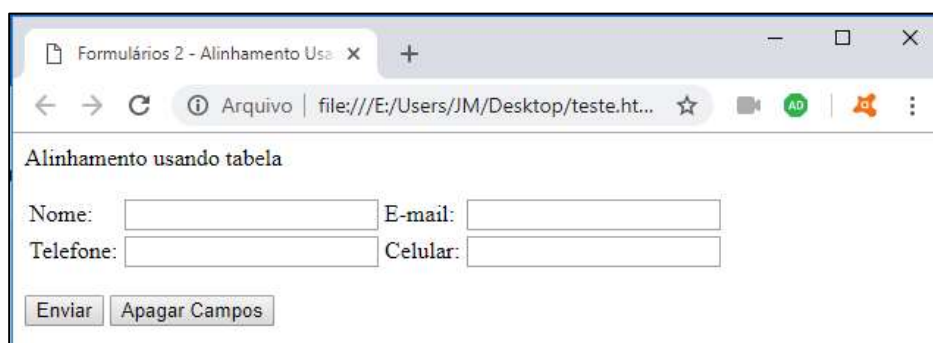
Uma técnica simples de alinhamento consiste em envolver os comandos relativos aos campos de formulário pelas marcações <PRE> e </PRE>.

O texto formatado pelo comando PRE é exibido em uma fonte de largura fixa, ou seja, todas as letras possuem a mesma largura. Desta forma, basta acrescentar os espaços necessários entre o rótulo do campo e o comando INPUT para que seja feito o alinhamento.

Usando tabelas

Uma técnica mais versátil consiste na utilização de tabelas para alinhar os campos. Nesta técnica a tabela deve possuir colunas de títulos e colunas de campos. Para ter uma noção mais clara desta técnica, digite o exemplo abaixo. Salve-o como 'forms2.html'.

```
<HTML>
<HEAD><TITLE> Formulários 2 - Alinhamento Usando Tabelas</TITLE></HEAD>
<BODY><P> Alinhamento usando tabela
<FORM>
<TABLE>
<TR>
<TD>Nome:</TD>
<TD><INPUT TYPE=text NAME=nome></TD>
<TD>E-mail:</TD>
<TD><INPUT TYPE=text NAME=email></TD>
</TR>
<TR>
<TD>Telefone:</TD>
<TD><INPUT TYPE=text NAME=tel></TD>
<TD>Celular:</TD>
<TD><INPUT TYPE=text NAME=cel></TD>
</TR>
</TABLE>
<P> <INPUT TYPE=submit VALUE=Enviar> <INPUT TYPE=reset VALUE="Apagar Campos">
</FORM>
</BODY></HTML>
```



Formulários 2 - Alinhamento Usando Tabelas

Alinhamento usando tabela

Nome: E-mail:

Telefone: Celular:

Agrupamento de campos com o marcador FIELDSET

O marcador FIELDSET é utilizado para envolver campos com uma moldura.

Basta que os campos sejam criados entre as marcações <FIELDSET> e </FIELDSET>.

Sua sintaxe básica é: <FIELDSET> [Campos] </FIELDSET>.

Além disso, o comando LEGEND, definido dentro do comando FIELDSET, permite especificar nomes para os conjuntos de campos e criar teclas

de atalho para eles. Sua sintaxe básica é: <LEGEND> Nome do conjunto de campos </LEGEND>.

Estes recursos não são compatíveis com todos os Browsers.

Digite o exemplo abaixo e salve-o como 'forms3.html'.

```
<HTML>
<HEAD><TITLE> Formulários 3 - FIELDSET </TITLE></HEAD>
<BODY>
<FORM>
<FIELDSET>
<LEGEND> Informações de Contato</LEGEND>
Nome:<INPUT TYPE=text NAME=nome><BR>
E-mail:<INPUT TYPE=text NAME=email><BR>
Telefone:<INPUT TYPE=text NAME=tel>
</FIELDSET>
<FIELDSET>
<LEGEND>Cartão de crédito</LEGEND>
<INPUT TYPE=radio NAME=card VALUE=visa> Visa
<INPUT TYPE=radio NAME=card VALUE=mc> Master Card
<BR>
Número: <INPUT TYPE=text NAME=number><BR>
Data de validade: <INPUT TYPE=text SIZE=7 NAME=data
VALUE="mm/aaaa">
</FIELDSET>
</FORM>
</BODY></HTML>
```


Teclas de atalho para conjuntos de campos

O parâmetro ACCESSKEY do marcador LEGEND permite ao programador definir teclas que, quando pressionadas juntamente com a tecla ALT, movem o cursor para o primeiro campo de cada conjunto.

Sintaxe: <LEGEND ACCESSKEY=A> Nome do conjunto de campos</LEGEND>

A sintaxe acima define a tecla 'a' como tecla de atalho para o conjunto de campos 'Nome do conjunto de campos'.

Modifique o exemplo 'forms3.html' para que fique como mostrado abaixo.

Teste as teclas de atalho.

```
<HTML>
<HEAD><TITLE> Formulários 3 - FIELDSET </TITLE></HEAD>
<BODY>
<FORM>
<FIELDSET>
<LEGEND ACCESSKEY=I>
<U>I</U>nformações de Contato</LEGEND>
Nome:<INPUT TYPE=text NAME=nome><BR>
E-mail:<INPUT TYPE=text NAME=email><BR>
Telefone:<INPUT TYPE=text NAME=email>
</FIELDSET>
<FIELDSET>
<LEGEND ACCESSKEY=C> <U>C</U>artão de
crédito</LEGEND>
<INPUT TYPE=radio NAME=card VALUE=visa> Visa
<INPUT TYPE=radio NAME=card VALUE=mc> Master Card<BR>
Número: <INPUT TYPE=text NAME=number><BR>
Data de validade: <INPUT TYPE=text SIZE=7 NAME=data VALUE="00/0000">
</FIELDSET>
</FORM>
</BODY></HTML>
```

Teclas de atalho para campos

Outro marcador que permite ao usuário navegar rapidamente pelos campos de um formulário através de teclas de atalho é o marcador LABEL. Esta tag especifica uma tecla que, ao ser pressionada juntamente com a tecla ALT, move o cursor diretamente para o campo especificado.

Para utilizar o marcador LABEL é preciso identificar cada campo através do parâmetro ID do marcador INPUT.

A sintaxe usada é: <INPUT TYPE="tipo" NAME="nome" ID="String de identificação">.

Uma vez identificado o campo, a tecla de atalho para ele será definida através do parâmetro ACCESSKEY do marcador LABEL. O parâmetro FOR deste mesmo marcador indicará o campo a que a tecla de atalho levará.

A sintaxe usada é: <LABEL ACCESSKEY=A FOR= "String de identificação"> Rótulo do campo </LABEL>[Campo].

O marcador LABEL pode também ser fechado logo após sua abertura ou até mesmo depois da tag de criação do campo.

Modifique novamente o exemplo 'forms3.html' para que fique como mostrado abaixo. Teste agora as teclas de atalho para cada campo.

```
<HTML>
<HEAD><TITLE> Formulários 3 - FIELDSET </TITLE></HEAD>
<BODY>
<FORM>
<FIELDSET>
<LEGEND ACCESSKEY=I> <U>I</U>nformações de Contato</LEGEND>
<LABEL ACCESSKEY=N FOR=nome> <U>N</U>ome:
</LABEL><INPUT TYPE=text NAME=nome ID=nome><BR>
<LABEL ACCESSKEY=E FOR=email><U>E</U>-mail:
</LABEL><INPUT TYPE=text NAME=email ID=email><BR>
<LABEL ACCESSKEY=T FOR=tel> <U>T</U>ELEFONE:
</LABEL><INPUT TYPE=text NAME=tel ID=tel>
</FIELDSET>
<FIELDSET>
<LEGEND ACCESSKEY=C> <U>C</U>artão de crédito</LEGEND>
<LABEL ACCESSKEY=V FOR=visa><INPUT TYPE=radio NAME=card VALUE=visa ID=visa>
<U>V</U>isa
</LABEL>
<LABEL ACCESSKEY=M FOR=master><INPUT TYPE=radio NAME=card VALUE=mc
ID=master><U>M</U>aster Card </LABEL><BR>
<LABEL ACCESSKEY=U FOR=num>
N<U>ú</U>mero: </LABEL><INPUT TYPE=text
NAME=number ID=num><BR>
<LABEL ACCESSKEY=D FOR=data> <U>D</U>ata de validade: </LABEL> <INPUT TYPE=text SIZE=7
NAME=data VALUE="00/0000" ID=data>
</FIELDSET>
</FORM>
</BODY></HTML>
```

Frames

O recurso de frames permite ao programador criar páginas HTML que podem ser visualizadas simultaneamente na janela do Browser. Desta forma, a janela fica dividida em frames que compartilham o espaço disponível.

Este recurso não é compatível com todos os clientes Web, porém é aceito pelas versões mais recentes dos navegadores mais populares.

O uso de frames requer um planejamento prévio, que consiste na criação da estrutura de janelas. O programador irá especificar a quantidade de frames, a disposição que eles terão na tela, a largura e a altura de cada frame, assim como o conteúdo que será exibido em cada um deles.

O frame document

O uso de frames exige a criação de um documento HTML especial que contém todas as definições dos frames. Este documento se diferencia dos demais por não utilizar o comando BODY. Em seu lugar é usado o comando FRAMESET.

Dentro do par de marcações <FRAMESET></FRAMESET> são especificados todos os atributos dos frames que serão criados, tais como sua quantidade e disposição horizontal e vertical.

Cada frame especificado precisará de um outro comando para definir suas características individuais e, principalmente, seu conteúdo. Para tanto é utilizado o comando FRAME.

O marcador FRAMESET

Os principais parâmetros do marcador FRAMESET são:

COLS="tamanhos" - É usado para criar um documento com frames dispostos em colunas. Nele deverá ser especificada a largura de cada coluna do documento. Os valores podem ser especificados em pixels, percentagem ou tamanho relativo.

ROWS="tamanhos" - É usado para criar um documento com frames dispostos em linhas. Nele deverá ser especificada a altura de cada linha do documento. Os valores podem ser especificados em pixels, percentagem ou tamanho relativo.

FRAMEBORDER=1|0 - Mostra ou inibe borda para os frames, caso receba os valores 1 ou 0 respectivamente.

FRAMESPACING="valor" - Cria espaço adicional em pixels entre os frames.

Sintaxe básica: `<FRAMESET COLS | ROWS="tamanhos">` [Especificação individual de cada frame] `</FRAMESET>`.

Sintaxe completa: `<FRAMESET COLS | ROWS="tamanhos" FRAMEBORDER=1|0 FRAMESPACING="valor">` [Especificação individual de cada frame] `</FRAMESET>`.

Especificação de tamanho de frames

Usando valores em pixels

Como vimos, pode-se especificar o tamanho de frames no comando FRAMESET através de seus parâmetros ROWS e COLS. A maneira mais simples de fazê-lo é atribuir valores em pixels à dimensão com que se está trabalhando, altura no caso do parâmetro ROWS e largura no caso do parâmetro COLS.

As dimensões dos frames devem estar separadas entre si por vírgulas.

Veja os exemplos:

`<FRAMESET COLS="100,300,200">` [Frames] `</FRAMESET>` - Cria três frames dispostos em colunas, cujas larguras, da esquerda para a direita, são de 100, 300 e 200 pixels.

`<FRAMESET ROWS="60,100,100,200">` [Frames] `</FRAMESET>` - Cria quatro frames dispostos em linhas, cujas alturas, de cima para baixo, são de 60, 100, 100 e 200 pixels.

Nota: Quando a dimensão especificada para um determinado frame não é suficiente para exibir seu conteúdo, barras de rolagem aparecem automaticamente para auxiliar o usuário.

Usando valores relativos

Para dividir frames igualmente pela dimensão da janela do Browser, a melhor opção é usar o sinal de asterisco para cada linha ou coluna.

Veja os exemplos:

`<FRAMESET COLS="*,*,*">` [Frames] `</FRAMESET>` - Cria três frames dispostos em colunas cujas larguras são iguais.

`<FRAMESET ROWS="*,*,*,*">` [Frames] `</FRAMESET>` - Cria cinco frames dispostos em linhas cujas alturas são iguais.

`<FRAMESET COLS="*,2*">` [Frames] `</FRAMESET>` - Cria dois frames dispostos em colunas, sendo que o da direita ocupa dois terços da tela.

`<FRAMESET ROWS="*,300,*">` [Frames] `</FRAMESET>` - Cria um frame com altura de 300 pixels no centro da janela e, acima e abaixo dele, dois frames de mesma altura.

Usando valores percentuais

A terceira forma de especificar tamanhos de frames é utilizando valores na forma percentual, totalizando 100%.

Veja os exemplos:

<FRAMESET ROWS="30%, 70%"> [Frames] </FRAMESET> - Cria dois frames dispostos em linhas, sendo que o de cima ocupa 30% da tela, e o de baixo ocupa 70%.

<FRAMESET COLS="30%, 50%, 20%"> [Frames] </FRAMESET> - Cria três frames dispostos em colunas, sendo que o da esquerda ocupa 30% da tela, o do centro 50% e o da direita 20%.

<FRAMESET ROWS="25%, *, 100"> [Frames] </FRAMESET> - Cria três frames dispostos em linhas, sendo que o de cima ocupa 25% da tela, o de baixo tem altura de 100 pixels e o do centro ocupa o espaço restante.

Nota: Como você deve ter percebido pelos exemplos, os diferentes formatos de valores podem ser combinados entre si.

O marcador FRAME

Enquanto o comando FRAMESET cria o layout geral dos frames, o comando FRAME é o responsável pela atribuição do conteúdo de cada frame e das características individuais de cada um deles. Os parâmetros do comando FRAME se sobrepõem aos parâmetros de mesmo nome do comando FRAMESET.

Principais parâmetros do comando FRAME:

SRC= "Endereço" - Este parâmetro é o único indispensável, pois especifica o nome ou a URL do documento que será exibido no frame.

ALIGN="posição" - Ajusta o alinhamento do frame ou do texto. As opções permitidas para este parâmetro são:

- Top - O texto ao redor do frame é alinhado pela parte superior do frame.
- Middle - O texto ao redor do frame é alinhado pelo meio do frame.
- Bottom - O texto ao redor do frame é alinhado pela parte inferior do frame.
- Left - O frame é alinhado à esquerda, deixando o texto posicionado no seu lado direito.
- Right - O frame é alinhado à direita, deixando o texto posicionado no seu lado esquerdo

FRAMEBORDER=1|0 - Idêntico ao parâmetro de mesmo nome do comando FRAMESET, ele ativa ou desativa a exibição de borda para o frame atual.

MARGINHEIGHT="altura" - Especifica a altura das margens superior e inferior do frame em pixels.

MARGINWIDTH="largura" - Especifica a largura das margens esquerda e direita do frame em pixels.

NAME="nome" - Atribui um nome para o frame, de maneira que possa ser identificado e localizado para carregar documentos.

NORESIZE - Evita que o frame seja redimensionado pelo usuário.

SCROLLING=yes| no - Quando ajustado para o valor 'no', não permite que o frame possua barras de rolagem.

Sintaxe básica: <FRAME SRC="nome ou URL do arquivo">.

Sintaxe completa: <FRAME SRC="nome ou URL do arquivo" ALIGN="posição" FRAMEBORDER=1 | 0 MARGINHEIGHT="altura" MARGINWIDTH="largura" NAME="nome" NORESIZE SCROLLING=yes | no>.

Combinando ROWS e COLS

Pode-se combinar a criação de frames usando os dois atributos. O programador pode, por exemplo, criar uma janela dividida em dois frames horizontais, onde o de baixo é dividido em duas colunas.

Para que isto seja possível, basta aninhar conjuntos de comandos FRAMESET.

Para exemplificar, vamos criar, primeiramente, os documentos que serão exibidos por cada frame:

Documento 1:

```
<HTML>
  <HEAD><TITLE> Documento 1 </TITLE></HEAD>
  <BODY>
    <CENTER><H1> Documento 1 </H1></CENTER>
  </BODY>
</HTML>
```

Salve-o com o nome de 'doc1.html'.

Documento 2:

```
<HTML>
  <HEAD><TITLE> Documento 2 </TITLE></HEAD>
  <BODY>
    <CENTER><H1> Documento 2 </H1></CENTER>
  </BODY>
</HTML>
```

Salve-o com o nome de 'doc2.html'.

Documento 3:

```
<HTML>
  <HEAD><TITLE> Documento 3 </TITLE></HEAD>
  <BODY>
    <CENTER><H1> Documento 3 </H1></CENTER>
  </BODY>
</HTML>
```

Salve-o com o nome de 'doc3.html'.

Agora, digite o seguinte código, salvando-o como 'frames1.html':

```
<HTML>
  <HEAD>
    <TITLE> Frames 1 </TITLE>
  </HEAD>
  <FRAMESET ROWS=40%, 60%>
    <FRAME SRC="doc1.html">
      <FRAMESET COLS=*,*>
        <FRAME SRC="doc2.html">
          <FRAME SRC="doc3.html">
        </FRAMESET>
      </FRAMESET>
  </FRAMESET>
</HTML>
```

Abra o arquivo 'frames1.html' e visualize o resultado.

Interligação de frames

Uma aplicação interessante do recurso de frames é a criação de páginas que exibam, na mesma tela, uma lista de itens permanente e o conteúdo do item selecionado. Para fazer isto precisaremos de dois frames, um para exibir a lista e outro, os conteúdos.

Os frames devem receber nomes, através do parâmetro NAME do comando FRAME, para que o Browser possa abrir os documentos no local correto.

No campo do cabeçalho do documento que contém os links, deve ser usada a tag <BASE TARGET="nome do frame">, contendo o nome do frame que abrirá os conteúdos dos itens. Pode-se também utilizar o parâmetro TARGET do comando A para este fim. A sintaxe utilizada neste caso seria:

```
<A HREF="nome ou URL do documento" TARGET="nome da janela" > Texto do link </A>.
```

Para facilitar, utilizaremos em nosso exemplo as páginas 'doc1.html', 'doc2.html' e 'doc3.html' criadas para o exemplo anterior. Entretanto, precisaremos criar o documento que funcionará como menu.

Digite o código a seguir e salve-o como 'menu.html':

```
<HTML>
<HEAD><TITLE> Menu </TITLE>
<BASE TARGET="frame2"></HEAD>
<BODY>
<H2> Lista de itens: </H2>
<OL>
<LI> <A HREF="doc1.html"> Link1 </A>
<LI> <A HREF="doc2.html"> Link2 </A>
<LI> <A HREF="doc3.html"> Link3 </A>
</OL>
</BODY>
</HTML>
```

Agora, vamos criar a estrutura dos frames:

```
<HTML>
<HEAD><TITLE> Frames 2 </TITLE></HEAD>
<FRAMESET COLS="0%, 70%">
<FRAME NAME="frame1" SRC="menu.html">
<FRAME NAME="frame2" SRC="doc1.html">
</FRAMESET>
</HTML>
```

Salve com o nome de 'frames2.html'.

Abra o arquivo 'frames2.html' e teste os links do frame da esquerda.

Aprecie o efeito na tela.

O marcador NOFRAMES

Como foi dito no início deste capítulo, o recurso de frames não é compatível com todos os browsers. Portanto, se algum usuário acessar uma página que tenha frames com um Browser que não os reconheça, nada será exibido.

Para solucionar este problema existe o comando NOFRAMES, que permite a inclusão de avisos ou comandos HTML que serão visualizados pelo browser não compatível com frames.

As marcações <NOFRAMES> e </NOFRAMES> são definidas dentro da estrutura do campo de atuação do comando FRAMESET. Entre elas podem ser especificados quaisquer comandos HTML, que só serão executados se o frameset não puder ser carregado pelo Browser.

Considerando a hipótese de lidar com Browsers que não reconheçam frames, sugere-se as seguintes alternativas ao criar sua página:

1. Use o comando NOFRAMES e emita um aviso de que o Browser usado não aceita frames e aproveite para sugerir endereços para que o usuário possa 'baixar' versões mais novas.
2. Crie uma página inicial que permita ao usuário escolher entre visualizar uma versão com frames e outra sem frames.
3. Use o comando NOFRAMES para carregar automaticamente uma versão sem frames do seu site.

O exemplo abaixo é uma modificação do arquivo 'frames2.html', utilizando o comando NOFRAMES. Modifique-o também em seu computador e salve com o nome de 'frames3.html':

```
<HTML>
<HEAD><TITLE> Frames 2 </TITLE></HEAD>
<FRAMESET COLS="30%, 70%">
<FRAME NAME="frame1" SRC="menu.html">
<FRAME NAME="frame2" SRC="item1.html">
<NOFRAMES>
```

Esta página usa frames, um recurso não suportado pelo Browser que está sendo utilizado.
Sugerimos obter uma versão atualizada do Netscape Navigator ou do Internet Explorer.

```
</NOFRAMES>
</FRAMESET>
</HTML>
```

Janelas especiais para frames

Para dominar completamente o uso de frames, é importante conhecer a finalidade de quatro nomes especiais de janelas que são predefinidos.

Usando estes nomes no parâmetro TARGET do comando A, o Browser irá carregar o documento em determinadas janelas.

TARGET="_blank" - Faz com que uma nova janela seja aberta para carregar o documento.

TARGET="_self" - Faz com que o documento seja aberto na mesma janela em que está o link.

TARGET="_parent" - Faz com que o documento seja carregado no frameset de nível superior ao do documento atual.

TARGET="_top" - Faz com que o documento seja carregado usando a janela inteira do Browser, ocultando, assim, a exibição dos frames do documento.

Imagens Mapeadas

Mapas de imagem (ou image maps) são figuras colocadas na página que, ao serem clicadas em regiões específicas, permitem o acesso a outros documentos.

É como se determinadas partes da imagem funcionassem como links.

As imagens mapeadas podem ser criadas através de programas denominados mapeadores de imagem, tal como PaintBrush.

Os arquivos gerados por este tipo de programa (os mapeadores) possuem a extensão '.map', mas alguns possuem a opção de criar diretamente um arquivo HTML que contém apenas a área de definição do mapa.

Nesta apostila, entretanto, não estaremos abordando a criação de mapas através de programas mapeadores especiais. Usaremos o próprio paintbrush do Windows para isso.

Existem dois tipos de mapas de imagem. O primeiro é o server-side (lado do servidor) e, o segundo, o client-side (lado do cliente).

Mapas do tipo server-side

Neste tipo de mapa de imagem, fica localizado no servidor.

Quando o usuário clica na imagem, o Browser envia uma solicitação ao servidor perguntando o endereço (URL) do documento associado à área clicada.

O servidor envia, então, o endereço de volta ao Browser que lhe retorna uma solicitação para abrir aquele documento.

Não entraremos em detalhes sobre o funcionamento de mapas do tipo server-side, pois seu mecanismo não envolve diretamente programação HTML.

Basta saber que, para inserir um mapa deste tipo num documento, é necessário criar uma âncora da imagem para o programa que interpreta mapas no servidor com a identificação do mapa associado a esta imagem.

A sintaxe utilizada para isso é:

``.

O parâmetro ISMAP do comando IMG indica que a imagem é do tipo clicável, ou seja, uma imagem mapeada.

Mapas do tipo client-side

A definição deste tipo de mapa é feita inteiramente no documento HTML que o contém. Esta definição pode ser feita totalmente a mão ou utilizando recursos de programas mapeadores.

Nesta apostila trataremos apenas da definição à mão através de comandos da linguagem HTML.

O marcador MAP

Cada imagem inserida como mapa precisa ter comandos de mapeamento associados a ela. Tais comandos correspondem à definição do mapa propriamente dita.

Todos os comandos de uma imagem mapeada como client-side devem ser envolvidos pelas marcações <MAP> e </MAP>.

O principal parâmetro do comando MAP é NAME="nome". Ele atribui à definição de um mapa, um nome único que será referido pelo parâmetro USEMAP do comando IMG utilizado para colocar a imagem na página.

Sintaxe básica:

<MAP NAME="nome do mapa"> [Definição de áreas sensíveis] </MAP>

Para chamar um mapa de imagem client-side através do comando IMG é usada a sintaxe: . Note que o símbolo # deve anteceder o nome do mapa.

O marcador AREA

A marcação <AREA> é utilizada para especificar as áreas sensíveis, ou seja, que podem ser clicadas, de um mapa de imagem.

Seus principais parâmetros são:

SHAPE="forma" - Determina o formato da área sensível. Este formato será usado para a especificação das coordenadas e pode receber os seguintes valores:

RECT (retângulo) - Exige quatro coordenadas, x1, y1, x2 e y2. As coordenadas x1 e y1 correspondem ao vértice superior esquerdo do retângulo, enquanto que x2 e y2 correspondem ao vértice inferior direito.

CIRC (círculo) - Exige três coordenadas xC, yC e r. As coordenadas xC e yC correspondem ao centro do círculo (posição X e Y), e a coordenada r ao seu raio.

POLY (polígono) - Exige um par de coordenadas para cada vértice do polígono descrito. No caso, xN e yN corresponderiam à localização do vértice N do polígono.

DEFAULT - Não exige coordenada nenhuma. Pode ser usada para definir uma URL que será carregada se o usuário clicar sobre uma área da imagem que não foi mapeada.

COORDS="coordenadas" - As coordenadas dependem diretamente do valor do parâmetro SHAPE, conforme explicitado acima, e seus valores são medidos em pixels. Num mapa retangular de área NxM, sendo N sua largura e M sua altura em pixels, a coordenada (x=0,

y=0) corresponde ao seu vértice superior esquerdo, enquanto que (x=N, y=M) correspondem ao seu vértice inferior direito.

HREF="nome ou URL do arquivo" - Especifica o caminho para o documento a que determinada região da imagem levará, se clicada.

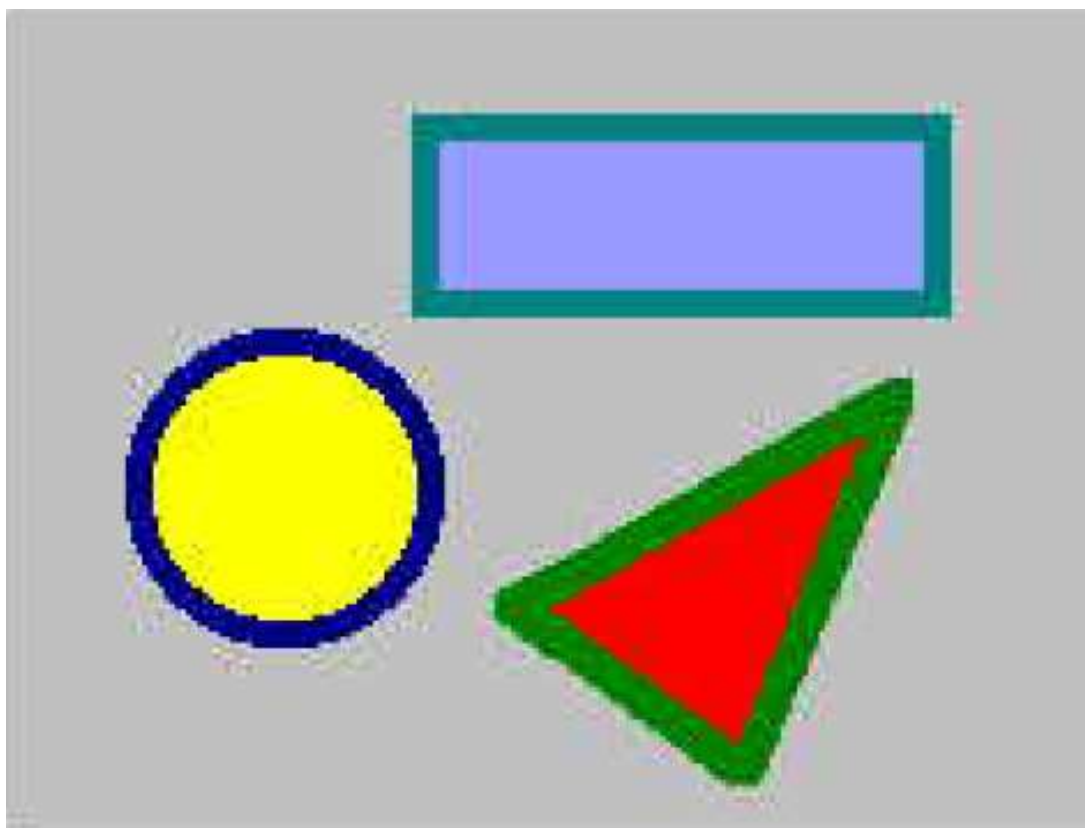
TARGET="janela" - Especifica o nome da janela onde o documento deve ser carregado. Pode receber os seguintes valores:

- **_blank** - Abre o documento em uma nova janela
- **_parent** - Abre o documento na janela de nível imediatamente superior (principal)
- **_self** - Abre o documento na própria janela do link.
- **_top** - Abre o documento na janela de nível mais alto e ocupando toda a área.

ALT="texto" - Exibe o texto especificado quando o cursor passa por cima da área.

Para exemplificar a criação de um mapa de imagem, considere a imagem seguinte, que possui um retângulo, um círculo e um triângulo.

Vamos fazer com que cada uma dessas figuras, ao ser clicada, leve a um documento HTML.



Crie no paint esta imagem. Marque suas coordenadas e salve-a como 'mapa.gif'.
Digite os documentos à seguir:

Documento 1:

```
<HTML>
<HEAD><TITLE> Documento 1 </TITLE></HEAD>
<BODY>
<CENTER><H1> Você clicou sobre o retângulo!</H1></CENTER>
</BODY></HTML>
```

Salve-o com o nome de 'doc-ret.html'.

Documento 2:

```
<HTML>
<HEAD><TITLE> Documento 2 </TITLE></HEAD>
<BODY>
<CENTER><H1> Você clicou sobre o círculo!</H1></CENTER>
</BODY></HTML>
```

Salve-o com o nome de 'doc-circ.html'.

Documento 3:

```
<HTML>
<HEAD><TITLE> Documento 3 </TITLE></HEAD>
<BODY>
<CENTER><H1> Você clicou sobre o triângulo!</H1></CENTER>
</BODY></HTML>
```

Salve-o com o nome de 'doc-tri.html'.

Documento 4:

```
<HTML>
<HEAD><TITLE> Mapa de imagem </TITLE></HEAD>
<BODY>
<CENTER>
<H1> Clique sobre as figuras abaixo: </H1>
<IMG SRC="mapa.gif" USEMAP=#mapa>
</CENTER>
<MAP NAME=mapa>
<AREA SHAPE=rect COORDS="74,19,170, 55" HREF="doc-ret.html" >
<AREA SHAPE=circ COORDS="50,86,30" HREF="doc-circ.html">
<AREA SHAPE=poly COORDS="87,107,154,66,134,142"
HREF="doc-tri.html">
</MAP>
</BODY></HTML>
```

Salve-o como 'mapa.html'.

Execute-o e observe os resultados !