



HTML

ÍNDICE



INTRODUÇÃO

1. O QUE É A LINGUAGEM HTML?
2. O QUE É UM DOCUMENTO HIPERTEXTO?
3. ESTRUTURA GERAL DE UM DOCUMENTO HTML
4. CABEÇALHOS
5. PARÁGRAFOS E QUEBRAS DE LINHA
6. FORMATANDO TEXTOS
 - 6.1. Tags Físicas de Formatação
 - 6.2. Tags Lógicas de Formatação
 - 6.3. Alterando o tamanho da fonte
7. CARACTERES ESPECIAIS
8. LISTAS
 - 8.1. Listas Ordenadas
 - 8.2. Listas Não Ordenadas
 - 8.3. Listas de Definição
9. *LINKS DE HIPERTEXTO*
10. *URLS*
11. LINKANDO SEÇÕES DE UMA PÁGINA
 - 11.1. Criando uma âncora
 - 11.2. Criando um *link* para a âncora
 - 11.3. Linkando arquivos locais
 - 11.4. Linkando arquivos de outros diretórios
 - 11.5. Linkando arquivos de outros computadores
12. IMAGENS
 - 12.1. Opções importantes da tag
 - 12.2. Usando imagens como *links*
 - 12.3. *Links* para imagens
13. CORES E PADRÕES DE FUNDO
 - 13.1. Backgrounds com imagens
14. REFINANDO A PÁGINA
 - 14.1. Alinhando parágrafos
 - 14.2. Alinhando o texto com a tag <DIV>
 - 14.3. Alinhando cabeçalhos
 - 14.4. Bordas de imagens usadas como *links*
 - 14.5. Melhorando as Listas Ordenadas



- 15. TABELAS
 - 15.1. Criando uma tabela simples
 - 15.2. Alterando as bordas da tabela



- 16. TÓPICOS AVANÇADOS
 - 16.1 Frames
 - 16.2 Folhas de Estilo
 - 16.3 Formulários



- 17. HOSPEDAGEM
 - 17.1 Como hospedar
 - 17.2 Utilização de FTP Cliente

GLOSSÁRIO



INTRODUÇÃO



Esta apostila tem como um de seus objetivos, servir de apoio ao curso **HTML** oferecido pela Seção Técnica de Informática da Faculdade de Filosofia, Letras e Ciências Humanas da Universidade de São Paulo, e também dar um embasamento técnico a um estudo mais detalhado sobre o assunto. Outro objetivo é fornecer uma visão geral dos recursos mais importantes, permitindo que o usuário possa iniciar o uso desta linguagem.

Assumimos que o usuário tenha conhecimentos básicos sobre **Windows** e **Internet**. A intenção é mostrar, através de exemplos, usos práticos de **HTML**. Vale lembrar que um melhor aproveitamento, demanda por parte do usuário um grande empenho para aprofundar-se nas peculiaridades da linguagem em estudos individuais.



1. O QUE É A LINGUAGEM HTML?



HTML é a sigla da expressão **HyperText Markup Language** que traduzindo significa **Linguagem de Marcadores de Hipertexto**. É uma linguagem de formatação muito simples, utilizada para criar **documentos hipertexto**. Hipertexto quer dizer, em outras palavras, além do texto, ou seja, além do texto o documento pode ter imagens, tabelas, formatações especiais, links etc., que tornam o documento muito mais atraente.

Um **documento hipertexto** é constituído de trechos de texto, muito parecidos com aqueles que você produz num **editor de textos**. Na verdade, se quiser produzir páginas codificadas em **HTML**, basta utilizar um editor de textos simples, como o Bloco de Notas do Windows.

Porém existem programas sofisticados que "constroem" todo o código **HTML** de forma simples e rápida. Exemplos desses programas são o **Microsoft Front Page** e o **Macromedia Dream Weaver**. Existem outras vertentes da linguagem **HTML** como por exemplo **DHTML**, **SGML**, **XML** etc., mas o foco deste material é específico à linguagem **HTML**.

Diferentemente de outras linguagens como C e Pascal, a linguagem **HTML** não é compilada e sim interpretada, e no caso pelo *browser*. Teoricamente só podemos perceber algum erro com o código **HTML** quando o arquivo que contém este código for aberto, enquanto as outras linguagens possuem um recurso de compilação que verifica no código, erros de digitação, entre outros erros.

2. O QUE É UM DOCUMENTO HIPERTEXTO?



O que diferencia um **documento hipertexto** de um documento comum? É a possibilidade, entre outras, de “ligar” uma palavra ou um trecho de texto com outras partes desse ou de outros documentos. Esta ligação é chamada **link de hipertexto** ou, simplesmente, **link**. Ao dar um clique em um **link**, o navegador irá se encarregar de desviar sua leitura para um novo trecho do documento, correlacionado com o assunto apontado por essa ligação.

A ligação pode ser feita com um outro arquivo **HTML**, com um arquivo de imagem, com um arquivo de som ou qualquer outro tipo de arquivo. Esses arquivos podem estar no próprio computador ou em qualquer computador em qualquer parte do mundo.

Outra diferença entre um documento hipertexto e um documento comum é justamente a possibilidade de se ter no documento, além do texto, imagem, cores, sons, vídeos, animações etc. Um outro exemplo de arquivos hipertexto, além dos arquivos **HTML**, são os arquivos de Ajuda do Windows. Embora o formato seja diferente (estes arquivos possuem extensões .chm, .hlp etc) os mesmos são arquivos hipertextos também.

3. ESTRUTURA GERAL DE UM DOCUMENTO HTML

Todo documento **HTML** deve começar pela declaração de si mesmo. Você faz isto utilizando a tag `<HTML>`, colocada na primeira linha do arquivo. Tag é o código **HTML** propriamente dito, ou seja, tudo que estiver entre os sinais de menor (`<`) e maior (`>`) será chamado de tag. A última linha do documento será a tag `</HTML>`. Assim pode-se perceber que toda tag contenedora utiliza a barra “/” para sinalizar o fechamento do trecho que está sendo formatado. Tudo mais que a página terá, deve ser colocado entre as duas tags informadas anteriormente.

Geralmente todo documento tem duas divisões: o **cabeçalho** e o **corpo**. O início do cabeçalho é definido pela tag `<HEAD>` e o fim é definido pela tag `</HEAD>`. O início do corpo é definido pela tag `<BODY>` e o fim pela tag `</BODY>`.

No cabeçalho existe uma tag, entre outras, que **sempre** deverá ser inserida: trata-se da tag que marca o título da página. O título será contido pelo par `<TITLE>` `</TITLE>`. O título de um documento não é mostrado na página mas os **browsers** o utilizam para titular a janela de visualização. No entanto, o uso mais importante do título é nos **bookmarks** do **browser**. Lembre-se de que os **bookmarks** são preciosos para todo internauta. Daí, nunca se esqueça de colocar um título bem claro e explicativo para seus documentos.

No corpo é que será visualizada na página propriamente dita, com todo texto, imagens, tabelas e formatações que forem definidas pelo código. Assim, um documento típico seria codificado da seguinte maneira:

```
<HTML>
<!--**** Isto é um comentário **** -->
<HEAD>
  <TITLE>Meu primeiro documento HTML</TITLE>
</HEAD>
<BODY>
<!--**** Isto é um outro comentário **** -->
  Textos e mais textos, imagens, tabelas etc.
</BODY>
</HTML>
```

Lembraremos três coisas:

- Podemos colocar comentários em qualquer parte do programa, para isso basta colocarmos o mesmo dentro de `<!-- Aqui um comentário... -->`.
- Espaços em branco são ignorados pelo interpretador **HTML**, portanto o código acima poderia ter sido escrito da seguinte forma:

```
<HTML><!-- **** Isto é um comentário ****--><HEAD><TITLE>Meu primeiro
documento HTML</TITLE></HEAD><BODY><!--**** Isto é um outro comentário
**** -->Textos e mais textos, imagens, tabelas etc...</BODY></HTML>
```

Isto certamente economiza espaço em disco, porém, trata-se de uma maneira muito prolixa de programar e muito complexa de se entender.

- As *tags* **HTML** não são sensíveis à caixa, ou seja, tanto faz escrever `<HTML>`, `<HtmL>`, `<html>`, `<HtMl>` etc. Como sugestão escreva todas as tags em caixa alta (letras maiúsculas).

O corpo do documento é todo texto e imagens, hiperlinkados ou não, contidos entre as tags `<BODY>` e `</BODY>`. É a página propriamente dita, aquilo que será mostrado na tela do monitor.

Veremos, aqui, como formatar o seguinte:

- Textos de cabeçalho ou de titulamento (*headings*)



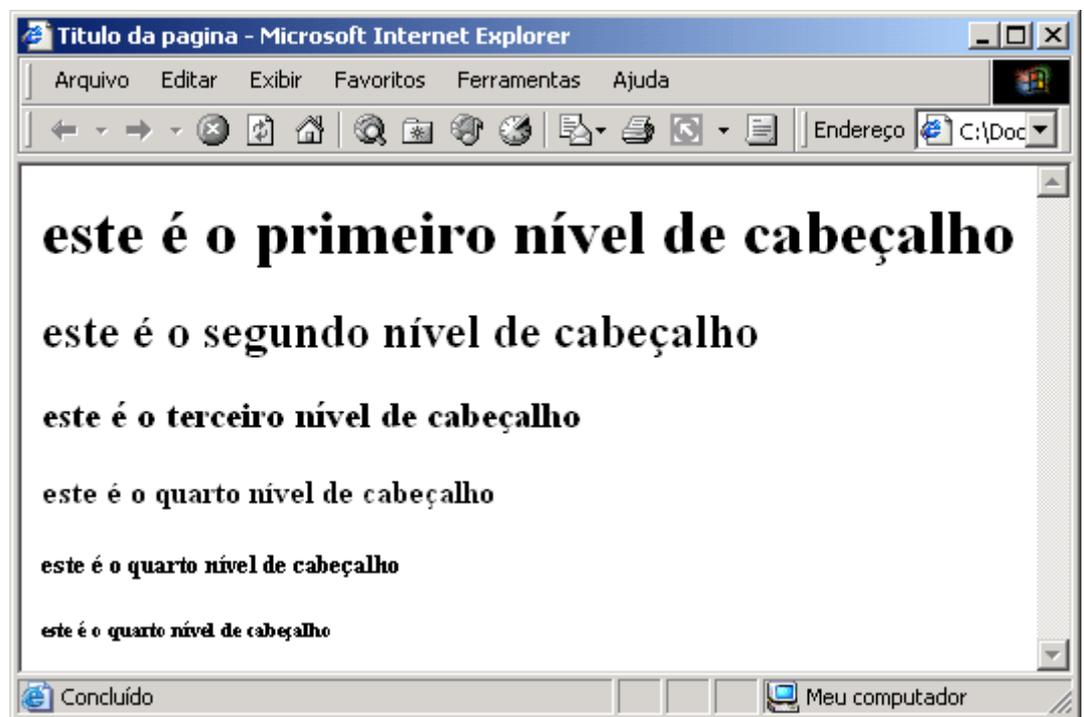
- Parágrafos de texto
- Listas de itens (*lists*)
- Listas de definição (*definition lists*)
- Links e âncoras de hipertextos (*links*)
- Imagens (*images*)
- Links utilizando imagens

4. CABEÇALHOS

Apesar de não ser exigido, é uma boa prática escrever seus documentos de forma que os “níveis” dos tópicos e subtópicos reflitam sua organização. Por exemplo, o primeiro tópico deveria ter “nível 1”, subtópicos começariam com “nível 2” e assim por diante.

A tag de cabeçalho é definida pelo par `<Hn> . . . </Hn>` onde n representa seu nível, que varia de 1 até 6. Exemplo:

```
<HTML>
<HEAD>
<TITLE>Titulo da pagina</TITLE>
</HEAD>
<BODY>
<H1> este é o primeiro nível de cabeçalho</H1>
<H2> este é o segundo nível de cabeçalho</H2>
<H3> este é o terceiro nível de cabeçalho</H3>
<H4> este é o quarto nível de cabeçalho</H4>
<H5> este é o quinto nível de cabeçalho</H5>
<H6> este é o sexto nível de cabeçalho</H6>
</BODY>
</HTML>
```



As tags de cabeçalhos inserem, além do texto, uma linha em branco antes e depois do texto.

Existe, também, a tag `<HR>` que é usada para inserir uma linha horizontal no local onde foi especificada. Esta tag não precisa ser fechada, isto é, não é preciso utilizar um par `<HR> . </HR>`.

5. PARÁGRAFOS E QUEBRAS DE LINHA



Você escreve seus parágrafos de texto da mesma maneira que se estivesse usando um **editor de textos**. A única diferença reside em ter de forçar os espaçamentos entre cada parágrafo. Você faz isto usando a tag `<P>` sempre que quiser começar um novo parágrafo. Exemplo:

```
<HTML>
<HEAD>
<TITLE>Titulo da pagina</TITLE>
</HEAD>
<BODY>
<P>Este é o primeiro parágrafo
<P>Este é o segundo parágrafo
</BODY>
</HTML>
```

resultaria em:

Este é o primeiro parágrafo

Este é o segundo parágrafo

Se não tivesse colocado a tag `<P>`, o resultado seria este:

Este é o primeiro parágrafoEste é o segundo parágrafo

Ou seja, você não obterá a separação entre os dois parágrafos nem a quebra de linha. Caso queira apenas quebrar a linha, você deve inserir a tag `
`. Esta tag também não precisa ser fechada, isto é, não precisamos utilizar um par `
.</BR>`.

A linguagem **HTML** permite definir formatação de textos, físicas e lógicas. Na tela do monitor eles poderão ter o mesmo resultado. Entretanto, sua utilidade ficará mais evidente após a leitura do que se segue.



6. FORMATANDO TEXTOS

6.1. Tags Físicas de Formatação

Todas as tags são contenedores:

·. : texto em **negrito**.

·<I>. </I>: texto em *itálico*.

·<TT>. </TT>: texto escrito com alguma fonte monoespçada, geralmente Courier.

·<U>. </U>: texto sublinhado.

·[.]: eleva o texto e diminui seu corpo. Texto ^{sobrescrito}

·_.: rebaixa o texto e diminui seu corpo. Texto _{subscrito}

Exemplo:

```
<BODY>
Meu caro surfista, <B>nao se esqueca</B> de configurar as fontes de seu
<I>browser</I>, <U>antes</U> de iniciar seu trabalho. Utilize como
segunda fonte a Courier <TT>monoespçada</TT>.
</BODY>
```

resultaria em:

Meu caro surfista, **nao se esqueça** de configurar as fontes de seu *browser*, antes de iniciar seu trabalho. Utilize como segunda fonte a Courier monoespçada.

6.2. Tags Lógicas de Formatação

Existem vários, mas veremos apenas alguns deles:

<CODE> . . . </CODE>

Utilizada para relacionar o código-fonte de programas. Geralmente será visualizado em fonte monoespçada.

** . . . **

Utilizada para dar ênfase especial a um trecho de texto. Geralmente será visualizado em fonte negritada.

<DFN> . . . </DFN>

Utilizada para destacar um definição. Geralmente será visualizado em fonte *itálica*.

<CENTER> . . . </CENTER>

Utilizada para centralizar um texto na tela.

<PRE> . . . </PRE>

Permite criar textos pré-formatados semelhantes aos produzidos em **editores de texto**. Espaços em branco, marcas de tabulação e quebras de linha são reconhecidos e colocados nos respectivos pontos de inserção. Textos pré-formatados são representados pela fonte monoespçada definida pelo usuário.

Dentro de um texto pré-formatado:

·É permitido utilizar **links** e âncoras. Entretanto, não existe certeza de como o **browser** interpretará essas marcações.



·A tag <P> não deve ser utilizado.

·Não devem ser usados outros comandos de formatação, tais como os de listas e de cabeçalhos.

Exemplo:

```
<HTML>
<HEAD>
<TITLE>Titulo da Pagina</TITLE>
</HEAD>
<BODY>
<PRE>
Diretório de arquivo EXEMPLO:
EXEMPLO1.DOC  5,760 bytes  18/09/95  2:34 pm
EXEMPLO2.DOC  13,280 bytes  15/09/95  7:45 pm
EXEMPLO3.DOC  4,562 bytes  21/08/95  6:19 am
IMAGEM1.GIF   143,218 bytes  12/07/95  10:56 pm
</PRE>
</BODY>
</HTML>
```

que resultaria na seguinte representação na tela:

```
Diretório de arquivo EXEMPLO:
EXEMPLO1.DOC  5,760 bytes  18/09/95  2:34 pm
EXEMPLO2.DOC  13,280 bytes  15/09/95  7:45 pm
EXEMPLO3.DOC  4,562 bytes  21/08/95  6:19 am
IMAGEM1.GIF   143,218 bytes  12/07/95  10:56 pm
```

<ADDRESS> . </ADDRESS>

Formata informações tais como o endereço, a assinatura e a autoria de quem publicou a página. Normalmente, estes dados são colocados no final de um documento e, geralmente, serão visualizados em itálico. Exemplo:

```
<ADDRESS>
Página atualizada em: 13/01/03.<BR>
Criada por Antonio João<BR>
Email: <A HREF="mailto:goncalves@hotmail.com.br">antonio@hotmail.com.br</A>
</ADDRESS>
```

O resultado seria:

```
Página atualizada em: 13/01/03.
Criada por Antonio João
Email: antonio@hotmail.com.br
```

6.3. Alterando o Tamanho da fonte

Um dos recursos de tratamento do texto mais interessante da linguagem **HTML** é a possibilidade de alterar o tamanho, cor e tipo da fonte do texto utilizado. A tag responsável por isso é . Exemplo:



```
<FONT SIZE=7 FACE="courier" COLOR=green>fontes e mais fontes...</FONT>
```

O resultado seria:

fontes e mais fontes...

Uma novidade vista aqui é que algumas tags aceitam opções para definir alguma característica referente à própria tag. No caso da tag `` suas opções são:

·`SIZE=n` - Esta opção especifica o tamanho da fonte utilizada;

·`FACE="name"` - Esta opção permite que seja definida uma fonte diferente para o texto;

·`COLOR=cor` - Esta opção especifica a cor do texto. Seu valor pode ser especificado pelo código hexadecimal ou, para algumas cores, pelo seu nome em inglês (blue, black, white etc.).

Uma relação com alguns códigos de cores pode ser vista no capítulo “**13. Cores e Padrões de Fundo**”.



7. CARACTERES ESPECIAIS

Não é recomendável digitar caracteres acentuados e símbolos tipográficos especiais diretamente em strings ou em valores de opções. Se fizer isto, os **browsers** podem truncá-los ou representar outros caracteres especiais. Exemplo:

Se você escrever:

```
<OL>
<LI>Este é um parágrafo
</OL>
```

o resultado na tela, usando algum navegador poderia ser:

Este / um parWgrafo

Para obter a representação correta, você terá de utilizar caracteres-entidade padronizados pela ISO. No exemplo anterior, o correto seria codificar o seguinte:

```
<LI> Este &eacute; um par&aacute;grafo</LI>
```

e o resultado na tela, seria:

Este é um parágrafo

Segue abaixo uma relação com os caracteres-entidades mais utilizados. Além desses caracteres, existem seis **caracteres de escape** importantes.

Entidade	Caracter	Entidade	Caracter
á	á	Á	Á
â	â	Â	Â
à	à	À	À
ã	ã	Ã	Ã
ç	ç	Ç	Ç
é	é	É	É
ê	ê	Ê	Ê
í	í	Í	Í
ó	ó	Ó	Ó
ô	ô	Ô	Ô
õ	õ	Õ	Õ
ú	ú	Ú	Ú
ü	ü	Ü	Ü

Entidade	Caracter
<	<
>	>
&	&
®	®
©	©
§	§

Obs.: Os **browsers** Navigator 4.0 ou superior e Internet Explorer 5.0 ou superior reconhecem os caracteres especiais sem a necessidade de colocarmos o seu código.

8. LISTAS

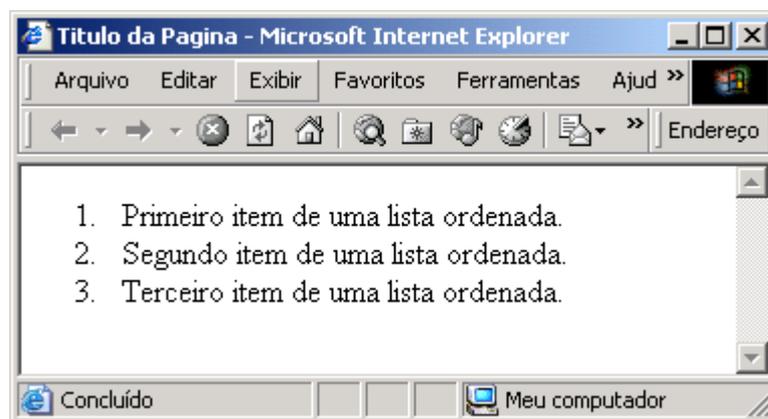
A linguagem **HTML** permite definir três tipos distintos de listas: ordenadas, não ordenadas e uma especial, chamada **lista de definição**. As **listas ordenadas** são numeradas. As **listas não ordenadas** utilizam um marcador para demarcar cada uma das suas linhas.

8.1. Listas Ordenadas

Em listas ordenadas, o **browser** se encarrega de colocar os números que referenciam cada linha. Isto facilita o trabalho do programador. Se for apagada ou inserida uma linha, em qualquer posição da lista, não é preciso se preocupar em renumerar a lista. O início de uma lista ordenada deve ser marcado pela tag `` e o fim pela tag ``. Para definir cada item da lista utilize a tag ``. Exemplo:

```
<HTML>
<HEAD>
<TITLE>Titulo da Pagina</TITLE>
</HEAD>
<BODY>
  <OL>
    <LI>Primeiro item de uma lista ordenada.
    <LI>Segundo item de uma lista ordenada.
    <LI>Terceiro item de uma lista ordenada.
  </OL>
</BODY>
</HTML>
```

Isto produz o seguinte resultado:

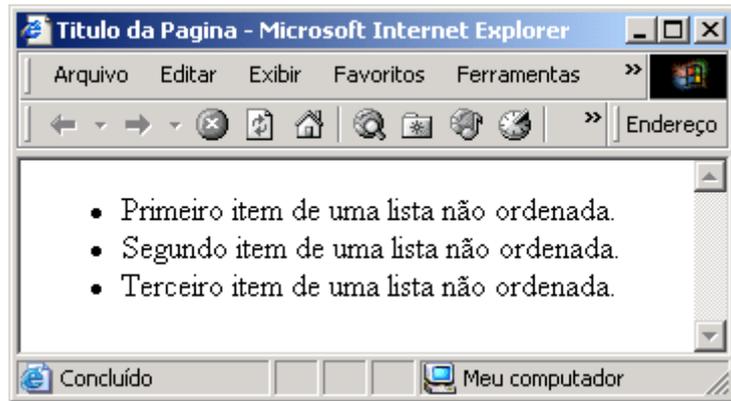


8.2. Listas Não Ordenadas

Uma lista não ordenada deve ser envolvida pelo par ` `. Exemplo:

```
<HTML>
<HEAD>
<TITLE>Titulo da Pagina</TITLE>
</HEAD>
<BODY>
  <UL>
    <LI>Primeiro item de uma lista não ordenada.
    <LI>Segundo item de uma lista não ordenada.
    <LI>Terceiro item de uma lista não ordenada.
  </UL>
</BODY>
</HTML>
```

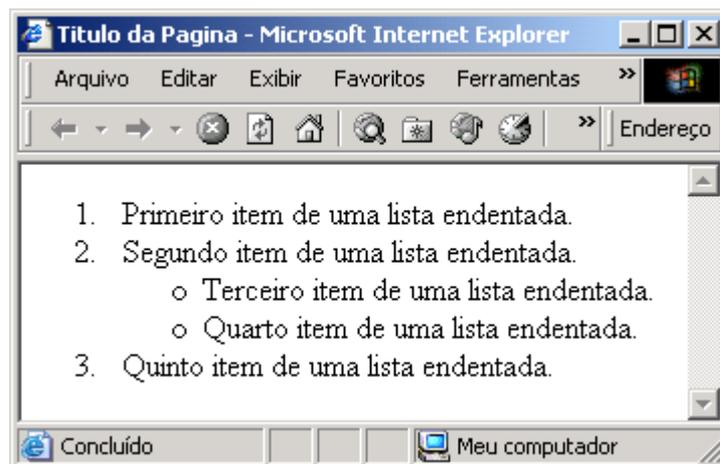
Isto produz o seguinte resultado:



O próximo exemplo produz uma lista endentada. Você pode endentar tantas listas quantas quiser, desde que envolva cada nível de endentamento com `` ou ``.

```
<HTML>
<HEAD>
<TITLE>Titulo da Pagina</TITLE>
</HEAD>
<BODY>
  <OL>
    <LI> Primeiro item de uma lista endentada.
    <LI> Segundo item de uma lista endentada.
      <UL>
        <LI> Terceiro item de uma lista endentada.
        <LI> Quarto item de uma lista endentada.
      </UL>
    <LI>Quinto item de uma lista endentada.
  </OL>
</BODY>
</HTML>
```

Isto produz o seguinte resultado:



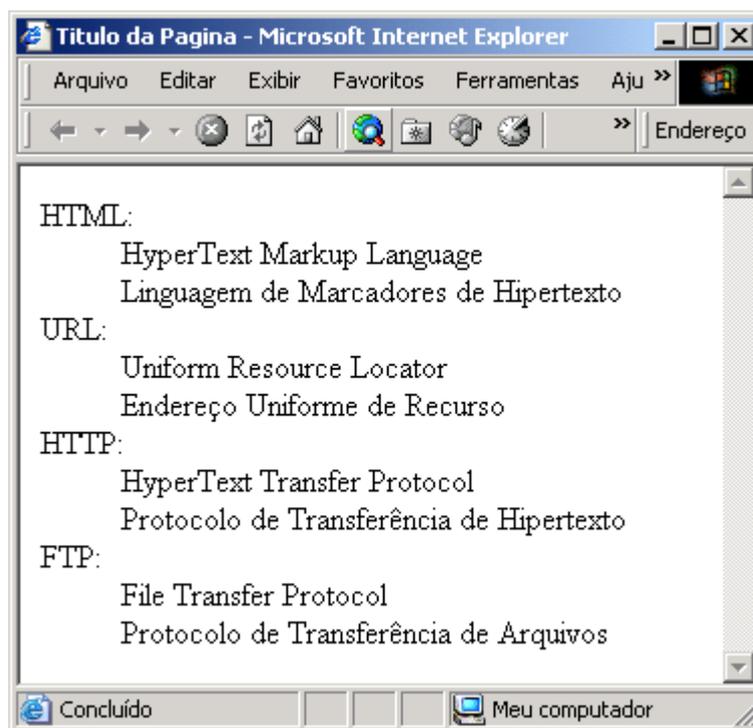
Observe que a numeração das listas obedece ao nível de encadeamento das respectivas linhas. Outra coisa interessante é que as tags de listas forçam uma quebra de linha automaticamente.

8.3. Listas de Definição

Uma lista de definição é um tipo especial de lista extremamente útil. Ela permite criar diagramações endentadas em vários níveis, sem que sejam colocados símbolos especiais na frente de cada uma de suas linhas. Exemplo:

```
<HTML>
<HEAD>
<TITLE>Titulo da Pagina</TITLE>
</HEAD>
<BODY>
  <DL>
    <DT>HTML:</DT>
    <DD>HyperText Markup Language</DD>
    <DD>Linguagem de Marcadores de Hipertexto</DD>
    <DT>URL:</DT>
    <DD>Uniform Resource Locator</DD>
    <DD>Endereço Uniforme de Recurso</DD>
    <DT>HTTP:</DT>
    <DD>HyperText Transfer Protocol</DD>
    <DD>Protocolo de Transferência de Hipertexto</DD>
    <DT>FTP:</DT>
    <DD>File Transfer Protocol</DD>
    <DD>Protocolo de Transferência de Arquivos</DD>
  </DL>
</BODY>
</HTML>
```

Veja como este exemplo seria reproduzido na tela:



Fica fácil perceber que uma lista de definição utiliza o par <DL> </DL> para estabelecer esta relação. Ela também permite encadeamentos. Existem dois níveis adicionais de formatação: <DT> estabelece a frase a ser definida e <DD> a definição propriamente dita.

9. LINKS DE HIPERTEXTO



Uma das grandes vantagens da linguagem **HTML** é a possibilidade de linkar (ligar) um documento a um outro documento quer esteja este no mesmo computador ou em outro computador em qualquer lugar do mundo. Este link poder ser feito não só para outro arquivo **HTML** mas, à principio para qualquer tipo de arquivo, seja ele uma imagem, um arquivo de som, um arquivo de programa etc. A tag utilizada para criar links é `<A HREF>`. `` que basicamente tem a seguinte sintaxe:

```
<AHREF="URL do link">Texto que vai servir como link</A>
```

Exemplo:

```
<HTML>
<HEAD>
<TITLE>Titulo da Pagina</TITLE>
</HEAD>
<BODY>
<A HREF="http://www.fflch.usp.br/sti">Site da STI</A>
<A HREF="http://www.fflch.usp.br">Site da FFLCH</A>
<A HREF="http://www.fflch.usp.br"><IMG SRC="../imagem/fflch.gif"></A>
<A HREF="http://www.usp.br">Site da USP</A>
</BODY>
</HTML>
```



É possível, como no último exemplo, usar uma imagem como *link* e assim sendo quando a imagem é clicada o endereço definido será alcançado.

Praticamente é possível utilizar qualquer pedaço do arquivo **HTML** para servir como *link*, esteja ele em uma tabela, em uma lista etc.

Na linguagem **HTML** existe uma outra tag utilizada para marcar um pedaço de um documento para ser utilizado em conjunto com a tag `<A HREF>`. A tag `<A NAME>` define um local no documento atribuindo um nome para o mesmo. Exemplo:

```
<A NAME="Capitulo10">Capítulo 10 - Back up</A>
```

Assim podemos criar um link diretamente para esse pedaço do documento utilizando a tag `<A HREF>`. Exemplo:

```
<A HREF= "www.fflch.usp.br/curso/windows#Capitulo10">Capítulo 10</A>
```



10. URLS



Uniform Resource Locator (URL) é o que movimenta toda a **Internet**. Sem este recurso, seria impossível "navegar" na "Rede das Redes". **URL** é considerado o endereço de cada computador ligado na rede. Acontece que, no sentido tecnicamente exato, **URL** não é apenas um endereço e sim um protocolo de comunicação associado a um endereço virtual que identifica um documento em um computador qualquer. Eis alguns exemplos de **URL**:

<http://www1.50megs.com/odisseus>
<http://www.fflch.usp.br/sti>
<http://www.receita.fazenda.gov.br>
<http://www.estadao.com.br>



Observe que não existem espaços em branco entre as diversas partes de um endereço **URL**. Se você colocar algum espaço em branco, o **browser** vai se perder!
 Os **protocolos de Internet** mais comuns são:

- http**:- **World Wide Web** à usado para acesso às páginas **Web**;
- ftp**:- **File Transfer Protocol** à usado para transferência de arquivos;
- mailto** - usado para acesso às caixas postais eletrônicas.



Para dominar o uso do hipertexto, deve-se conhecer bem o funcionamento das duas tags da linguagem: a tag <A NAME> e a tag <A HREF>.

A maioria dos endereços dos **sites** na **Internet** segue um determinado padrão. Eles geralmente seguem o formato: **http:// + www. + nome + .com + .br** onde:

- http://** determina a forma de comunicação com o computador onde o **site** é mantido;
- www.** é padrão e em alguns poucos endereços pode não aparecer;
- nome** determina a empresa ou entidade responsável pelo **site**;
- .com** determina que o **site** é comercial e, portanto, alguns **sites** podem não ter essa indicação;
- .br** determina a sigla do país onde o **site** está localizado e se não houver essa indicação, o **site** é geralmente mantido nos Estados Unidos;

Com base nisso, se você quiser acessar o **site** da Nestlé no Brasil você digitaria <http://www.nestle.com.br>. Se quisesse acessar o **site** do Banespa, você digitaria <http://www.banespa.com.br>, e assim por diante.



11. LINKANDO SEÇÕES DE UMA PÁGINA



11.1. Criando uma âncora

Para linkar partes de uma página, você precisa criar uma âncora no início de cada seção que será referenciada por um *link*.

Uma âncora é um ponto de referência ou endereço que será acessado por um *link*. Uma âncora é usada dentro de um documento para marcar o início de uma seção do documento. Essa âncora recebe um nome que será mencionado pelo *link* que a acessará. A tag usada para criar uma âncora, como visto anteriormente, é:

```
<A NAME="nome da âncora">Texto para linkar</A>
```



11.2. Criando um *link* para a âncora

Para criar um *link* para uma âncora, você deve usar a tag `<A HREF>`, especificando o nome da âncora que deseja acessar e um texto que será convertido em hipertexto, e indicando que ele faz *link* com algum objeto. O texto especificado mudará de cor e ficará sublinhado para se destacar do resto do texto. O comando usado para criar um *link* para uma âncora é:

```
<A HREF="#nome da âncora">Texto usado como hipertexto</A>
```

O símbolo “#” avisa o *browser* para procurar o *link* no documento atual. Quando você dá um clique no hipertexto, ele exibe a parte do documento que contém a âncora, colocando-a, sempre que possível, no alto da tela.



11.3. Linkando arquivos locais

Fazer a ligação de um texto com uma página local é uma tarefa bastante simples. Você precisa apenas especificar o nome completo do arquivo que será chamado através da tag `<A HREF>`. A tag ficaria assim:

```
<A HREF="documento2.htm">Documento2</A>
```

O comando abaixo:

```
<A HREF="documento2.htm#parte5">Documento2 - Seção E</A>
```

vai um pouco mais longe e faz referência a uma âncora chamada “parte5” do documento2. Note que uma âncora é sempre identificada pelo símbolo “#”.



11.4. Linkando arquivos de outros diretórios

Você pode especificar o endereço de outros diretórios usando um esquema de caminhos parecido com o do “*DOS*”. Existe uma grande diferença entre esses caminhos. Na *Web* você deve inverter a barra de espaços, usando a barra “/” no lugar da barra “\” para separar diretórios.

Além disso, o *browser* usa como referencial o diretório atual para iniciar a busca de páginas. Você não pode especificar um caminho do *DOS*, mas sim especificar um caminho relativo ao diretório atual. Exemplo:

Para especificar um arquivo que esteja no diretório PERA, que fica abaixo do diretório XUXU, os quais se encontram ambos abaixo do diretório-raiz, você deve usar os sinais “../” para subir um nível, acessando assim a raiz e depois o caminho do diretório XUXU (endereço relativo). Neste caso, o



caminho para acessar um arquivo chamado legal.htm seria:

```
<A HREF=" ../XUXU/PERA/legal.htm">
```

Se o diretório a ser acessado estiver abaixo do diretório atual basta especificar o nome do subdiretório sem nenhum símbolo. Para acessar um arquivo chamado legal.htm que se encontra no diretório DEMO abaixo do diretório atual, o caminho para seu **link** seria:

```
<A HREF="DEMO/legal.htm">
```

11.5. Linkando arquivos de outros computadores

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

Um **URL** é composto por duas partes principais. A primeira é o protocolo **Internet** do documento e a segunda parte é o endereço do servidor e da página. A sintaxe é Protocolo://nomedoservidor/arquivo, onde:

Protocolo: é o tipo de comunicação que está sendo executada;

Nome do servidor: nome do computador que contém a página;

Arquivo: é o caminho do arquivo a ser acessado.

Para acessar a página da FFLCH, o **URL** da página é: <http://www.fflch.usp.br>. Esse endereço pode ser especificado na linha de **URL** do **browser**, para acessar diretamente a página ou então, dentro de um documento **HTML**, através da tag `<A HREF>` que cria um **link** para o **URL** especificado. Exemplos:

```
<A HREF="http://www.fflch.usp.br">Home Page da FFLCH</A>
<A HREF="ftp://www.fflch.usp.br">Home Page de ftp da FFLCH</A>
<A HREF="mailto:a.silva@usp.br">Antonio</A>
```

A inclusão de imagens nas páginas da **Web** deu uma nova dimensão para a distribuição da informação. A inclusão de uma imagem torna a informação muito mais atraente, e, como diz o ditado, uma imagem vale mais do que mil palavras.

Incluir uma imagem na sua página é uma tarefa bem fácil. Ela exige apenas que você tenha disponível a imagem que será exibida, e que ela esteja em um dos formatos aceitos pelo **browser**. O formato de arquivo gráfico padrão aceito pela **Web** é o formato **GIF**; outro formato é o **JPG**. Antes de inundar suas páginas com muitas imagens, lembre-se de que, ao usar a rede para transmitir suas imagens, a velocidade de transmissão é muito reduzida. Portanto, uma imagem de 100KB pode levar vários minutos para ser exibida, frustrando o "navegador".

Para facilitar as coisas, coloque os arquivos de imagem no mesmo diretório da página. A tag **HTML** usado para inserir uma imagem é o seguinte:

```
<IMG SRC="NomeDoArquivo.gif"> OU <IMG SRC="NomeDoArquivo.jpg">
```

Se não for especificado um **URL**, o **browser** procura a imagem no diretório atual.

12. IMAGENS



12.1. Opções Importantes da tag

Se você colocar uma imagem perto de um texto, poderá especificar o alinhamento da imagem com relação ao texto que está ao seu redor. Como padrão, a imagem é alinhada colocando-se o seu fundo (parte de baixo) na mesma linha do texto. Através da diretiva **ALIGN**, você pode alinhar o texto pelo fundo (*bottom*), que é o padrão, pelo meio (*middle*) e por cima (*top*):

```
<IMG ALIGN="top" SRC="imagem.gif">
<IMG ALIGN="middle" SRC="imagem.jpg">
<IMG ALIGN="bottom" SRC="imagem.gif">
```

Para exibir uma imagem sem texto ao seu redor, utilize a tag e depois a tag
.



12.2. Usando imagens como *links*

Assim como um texto pode ser especificado para servir de *link*, você pode usar uma imagem para servir de link. Para isso deve-se inserir uma imagem no lugar do hipertexto, isto é:

```
<A HREF="documento.htm"><IMG SRC="imagem.gif"></A>
```

12.3. Links para imagens

Colocar imagens dentro da página (*in-line*) é muito interessante. Contudo, isso provoca uma lentidão na carga da página, principalmente quando existem várias imagens e de tamanho grande. Uma técnica muito útil é a de fazer um *link* para a imagem usando o comando <A HREF>. Dessa forma, o leitor que desejar ver determinada foto dá um clique sobre sua referência e não é onerado com o tempo da carga automática das imagens de quando se usa apenas a tag IMG, por exemplo:

```
<A HREF="imagem.gif">Imagem</A>
```



13. CORES E PADRÕES DE FUNDO



O mundo é colorido, não é? Então por que usar aquele fundo cinza que aparece como padrão toda vez que a página é criada? A maioria dos **browsers** permite a utilização de cores no segundo plano da tela (*background*) ou a utilização de imagens para enfeitar o fundo da tela. Você pode também alterar a cor do texto e dos **links** de forma a criar uma página bastante diferente do padrão do **browser**.

Neste capítulo, você aprenderá a usar e alterar as opções da tag <BODY>, responsável pelas cores da página. A sintaxe da tag <BODY> com relação às cores é a seguinte:

```
<BODY BGCOLOR=#xxxxxxTEXT=#xxxxxx LINK=#xxxxxx VLINK=#xxxxxx
ALINK=#xxxxxx>
```

onde #xxxxxx deve ser substituído por um código que identifique a cor desejada para aquela opção.



BGCOLOR: é a cor do background, cujo padrão é cinza.

TEXT: é a cor do texto, e o seu padrão é preto.

LINK: é a cor do texto do **link**, e o seu padrão é azul.

VLINK: é a cor do **link** que já foi visitado e o padrão é vermelho-púrpura.

ALINK: é a cor do **link** ativo, selecionado.



Veja o código de algumas cores e lembre-se de que você poderá colocar o nome da cor, em inglês, no lugar de colocar o código:

Black (Preto) - #000000

White (Branco) - #FFFFFF

Blue (Azul) - #0000FF

Yellow (Amarelo) - #FFFF00

Green (Verde) - #00FF00

Cyan (Ciano) - #00FFFF

Red (Vermelho) - #FF0000

Magenta - #FF00FF

Dark green (Verde escuro) - #2F4F2F

Brown (Marrom) - #A62A2A



13.1. Backgrounds com imagens

Usar um *background* colorido dá outra vida à sua página. Você pode ir mais além e usar um background muito mais elaborado através do uso de pequenas imagens gravadas nos formatos **GIF** e **JPG**. Em vez de usar um código de cor, você pode especificar o nome de um arquivo de imagem na opção *background* do comando <BODY>. Essa imagem será espalhada por toda a página de forma a preencher o fundo da tela.

O uso desta técnica permitirá que você crie uma imagem em um programa de edição de imagens e a utilize na sua página. Por exemplo, você pode criar um logotipo da sua empresa, colocar o distintivo do seu clube de futebol, a foto da(o) namorada(o) etc.

Procure usar imagens com menos de 10KB de tamanho e com poucas cores para evitar que o texto fique difícil de ler. Algumas imagens fantásticas ficam horríveis como background. A sintaxe para especificar uma imagem é:





<BODY BACKGROUND="imagem.gif">

14. REFINANDO A PÁGINA



Este capítulo é uma coletânea de recursos da linguagem **HTML** associada às tags que você já aprendeu. Você verá como funcionam alguns propriedades que alteram o funcionamento padrão de uma tag, proporcionando mais flexibilidade e qualidade para a criação de páginas.

Alinhando parágrafos;

Alinhando cabeçalhos;

Tirando a borda de imagens usadas como *links*;

Melhorando as listas ordenadas.



14.1. Alinhando parágrafos

A tag `<P>` inicia um novo parágrafo, onde o texto é automaticamente alinhado pela margem esquerda da tela. Através da propriedade `ALIGN`, pode-se mudar o alinhamento de forma que o texto do parágrafo inteiro fique centralizado ou alinhado à direita. Exemplo:

```
<P ALIGN=center>Parágrafo centralizado</P>
<P ALIGN=right>Parágrafo alinhado à direita</P>
```

Parágrafo centralizado

Parágrafo alinhado à direita



14.2. Alinhando o texto com a tag `<DIV>`

A tag `<DIV>` envolve uma área do texto ou `DIV`isão que pode receber parâmetros específicos de alinhamento. Este comando é bastante usado para alinhar um grupo de parágrafos pela direita sem que seja necessário fazer o alinhamento para cada um deles. Veja um exemplo:

```
<DIV ALIGN=right>
<P>Primeiro parágrafo</P>
<P>Segundo parágrafo</P>
<P>Terceiro parágrafo</P>
</DIV>
```

O resultado na tela seria:

Primeiro parágrafo

Segundo parágrafo

Terceiro parágrafo



14.3. Alinhando cabeçalhos

Como padrão, um cabeçalho gerado pelo comando `<Hn>` é sempre posicionado do lado esquerdo da linha. Usando o parâmetro `ALIGN`, pode-se centralizar ou ajustar o cabeçalho à direita da linha. Exemplo:

```
<H2>Cabeçalho sem alinhamento especial</H2>
```

Este é o padrão quando um cabeçalho é exibido.

```
<H2 ALIGN=center>Cabeçalho com alinhamento centralizado</H2>
<H2 ALIGN=right>Cabeçalho com alinhamento à direita</H2>
```





14.4. Bordas de imagens usadas como *links*

Quando você usa uma imagem como *link*, ela fica automaticamente envolvida por uma borda com a cor padrão de *link*. Esta borda serve para indicar se o *link* foi ou não acessado, assim como um texto usado para *link*. Muitas vezes, você pode querer omitir esta borda em função do visual da página, ou até mesmo aumentá-la. Isso tudo é possível com o uso da propriedade BORDER da tag . Exemplo:

```
<A HREF="documento.htm"><IMG SRC="imagem.gif" BORDER=0>Imagem sem borda</A>
<A HREF="documento.htm"><IMG SRC="imagem.gif" BORDER=10>Imagem com borda tamanho 10</A>
```



14.5. Melhorando as Listas Ordenadas

As listas ordenadas possibilitam várias modificações. Seu padrão é sempre iniciar a lista pelo número 1. Você pode mudar o número inicial através da propriedade START e o tipo de numeração da lista usando a propriedade TYPE. Os valores permitidos para TYPE são:

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

TYPE=a - Cria uma lista alfabética com letras minúsculas

TYPE=I - Cria uma lista numérica com numerais romanos (maiúsculos)

TYPE=i - Cria uma lista numérica com numerais romanos (minúsculos)



No caso de listas alfabéticas ou numerais romanos, esse valor indica a enésima ocorrência da lista. Exemplo:

```
<OL START=25>
<LI>primeiro item
<LI>segundo item
<LI>terceiro item
</OL>
<BR>
<OL TYPE=a START=10>
<LI>primeiro item
<LI>segundo item
<LI>terceiro item
</OL>
```



O exemplo ficaria assim:

25. primeiro item
26. segundo item
27. terceiro item



j. primeiro item
k. segundo item
l. terceiro item

15. TABELAS

As tabelas têm uma estrutura parecida com a de uma planilha eletrônica. Elas são compostas por linhas e colunas cujas intersecções formam o que se chama de uma célula. Uma célula pode conter um texto simples, um hipertexto ou até mesmo uma imagem. A seguir, daremos uma olhada geral nas tags relativas a tabelas.

<TABLE> . </TABLE>

Uma tabela é criada com pelas tags <TABLE> e </TABLE>. Entre estas devem ser especificados as demais tags responsáveis pela criação das linhas e células, títulos, bordas e alinhamento da tabela. Como padrão, uma tabela é criada sem borda, ou seja, não aparecem as linhas horizontais e verticais que separam as células.

<TR> . </TR>

Abreviação de *Table Row*, esta tag é usada para criar uma linha da tabela. Se uma tabela tiver cinco linhas, deverão ser especificados cinco pares deste comando.

<TD> . </TD>

Abreviação de *Table Data*, estes comandos são usados para especificar o conteúdo de uma célula. Estes comandos devem ser especificados em pares para cada célula da linha. Eles devem ser usados entre os comandos <TR> e </TR>. Ao contrário das planilhas, uma tabela não precisa ter todas as linhas com o mesmo número de células. A largura de uma coluna é definida pela largura da maior célula que faz parte da coluna.

O conteúdo de uma célula é alinhado pela opção **ALIGN**. Como padrão, o alinhamento horizontal do texto é à esquerda (ALIGN=left) e o alinhamento vertical, no meio da célula (VALIGN=middle).

BORDER

Este parâmetro pertence ao comando <TABLE>. Se for especificado sem nenhum valor, uma linha fina é criada em volta de todas as células. O valor 0 é equivalente a não ter borda. Quanto maior o valor especificado, mais grossa é a borda.

COLSPAN

Este parâmetro pode aparecer em qualquer célula da tabela e especifica quantas colunas da tabela a célula deve ocupar. O valor padrão é 1.

ROWSPAN

Este parâmetro funciona como o anterior, só que especifica quantas linhas, para baixo, a célula deve ocupar. O valor padrão é 1.

15.1. Criando uma tabela simples

A lógica e sequência de criação de uma tabela é a seguinte:

1. Use a tag <TABLE> para especificar que uma tabela está sendo criada;
2. Use a tag <TR> . </TR> para criar uma linha da tabela;
3. Use a tag <TD> . </TD> para criar cada célula de uma linha;
4. Use a tag </TABLE> para encerrar a tabela;
5. Use as propriedades para mudar a aparência da tabela;

Exemplo:

```
<HTML>
<HEAD>
<TITLE>Titulo da Pagina</TITLE>
</HEAD>
<BODY>
```

```

<TABLE BORDER>
  <TR>
    <TD>Linha 1 Coluna 1</TD>
    <TD>Linha 1 Coluna 2</TD>
    <TD>Linha 1 Coluna 3</TD>
  </TR>
  <TR>
    <TD>Linha 2 Coluna 1</TD>
    <TD>Linha 2 Coluna 2</TD>
    <TD>Linha 2 Coluna 3</TD>
  </TR>
  <TR>
    <TD>Linha 3 Coluna 1</TD>
    <TD>Linha 3 Coluna 2</TD>
    <TD>Linha 3 Coluna 3</TD>
  </TR>
</TABLE>
</BODY>
</HTML>

```

Este código gera a seguinte tabela:



Note que, na tag <TABLE>, existe a opção **Border**. É ela que cria a borda que envolve as células da tabela. Se você não a especificar, a tabela é montada normalmente, porém sem a borda. Agora, faremos um exemplo que mostra o uso da opção **COLSPAN**, isto é, faz com que a célula ocupe colunas adjacentes. Note que o conteúdo desta célula está centralizado pelo comando ALIGN.

```

<HTML>
<HEAD>
<TITLE>Titulo da Pagina</TITLE>
</HEAD>
<BODY>
<TABLE BORDER>
  <TR>
    <TD>Linha 1 Coluna 1</TD>
    <TD ALIGN=center COLSPAN=2>Linha 1 Coluna 2-3</TD>
    <TD>Linha 1 Coluna 4</TD>
  </TR>
  <TR>
    <TD>Linha 2 Coluna 1</TD>
    <TD>Linha 2 Coluna 2</TD>
    <TD>Linha 2 Coluna 3</TD>
    <TD>Linha 2 Coluna 4</TD>
  </TR>
  <TR>
    <TD>Linha 3 Coluna 1</TD>
    <TD>Linha 3 Coluna 2</TD>
    <TD>Linha 3 Coluna 3</TD>
    <TD>Linha 3 Coluna 4</TD>
  </TR>

```

```
</TABLE>
</BODY>
</HTML>
```

Este código gera a seguinte tabela:



O próximo exemplo, mostra o uso da opção **ROWSPAN** para aumentar o tamanho de uma célula e fazer com que ela ocupe linhas adjacentes. Note que o conteúdo desta célula está alinhado pelo comando VALIGN=top.

```
<HTML>
<HEAD>
<TITLE>Titulo da Pagina</TITLE>
</HEAD>
<BODY>
<TABLE BORDER>
  <TR>
    <TD>Linha 1 Coluna 1</TD>
    <TD>Linha 1 Coluna 2</TD>
    <TD>Linha 1 Coluna 3</TD>
  </TR>
  <TR>
    <TD>Linha 2 Coluna 1</TD>
    <TD rowspan=2>Linhas 2-3 Coluna 2</TD>
    <TD>Linha 2 Coluna 3</TD>
  </TR>
  <TR>
    <TD>Linha 3 Coluna 1</TD>
    <TD>Linha 3 Coluna 3</TD>
  </TR>
</TABLE>
</BODY>
</HTML>
```

Veja o efeito desse código:



15.2. Alterando as bordas da tabela

A criação de bordas pode ser útil em muitas tabelas, porém elas não são necessárias em muitos casos. Você viu até agora várias tabelas que usaram uma borda gerada pela edição da opção **BORDER** ao comando <TABLE>.

O próximo exemplo, mostra três tabelas onde são especificados os valores 10 e 5 para a borda e outra em que nenhum valor para borda foi especificado.

```
<HTML>
<HEAD>
<TITLE>Titulo da Pagina</TITLE>
</HEAD>
<BODY>
<TABLE BORDER=10>
  <TR>
    <TD>Linha 1 Coluna 1</TD>
    <TD>Linha 1 Coluna 2</TD>
  </TR>
  <TR>
    <TD>Linha 2 Coluna 1</TD>
    <TD>Linha 2 Coluna 2</TD>
  </TR>
  <TR>
    <TD>Linha 3 Coluna 1</TD>
    <TD>Linha 3 Coluna 2</TD>
  </TR>
</TABLE>

<TABLE BORDER=5>
  <TR>
    <TD>Linha 1 Coluna 1</TD>
    <TD>Linha 1 Coluna 2</TD>
  </TR>
  <TR>
    <TD>Linha 2 Coluna 1</TD>
    <TD>Linha 2 Coluna 2</TD>
  </TR>
  <TR>
    <TD>Linha 3 Coluna 1</TD>
    <TD>Linha 3 Coluna 2</TD>
  </TR>
</TABLE>

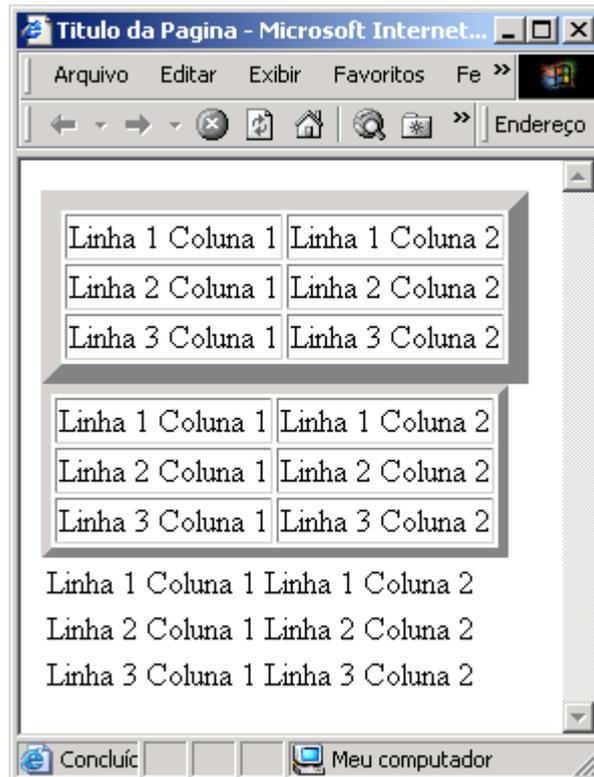
<TABLE>
  <TR>
    <TD>Linha 1 Coluna 1</TD>
    <TD>Linha 1 Coluna 2</TD>
  </TR>
  <TR>
    <TD>Linha 2 Coluna 1</TD>
```

```

<TD>Linha 2 Coluna 2</TD>
</TR>
<TR>
<TD>Linha 3 Coluna 1</TD>
<TD>Linha 3 Coluna 2</TD>
</TR>
</TABLE>
</BODY>
</HTML>

```

Veja o resultado:



Você pode criar uma tabela sem bordas para montar um menu onde as opções não fiquem unicamente na vertical, como é o caso quando você usa os comandos para a criação de listas.

17. TÓPICOS AVANÇADOS

17.1. FRAMES

Também conhecidos como molduras ou quadros, permitem a divisão da área de visualização dos navegadores em diversas partes diferentes. Em cada uma dessas áreas é possível exibir uma página **HTML** diferente. Quando bem utilizados, viabilizam a organização das informações que serão disponibilizadas em um *website*. As principais vantagens de uso são:

1. Permitem estruturar melhor a navegação em um website;
2. Possibilitam evitar ou reduzir a recarga de imagens e logotipos;
3. Propiciam manutenções mais organizadas;
4. Permitem mascarar os endereços dos links na barra de status.

Criando Quadros

Para se trabalhar com frames, é necessário criar uma página **HTML** especial, que informará em quantos quadros a tela será dividida, qual o tamanho terá cada um e qual o conteúdo será visto em cada quadro. A estrutura dessa página especial é um pouco diferente de uma estrutura comum. No lugar das tags `<BODY>`, `<BODY>` aparecem:

```
<HTML>
  <HEAD>
    <TITLE> MOLDURAS (FRAMES) </TITLE>
  </HEAD>
  <FRAMESET atributo1 atributo2 ...>
    <FRAME src="página1.html" atributo1 atributo2 ...>
    <FRAME src="página2.html" atributo1 atributo2 ...>
    ...
  </FRAMESET>
</HTML>
```

Tag Frameset

Essa tag é responsável pela divisão da área da tela dos navegadores em colunas ou linhas. Ela possui três atributos:

Cols: Divide a área em quadros posicionados horizontalmente. Deve ser preenchido com a altura de cada quadro em pixels, percentual ou com tamanhos relativos. Exemplos:

```
<FRAMESET COLS="20%, 20%, *">
```

Divide a tela em três colunas: duas com vinte por cento do espaço da tela e a última com o restante do espaço (sessenta por cento, no caso).

```
<FRAMESET COLS="20, 20%, *">
```

Divide a tela em três colunas: uma com 20 pixels de tamanho, outra com 20% do espaço e a terceira com o restante.

```
<FRAMESET COLS="2*, *">
```

Divide a tela em dois quadros: um com o dobro do tamanho do outro (equivalente a um frame ocupando 33,33% e outro 66,66% do espaço da tela).

Rows: Usado como o atributo Cols, só que dispõe em quadros verticais. Exemplos:

```
<FRAMESET ROWS="10%, 50, *">
```

Serão criados três quadros, um com o tamanho de 10% da tela, outro com tamanho de 50 pixels e

o último com o restante do espaço.

Framespacing: Define um afastamento, em pixels, entre os quadros

```
<FRAMESET ROWS="20%, *" FRAMESPACING=10>
```

Serão criados dois quadros, um com o tamanho de 20% da tela e outro com 80%. Entre eles haverá uma borda de 10 pixels.

Tag Frame

Essa tag é responsável pela apresentação do conteúdo de cada quadro. Deverá haver uma tag <FRAME> para cada quadro criado. É usada com as seguintes propriedades:

Name: atribui um nome ao quadro para poder ser identificado e utilizado como alvo para inserção de outras páginas com o atributo target.

Src: define o endereço da página que será mostrada no quadro.

Noresize: estabelece que o quadro não poderá ter o tamanho reajustados pelo usuário.

Scrolling: Mostra ou esconde as barras de rolagem. As opções possíveis são: **auto** insere caixas de rolagem vertical e/ou horizontal cada vez que o tamanho do quadro; **yes** insere sempre as barras de rolagem no quadro; **no** nunca insere caixas de rolagem, independentemente do conteúdo do quadro.

Frameborder: Define se uma borda (moldura) será mostrada ou não na junção entre dois quadros diferentes. Os valores possíveis são: 1, mostra a borda; 2, não mostra.

Marginwidth: Define a largura da margem esquerda e direita de um quadro em pixels.

Marginheight: Define a altura da margem superior e inferior de um quadro em pixels.

Exemplos:

```
<FRAME NAME="menuvertical" SRC="http:www.xxx.com.br" FRAMEBORDER=0>
```

Carrega uma página da Web, nomeando esse quadro de menuvertical e não mostra a borda.

```
<FRAME SRC="TESTE.HTML" SCROLLING=auto NORESIZE MARGINWIDTH=10
MARGINHEIGHT=10>
```

Carrega a página teste.html, insere caixas de rolagem vertical e horizontal, se necessitar, não dá a possibilidade do usuário redimensionar o tamanho do quadro e define as margens superiores, inferiores, da esquerda e da direita em 10 pixels.

Frames aninhados

É possível aninhar os comandos para subdividir uma linha em colunas ou vice-versa. O exemplo listado a seguir mostra como dividir a tela em três quadros verticais, sendo que o do meio deverá ser subdividido em dois outros quadros horizontais:

```
<FRAMESET COLS="33%, 33%, 34%">
  conteúdo do primeiro quadro vertical
  <FRAMESET ROWS="50%, 50%">
    conteúdo do segundo quadro vertical, primeira linha
    conteúdo do segundo quadro vertical, segunda linha
  </FRAMESET>
  conteúdo do terceiro quadro vertical
</FRAMESET>
```

Tag Noframes

Os navegadores mais antigos não suportam a navegação com o uso de quadros. Se uma página

com quadros é carregada por um navegador que não os suporta, não será possível visualizar corretamente o seu conteúdo.

Para contornar esse obstáculo, coloca-se a tag `<NOFRAMES>` que possibilita a inserção de avisos e até páginas inteiras dentro das páginas que criam a estrutura de quadros.

```
<HTML>
  <FRAMESET ...>
    <FRAME ...>
    ...
  </FRAMESET>
  <NOFRAMES>
  <BODY ...>
    ...
    conteúdo da página sem quadros
    ...
  </BODY>
</HTML>
```

O exemplo acima mostra a utilização geral que inclui uma página sem frames.

17.2. FOLHAS DE ESTILO

São formatações de textos e cores que são definidos para serem aplicadas em um conjunto de elementos. Há três formas para a inserção de folhas de estilo em páginas **HTML**: é possível incorporar um estilo ao código, é possível conectar uma página a um estilo separado e, por fim, é possível definir um estilo para um comando.

Incorporando Folhas de Estilos

Para incorporar uma folha de estilo a uma página HTML, é necessário incluir a tag `<STYLE>`. `</STYLE>` dentro do cabeçalhos que informará ao navegador o seu início e final. Exemplo:

```
<HTML>
  <HEAD>
    ...
    <STYLE atributo1 atributo2 ...>
    ...
    comandos de folha de estilo ou linhas de chamadas de estilo.
    ...
  </STYLE>
  </HEAD>
  <BODY>
    ...
    conteúdo da página
    ...
  </BODY>
</HTML>
```

A sintaxe das folhas de estilo é bastante simples e pode ser escrita genericamente:

Para folhas de estilo incorporadas na página ou em um arquivo externo:

comando `{propriedade1: valor atribuído; propriedade2: valor atribuído...}`

Para folhas de estilo incorporadas diretamente ao comando:

```
<comando STYLE="propriedade1: valor atribuído; propriedade2: valor
atribuído...">
Assim, poderíamos atribuir a fonte Arial a todos os textos inseridos com
o comando <P> de dois modos:
p {font: Arial} ou
<P STYLE="font: Arial">
```

Estilos de Fontes e Textos



Font: define qual fonte será utilizada no comando. O nome das fontes deve ser escrito respeitando a grafia disponível nos aplicativos Windows.

```
{font: Garamond}
```

Font-family: define um conjunto de fontes a ser utilizado. Na ausência da primeira no computador do internauta, a segunda será usada. Cada nome de fonte deve ser separado por vírgulas.

```
{font-family: Arial, Courier, Garamond}
```

Font-size: define o tamanho da fonte a ser utilizado. Devem ser usados valores numéricos acompanhados das seguintes unidades: pt (pontos), in (polegadas), cm (centímetros) e px (pixels).

```
{font-size: 12pt}
```

Font-weight: define a espessura do traçado da fonte. As opções disponíveis são: extra-light, light, demi-light, medium, demi-bold, bold e extra-bold.

```
{font-weight: extra-bold}
```

Font-style: Define o estilo italic (itálico) ou normal.

```
{font-style: italic}
```

Text-decoration: define os estilos de underline (sublinhado), italic (itálico) e line-through (tachado)

```
{text-decoration: underline}
```

Text-align: Define o alinhamento do texto em: left (esquerdo), center (centralizado) ou right (direito).

```
{text-align: center}
```

Text-indent: Define a orientação do texto, ou seja, o recuo do texto com relação à margem esquerda. Deve ser utilizado um valor numérico com as mesmas unidades mostradas na propriedade font-size.

```
{text-indent: 6pt}
```

Line-height: Define o espaçamento entre linhas de um parágrafo. Deve ser usado um valor numérico com as mesmas unidades da propriedade anterior, mas podendo-se utilizar ainda a unidade % (valor percentual)

```
{line-height: 30pt}
```

Margin-left: Ajusta o tamanho da margem esquerda. Deve ser usado um valor numérico com as mesmas unidades da propriedade font-size.

```
{margin-left: 50px}
```

Margin-right: Ajusta o tamanho da margem direita. Deve ser usado um valor numérico com as mesmas unidades da propriedade font-size.

```
{margin-right: 50px}
```

Margin-top: Ajusta o tamanho da margem superior. Deve ser usado um valor numérico com as mesmas unidades da propriedade font-size.

```
{margin-top: 5pt}
```

Estilos de Fontes e Textos

Color: define a cor de um texto ou de um link.

```
{color: #FF0000}
```

define a cor do texto para vermelho.

```
A:link {color:#0000FF}
```

define a cor dos links ativos para azul.

```
A:active {color:#00FF00}
```

define a cor dos links acionados para verde.

```
A:visited {color:#FFFF00}
```

define a cor dos links ativos para amarelo.

Estilos de Imagens de Fundo

Background: define a cor de fundo de um texto ou define a imagem a ser utilizada como fundo da página



```
{background: #FF0000}
    define a cor do fundo do texto para vermelho.
{background: url(http://www.site.com.br/fundo.gif)}
    define a imagem para ser utilizada como fundo.
```

É possível ajustar, utilizando quatro parâmetros, o comportamento de uma imagem de fundo com relação à cobertura da página:

Repeat

Este comportamento é padrão e permite repetir uma imagem horizontal e verticalmente de modo que seja obtida uma cobertura total do fundo da página. Vale lembrar que ele só é possível no caso de imagens cujas dimensões sejam menores do que as da página Web onde forem inseridas.

```
BODY {BACKGROUND: URL(fundo.gif) repeat}
```

No-repeat

É um atributo de comportamento inverso do anterior: não permite que a figura seja repetida até cobrir todo o fundo da janela. Só é eficaz quando a figura tiver uma dimensão menor que a página onde for inserida.

```
BODY {BACKGROUND: URL(fundo.gif) no-repeat}
```

Scroll

É um atributo padrão de comportamento que faz com que a imagem seja tratada como se estivesse presa a página. Caso seja preciso rolar a página, a imagem de fundo será rolada também.

```
BODY {BACKGROUND: URL(fundo.gif) scroll}
```

Fixed

É um atributo de comportamento inverso do anterior. Ele impede que a imagem seja tratada como se estivesse presa a página. Caso seja preciso rolar a página, a imagem de fundo ficará parada.

```
BODY {BACKGROUND: URL(fundo.gif) fixed}
```

Conectando uma página a uma folha de estilo

A opção de conectar uma folha de estilo a uma página, necessitará de um arquivo tipo ASCII com extensão .CSS, contendo as instruções para cada estilo em seu interior. A instrução que conecta o arquivo de estilo à página (linha de chamada) também deve ser inserida entre as tags <HEAD>. </HEAD>.

```
<HTML>
  <HEAD>
    ...
    <TITLE> TÍTULO </TITLE>
    <STYLE atributo1 atributo2 ...>
    ...
    <LINK REL=STYLESHEET HREF="ESTILO.CSS" TYPE="TEXT/CSS"
    MEDIA="PRINT">
    ...
  </STYLE>
</HEAD>
<BODY>
  ...
  conteúdo da página
  ...
</BODY>
</HTML>
```

Nesse caso, foram utilizados duas propriedades:

Type: define o tipo de linguagem utilizado no estilo

Media: define o dispositivo em que o estilo será apresentado. As opções são:

- screen** (monitor) – o padrão;
- tty** – para monitores com capacidade limitada de resolução como terminais e teletipo;
- tv** – para monitores tipo TV de baixa resolução;
- projection** – para projetores tipo datashow;
- handheld** – para monitores de dispositivos portáteis (cristal líquido de baixa resolução);
- print** – para impressora ou monitores no modo de visualização de impressão;
- braille** – para dispositivos com saída em braille;
- aural** – para dispositivos de sintetização de voz;
- all** – para todos os dispositivos.

Definindo um estilo

Essa é a terceira opção de se colocar um estilo em uma página. Nesse caso, servirá para incorporar um estilo diretamente a um comando específico, em que, por exemplo, apenas um comando <p> deva ser diferente dos demais. Para isso, basta incorporar as instruções diretamente ao comando utilizado. O exemplo seguinte mostra um estilo sendo definido para um único parágrafo:

```
...
<p> parágrafo comum
...
<p STYLE="font-family: Arial, serif; font-weight: bold; text-decoration= underline">
Este texto será apresentado em Arial, negrito e sublinhado.
...
```

17.3. FORMULÁRIOS

Formulários são home pages que permitem o internauta preencher campos e submetê-los para alguém. Os formulários podem ser utilizados para praticamente qualquer entrada de dados em home pages. O objetivo é permitir que um visitante da home page envie informações para o responsável pela página.

Controles

Existem diversos tipos de controles diferentes para formulários. Cada um permite a entrada de um determinado tipo de dado. Qualquer campo possui o seguinte formato: **<input type=tipo name=nome >**. O tipo será discutido abaixo e o nome é utilizado para identificar os campos no formulário. Os tipos de controles disponíveis são:

type=text: controle de entrada de texto comum;

type=password: controle de entrada senhas (o texto não é mostrado para o usuário);

type=checkbox: botões do tipo liga/desliga;

 Gostei da sua homepage!

type=radio: botões seletores (iguais os seletores de radio FM/AM) que permitem que somente um seja acionado de cada vez;

 Sim Não



type=hidden: controle escondido (serve para passar algum dado que o usuário não deva ver para o destinatários do formulário);

type=submit: botão de envio do formulário (os botões são considerados campos também);

type=image: botão de envio do formulário (funciona exatamente igual ao anterior, mas apresenta uma imagem ao invés de texto);

type=reset: botão para apagar os dados do formulário para que o usuário reinicie a entrada

Para completar a funcionalidade selecionada por cada tipo mencionado acima, existem ainda os seguintes atributos permitidos para a tag **HTML** input que podem ser utilizados para otimizar um determinado campo de um formulário:

value=valor: inicializa o controle com um determinado valor;

disabled: quando presente determina que o controle não pode ser modificado pelo usuário, funcionando como um controle para exibição de dados somente;

checked: quando presente em um controle do tipo checkbox determina que o campo seja inicializado marcado;

size: tamanho: indica qual será o tamanho visível do campo na tela em caracteres;

maxlength: entrada máxima, indica qual é o número de caracteres permitidos para entrada no controle;

min: valor mínimo, indica qual será o menor valor numérico aceito no controle;

max: valor máximo, indica qual será o maior valor numérico aceito no controle;

Existe um controle de entrada adicional utilizado para a entrada de texto. É o controle TEXTAREA, cuja sintaxe é mostrada a seguir:

```
<textarea name="nome" rows=linhas cols=colunas>
texto inicial do textarea
</textarea>
```

Tags para Formulários

Formulários são agregados de campos usados para criar uma página de entrada de dados. Existe um comando **HTML** para definir formulários. É o comando form cuja sintaxe é:

```
<form method=método action=ação>
```

O método do form é usado para especificar como o formulário será enviado para o servidor. O action é a URL (endereço WWW) do CGI (common gateway interface) que fará o tratamento dos dados do formulário. O CGI é o programa que reconhece os dados enviados pelo usuário e o envio do formulário para o destinatário.



18. HOSPEDAGEM



18.1. Como hospedar

Para se hospedar um site com o intuito de que o mesmo seja acessível pela Internet é necessário que as páginas criadas sejam colocadas em um computador que possua um programa que efetuará o gerenciamento WEB, normalmente um computador servidor. Dentre os gerenciadores mais conhecidos citamos os seguintes: **IIS** - Internet Information Server (para o ambiente Windows NT e 2000 Server), **Apache** (para ambientes UNIX, LINUX etc) e o **WPS** - Windows Personal Server (para o ambiente Windows 98).

Outro serviço normalmente gerenciado por estes programas é o FTP (File Transfer Protocol) ou protocolo de transferência de arquivos. Trata-se de um protocolo (conjunto de regras) para se efetuar a troca de arquivos entre computadores por meio da Internet. É a maneira mais fácil para se efetuar UPLOADS (envio de arquivos) e DOWNLOADS (recebimento de arquivos). Quando fazemos o download de um arquivo da WEB normalmente estamos acessando a área FTP de um site, porém, nesse caso, via de regra, nosso acesso é anônimo, ou seja, na maioria das vezes não nos pedem identificação para efetuar o download. O acesso para manutenção de uma página pelo contrário, normalmente, é controlado, precisamos de um Login (nome de usuário) e de um Password (senha) para conseguir a conexão. Outra prática recomendada é a utilização de um programa cliente FTP que irá facilitar a comunicação entre nossa máquina e o Servidor que hospeda nossa página WEB.

18.2. Utilização de FTP Cliente

Como vimos, a utilização de um programa FTP cliente poderá facilitar muito a manutenção de nossas páginas WEB. Através destes programas podemos nos conectar, enviar, remover, atualizar e baixar arquivos a partir de nosso hospedeiro com bastante facilidade, quase como se estivéssemos manipulando arquivos e diretórios em nossa própria máquina.

Um dos mais populares programas de FTP Cliente é o WS-FTP. Procuraremos passar alguns detalhes sobre a funcionalidade deste programa e como ele poderá nos auxiliar na manutenção de nossas páginas WEB utilizando as facilidades do protocolo FTP.

Embora em inglês a utilização do programa é bastante simples. O programa fará a conexão com o servidor FTP dando acesso às áreas permitidas e pré-configuradas por nosso hospedeiro.

No caso do WS-FTP (e da maioria dos outros programas do tipo, também) as informações prévias que necessitaremos são:

Host Name: Número IP ou endereço completo do servidor para acesso internet.

User ID: Login, nome do usuário que foi cadastrado.

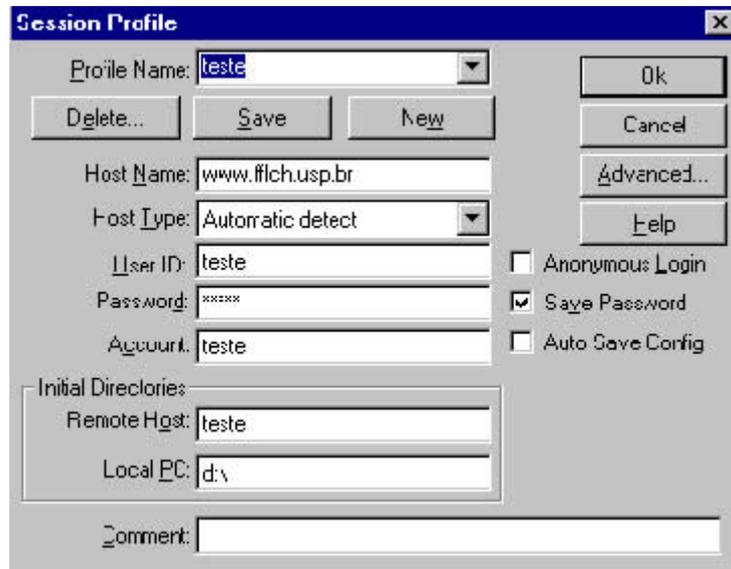
Password: Senha de acesso com que aquele usuário foi cadastrado.

Account: Conta, normalmente será o mesmo nome do usuário ou User ID.

Remote Host: O nosso diretório ou área de trabalho criado no servidor.

Algumas destas informações poderão ser definidas pelo usuário, mas serão implementadas pelo administrador do servidor, desta forma convém confirmar com ele como foram efetivamente cadastradas.

De posse destas informações podemos facilmente configurar a tela de conexão do WS_FTP para nos conectarmos ao servidor:



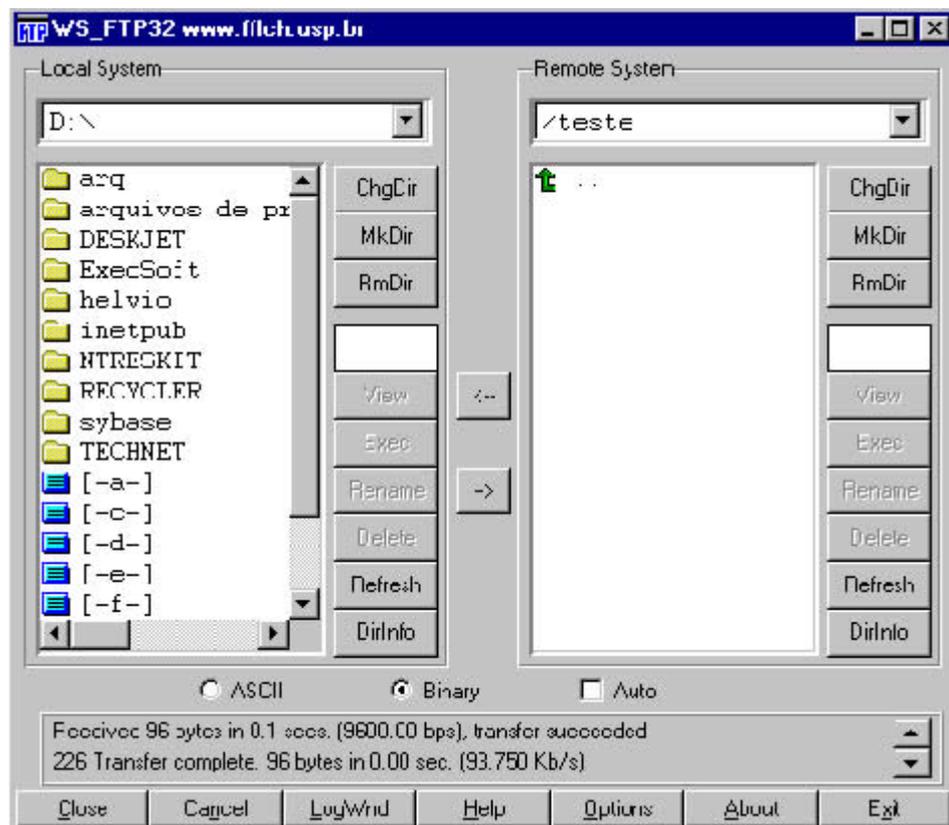
Como você pode ver na figura acima, existem outros campos a serem preenchidos, porém estes não são essenciais podendo até, em alguns casos, permanecerem em branco:

Profile Name: É um nome aleatório que podemos dar à nossa conexão. Por meio deste nome e tendo tomado o cuidado de marcar a opção Auto Save Config, podemos facilmente nos reconectar bastando especificar o nome da conexão. O WS_FTP permite que tenhamos diversas conexões com diversos servidores ou com diversas contas, também pode ser usado por diversos usuários cada um com suas características de usuários e senhas diferentes.

Host Type: A opção default (que já vem escolhida) é Automatic detect, convém deixar esta opção ativa, pois ela atenderá a mais de 90% dos casos. Caso queira e saiba, poderá escolher entre os diversos tipos de Servidores disponíveis.

Local PC: Como veremos a seguir o WS-FTP abrirá uma tela dividida em esquerda e direita, sendo que a parte esquerda representará nossa máquina (cliente) e o lado direito a máquina remota (servidor). Com a opção Remote Host, como já foi visto, definimos nosso diretório no servidor e com a opção Local PC definimos a área em nossa máquina onde colocamos os dados referentes a nossa página. A mudança de diretórios em nossa máquina normalmente é livre, podemos ir para qualquer drive ou diretório, já na máquina servidora, normalmente, haverá limitações. O que escolhemos aqui é só a área ou diretório inicial, tanto no cliente como no servidor.

Após a conexão com o servidor a tela do WS_FTP que surgirá será semelhante à apresentada a seguir:



Como já comentamos, a tela é essencialmente dividida em duas partes, a da esquerda refere-se à nossa máquina cliente (Local System) e a da direita ao servidor (Remote System).

Para efetuarmos a troca de arquivos basta que os selecionemos e, em seguida cliquemos nas setas existentes entre as duas divisões: , ou seja estaremos selecionando algo do lado direito

da tela (servidor) e a enviaremos para o nosso computador cliente, e , selecionaremos algo em nossa máquina (cliente) e a enviaremos para o servidor. A parte superior indica o local onde estamos atuando naquele momento em cada um dos equipamentos, portanto será para onde as transferências serão completadas.

Vamos esclarecer os outros botões existentes em cada uma das duas telas, ele nos ajudarão a organizar as tarefas.

ChgDir: Mudar de diretório;

MkDir: Criar diretório;

Rmdir: Apagar diretório;

View: Ver o que esta selecionado;

Exec.: Executar o que esta selecionado;

Rename: Renomear;

Delete: Apagar;

Refresh: Atualizar;

DirInfo: Mostra conteúdo do diretório corrente. Aquele que está marcado na parte de cima.

GLOSSÁRIO



A

<A HREF> - usada para criar os *links* entre páginas e servidores da Internet. Ele também tem a função de definir quais áreas do texto ou quais imagens serão usadas para ligar as diversas páginas. Exemplo:

```
<A HREF=http://www.fflch.usp.br>Site da FFLCH</A>
<A HREF=http://www.usp.br target=blank>Site da USP</A>
<A HREF=mailto:fflchsti@usp.br>fflchsti@usp.br</A>
```

<A NAME> - define uma área dentro de uma página **HTML** e atribui um nome a essa área. Dessa forma, a região criada com a tag **<A NAME>** pode ser chamada diretamente, usando a tag **<A HREF>**. Exemplos:

```
<A NAME="Area51">Essa área do arquivo foi marcada como Areal</A><P>
<A HREF="www.tcinet.com.br/arquivo.html#Areal">Clique para ir à
Area51</A><P>
```

<ABBR> - serve para indicar os significados de abreviações, como fax, Corp., Av. etc. **<ABBR>** não modifica a exibição das páginas **HTML**, mas fornece dados para mecanismos que interpretam dados em uma página, como sistemas de busca, corretores ortográficos, sintetizadores de voz e sistemas de tradução. Exemplo:

```
<ABBR TITLE="Avenida">Av.</ABBR>
```

<AREA> - usada para determinar quais áreas de uma imagem serão "clicáveis" e quais os *links* relacionados com cada uma dessas imagens. Pode-se, por exemplo, dividir uma imagem em quatro partes, cada uma com *links* para páginas diferentes dentro de um *site*. **<AREA>** deve ser usado com a tag **<MAP>** para criar imagens mapeadas. Exemplo:

```
<MAP NAME="mapal">
<AREA SHAPE="rect" COORDS="0,0,200,100"
  HREF="http://www.tcinet.com.br/exemplo1.html">
<AREA SHAPE="circle" COORDS="100,85,10"
  HREF="http://www.tcinet.com.br/exemplo2.html">
<AREA SHAPE="poly" COORDS="10,80,15,80,0,110" HREF="http://
www.tcinet.com.br/exemplo3.html">
</MAP>
```

B

**** - Uma das tags mais simples de **HTML**. É usada para aplicar negrito em textos. Exemplo:

```
<B>Texto</B>
```

<BASE> - Em **HTML**, os *links* para outros documentos e imagens que podem ser exibidos em uma página são especificados por meio de URLs. As URLs são os endereços na Internet. Por exemplo, uma foto pode estar no endereço <http://www.fotos.com/foto.jpg>. A tag **<BASE>** é usada para facilitar o trabalho dos programadores e estabelece um endereço comum, onde os objetos são encontrados. Exemplo:

```
<BASE HREF="http://www.teinet.com.br">
```

Com esse endereço especificado, elimina-se a necessidade de digitar o endereço `http://www.teinet.com.br` toda vez que for preciso chamar uma página ou imagem do site. Um *link* para uma página de notícias, por exemplo, ficaria dessa forma:

```
<A HREF="noticias.html">
```

Ou então para uma imagem:

```
<IMG SRC="../imagens/imagem.gif">
```

A tag `<BASE>` deve ser usado sempre entre os elementos `<HEAD>` e `</HEAD>` e é obrigatório o uso do atributo `HREF` para especificar um endereço.

<BASEFONT> - usada para modificar os atributos das fontes exibidas em uma página **HTML**. Com ele, é possível determinar o tamanho, cor e o tipo da fonte. Funciona exatamente como o elemento ``. Exemplo:

```
<BASEFONT SIZE="3" FACE="Arial" COLOR="#000000">Texto escrito com fonte arial, tamanho 3 e cor preta</BASEFONT>
```

<BIG> - serve para destacar uma palavra, frase ou bloco de texto em uma página **HTML**, aumentando o tamanho da fonte especificada. Em alguns casos, esta tag pode substituir a tag `` para fazer pequenas mudanças nas fontes usadas em um documento. Exemplo:

```
<BIG>Aumente o tamanho deste texto</BIG>
```

<BLINK> - Faz com que o texto pisque na tela. Suportado apenas pelo Netscape. Exemplo:

```
<blink>Texto que pisca</blink>
```

<BODY> - a presença da tag `<BODY>` é obrigatória em praticamente todas as páginas **HTML**. Junto com este elemento, são colocadas todas as especificações de um documento, desde o tamanho e tipo da fonte até a cor da página e até mesmo uma imagem para ser usada como fundo de tela. Exemplo:

```
<HTML>
<HEAD>
<TITLE>Exemplo 1</TITLE>
</HEAD>
<BODY BGCOLOR="ffffff">
Esta página tem fundo branco
</BODY>
</HTML>
```

**
** - indica uma quebra de linha em um determinado local de uma página **HTML**. Atributos usados com este elemento: `ID`, `CLASS`, `TITLE`, `STYLE` e `CLEAR` Exemplo:

```
Esta é a primeira linha.<BR>
Esta é a segunda linha.
```

<BUTTON> - usada para criar botões em documentos **HTML**. Ao contrário da tag `<INPUT>`, que pode ser usada para a mesma função, esta tag permite que se criem botões com imagens e até

mesmo com formatação de texto em **HTML**. O tamanho dos botões é ajustado automaticamente para acomodar as imagens e textos acrescentados. Exemplo:

```
<BUTTON NARRE="submit" VALUE="Submit" TYPE="submit" WIDTH="64"
HEIGHT="25">
<IMG SRC="http://www.tclnet.com.br/imagens/hit.gif" ALT="Clique aqui"
WIDTH="65" HEIGHT="25">
</BUTTON>
```

C

<CAPTION> - usada sempre depois da tag **<TABLE>** e serve para mostrar o nome ou título de uma tabela. Atributos usados com este elemento: **ALIGN**, **VALIGN**, **ID**, **CLASS**, **LANG**, **DIR**, **TITLE**, **STYLE**, **ONCLICK**, **ONDBLCLICK**, **ONMOUSEDOWN**, **ONMOUSEUP**, **ONMOUSEOVER**, **ONMOUSEMOVE**, **ONMOUSEOUT**, **ONKEYPRESS**, **ONKEYDOWN**, **ONKEYUP** Exemplo:

```
<HTML>
<HEAD>
<TITLE>Exemplo do elemento <CAPTION></TITLE>
</HEAD>
<BODY>
<TABLE BORDER="1" ALIGN="left" BORDERCOLOR="#0000FF" WIDTH="100%">
<CAPTION ALIGN="bottom">Exemplo do uso do CAPTION</CAPTION>
<TR><TH>Texto A</TH> <TH>Texto B</TH></TR>
<TR> <TD>Texto C</TD> <TD>Texto D</TD> </TR>
<TR> <TD>Texto E</TD> <TD>Texto F</TD> </TR>
</TABLE>
</BODY>
</HTML>
```

<CENTER> - usada para alinhar um texto, tabelas e imagens, no centro da página. Este tag pode ser substituída pela tag **<DIV>**. Exemplo:

```
<CENTER>Esta frase fica no centro da tela</CENTER>
```

<CODE> - usada para formatar código de programação em páginas **HTML**. Os textos colocados entre a tag **<CODE>** são apresentados em uma fonte fixa, que na maioria dos casos é a fonte Courier New. Exemplo:

```
<CODE>Texto</CODE>
```

<COL> - permite que sejam criadas colunas em uma tabela e determina as propriedades para cada uma delas isoladamente. Deve ser usada juntamente com a tag **<COLGROUP>**, que é usada para criar e definir as propriedades gerais de um grupo de colunas. Exemplo:

```
<TABLE>
<COLGROUP>
<COL>
<COL SPAN="2">
</COLGROUP>
<TR>
<TD>...</TD>
<TD>...</TD>
<TD>...</TD>
</TR>
</TABLE>
```

<COLGROUP> - Este elemento permite que seja determinado um grupo de colunas dentro de uma tabela, bem como as propriedades destas colunas. No exemplo, o **<COLGROUP>** foi usado

para criar uma tabela com uma linha e cinco colunas com 50 pixels de largura cada. A tag <COL> é usada para determinar as propriedades das colunas separadamente. Exemplo:

```
<TABLE>
<COLGROUP span="5" width="0"></COLGROUP>
<TR>
<TD>...</TD>
<TD>...</TD>
<TD>...</TD>
<TD>...</TD>
<TD>...</TD>
</TR>
</TABLE>
```

D

<DD> - mostra a definição de um termo em uma lista de definições de termos. É usada junto com as tags <DL> (Definition List) e <DT> (Definition Term). Atributos usados com este elemento: ID, CLASS, LANG, DIR, TITLE, STYLE, ONCLICK, ONDBLCLICK, ONMOUSEDOWN, ONMOUSEUP, ONMOUSEOVER, ONMOUSEMOVE, ONMOUSEOUT, ONKEYPRESS, ONKEYDOWN, ONKEYUP. Exemplo:

```
<DL>
<DT>Termo 1</DT>
<DD>Definição do Termo 1 </DD>
<DT>Termo 2</DT>
<DD>Definição do Termo 2</DD>
</DL>
```

 - usado junto com o elemento <INS> para indicar textos ou palavras que foram apagados da versão original e inseridos em uma versão editada. Um traço é colocado sobre as palavras ou frases marcadas com este elemento. Exemplo:

Este texto foi apagado, mas esta <INS>frase foi inserida</INS>

<DIR> - desenvolvida para criar listas desordenadas. Entretanto, foi substituída pela tag . Exemplo:

```
<DIR>
<LI>Item 1
<LI>Item 2
<LI>Item 3
</DIR>
```

F

 - usada para modificar os atributos das fontes exibidas em uma página. Com ela, é possível determinar o tamanho, cor e o tipo da fonte. Funciona exatamente como o elemento <BASEFONT>. Exemplo:

```
<FONT SIZE="4" FACE="impact" COLOR="#ffffff">Texto escrito com fonte Impact, tamanho 4 e cor branca</FONT>
```

<FORM> - usada para delimitar a quantidade de campos de entrada de um formulário e também

para definir como esses dados serão processados. A tag geralmente é usada com a tag `<INPUT>`. Exemplo:

```
<FORM METHOD="post" ACTION="http://www.exemplo.com.br/cgi-bin/query">
<INPUT TYPE="text" NAME="exemplo">
</FORM>
```

<FRAME> - usada, basicamente, para especificar um endereço ou documento de onde as informações dos quadros serão retiradas. Exemplo:

```
<FRAMESET COLS="30%,70%">
<FRAMESET ROWS="40%,60%">
<FRAME SRC="arquivo frame1.html">
<FRAME SRC="arquivo frame2.html">
</FRAMESET>
<FRAME SRC="arquivo frame3.html">
</FRAMESET>
```

<FRAMESET> - usada para determinar as características de cada um dos frames de um documento, como a largura das colunas. Exemplo:

```
<FRAMESET COLS="200, 400">
<FRAME SRC="arquivo_frame1.html">
</FRAMESET>
```

H

<Hn> - usada para descrever títulos de uma sentença. Entre os seis níveis, o primeiro é o mais importante e o último, o menos importante. Exemplo:

```
<H1>Este título é mais importante ... </H1>
<H2> ...do que este outro título...</H2>
<H3> ...e assim por diante...<H3>
```

<HEAD> - traz informações sobre um documento **HTML**, como o título, palavras-chave que podem ser úteis para mecanismos de busca e outros dados que não são considerados como parte do conteúdo de um documento. Exemplo:

```
<HEAD>
<TITLE>Novo Documento</TITLE>
<META HTTP-EQUIV="Content-Language" CONTENT="pt">
</HEAD>
```

<HR> - cria uma linha horizontal entre duas seções de um documento **HTML**. A linha costuma ser acinzentada e ocupa toda a largura da tela. Essas características, entretanto, podem ser alteradas. Exemplo:

```
<HR WIDTH=50% SIZE=10 COLOR=BLUE>
```

<HTML> - indica que tudo o que está na página deve ser interpretado como elementos para marcação de texto no padrão **HTML**. Exemplo:

```
<HTML>
<HEAD>
<TITLE>Uma página HTML</TITLE>
```

```

</HEAD>
<BODY>
Texto do documento ....
</BODY>
</HTML>

```

I

<I> - usada para formatar textos em itálico. Exemplo:

```
<I>Texto</I>
```

<IFRAME> - permite a inclusão de um frame em uma página HTML da mesma forma que uma imagem ou outro objeto é incluído. Na prática, é como se você estivesse incluindo uma página HTML dentro de outra página. No entanto, só é suportado pelo Internet Explorer (versão 3.0 ou superior) e pelo Netscape 6 ou superior. Exemplo:

```

<IFRAME SRC="http://www.tcinet.com.br/noticias.html" width="300"
height="200" scrolling="auto" frameborder="0"></IFRAME>
<P>
HTML<BR>
<IFRAME SRC="http://www.tcinet.com.br" width="200" height="150"
bcrolling="auto" frameborder="0"></IFRAME>

```

**** - serve para inserir uma imagem em uma página. A propriedade SRC deve sempre ser usada para identificar a origem da imagem utilizada. Exemplo:

```
<IMG SRC="http://www.tcinet.com.br/imagens/estrela.gif">
```

<INPUT> - cria campos de formulários para a entrada de dados. É possível criar caixas de entrada para textos, botões para enviar os dados etc. Exemplo:

```

<DIV ALIGN="JUSTIFY"><FONT FACE="ARIAL" SIZE="3" COLOR="#000000">
Nome: <INPUT TYPE="text" NAME="nome" SIZE="100"><BR>
Sobrenome: <INPUT TYPE="text" NAME="nome" SIZE="100"><BR>
E-mail: <INPUT TYPE="text" NAME="nome" SIZE="100"><BR>
<INPUT TYPE="radio" NAME="sexo" VALUE="Homem">Homem<BR>
<INPUT TYPE="radio" NAME="sexo" VALUE="Mulher">Mulher<BR>
</DIV>

```

K

<KBD> - usada para indicar textos que foram digitados pelo usuário. Os textos colocados entre a tag <KBD> são apresentados em uma fonte fixa, que na maioria dos casos é Courier New. Exemplo:

```
<KBD>Texto em fonte fixa</KBD>
```

L

<LABEL> - serve para associar dois objetos diferentes em um documento. Não modifica a exibição de uma página, mas é um recurso importante para a manipulação de dados em formulários. Exemplo:

```

<FORM ACTION= METHOD="post">
<TABLE>
<TR>
<TD><LABEL for="pnome">Nome</LABEL>
<TD><INPUT type="text" narre="pnmeironome" id="pnome">
<TR>
<TD><LABEL for="snome">Last Narre</LABEL>
<TD><INPUT type="text" narre="sobrenome" id="sname">
</TABLE>
</FORM>

```

<LEGEND> - define um nome para identificar o conteúdo da tag **<FIELDSET>**. Exemplo:

```

<FIELDSET>
<LEGEND>Dados pessoais</LEGEND>
Nome: <INPUT NARRE="primeiro nome" TYPE="text" TABINDEX="1"><BR>
Sobrenome: <INPUT NARRE="sobrenome" TYPE="text" TABINDEX="2"><BR>
Endereço: <INPUT NARRE="endereco" TYPE="text" TABINDEX="3">
</FIELDSET>

```

**** - usado para criar itens de uma lista, que pode ser desordenada ou ordenada. Exemplo:

```

<UL>
<LI>Um item
<LI>Outro item
</UL>

```

M

<MAP> - usada para criar um mapa, que posteriormente será aplicado a uma imagem. É usada para definir um nome para o mapa e também quais as partes de uma imagem que serão clicáveis e quais os endereços relacionados com essas partes. Exemplo:

```

<MAP NAME="mapal">
<P>Navegando pelo site:
<A HREF="intro.html" SHAPE="rect" COORDS="0,0,118,28">Introdução</A>
<A HREF="news.html" SHAPE="rect" COORDS="118,0,184,28">Novidades</A>
<A HREF="search.html" SHAPE="circle" COORDS="184,200,60">Pesquisa</A>
<A HREF="links.html" SHAPE="poly"
COORDS="276,0,276,28,100,200,50,50,276,0">Links</A>
</MAP>

```

<MARQUEE> - movimenta um texto de um lado para outro da tela. Interpretado somente pelo Internet Explorer. Exemplo:

```

<MARQUEE>Texto</MARQUEE>

```

<META> - usada para indicar diversas propriedades de um documento, como, por exemplo, o autor da página, a linguagem utilizada, o assunto do documento etc. Deve ser colocada entre a tag **<HEAD>** e não produz nenhuma informação visível em uma página **HTML**. Em compensação, ela contém informações úteis para mecanismos de busca e outros sistemas de tratamento de dados. Exemplo:

```

<META NAME="Author" LANG="pt" CONTENT="Jose da Silva">
<META HTTP-EQUIV="Content-Type" CONTENT="text/html";
CHARSET=windows-1252">

```

N

<NOFRAMES> - especifica um conteúdo que deve ser exibido apenas por navegadores mais antigos, que não conseguem exibir frames. Exemplo:

```
<FRAMESET cols="30%, 70%">
<FRAME SRC="principal.html">
<FRAME SRC="secundario.html">
<NOFRAMES>
<P> O seu navegador não suporta frames. Por favor, copie o Netscape
Navigator 3 ou o Internet Explorer 3<P>
</NOFRAMES>
</FRAMESET>
```

<NOSCRIPT> - quando os comandos de um script não podem ser interpretados por um navegador, os autores podem usar a tag **<NOSCRIPT>** para exibir um conteúdo alternativo. É sempre usada com a tag **<SCRIPT>**. Exemplo:

```
<SCRIPT TYPE="text/tcl">
...alguns comandos do script...
</SCRIPT>
<NOSCRIPT>
<P>
<A HREF="http://www.exemplo.com.br/noscript.html">Clique aqui para ver
o conteúdo desta página</A>
</NOSCRIPT>
```



<OBJECT> - a maioria dos navegadores tem mecanismos próprios para exibir imagens, fontes diferentes, tabelas etc. Quando um browser não tem um mecanismo para exibir um determinado objeto (um arquivo de música ou um documento do Word, por exemplo) ele pode usar aplicativos externos para mostrar esses objetos. A tag **<OBJECT>** define exatamente qual o arquivo que o browser terá que interpretar e qual o aplicativo externo que será usado para executar a tarefa. É possível usar aplicações em Java e diversos plug-ins, como o Real Player, Flash, Adobe Acrobat. Exemplo:

```
<OBJECT CLASSID="http://www.exemplo.com.br/filmes/filme.mpeg"
TYPE="application/mpeg" STANDBY="Carregando vídeo">
</OBJECT>
```

**** - cria listas ordenadas em um documento. Os itens em uma lista ordenada são numerados. Exemplo:

```
<OL>
<LI>Primeiro item da lista
<LI>Segundo item da lista
<LI>Terceiro item da lista
</OL>
```

<OPTION> - define quais as opções que serão listadas em um menu criado com a tag **<SELECT>**. Exemplo:

```
<FORM>
<SELECT NAME="menu1" MULTIPLE>
<OPTION VALUE="Escolha1">Item 1
<OPTION VALUE="Escolha2">Item 2
<OPTION VALUE="Escolha3">Item 3
</SELECT>
</FORM>
```

P

<P> - serve para criar parágrafos em documentos. Depois de posicionado, a tag **<P>** cria duas quebras de linha. Exemplo:

```
<P>Paragrafo</P>
```

<PRE> - indica que o texto seguinte é pré-formatado. Dessa forma, os espaços, tabulações entre outros caracteres que normalmente são ignorados pelo **HTML**, são exibidos corretamente. Esta tag exibe o conteúdo com uma fonte fixa. Exemplo:

```
<PRE>Texto com formatação original.</PRE>
```

S

<SELECT> - usada para criar menus. É usada sempre com a tag **<OPTION>**, que define quais opções de escolha o menu oferece. Exemplo:

```
<FORM>
<SELECT NAME="menu" MULTIPLE>
<OPTION VALUE="Escolha1">Item 1
<OPTION VALUE="Escolha2">Item 2
<OPTION VALUE="Escolha3">Item 3
</SELECT>
</FORM>
```

<SMALL> - exibe o texto num tamanho de fonte menor do que o padrão. Exemplo:

```
<SMALL>Texto em fonte menor do que a normal</SMALL>
```

**** - usada para formatar blocos com textos, tabelas e imagens em um documento e funciona de maneira semelhante a tag **<DIV>**. A diferença é que esta força uma quebra de linha. Com o uso de Style Sheets, pode ser usada para definir estruturas de bancos de dados em páginas **HTML**. No primeiro exemplo, ela define apenas o alinhamento do texto. Já no segundo caso, a tag é usada para criar a estrutura de um banco de dados de clientes. Exemplo:

```
<SPAN ALIGN="CENTER">Este texto aparecerá no centro da tela</SPAN>

<SPAN ID="client-silva" class="client">
<P>
<TABLE CLASS="client-data">
<TR><TH>Nome:<TD>Mario</TR>
<TR><TH>Sobrenome:<TD>Silva</TR>
<TR><TH> e-mail:<TD>mariosilva@exemplo.com.br</TR>
</TABLE>
</SPAN>
```

<STYLE> - usada para determinar diversas características de Style Sheets dentro de um documento. Deve ser usada dentro do cabeçalho de uma página e permite modificar as propriedades de praticamente todas as tags de **HTML**. As tags **<LINK>** e **<META>** oferecem outros métodos para incluir Style Sheets em documentos. Exemplo:

```
<HEAD>
```



```
<STYLE type="text/css">
H1.meuexemplo {border-width: 1; border: solid; color: blue; text-align:
center}
</STYLE>
</HEAD>
<BODY>
<H1 class="meuexemplo">Esta frase vai aparecer com uma borda e será
azul</H1>
<H1> Mas essa continuará normal </H1>
</BODY>
```

<SUB> - diminui o tamanho do texto e o coloca um pouco abaixo do alinhamento normal. Útil para notações científicas. Exemplo:

```
3 <SUB>12</SUB>
```



<SUP> - Semelhante à tag <SUB>. A diferença é que esta tag alinha o texto um pouco acima do nível normal. Útil para notações matemáticas envolvendo potências ou abreviaturas. Exemplo:

```
2<SUP>o</SUP> grau; 3<SUP>10</SUP>
```

T

<TABLE> - cria tabelas. Contém todas as definições de fontes, linhas, células, cabeçalho e rodapé de uma tabela. Exemplo:



```
<TABLE BORDER="1" BORDERCOLOR="#0000FF" CELLPADDING="10" WIDTH="50%">
<CAPTION>Exemplo de tabela</CAPTION>
<TR><TH>Texto 1</TH><TH> Texto 2</TH><TH> Texto 3</TH>
</TR><TR><TD>Continuação do texto 1 </TD>
<TD> Continuação do texto 2</TD>
<TD> Continuação do texto 3</TD>
</TR> <TR><TD> Continuação do texto 1 </TD>
<TD> Continuação do texto 2</TD>
<TD> Continuação do texto 3</TD>
</TR>
<TFOOT>
<TR><TH CO LSPAN="4">Este é apenas um pequeno exemplo</TH></TR> </
TFOOT>
</TABLE>
```

<TBODY> - usada para especificar o conteúdo da tabela. Dentro estão as tags que criam as linhas e colunas da tabela. O uso desta tag, no entanto, não é necessária. Exemplo:

```
<TABLE>
<TBODY>
<TR><TD>Este é um exemplo</TD></TR>
<TR><TD>Este é outro exemplo<TD></TR>
</TBODY>
</TABLE>
```



<TD> - todo o conteúdo que é exibido em uma tabela deve ser inserido dentro de uma célula criada com a tag <TD> ou <TH>. Exemplo:

```
<TABLE>
<TR><TD>Primeira célula da tabela</TD><TD>Segunda célula da
tabela<TD><TD>Terceira célula da tabela</TD></TR>
<TR><TD COLSPAN=3> </TD>
</TABLE>
```





<TEXTAREA> - usada para criar caixas de entrada de texto em páginas HTML. Exemplo:

```
<FORM ACTION="http://www.tcinet.com.br/cgi-bin/pegar-texto"
method="post">
<P>
<TEXTAREA NARRE="texto" ROCAS="25" COLS="60">
</TEXTAREA>
<INPUT TYPE="submit" VALUE="Enviar"><INPUT TYPE="reset" VALUE="Limpar">
</P>
</FORM>
```



<TFOOT> - usada para exibir conteúdo no rodapé de uma tabela. Deve haver, pelo menos, uma linha de dados, criada com a tag <TD>. Exemplo:

```
<TABLE BORDER="1" CELLPADDTNG="10" CELLSPACING="0" WIDTH="50%">
<TR> <TD>Texto A</TD>
<TD>Texto B</TD>
<TD>Texto C</TD>
</TR>
<TFOOT>
<TR><TH COLSPAN=3>Fim da tabela</TH></TR>
</TFOOT>
</TABLE>
```



<TITLE> - usada para identificar o título de um documento. Esta tag deve sempre ser colocada no cabeçalho de uma página. Exemplo:

```
<HEAD>
<TITLE>Exemplos de HTML</TITLE>
</HEAD>
```



<TR> - define as linhas de uma tabela. Toda linha deve ter pelo menos uma tag <TD>, para exibir o conteúdo da tabela. Exemplo:

```
<TABLE BORDER="1" BORDERCOLOR="#000000">
<TR ID="primeiralinha"><TD>Esta é a primeira linha de uma tabela</
TD>
<TR ID="segundalinha"><TD>Esta é a segunda linha de uma tabela</TD>
</TABLE>
```



<TT> - usada para exibir textos com fontes fixas, como Courier New. Exemplo:

```
<TT>Um texto com fonte fixa</TT>
```

U

<U> - uma das tags mais simples de HTML. Usada para sublinhar textos. Exemplo:

```
<U>Texto</U>
```

**** - usada para criar listas sem ordem nenhuma. Exemplo:

```
<UL>
<LI>Um item qualquer
<LI>Outro item qualquer
</UL>
```



Universidade de São Paulo
Reitor: Prof. Dr. Adolpho José Melfi
Vice-Reitor: Prof. Dr. Hélio Nogueira da Cruz



Faculdade de Filosofia, Letras e Ciências Humanas
Diretor: Prof. Dr. Sedi Hirano
Vice-Diretor: Profa. Dra. Eni de Mesquita Samara



Seção Técnica de Informática
Chefe: Augusto César Freire Santiago
Chefe-substituta: Néli Maximino

Título: HTML

Elaboração: Seção Técnica de Informática

Organização: Seção Técnica de Informática

Revisão: Augusto César Freire Santiago

Diagramação: Patrícia Carla dos Santos e Marcelo Marcos Benedito

Bibliografia: Cintra, Glauber - *Construção de páginas na WWW: HTML e CGI - versão 1.3* - 1996.
ICMC / USP - *Tutorial HTML* - [<http://www.icmsc.sc.usp.br/manuals/HTML/>] - 1999.
INET - *Tutorial HTML* - [<http://www.inet.com.br/suporte/home03.htm>] - 1999.

Versão: 3.0
Última Atualização: 05/05/03