

Apostila de HTML

**Elaborada pelo
Professor Sérgio Furgeri**

**FIEC – Fundação Indaiatubana de
Educação e Cultura**

O QUE É HTML?

Abreviação de HiperText Markup Language é a linguagem de programação que usamos para criar uma página Web, que por sua vez será composta de textos e comandos especiais que chamaremos de Tags. Não se assuste com o nome, pois se trata de uma linguagem muito simples, e tem como finalidade básica formatar o texto e imagens exibidas e criar ligações entre páginas Web, criando assim documentos com o conceito de Hipertexto.

Como em qualquer outra linguagem, o programador deve escrever o código - fonte seguindo as regras de sintaxe da linguagem. Este código – fonte é posteriormente interpretado pelo browser, que se encarregará de executar os comandos ou tags do código para formatar e acessar recursos da Web.

Em março de 1989, Tim Berners-Lee do European Laboratory for Particle Physics (mais conhecido como CERN), propôs um novo conjunto de protocolos para um sistema de distribuição de informações da Internet a ser usado entre os diversos grupos de pesquisa em física de altas energias. Os protocolos da WWW foram logo adotados por outras organizações e foi formado um consórcio de organizações, chamado de W3 Consortium, que uniu seus recursos de modo a prosseguir com o desenvolvimento de padrões WWW.

Esse consórcio é liderado pelo MIT (Massachusetts Institute of Technology), pelo CERN e pelo INRA (French National Institute for Research in Computer Science and Control). Ele propõe recursos mais novos e sofisticados de HTML, avalia sugestões e implementações alternativas e publica novos "níveis" de versões do padrão HTML.

Existem vários programas para editoração HTML, que tornaria nosso trabalho muito mais fácil e nos acomodaria como usuários – finais que não entendem nada do que estão fazendo. Por isso, sofreremos um pouco e usaremos o incrível, sensacional, fantástico e estupendo Bloco de Notas (NotePad) do Windows.

A HyperText Markup Language (HTML) é a linguagem da World Wide Web. Além do próprio software servidor Web, a HTML é o coração da Web. Todos os documentos, hiperlinks clicáveis, imagens gráficas, documentos de multimídia, formulários e todas as outras coisas da Web que você (e seu chefe) já viram estão baseadas em HTML.

Alguns chamam a HTML de “linguagem de programação”, mas isso não deverá assustá-lo. A HTML é uma simples linguagem de marcação fácil de aprender. Até os novatos na área de computação podem assimilar os conhecimentos básicos rapidamente. Se você já aprendeu a usar o seu processador de texto, então aprendeu algo tão complexo quanto a HTML.

Embora seja verdade que a HTML e a própria World Wide Web apareceram primeiro nos sistemas de computadores UNIX, você pode aprender e usar a HTML em qualquer sistema de computador, incluindo o seu PC, usando ferramentas com as quais já está familiarizado.

A HTML é completamente independente do sistema. Não importa se você está trabalhando em um PC Windows, um Macintosh, uma estação de trabalho UNIX de alto desempenho ou um cluster VAX/VMS multiusuário: a HTML é a mesma em qualquer lugar.

VANTAGENS DA HTML

Aqui está uma pequena lista das principais características da HTML:

- Formatação de documento usando vários estilos de letras, incluindo uma variedade de títulos.
- A característica de incluir hiperlinks que apontam para outros documentos Web ou serviços em sistemas de computadores por toda a Internet.
- Uma grande variedade de recursos de organização em lista.
- A capacidade para criar texto pré-formatado, incluindo texto tabulada.
- A capacidade de encaixar imagens gráficas diretamente em um documento HTML, que podem ser hiperlinks para outros documentos.
- Mapas de imagens clicáveis, com “host spots” (pontos quentes) para levá-lo a vários lugares, dependendo do ponto em que você deu o clique na imagem.
- Documentos HTML são arquivos escritos em ASCII - texto.
- Podem ser criados em qualquer editor de texto (vi, emacs, edit, notepad)
- Existem editores específicos para várias plataformas
- Existem conversores de vários formatos, por exemplo, doc para html
- A unidade mínima de informação é a “página”

LIMITAÇÕES DA HTML

Antes de entrarmos nos detalhes da HTML, precisamos discutir algumas limitações fundamentais da linguagem, não para desencorajá-lo a aprender a usá-la, mas para lhe dar uma perspectiva geral daquilo que pode esperar.

→ Não é uma linguagem tipográfica

A simples justificação da linha é fácil – ela está sempre alinhada à esquerda. Você não tem outra escolha. Os navegadores Web devoram quaisquer espaços em branco extras que você colocar nos seu documentos, a menos que sejam tomadas medidas especiais para preservá-los. Você não pode especificar um estilo de fonte, nem mesmo um corpo. E você pode conduzir um usuário a uma imagem gráfica, mas ele não irá necessariamente vê-la.

DIFERENÇAS FÍSICAS EM HARDWARE

Há uma grande variedade de sistemas de computadores, com diferentes hardware e diferentes recursos, e cada um reproduzirá o seu documento HTML de maneira diferente.

As diferenças de hardware constituem a limitação mais importante que afeta a HTML. Você não pode esperar que monitores monocromáticos mostrem cores nem pode esperar também que monitores de baixa resolução reproduzam imagens gráficas, e esses terminais não-gráficos não podem mostrar outra coisa a não ser texto puro, com uma única e monótona fonte fixa. Essas diferenças físicas significam que os seu documentos HTML serão reproduzidos diferentemente nos diversos PCs, nas estações de trabalho e nos diversos terminais. Por esses motivos, nem sempre é vantajoso se preocupar com os detalhes na apresentação com HTML. Por exemplo, querer dar o tom de cor perfeito, para um logotipo qualquer.

QUAIS AS FERRAMENTAS NECESSÁRIAS?

Embora haja disponíveis algumas ferramentas especializadas para criá-los e editá-los, você pode começar a criar HTML imediatamente com ferramentas que já tem à mão. Elas são:

- Os editores de texto internos do seu computador, como o Bloco de Notas do Microsoft Windows, o DOS Edit, o Macintosh Teach Text, UNIX vi ou emacs, ou VAX/VMS edt.
- O seu processador de texto preferido, como o PC-Write, Word, Word-Perfect, ou qualquer outro que você tenha usado sem formatação de texto puro, ou modo ASCII.

Na verdade, você pode usar qualquer ferramenta com a qual se sinta bem para criar documentos, tornando o processo de aprendizado HTML muito fácil. Você pode focalizar a sua atenção no aprendizado e criação HTML, sem ter de ao mesmo tempo aprender uma ferramenta especial.

EDITORES DE HTML

É possível desenvolver páginas HTML super-sofisticadas usando um simples editor de texto, mas certamente não é a forma mais cômoda e produtiva. Existem vários tipos diferentes de **editores e conversores para HTML** que podem facilitar muito a criação ou transformação de informação para o WWW. Há basicamente três tipos de ferramentas:

- Editores HTML
- Macros para processadores de texto
- Conversores

Os **editores HTML** são as ferramentas que geralmente oferecem mais liberdade para a criação de páginas. Isto é importante, pois o HTML é um padrão em constante desenvolvimento e a cada dia surgem novos elementos. A maioria desses editores não oferece uma interface WYSIWYG e geralmente dependem de um *browser* para se ter uma idéia do resultado final.

As **macros** têm a vantagem de usar o ambiente de processamento de textos, que geralmente é familiar ao usuário. Permitem que se crie uma página HTML sem escrever uma linha sequer de HTML, e nem ver o código. Essa facilidade de uso tem o preço de não permitir o uso de recursos mais sofisticados da linguagem. Existem macros para Word for Windows, WordPerfect e Ami Pro.

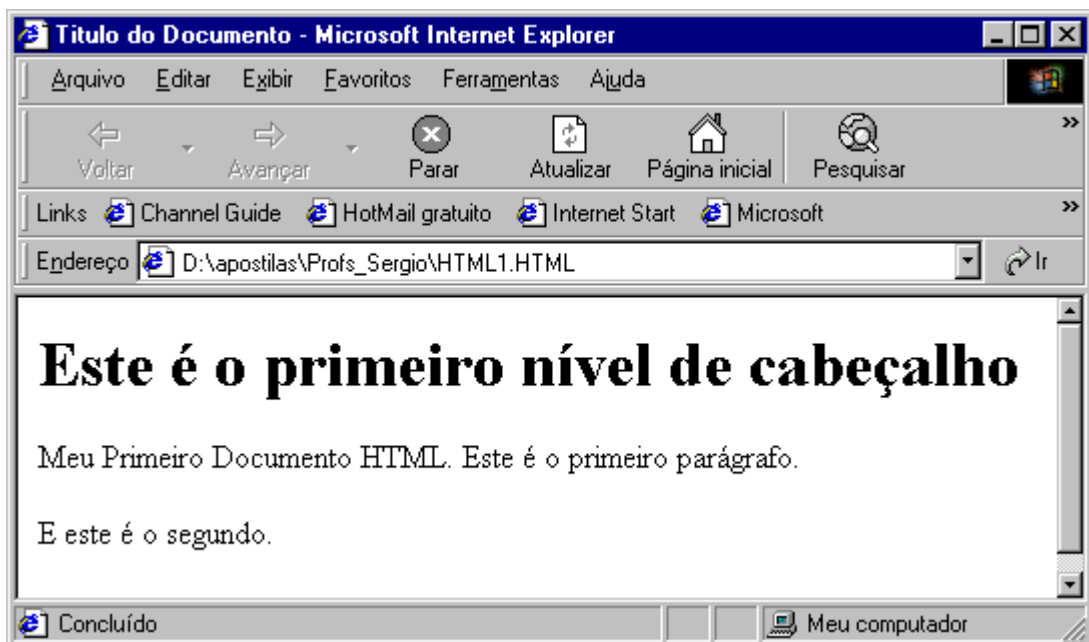
Finalmente, há os **conversores**, que permitem converter diversos formatos de texto para HTML, como PostScript, Texinfo, LaTeX, Framemaker, Word, RTF, WordPerfect, nroff, etc. Em geral, são limitados. Quase sempre é necessário fazer alguma edição após a conversão, mas são extremamente úteis quando se tem muita informação em um formato diferente e deseja-se torná-la disponível rapidamente no WWW.

A ESTRUTURA DE UM DOCUMENTO HTML

Todos os descritores HTML devem ser cercados por colchetes angulares (< e >) e tem de ter marcadores de início e fim. Todo documento deve ser identificado como HTML (<html> </html>), ter uma área de cabeçalho (<head></head>) com o título para o documento (<title> </title>), um título principal e uma área definida como corpo(<body></body>) do conteúdo do documento. Como o exemplo a seguir:

```
<HTML>
<HEAD>
<TITLE>Titulo do Documento</TITLE>
</HEAD>
<BODY>
<H1>Este é o primeiro nível de cabeçalho</H1>
Meu Primeiro Documento HTML.
Este é o primeiro parágrafo.<P>
E este é o segundo.<P>
</BODY>
</HTML>
```

Esta página vai aparecer assim:



OS MARCADORES 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, 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. 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

- A notação: <TITLE> (e a correspondente </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.

QUESTÕES REFERENTES À ACENTUAÇÃO

As notações presentes em marcações devem ser representadas com notações especiais para que possam ser exibidas em tela. Estas notações especiais sempre se iniciam por & (e comercial) e encerram-se com ; (ponto e vírgula).

Notação	descrição	Aparência
<	“Maior que”	>
>	“Menor que”	<
&	E comercial	&
"	aspas duplas	“
®	marca registrada	®
©	copyrights	©

É possível utilizar facilidades de acentuação (padrão do Windows por exemplo) em documentos HTML. No entanto, desta forma, apenas poderá visualizar a acentuação o computador que reconhecer este padrão específico.

Segue abaixo o padrão para acentuação - ISO Latin-1 alphabet -, semelhante ao utilizado para exibir caracteres das marcações, e que poderá ser visualizado pela grande maioria das máquinas.

Caracter	Notação
Acento agudo	&xacute; onde x é uma letra qualquer, maiúscula ou minúscula
Acento grave	&xgrave; onde x é uma letra qualquer, maiúscula ou minúscula
Acento circunflexo	◯ onde x é uma letra qualquer, maiúscula ou minúscula
Letra com til	&xtilde; onde x é uma letra qualquer, maiúscula ou minúscula
Letra com trema	&xuml; onde x é uma letra qualquer, maiúscula ou minúscula
Letras unidas	&Aelig; = Æ e æ = æ
Letra com argola	Å = Å e å = å
Cedilha	Ç = Ç e ç = ç
N com til	Ñ = Ñ e ñ = ñ
O cortado	Ø = Ø e ø = ø
THORN maiúsculo	Þ = Þ (Icelandic)
thorn minúsculo	þ = þ (Icelandic)
eth minúsculo	ð = ð (Icelandic)
eth maiúsculo	Ð = Ð (Icelandic)
Caracter alemão	ß = ß

MARCAÇÕES BÁSICAS

Títulos

Todo documento em HTML deve possuir um título. De um modo geral o título aparece em lugar separado da página (por exemplo, alto da tela no Netscape), e é utilizado para identificar o documento em outros contextos (por exemplo, buscas Wais). É interessante que o título possa sugerir claramente o conteúdo do documento.

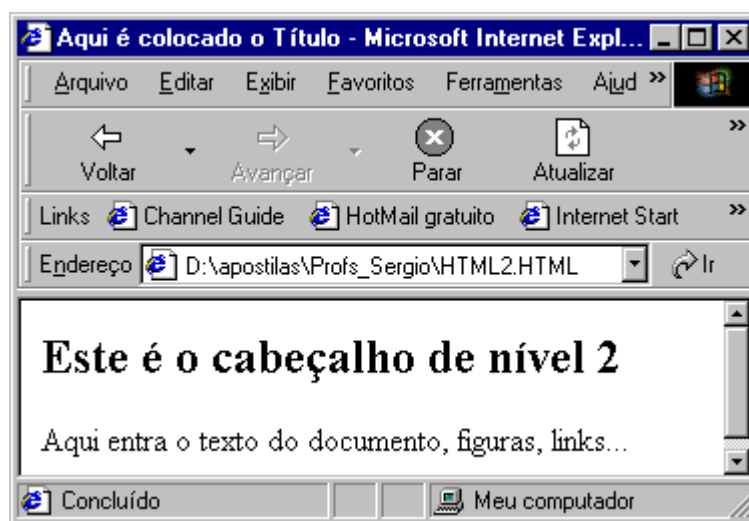
Atenção porque o conceito de título é diferente de cabeçalho. O título está mais para o nome do arquivo. Não é um elemento relevante na visualização do documento como acontece com o cabeçalho.

A marcação utilizada para títulos é <title> e seu par </title>.

Escrito desta forma:

```
<html>
<title> Aqui é colocado o Título</title>
<body>
<h2>Este é o cabeçalho de nível 2</h2>
Aqui entra o texto do documento, figuras, links...
</body>
</html>
```

Apresenta-se assim:



Cabeçalhos

"Cabeçalhos" normalmente são usados para títulos e sub-títulos de uma página.

O HTML possui seis níveis de cabeçalhos, numerados de 1 a 6, sendo o número 1 o de maior destaque. Cabeçalhos são exibidos em letras maiores e em negrito. O primeiro cabeçalho em cada documento deve estar marcado como <H1>.

ATENÇÃO: ao definir o tamanho de um cabeçalho, você não está definindo o tamanho da letra (fonte 10, fonte 14). Você apenas define que ele aparecerá com maior tamanho e destaque que o resto do texto. O tamanho exato com que ele será visualizado é definido pelo programa visualizador de html (browser) de cada pessoa que acessar a informação.

As notações relativas a cabeçalhos são:

<Hy>Texto do cabeçalho </Hy >

onde y é um número entre 1 e 6 especificando o nível do cabeçalho.

O conjunto de notações possíveis é o que segue:

<H1>Headings</H1>

<H2>Headings</H2>

<H3>Headings</H3>

<H4>Headings</H4>

<H5>Headings</H5>

<H6>Headings</H6>

Parágrafos

A marcação <p> é utilizada para definir o início de um novo parágrafo, deixando uma linha em branco entre cada parágrafo. HTML não reconhece o caracter de quebra de linha dos editores de texto. Mesmo que exista uma linha em branco, os clientes Web só reconhecem o início de um novo parágrafo mediante a marcação apropriada.

Existem algumas exceções. Por exemplo, inserindo uma marcação de parágrafo depois de <P>, , <Hy> e outras, esta será ignorada.

Quebras de linha

A marcação
 faz uma quebra de linha sem acrescentar espaço extra entre as linhas. Veja a diferença do uso de <p> e
 ,nos exemplos a seguir:

```
<html>
```

```
<title>Quebra de Linha</title>
```

```
<body>
```

```
<h1>Utilizando p</h1>
```

```
Vamos separar esta sentença com marcação de parágrafo.<p>
```

```
Veja o espaço extra entre as linhas.
```

```
</body>
```

```
</html>
```

```
<html>
```

```
<body>
```

```
<h1>Utilizando br</h1>
```

```
Veja a diferença quando separamos duas linhas utilizando<br>
```

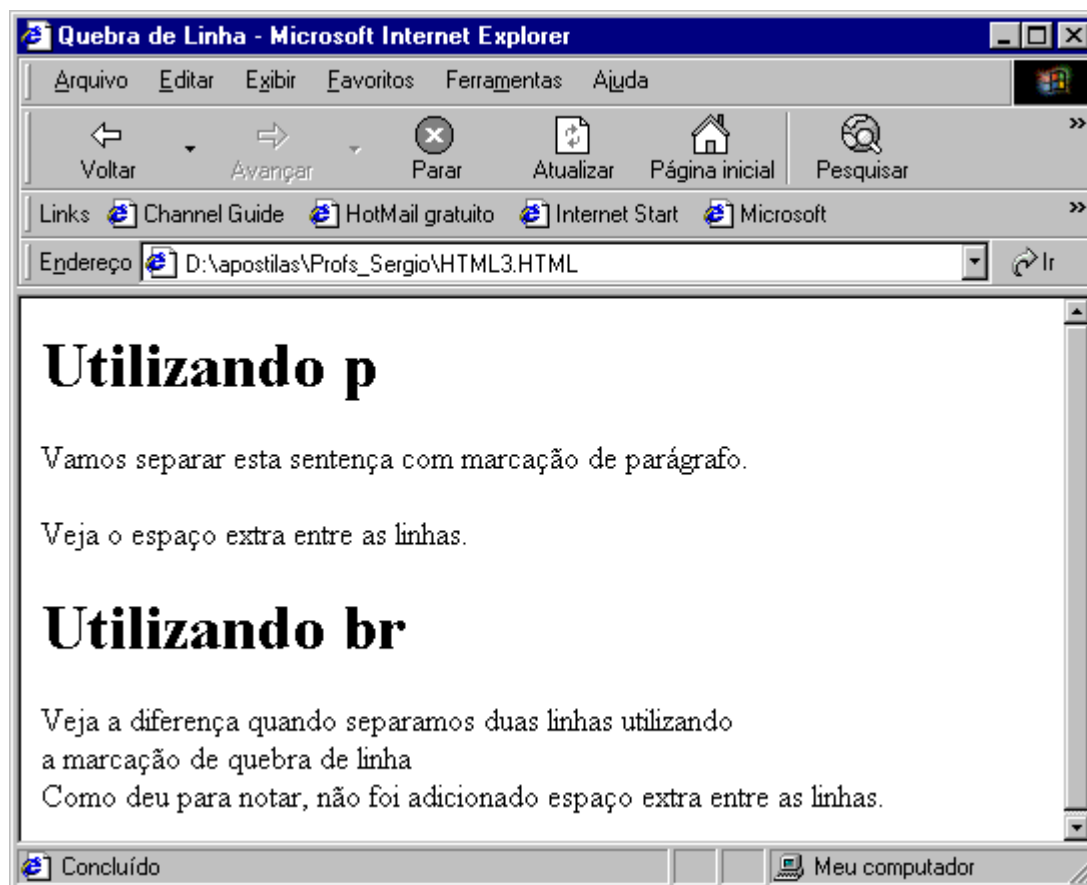
```
a marcação de quebra de linha<br>
```

```
Como deu para notar, não foi adicionado espaço extra entre as linhas.
```

```
</body>
```

```
</html>
```

Veja a diferença no uso das duas notações:



INTERLIGANDO DOCUMENTOS

Para começar

O principal poder do HTML vem da sua capacidade de interligar partes de um texto (e também imagens) a outros documentos. Os clientes exibem em destaque estas áreas ou pontos chaves (normalmente com cores diferentes ou sublinhado) para indicar que se trata de um link, ou interligação, no hipertexto.

A marcação `<A>`, que define o ponto de partida para os links, é denominada de âncora. Para incluir uma âncora em seu documento:

1. Inicie a âncora com `<A .` (Há um espaço depois de A.)
2. Especifique o documento a ser interligado, inserindo parâmetro `href="filename"` seguido do sinal: `>`
3. Insira o texto que vai funcionar como link no documento corrente
4. Anote a marcação de final da âncora: ``.

Um exemplo de referência a um hipertexto:

```
<A href="ListaPraias.html">Praias</A>
```

A palavra "Praias" é definida como o marcador do link para se chegar ao documento `ListaPraias.html`, que está no mesmo diretório do documento corrente. Ou seja, "Praias" aparece em negrito e se eu clicar nessa palavra será exibido o documento apontado - `ListaPraias.html`

Interligando Documentos em Outros Diretórios

É possível interligar documentos em outro diretório especificando-se o caminho relativo a partir do documento corrente, em relação ao que está sendo interligado.

Por exemplo, um link para o arquivo Sergipe.html localizado no subdiretório Estados seria assim:

```
<A HREF="Estados/Sergipe.html">Sergipe</A>
```

Estes são os denominados links relativos. É também possível usar o caminho completo (pathname absoluta) do arquivo desejado. Para isso, utiliza-se a sintaxe padrão do sistema UNIX.

Importante: se você quiser referenciar um diretório a partir do “raiz” do seu servidor www, inicie a notação com /. Isto é, se você tiver uma notação desta forma , significa que o arquivo incon1.gif deverá ser buscado no diretório imagens que está imediatamente acima do diretório raiz do seu servidor WWW.

Ligando URLs

O World Wide Web utiliza-se das Uniform Resource Locators (URLs) para especificar a localização de arquivos em servidores. Uma URL inclui o tipo de recurso acessado (p.e, gopher, WAIS), o endereço do servidor, e a localização do arquivo.

Se escreve de acordo com o esquema abaixo:

```
protocolo://servidor[:port]/path/filename
```

Onde protocolo é uma das seguintes definições

file	um arquivo no seu sistema local, ou servidor de ftp
http	um arquivo em um servidor World Wide Web
gopher	um arquivo em um servidor Gopher
WAIS	um arquivo em um servidor WAIS
news	um Usenet newsgroup
telnet	uma conexão Telnet

Por exemplo, para incluir um link para o site de busca CADÊ, em um documento, deve ser usado:

```
<A HREF = "http://www.cade.com.br">Cade – Site de busca </A>
```

O que torna a sentença “Cade – Site de busca” um link para tal documento.

Ligando sessões específicas em outro documento

Para fazer links com sessões de outros documentos o processo é o mesmo da sessão anterior. Faz-se uma âncora normalmente, e apenas é acrescentado, após a url do documento de destino, o sinal # e uma palavra, ou identificador. O identificador é "volta" no nosso exemplo:

Este é o meu `link` para o último item do índice deste tutorial.

No índice do tutorial - arquivo index20.htm - deverá haver uma notação que reconheça a palavra/identificador informado no ponto de partida. Pode ser: `Formulários Eletrônicos`.

Interligando sessões em uma página

HTML permite que se façam ligações entre diferentes trechos de um documento. Você pode apontar uma palavra ou trecho de um texto utilizando uma âncora de nome (named anchor). Este recurso é bastante usado na geração de índices de conteúdo no início de uma página.

O passo a passo é :

1. Defina o "ponto de partida", atribuindo a ele um nome qualquer - no exemplo item1-, precedido do caracter #, da seguinte forma:

Veja o início da `segunda sessão`.

2. Defina o "ponto de chegada", atribuindo a ele uma âncora de nome, com o nome correspondente ao do ponto de partida, assim:

Esta `segunda sessão` trata de..

LISTAS

Listas Não Numeradas

Para criar uma lista não numerada:

1. Comece com a marcação de início de lista: ``.
2. Insira a marcação `` antes de cada item da lista item. (Não é necessário fechar a marcação com ``)
3. Encerre com a marcação de fim de lista: ``.

Abaixo o exemplo de uma lista com quatro itens:

```
<UL>
<LI> maçãs
<LI> uvas
<LI> laranjas
<LI> bananas
</UL>
```

Vai aparecer assim:



Cada marcação pode conter vários parágrafos.

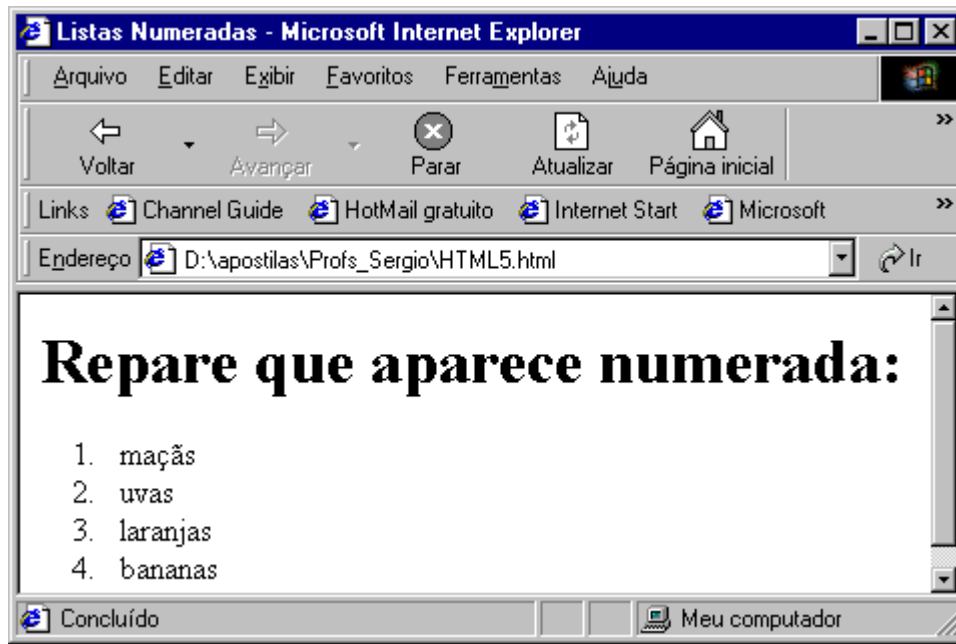
Listas Numeradas

Uma lista numerada - ou ordenada - é semelhante a uma lista não numerada, exceto porque utiliza marcação ao invés de . Os itens são identificados utilizando-se a mesma notação .

Uma lista criada assim:

```
<OL>  
<LI> maçãs  
<LI> uvas  
<LI> laranjas  
<LI> bananas  
</OL>
```

Resulta neste formato:



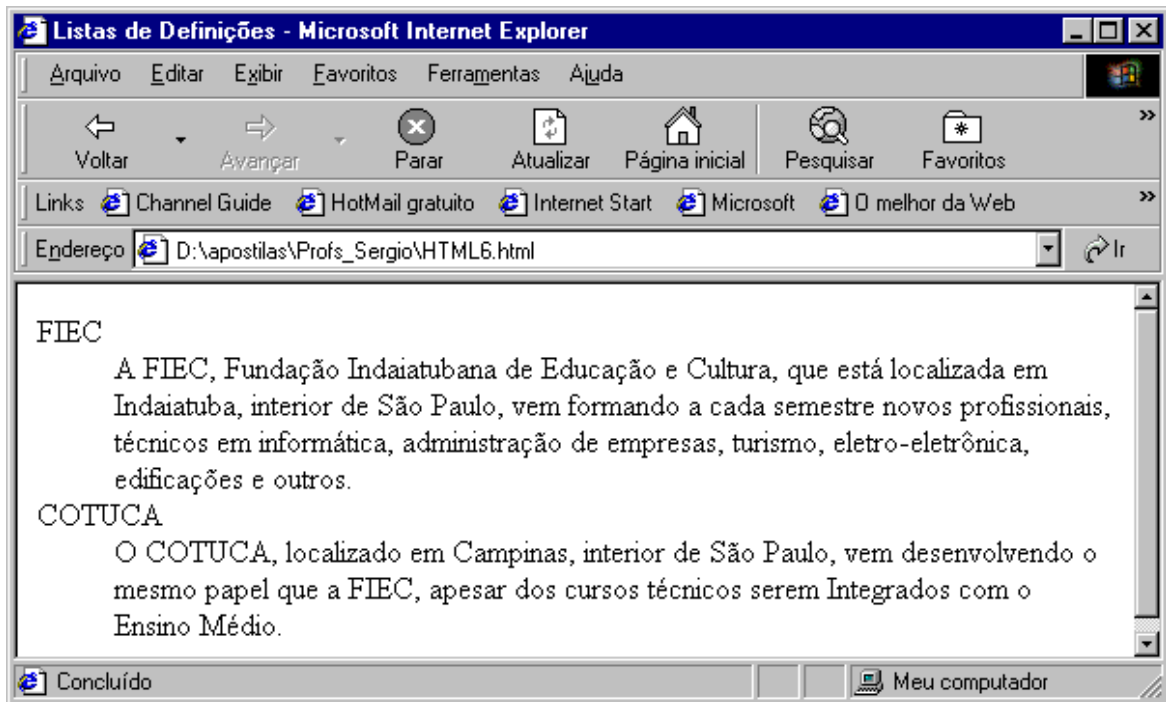
Listas de Definições

Uma lista de definições normalmente consiste em alternar um termo (abreviado como DT) e uma definição (abreviado como DD). Clientes Web browsers geralmente mostram a definição em nova linha com outro alinhamento.

Exemplo de uma lista de definição:

```
<DL>
<DT> FIEC
<DD> A FIEC, Fundação Indaiatubana de Educação e Cultura, que está localizada em
    Indaiatuba, interior de São Paulo, vem formando a cada semestre novos profissionais,
    técnicos em informática, administração de empresas, turismo, eletro-eletrônica,
    edificações e outros.
<DT> COTUCA
<DD> O COTUCA, localizado em Campinas, interior de São Paulo, vem desenvolvendo
    o mesmo papel que a FIEC, apesar dos cursos técnicos serem Integrados com o Ensino
    Médio.
</DL>
```

Será vista desta forma:



As entradas <DT> e <DD> podem conter vários parágrafos (separados por <P>), listas, ou outras definições.

A marcação <DD> pode também ser utilizada, fora de uma lista de definições como se fosse uma tabulação (insere um espaço no início da frase).

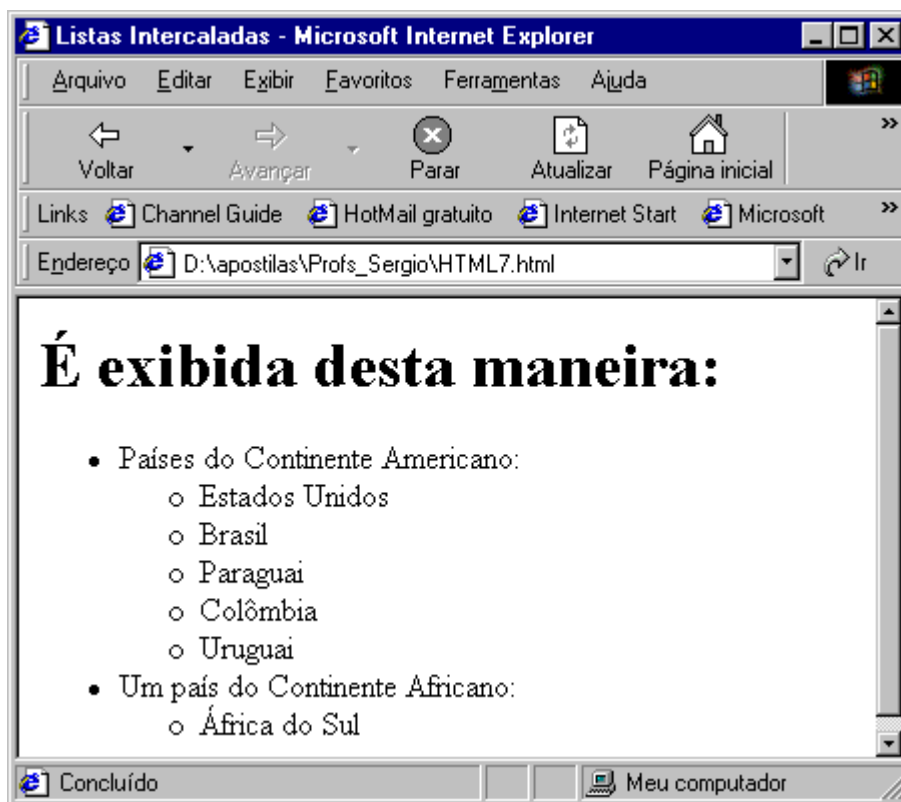
Listas Intercaladas

As listas podem ser encadeadas ou intercaladas arbitrariamente, produzindo resultados bastante interessantes. A prática é que vai mostrar qual o número máximo de listas que vale a pena intercalar. Pode-se inclusive ter um parágrafo, intercalado com uma lista que contenha um único item.

Exemplo de lista intercalada:

```
<UL>
  <LI> Países do Continente Americano:
    <UL>
      <LI> Estados Unidos
      <LI> Brasil
      <LI> Paraguai
      <LI> Colômbia
      <LI> Uruguai
    </UL>
  <LI> Um país do Continente Africano:
    <UL>
      <LI> África do Sul
    </UL>
</UL>
```

É exibida com este formato:



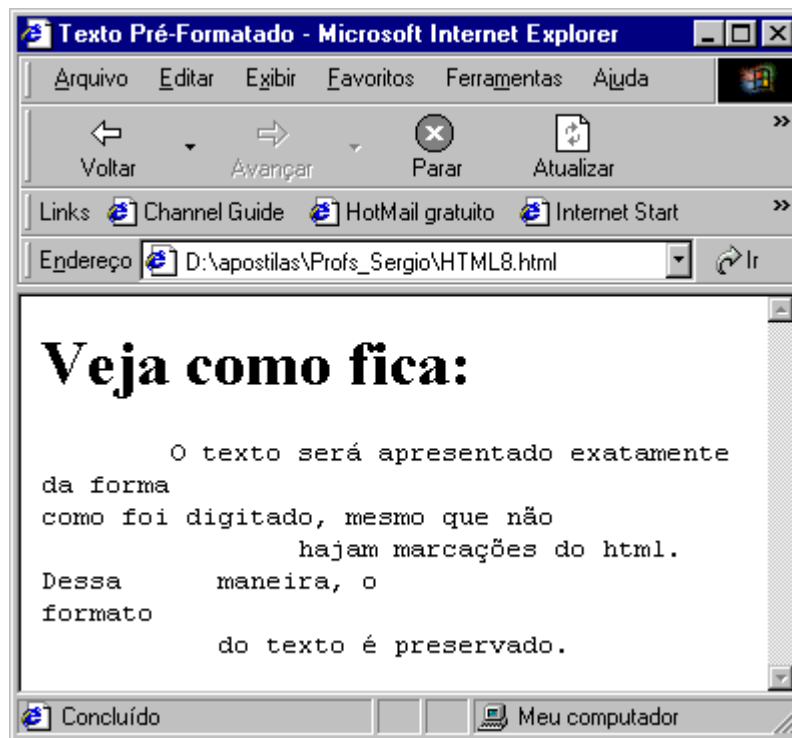
TEXTO PRÉ-FORMATADO

A marcação <PRE>(derivada de "preformatted") delimita uma área de texto em que espaços, novas linhas e tabulações são mantidas. Ou seja, o texto será apresentado exatamente da forma como foi digitado, mesmo que não hajam marcações do html. É uma forma de se preservar o formato de um texto

Este recurso é utilizado por exemplo para listas de programas. Vejamos um exemplo:

```
<PRE>      O texto será apresentado exatamente
da forma
como foi digitado, mesmo que não
                hajam marcações do html.
Dessa  maneira, o
formato
                do texto é preservado.
</PRE>
```

Apresenta-se assim:



Podem ser definidos links dentro de textos pré-formatados. No entanto, deve-se evitar utilizar outros caracteres das marcações em áreas <PRE>. Isto porque caracteres como <, >, and & possuem um significado específico no HTML.

O ideal é utilizar as sequências especiais, (<, >, and &, respectivamente) sempre que precisar representar esses caracteres. Veremos mais sobre no próximo item.

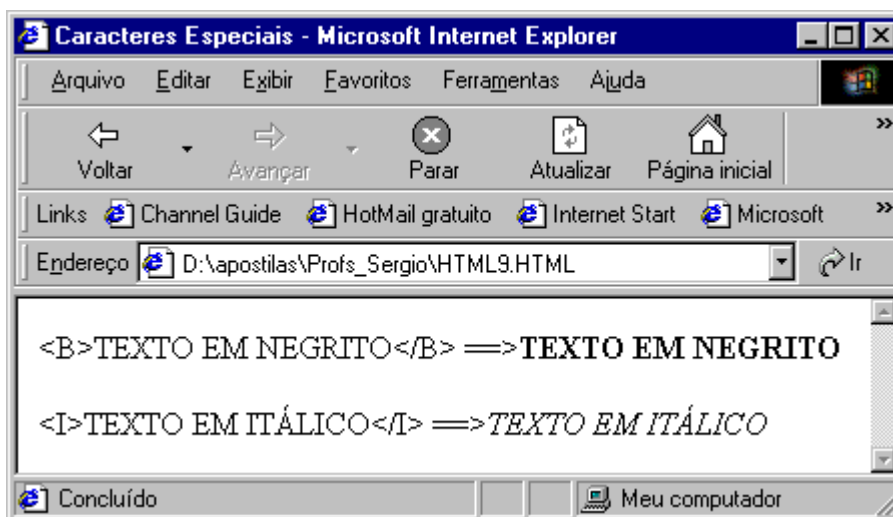
CARACTERES ESPECIAIS

A HTML permite que se apliquem estilos específicos a palavras ou sentenças. Vamos ver algumas opções:

 texto em negrito

<I></I> texto em itálico

E o efeito de cada uma delas:



Para representar os caracteres das tags, utilize as seguintes notações:

- < para representar <
- > para representar >
- & para representar &
- " para representar "

Uma lista completa dessas notações especiais pode ser obtida no CERN, em:
<http://www.w3.org/hypertext/WWW/MarkUp/ISolat1.html>

Caso o editor que você esteja utilizando não acentue, é possível também utilizar notações especiais para acentuação.

Atenção: Ao contrário do resto do HTML, as notações especiais diferenciam letras maiúsculas de minúsculas. Não é possível usar < no lugar de <.

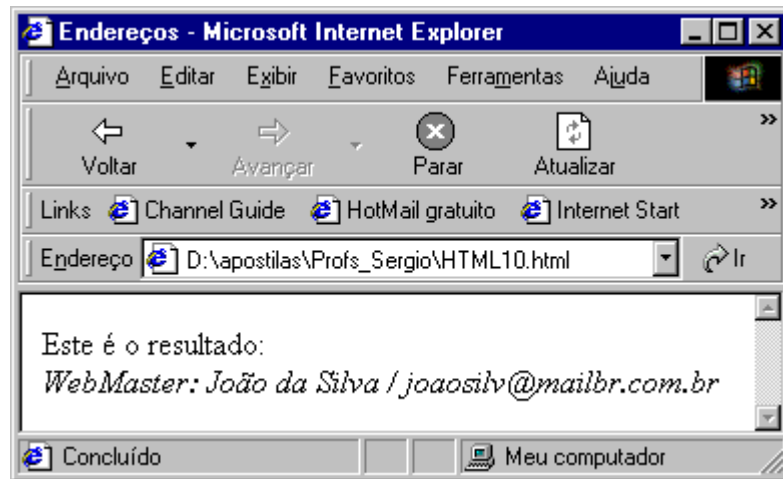
ENDEREÇOS

A marcação <ADDRESS></ADDRESS> é habitualmente utilizada para especificar o autor do documento e a forma de contato com ele (por exemplo, o endereço eletrônico). Geralmente é o último item de uma página.

Por exemplo:

```
<ADDRESS>  
WebMaster: João da Silva / joaosilv@mailbr.com.br  
</ADDRESS>
```

E o resultado é:



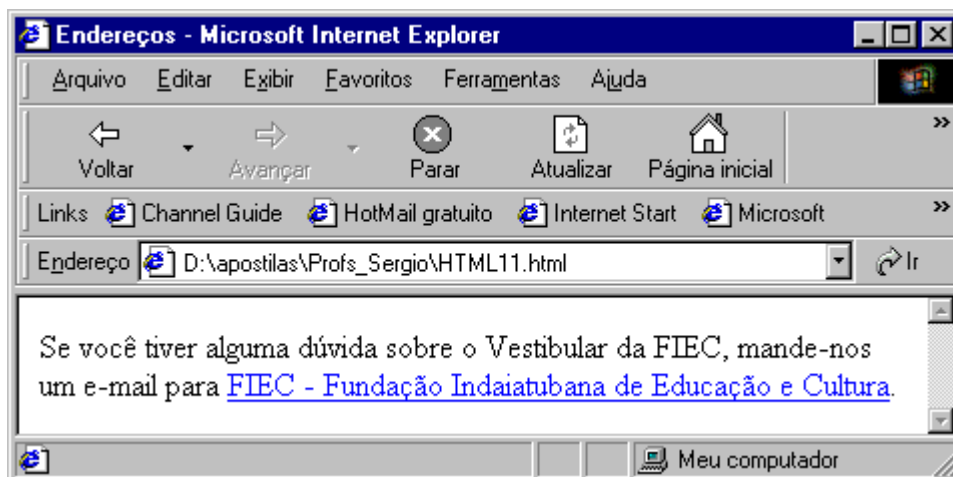
Existe ainda um recurso para criar um item na página que, ao ser selecionado, abre uma janela, e permite que se envie uma mensagem eletrônica para um endereço específico. Para isso utiliza-se a variável "mailto:"

Veja o exemplo abaixo, para enviar mensagens de dúvidas para o endereço eletrônico apoio@cr-df.rnp.br

Se você tiver alguma dúvida sobre o Vestibular da FIEC, mande-nos um e-mail para

[FIEC – Fundação Indaiatubana de Educação e Cultura](mailto:fiec@fipec.com.br).

Fica assim:



E ao se clicar em “FIEC – Fundação Indaiatubana de Educação e Cultura”, se abrirá uma janela para envio de mensagem eletrônica.

INSERIR IMAGENS

As imagens dentro de uma página devem estar preferencialmente no formato ".gif". Este é o formato mais universalmente aceito pelos visualizadores de WWW (browsers). O

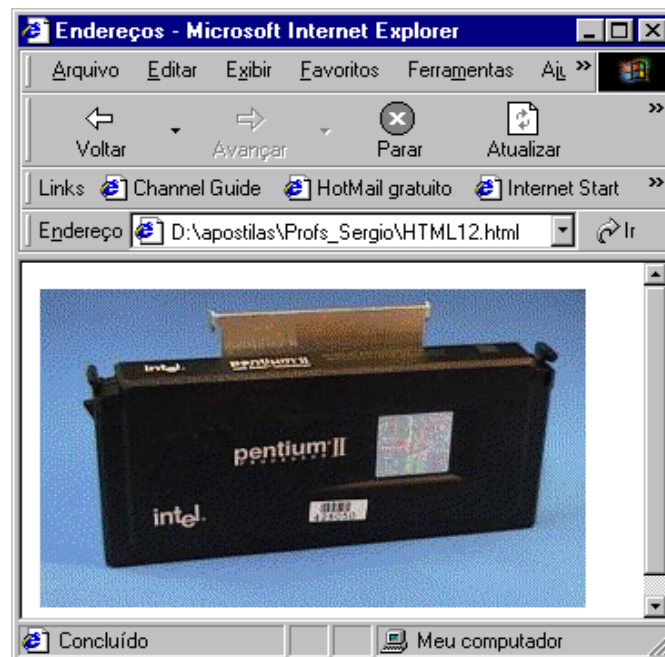
Netscape consegue decodificar outros formatos como o ".jpg" e ".rgb", mas esta capacidade não é comum a todos os programas.

Utiliza-se a seguinte notação para incluir uma imagem:

```

```

No exemplo:



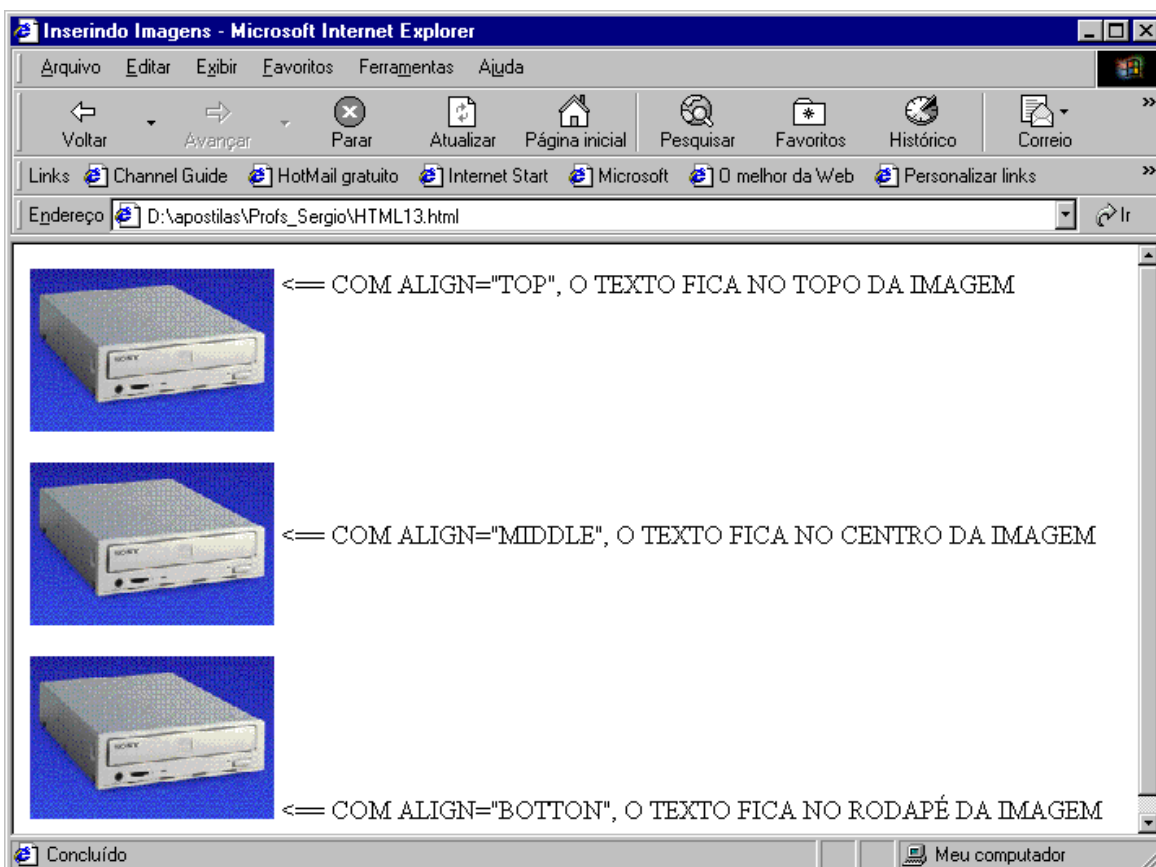
- Alinhamento das imagens

Toda imagem incluída pode ser posicionada (align=...) na tela das seguintes formas: TOP, MIDDLE, BOTTOM, LEFT e RIGHT. Esse comando é colocado conforme a seguinte notação:

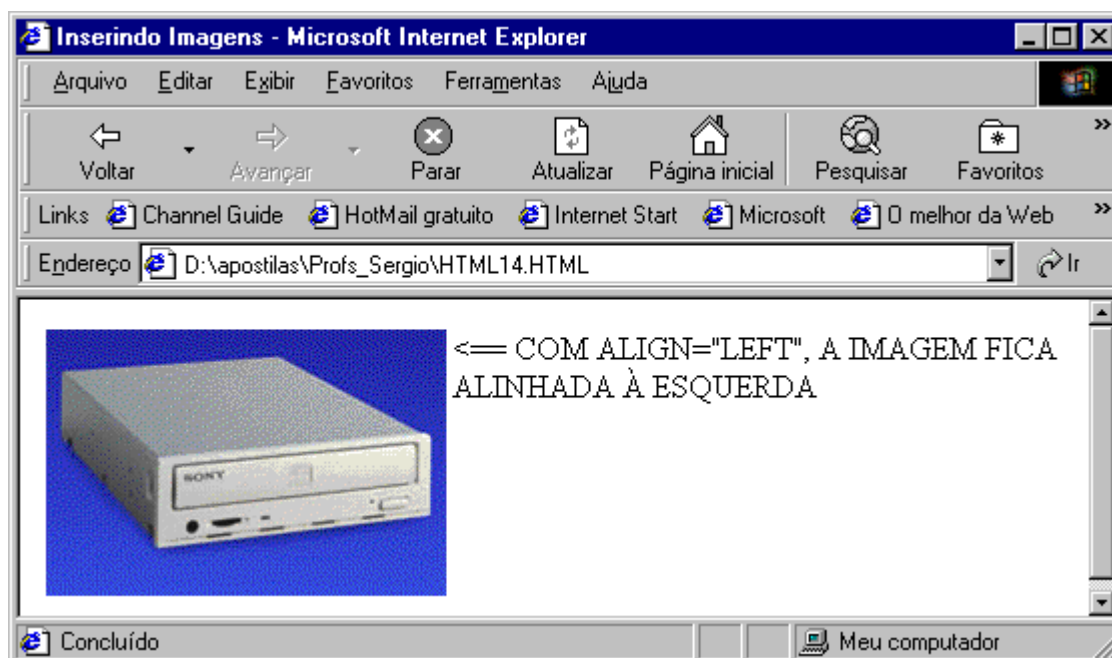
```

```

As três primeiras opções definem a posição da imagem com relação ao texto. Veja os exemplos:

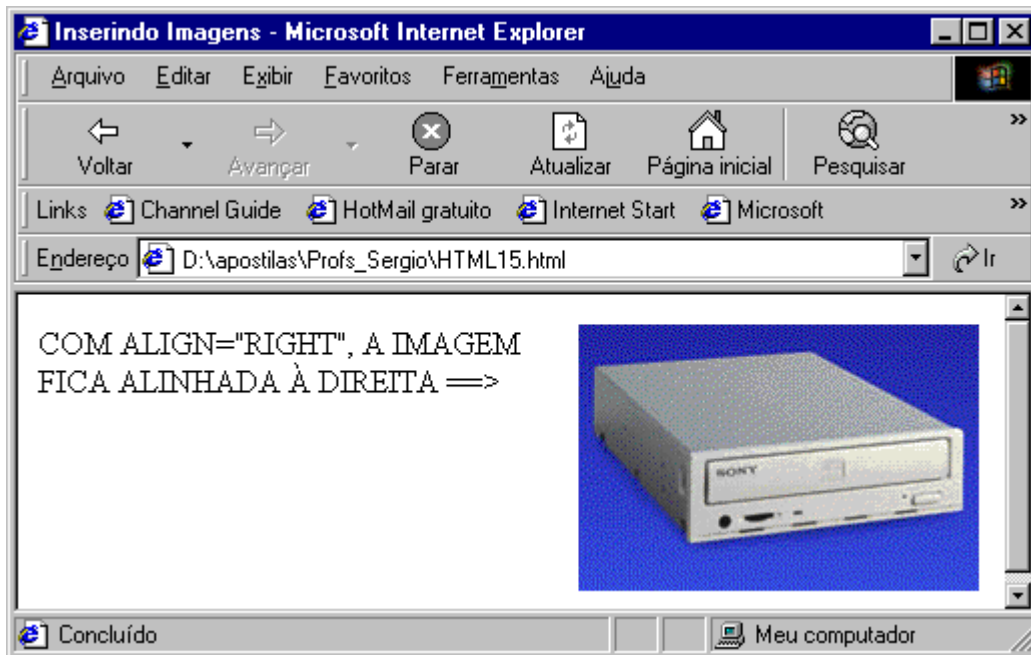


As duas últimas definem a posição da imagem na página, a direita ou esquerda:



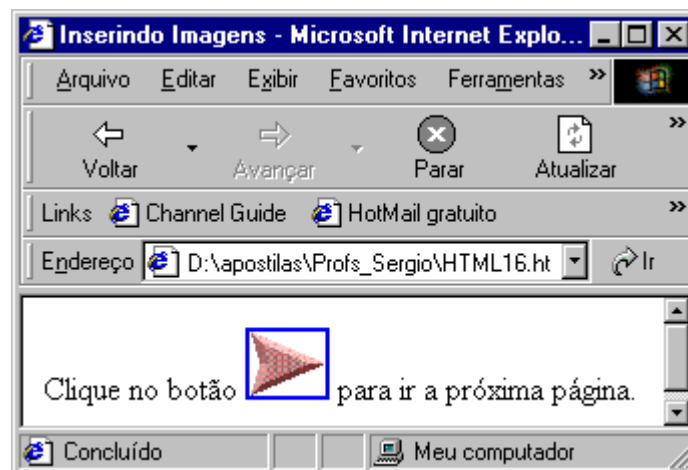
Atenção: De um modo geral não é necessário incluir essa anotação para alinhar a imagem a esquerda. Caso seja omitida a opção "align", a imagem aparecerá a esquerda da página.

Uma imagem pode ser também alinhada à direita da página:



Uma imagem pode ser ainda um botão sensível: Para isso basta incluir uma imagem no interior de uma âncora.

```
<a href="/index.htm"></a>
```



TABELAS EM HTML

Principais Marcações

Segue abaixo o conjunto de marcações essenciais para desenhar tabelas em HTML. Alguns atributos podem ser definidos para cada uma dessas marcações. Falaremos desses atributos mais adiante.

- `<TABLE></TABLE>`

Toda tabela deve ser iniciada com a marcação `<table>` e encerrada com `</table>`.

ATENÇÃO: Dessa forma a tabela vai aparecer sem bordas, mas dividindo linhas e células.

Para que a tabela apareça com bordas simples, defina dessa forma:

```
<table border></table>
```

Antes e depois de uma tabela, acontece sempre uma quebra de linha:

```
<TR></TR>
```

Cada linha de uma tabela deve sempre aparecer entre as marcações <tr> e </tr>. (TR = Table Row)

- <TD></TD>

Esta é a marcação que define cada célula de uma tabela. As células de uma tabela devem sempre aparecer entre as marcações de linhas (<tr> e </tr>). Como padrão, o texto nas células é alinhado a esquerda.

- <TH></TH>

Desta forma são definidos os títulos de uma tabela. Estes podem ser posicionados em qualquer célula. A diferença entre a marcação de célula e título de célula é que o título aparece em negrito.

Vamos ver então o exemplo de uma tabela utilizando essas marcações básicas.

```
<table border>
```

```
<tr>
```

```
<td>Automóveis/Descrição</td>
```

```
<th>Preço</th><th>Ano</th><th>Cor</th>
```

```
</tr>
```

```
<tr>
```

```
<th>Omega</th><td>94</td><td>R$ 16.000,00</td><td>Preto</td>
```

```
</tr>
```

```
<tr>
```

```
<th>Vectra</th><td>95</td><td>R$ 19.000,00</td><td>Cinza</td>
```

```
</tr>
```

```
<tr>
```

```
<th>Pálio</th><td>97</td><td>R$ 8.000,00</td><td>Branco</td>
```

```
</tr>
```

```
</table>
```

Vai aparecer dessa forma:



Atributos

As marcações das tabelas, podem apresentar resultados diferentes, se acompanhadas de alguns atributos. Vamos ver os principais:

- **<BORDER>**

Esse atributo aparece junto a marcação TABLE. Caso esse atributo não apareça, a tabela não terá bordas.

Ou seja, como já vimos, para obter uma tabela com bordas, ela deve ser delimitada pelas marcações `<table border>` `</table>`.

- **<ALIGN>**

Este atributo pode ser aplicado a TR, TH e TD, e controla como será o alinhamento do texto dentro de uma célula, com relação às bordas laterais.

Aceita os valores left, center, right, respectivamente para alinhar a esquerda, centralizar ou alinhar a direita.

Veja o exemplo:

```
<table border>
```

```
<tr>
```

```
<td>Primeira Célula</td><td>Segunda Célula</td><td>Terceira Célula</td>
```

```
</tr>
```

```
<tr>
```

```
<td align=center>centro</td>
```

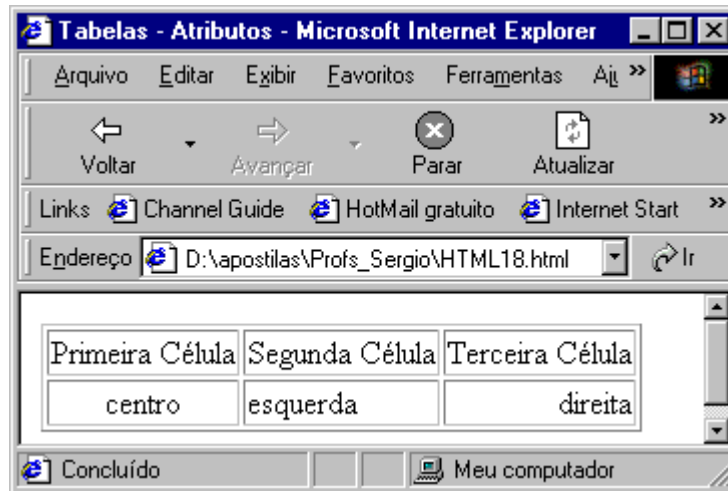
```
<td align=left>esquerda</td>
```

```
<td align=right>direita</td>
```

```
</tr>
```

</table>

Aparece assim:



- <VALIGN>

Pode ser aplicado a TR, TH e TD, e define o alinhamento do texto nas células com relação a borda superior e inferior.

Aceita os valores *top*, *middle* e *bottom*. Exemplo:

<table border>

<tr>

<td> Veja esse teste para alinhamento

Preste atenção nas bordas

inferior e superior

</td>

<td valign=top>Com TOP </td>

<td valign=middle>Com MIDDLE</td>

<td valign=bottom>Com BOTTOM</td>

</tr>

</table>

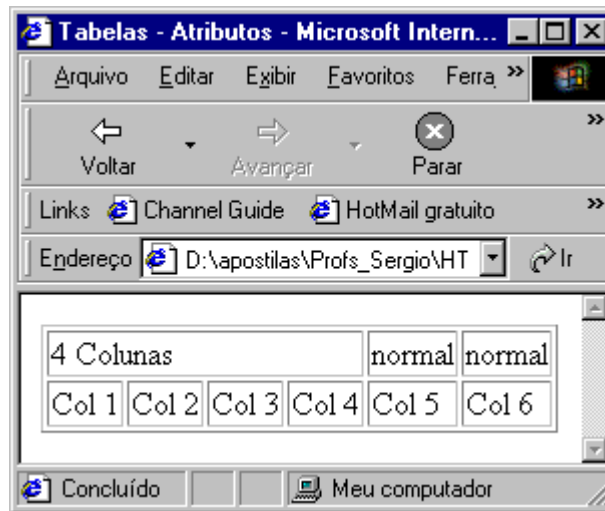
Aparece assim:



- **<NOWRAP>**
Este atributo evita que haja uma quebra de linha dentro de uma célula. Cautela ao utilizá-lo, para não produzir células muito largas.
- **<COLSPAN>**
Pode ser aplicado a TH ou TD. Define quantas colunas uma célula poderá abranger. Por padrão cada célula corresponde a uma coluna, ou seja COLSPAN=1.
Vejam os exemplos

```
<table border=1>
<tr>
<td colspan=4>4 Colunas</td><td>normal</td><td>normal</td>
</tr>
<tr>
<td>Col 1</td><td>Col 2</td><td>Col 3</td><td>Col 4</td><td>Col 5</td><td>Col
6</td>
</tr>
</table>
```

Que fica assim:



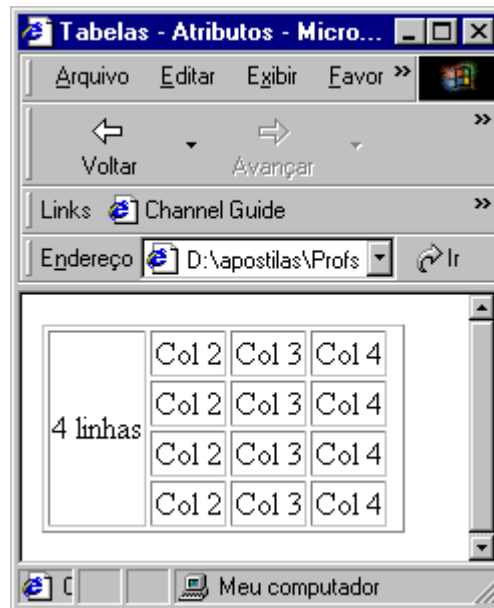
- <ROWSPAN>

Este atributo pode ser aplicado a células (TH e TD) e define quantas linhas uma mesma célula pode abranger. Assim como na marcação anterior, o padrão é uma célula corresponder a uma linha.

Novamente, vamos ver exemplos:

```
<table border>
<tr>
<td rowspan=4>4 linhas</td>
<td>Col 2</td><td>Col 3</td><td>Col 4</td><td>
<tr>
<td>Col 2</td><td>Col 3</td><td>Col 4</td><td>
</tr>
<tr>
<td>Col 2</td><td>Col 3</td><td>Col 4</td><td>
</tr>
<tr>
<td>Col 2</td><td>Col 3</td><td>Col 4</td><td>
</tr>
</table>
```

Apresenta-se dessa forma:



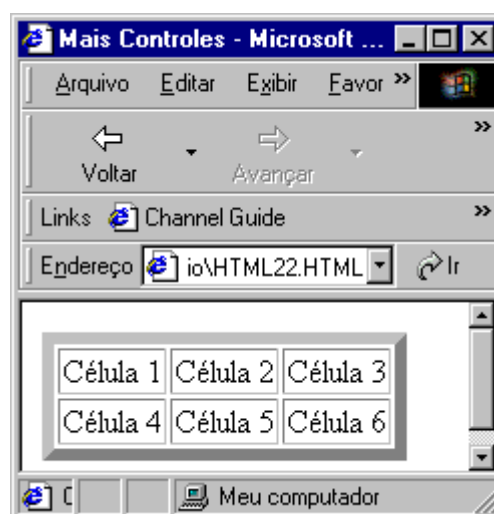
Mais elementos de controle

Vamos ver agora mais um conjunto de extensões que permitem maior controle sobre tabelas. Estas são especialmente úteis na criação de múltiplas tabelas intercaladas.

- **BORDER=<value>**

Acrescentando um valor ao atributo **BORDER** é possível colocar tabelas em maior destaque:

```
<table border=6>
<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>
```



- **CELLSPACING=<value>**

Este atributo é aplicável à marcação TABLE. Como padrão, o Netscape 1.1 utiliza espaço 2 entre as células. Este atributo define o espaço entre cada célula na tabela.

```
<table border cellspacing=7>
<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>
```



- **CELLPADDING=<value>**

Este atributo é aplicado também à marcação TABLE e define a distância entre o texto e a borda de cada célula.

Veja exemplo:

```
<table border cellpadding=8>
<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>
```



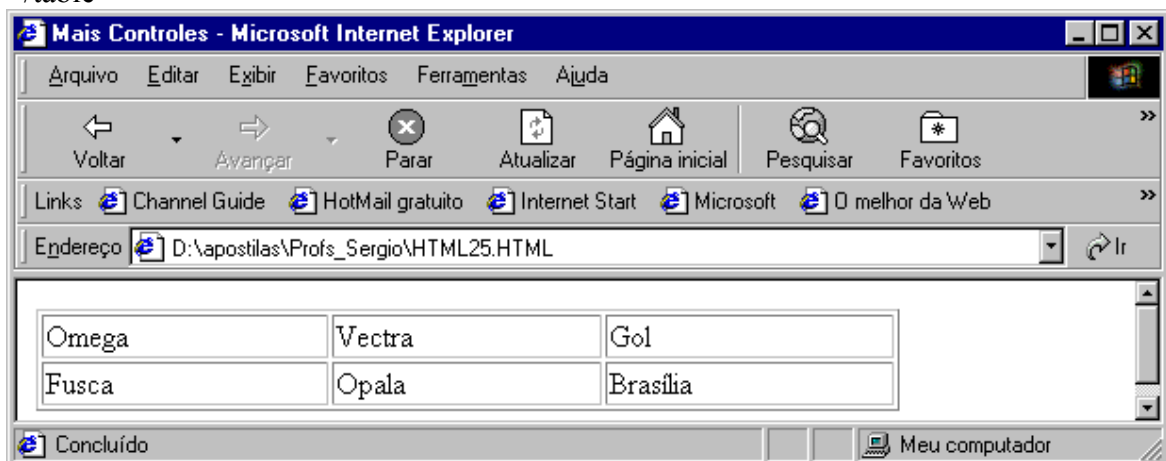
A tabela mais compacta possível seria definida como: `<table border=0 cellpadding=0 cellspacing=0>`

- `WIDTH=<value or percent>`

Este atributo pode ser aplicado tanto a TABLE como a TH e TD. A ele tanto pode ser associado um valor em "pixels" (ponto em tela gráfica), como um valor percentual. Ou seja, determina o quanto da tela uma tabela ou célula de tabela deverá ocupar.

Vamos ver os exemplos:

```
<table border width=80%>
<tr>
<td>Omega</td>
<td>Vectra</td>
<td>Gol</td>
</tr>
<tr>
<td>Fusca</td>
<td>Opala</td>
<td>Brasília</td>
</tr>
</table>
```



Segundo exemplo - aplicando width a uma célula:

```
<table border>
<tr>
<td width=60%>Omega</td>
<td>Vectra</td>
<td>Gol</td>
</tr>
<tr>
<td>Fusca</td>
<td>Opala</td>
<td>Brasília</td>
</tr>
</table>
```

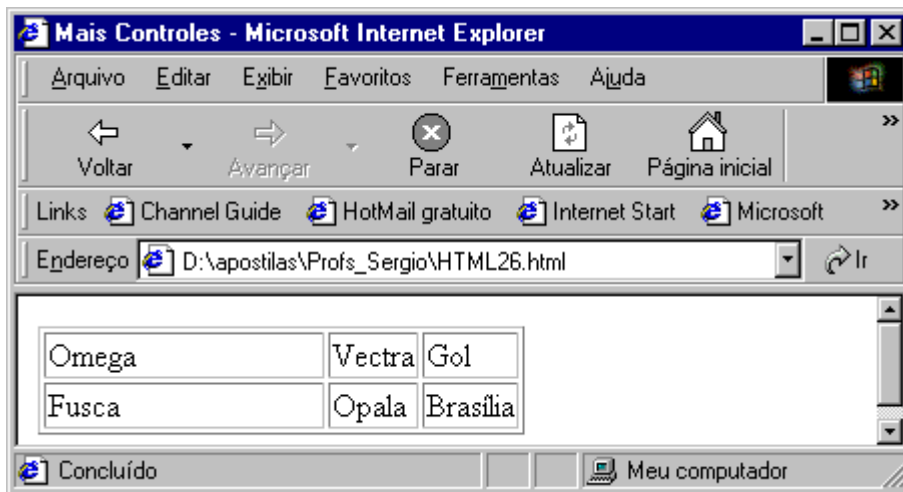


Tabela de comandos

- **Marcações:**

`<table>` e `</table>` = define início e fim da tabela.

`<tr></tr>` = define cada linha da tabela

`<td></td>` = define cada célula da tabela. Um conjunto de células aparece sempre delimitado pelas marcações de linha (`<tr></tr>`).

`<th></th>` = define títulos na tabela. Na prática são células onde o texto aparece em destaque (negrito). Podem aparecer em qualquer posição na tabela.

- **Atributos**

`<border>` = determina que uma tabela tenha bordas. Aparece sempre associado a marcação `<TABLE BORDER>`.

`<align>` = Controla o alinhamento horizontal do texto em uma linha ou célula. Pode aparecer associado a TR, TD ou TH. Aceita os valores:

left = alinha à esquerda	right = alinha à a direita	center = centraliza
--------------------------	----------------------------	---------------------

Como nos exemplos:

<tr align=center>	<td align=left>	<th align=right>
-------------------	-----------------	------------------

<valign> = controla o alinhamento vertical do conteúdo de cada célula da tabela. Pode aparecer associado a TR, TD ou TH. Aceita os valores:

top = alinha com o alto da célula

middle = alinha no meio

bottom = alinha com a parte de baixo da célula

Como nos exemplos:

<tr align=top>

<td align=middle>

<th align=bottom>

<rowspan> = Aparece associado a células (TD ou TH) e determina quantas linhas uma célula abrange.

Ex: <rowspan=2> , <rowspan=3> ...

<colspan> = Aparece associado a células (TD ou TH) e determina quantas colunas uma célula abrange.

Ex: <colspan=2> , <colspan=3> ...

- **Atribuição de Valores**

BORDER=<value> = define a largura da borda;

CELLSPACING=<value> = define espaçamento entre células ;

CELLPADDING=<value> = define distância entre o texto e a borda das células ;

WIDTH=<value or percent> = determina o quanto da tela uma tabela deve ocupar;

FORMULÁRIOS

Codificação básica

Um formulário começa com a marcação <FORM> e termina com </FORM> , mas Outros itens devem ser especificados:

- Primeiro, o formulário precisa saber como enviar a informação para o servidor. Existem dois métodos, GET e POST.

METHOD="GET"

A maioria dos documentos HTML são recuperados a partir da requisição de uma única URL ao servidor. Assim, um formulário que utilize este método, envia toda sua informação ao final da URL ativada.

METHOD="POST"

Este método transmite toda a informação fornecida via formulário, imediatamente após a URL ativada. Ou seja, quando o servidor recebe uma ativação de um formulário utilizando POST ele sabe que precisa continuar "ouvindo" para obter a informação. Este é o melhor método.

- Segundo, o formulário precisa saber para onde enviar a informação. Esta é a URL sendo ativada a partir do formulário, e ela é referenciada através da marcação ACTION. Esta URL em geral aponta para um script CGI que irá receber e decodificar os resultados. Lembre-se que se você está referenciando um script que reside no mesmo servidor do formulário, você não precisa incluir a URL completa.

ACTION="/cgi-bin/post-query" para um script no seu servidor

ACTION="http://www.cr-df.rnp.br/cgi-bin/post-query" para um script no servidor do CR/DF

Após você construir estas marcações, seu formulário geralmente terá a seguinte estrutura:

```
<FORM METHOD="POST" ACTION="/cgi-bin/post-query">
```

Marcações de campos de entrada e HTML em geral

```
</FORM>
```

Observe que este formulário utiliza o método POST e envia as informações digitadas para um script local chamado post-query no diretório /cgi-bin do servidor.

Outra informação importante: cada marcação de entrada em um formulário tem uma opção NAME associada, de tal forma que o script saiba qual o nome, isto é, como chamar cada valor digitado. Certamente você pode definir mais de um campo de entrada textual ou menu dentro de um formulário, mas certifique-se de que cada um possui um nome diferente.

Entrada de texto comum - TEXT

A forma mais simples de campo de entrada é a marcação text. Este campo permite a digitação de uma única palavra ou linha de texto, e possui uma largura default de 20 caracteres.

Opcões:

VALUE="" OPCIONAL

Utilizando a marcação VALUE você especifica que texto aparecerá no campo quando o formulário for exibido.

SIZE="" OPCIONAL

Esta marcação altera o tamanho deste campo exibido na tela.

Obs.: o usuário sempre poderá digitar mais caracteres do que o tamanho do campo na tela, pois o texto irá se deslocar a esquerda dentro do campo.

MAXLENGTH="" OPCIONAL

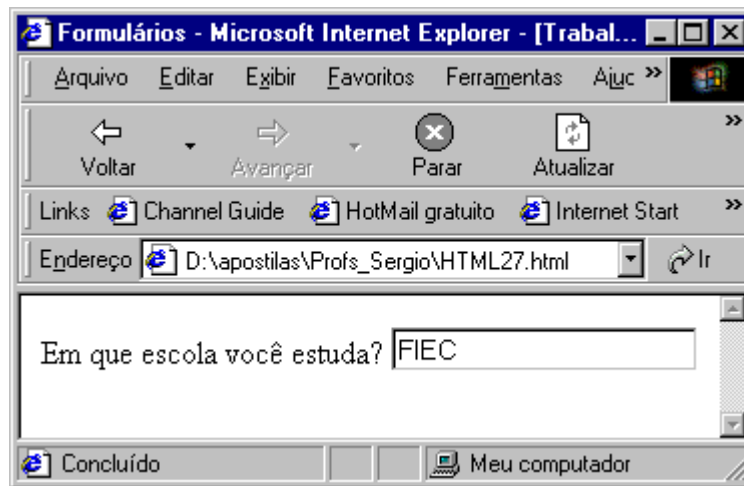
Se você deseja limitar o número de caracteres que o usuário pode digitar, basta usar esta marcação. O formulário irá emitir um bip de erro se o usuário tentar digitar além do permitido em MAXLENGTH.

```
<FORM METHOD="POST" ACTION="/cgi-bin/post-query">
```

Em que escola você estuda?

```
<INPUT TYPE="text" NAME="escola_nome" VALUE="FIEC" SIZE="20"
MAXLENGTH="25">
```

```
</FORM>
```



Obs.: O valor informado em NAME deve utilizar sublinhado em vez de espaços em branco. Inserir espaços em branco nestes nomes pode causar problemas no servidor na decodificação dos valores informados.

Se seu formulário possui apenas um campo de entrada textual, ao teclar ENTER neste campo o formulário será submetido, como se o usuário tivesse acionado o botão SUBMIT.

Entrada de texto protegido, senha - PASSWORD

Marcações de entrada do tipo password são idênticas aos campos do tipo text, exceto pelo fato de todos os caracteres serem exibidos como asteriscos (*).

Opcões:**VALUE="" OPCIONAL**

A marcação VALUE especifica um valor default para este campo.

SIZE="" OPCIONAL

Esta troca o tamanho do campo de password exibido na tela.

MAXLENGTH="" OPCIONAL

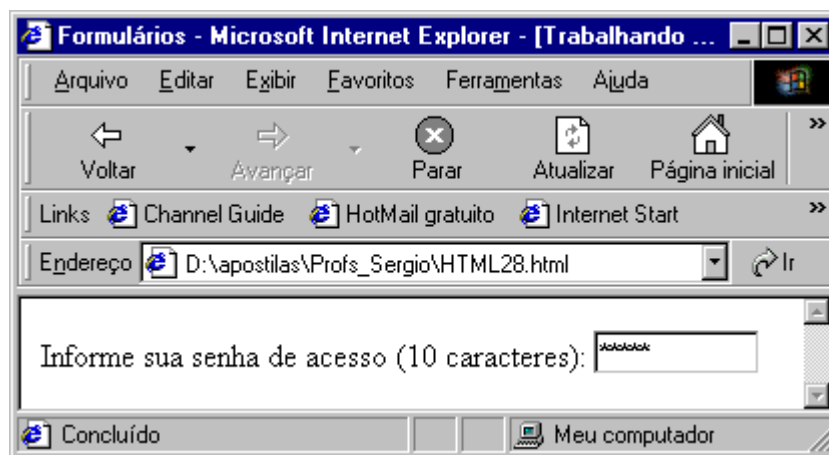
Limita o número de caracteres que o usuário pode informar como password.

```
<FORM METHOD="POST" ACTION="/cgi-bin/post-query">
```

Informe sua senha de acesso (10 caracteres):

```
<INPUT TYPE="password" VALUE="profs" NAME="sua_senha" SIZE="10"  
MAXLENGTH="10">
```

```
</FORM>
```



Obs.: Se você pretende utilizar MAXLENGTH para limitar o número de caracteres informados, tenha certeza de especificar um SIZE com o mesmo tamanho, para fornecer ao usuário o sentimento de qual a largura do campo. Esta regra não foi utilizada no primeiro exemplo, que possui SIZE igual a 20, mas MAXLENGTH de 25, o que faz o campo se deslocar nos últimos 5 caracteres caso o valor informado possua mais do que 20 caracteres.

Mesmo que o valor default VALUE esteja representado por asteriscos, o usuário pode visualizar o seu valor através da recuperação do fonte do hiperdocumento em HTML (uma opção comum na maioria dos clientes WWW).

Entrada oculta - HIDDEN

Além da marcação PASSWORD com um valor VALUE default, é possível "esconder" informação passada dentro de blocos FORM com uma marcação HIDDEN. Esta informação é recebida pelo script de decodificação no servidor, mas não é diretamente exibida ao usuário no formulário.

Opções:**VALUE="" OBRIGATÓRIO**

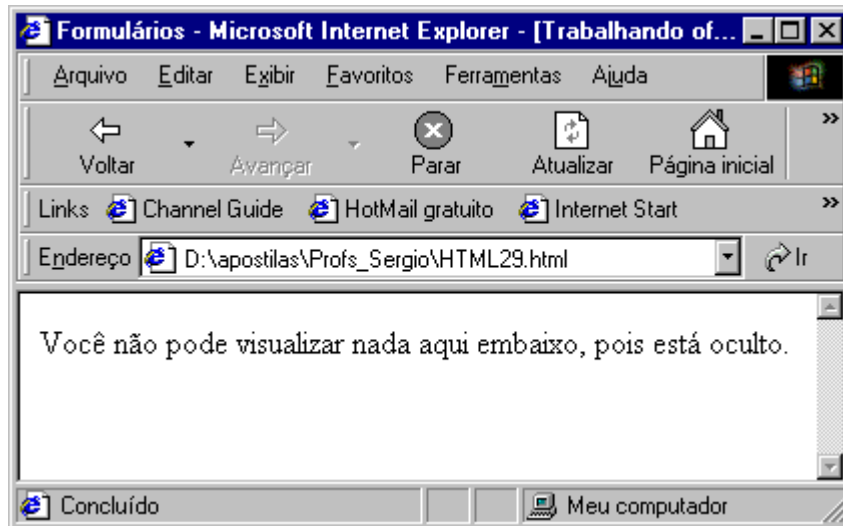
Através da marcação VALUE, você deve especificar o texto oculto a ser enviado ao script processador do formulário.

```
<FORM METHOD="POST" ACTION="/cgi-bin/post-query">
```

Você não pode visualizar nada aqui embaixo, pois está oculto.

```
<INPUT TYPE="hidden" NAME="nome_oculto" VALUE="FIEC">
```

```
</FORM>
```



Obs.: Mesmo que o valor especificado em HIDDEN VALUE não seja exibido no formulário, o usuário pode vê-lo através da operação de visualização do fonte do hiperdocumento, da mesma forma como no campo PASSWORD.

Entrada de várias linhas de texto - TEXTAREA

A marcação TEXTAREA não utiliza o formato convencional INPUT TYPE="text" dos exemplos anteriores. Ao contrário, uma marcação <TEXTAREA> delimita o seu início e a marcação </TEXTAREA> o seu fim.

Opcões:

ROWS="" OBRIGATÓRIO

Especifica o número de linhas da entrada textual.

COLS="" OBRIGATÓRIO

Especifica o número de colunas da entrada textual.

Texto default OPCIONAL

Se você deseja que um texto seja exibido no campo textual ao abrir o formulário, simplesmente coloque este texto entre as marcações de início e fim da TEXTAREA.

```
<FORM METHOD="POST" ACTION="/cgi-bin/post-query">
```

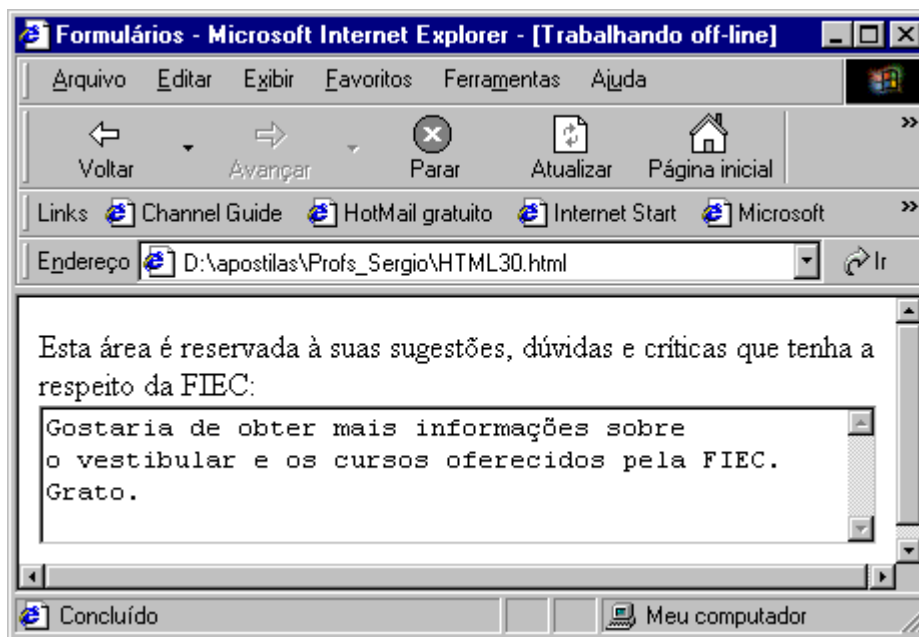
Esta área é reservada à suas sugestões, dúvidas e críticas que tenha a respeito da FIEC:


```
<TEXTAREA NAME="critica" ROWS="4" COLS="50">
```

Gostaria de obter mais informações sobre o vestibular e os cursos oferecidos pela FIEC. Grato.

```
</TEXTAREA>
```

```
</FORM>
```



Obs.: O usuário dispõe de Scrollbars para digitar além do número de linhas e colunas definidas em ROWS e COLS.

MENUS

Menus com opções - SELECT

Menu de opção única

A marcação SELECT segue a mesma convenção de TEXTAREA. Ou seja, as opções de menu ficam entre a marcação de início <SELECT> e a de fim </SELECT>.

Opções:

OPTION OBRIGATÓRIO

Especifica uma opção presente no menu.

VALUE="" OPCIONAL

Especifica o valor da opção retornada ao servidor. Se não for definido, o nome da opção é enviado ao servidor.

SELECTED OPCIONAL

Por default, a primeira OPTION é exibida no menu. Esta marcação estabelece uma opção de menu a ser exibida inicialmente, quando não se deseja que seja a primeira OPTION.

```
<FORM METHOD="POST" ACTION="/cgi-bin/post-query">
```

```
  Em que cidade você pretende estudar?
```

```
  <SELECT NAME="cidades_estudar">
```

```
    <OPTION>Campinas
```

```
    <OPTION SELECTED>Indaiatuba
```

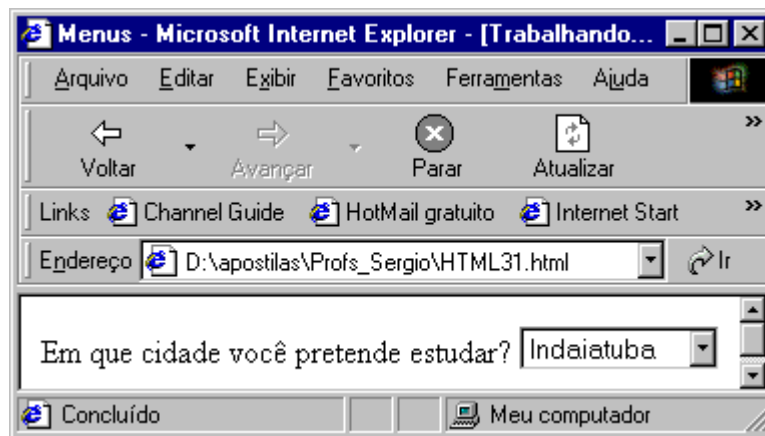
```
    <OPTION>Itu
```

```
    <OPTION VALUE="capital">São Paulo
```

```
    <OPTION VALUE="outra_cidade">Outra cidade
```

```
  </SELECT>
```

```
</FORM>
```



Obs.: Se você possui mais do que 3 ou 4 opções de escolha, e o usuário só pode selecionar uma, então este elemento de entrada é o melhor. As pessoas frequentemente utilizam vários RADIOBUTTONS, que veremos a seguir.

O script decodificador, no servidor, provavelmente apreciará uma única palavra para VALUE, em vez de múltiplas palavras. Você sempre pode utilizar sublinhado para separar expressões compostas.

Listas pagináveis - SELECT com SIZE

A única diferença entre este elemento de entrada e o anterior SELECT é a introdução da opção SIZE. Ela especifica quantas linhas com opções de menu serão exibidas na janela.

Opções:

MULTIPLE OPCIONAL

Especifica que múltiplas opções podem ser selecionadas, em oposição ao exemplo anterior do SELECT, onde apenas uma opção pode ser selecionada no menu.

SIZE="" OBRIGATÓRIO

Número de linhas (opções de menu) exibidas na janela.

OPTION OBRIGATÓRIO

Especifica uma opção da lista.

VALUE="" OPCIONAL

Especifica o valor da opção retornada ao servidor.

Se não for definido, o nome da opção é enviado ao servidor.

SELECTED OPCIONAL

Esta opção determina uma OPTION default para ser selecionada.

```
<FORM METHOD="POST" ACTION="/cgi-bin/post-query">
```

```
Que cursos você deseja fazer na FIEC?<BR>
```

```
<SELECT NAME="opcional" MULTIPLE SIZE="5">
```

```
<OPTION>Administração de Empresas
```

```
<OPTION>Alimentos
```

```
<OPTION VALUE="pd" SELECTED>Técnico em Informática
```

```
<OPTION>Eletrônica
```

```
<OPTION SELECTED>Edificações
```

```
<OPTION>Turismo
```

```
</SELECT>
```

```
</FORM>
```



Obs.: Este elemento de entrada é bom para listas com muitas opções, porque é possível controlar quantas serão exibidas por vez.

Em alguns clientes WWW, é necessário teclar simultaneamente as teclas CONTROL ou SHIFT para selecionar múltiplos itens.

BOTÕES

Botões sim ou não - CHECKBOX

Retornando ao formato de INPUT TYPE="", a marcação CHECKBOXES é perfeita para escolher entre duas opções.

Opções:

VALUE="" OPCIONAL

Especifica o valor da opção enviado ao servidor. Se não for definido, o valor "on" é enviado ao script decodificador.

CHECKED OPCIONAL

Esta marcação define a opção selecionada por default.

```
<FORM METHOD="POST" ACTION="/cgi-bin/post-query">
```

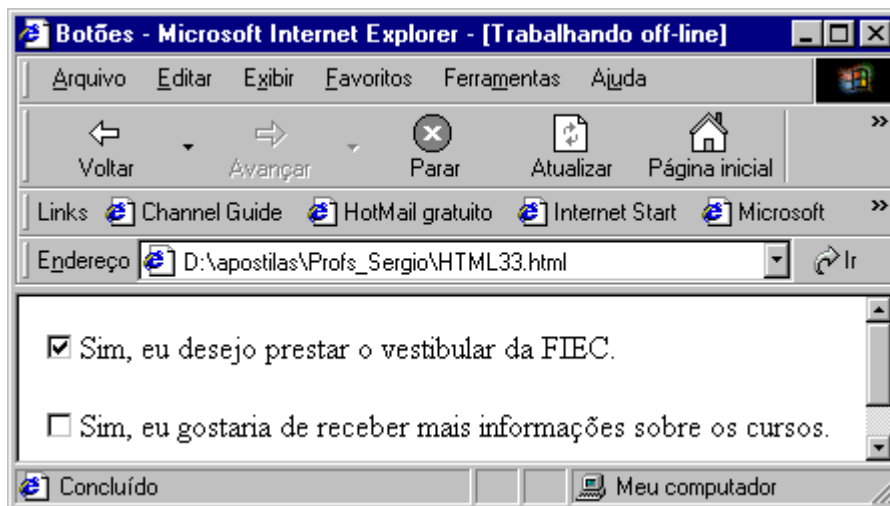
```
<INPUT TYPE="checkbox" NAME="boletim" CHECKED>Sim, eu desejo prestar o vestibular da FIEC.
```

```
<P>
```

```
<INPUT TYPE="checkbox" NAME="info">Sim, eu gostaria de receber mais informações sobre os cursos.
```

```
</DL>
```

```
</FORM>
```



Obs.: Este elemento funciona melhor para respostas do tipo "sim/não" ou "on/off".

Você pode simular uma lista SELECT MULTIPLE criando CHECKBOXES com o mesmo NAME, mas diferentes VALUES

Botões com opções - RADIO

RADIOBUTTONS utilizam uma notação similar a de CHECKBOXES, contudo, apenas uma opção pode ser escolhida.

Opcões:**VALUE="" Obrigatório**

Especifica o valor da opção a ser enviado para o servidor. Se não for definido, um valor "on" é enviado ao script decodificador.

CHECKED RECOMENDADO

Esta marcação especifica qual botão estará selecionado por default. Uma vez que uma seleção precisa ser feita, é melhor prover uma opção pré-selecionada.

```
<FORM METHOD="POST" ACTION="/cgi-bin/post-query">
```

Suas preferências:

```
<DL>
```

```
<DD>Qual a linguagem de programação que você mais gosta?
```

```
<INPUT TYPE="radio" NAME="linguagem" VALUE="de" CHECKED>Delphi
```

```
<INPUT TYPE="radio" NAME="linguagem" VALUE="vb">Visual Basic
```

```
<INPUT TYPE="radio" NAME="linguagem" VALUE="fp"> Visual FoxPro
```

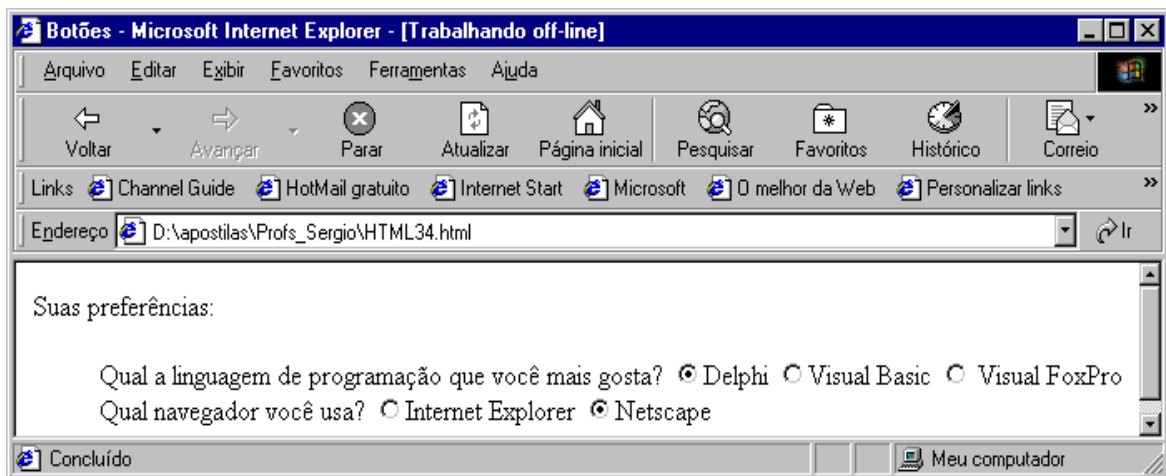
```
<DD>Qual navegador você usa?
```

```
<INPUT TYPE="radio" NAME="navegador" VALUE="ie">Internet Explorer
```

```
<INPUT TYPE="radio" NAME="navegador" VALUE="ns">Netscape
```

```
</DL>
```

```
</FORM>
```



Obs.: Como já foi mencionado, RADIOBUTTONS não são boa escolha para listas com muitos itens, porque o cliente tem problemas para exibir muitos botões. É melhor utilizar o menu SELECT.

Uma vez que um botão é selecionado, ele não pode ser desmarcado sem selecionar outro botão que possua o mesmo NAME. A seleção default pode ser restaurada com a utilização do botão RESET, introduzido a seguir.

Se você utiliza vários RADIOBUTTONS com o mesmo NAME e nenhum VALUES, o servidor não será capaz de dizer que botão foi selecionado, uma vez que o valor "on" será retornada para qualquer um deles.

Botões de submissão e limpeza - RESET E SUBMIT

Em vez do usuário corrigir cada INPUT, um botão RESET pode ser utilizado para restaurar todos os campos a seus valores default, como se nenhuma informação houvesse sido digitada.

E finalmente, o FORM precisa de uma opção para enviar toda a informação digitada para o servidor, uma vez que o usuário terminou de preencher todos os campos de entrada. O botão SUBMIT transfere toda a informação para a URL especificada no elemento ACTION.

Opcões:

VALUE="" OPCIONAL

Especifica o texto a ser exibido no botão. Se não for especificado, os textos default "Reset" e "Submit Query" serão colocados nos botões RESET e SUBMIT, respectivamente.

NAME="" OPCIONAL

Se NAME for definido em um botão SUBMIT, o formulário irá transmitir o valor do conteúdo do elemento VALUE, permitindo que você tenha múltiplos botões SUBMIT numa espécie de versão simplificada de um RADIOBUTTONS.

```
<FORM METHOD="POST" ACTION="/cgi-bin/donothing">
```

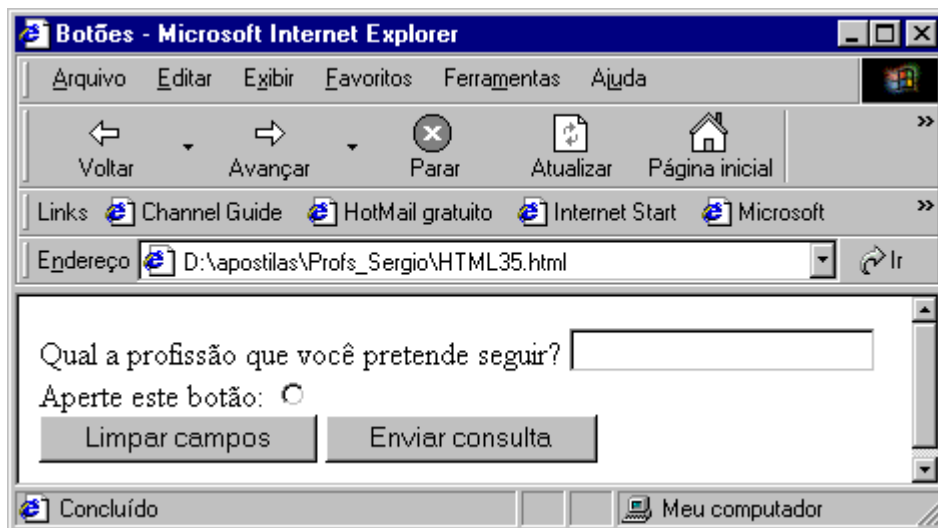
```
Qual a profissão que você pretende seguir? <INPUT TYPE="text"><BR>
```

```
Aperte este botão: <INPUT TYPE="radio" NAME="nada"><BR>
```

```
<INPUT TYPE="reset" VALUE="Limpar campos">
```

```
<INPUT TYPE="submit">
```

```
</FORM>
```



Obs.: Normalmente, o script decodificador retorna um hiperdocumento após ele processar a informação do formulário.

Nem todos os clientes suportam múltiplos botões SUBMIT. É melhor utilizar apenas um único botão SUBMIT para transmitir a informação digitada no formulário.

Em alguns navegadores, quando não se define o VALUE do botão SUBMIT ou do RESET, é colocado por default a tradução, como no exemplo acima.

Estrutura Básica - Frame Document

O "Frame Document" é um arquivo onde se define a estrutura das janelas para seu hiperdocumento em HTML. Quantas serão e qual sua distribuição em tela. Neste documento as marcações <body> e seu par </body> são substituídas por <frameset> e </frameset>. Tal arquivo pode ser, por exemplo, o arquivo index.html de qualquer nível na estrutura do seu serviço de informações baseado em WWW.

Pronto, agora é preciso definir cada frame/janela internamente à(s) área(s) de frameset. Ou seja, as características de cada janela e seus "conteúdos" - URL inicial. Cada janela/frame é antecedido da marcação <frame>, como numa lista cada item é antecedido por . Usualmente, uma URL virá associada a cada frame.

A estrutura mínima do *frame document* será então:

<pre><html> <head> <title></title> </head> <frameset ...> <frame src="URL"> <frame src="URL"> </frameset> </html></pre>	<p>Atenção: Note que com apenas essas marcações o documento da estrutura - Frame Document - não funciona. É necessário que esteja definido algum atributo para as áreas de frameset, assim como associadas urls a cada frame, como veremos mais adiante.</p> <p>Atenção II: Você poderá ter - e provavelmente terá - vários <frameset> intercalados. Da mesma forma como é possível intercalar listas, ou tabelas em HTML.</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Cada uma destas marcações - <frameset> e <frame> - aceita extensões, valores e atributos, como veremos a seguir.

Sintaxe

Frameset

Frameset aceita os atributos ROWS e COLS, referentes divisões horizontais (como linhas em uma tabela) e verticais (como colunas) entre janelas na tela. Internamente a marcações FRAMESET só poderão aparecer outras FRAMESET, FRAME ou NOFRAMES.

Importante: Não podem ser utilizadas as marcações válidas entre marcações <BODY></BODY> nem internamente a marcações FRAMESET, nem antes dela, senão FRAMESET será ignorada.

Atributos

1. ROWS (<frameset rows="valor, valor, valor...">)

Define divisões horizontais entre janelas. Vem sempre acompanhado de valores associado, que definem quanto da tela cada janela vai ocupar. Dessa forma, para cada janela a ser criada deverá haver um valor associado. Estes valores devem vir separados por vírgulas.

Este valor poderá ser:

- **Numérico – pixels** ⇒ (*ROWS="30, 50"*) Refere-se a quantos pixels cada frame (ou janela) deve ocupar. A desvantagem desta notação é que não é possível ter controle do valor total de pixels que o cliente do usuário compreende.
- **Percentual** ⇒ (*ROWS="25%, 25%, 50%"*) Pode ainda ser um valor percentual, sempre somando um valor de 100%. (É o método mais simples).
- **Relativo** ⇒ (*ROWS="2*, *"*) Desta forma, define-se um valor relativo. No exemplo, o primeiro frame vai ocupar dois terços da tela, e o segundo um terço.

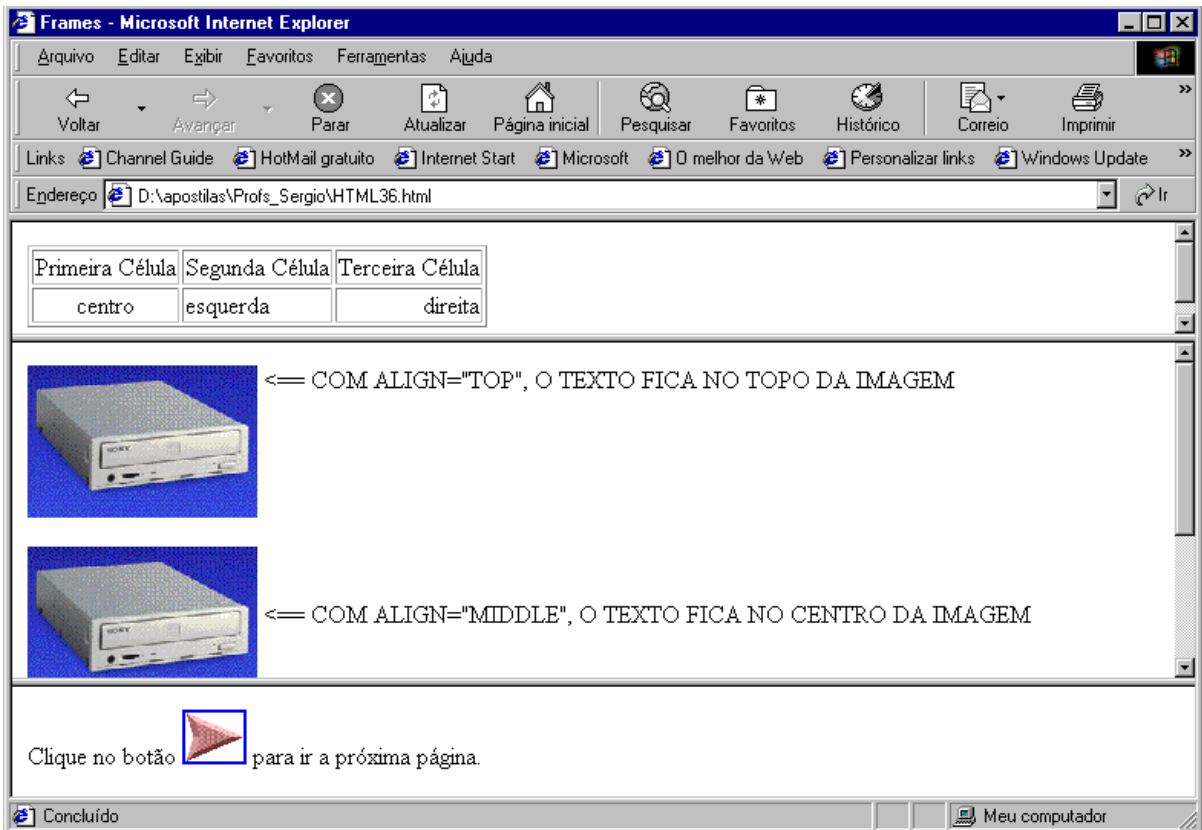
Exemplos :

1. Para dividir a tela do browser em três janelas horizontais, sendo que a do meio é mais larga que as de cima e de baixo:

```
<html>
<head>
<title>Frames</title>
</head>

<frameset rows="20%, 60%, 20%">
<frame src="html18.html">
<frame src="html13.html">
<frame src="html16.html">
</frameset>

</html>
```



2. Três janelas horizontais, sendo que a primeira e a segunda tem uma altura fixa, e o frame central deverá ocupar o restante do espaço:

```
<html>
```

```
<head>
```

```
<title>Frames</title>
```

```
</head>
```

```
<frameset rows="75, *, 60">
```

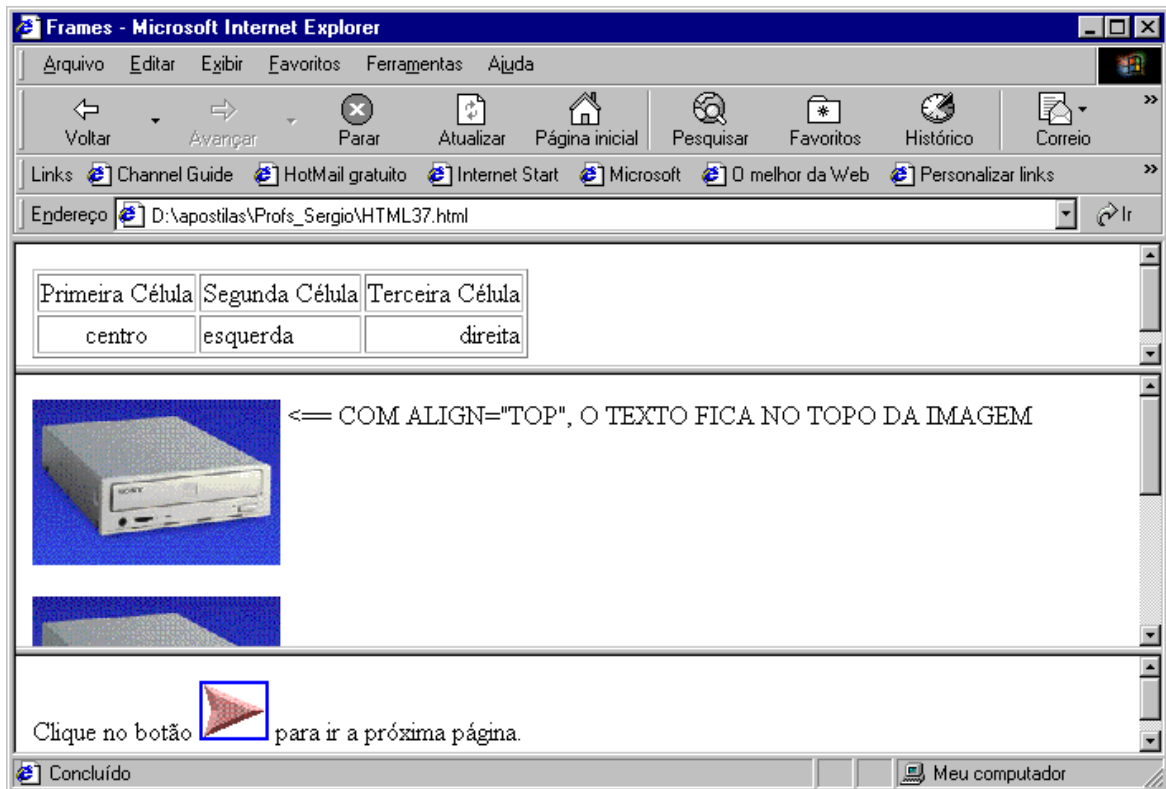
```
<frame src="html18.html">
```

```
<frame src="html13.html">
```

```
<frame src="html16.html">
```

```
</frameset>
```

```
</html>
```



2. COLS (<frameset cols="valor, valor, valor.. ">)

Funciona exatamente como a marcação anterior, no entanto, divide a tela em frames ou janelas verticais.

Exemplos

Para dividir a tela do browser em três janelas verticais:

1. Três colunas, sendo que a do meio é mais larga que as de cima e de baixo:

```
<html>
```

```
<head>
```

```
<title>Frames</title>
```

```
</head>
```

```
<frameset cols="25%, 50% ,25%">
```

```
<frame src="html4.html">
```

```
<frame src="html12.html">
```

```
<frame src="html5.html">
```

```
</frameset>
```

```
</html>
```



2. Três janelas verticais, sendo que a primeira e a segunda tem uma altura fixa, e o frame central deverá ocupar o restante do espaço:

```
<html>
```

```
<head>
```

```
<title>Frames</title>
```

```
</head>
```

```
<frameset cols="200, *, 300">
```

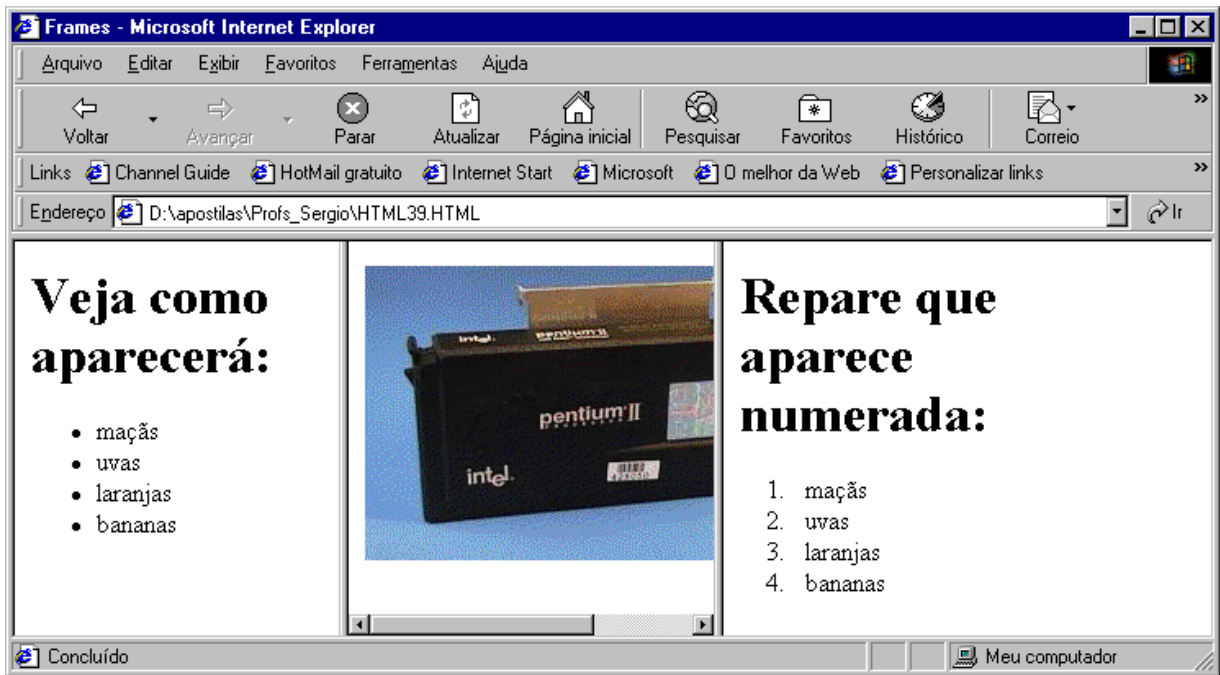
```
<frame src="html4.html">
```

```
<frame src="html12.html">
```

```
<frame src="html5.html">
```

```
</frameset>
```

```
</html>
```



Atenção, para intercalar janelas verticais e horizontais, será necessário definir várias áreas "FRAMESET". Para cada área delimitada como "FRAMESET" você poderá definir número de linhas OU colunas. Veremos como fazê-lo mais adiante. Não é possível definir COLS e ROWS para uma mesma área "FRAMESET".

Frame

A marcação FRAME define cada janela contida em uma área "frameset". Esta marcação não necessita de uma tag de finalização (</frame>) e aceita 6 atributos possíveis:

1. SRC="url"

O atributo SRC define a URL que será exibida em cada frame.

2. NAME="nome_da_janela"

Este atributo é utilizado para associar um nome a uma janela. Deve ser usado quando uma janela - frame - for o destino de um link em outro documento (normalmente é no mesmo documento. Um frame apontando para outro. Tipo um índice abrindo várias urls em outras janelas).

Este atributo é opcional. Por padrão, os frames não tem nome. Importante, o nome_da_janela deve começar por caracter alfanumérico.

3. MARGINWIDTH="valor"

Esse atributo controla as margens esquerda e direita de cada frame, ou seja, a distância entre o conteúdo da página e as margens da janela. O valor associado será um valor absoluto em pixels. O menor valor aceito será 1.

Este atributo é opcional, caso não venha definido, o browser usará o seu padrão para definir as margens no frame/janela.

4. MARGINHEIGHT="valor"

Também é um atributo opcional. Funciona exatamente como o anterior, só que determina as margens superior/inferior em cada frame.

5. SCROLLING="yes/no/auto"

Novamente, é um atributo opcional, que define se uma janela deve possuir barra de rolagem ou não. Caso seja definido como YES, a janela sempre possuirá uma barra de rolagem visível. Caso seja definido como NO, nunca haverá barra de rolagem. E, finalmente, se vier como AUTO, o browser aplicará a barra quando necessário.

O padrão é AUTO, portanto, se o atributo scrolling não vier definido, o browser aplicará a barra de rolagem, sempre que necessário.

6. NORESIZE

Este atributo não possui valor associado. Quando ele não aparece, o usuário poderá alterar o tamanho da janela, "arrastando" com o cursor a borda da mesma. O padrão é que todas as janelas possam ter seu tamanho alterado. NORESIZE é opcional

Um frame, com todos os atributos definidos seria escrito assim, por exemplo:

```
<frame src="http://www.cr-df.rnp.br" name="home" marginwidth="5"
marginheight="5" scrolling="auto" noresize>
```

Noframes

Esta marcação possibilita que se crie uma opção de navegação na página para quem não possui um browser que entende frames.

Esta marcação aparece no documento de estrutura "Frame document", e sempre é usada em pares (<noframes>.. conteúdo... </noframes>).

Quando o acesso for feito através de um browser que entenda frames, o que estiver entre as marcações "noframes" será simplesmente ignorado.

Internamente a uma área "noframes" deverá ser usada a estrutura padrão de documentos html (<head> </head>, <title> </title> <body> </body>, etc...).

Intercalando Frameset - exemplos

Para tornar mais claro como se constróem múltiplas e intercaladas janelas - divisões horizontais e verticais - utilizando das marcações de frames, seguem abaixo dois exemplos de construção de documentos com múltiplas janelas.

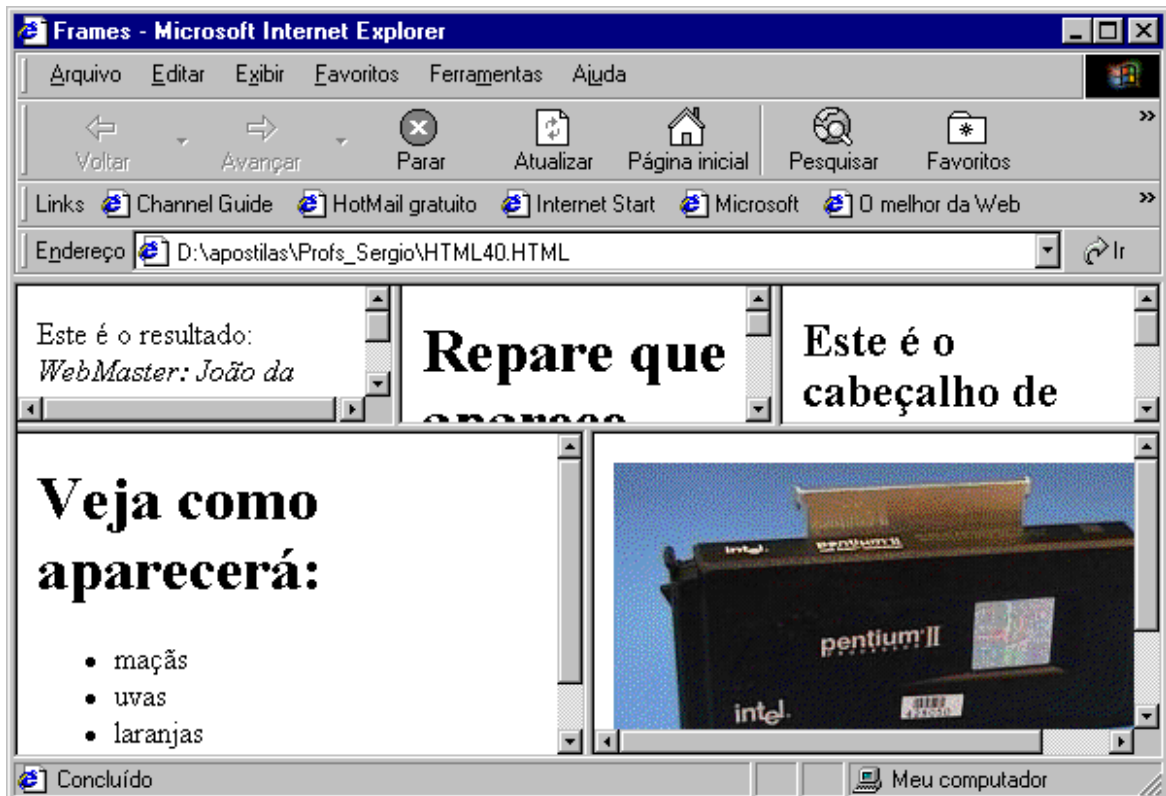
Exemplo 1

```
<html>
<head>
<title>Frames</title>
</head>
<frameset rows="30%, 70%">
<frameset cols="33%, 33%, 33%">
<frame src="html10.html">
<frame src="html5.html">
```

```
<frame src="html2.html">
</frameset>
```

```
<frameset cols="50%, 50%">
<frame src="html4.html">
<frame src="html12.html">
</frameset>
</frameset>
```

```
</html>
```



Exemplo 2:

```
<html>
```

```
<head>
```

```
<title>Frames</title>
```

```
</head>
```

```
<frameset rows="15%, 85%">
```

```
<frame src="html10.html">
```

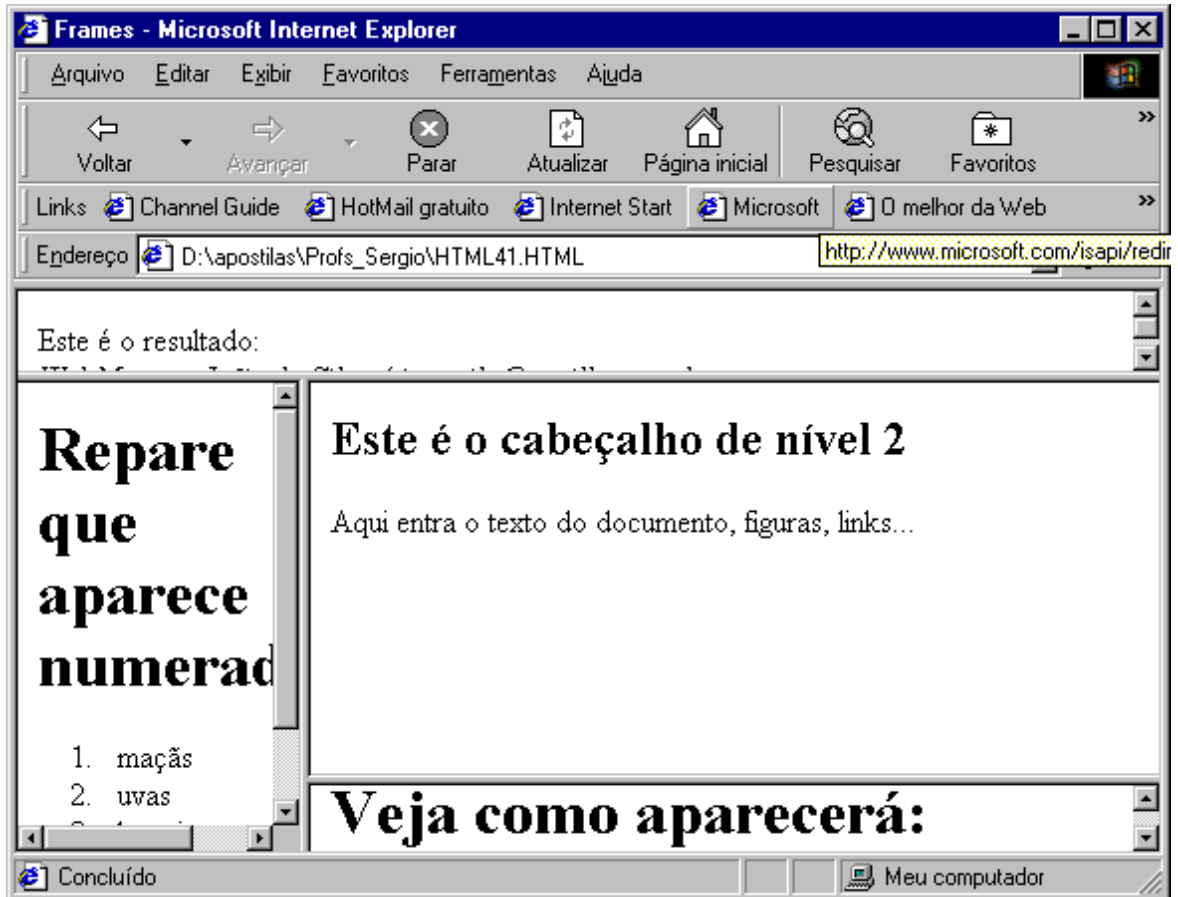
```
<frameset cols="25%, 75%">
```

```
<frame src="html5.html">
```

```
<frameset rows="85%, 15%">
```

```
<frame src="html2.html">
```

```
<frame src="html4.html">
</frameset>
</frameset>
</frameset>
</html>
```



Links entre frames

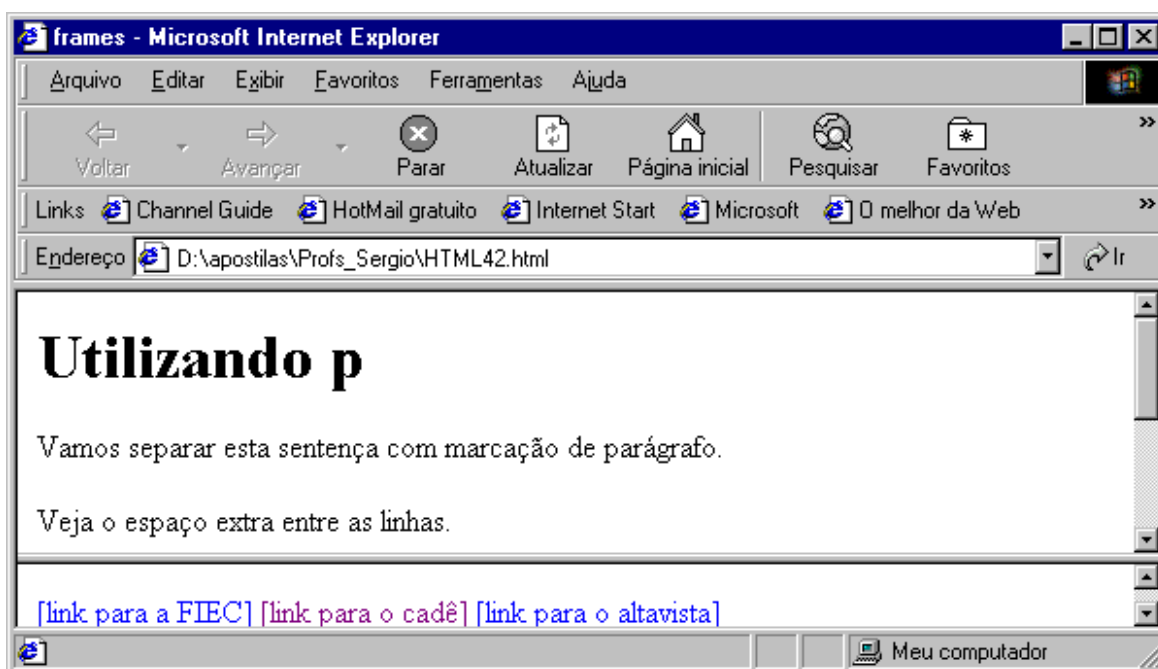
A marcação TARGET, permite que se controle em qual janela um link específico será exibido quando o usuário clicar sobre ele. Por exemplo, pode-se ter uma janela lateral com uma espécie de índice do serviço de informação e outra janela em que a navegação propriamente dita vai ocorrer. Desta forma, o índice está permanentemente disponível durante toda a consulta.

Para utilizar este recurso você deverá:

1. Atribuir um "name" a cada frame em seu "frame document" (name="valor").
2. No documento onde será criado um link, que vai aparecer em outra janela, ao criar o link, acrescentar a marcação target="valor" à âncora, da seguinte forma:

Onde este valor é idêntico àquele associado à marcação name no frame document.

Exemplo:



O FRAME DOCUMENT DEVERÁ SER ESCRITO ASSIM:	O DOCUMENTO QUE CONTÉM O ÍNDICE DE NAVEGAÇÃO (LINKS.HTML), SERÁ ASSIM:
<pre><html> <head> <title>frames</title></head> <frameset rows="80%, 20%"> <frame name="navegar" src="html3.html"> <frame src="links.html"> </frameset> </html></pre>	<pre><html> <head><title>Frames</title> </head> <body> [link para a FIEC] [link para o cadê] [link para o altavista] </body> </html></pre>

Cartela de Comandos

Marcação	Função / características
Estrutura Básica	
<html> e </html>	Delimita o "frame document", como em qualquer arquivo html normal.
<head></head>	Define cabeça do documento, novamente, como em qualquer arquivo html.
<frameset></frameset>	Delimita as janelas a serem usadas no documento.

	Podem haver vários frameset intercalados. Controla o comportamento das janelas através de seus atributos.
<code><frame src="URL"></code>	Define cada janela/frame. Normalmente possui uma URL associada. Aceita atributos e valores.
Sintaxe	
Para Frameset	
<code><frameset rows="valor, valor, valor"></code>	Define quantas linhas - janelas horizontais - e sua ocupação na tela, para cada área frameset. Aparecerão tantos valores quantas forem as janelas. Os valores relativos a cada janela deverão ser separados por vírgulas. Cada valor associado poderá ser: Numérico - em pixels = <code><frameset rows="30, 50"></code> Percentual = <code><frameset rows="25%, 25%, 50%"></code> Relativo = <code><"2*,*"></code> ;
<code><frameset cols="valor, valor, valor"></code>	Define quantas colunas - janelas verticais - e sua ocupação da tela. Funciona exatamente como a marcação interior (associação de valores, com as mesmas opções).
Para Frame	
<code><frame SRC="URL"></code>	associa uma url a uma janela
<code><frame NAME="valor"></code>	associa um nome a uma janela
<code><frame MARGINWIDTH="valor"></code>	determina margem direita/esquerda em cada frame
<code><frame MARGINHEIGHT="valor"></code>	determina margem superior/inferior em cada frame
<code><frame SCROLLING="yes/no/auto"></code>	define presença, ausência, ou atribuição automática (pelo browser) de barras de rolagem
<code><frame NORESIZE ></code>	mantém fixo o tamanho de cada janela/frame
Para Noframes	
<code><noframe></noframe></code>	delimita área para navegação opcional, destinada a quem usa browser que não entenda 'frame'. Internamente, recebe a marcação <code><body></body></code> delimitando o documento, e todas as demais marcações HTML.
Target	
<code></code>	Define a janela em que será exibido um link ao ser ativado. O nome associado deverá ser idêntico ao nome atribuído a janela de destino no "frame document".

Folhas de Estilo ou *Style Sheets*

Introdução

Se você se interessa pela criação de home pages já deve ter ouvido falar em *Cascading Style Sheets*, ou *CSS* apenas. Trata-se de um padrão de formatação para páginas web que vai além das limitações impostas pelo html. Proposto pelo W3 Consortium <<http://www.w3.org>> - uma espécie de comitê que define os padrões de programação para a WWW - o *CSS* foi introduzido pela primeira vez pela Microsoft, no lançamento do Internet Explorer 3.0.

O *Cascading Style Sheet* permite uma versatilidade maior na programação do layout de páginas web sem aumentar o seu tamanho em Kb, pois oferece várias possibilidades que antes só eram conseguidas com a utilização de gifs e jpbs. Basicamente, o *CSS* permite ao designer um controle maior sobre os atributos tipográficos de uma home page, como tamanho e cor das fontes, espaçamento entre linhas e caracteres, margem do texto, entre outros. Introduziu também às páginas a utilização de *layers*, permitindo a sobreposição de texto sobre texto ou texto sobre figuras.

Para visualizar inovações de layout programadas em *Cascading Style Sheet* é necessário um Internet Explorer versão 3.0 ou superior. O Netscape incorporou o *CSS* em seu código a partir da versão 4.0. Se você ainda não possui um browser 4.0, pode atualizá-lo na nossa seção de download.

O que *Style Sheets* podem fazer

Utilizando uma boa combinação de *tags* do código de programação html, como , <I>, <H1> e , podemos criar páginas interessantes sob o ponto de vista do design. Entretanto, as possibilidades que elas oferecem são bastante limitadas se comparadas às oferecidas pelo *CSS*.

Comandos de *Style Sheets* podem ser aplicados a **toda e qualquer tag**, modificando seus atributos. Até mesmo a *tag* <P> possui um </P> opcional e permite que você defina os atributos de tudo o que for compreendido entre as duas.

Você pode, por exemplo, trocar os atributos de uma *tag* , que tradicionalmente adiciona negrito a uma palavra.

O *Style Sheets* permite a você, por exemplo, configurar todos os comandos em uma página ou em um site inteiro de uma só vez.

Adicionando Estilo a Sua Página

Style Sheets podem ser utilizados de três maneiras diferentes: local (modificando uma *tag* específica de uma página), geral (modificando determinados atributos para a toda a página) ou global (quando criamos um modelo que será aplicado a várias páginas simultaneamente).

• Adicionando estilo localmente:

Style Sheet pode modificar os atributos de uma única *tag* específica, em um determinado ponto de uma página.

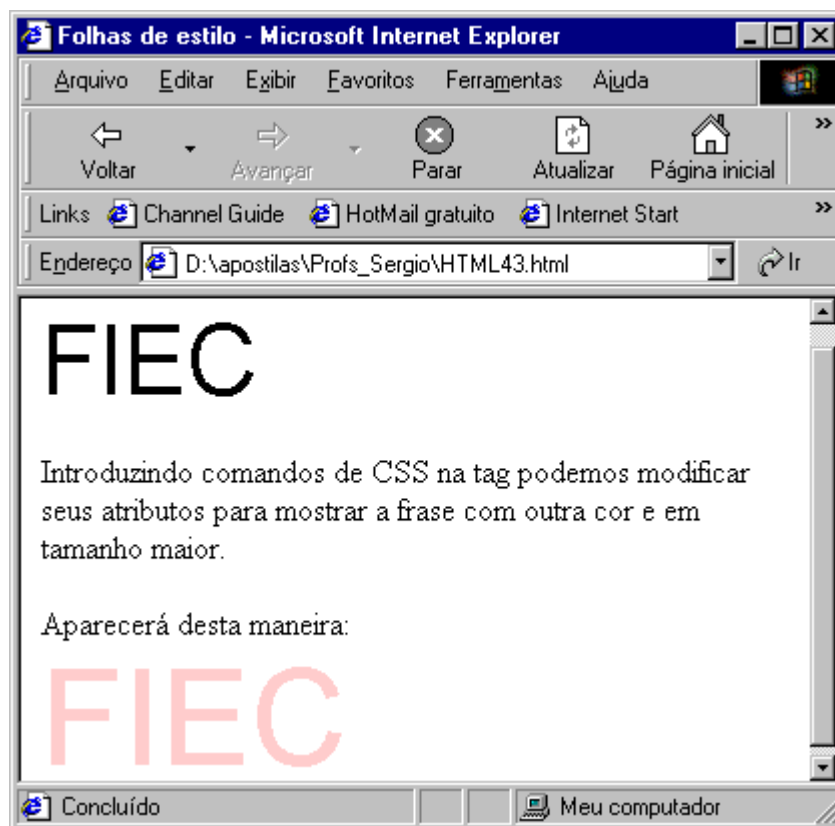
Os comandos em *CSS* aplicados localmente seguem a seguinte sintaxe:

```
<tag STYLE="propriedade:valor; propriedade, valor;"...>
```

Pelo código html normal, o tamanho máximo de uma fonte que podemos obter é estipulado pela tag . Utilizando o CSS podemos aumentar o tamanho com que as letras seriam tradicionalmente mostradas.

```
<HTML>
<HEAD>
<TITLE>Folhas de estilo</TITLE>
</HEAD>
<BODY>
<FONT FACE="Trebuchet MS, Arial, Helvetica" SIZE="+7">FIEC</FONT>
<P>Introduzindo comandos de CSS na tag podemos modificar seus atributos para mostrar
a frase com outra cor e em tamanho maior.
<P>Aparecerá desta maneira:<P>
<FONT style="font-size:50pt; color:pink; line-height:30pt; font-family:Trebuchet MS,
Arial, Helvetica;">FIEC</FONT>
</BODY>
</HTML>
```

Veja o resultado:



Com o CSS não há limites para o tamanho da fonte. Ele pode ser definido em pontos, pixels e outras unidades. Veremos isso em detalhes mais adiante.

- **Adicionando estilo geral:**

Para criar um modelo padrão de CSS que será aplicado a toda uma página, introduzimos o comando STYLE no cabeçalho do documento, entre a tag HTML e a tag

BODY. Dessa forma, tudo o que vier no conteúdo da página obedecerá aos comandos CSS estipulados uma única vez. Veja a seguir:

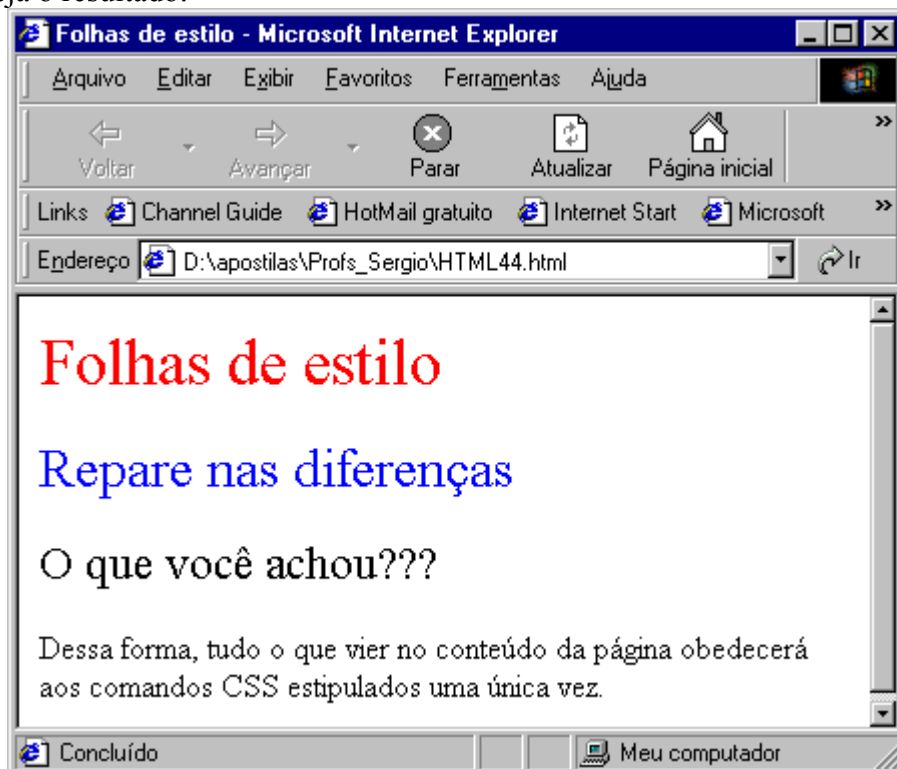
```
<HTML>
<TITLE>Folhas de estilo</TITLE>
<HEAD>
<STYLE type="text/css">

<!--
h1 {font: 25pt "Trebuchet MS, Arial, Helvetica"; color: red}
h2 {font: 20pt "Trebuchet MS, Arial, Helvetica"; color: blue}
p {font: 17pt "Corrier, Times"; color: black}
-->
</STYLE>
</HEAD>
<BODY>
<H1>Folhas de estilo</H1>
<H2>Repare nas diferenças</H2>
<P> O que você achou??? </P>
```

Dessa forma, tudo o que vier no conteúdo da página obedecerá aos comandos CSS estipulados uma única vez.

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

Veja o resultado:



Note que o que está dentro da *tag* STYLE aparece em comentário <!-- -->. Isso serve para evitar problemas com *browsers* que não suportam CSS. Estando entre

comentários, os atributos funcionarão normalmente em browsers com capacidade para CSS e serão desprezados por *browsers* mais antigos.

Classes

Podemos também aplicar classes nas especificações gerais de CSS para definir diferentes atributos para uma mesma tag. Se por exemplo você quiser um parágrafo em vermelho para grifar textos importantes, outro em cinza para textos normais, deverá proceder desta maneira:

- **Especificação geral:**

```
<!--
```

```
P.importante {font: 20pt "Trebuchet MS, Arial, Helvetica"; color: red}
```

```
P.normal {font: 15pt "Trebuchet MS, Arial, Helvetica"; color: gray}
```

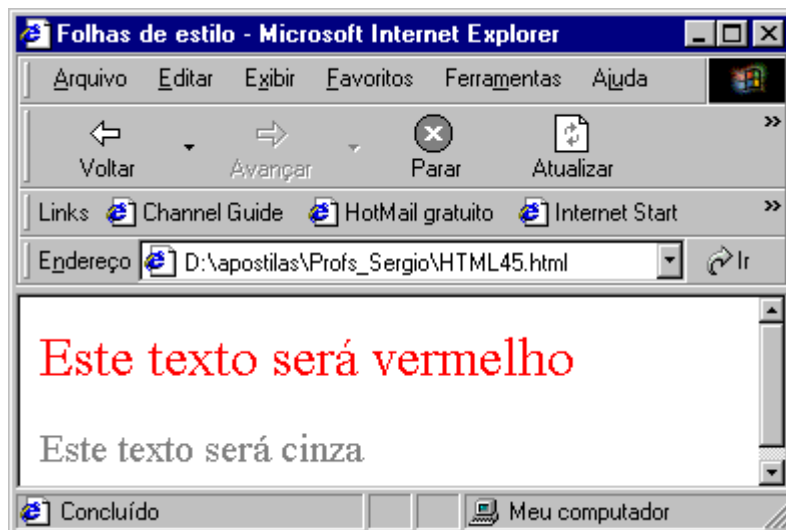
```
-->
```

- **No HTML os Códigos parecerão assim:**

```
<P CLASS="importante"> Este texto será vermelho</P>
```

```
<P CLASS="normal">Este texto será cinza</P>
```

Veja como ficará:



- **Adicionando estilo global:**

Podemos criar um único modelo de *Style Sheets* para ser aplicado a múltiplas páginas. Para tanto, devemos construir um modelo separado de *Style Sheets* e salvá-lo em um arquivo de terminação **.css**. Esse modelo pode ser aplicado a qualquer página, apenas referindo-se ao arquivo **.css** no seu cabeçalho.

A seguir um típico modelo global de *CSS*, definindo margens, cores, fontes e outros atributos:

```
<HTML>
```

```
<style type="text/css">
```

```
<--
```

```
body {font: 10pt "Verdana, sans-serif"};
```

```
h2 {font: 50pt "Verdana, sans-serif"; font-weight: bold; color: #58F734}
```

```
h3 {font: 13pt/15pt "Verdana, sans-serif"; font-weight: bold; color: maroon; margin-left:
```

```

0.5in; margin-top: -10px; line-height: 30px}
em {margin-top: -105px}
p {font: 10pt/12pt "Verdana, sans-serif"; color: black; margin-left: 0.5in; margin-top: -
10px; line-height: 20px}
-->
<BODY></BODY>
</HTML>

```

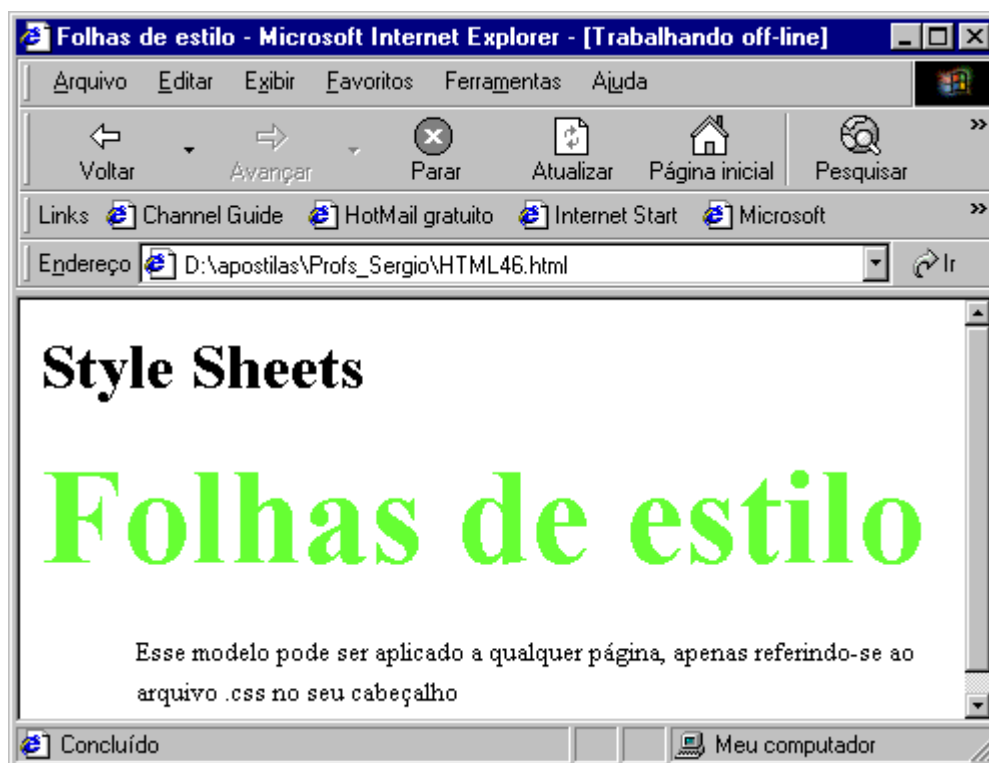
Para aplicar o modelo a uma página html específica, basta chamar o arquivo .css no seu cabeçalho:

```

<HEAD>
< link rel=stylesheet href="exemplo.css" type="text/css">
</HEAD>

```

Veja o resultado:



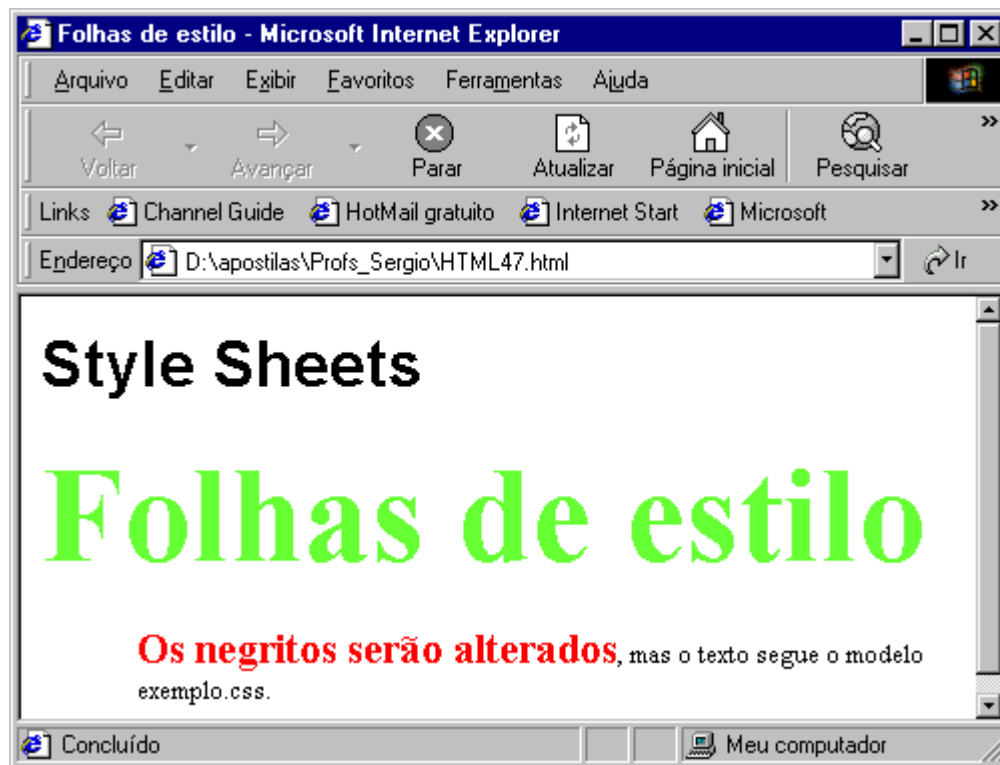
Outra maneira de aplicar o modelo CSS é importá-lo. A diferença entre linká-lo a página como no exemplo acima e importá-lo é que a segunda maneira permite acrescentar estilos na própria página que está editando, sem precisar alterar o modelo global (o arquivo .css). Veja o exemplo a seguir:

```

<HTML>
<STYLE TYPE="text/css">
<!--
@import url(exemplo.css);
B { color:red; font-size:16pt }
H1 { font-family: Arial, Helvetica }

```

```
</STYLE>
<HEAD>
<TITLE>Folhas de estilo</TITLE>
</HEAD>
<BODY>
<H1>Style Sheets</H1>
<H2>Folhas de estilo</H2>
<P>
<B>Os negritos serão alterados</B>, mas o texto segue o modelo exemplo.css.
</P>
</BODY>
</HTML>
```



IMPORTANTE: No caso de uma mesma página ter mais de um estilo aplicado, para saber qual predominará deve-se pensar da seguinte forma: o estilo mais específico prevalece.

No caso, se uma página contém uma especificação de estilo geral ou global, você pode acrescentar novos atributos locais diferentes. Dessa forma:

- Especificações locais prevalecem sobre
- Especificações gerais que prevalecem sobre
- Especificações globais que, por sua vez, prevalecem sobre
- Especificações default dos comandos de html

ARRUMANDO A PARTIR DAKI

Aplicando CSS a fontes

Comandos:

- font-family
- font-size
- font-weight
- font-style
- text-transform
- text-decoration

Font-family é o comando CSS utilizada para indicar qual a fonte que uma tag ou documento irá mostrar. Em uma aplicação geral de CSS a uma página, a linha de comando parecerá com esta:

```
P { font-family: Trebuchet MS, Arial, Helvetica }
```

No exemplo acima, tudo o que estiver entre as tag <P> e </P> no seu documento html será visualizado em Trebuchet MS. A razão de definir mais de uma fonte deve-se ao fato de que nem todos os computadores possuem as mesmas fontes instaladas. No caso, se o computador não possuir a Trebuchet MS instalada em seu hard disk, automaticamente passará para a segunda opção e mostrará o texto em Arial.

Deve-se observar que o nome da fonte especificadas na linha de comando do CSS deve ser exatamente igual ao nome da mesma no diretório "fontes" do computador. A fonte "Courier", por exemplo, leva este nome na plataforma Mac, mas se chama "Courier New" no Windows.

Font-size é utilizado para definir o tamanho pelo qual as fontes serão apresentadas na tela. Há três formas básicas para definição de tamanho:

1. pontos, pixels, cm e outras unidades de medida;
2. palavras-chave;
3. porcentagem.

Unidades de medida:

Ponto é uma medida familiar para designers, e pode ser usada para definir o tamanho de um elemento em uma página html. Tem sido a medida mais utilizada na programação em CSS pois funcionam bem em todos os browsers e plataformas operacionais. A única observação sobre esta medida é que as fontes costumam aparecer um pouco maiores na tela dos PCs do que nos monitores da linha Mac. Uma linha de comando geral em CSS parecerá assim:

```
P { font-size: 20pt } para pontos ou: B { font-size: 20px } para pixels, por exemplo.
```

As unidades de medida que podem ser utilizadas são:

- pt: pontos
- px: pixels
- in: inches
- cm: centímetros
- mm: milímetros
- pc: picas
- em: sem
- ex: x-weight

Pixel é uma medida familiar aos web designers. Pode dar uma boa noção do tamanho que a fonte aparecerá na tela, mas infelizmente a utilização dessa medida tem causado problemas para na impressão das páginas em papel.

Como já foi dito anteriormente, não há limites para a especificação do tamanho de uma fonte em CSS. Para quem estava acostumado a utilizar as tag que permitia apenas sete tamanhos para a visualização de uma palavra, a possibilidade de aplicar qualquer tamanho através do atributo font-size será bastante útil. Compare:

- **HTML tradicional: tag **

```
<HTML>
<HEAD>
<TITLE>Folhas de estilo</TITLE>
</HEAD>
<BODY>
<CENTER>
<H1>Veja a diferença com a tag FONT</H1>

<Font size= "1">Tamanho 1</Font> <Br>
<Font size= "2">Tamanho 2</Font> <Br>
<Font size= "3">Tamanho 3</Font> <Br>
<Font size= "4">Tamanho 4</Font> <Br>
<Font size= "5">Tamanho 5</Font> <Br>
<Font size= "6">Tamanho 6</Font> <Br>
<Font size= "7">Tamanho 7</Font>

</CENTER>

</BODY>
</HTML>
```



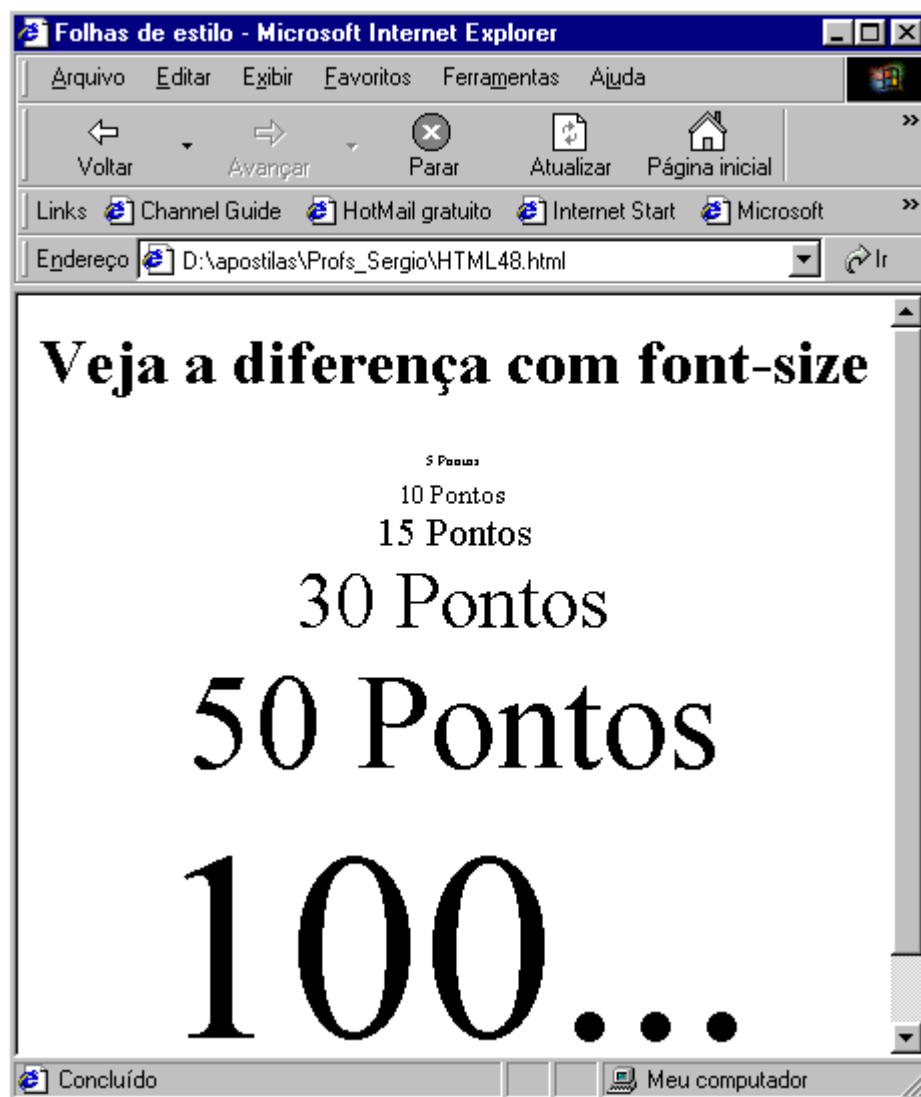
- **Comando Font-size:**

```
<HTML>
<HEAD>
<TITLE>Folhas de estilo</TITLE>
</HEAD>
<BODY>
<CENTER>
<H1>Veja a diferença com font-size</H1>
```

```
<A STYLE="font-size: 5pt">5 Pontos</A> <Br>
<A STYLE="font-size: 10pt">10 Pontos</A> <Br>
<A STYLE="font-size: 15pt">15 Pontos</A> <Br>
<A STYLE="font-size: 30pt">30 Pontos</A> <Br>
<A STYLE="font-size: 50pt">50 Pontos</A> <Br>
<A STYLE="font-size: 100pt">100...</A>
```

```
</CENTER>
```

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



Palavras-chave:

Outra maneira de determinar tamanhos de fonte em CSS é utilizar palavras chave. A seguir, a lista de comandos em ordem crescente:

- xx-small
- x-small
- small
- medium
- large
- x-large
- xx-large

Esses tamanho pré-configurados deixam o browser decidir qual o tamanho exato de fonte apropriado para cada palavra-chave. Por exemplo, x-large é mostrado em 28 pontos no Netscape 4.0, em 24 pontos no IE 4 e em 18 pontos no Internet Explorer 3 para Windows 95.

Porcentagem:

O tamanho das fontes também pode ser determinado segundo regras de porcentagem:

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

```
I { font-size: 200% }
```

No exemplo acima, todo o texto que estiver compreendido entre as tags <P> e </P> no corpo da página será mostrados em 12 pontos, assim como todas as palavras em itálico entre as duas tags serão apresentadas em um tamanho duas vezes maior.

font-weight é o comando CSS que controla a propriedade bold (negrito) de uma tag:

```
H1 { font-weight: bold }
```

No exemplo acima, todas as palavras da página que estiverem compreendidas entre as tags <H1> e </H1> serão mostradas em negrito. Da mesma forma você pode configurar "font-weight : normal" se quiser desativar todos os bolds de uma página. Podemos também especificar as características do negrito através de valores numéricos: 100, 200, 300...

400 é o valor normal de uma fonte sem negrito. 900 é o valor máximo de negrito disponível. Para uma fonte Arial, por exemplo, um browser com apenas a versão normal e a versão bold poderá mostrar a versão normal para valores de 100 à 600 e a versão bold para valores de 700 à 900.

Font-style é o comando CSS que controla a propriedade italic de uma tag.

```
H1 { font-style: italic }
```

No exemplo acima o browser irá procurar uma versão itálico da fonte para apresentar o texto compreendido entre as tags <H1> e </H1>. Algumas fontes tem uma versão chamada "oblique" em lugar de "italic". Nesse caso, use a palavra oblique como valor para font-style.

Text-transform serve para controlar os atributos maiúsculas e minúsculas de um texto:

```
P { text-transform: uppercase }
```

No exemplo, tudo o que estiver compreendido entre as tags <P> e </P> será mostrado em maiúsculas COMO NESTE EXEMPLO.

Os outros valores possíveis para o text transform são:

P { text-transform: lowercase }

Tudo entre as tags <P> e </P> será mostrado em maiúsculas como neste exemplo.

P { text-transform: capitalize }

Tudo entre as tags <P> e </P> será mostrado com as iniciais maiúsculas Como

Neste Exemplo.

P { text-transform: none }

Tudo entre as tags <P> e </P> será mostrado normalmente.

Observação: O Internet Explorer 3.0 não aceita o comando text-transform.

Text-decoration serve para controlar o sublinhado das palavras.

B { text-decoration: underline }

No exemplo acima todas as palavras em negrito da página aparecerão também sublinhadas. As opções são:

- underline: sublinha as palavras compreendidas e dentro de uma tag. Exemplo
- overline: coloca uma linha em cima da palavra compreendida entre as tags.

Exemplo

- line-trought: coloca uma linha em sobre o texto. Exemplo
- blink: sua palavra pisca (lembram da tag blink?). Exemplo
- none: tira o sublinhado de uma palavra (inclusive de um link). Exemplo

Observação: o atributo overline ainda não é mostrado no Netscape e o atributo blink não funciona em nenhuma das versões do Internet Explorer.

O mais interessante do comando text-decoration é que utilizando o atributo none finalmente podemos eliminar o sublinhado de um link! Se você inserir as linhas abaixo no cabeçalho de suas páginas, todos os sublinhados dos links desaparecerão. Mas não esqueça de configurá-los para serem de outra cor, ou ninguém saberá o que é e o que não é um link...

```
<style type="text/css">
<!--
A { text-decoration: none }
-->
</style>
```