

HTML

Hyper Text Mark-up Language

Prof. Jairo - jairo@uninove.br

Primeiro Semestre/2002

1.1 - INTRODUÇÃO: CONCEITOS BÁSICOS.....	3
1.1.1 - WWW	3
1.1.2 - INTERNET	3
1.1.3 - PROTOCOLO TCP/IP	3
1.1.4 - IP.....	3
1.1.5 - DNS	4
1.1.6 - HTTP	4
1.1.7 - PORTA	4
1.1.8 - SERVIDOR	4
1.1.9 - BROWSER	4
1.1.10 - HTML	5
1.1.11 - SITE	5
1.1.12 - FERRAMENTAS PARA CRIAR HTML.....	5
1.1.13 - HTTP/1.0 E HTTP/1.1	5
1.1.14 - BROWSERS QUE SUPORTAM HTTP/1.1	5
1.1.15 - URL	6
1.2 - A LINGUAGEM HTML	7
1.2.1 - DOCUMENTO HTML MÍNIMO.....	7
1.2.2 - DESCRIÇÃO DAS PRINCIPAIS TAGS	8
1.2.3 - TAGS MAIS USADAS NO INTERIOR DO CORPO DO DOCUMENTO HTML	8
<i>1.2.3.1 - Headings</i>	8
<i>1.2.3.2 - Paragraphs</i>	9
<i>1.2.3.3 - Lists</i>	9
<i>1.2.3.4 - Nested Lists</i>	10
<i>1.2.3.5 - Definition Lists</i>	11
<i>1.2.3.6 - Preformatted Text</i>	11
<i>1.2.3.7 - Plaintext</i>	11
<i>1.2.3.8 - Extended Quotations</i>	12
<i>1.2.3.9 - Forced Line Breaks</i>	12
<i>1.2.3.10 - Horizontal Rules</i>	12
<i>1.2.3.11 - Formatação de caracteres</i>	13
<i>1.2.3.12 - Estilo Físico de Caracteres</i>	13
<i>1.2.3.13 - Estilo lógico de caracteres</i>	14
<i>1.2.3.14 - Escape sequences</i>	14
<i>1.2.3.15 - Comentários HTML</i>	14
1.2.4 - TAGS MAIS USADAS NO INTERIOR DO HEAD DO DOCUMENTO HTML.....	14
<i>1.2.4.1 - Título do documento</i>	15
<i>1.2.4.2 - Script (roteiro)</i>	15
<i>1.2.4.3 - Meta-Informação</i>	15
1.3 - PROGRAMANDO EM HTML.....	17
1.3.1 - INSERÇÃO DE IMAGENS	17
1.3.2 - LINKS	17
1.3.3 - MAPA CLICÁVEL.....	18
1.3.4 - TABELAS	18
1.3.5 - MÚLTIPLAS JANELAS (FRAMES).....	20
1.3.6 - FORMULÁRIOS	21
1.4 - EXERCÍCIOS.....	23
1.5 - RESPOSTAS DOS EXERCÍCIOS	25

1 - HTML

1.1 - INTRODUÇÃO: CONCEITOS BÁSICOS

1.1.1 - WWW

World Wide Web é a "teia de alcance mundial" - "world wide" significa "grande como o mundo" e "Web" significa "teia". A "WWW" foi o mecanismo que permitiu a explosão da Internet porque possibilitou a construção de páginas gráficas, bonitas, coloridas, compostas com as fotos e animações que se podem ver hoje. Além disso, sua maior característica é a possibilidade de navegação através dos "links" (ligação). Pode-se dizer que a WWW é recente, pois surgiu no início dos anos 90 (1992).

1.1.2 - Internet

É a rede das redes. Nasceu após uma experiência militar para conexão de computadores em várias partes do mundo, germinou na experiência de conexão de computadores de diversas universidades e explodiu na conexão de computadores em qualquer lugar do mundo através de uma linha telefônica. A Internet só foi possível depois que inventaram um protocolo de fácil manipulação e que poderia trafegar em qualquer equipamento de informática, o TCP-IP.

A Internet teve seu início nos Estados Unidos em 1969 com o nome de ARPANET, que pertencia ao Depto. de Defesa Norte-Americano.

A partir de 1982 o uso da Arpanet tornou-se maior no âmbito acadêmico, o nome Internet veio depois que a rede se expandiu para outros países (inicialmente países europeus), e chegou ao Brasil em 1989.

Em 1987 foi liberado o uso comercial nos EUA, mas a Internet comercial somente chegou ao Brasil em 1995.

1.1.3 - Protocolo TCP/IP

Transmission Control Protocol/Internet Protocol, ou Protocolo de Controle de Transmissão ou Transferência/Protocolo da Internet, se refere ao conjunto de protocolos utilizados na Internet. Ele inclui uma série de padrões que especificam como os computadores e equipamentos de rede (hubs, switches, roteadores) vão se comunicar e cria convenções para interconectar redes e para o roteamento através dessas conexões.

Características:

- Estrutura cliente-servidor robusta: O TCP/IP é uma excelente plataforma cliente-servidor, especialmente em ambientes WAN (wide-area network, ou redes de grande alcance).
- Compartilhamento de informações: Milhares de organizações militares, educacionais, científicas e comerciais compartilham dados, correio eletrônico (e-mail) e outros serviços na Internet usando o TCP/IP.
- Ampla disponibilidade: Implementações do TCP/IP estão disponíveis em praticamente todos os Sistemas Operacionais populares. Seu código fonte é amplamente disponível em várias implementações.

O protocolo TCP/IP permite que máquinas conectadas numa mesma rede troquem informações entre si, que ocorre na forma de pacotes.

1.1.4 - IP

Existem algumas analogias entre computadores e telefones e o número IP é uma delas. Pode-se imaginar o número IP como um número de telefone com todos os códigos de discagem internacionais. Isto significa que qualquer máquina pode contactar outra máquina usando o número IP, bastando apenas que exista um caminho entre essas duas máquinas. Para acessar a Internet, toda máquina necessita ter um número IP diferente do número IP das outras máquinas que estejam acessando a Internet naquele mesmo instante. Isto também significa que duas máquinas na mesma rede NÃO podem ter o mesmo número IP, mas duas máquinas podem ter o mesmo IP desde que cada uma faça parte de um rede isolada. Essa isolamento pode muito bem ser um "firewall" (parede de fogo), que normalmente não impede o acesso dessas máquinas à Internet.

Ainda fazendo analogia com telefones, pode-se imaginar duas pessoas morando em países diferentes que possuam o mesmo número de telefone local, mas que usam prefixos diferentes para as ligações internacionais.

O número de IP tem 4 bytes de tamanho e tem um formato específico, xxx.xxx.xxx.xxx (exemplo: 143.107.130.1). Cada agrupamento xxx só pode ir de 0 a 255, pois essa é a capacidade de 1 byte.

1.1.5 - DNS

Domain Name System, ou servidor de nomes, serve para traduzir IPs em nomes de domínios e vice-versa. Por exemplo, quando se quer visitar o "endereço" 143.107.130.1 digita-se FGE.IF.USP.BR, pois é muito mais fácil decorar um nome que um número extenso como um IP, e o servidor DNS traduz esse nome para um IP. Graças ao DNS, os usuários tem o compromisso de saber apenas o nome do domínio que querem visitar e no "header" (cabeçalho) dos pacotes TCP/IP vão os IPs (de destino e de retorno).

No exemplo acima, o usuário digitou FGE.IF.USP.BR e abriu a página do Departamento de Física Geral da USP, isso porquê ele enviou pela Internet pacotes TCP/IP que carregavam nos headers o seu próprio IP (para retornarem a ele trazendo a página) e também o IP de destino, 143.107.130.1, que é o local de onde vem página.

Um domínio é composto por um nome e uma extensão, por exemplo USP.BR (IP: 143.107.xxx.xxx).

O DNS é fundamental para o funcionamento da Internet, e é controlado por uma entidade chamada Internic, localizada nos EUA. No Brasil, a autoridade sobre os domínios é a FAPESP (Fundação de Apoio à Pesquisa de São Paulo).

Para ter um domínio na Internet é necessário registrá-lo. Todos os domínios com extensão ".br" são registrados na FAPESP, normalmente através do preenchimento de um formulário de cadastro numa página de "www.registro.br".

1.1.6 - HTTP

Hyper Text Transfer Protocol ou protocolo de transferência de hipertexto é o protocolo utilizado pela WWW. Esse protocolo transmite textos, gráficos e qualquer outro tipo de arquivo além de permitir a navegação através de hipertexto (HTML). HTTP é o protocolo que tem impulsionado o crescimento rápido e recente da Internet, pois permite o funcionamento da face gráfica, muito mais bonita e amigável, além de sons e animações.

Atualmente o protocolo HTTP é o mais usado do conjunto de protocolos TCP/IP.

1.1.7 - Porta

Servidores HTTP geralmente estão na porta 80, por isto a porta 80 é chamada de porta da Internet, embora o mais correto seria chamá-la de porta da WWW. Exemplos de portas também muito usadas são: 21 (FTP, File Transfer Protocol), 23 (Telnet, conexão remota), 25 (SMTP, Servidor de email), 53 (DNS), 110 (POP3, para ler emails), 443 (HTTPS, ou HTTP criptografado, seguro).

1.1.8 - Servidor

É o computador equipado com software que permite "servir" a uma rede de computadores. Quanto mais potente o servidor e mais modernos seus programas, maior e melhor poderá ser a rede por ele atendida.

Exemplos de servidores WEB: Apache (Open Source - free, mais usado atualmente), Microsoft Internet Information Server, Netscape Enterprise Server.

1.1.9 - Browser

Para podermos visualizar as páginas (documentos) HTML é usual utilizar um software específico que interprete essa linguagem e apresente os objetos HTML (o conteúdo da página) na tela do computador do cliente (mais precisamente janela browser). É na interpretação do documento HTML que o browser inclui as figuras, mostra cores, links, sons, animações, etc.

Numa definição simples, web browser - ou simplesmente browser - é o Navegador, o software que permite explorar textos, fotos, gráficos, sons e vídeos na Internet. O browser permite pular de uma página para outra com um simples clique nos "links", que são palavras ou expressões grifadas ou qualquer objeto sobre o qual surgir uma mãozinha quando passa-se a seta do mouse sobre ela.

O browser foi desenvolvido na Universidade de Illinois (EUA), no começo dos anos 90, e o primeiro chamava-se Mosaic.

Exemplos de navegadores: Netscape, Internet Explorer (mais usado atualmente), Mozilla, Opera, Konqueror.

Um ponto importante a ser notado é que cada browser e cada versão de browser interpreta de maneira um pouco diferente o conteúdo do documento HTML, e que essas diferenças se acentuam à medida que aumenta a complexidade da página.

1.1.10 - HTML

"Hyper Text Mark-up Language", ou linguagem para marcação de hipertexto, é uma linguagem que usa indicações ou marcações no texto ASCII para disponibilizar na Internet imagens, sons ou animações (os documentos HTML são arquivos de texto puro, ou seja, não binários).

HTML é uma linguagem interpretada onde um programa especial (usualmente o browser) interpreta as marcações no texto (tags) e apresenta seus efeitos na tela para realçar, animar, estruturar, formatar e ilustrar o conteúdo do documento. Uma "tag" é um marcador de formatação de texto.

1.1.11 - Site

Um "lugar", ou seja, um computador ou uma rede de computadores conectada à Internet onde se tem acesso a informações usando um navegador ou outro instrumento de navegação. As páginas da Internet estão nos sites e têm como linguagem básica o HTML.

Numa linguagem simples, um portal é um site com conteúdo tal que atraia muitos clientes HTTP.

1.1.12 - Ferramentas para criar HTML

Como os documentos HTML são arquivos de texto puro, eles podem ser criados com qualquer editor de textos disponível, tais como o Bloco de Notas e o Wordpad (Windows), Edit (DOS), VI, Kedit, Kwrite e Emacs (Linux/Unix), etc. Para que o browser reconheça esses arquivos como documentos HTML eles devem ser salvos com extensões .html ou .htm.

Existem também vários editores HTML WYSIWYG (What You See Is What You Get, o que você vê é o que você consegue) onde a página pode ser criada simplesmente selecionando com o mouse objetos de um menu, e à medida em que é criado o documento já aparece na forma aproximada com que será visualizado pelos navegadores. Porém, neste caso também é recomendável que se conheça um pouco da linguagem HTML antes de usá-los, pois os editores WYSIWYG carregam os documentos com "sujeira". Os editores WYSIWYG são mais apropriados para webdesigners, não para programadores HTML.

Exemplos: FrontPage (Microsoft), DreamWeaver, Netscape Composer (free), Bluefish (free, Linux/Unix).

1.1.13 - HTTP/1.0 e HTTP/1.1

Logo após a criação do protocolo HTTP ele passou a dominar o tráfego de pacotes na Internet. Com o propósito de permitir uma WWW mais rápida e eficiente e remediar as fraquezas do protocolo original HTTP/1.0 foi criado o protocolo HTTP/1.1 com as principais características:

- A. Suporte a Hospedagem Virtual não IP, ou seja, múltiplas hospedagens de domínios virtuais num mesmo IP (isso é justificado pela carência e custo de IPs).
- B. Como o HTTP pertence ao conjunto de protocolos TCP/IP (pacotes TCP/IP são enviados e recebidos pela rede), o HTTP/1.1 evita algumas fraquezas do TCP/IP e aproveita algumas de suas vantagens. Por exemplo, quando um cliente acessa um documento, uma imagem, etc., HTTP/1.0 cria uma nova conexão TCP que é desfeita após realizada a tarefa, ou seja, fica criando novas conexões o tempo todo, que implica em muito tráfego TCP/IP e pouca informação transmitida. O HTTP/1.1 corrige isso de 2 maneiras:
 - i. Conexão persistente (persistent connections) permite ao cliente reutilizar a mesma conexão TCP enquanto estiver "conversando" com o mesmo servidor Web;
 - ii. Otimiza a informação transportada pela rede, evitando retornar e buscar outras informações que ficaram para trás. Como resultado temos menos "engarrafamentos de trânsito de pacotes TCP";
- C. Cache mais eficiente: permite buscar a informação que esteja mais próxima ao cliente, sem ter necessidade de retornar novamente a um servidor Web distante;
- D. Autenticação: senhas podem ser enviadas encriptadas, muito mais seguro que simplesmente texto (poderiam haver sniffers no caminho), favorecendo o comércio eletrônico.

1.1.14 - BROWSERS QUE SUPORTAM HTTP/1.1

Apenas as versões 3.0 ou maiores de browsers suportam HTTP/1.1. Por esse motivo, e a julgar pelo rumo que a WWW tem tomado (cada vez mais gráfica, dinâmica, interativa e animada), os browsers versão 2 ou menores parecem mesmo destinados à extinção.

1.1.15 - URL

Uniform Resource Locators especifica a localização dos arquivos nos servidores. A URL inclui o tipo do recurso que está sendo acessado, o endereço do servidor e a localização do arquivo.

A estrutura completa de uma URL é:

acesso://host.domain:porta/path/filename, onde

acesso é usualmente do tipo

file um arquivo no disco local;
ftp um arquivo num servidor de FTP anônimo;
http um arquivo da WWW

porta pode ser qualquer uma entre 1 e 65536 desde que esteja disponível pelo Sistema Operacional e que tenha um servidor Web instalado nela. Quando a porta for omitida será usada a porta 80 padrão, que é o "default" do browser e do servidor Web

host (caso haja) é o nome (hostname) dado ao computador na intranet que constitui o domínio "domain"

domain é o nome do domínio

path é caminho ou estrutura de diretório ou subdiretório. A barra "/" é invertida em relação ao DOS ("\\") pois é assim que funciona a estrutura de diretórios no Unix, onde iniciou a Internet

filename é o nome do documento. Se "filename" for omitido ou não existir, o servidor Web entregará (caso haja) uma página "default"

Exemplo:

<http://fge.if.usp.br/Welcome.html>

onde,

fge é o nome de um computador (host) da rede do Instituto de Física,
if.usp.br é o domínio (domain) e Welcome.html é o nome do documento HTML.
Neste caso foi omitida a porta, e por isso usada a 80.

1.2 - A LINGUAGEM HTML

Para entendermos a linguagem HTML precisamos ter em mente que os documentos HTML são arquivos de texto puro (ASCII) que contém "marcações" (tags HTML) no texto, e essas marcações consistem de um sinal "<" (bracket), o nome da tag e o sinal ">". As tags usualmente ocorrem em pares (ex: <H1> and </H1>) para expressarem corretamente o início e final da instrução, onde a tag final é igual à inicial, porém uma barra "/" precede o texto no interior dos brackets.

Na interpretação do documento HTML, o browser irá considerar as tags como elementos de inclusão dos objetos HTML.

Alguns desses elementos podem incluir um atributo, que é uma informação adicional a ser incluída no interior das tags. Por exemplo, pode-se especificar um alinhamento de imagem (top, middle, or bottom - topo, meio e baixo) adicionando-se o atributo apropriado na tag de inclusão de imagem IMG:

```
<IMG SRC="c:\imagens\carro.jpg" ALIGN="top">
```

Normalmente o conteúdo do atributo é posicionado entre aspas duplas, isso por uma questão de padronização, mas na prática nem seria necessário pois atualmente os browsers já são desenvolvidos tendo em vista a falta de padronização dos documentos HTML que compõem grande parte das páginas da WWW.

1.2.1 - DOCUMENTO HTML MÍNIMO

Todo documento HTML precisa conter certas tags padronizadas, no mínimo <HEAD>, </HEAD>, <BODY> e </BODY>, pois elas constituem as duas partes básicas de um documento HTML que são: o HEAD (cabeçalho) e o BODY (corpo do documento).

O <HEAD> contém, entre outras coisas, o <TITLE> (título), e o no <BODY> vai todo o conteúdo do documento, normalmente composto de parágrafos, listas, tabelas, etc.

Para interpretar corretamente o conteúdo do documento HTML, os browsers esperam informações em acordo com as especificações HTML.

Exemplo de documento HTML simples:

```
<HTML>
<HEAD>
<TITLE>Um Exemplo simples de HTML</TITLE>
</HEAD>
<BODY>
<H1>HTML é fácil de aprender</H1>
<P>Bemvindo ao mundo do HTML.
Este é o primeiro parágrafo.</P>
<P>Este é o segundo parágrafo.</P>
</BODY>
</HTML>
```

Para transformar o código acima num documento HTML basta apenas selecionar o conteúdo desde <HTML> até </HTML> e copiar para, por exemplo, o bloco de notas do Windows. Depois disso basta salvar com qualquer nome e extensão .html ou .htm (tomando o cuidado de não deixar o bloco de notas dar a sua contribuição, que é adicionar um .txt no final do nome), e então abrir esse documento com um browser.

NOTA:

HTML não é "case sensitive" (caixa alta ou baixa não faz diferença), por exemplo <title> é equivalente a <TITLE> ou <TiTIE>, porém existem algumas poucas exceções a essa regra.

IMPORTANTE:

Nem todas as tags são suportadas por todos os Web browsers, e se um browser não suporta uma tag ele simplesmente a ignora. Entretanto, qualquer texto posicionado entre um par de tags desconhecida é mostrado pelo browser.

Inclusive, alguns browsers conseguem mostrar um arquivo de texto mesmo que as tags <HTML>, <HEAD>, <TITLE> e <BODY> não tenham sido incluídas.

DETALHE:

Para ver uma cópia do arquivo que o browser leu para interpretar e mostrar a página, basta selecionar "View Source" (Ver Código Fonte ou equivalente) num dos menus do browser. Um modo mais rápido e fácil de acessar esse menu é clicar com o botão direito do mouse sobre a página.

Observar como é escrito o código fonte das páginas HTML é uma maneira bem simples de aprender a linguagem HTML, inclusive porque uma vez visualizado o código fonte pode-se salvar o arquivo em disco e usá-lo como ponto de partida para construção de um novo documento HTML.

1.2.2 - DESCRIÇÃO DAS PRINCIPAIS TAGS

<HTML> - Esse elemento transmite ao browser que o arquivo que ele leu contém informação de código HTML. Na verdade essa tag é usada por uma questão de padronização, pois a extensão .html do arquivo já garante ao browser que se trata de um documento HTML.

<HEAD> - O elemento HEAD identifica a primeira parte do documento HTML. É no HEAD que se deve incluir, entre outras coisas, o título <TITLE> do documento, que será apresentado como título da página na barra de título da janela browser.

<TITLE> - O Título identifica a página num contexto geral, pois o que for definido na tag <TITLE> será mostrado na barra de título da janela do browser e por isso deve ser escolhido um título único, descritivo e relativamente curto. O título também é usado para identificar a página para os motores de busca (robots), tais como HotBot, Infoseek ou Google (google.com). Geralmente o título deve ter no máximo 64 caracteres.

<BODY> - A segunda (e maior) parte do documento HTML é o corpo, que inclui todo o conteúdo da página. Esse conteúdo é mostrado no interior da janela do browser.

Exemplos de atributos para <BODY>:

A tag

<BODY BGCOLOR="#000000" TEXT="#FFFFFF" LINK="#9690CC" VLINK="#9690CC" ALINK="#00FF00">
torna o corpo do documento com fundo na cor preta (BGCOLOR), texto branco (TEXT) e hiperlinks prata (LINK) que permanecem na mesma cor após o clique (VLINK - Visited Link). No momento do clique o link adquire a cor verde (ALINK - Active Link).

Em lugar de BGCOLOR poderia ser usado BACKGROUND.

Exemplo 1.2.2.1:

<BODY BACKGROUND="/util/www/img2/stars.jpg">
que inclui como tema de fundo do documento a figura "stars.jpg".

NOTA 1:

Em HTML as cores RGB (Red, Green, Blue) são obtidas pela combinação de 6 caracteres hexadecimais (base 16) tomados em pares RRGGBB, onde cada par está no intervalo de intensidade 00-FF (cada caractere do par está no intervalo de 0 a 15 na base decimal). Por exemplo, 000000 é totalmente preto (intensidade zero para as três cores), FF0000 é vermelho, 00FF00 é verde, 0000FF é azul e FFFFFFFF é branco.

Quanto maior o número hexadecimal mais intensa a cor, por isso 0000FF é mais azul que 00000F, pois FF é maior que 0F (na verdade 00000F é quase totalmente preto, pois a contribuição do azul é muito pequena).

Usando esse sistema para as cores é possível representar 256x256x256 cores (16 milhões), pois 16x16=256. Mas convém notar que uma coisa são as cores possíveis de serem apresentadas no documento HTML, outra é a capacidade do monitor do cliente HTTP em reproduzir essas cores.

NOTA 2:

As cores básicas, tais como vermelho, azul, verde, etc. também podem ser representadas pelo seu respectivo nome em inglês, por exemplo

<BODY BGCOLOR="red">

1.2.3 - TAGS MAIS USADAS NO INTERIOR DO CORPO DO DOCUMENTO HTML

1.2.3.1 - Headings

<Hx>, onde x vai de 1 a 6 (não confundir HEADING com HEAD). Quanto maior o número x, menor o tamanho do texto.

Exemplo:

<BODY>

```
<H2>Texto "aumentado" de tamanho</H2>
</BODY>
```

1.2.3.2 - Paragraphs

<P>, delimita parágrafos no documento.

Exemplo 1.2.3.2.1:

```
<BODY>
<P ALIGN="center">
Este parágrafo
    está
                centralizado !
</P>
</BODY>
```

Vai mostrar na janela browser:

Este parágrafo está centralizado !

IMPORTANTE:

Ao contrário da maioria dos processadores e editores de texto, em HTML espaços em branco, "carriage return" ("Enter") e tabulações são mostrados comprimidos em um único espaço em branco. A vantagem é que não é necessário se preocupar com a extensão das linhas, pois o browser é que vai se encarregar da melhor apresentação do conteúdo do documento na janela. Exceção: tag <PRE> (texto preformatado).

NOTA:

Na linguagem HTML toda a formatação do texto é obtida com tags apropriadas. As marcações inseridas pelos editores de texto, tais como quebras de linha, tabulações e parágrafos são desprezadas pelo browser.

1.2.3.3 - Lists

As listas são usadas normalmente para criar índices de páginas HTML, tanto índices numerados quanto não-numerados. A tag determina listas não numeradas (unnumbered) e determina listas numeradas.

Exemplo 1.2.3.3.1:

```
<UL TYPE="circle">
    <LI> maçã
    <LI> pêra
    <LI> laranja
</UL>
```

irá mostrar:

```
o maçã
o pêra
o laranja
```

Atributos:

```
<UL TYPE="disc|square|circle"></UL>
```

```
<LI TYPE="disc|square|circle">
```

onde,

DISC é um círculo preenchido, SQUARE é um quadrado vazado e CIRCLE é um círculo vazado.

Exemplo 1.2.3.3.2:

```
<OL>
    <LI> morango
    <LI> pêssego
    <LI> uva
</OL>
```

irá mostrar

```
1. morango
2. pêssego
3. uva
```

Atributos:

```
<OL TYPE="1|a|A|i|I" START="n"></OL>
```

```
<LI TYPE="1|a|A|i|I" VALUE="n">
```

onde,

TYPE="1" é o padrão de numeração, a e A são numerações alfabéticas, i e I são numerações em algarismos romanos e 'n' em START="n" é um número onde se queira iniciar a listagem e VALUE="n" é um valor a ser atribuído à tag .

Exemplo 1.2.3.3.3:

```
<OL TYPE="1" START="7">
```

```
<LI> banana
```

```
<LI> melancia
```

```
<LI> manga
```

```
</OL>
```

irá mostrar:

7. banana

8. melancia

9. manga

Exemplo 1.2.3.3.4:

```
<OL TYPE="a">
```

```
<LI> morango
```

```
<LI> abacaxi
```

```
<LI TYPE="I"> pêssego
```

```
<LI> uva
```

```
<LI TYPE="A" VALUE="3"> laranja
```

```
</OL>
```

irá mostrar:

a. morango

b. abacaxi

III. pêssego

IV. uva

C. laranja

1.2.3.4 - Nested Lists

Listas aninhadas: consiste em juntar as listas numeradas e não-numeradas para obter um índice mais elaborado.

Exemplo 1.2.3.4.1:

```
<OL>
```

```
<LI> A few New England states:
```

```
<UL TYPE="circle">
```

```
<LI> Vermont
```

```
<LI> New Hampshire
```

```
<LI> Maine
```

```
</UL>
```

```
<LI> Two Midwestern states:
```

```
<UL TYPE="circle">
```

```
<LI> Michigan
```

```
<LI> Indiana
```

```
</UL>
```

```
</OL>
```

Irá mostrar:

1. A few New England states:

o Vermont

o New Hampshire

o Maine

2. Two Midwestern states:

o Michigan

o Indiana

1.2.3.5 - Definition Lists

Listas de definição: normalmente consistem em alternar um termo de definição, definition term <DT>, e uma definição de definição, definition definition <DD>. Os browsers normalmente formatam a definição numa linha nova com parágrafo (é a tabulação em HTML).

Exemplo 1.2.3.5.1:

```
<DL>
  <DT> UNINOVE
  <DD> Universidade Nove de Julho.
  <DT> USP
  <DD> Universidade de São Paulo.
</DL>
```

irá mostrar:

```
UNINOVE
  Universidade Nove de Julho.
USP
  Universidade de São Paulo.
```

NOTA:

As entradas nas tags <DT> e <DD> podem conter múltiplos parágrafos <P>, listas ou outras informações de definição.

1.2.3.6 - Preformatted Text

Texto Preformatado: a tag <PRE> torna os espaços, linhas e tabulações do editor de textos significantes. É útil para a listagem de programas, entre outras coisas, mas tem a desvantagem de não permitir ao texto "se acomodar" no interior da janela browser caso a linha de texto seja mais extensa que a janela.

Exemplo 1.2.3.6.1:

```
<PRE>
#!/usr/bin/perl

print "Bemvindo ao mundo do Perl !!!\n";

exit;
</PRE>
```

irá mostrar

```
#!/usr/bin/perl

print "Bemvindo ao mundo do Perl !!!\n";

exit;
```

Atributo:

```
<PRE COLS="n" WRAP></PRE>
```

especifica o número máximo de caracteres que serão mostrados na mesma linha. Esse atributo efetivamente quebra ou ajusta (WRAP) as linhas de texto na extensão dada por "n".

OBSERVAÇÃO:

A tag <PRE> não lista as tags HTML, que são interpretadas pelo browser.

1.2.3.7 - Plaintext

Texto Puro: tudo que estiver APÓS a tag <PLAINTEXT> será interpretado como texto puro, incluindo tags HTML que não serão mais interpretadas pelo browser.

Exemplo 1.23.7.1:

```
<PLAINTEXT>
Exemplo de documento HTML mínimo:
<HTML>
```

```
<HEAD>
<TITLE>Documento mínimo</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<P ALIGN="CENTER">Documento HTML mínimo</P>
</BODY>
</HTML>
```

OBSERVAÇÃO 1:

A tag <PLAINTEXT> NÃO tem </PLAINTEXT>, o que significa que após inserida essa tag todas as tags restantes do documento HTML não serão mais interpretadas pelo browser, apenas listadas como texto.

OBSERVAÇÃO 2:

Uma tag similar à <PLAINTEXT> é <XMP>, que tem a vantagem de admitir </XMP>.

1.2.3.8 - Extended Quotations

Citações Estendidas: a tag <BLOCKQUOTE> é usada para incluir texto num bloco separado. Muitos browsers geralmente mudam a margem esquerda para separar a aparência do bloco na janela browser.

Exemplo 1.2.3.8.1:

```
<P>Omitir palavras desnecessárias:</P>
<BLOCKQUOTE>
<P>"Uma sentença não deve conter palavras desnecessárias, um parágrafo não deve conter sentenças
desnecessárias."</P>
<P>--O autor </P>
</BLOCKQUOTE>
```

irá mostrar

Omitir palavras desnecessárias:

"Uma sentença não deve conter palavras desnecessárias,
um parágrafo não deve conter sentenças desnecessárias."

--O autor

1.2.3.9 - Forced Line Breaks

Quebra de linha forçada: a tag
 força a quebra da linha no local onde for inserida.

Exemplo 1.2.3.9.1:

A linha será quebrada
na tag Forced Line Break.

irá mostrar:

A linha será quebrada
na tag Forced Line Break.

atributos:

```
<BR CLEAR="left|all|right|none">
```

permite que se quebre a linha de texto e se consiga alinhamento mesmo que essa quebra tenha ocorrido, por exemplo, logo após a inserção de uma figura bem mais alta que a altura padrão da linha de texto.

NOTA:

Parece evidente que a tag
 NÃO deva ter </BR>.

1.2.3.10 - Horizontal Rules

Réguas horizontais: a tag <HR> produz uma linha horizontal na janela do browser.

Exemplo 1.2.3.10.1:

```
<HR SIZE="4" WIDTH="50%" NOSHADE>
```

irá mostrar uma linha horizontal com altura 4 pixels, extensão 50% da janela browser e centralizado. O atributo NOSHADE (sem sombra) torna a régua horizontal uma linha sólida.

Exemplo 1.2.3.10.2:

```
<HR WIDTH="50%" ALIGN="right">
```

irá mostrar uma linha "sombreada" com extensão 50% da janela browser e alinhada à direita.

atributos:

```
<HR SIZE="n" WIDTH="n|n%" ALIGN="left|center|right" NOSHADE>
```

OBSERVAÇÃO:

Se não houver o atributo WIDTH a linha ocorrerá em toda a extensão da janela.

Se não houver o atributo ALIGN o alinhamento será à esquerda da janela browser.

Essa tag NÃO tem </HR>.

1.2.3.11 - Formatação de caracteres

A tag formata a fonte usada para apresentar o texto.

Exemplo 1.2.3.11.1:

```
<FONT FACE="arial" COLOR="#00F000" SIZE="2">
```

formata a fonte para tipo arial na cor verde e tamanho 2.

Exemplo 1.2.3.11.2:

```
<FONT FACE="arial, helvetica,sans-serif" COLOR="#00000F" SIZE="3">fonte com cor 00000F<BR>
```

```
<FONT COLOR="#0000F0"> fonte com cor 0000F0</FONT></FONT>
```

irá mostrar:

fonte com cor 00000F

fonte com cor 0000F0

onde o texto da primeira linha tem cor preta e da segunda azul.

NOTA:

Quando o browser interpreta o texto HTML ele procura pela primeira fonte que encontra no atributo FACE e, caso essa fonte esteja instalada ela será usada, caso contrário o browser tenta a próxima fonte.

atributos:

```
<FONT FACE="fonte1, fonte2, etc." COLOR="#RRGGBB" SIZE="n"></FONT>
```

OBSERVAÇÃO 1:

A escolha da fonte é importante não somente pela melhor apresentação do texto, mas também porquê os nomes de fontes são dependentes de plataforma, isto é, diferentes Sistemas Operacionais normalmente não têm nomes de fontes em comum. Por exemplo, a fonte ARIAL é usada no Windows, seu equivalente no Linux/Unix é HELVETICA e no Imac (Apple) é SANS-SERIF.

No atributo FACE é interessante definir vários nomes de fontes para poder atender as principais plataformas atuais (Windows, iMac, Unix/Linux).

OBSERVAÇÃO 2:

Caso a fonte não exista (ou não esteja instalada), o browser irá fazer uma aproximação, ou seja, ele irá escolher uma fonte parecida com a que aparece no atributo FACE da tag .

1.2.3.12 - Estilo Físico de Caracteres

Os caracteres podem ser realçados/formatados com as seguintes tags:

```
<B>Bold, negrito</B>
```

```
<I>Itálico</I>
```

```
<TT>Teletype ou typewriter</TT>
```

```
<U>Sublinhado</U>
```

```
<SMALL>Reduzido</SMALL>
```

```
<STRIKE>Caracteres riscados</STRIKE>
```

```
<BIG >Aumentado</BIG>
```


O HEAD do documento HTML pode conter, entre outras, as seguintes tags: <TITLE>, <SCRIPT>, <META>.

1.2.4.1 - Título do documento

A tag <TITLE> dá um título ao documento HTML.

Exemplo:

```
<HTML>
<HEAD>
<TITLE>Este é o meu documento</TITLE>
</HEAD>
<BODY>
Segue o documento...
</BODY>
</HTML>
```

irá mostrar o documento HTML com o título "Este é o meu documento" na barra de título da janela browser.

1.2.4.2 - Script (roteiro)

Essa tag pode ter inúmeras funções, pois no interior dela é possível trabalhar com estruturas semelhantes a programas. Entre outras funções essa tag possibilita interatividade entre o documento HTML e o usuário HTTP. No tópico JavaScript aprenderemos mais sobre <SCRIPT>.

Exemplo:

```
<SCRIPT LANGUAGE="JavaScript">
```

NOTA:

A tag <SCRIPT> não está confinada somente ao <HEAD> do documento HTML, ela pode aparecer também no interior do <BODY>.

1.2.4.3 - Meta-Infomação

Essa tag contém meta-informação ou especifica cabeçalho através de dois atributos: NAME e HTTP-EQUIV.

NAME é uma informação que pode ser aproveitada por alguém que visualize o código fonte ou por ferramentas de busca tais como Alta Vista, Infoseek ou Google.

HTTP-EQUIV envia seu valor e o do atributo CONTENT para ser usado pelo servidor como HEADER (cabeçalho) do documento.

Atributos:

```
<META NAME="string"|HTTP-EQUIV="string" CONTENT="string">
```

Exemplo 1.2.4.3.1:

```
<HEAD>
<META NAME="description" CONTENT="Descrição da página">
<META NAME="keywords" CONTENT="Palavras-chave que descrevem o conteúdo da página. Robots gostam disso">
<META NAME="author" CONTENT="Nome do Autor">
<META NAME="generator" CONTENT="Criado na Unha Mesmo">
</HEAD>
```

Exemplo 1.2.4.3.2:

```
<HEAD>
<META HTTP-EQUIV="IFUSP - Instituto de Física da USP" CONTENT="no-cache">
<META HTTP-EQUIV="Expires" CONTENT="Mon, 01 Jan 1990 12:00:00 GMT">
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
</HEAD>
```

Neste exemplo <META> é usado para forçar o browser a não usar seu cache em disco local na hora de apresentar o conteúdo do documento ao cliente HTTP. Se esse procedimento tem a vantagem de obrigar o cliente a pegar a página no servidor Web, por outro lado acarreta fluxo desnecessário de pacotes TCP/IP.

OBSERVAÇÃO:

Os browsers atuais têm fama de serem rápidos, mas na verdade boa parte dessa "rapidez" tem origem na leitura da página em disco local (cache), uma vez que após acessada pela primeira vez essa página é salva no cache do browser. Com isso economiza-se o download da página pela Internet, mas traz também o inconveniente do browser se recusar a mostrar aquela atualização da página que acabou de ser feita.

Exemplo 1.2.4.3.3:

```
<HEAD>  
<META HTTP-EQUIV="Refresh" CONTENT="5; URL=another_page.html">  
</HEAD>
```

Neste exemplo <META> é usado para efetuar um "refresh" (renovação) HTML, ou seja, 5 segundos após acessada a página que contém o META-REFRESH o cliente HTTP é forçado a receber a página "another_page.html".

NOTA:

Por ser muito simples e prático, o META-REFRESH é um método de redirecionamento muito usado na WWW.

1.3 - PROGRAMANDO EM HTML

1.3.1 - Inserção de imagens

Em HTML as imagens são inseridas no documento através da tag .

Exemplo 1.3.1.1:

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

Exemplo 1.3.1.2:

```
<IMG SRC="http://www.dominio.com.br/imagens/imagem.jpg">
```

NOTA 1:

A maioria dos browsers só reconhece imagens nos formatos GIF e JPG. Browsers mais recentes reconhecem tanto imagens GIF com fundo transparente quanto GIF animados, estes últimos muito usados na construção de "banners".

Atributos:

```
<IMG SRC="URL" ALIGN="left|right|top|middle|bottom" HEIGHT="n" WIDTH="n" VSPACE="n" HSPACE="n" BORDER="n" ALT="string" USEMAP="#string" ISMAP>
```

onde,

URL é o endereço da imagem,

ALIGN define o alinhamento da imagem (à esquerda ou à direita em relação ao bloco e superior, no centro ou inferior em relação à linha),

HEIGHT e WIDTH especificam dimensões em pixels na horizontal e na vertical que a imagem assumirá na janela do browser,

VSPACE e HSPACE especificam a distância vertical e horizontal em pixels entre a imagem e o texto,

BORDER define a espessura da borda em pixels,

ALT define um texto alternativo que normalmente é mostrado quando se põem a seta do mouse em cima da imagem,

USEMAP e ISMAP será mostrado logo adiante.

Exemplo 1.3.1.3:

```
<P ALIGN="center">
```

Este é um bloco de texto. No seu interior será inserida a imagem que será visualizada no interior desse bloco de texto, alinhada à direita.

```
</P>
```

NOTA 2:

Imagens muito grandes (tamanho em bytes) costumam demorar mais que o próprio documento HTML para fazer o download pela Internet, mas se o browser já souber as dimensões que a imagem ocupará na página (atributos HEIGHT e WIDTH) ele poderá interpretar e apresentar a página, mesmo sem ter finalizado o download da imagem.

NOTA 3:

A tag NÃO tem

1.3.2 - Links

Os links ou hyperlinks permitem "interligar" as páginas na Internet, ou seja, através de links se vai de uma página para outra com um simples clique do mouse.

Atributos:

```
<A HREF="URL|string" TARGET="string"></A>
```

onde,

URL é o endereço do documento,

HREF="string" é o caminho e o nome de arquivo e

TARGET="string" define a janela browser onde o endereço referenciado no atributo HREF será exibido (o "default" é "_self").

Sobre o atributo TARGET veremos mais detalhes à frente, em FRAMES.

Exemplo 1.3.2.1:

```
<A HREF="http://www.if.usp.br">Instituto de Física da USP</A>
```

Exemplo 1.3.2.2:

```
<A HREF="page.html">  
<IMG SRC="amigo.jpg" HEIGHT="80" WIDTH="30" ALIGN="bottom" ALT="Domínio" BORDER="0">  
</A>
```

Exemplo 1.3.2.3:

```
<A HREF="mailto:jairo@if.usp.br">Jairo</A>
```

Neste caso, o cliente HTTP precisa ter um mailer instalado para que o link funcione. Para um usuário Windows temos como exemplos de mailers o Outlook, o Netscape Messenger e o Eudora.

Exemplo 1.3.2.4:

```
<A HREF="page2.html#Internet">Internet</A>
```

Neste exemplo também se faz necessário introduzir a tag <A NAME> no local do BODY do documento page2.html para onde se quer a barra de rolagem do navegador vá, da seguinte maneira:

```
<A NAME="Internet">Algum texto aqui</A>
```

NOTA:

O caso do link com tag <A NAME> é muito útil em páginas muito extensas, onde o cliente HTTP tem que usar a barra de rolagem vertical do browser para visualizar o conteúdo da página. Uma solução para facilitar a vida do usuário é usar links conforme o exemplo 1.3.1.4, que ao ser "clicado" faz a tarefa de movimentar a barra de rolagem para o local desejado da página.

Exemplo 1.3.2.5:

```
<A HREF="figura.jpg">Clique na figura</A>  
<A HREF="arquivo.doc">Clique no DOC do Word</A>
```

1.3.3 - Mapa clicável

A tag <MAP> permite que se defina "regiões clicáveis" em uma imagem, e a essas regiões pode-se associar documentos HTML para onde o navegador vai caso o cliente clique na região.

Essa tag tem algo de parecido com a tag <A HREF> acima, porém é um poucos mais sofisticada.

NOTA 1:

No caso do link comum, qualquer lugar da imagem que for "clicado" levará ao mesmo documento HTML.

Exemplo:

```
<IMG SRC="figura.gif" ISMAP USEMAP="#meumapa" BORDER="0" ALT="Este é meu mapa">  
<MAP NAME="meumapa">  
  <AREA HREF="page1.html" ALT="Página1" SHAPE="circle" COORDS="50,80,30">  
  <AREA HREF="page2.html" ALT="Página2" SHAPE="rect" COORDS="110,140,160,180">  
  <AREA HREF="page3.html" ALT="Página3" SHAPE="poly" COORDS="70,175,110,175,100,240">  
</MAP>
```

onde,

o valor de COORDS deve ser definido de acordo com o formato da região:

rect: retângulo - devem ser especificadas as coordenadas do canto superior esquerdo e do canto inferior direito;

circle: círculo - devem ser especificadas as coordenadas do centro do círculo e do raio;

poly: polígono - devem ser especificadas as coordenadas dos vértices do polígono.

NOTA 2:

Determinar as coordenadas do mapa clicável pode ser uma tarefa árdua caso não se disponha de um bom editor WYSIWYG.

1.3.4 - Tabelas

A tag <TABLE> permite criar tabelas nos documentos HTML.

Atributos de <TABLE>:

```
<TABLE BORDER="n" CELSPACING="n" CELLPADDING="n" WIDTH="n" ALIGN="left|center|right"  
BGCOLOR="#RRGGBB"></TABLE>
```

onde,

BORDER determina bordas na tabela,

CELLSPACING define o número de pixels entre o conteúdo da célula e suas bordas,

CELLPADDING define o número de pixels entre as células,
WIDTH define a largura da tabela em pixels,
ALIGN é o alinhamento da tabela em relação à página (vários browsers não levam esse atributo em conta!) e
BGCOLOR é a cor de fundo da tabela (se não for definida cor de fundo a tabela é transparente).

Para criar as linhas da tabela usa-se a tag <TR>.

Atributos de <TR>:

```
<TR ALIGN="left|center|right" VALIGN="top|middle|bottom|baseline" BGCOLOR="#RRGGBB"></TR>
```

onde,

ALIGN é o alinhamento horizontal de todas as células da linha,

VALIGN é o alinhamento vertical e

BGCOLOR define a cor de fundo para a linha e tem precedência sobre a cor de fundo da tabela.

Para criar as células da tabela usa-se a tag <TD>.

Atributos de <TD>:

```
<TD ALIGN="left|center|right" VALIGN="top|middle|bottom|baseline" NOWRAP WIDTH="n" HEIGHT="n" ROWSPAN="n" COLSPAN="n" BGCOLOR="#RRGGBB"></TD>
```

onde,

NOWRAP faz com que o texto contido na célula NÃO seja quebrado em várias linhas,

WIDTH e HEIGHT permitem indicar a largura e altura da célula em pixels,

ROWSPAN e COLSPAN permitem que a célula ocupe respectivamente mais de uma linha ou mais de uma coluna, isto é, ROWSPAN funde "n" linhas e COLSPAN funde "n" colunas e

BGCOLOR define a cor de fundo da célula tem precedência sobre a cor da tabela ou linha.

Para definir células de cabeçalho usa-se a tag <TH>.

Atributos de <TH>:

```
<TH ALIGN="left|center|right" VALIGN="top|middle|bottom|baseline" NOWRAP WIDTH="n" HEIGHT="n" ROWSPAN="n" COLSPAN="n" BGCOLOR="#RRGGBB"></TH>
```

que é análogo ao caso do <TD> acima.

Exemplo 1.3.4.1:

```
<TABLE>
  <TH ALIGN="center" COLSPAN="2">Cabeçalho</TH>
  <TR>
    <TD>Célula1</TD>
    <TD>Célula2</TD>
  </TR>
  <TR>
    <TD>Célula3</TD>
    <TD>Célula4</TD>
  </TR>
</TABLE>
```

Exemplo 1.3.4.2:

```
<TABLE BORDER="2" CELSPACING="3" CELLPADDING="2" BGCOLOR="#FFFF00">
  <TR>
    <TD>1.1</TD>
    <TD>1.2</TD>
    <TD>1.3</TD>
    <TD>1.4</TD>
  </TR>
  <TR>
    <TD>2.1</TD>
    <TD COLSPAN="2">2.2</TD>
    <TD>2.4</TD>
  </TR>
  <TR>
    <TD>3.1</TD>
    <TD>3.2</TD>
    <TD ROWSPAN="2">3.3</TD>
    <TD>3.4</TD>
  </TR>
```

```

<TR>
  <TD>4.1</TD>
  <TD>4.2</TD>
  <TD>4.4</TD>
</TR>
<TR>
  <TD>5.1</TD>
  <TD>5.2</TD>
  <TD>5.3</TD>
  <TD>5.4</TD>
</TR>
</TABLE>

```

NOTA:

Apesar de aparentemente simples e descomprometida, é a tag <TABLE> a responsável pela grande variedade de "design" de páginas bem estruturadas que se vê na WWW atualmente. Já houve época em que a ênfase estava nos FRAMES (logo abaixo) como estrutura básica de criação de páginas bem elaboradas.

1.3.5 - Múltiplas janelas (Frames)

Para dividir uma página em várias janelas, usa-se a tag <FRAMESET>.

IMPORTANTE 1:

Com exceção do Netscape 2, somente browsers versões 3 ou maior interpretam esta tag.

Atributos:

```
<FRAMESET ROWS="n,r,s,...,*" | COLS="n,r,s,...,*" FRAMEBORDER="yes|no" BORDER="n"></FRAMESET>
```

onde,

ROWS ou COLS definem uma divisão do documento em faixas ou colunas, que irão conter as janelas com as dimensões especificadas em pixels (n,r,...) ou em percentual (ex: 20%, 30%, etc.) ou valores relativos usando o caracter "*" que "completa" a janela com o restante,

FRAMEBORDER define a existência ou não de uma sombra de uma borda em torno da janela selecionada e BORDER define a largura da borda em pixels.

IMPORTANTE 2:

Dentro de um <FRAMESET> só podem ser colocadas as tags <FRAME> e <NOFRAME>.

Como exemplo temos o documento HTML page1.html que inclui em estrutura de FRAMES os documentos page2.html, page3.html e page4.html.

DETALHE:

O browser abre o documento page1.html mas irá mostrar apenas as páginas page2.html, page3.html e page4.html.

Exemplo 1.3.5.1:

```

<HTML>
<HEAD>
<TITLE>Exemplo de FRAME (documento page1.html)</TITLE>
</HEAD>
<FRAMESET COLS="100,*" BORDER="10">
  <NOFRAME>
  Infelizmente seu browser não suporta frames !
  </NOFRAME>
  <FRAME SRC="page2.html" NAME="janela1" SCROLLING="yes" NORESIZE>
  <FRAMESET ROWS="40%,*">
    <FRAME SRC="page3.html" NAME="janela2">
    <FRAME SRC="page4.html" NAME="janela3">
  </FRAMESET>
</FRAMESET>
</HTML>

```

Exemplo 1.3.5.2:

Se houvesse um link em page2.html para abrir o documento page5.html, e quiséssemos abri-lo no frame 'janela2', o link necessitaria o atributo TARGET:

```
<A HREF="page5.html" TARGET="janela2">link para page5.html na janela2</A>.
```

Outros 'TARGETS' que também poderiam ser usados:

_self: a própria janela (é o "default");

_top: destrói toda a estrutura de frames, pois o documento abre "sozinho" na janela browser;

_parent: destrói o frameset da janela que contém o link;

_blank: o browser criará uma nova janela browser para exibir o documento;

window, new: o mesmo que "_blank".

1.3.6 - Formulários

A tag <FORM> permite ao usuário enviar informações a um banco de dados através do servidor Web. Como exemplo, as informações enviadas no FORM podem tanto ser um cadastro num banco de currículos na Internet quanto verificar a disponibilidade de horários de vôos de determinada empresa de aviação.

Exemplo 1.3.6.1:

```
<FORM ACTION="/cgi-bin/formulario.cgi" METHOD="post">
```

Time favorito:

```
<INPUT TYPE="text" NAME="time" VALUE="Nenhum" SIZE="15" MAXLENGTH="30">
```

```
</FORM>
```

Atributos:

```
<FORM ACTION="URL" METHOD="get|post" TARGET="string"></FORM>
```

onde,

o atributo obrigatório ACTION define o endereço do programa (ou script) que vai receber os dados do formulário enviado pelo cliente (veremos mais detalhes no tópico cgi-bin, mais à frente),

METHOD especifica o método pelo qual os dados do formulário serão enviados, normalmente é usado POST pois suporta mais de 255 caracteres enviados no FORM,

TARGET especifica a janela onde o documento HTML "gerado" pelo programa (ou script) será exibido e tem como default o valor "_self" (mesma janela do formulário).

1.3.6.1 - Campos em um formulário - os campos do formulário podem ser obtidos usando as 3 seguintes tags:

1.3.6.1.1 - INPUT, que é uma caixa genérica

Atributos:

```
<INPUT TYPE="text|password|hidden|checkbox|radio|reset|submit|image|file" NAME="string" VALUE="string" SIZE="n" MAXLENGTH="n" ALIGN="left|right|top|middle|bottom" CHECKED SRC="URL">
```

onde,

TYPE é o tipo do campo do formulário (texto, senha, oculto, caixa de verificação, botão de rádio, limpar o campo do formulário, botão submeter, figura, arquivo),

NAME é o nome do campo e será usado pelo script (ou programa) para diferenciar os diversos parâmetros recebidos no FORM,

VALUE permite definir um valor default que já irá aparecer na tela quando a página do formulário for acessada,

SIZE define o tamanho da caixa de texto em caracteres (TYPE="text" ou TYPE="password"),

MAXLENGTH define o tamanho máximo da linha de texto em caracteres (TYPE="text" ou TYPE="password"),

ALIGN é o alinhamento do campo em relação ao documento HTML,

CHECKED já seleciona o campo (TYPE="checkbox" ou TYPE="radio"),

SRC é o endereço da imagem (TYPE="image").

Exemplo:

```
<FORM METHOD="post" ACTION="http://www.dominio.com.br/cgi-bin/forms/formulario.cgi">
```

Seu nome:

```
<INPUT TYPE="text" NAME="nome" SIZE="30" MAXLENGTH="50"><br><br>
```

Você é estudante?

```
&nbsp; sim:<INPUT TYPE="radio" NAME="estuda" VALUE="sim" CHECKED>
```

```
&nbsp; não:<INPUT TYPE="radio" NAME="estuda" VALUE="nao"><br><br>
```

Você gostaria de ganhar:


```
&nbsp; Carro<INPUT TYPE="checkbox" NAME="carro" VALUE="carro" CHECKED>
```

```
&nbsp; Moto<INPUT TYPE="checkbox" NAME="moto" VALUE="moto">
```

```
&nbsp; Carroça<INPUT TYPE="checkbox" NAME="carroca" VALUE="carroca"><br><br>
<INPUT TYPE=submit NAME="enviar" VALUE="Enviar o formulário">
</FORM>
```

1.3.6.1.2- TEXTAREA, que é uma área de texto com quebra de linhas.

Atributos:

```
<TEXTAREA NAME="string" ROWS="n" COLS="n"></TEXTAREA>
```

onde,

NAME é o nome do campo,

ROWS e COLS são os números de linhas e colunas que a caixa de diálogo irá ocupar no documento HTML.

NOTA:

Parece evidente que o campo TEXTAREA deve usar o método "post" do FORM, em vista da limitação de 255 caracteres imposta pelo método "get".

Exemplo:

```
<FORM METHOD="post" ACTION="/cgi-bin/cadastro.cgi">
```

Escreva suas sugestões:


```
<TEXTAREA NAME="sugerir" ROWS="3" COLS="50">
```

Gostaria de obter mais informações a respeito ...

```
</TEXTAREA>
```

```
</FORM>
```

1.3.6.1.3 - SELECT, que é uma caixa de lista suspensa.

Atributos:

```
<SELECT NAME="string" SIZE="n" MULTIPLE></SELECT>
```

onde,

NAME é o nome do campo,

SIZE é o número de linhas que será mostrada da caixa de lista suspensa (se for omitido será apenas uma linha, que é o "default"),

MULTIPLE só deve ser usado quando SIZE for maior que 1.

Exemplo 1.3.6.1.3.1:

```
<FORM METHOD="post" ACTION="/cgi-bin/cadastro.cgi">
```

Local das férias:


```
<SELECT NAME="ferias">
```

```
  <OPTION VALUE="manaus">Manaus</OPTION>
```

```
  <OPTION VALUE="fortaleza" SELECTED>Fortaleza</OPTION>
```

```
  <OPTION VALUE="salvador">Salvador</OPTION>
```

```
  <OPTION VALUE="saopaulo">São Paulo</OPTION>
```

```
</SELECT>
```

```
</FORM>
```

Exemplo 1.3.6.1.3.2:

```
<FORM METHOD="post" ACTION="/cgi-bin/cadastro.cgi">
```

Opcionais que você deseja:


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

```
  <OPTION VALUE="computador" SELECTED>Computador</OPTION>
```

```
  <OPTION VALUE="televisao">Televisão</OPTION>
```

```
  <OPTION VALUE="frigobar" SELECTED>Frigobar</OPTION>
```

```
  <OPTION VALUE="videogame">Video Game</OPTION>
```

```
  <OPTION VALUE="secretaria">Secretária Eletrônica</OPTION>
```

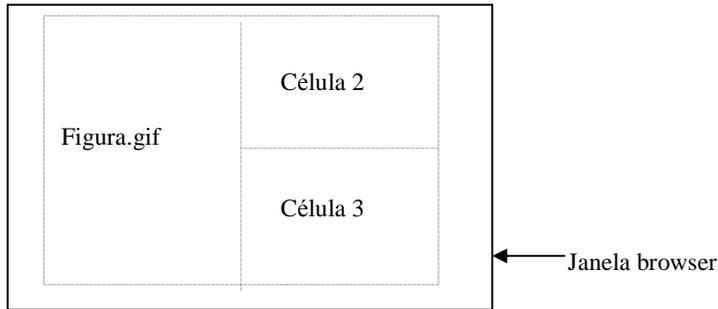
```
</SELECT>
```

```
</FORM>
```

1.4 - EXERCÍCIOS

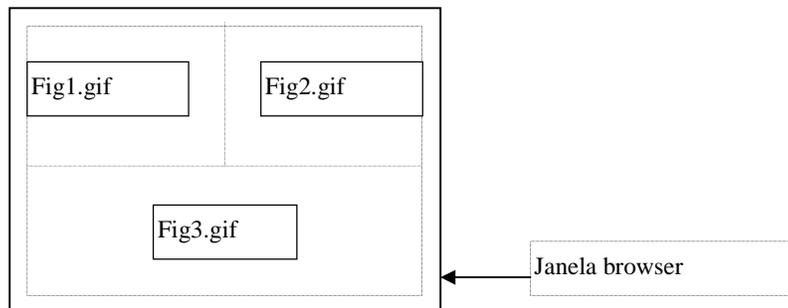
1.4.1 - Crie um documento HTML que apresente a cor de fundo verde e 1 figura alinhada à direita na página. A figura é um link para <http://www.uninove.br>.

1.4.2 - Crie um documento HTML que apresente a cor de fundo vermelha e a seguinte tabela centralizada na janela browser:



Ao clicar na figura, abre a página "page2.html" numa nova janela browser. NÃO use frames.

1.4.3 - Crie um documento HTML que tenha cor de fundo azul e que apresente três imagens distribuídas na página conforme a figura abaixo:



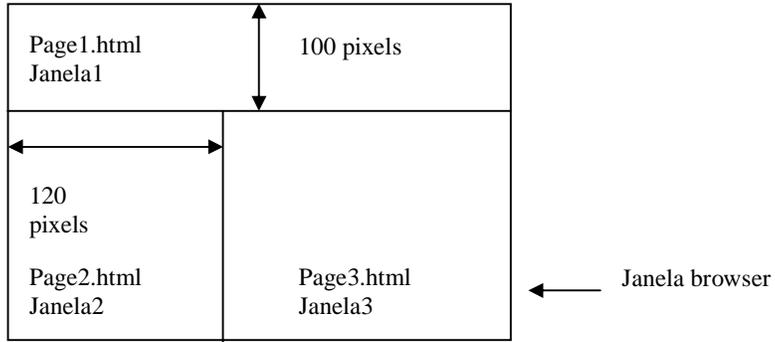
Ao redimensionar a janela browser, a Fig1.gif permanece alinhada à esquerda, a Fig2.gif permanece alinhada à direita e a Fig3.gif permanece centralizada.

1.4.4 - Crie um documento HTML que ao ser acessado mostre a seguinte mensagem centralizada na janela browser:

Mudamos de endereço, agora é <http://www.if.usp.br>
Se o seu browser não redirecionar em 10 segundos, clique [aqui](#)

onde [aqui](#) é um link para <http://www.if.usp.br> e, caso o usuário não clique no link, em 10 segundos ocorrerá o "refresh HTML" com redirecionamento para o endereço acima.

1.4.5 - Crie um documento HTML em FRAME de tal modo que, ao acessar o documento base.html mostre a seguinte página:



onde:

Page1.html tem cor de fundo verde e apresenta apenas uma imagem alinhada à direita da página;

Page2.html tem cor de fundo branca e apresenta 1 link para page4.html, que será carregada na Janela3;

Page3.html tem cor de fundo amarela e apresenta 1 link para page5.html, que será aberta numa nova janela browser.

1.5 - RESPOSTAS DOS EXERCÍCIOS

1.5.1 - Resposta do exercício 1.4.1:

```
<HTML>
<HEAD>
<TITLE>Exercício HTML 1.4.1</TITLE>
</HEAD>
<BODY BGCOLOR="#00FF00">
<P ALIGN="right">
<A HREF="http://www.uninove.br"><IMG SRC="/home/httpd/html/apache_pb.gif"></A>
</P>
</BODY>
</HTML>
```

1.5.2 - Resposta do exercício 1.4.2:

```
<HTML>
<HEAD>
<TITLE>Exercício HTML 1.4.2</TITLE>
</HEAD>
<BODY BGCOLOR="red">
<CENTER>
<TABLE>
  <TR>
    <TD ROWSPAN="2" VALIGN="middle">
      <A HREF="page2.html" TARGET=_blank>
      <IMG SRC="/home/httpd/html/apache_pb.gif" BORDER="0"></A></TD>
    <TD>Célula 2</TD>
  </TR>
  <TR>
    <TD>Célula 3</TD>
  </TR>
</TABLE>
</CENTER>
</BODY>
</HTML>
```

1.5.3 - Resposta do exercício 1.4.3:

```
<HTML>
<HEAD>
<TITLE>Exercício HTML 1.4.3</TITLE>
</HEAD>
<BODY BGCOLOR="#0000FF">
<TABLE WIDTH="100%">
  <TR>
    <TD ALIGN="left">
      <IMG SRC="/home/httpd/html/apache_pb.gif"></TD>
    <TD ALIGN="right">
      <IMG SRC="/home/httpd/html/apache_pb.gif"></TD>
  </TR>
  <TR>
    <TD COLSPAN="2" ALIGN="center">
      <IMG SRC="/home/httpd/html/apache_pb.gif"></TD>
  </TR>
</TABLE>
</BODY>
</HTML>
```

1.5.4 - Resposta do exercício 1.4.4:

```

<HTML>
<HEAD>
<TITLE>Exercício HTML 1.4.4</TITLE>
<META HTTP-EQUIV="Refresh" CONTENT="10; URL=http://www.if.usp.br">
</HEAD>
<BODY>
<P ALIGN="center">
Mudamos de endereço, agora é http://www.if.usp.br<BR>
Se o seu browser não redirecionar em 10 segundos, clique <A HREF="http://www.if.usp.br">aqui</A>
</P>
</BODY>
</HTML>

```

1.5.5 - Resposta do exercício 1.4.5:

Documento base.html:

```

<HTML>
<HEAD>
<TITLE>Exercício HTML 1.4.5</TITLE>
</HEAD>
<FRAMESET ROWS="100,*" BORDER="10">
  <NOFRAME>
  Infelizmente seu browser não suporta frames !
  </NOFRAME>

  <FRAME SRC="Page1.html" NAME="Janela1" SCROLLING="yes" NORESIZE>

  <FRAMESET COLS="120,*">
    <FRAME SRC="Page2.html" NAME="Janela2" SCROLLING="yes" NORESIZE >
    <FRAME SRC="Page3.html"NAME="Janela3" SCROLLING="yes" NORESIZE >
  </FRAMESET>
</FRAMESET>
</HTML>

```

Documento Page1.html:

```

<HTML>
<HEAD>
<TITLE>Exercício HTML 1.4.5 - Page1</TITLE>
</HEAD>
<BODY BGCOLOR="#00FF00">
<P ALIGN="right">
<IMG SRC="imagem.gif">
</P>
</BODY>
</HTML>

```

Documento Page2.html:

```

<HTML>
<HEAD>
<TITLE>Exercício HTML 1.4.5 - Page2</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<A HREF="Page4.html" TARGET="Janela3">Link para Page4</A>
</BODY>
</HTML>

```

Documento Page3.html:

```
<HTML>  
<HEAD>  
<TITLE>Exercício HTML 1.4.5 - Page3</TITLE>  
</HEAD>  
<BODY BGCOLOR="#FFFF00">  
<A HREF="Page5.html" TARGET="_blank">Link para Page5</A>  
</BODY>  
</HTML>
```