

# **Projeto e Criação de Páginas de Web**

**HTML**

Prof.: João Sérgio dos Santos Assis

e-mail: [joao@nce.ufrj.br](mailto:joao@nce.ufrj.br)

Tel.: 598-3214

# Projeto e Criação de Páginas de Web

## Índice

1. Bibliografia auxiliar	1
2. Introdução: A rede Internet e o serviço WWW	2
3. Características gerais de HTML	13
4. Comandos básicos de formatação	15
5. Listas e enumerações	19
6. Interligando documentos	22
7. Imagens e cores	25
8. Manipulação de Imagens	31
9. Mapas Clicáveis	34
10. A tag <META>	37
11. Multimídia e animações	41
12. Tabelas	46
13. Frames	49
14. Formulários	55
15. Folhas de estilo	61
Apêndice A – Tabela de caracteres	73
Apêndice B – Manipulação de imagens	76

## 1. Bibliografia auxiliar

### Livros:

- Aprenda em uma semana HTML 4  
Laura Lemay  
Editora Campus
- Home pages: Recursos e técnicas para criação de páginas WWW  
Andreia Alcântara e etc.  
Editora Campus
- HTML Dinâmico  
Ramalho  
Editora Berkeley

### Internet:

- Em português  
  
<http://www.icmsc.sc.usp.br/manuals/HTML>
- Documentação da NCSA  
  
<http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html>
- Documentação do Netscape  
  
<http://developer.netscape.com/docs/manuals/htmlguid/index.htm>

## 2. Introdução: A rede Internet e o serviço WWW

### A Rede Internet

A idéia inicial da Internet surgiu em 1969 como uma rede descentralizada para interligar instalações militares (ARPANET). Durante a década de 70 várias outras redes foram sendo interligadas a ARPANET e em 1980 ela passa a se chamar Internet. No início da década de 80 são criadas redes de interconexão de instituições científicas (BITNET, CSNET, NSFNET). O ano de 1983 é o marco de criação da Internet, quando os militares abandonam o controle da ARPANET.

A Internet surgiu no Brasil em 1991, num primeiro momento interligando instituições acadêmicas por meio da Rede Nacional de Pesquisa. A partir de 1995 começa a utilização comercial e a popularização da rede.

Com o surgimento dos provedores de acesso, a Internet chega ao cidadão comum. A exposição e divulgação da rede através dos meios de comunicação de massa dão um grande impulso ao crescimento da rede.

### Principais Serviços da Internet

- Correio Eletrônico
- Listas de Discussão
- Newsgroups
- Conexões Remotas (Telnet)
- Transferência de Arquivos - FTP
- World Wide Web

## Correio Eletrônico

Este serviço permite que qualquer usuário da Internet possa enviar (ou receber) mensagens para (de) outro usuário. O endereço eletrônico de um usuário contém todas as informações necessárias para que a mensagem chegue ao seu destino. Ele é composto de uma parte que identifica o destinatário (*username*) e uma parte relacionada a sua localização, no formato:

*username@subdomínios.domínio*

*Por exemplo: joao@nce.ufrj.br*

Uma mensagem é composta de cabeçalho e corpo. O cabeçalho informa a data do envio da mensagem, o endereço do emitente, um título sobre o assunto, além de informações de controle. O corpo da mensagem é o seu conteúdo em si.

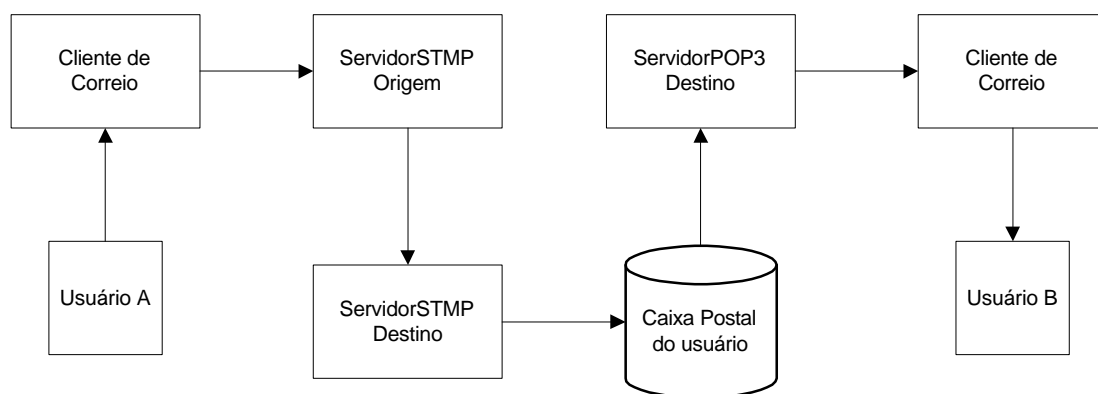
Embora a grande maioria das mensagens trocadas via rede sejam constituídas por informação puramente textual, também é possível enviar outros tipos de informação, tais como sons e imagens.

O envio de uma mensagem por um determinado usuário A para um usuário B através de correio eletrônico envolve vários programas:

- O usuário A prepara a sua mensagem num programa cliente de correio eletrônico e dá o comando de envio. Este cliente, por sua vez, manda a mensagem para um servidor de correio que reconheça o usuário A.
- Este servidor envia a mensagem para a máquina identificada pelo domínio do endereço eletrônico do usuário B.
- Na máquina destino, um outro servidor identifica o usuário e armazena a mensagem na sua caixa postal.
- Num determinado momento o usuário B executa o seu programa cliente de correio, que por sua vez consulta o servidor de correio que tem acesso à caixa postal do usuário. Caso haja alguma mensagem para o

usuário, o servidor envia para o programa cliente, que a exibe para o usuário.

Todo este procedimento envolve dois tipos de servidores: a troca de mensagens da máquina origem para a máquina destino é feita através de servidores SMTP; já o servidor que consulta a caixa postal de um usuário para saber se há alguma mensagem é um servidor POP3. Este caminho da mensagem é ilustrado pela figura abaixo:



Para ler mais a respeito do correio eletrônico:

<http://www.icmsc.sc.usp.br/manuals/BigDummy/Email.html>

## Listas de Discussão

São utilizadas para troca de mensagens entre grupos de usuários interessados em um determinado assunto. Com apenas um endereço eletrônico de destino é possível enviar mensagens para todos os participantes da lista.

A manutenção de uma lista pode ser feita de forma manual ou automática. Na forma automática há um programa que se encarrega de ler a caixa postal relativa ao endereço da lista e a reenviar as mensagens para todas as pessoas inscritas. Este programa também recebe, por meio de correio eletrônico, comandos para inclusão e exclusão de nomes da lista. Mesmo as listas automáticas contam com um administrador encarregado de resolver eventuais problemas que o programa de manutenção não consegue resolver.

As listas podem ser livres ou moderadas. No segundo caso, as mensagens não são imediatamente repassadas para os assinantes da lista. Antes disso elas devem ser aprovadas por uma pessoa chamada moderador que decide sobre a pertinência ou não do assunto da mensagem. Existe uma infinidade de listas de discussão, sobre os mais variados assuntos.

Para ler mais a respeito:

<http://www.icmsc.sc.usp.br/manuals/BigDummy/Listas.html>

## Newsgroups da Usenet

As listas de discussão tem um inconveniente de encher a caixa postal do participante com centenas de mensagens. Muitas vezes um usuário sai de férias e fica muito tempo sem ler sua caixa postal e ao voltar se depara com uma quantidade absurda de mensagens para ler. Uma outra forma de participar de listas de discussão é através dos grupos da usenet. Diferente das listas comuns, as mensagens para os grupos não são reenviadas para os participantes da lista. Ao invés disso são armazenadas em servidores espalhados pela rede. De tempos em tempos, as mensagens mais antigas são removidas de cada grupo. Um usuário pode a qualquer momento se conectar a um destes servidores e receber as mensagens dos grupos em que esteja inscrito.

Os grupos são organizados em uma estrutura hierárquica, explicitada pelo seu nome. Assim os newsgroups cujo nome começam pelo prefixo news servem para discutir assuntos relacionados aos newsgroups, os com prefixo soc agrupam listas de assuntos relacionados à sociedade e aqueles com o prefixo Comp assuntos relativos a computação ou computadores. Por Exemplo:

- news.announce.newusers – informações sobre a usenet para novos usuários.
- soc.culture.brasil – Grupo de discussão de assuntos brasileiros.
- comp.sources.games – Informação e disponibilização de código fonte de jogos de computador.

## Conexões Remotas (Telnet)

Sistema que permite que sua máquina possa ser um terminal de outra máquina na Internet. Para isso o usuário deve ter uma conta (*login*) na máquina remota. A utilidade principal do Telnet é a possibilidade de executar programas na máquina remota.

## Transferência de Arquivos - FTP

É o protocolo usado na Internet para transferência de arquivos entre computadores. O FTP é um dos recursos mais importantes disponíveis na Internet, e também responsável por um grande volume de tráfego de dados.

Para que se possa conectar a uma máquina remota através do FTP é necessário que esta máquina esteja executando um programa servidor de FTP. O FTP funciona da seguinte maneira: um usuário executa o programa FTP (um cliente FTP) em sua máquina (máquina local), cria uma conexão com outra máquina (com o servidor de FTP desta máquina), localizada em qualquer ponto da rede (máquina remota), e envia (ou recebe) arquivos para (de) esta máquina.

Ao se conectar a uma máquina remota, a pessoa que está utilizando o cliente FTP pode ser usuário desta máquina ou não. Caso seja usuário, deve se identificar através do nome de sua conta (*login*) e de sua senha (*password*). Após a validação destes dados, tem acesso aos seus arquivos e diretórios na máquina remota podendo: listar o conteúdo dos diretórios, mudar de diretório, enviar arquivos e recuperar arquivos.

Muitas máquinas oferecem áreas públicas que podem ser acessadas por não usuários. Para Ter acesso a esta área basta o usuário responder ao *login* com o nome ***anonymous*** e como *password* escrever o seu endereço de correio eletrônico. Caso a máquina permita esta forma de *login*, o usuário tem acesso à área pública podendo: listar o conteúdo dos diretórios, mudar de diretório e recuperar arquivos (normalmente, por questões de segurança, não é permitido enviar arquivos).



Existem muitos gigabytes de informações disponíveis em FTP anônimo nas máquinas da Internet ao redor do mundo. Nesses locais podem-se encontrar programas fontes, programas objetos, sistemas inteiros, documentos, ou seja, todo tipo de informação em arquivos de computador.

O clientes FTP mais antigos apresentavam ao usuário uma interface parecida com a do DOS, oferecendo vários comando, dentre os quais os mais comuns são:

- OPEN <endereço> - faz a conexão com a máquina identificada pelo <endereço> e pede o *login* e a *password*.
- CD <diretório> - muda de diretório na máquina remota.
- PWD – Informa o diretório atual na máquina remota.
- DIR – Lista o diretório atual da máquina remota.
- LCD <diretório> - muda de diretório na máquina local.
- !DIR – Lista o diretório atual da máquina local.
- PUT – Envia um arquivo da máquina local para a remota.
- GET – Envia um arquivo da máquina remota para a local.

Um bom site de ajuda à utilização do FTP é

<http://www.icmsc.sc.usp.br/manuals/BigDummy/ftp.html>

## World Wide Web

Foi desenvolvida no final da década de 80 numa instituição de pesquisa (CERN) como uma forma de facilitar aos pesquisadores acesso a documentos científicos. Os trabalhos de cada pesquisador, muitas vezes interrelacionados, estavam espalhados por várias máquinas diferentes. Através da Web é possível, partindo de um determinado ponto, pesquisar documentos

relacionados, independente de sua localização física (que o usuário não tem a mínima necessidade de conhecer).

Com a Web ganha força a utilização em larga escala da técnica de Hipertextos. Um Hipertexto é um documento onde é possível incluir referências (em inglês, *Links*) a outros documentos. A seleção de uma destas referências, leva o usuário ao documento referenciado.

Inicialmente a informação disponível na Web era encontrada principalmente na forma de texto. Aos poucos foram sendo incorporados elementos gráficos e animações aos documentos. Os novos recursos gráficos ajudaram muito a popularizar a Web, aproximando-a mais do cidadão comum.

A popularização da Web criou uma série de novas aplicações, fazendo com que ela seja utilizada hoje para: divulgação científica, trabalho cooperativo, divulgação de informações culturais, livro eletrônico, promoção de produtos e serviços, comércio eletrônico, suporte técnico e várias outros usos que vão sendo inventados a cada instante.

## Conceitos Básicos da Web

Uma característica importante da Web é que a informação é disponibilizada de forma independente da plataforma que será utilizada para a sua visualização. Para isso é necessário que os documentos sejam escritos utilizando um formato padronizado.

Esta padronização é obtida através de uma linguagem de marcação chamada HTML. O código HTML contém instruções de visualização de texto e informação de localização de outros documentos (*Links*). Para poder visualizar corretamente o documento o usuário deve ter um programa (navegador) capaz de interpretar esta linguagem, reconstituindo a informação segundo as instruções contidas no código.

Para obter algum documento na rede o usuário deve informar ao navegador um conjunto de informações sobre a sua localização na Internet:

protocolo, máquina onde se encontra o documento, diretório onde ele se encontra na máquina e nome do documento. Estes dados devem ser informados ao navegador segundo um formato padronizado que chamamos de URL.

A máquina que contém o documento a ser exibido deve estar executando um programa que se encarregada de receber o pedido do documento, localizá-lo no disco e enviá-lo para a máquina que o pediu. Este programa é chamado de Servidor de Web e o navegador que pede o documento é chamado de Cliente.

### Navegadores (Browsers)

Para navegar pela Web é necessário um software cliente que chamamos de navegador (em inglês, *Browser*). Ele é responsável por: solicitar documentos na Internet e interpreta-los, exibindo-os para o usuário. Exemplos de navegadores: Netscape, Internet Explorer, Mosaic, Lynx

### Servidores (Web Servers)

São programas sendo executados nas máquinas onde estão guardados os documentos. Estes programas são os responsáveis por enviar os documentos para as máquinas que os solicitam. Exemplos: NCSA, CERN, Apache (em ambiente UNIX), PWS, IIS, Netscape, WebSite (em ambiente Windows).

### HTML (Hypertext Markup Language)

É a linguagem padrão usada para a escrita de páginas da Web. O HTML é uma linguagem de marcação, ou seja, seus comandos (chamados *Tags*) servem para informar ao cliente os elementos que serão exibidos na página: cabeçalhos, textos em itálico, links, imagens, etc. O cliente Web interpreta estes comandos e exibe a página para o usuário. Um texto HTML não define a forma exata como o documento vai ser exibido. Isto depende do cliente e de definições feitas pelo usuário.

## URL (Uniform Resource Locator)

A URL é a convenção utilizada para indicar ao navegador a forma de localizar um endereço na rede. Uma URL obedece ao seguinte formato:

**protocolo://servidor /caminho/arquivo**

Onde:

Protocolo – Indica a forma como vai ser realizada a comunicação entre o servidor e o cliente e o tipo de serviço que será prestado. Os mais comuns são http (Hypertext Transfer Protocol - páginas comuns de hipertexto) e ftp (File Transfer Protocol - transferências de arquivos)..

Servidor – Endereço do servidor na Internet. Pode ser dado na forma nome\_da\_máquina.domínio (como em www.nce.ufrj.br) ou através do endereço IP da máquina (como em 146.164.2.68).

Caminho – Localização do arquivo no disco do servidor através de uma lista de diretórios.

Arquivo - Nome do arquivo desejado. Esta informação pode ser omitida e o servidor assume um nome padrão. Este nome pode variar de instalação para instalação mas normalmente é home.html ou index.html.

## Home Pages

Documento projetado para ser a página principal de um *Site*. Funciona como a porta de entrada e deve conter um ou mais Links para as demais páginas do *Site* ou para outros *Sites* relacionados.

## Como projetar bons documentos ?

Passos a seguir no processo de criação:

- 1) Avaliação do público alvo
- 2) Definição do conteúdo
- 3) Organização do conteúdo a estrutura pode ser linear ou hierárquica.
- 4) Redação do conteúdo
- 5) Programação Visual e Implementação

Boas práticas na confecção de um documento

Portabilidade – Os documentos devem ser projetados de forma que possam ter (mais ou menos) a mesma aparência na maior variedade possível de navegadores. Na prática leva-se em conta principalmente o Internet Explorer e o Netscape que monopolizam o mercado de browsers.

Imagens – Deve-se levar em conta que os meios de transmissão ainda são lentos para a maioria dos usuários. Não se deve abusar de muitas imagens de tamanho muito grande pois o usuário pode acabar desistindo de carregar a página se ela demora muito para vir.

Links – Deve-se sempre verificar se os links de sua página ainda são válidos. A rede é muito dinâmica e por vezes páginas desaparecem ou são mudadas de lugar.

URLs relativas X absolutas – Ao fazer referências a páginas do mesmo *site* convém utilizar URLs relativas ao invés de absolutas. Isto facilita quando é necessário mudar todo o site de um lugar para o outro. Além disso muitas vezes utiliza-se uma máquina para desenvolvimento do site e outra para publicação.

O título do documento – Os documentos HTML contém um título que, normalmente, não é visualizado no interior da página. Este título porém é utilizado pelos navegadores para nomear a página no arquivo de bookmarks. É

interessante portanto criar um título relativamente pequeno e suficientemente descritivo.

Tipografia – Os navegadores costumam oferecer ao usuário um conjunto de tipos de caracteres que podem ser visualizados. Os tipos mais comuns (Times News Roman, Arial, Courier) costumam estar disponíveis em todas as instalações. Porém, por vezes, pode ser interessante incluir textos utilizando tipos de caracteres mais sofisticados. Neste caso, deve-se utilizar um programa de edição de imagens e gerar um arquivo no formato gif ou jpeg com o texto e depois incluí-lo na página na forma de imagem.

Cores – Cores devem ser escolhidas com cuidado de forma a não dificultar a visualização das informações contidas na página.

## Editores de HTML

Atualmente existe uma grande quantidade de ferramentas para auxiliar o desenvolvimento de páginas de Web. Há vários níveis de ferramentas desde aquelas destinadas a converter documentos de um determinado formato para HTML, passando por editores visuais simples até ferramentas complexas de gerenciamento de sites. Exemplos:

- Filtros conversores: rtfhtml, psthtml
- Editores de tags: HTML Assistant, HotDog, W3e
- Editores Wysiwyg: Netscape Composer, MS Internet Assistant, MS FrontPage Express
- Gerenciadores de site: MS FrontPage, AOLPress, Macromedia Dreamweaver

Estas ferramentas porém não eliminam a necessidade de conhecer a linguagem HTML. Mesmo as ferramentas mais complexas mantêm a facilidade de visualizar e editar diretamente o código HTML. Muitas vezes esta é a forma mais fácil de se obter o efeito desejado na página.

### 3. Características gerais de HTML

Um documento escrito em HTML é um arquivo ASCII comum, contendo apenas caracteres visíveis. O navegador ignora qualquer caracter especial, inclusive aqueles que sugerem algum tipo de formatação ao texto (como TAB, CR, LF). Qualquer tipo de formatação deve ser informada através das Tags. As Tags se diferenciam do texto comum por estarem contidas entre o caracter < e o caracter >. Algumas tags contem atributos que permitem que configurar algumas características. Estes atributos são colocados entre os delimitadores (< e >), após o nome da tag.

#### Tipos de Tags

Há dois tipos de tags:

Container Tags - Servem para definir um efeito sobre um trecho do documento. Estas tags vêm sempre aos pares: uma tag indica o início do trecho e uma outra tag derivada (igual à inicial exceto por conter o caracter /) indica o fim. Todo o texto escrito entre as duas tags é sofre o efeito indicado por elas.

Por exemplo, para indicar que uma parte do texto deve ser exibida em negrito utilizo o para de tags <B> e </B>. O seguinte trecho HTML:

Uma palavra em <B>negrito</B> fica realçada

Seria exibida da seguinte forma:

Uma palavra em **negrito** fica realçada.

As tags container podem ser colocadas umas dentro das outras. O texto contido na tag mais interior, sofre o efeito cumulativo de todas as tags que o contêm.

Empty Tags – São tags que produzem efeitos locais, normalmente introduzindo algum elemento no texto, e, portanto, não precisam de uma tag

finalizadora. Um exemplo é a tag <BR> que insere no texto uma mudança de linha.

A linguagem HTML não faz diferenciação entre letras maiúsculas e letras minúsculas, ou seja, escrever <BR> ou <br> tem o mesmo significado. Como em outras linguagens, é possível inserir comentários num texto HTML. Todo texto incluído entre <!-- e --> é ignorado pelo *Browser*.

## Estrutura de um documento HTML

Todo documento HTML tem a seguinte estrutura:

```
<HTML>
  < HEAD>
    <TITLE> ... </TITLE>
  </HEAD>
  <BODY>
    ...
  </BODY>
</HTML>
```

A tag HTML indica a área onde deve estar contido o documento HTML. Isto não quer dizer que o Browser não vai exibir um texto colocado fora desta tag, mas não é uma boa prática colocar texto fora desta área.

A tag HEAD é o cabeçalho do documento. Nesta área são colocadas tags com informações relativas ao documento. A mais importante destas informações é contida na tag TITLE que deve sempre ser incluída em todas as páginas. O texto do TITLE é utilizado pelo navegador para nomear os links adicionados ao arquivo de bookmarks. Normalmente esta informação é visualizada na barra de títulos da janela do browser. A tag BODY contém o documento propriamente dito. Nesta área deve ser colocada a página a ser visualizada.



#### 4. Comandos Básicos de Formatação

A linguagem HTML possui duas classes de elementos utilizados para modificar o estilo de apresentação de partes do texto: tags físicas e tags lógicas.

As tags físicas indicam explicitamente a forma como o autor deseja ver exibido o seu texto. São elas:

Elemento	Descrição	Efeito
<B>...</B>	Negrito	<b>texto</b> normal
<i>...</i>	Itálico	<i>texto</i> normal
<U>...</U>	Sublinhado	<u>texto</u> normal
<TT>...</TT>	Letras igualmente espaçadas	texto normal
<SUB> ... </SUB>	Subescrito	texto normal
<SUP> ... </SUP>	Sobrescrito	texto normal
<STRIKE> ... </STRIKE>	Riscado	<del>texto</del> normal
<BIG> ... </BIG>	Fonte grande	texto normal
<SMALL> ... </SMALL>	Fonte pequena	texto normal

As tags lógicas expressam uma idéia que deve ser passada ao usuário e a forma como o texto será exibido é dependente do navegador. Algumas delas:

Elemento	Descrição	Explorer e Netscape
<STRONG>...</STRONG>	Texto forte	Negrito
<EM>...</EM>	Texto enfatizado	Itálico
<CITE>...</CITE>	Citação	Itálico
<CODE> ... </CODE>	Código de programa	Monoespaçado
<ADRESS> ... </ADDRESS>	Endereço	Itálico

## Separadores

Como foi dito anteriormente, o navegador ignora os caracteres de mudança de linha existentes no arquivo HTML. Quando é necessário incluir uma mudança de linha num documento deve-se incluir uma das seguintes tags:

Elemento	Descrição	Atributos
<P> <sup>1</sup>	Início de parágrafo	ALIGN
 	Quebra de linha	Nenhum
<HR>	Linha horizontal	SIZE, WIDTH, ALIGN e NOSHADE

### Atributos de <HR>

O atributo SIZE é utilizado para definir a espessura da linha em número de pontos. O atributo WIDTH serve para definir a largura da linha. Esta largura pode ser especificada em número de pontos ou com o percentual da janela que será ocupado pela linha. A simples inclusão do atributo NOSHADE faz com que a linha não seja desenhada com efeito sombreado. O atributo ALIGN será descrito mais adiante na seção Alinhamento de Texto. O exemplo abaixo inclui uma linha de 4 pontos de espessura, ocupando 50% da janela e sem sombreado:

```
<HR SIZE=2 WIDTH=50% NOSHADE>
```

## Cabeçalhos

Ao se redigir um documento é conveniente organizá-lo de forma clara atribuindo títulos e subtítulos às suas diversas partes. A linguagem HTML oferece um conjunto de 6 cabeçalhos pré-definidos que podem ser incluídos no

---

<sup>1</sup> Em versões antigas do HTML esta era uma tag do tipo container. Atualmente não é mais necessário colocar a tag de fim de parágrafo (</P>), embora alguns editores insistam colocar. Isto não causa problema algum pois os navegadores ignoram o </P>.

documento através da tag <Hn> ... </Hn>, onde n pode ser um número de 1 a 6. Os cabeçalhos são em negrito, centralizados e seu tamanho varia do maior H1 até o menor H6.

## Texto Pré-formatado

Muitas vezes é interessante fazer com que o navegador reproduza exatamente a formatação do texto escrita no arquivo HTML, sem ignorar espaços, mudanças de linha, tabulações e utilizando um fonte não proporcional para exibir o texto.

Um exemplo típico disso é quando está se fazendo uma página sobre alguma linguagem de programação e se deseja incluir um trecho do código de um programa como exemplo. Desta maneira o programa é exibido respeitando a indentação.

Para incluir um texto pré-formatado utiliza-se a tag <PRE> ... </PRE>. Esta tag, porém, não impede o navegado de interpretar outras tags que estejam em seu interior, permitindo realizar mudanças de fonte, estilo e cor do texto.

## Alinhamento de Texto

Os textos contidos nos arquivos HTML são exibidos, salvo indicação em contrário, alinhados à esquerda da janela. Para ter o texto alinhado de forma diferente é necessário modificar o atributo ALIGN existente em algumas tags (como cabeçalhos, <P> e <HR>). Este atributo pode assumir os seguintes valores: RIGHT (direita), CENTER (centro) ou LEFT (esquerda). O alinhamento centralizado também pode ser obtido através da tag <CENTER> ... </CENTER>. A tag <DIV> ... </DIV> permite definir o alinhamento default para tudo que ela contiver. Exemplos:

```
<P align=right>  
<H1 align=center>  
<center> texto centralizado </center>  
<DIV align=left> texto alinhado à esquerda </DIV>
```

## Acentuação

Não existe uma padronização universal para definir os códigos associados aos caracteres acentuados. Quando produzimos uma página no ambiente Windows, utilizando diretamente os caracteres do Windows esta página será visualizada sem problemas em grande parte das máquinas que utilizam este mesmo sistema (se o sistema estiver configurado para utilizar a norma ISO Latin 1), mas isso não será verdade para qualquer ambiente.

É possível que um texto cheio de caracteres acentuados que aparece perfeito na tela de quem o produziu apareça cheio de caracteres estranhos na tela de alguém que esteja trabalhando em um ambiente diferente. Para garantir a portabilidade dos documentos nas mais diferentes plataformas o HTML prevê uma série de códigos que devem ser utilizados no lugar de caracteres acentuados e outros caracteres especiais:

Codificação em HTML			
á &aacute;	é &eacute;	Ç &ccedil;	& &amp;
ã &atilde;	ê &ecirc;	Ç &Ccedil;	“ &quot;
â &acirc;	ó &oacute;	< &lt;	® &reg;
à &agrave;	ü &uuml;	> &gt;	© &copy;

A tabela completa pode ser encontrada no apêndice A das notas de aula.

## 5. Listas e enumerações

Uma forma muito comum de organizar a informação é através de listas. A linguagem HTML oferece 3 formas diferentes de criar uma lista: listas não numeradas, lista numeradas e listas do tipo verbete de dicionário.

### Listas não numeradas

As listas não numeradas são formadas por itens precedidos de um símbolo gráfico. Em HTML utiliza-se a tag `<UL> ... </UL>` para delimitar a lista., sendo que cada item é indicado pela tag `<LI>`<sup>2</sup>. Tanto a tag `<UL>` como a tag `<LI>` tem o atributo `TYPE`. Este atributo indica qual o símbolo é colocado antes de cada item: **disc** (um círculo cheio, é o default), **circle** (um círculo vazado) e **square** (um quadrado cheio).

Por exemplo, o código HTML:

```
<UL>
<LI>Correio Eletrônico
<LI>Telnet
<LI>FTP
</UL>
```

Produz uma saída parecida com:

- Correio Eletrônico
- Telnet
- FTP

### Listas numeradas

As listas numeradas são formadas por itens precedidos de um número indicando a sua ordem. Em HTML utiliza-se a tag `<OL> ... </OL>` para delimitar a lista numerada., sendo que cada item é indicado pela tag `<LI>`. Tanto a tag

---

<sup>2</sup> Esta tag também era do tipo container nas versões mais antigas do HTML.

<OL> como a tag <LI> tem o atributo TYPE, que no caso deste tipo de lista indica qual a forma de numeração será usada em cada item: 1 (números arábicos, é o default), A (letras maiúsculas), a (letras minúsculas), I (números romanos com letras maiúsculas) e i (números romanos com letras minúsculas).

A tag <OL> tem também o atributo START, que indica qual será o primeiro número da lista. A numeração também pode ser modificada através do valor do atributo VALUE da tag <LI>

Por exemplo, o código HTML:

```
<OL TYPE=A>
<LI> Correio Eletrônico
<LI VALUE=4> Telnet
<LI> FTP
</OL>
```

Produz uma saída parecida com:

- A. Correio Eletrônico
- 1. Telnet
- 2. FTP

### **Listas de definições**

As listas de definição (ou verbete de dicionário) consistem de uma lista de termos, seguido de um parágrafo deslocado contendo sua descrição. Em HTML utiliza-se a tag <DL> ... </DL> para delimitar a lista de definições. Cada termo a ser definido é indicado pela tag <DT> ... </DT> e a sua definição é indicada pela tag <DD> ... </DD><sup>3</sup>.

Uma possível aplicação para as listas de definição é utilizar as tags <DL> ... </DL> e <DD> para criar listas indentadas que não são precedidas pelos

---

<sup>3</sup> <DT> e <DD> também podem ser usadas como Empty Tag nas versões mais novas de HTML.

símbolos gráficos padrão ou por numeração. O autor do documento pode então criar os símbolos que deseja por no início de cada item utilizando a tag que inclui imagens no documento (o que será visto mais adiante).

Por exemplo, o código HTML:

```
<DL>
<DT>Telnet</DT>
<DD>É o protocolo mais usado na Internet para criar uma
conexão com um nó remoto.</DD>
<DT>FTP</DT>
<DD>O "File Transfer Protocol" é o principal método de se
transferir arquivos pela Internet. </DD>
</DL>
```

Produz uma saída parecida com:

```
Telnet
    É o protocolo mais usado na Internet para criar uma
    conexão com um nó remoto.
FTP
    O "File Transfer Protocol" é o principal método de se
    transferir arquivos pela Internet.
```

## Resumo das Tags

Elemento	Descrição	Atributos
<UL> ... </UL>	Lista não ordenada	TYPE
<LI>	Item de lista não ordenada	TYPE
<OL> ... </OL>	Lista ordenada	TYPE, START
<LI>	Item de lista ordenada	TYPE, VALUE
<DL> ... </DL>	Lista de definições	nenhum
<DT>	Termo a ser definido	nenhum
<DD>	Definição do Termo	nenhum

## 6. Interligando documentos

Uma das principais características do hipertexto é a possibilidade de incluir no documento referências, cuja seleção leva a exibição do documento referenciado. As referências (*Links*) podem ser feitas a documentos na mesma máquina, em uma máquina completamente diferente (que pode estar até do outro lado do mundo) ou a um outro ponto do próprio documento.

Ao se fazer uma referência a um outro documento, é necessário indicar a sua URL. Caso o documento referenciado esteja na mesma máquina é possível (e recomendável) utilizar uma URL relativa (utilizando o caminho de diretórios para chegar ao novo documento a partir do atual) ao invés da URL absoluta (endereço completo na Internet do novo documento). Por exemplo: se na página [equipe.nce.ufrj.br/joao/Programacao](http://equipe.nce.ufrj.br/joao/Programacao) há uma referência a um arquivo que se encontra no endereço [equipe.nce.ufrj.br/joao/Exemplos/arq.html](http://equipe.nce.ufrj.br/joao/Exemplos/arq.html) não é necessário escrever a URL completa. Basta escrever o caminho relativo até ela: `../Exemplos/arq.html`<sup>4</sup> (a partir do diretório atual, vá ao diretório pai e de lá a um diretório filho chamado Exemplos, onde há o arquivo arq.html).

Referência a outro documento:

A forma de inserir referência em arquivo HTML é através da tag `<A> ... </A>` e de seus atributos. O atributo `HREF` serve para definir a URL que será aberta se o usuário selecionar com o mouse o texto contido entre o `<A>` e o `</A>`. Os navegadores costumam exibir este texto utilizando caracteres sublinhados e o cursor do mouse é modificado ao passar por cima dele. No exemplo a seguir, a seleção com o mouse do texto “Página do NCE” faz com que o navegador abra a página “<http://www.nce.ufrj.br>”:

```
<A HREF="http://www.nce.ufrj.br">Página do NCE</A>
```

---

<sup>4</sup> Os caminhos de diretórios seguem a convenção do Unix utilizando o caracter / ao invés do caracter \.



Uma observação importante a fazer é que, apesar da HTML não diferenciar maiúsculas de minúsculas, o mesmo não é válido para as URLs. É necessário indicar corretamente quais letras estão em minúsculas e quais estão em maiúsculas para que o arquivo possa ser encontrado.

O arquivo destino não precisa necessariamente ser um documento HTML, pode ser um arquivo de qualquer tipo: imagens, música, arquivos zipados, etc. Caso o navegador não saiba como exibir este arquivo, ele permite ao usuário receber o arquivo e salvá-lo em um diretório de sua escolha.

## Referências à própria página

Como foi dito anteriormente, em um arquivo HTML é possível fazer uma referência a uma outra parte do mesmo arquivo. Neste caso é necessário indicar ao navegador o ponto exato que será referenciado. Isto é feito colocando-se uma “âncora” neste ponto através da tag `<A> ... </A>` com a definição do atributo NAME. O código HTML do exemplo abaixo associa a um determinado ponto do documento o nome “inicio”:

```
<A NAME="inicio"></A>
```

Uma âncora colocada no meio do texto não tem nenhum efeito na visualização da página. O código HTML abaixo permite incluir uma referência ao ponto do documento marcado pela âncora acima:

```
<A HREF="#inicio">Início da Página</A>
```

As âncoras tem duas utilidades básicas: A primeira é permitir a criação de índices no início da página. Através da seleção do tópico de interesse no índice, o usuário é diretamente remetido a este tópico, sem precisar ficar avançando página a página a procura do item desejado. A segunda utilidade é permitir ao usuário voltar imediatamente ao início da página.

É possível fazer referência na URL a uma âncora de uma página:

```
<A HREF="http://www.desy.de/gna/interpedia/greek_myth/olympian.html#Athena">Atenas</A>
```

## Enviando mensagens de correio eletrônico

Além de criar links para a exibição de arquivos, a tag `<A>...</A>` serve para permitir que o usuário execute outros serviços da web: ftp, telnet, mail, news. No exemplo a seguir, quando o link for acionado, o navegador abre o programa de correio eletrônico, permitindo que o usuário envie uma mensagem para o endereço especificado (joao@nce.ufrj.br).

```
<A HREF="mailto:joao@nce.ufrj.br">Mande um mail</A>
```

## 7. Imagens e cores

Um dos comandos mais importantes de HTML é a tag <IMG>, através da qual é possível incluir arquivos de imagem no documento. O atributo SRC deve sempre estar presente para indicar, através de sua URL, qual arquivo contém a imagem (nada impede que a imagem esteja em outra máquina, mas isto não é conveniente). Arquivos de imagem são do tipo GIF ou do tipo JPEG. O exemplo abaixo faz com que a imagem eyesdown.gif seja exibida na junto com o texto, após a palavra “imagem” e antes de “no meio do texto”:

Imagem <IMG SRC="eyesdown.gif"> no meio do texto

Atributos de <IMG>	
Atributos	Descrição
SRC	URL do arquivo de imagem que será visualizado
ALT	texto alternativo mostrado no lugar da imagem
ALIGN	TOP, BOTTOM, MIDDLE, CENTER, LEFT, RIGHT
BORDER	largura da borda quando usada como âncora
WIDTH	largura em pixels
HEIGHT	altura em pixels
VSPACE	espaço vertical ao redor da imagem em pixels
HSPACE	espaço horizontal ao redor da imagem em pixels

O atributo ALT serve para associar à imagem uma descrição, que é mostrada pelo navegador em duas ocasiões: quando o usuário deixa o cursor do mouse sobre a imagem e enquanto a página está sendo carregada mas a imagem ainda não começou a ser carregada. Neste segundo caso o usuário pode decidir se deseja ou não carregar a imagem. Além disso, caso ela não

tenha sido carregada por algum motivo, o usuário tem como saber que naquele ponto da página havia uma imagem e o que ela retratava.

O atributo `ALIGN` define o alinhamento do texto próximo a tag em relação a imagem. Se este atributo receber os valores `LEFT` ou `RIGHT` a imagem é posicionada num canto da janela (esquerdo ou direito respectivamente) e o texto seguinte a tag é exibido ao lado da figura. As várias linhas do texto vão sendo posicionadas pela lateral da imagem, envolvendo-a, até que a lateral esteja toda tomada.

No caso do alinhamento definido como `TOP`, `BOTTOM`, `MIDDLE` ou `CENTER`, o texto que completar a linha é alinhado respectivamente pelo topo, base e meio da figura, mas apenas uma linha é colocada ao lado da imagem. As linhas seguintes do texto são posicionadas abaixo da imagem.

Muitas vezes é interessante permitir ao usuário selecionar um link através de uma imagem ao invés de um texto. Para que isso seja possível basta colocar a tag `<IMG>` dentro da tag `<A> ... </A>`, como é mostrado no exemplo a seguir:

```
<A HREF="link.htm"><IMG SRC="x.gif"></A>
```

Neste contexto que se situa o atributo `BORDER` da tag `<IMG>`. A borda da tag está para a imagem assim como o sublinhado está para um link textual. O valor default da borda é 1 e ela muda de cor conforme o estado do link (não visitado, ativo ou já visitado). Este efeito é muitas vezes indesejável e a forma de sumir com esta borda é incluir o atributo `BORDER=0` ou simplesmente `BORDER`.

Os atributos `WIDTH` e `HEIGHT` servem para indicar a largura e a altura da imagem para o navegador. Caso eles estejam presentes o navegador pode posicionar os elementos da página antes de carregar a imagem. Se os valores não corresponderem aos valores reais da imagem, esta é redimensionada para caber no espaço definido para ela.

Os atributos HSPACE e VSPACE servem para definir o espaçamento horizontal e vertical (em pixels) do texto em relação a figura, evitando que o texto fique muito próximo a sua borda.

### Definição de Cores

Há duas formas de definir cores em HTML: através de seu nome ou através dos valores RGB da cor. Para definir uma cor através do nome é preciso saber seu nome em inglês. A definição pelo código RGB é feita indicando a quantidade de vermelho (Red), verde (Green) e azul (Blue) necessária para compor a cor. Cada uma destas quantidades é indicada por um número entre 0 e 255 e o formato utilizado é #RRGGBB, com 2 dígitos para o R, 2 para o G e 2 para o B. Para que os valores possam caber em dois dígitos eles são escritos na base 16. Para saber mais sobre a base 16 veja o apêndice A. A tabela abaixo mostra algumas cores e seus valores RGB:

Nome	definição	R	G	B
White	#FFFFFF	255	255	255
Red	#FF0000	255	0	0
Green	#00FF00	0	255	0
Coral	#FF7F00	255	127	0
Yellow	#FFFF00	255	255	0
Gray	#C0C0C0	192	192	192

A seguinte página contém a tabela Completa:

<http://equipe.nce.ufrj.br/joao/programacao/privado1/exemplos/cores/cor.html>

## Imagem de fundo da página

A tag `<BODY> ... </BODY>` tem uma série de atributos que permitem definir características gerais do documento HTML: imagem de fundo da página (atributo `BACKGROUND`), cor de fundo da página (atributo `BGCOLOR`), cor do texto (atributo `TEXT`), cor de links ainda não visitados (atributo `LINK`), cor de links enquanto selecionados (atributo `ALINK`) e cor de links já visitados (atributo `VLINK`).

Atributos de <code>&lt;BODY&gt;</code>	
Atributos	Descrição
<code>BACKGROUND</code>	Imagem de fundo
<code>BGCOLOR</code>	Cor de fundo
<code>TEXT</code>	Cor do texto
<code>LINK</code>	Link não visitado
<code>ALINK</code>	Link sendo visitado
<code>VLINK</code>	Link já visitado

O atributo `BACKGROUND` define uma imagem que será utilizada como ladrilho de fundo da página. Ao criar esta imagem deve-se ter o cuidado de evitar que a junção dos lados dos ladrilhos fique nítida. Imagens em que os quatro lados são de uma única cor (sem variações de tonalidade) não apresentam o problema de bordar nítidas

Se o atributo `BACKGROUND` estiver definido a cor de fundo (`BGCOLOR`) só é visível até que a imagem de fundo seja carregada (o que as vezes pode demorar um pouco). No exemplo abaixo a definição de `fundo.gif` como imagem de fundo e da cor azul para o texto da página:

```
<BODY BACKGROUND="fundo.gif" TEXT="blue">
```

A tag <BODY> tem alguns atributos adicionais não muito bem documentados que permitem controlar o tamanho das margens do documento. Não existe compatibilidade entre os dois principais navegadores (Netscape e Explorer) quanto a estes atributos.

Atributos Netscape	Atributos Explorer	Descrição
marginHeight	topMargin	Margem superior
marginWidth	leftMargin	Margem esquerda

No exemplo a seguir mostra-se como eliminar a distância entre o conteúdo da página e as bordas da janela tanto para o Netscape quanto para o explorer:

```
<BODY topmargin=0 leftmargin=0 marginheight=0 marginwidth=0>
```

### Definindo a fonte e a cor dos Caracteres

A tag <FONT> ... </FONT> permite modificar o tamanho, cor e tipo de fonte de partes do documento (aqueles contidos dentro da tag).

Atributos de <FONT>	
Atributos	Descrição
SIZE	Tamanho do fonte
FACE	Tipo de fonte e alternativas
COLOR	Cor do fonte

O atributo SIZE permite definir o tamanho do FONT, que pode ser de 1 a 7, sendo que o tamanho default é 3. Ao se atribuir ao SIZE um valor precedido de + ou -, este valor tem um significado relativo ao tamanho atual.

O atributo FACE serve para definir tipo do fonte a ser utilizado e várias alternativas para o caso do primeiro fonte na estar disponível. Cada um dos tipos de fonte é separado do outro por vírgulas.

O atributo COLOR permite definir a cor em que será escrito o texto. No exemplo abaixo o texto “tudo verde” vai ser escrito em tamanho 4, na cor verde e com o fonte Arial:

```
<FONT SIZE=4 FACE="Arial" COLOR=green>tudo verde</FONT>
```



## 8. Manipulação de Imagens

### Quando usar GIF e quando usar JPEG

GIF é abreviação de “Graphics Interchange Format” e foi desenvolvido pela CompuServe. Neste formato as imagens são comprimidas através da codificação LZW (Lempel-Ziv and Welsh) e armazenadas em arquivos de extensão .gif. Os formatos GIF mais recentes são os GIF87 e GIF89a.

JPEG é um padrão internacional, proposto pelo comitê ISO “Joint Photographers Expert Group”. As imagens são codificadas por transformações matemáticas, o que causa a remoção de informações gráficas que o olho humano não consegue perceber, o que permite grande compressão, sem perda de qualidade perceptível. Este formato permite vários níveis de compressão o que permite escolher a melhor relação entre tamanho do arquivo e qualidade da imagem. Arquivos de imagens que seguem este formato têm extensão .jpeg ou .jpg.

A decisão a respeito de qual formato utilizar depende basicamente do tipo da imagem. Para alguns tipos de imagens, o formato GIF é superior em qualidade de imagem, tamanho de arquivo ou ambos; Para outros tipos é melhor utilizar o JPEG.

De maneira geral, JPEG é melhor aplicado a imagens com grandes variações de cor (fotografias digitalizadas, por exemplo) ou imagens com grandes variações de tonalidade. Quanto mais complexa a imagem, melhor o resultado com JPEG.

GIF é melhor para imagens com cores lisas, que utilizam até 256 cores. Para este tipo de imagem realiza uma compressão muito melhor do que o JPEG. Nestes casos, JPEG produz imagens desfocadas. Por isso, imagens em preto-e-branco puros (sem tons de cinza) não devem ser processados em JPEG.

## GIFs Entrelaçadas

O formato GIF permite uma variação na forma de carregamento do arquivo pelo navegador. Ao gerar um arquivo no formato GIF em um programa de manipulação de imagem deve-se indicar a forma de carregamento normal ou entrelaçada. As imagens entrelaçadas são carregadas em quatro passos pelo navegador. Após o primeiro passo já é possível visualizar toda a figura bastante desfocada. Muitas vezes esta primeira visualização já permite ao usuário decidir se deseja continuar a carregar a figura ou não, poupando tempo de transmissão. Os passos seguintes vão melhorando progressivamente a definição da imagem até a sua forma final.

## Imagens Transparentes

Quem já navegou pela web já deve ter encontrado sites com imagens que se misturam com o fundo da página, imagens que dão a impressão de não Ter a forma retangular dos arquivos de imagens comuns. Este efeito é obtido através da utilização do formato GIF89a.

Ao criar um arquivo GIF89a é possível indicar uma cor (alguns programas permite mais de uma) como sendo a cor de fundo da imagem. Quando encontra uma imagem deste tipo no documento, o navegador simplesmente não pinta as partes da imagem que tem a cor de fundo, mantendo a cor de fundo da página. Há vários programas no ambiente windows que permitem a criação GIFs transparente (por exemplo, Lview, PhotoShop, etc).

No apêndice B das notas de aula é descrito como criar imagens com fundo transparente no Lview e no Photoshop.

## Imagens para Web

A rede esta cheia de figuras que podem ser aproveitadas na confecção de uma página. Antes porém de utilizar uma imagem retirada da Web convém verificar se quem a publicou permite a sua livre utilização ou não. Há dezenas de site que oferecem imagens de domínio público, por exemplo:

- <http://www.sct.gu.edu.au/~anthony/icons/index.html>
- <http://www.iconbazaar.com>
- <http://www.gifworks.com>
- <http://www.aaaclipart.com>
- <http://daniweb.com/graphics>
- <http://www.clipart.com>

## 9. Mapas Clicáveis

Anteriormente vimos como é possível ativar um link através da seleção de uma imagem. Em muitas situações, porém, pode ser interessante associar vários links a uma imagem de forma que dependendo de qual parte dela foi selecionada o navegador exiba um documento diferente. O exemplo típico é quando temos uma figura com um mapa de um país e queremos que a seleção de um estado específico leve a uma página descrevendo este estado. A este tipo de facilidade da linguagem HTML chamamos de mapas clicáveis

### Server Side Maps

A primeira forma de implementar mapas clicáveis distribui o trabalho de interpretação de qual documento deve ser carregado, em função da seleção de um ponto da imagem, entre o cliente e o servidor.

Neste caso, o cliente deve ser capaz de poder determinar o ponto exato da figura que foi selecionado e enviar esta coordenada para um programa instalado no servidor. Este programa, por sua vez, determina qual URL está associada à aquele ponto e a envia para o cliente, que então finalmente carrega a página. No servidor há um arquivo associado a cada mapa clicável, onde estão definidas cada uma das regiões do mapa e a URL que deve ser ativada caso o ponto esteja dentro desta região. Em geral, é possível definir 3 tipos de região: circulares, retangulares e poligonais.

Esta forma de selecionar as regiões de uma imagem está caindo em desuso pois apresenta várias desvantagens. A primeira delas é que o formato deste arquivo de definição das áreas é dependente do tipo de servidor. Ao mudar uma página de um servidor para outro é necessário converter estes arquivos. Outra desvantagem é que costuma ser necessário pedir a intervenção do administrador do servidor para atualizar o arquivo. Além disso, a cada seleção é necessário uma consulta ao servidor para determinar o documento que foi selecionado (o que implica num maior tempo de resposta).

Por fim, o usuário não tem nenhuma indicação de qual URL ele está selecionando pois esta informação está no servidor.

## Client Side Maps

Neste tipo de mapa clicável, a definição das regiões de uma figura é feita no próprio arquivo HTML. É preciso que o navegador seja capaz tanto de saber determinar qual ponto da figura foi selecionado quanto descobrir em qual área está contido este ponto (o Netscape a partir da versão 2.0 e o Explorer a partir da versão 3.0 já implementam isso).

Para definir as diferentes regiões da figura e associá-las a diversas URLs são necessárias duas tags: a tag `<MAP> ... </MAP>` e a tag `<AREA>`.

A tag `<MAP> ... </MAP>` contém as tags que farão a definição de cada uma das áreas. Esta tag contém apenas o atributo `NAME` que serve para definir um nome que será utilizado para associar as regiões a um determinado arquivo de imagem (incluído através da tag `<IMG>`).

Uma ou mais tags `<AREA>` são colocadas dentro da tag `<MAP> ... </MAP>` para definir cada uma das áreas de uma figura. Há três tipos de áreas: circulares, retangulares e poligonais. O valor do atributo `SHAPE` especifica a forma da área e pode receber os seguintes valores: `CIRCLE`, `RECT` (valor default) e `POLYGON`. O atributo `COORDS` fornece um conjunto de coordenadas para definir a área. O significado e número de coordenadas varia de acordo com a forma escolhida, assim:

- `CIRCLE` – requer três valores `x`, `y` e `r`; onde `(x, y)` é o centro do círculo e `r` o raio. No exemplo abaixo é definida uma área circular de 20 pontos de raio, posicionada na coordenada `(100, 50)` da figura:

```
<AREA SHAPE=CIRCLE COORDS="100, 50, 20">
```

- `RECT` – requer quatro valores `x1`, `y1`, `x2`, `y2`; onde `(x1, y1)` é o vértice superior esquerdo do retângulo e `(x2, y2)` é o vértice inferior direito. No

exemplo abaixo é definido um retângulo de altura 50 e largura 100, posicionado na coordenada (30, 60) da figura:

```
<AREA SHAPE=RECT COORDS="30, 60, 130, 110">
```

- POLYGON – requer um numero par de valores x1, y1, ... xn, yn; onde cada par (xn, yn) corresponde a um dos vértices do polígono. No exemplo abaixo é definido um triângulo com os vértices (30, 40), (100, 80) e (30, 80):

```
<AREA SHAPE=POLYGON COORDS="30, 40, 100, 80, 30, 80">
```

A tag <AREA> deve ter definido o atributo HREF para indicar a URL que deve ser carregada após a seleção da região. É possível também utilizar o atributo ALT para exibir um texto explicativo quando o cursor do mouse passar sobre a área.

Nada impede que haja sobreposição de áreas. Quando a coordenada selecionada se encontra na interseção de duas áreas, a região selecionada será aquela que foi definida primeiro.

Após a definição do mapa, é necessário associar uma figura a ele. A tag <IMG> tem para isso o atributo USEMAP que deve receber como valor o nome do mapa associado.

A seguir um exemplo que ilustra a criação de um mapa clicável:

```
<MAP NAME="mainmap">
<AREA SHAPE=CIRCLE COORDS="50,50,25"
      HREF="http://home.netscape.com" >
<AREA SHAPE=RECT COORDS="50,50,100,100"
      HREF="http://developer.netscape.com" >
<AREA SHAPE=RECT COORDS="0,0,100,100"
      HREF="http://developer.netscape.com/docs" >
</MAP>
<IMG SRC="images/shapes.gif" WIDTH=100 HEIGHT=100
      ALT="shapes" USEMAP="#mainmap">
```

## 10. A tag <META>

Uma tag muito importante que pode ser colocada no interior da área de cabeçalho (área delimitada pela tag <HEAD> ... </HEAD>) é a tag <META>. Esta tag tem múltiplos usos: fornecer informações para sites de busca, guardar informações de documentação, enviar informações especiais para o navegador e fazer a troca automática de documento sendo exibido pelo navegador. Esta tag reconhece três atributos, cujos valores variam conforme a função desempenhada pela tag. Os atributos são: HTTP-EQUIV, NAME e CONTENT.

### Informação para Sites de Busca

Quando uma página é cadastrada num site de busca, dois tipos de informação são armazenados:

- Título da página – O título da página é definido pela tag <TITLE> e é a primeira informação exibida como resultado de uma busca.
- Descrição da página – Normalmente a descrição é formada pelos primeiros 200 caracteres a aparecer na página após a tag <BODY>. É possível, porém, indicar explicitamente esta descrição independente da informação exibida na página.

A descrição da página pode ser definida através da tag <META> com atributo NAME recebendo o valor “description” e o atributo CONTENT recebendo como valor a descrição da página.

A ordem em que uma página aparece como resultado de uma pesquisa depende do seu conteúdo. As páginas que contêm mais ocorrências das chaves da busca aparecem em primeiro lugar. É possível utilizar a tag <META> para fornecer mais informação ao mecanismo de busca definindo o atributo NAME com o valor “keywords” e o atributo CONTENT com uma série de palavras separadas por vírgulas. A informação da tag <META> é acrescentada a informação obtida no resto da página (não a substitui).

O exemplo a seguir ilustra a utilização desta tag:

```
<HEAD>
<TITLE>Minha Home Page</TITLE>
<META NAME="description"
      CONTENT="Esta é a descrição da minha página">
<META NAME="keywords"
      CONTENT="fotografia, programação, poesia">
</HEAD>
```

Observações a cerca dos sites de busca:

- Tamanho – Normalmente as informações de descrição são limitadas em 200 caracteres e as chaves em 1000 caracteres.
- Abuso de chaves – O mecanismo de busca ignora toda a lista de chaves quando uma chave é utilizada mais de 7 vezes numa tag <META>.
- Páginas com imagens – O atributo ALT das imagens também é levado em conta pelos mecanismos de busca. Assim páginas cujo conteúdo principal seja gráfico também podem ser encontradas.
- Páginas com javascript – Páginas começadas por programas em javascript devem utilizar a tag <META> para definir o conteúdo da página.

No endereço a seguir é possível encontrar informações acerca de como vários sites de busca obtêm informações nas páginas e da frequência que estas informações são atualizadas:

<http://www.eons.com/metatags.htm>

## Documentação

O exemplo a seguir mostra o cabeçalho de uma página criada no FrontPage. O editor inseriu uma série de informações apenas com o intuito de documentar a página (como autor e programa utilizado para gerar a página). A terceira tag <META> é interpretada pelo próprio FrontPage e apenas para ele faz sentido. Esta tag indica que na página qual estilo pré-definido foi utilizado.



```
<head>
<title>Home Page da Ana Lúcia</title>
<meta NAME="Author" CONTENT="João Sérgio S. Assis">
<meta NAME="GENERATOR" CONTENT="Microsoft FrontPage 3.0">
<meta NAME="Microsoft Theme" content="leaves 011, default">
</head>
```

## Informações especiais para o navegador

Alguns navegadores reconhecem algumas informações especiais e podem tomar atitudes em função delas. É possível por exemplo especificar uma data de validade da página (a data deve ser especificada no formato mostrado no exemplo abaixo), a partir da qual ela deve ser retirada do cache e recarregada. O exemplo a seguir ilustra esta utilização da tag:

```
<head>
<title>Minha Home Page</title>
<META HTTP-EQUIV="EXPIRES"
    CONTENT="Fri, 31 Dec 1999 00:00:01 GMT">
<META HTTP-EQUIV="CHARSET" CONTENT="ISO-8859-1">
<META HTTP-EQUIV="CONTENT-LANGUAGE" CONTENT="Portuguese">
</head>
```

## Troca automática de documento

De todas as utilidades da tag <META> esta possivelmente é a mais interessante de todas. É possível indicar ao navegador que após alguns segundos a página que está sendo exibida deve ser substituída por uma outra página. Esta tag utilizada desta forma pode ter várias aplicações:

- Redirecionamento - quando o endereço de uma página mudou mas se deseja que as pessoas que utilizam o endereço antigo continuem a ter acesso a página. É possível redirecionar automaticamente os que acessam o endereço antigo para o novo endereço.
- Apresentação de slides – Um conjunto de imagens (uma em cada página) é apresentado sem a intervenção do usuário.

- Animações – Neste caso só tem um resultado satisfatório quando se está carregando imagens locais ou numa Intranet, com tempos de acesso pequenos.

Para fazer a troca automática de documento deve-se definir na tag <META> o atributo HTTP-EQUIV com o valor “refresh”. O atributo CONTENT deve ser definido com o seguinte formato:

CONTENT="tempo; URL=página"

Onde:

- Tempo – tempo em segundos até carregar a nova página.
- Página – URL da nova página a ser carregada.

No exemplo a seguir a pagina nova.html é carregada após 10 segundos:

```
<META HTTP-EQUIV="refresh" CONTENT="10; URL=nova.html">
```

## 11. Multimídia e animações

### Multimídia

Os navegadores também podem ser utilizados para reproduzir arquivos de áudio e vídeo. Como há uma infinidade de formatos para este tipo de dado, a reprodução destes tipos de arquivo é realizada através de módulos incorporados ao navegador (plugin) ou a programas externos. Alguns navegadores costumam, durante a sua instalação, a instalar os pluggins para os tipos de arquivos mais comuns. Para outros pluggins é necessário que o próprio usuário faça a instalação. Quando o navegador não tem determinado plugin simplesmente ignora o arquivo requisitado. Na tabela a seguir alguns formatos comuns:

Formatos de áudio	
Wav	Formato padrão Windows.
Au	Formato padrão Unix
Mp3	Formato compactado.
Mid	Música produzida por sintetizador.
ra	Real Audio (execução por demanda).

Formatos de vídeo	
Avi	Formato padrão Microsoft.
Mov	Formato padrão Apple.
Mpeg	Formato compactado.

## Inserindo áudio e vídeo

Uma das formas fazer com que um arquivo de áudio ou vídeo seja executado é incluí-lo no documento na forma de um link. Desta forma ele só será executado se o usuário selecionar o link. No exemplo abaixo, a seleção do texto “link p/ música” faz com que o navegador carregue e execute o arquivo xmas.wav:

```
<A HREF="xmas.wav">link p/ música</A>
```

Algumas vezes é interessante executar o áudio ou vídeo independente da intervenção do usuário. Um exemplo deste tipo de aplicação é a inclusão de uma música de fundo numa página. Isso é um pouco mais complicado pois não há uma padronização muito rigorosa entre os navegadores.

Ambos os navegadores reconhecem a tag <EMBED>, que serve para exibir informação produzida por um módulo incorporado ao navegador (plugin). Para cada um deles, porém, é necessário definir determinados atributos para que seja produzido um resultado satisfatório. Em qualquer caso é fundamental definir o atributo SRC com a URL do arquivo que vai ser exibido (arquivo de som ou de vídeo). Para os demais atributos é necessário analisar caso por caso.

### A tag <EMBED> no Netscape

A inclusão da tag <EMBED> apenas com o atributo SRC definido faz com que o Netscape inclua na página o painel de controle do plugin. O som ou imagem não é exibido imediatamente mas o usuário pode, através do painel de controle, exibi-lo quantas vezes desejar.

Normalmente o Netscape não sabe que espaço reservar na página para o painel de controle e por isso é necessário definir também os atributos WIDTH e HEIGHT, informando a largura e a altura a ser ocupada.

A inclusão do atributo HIDDEN="True" faz com que, no caso de um arquivo de som, o painel de controle não seja exibido e o som seja tocado

imediatamente após o carregamento da página. No caso de um vídeo este atributo tem o mesmo efeito de definir a largura e a altura da janela de exibição como zero, o que faz com que o navegador não possa mostrar nada (normalmente isso causa um erro no plugin).

Para fazer com que o vídeo comece a ser exibido após o carregamento da página é necessário definir o atributo `AUTOSTART="True"`, além da altura e largura ocupadas pelo vídeo na janela. Quando este atributo está definido, o Netscape não exibe o painel de controle do plugin, apenas o próprio vídeo.

A tag `<EMBED>` também conta com o atributo `ALIGN` que pode receber os mesmos valores que recebe para a tag `<IMG>` e comporta-se da mesma maneira.

O atributo `LOOP` recebe um valor que indica o número de vezes que o som deve ser repetido (p. ex. `LOOP=10` faz com que o som seja tocado 10 vezes). Este atributo não tem nenhum efeito em arquivos de vídeo. O exemplo abaixo mostra a utilização desta tag tanto para exibir som como vídeo:

```
<EMBED SRC="x.avi" AUTOSTART="True" width=200 height=200>  
<EMBED SRC="x.wav" HIDDEN="True" LOOP=100>
```

### A tag `<EMBED>` no Explorer

Quando se define a tag `<EMBED>` apenas com o atributo `SRC`, o Explorer também exibe o painel de controle do plugin. Diferentemente do Netscape não é necessário definir a largura e altura pois o navegador consegue determinar o espaço ocupado pela janela do plugin. O atributo `ALIGN` tem o mesmo efeito do Netscape.

O atributo `HIDDEN` também tem o efeito de fazer desaparecer o painel de controle do plugin mas não faz com que ele seja executado automaticamente. Para que isso aconteça é necessário definir o atributo `AUTOSTART="True"` mesmo para arquivos de som.

O atributo LOOP tem efeito tanto sobre arquivos de som como de imagem e basta definir um valor diferente de zero para que o arquivo seja re-exibido infinitamente. O exemplo a seguir permite exibir arquivos de som e imagem em ambos os tipos de navegador (quase que com o mesmo efeito):

```
<EMBED SRC="x.avi" AUTOSTART="True" width=200 height=200>  
<EMBED SRC="x.wav" AUTOSTART="True" HIDDEN="True" LOOP=100>
```

No Explorer é possível utilizar a tag <IMG> para exibir vídeos definindo o atributo DYN SRC para indicar o arquivo a ser carregado. É conveniente, neste caso, definir o atributo SRC com uma imagem alternativa. O Explorer vai ignorar o SRC e exibir o vídeo, já o Netscape fará o contrário. Arquivos de som podem ser exibidos utilizando a tag <BGSOUND>. O exemplo acima poderia ser escrito então como:

```
<IMG DYN SRC="x.avi" SRC="alt.jpg" loop=yes>  
<BGSOUND SRC="x.wav" loop=yes>
```

Obviamente essas formatações não terão efeito algum se o browser não estiver configurado para tocar música ou se o computador que receber a página não tiver uma placa de som.

## Tags dependentes do navegador

Apenas a título de curiosidade são descritas nesta seção duas container tags implementadas apenas em um determinado tipo de navegador. Os navegadores que não as reconhecem vão simplesmente exibir o texto contido na tag sem realizar nenhuma ação especial

A tag <MARQUEE> ... </MARQUEE>, que funciona apenas no navegador Internet Explorer, serve para exibir um texto rolando na janela, da esquerda para a direita, num efeito similar a de um letreiro luminoso. O efeito causado por esta tag pode ser visualmente interessante, mas não convém utilizá-la por ser dependente do navegador. Através de Javascript é possível obter o mesmo efeito, com a vantagem da portabilidade.

A tag `<BLINK> ... </BLINK>`, que funciona apenas no navegador Netscape, serve para exibir um texto piscando na janela. Novamente, não é conveniente utilizar esta tag para manter a uniformidade de aspecto da página independente do navegador utilizado para visualização.

## Animações

Há várias formas de criar animações em páginas de Web:

- GIFs animadas
- Applets JAVA
- HTML dinâmico e Javascript

GIFS animadas são pequenas seqüências de animação compostas por várias imagens do tipo GIF agrupadas em um arquivo. A criação de uma GIF animada é muito simples. Basta criar cada uma das imagens que vão compor a seqüência, com pequenas variações no desenho para cada instante de tempo (como num desenho animado). Estes arquivos de imagens são posteriormente agrupados por meio de um programa. Há diversos programas que auxiliam a criação de GIFs animadas disponíveis na rede, apenas para citar alguns, por exemplo:

- <http://www.mindworkshop.com/alchemy/gifcon.html>
- [http://www.webutilities.com/ga/ga\\_main.htm](http://www.webutilities.com/ga/ga_main.htm)
- <http://rtlsoft.com/animagic/index.html>
- <http://www.moviegear.com/>

Para animações mais complexas e de figuras maiores pode-se optar por usar applets JAVA ou Javascript. Um applet é um pequeno programa em Java voltado para utilização na Web. Mesmo que não se saiba como criá-los é possível utilizar applets prontos, que permitam reconfiguração. Um exemplo é disponibilizado pela Sun no endereço:

<http://java.sun.com/applets/Animator/index.html>

## 12. Tabelas

Neste ponto chegamos a umas das tags mais importantes do HTML: a tag `<TABLE> .... </TABLE>`. Esta tag permite a definição de tabelas no documento. Mais do que isso porém, através das tabelas é possível forçar o posicionamento dos elementos em uma página. Esta tag deve vir sempre acompanhada das tags `<TR>...</TR>` e `<TD>...</TD>` (ou `<TH>...</TH>`), que servem para definir, respectivamente, as linhas e as células de cada coluna da tabela.

A tag `<TH>...</TH>` é utilizada para definir células que servirão de título da tabela (não é obrigatório que uma tabela tenha título) e o texto em seu interior é escrito em negrito. A não inclusão de linhas e colunas faz com que alguns navegadores simplesmente ignorem a tag `<TABLE>...</TABLE>`. O exemplo a seguir mostra uma tabela com duas linhas e três colunas:

```
<TABLE BORDER>
  <TR>
    <TD>Col 1 Lin 1</TD>
    <TD>Col 2 Lin 1</TD>
    <TD>Col 3 Lin 1</TD>
  </TR>
  <TR>
    <TD>Col 1 Lin 2</TD>
    <TD>Col 2 Lin 2</TD>
    <TD>Col 3 Lin 2</TD>
  </TR>
</TABLE>
```

Este trecho HTML vai produzir como saída a seguinte tabela:

Col 1 Lin 1	Col 2 Lin 1	Col 3 Lin 1
Col 1 Lin 2	Col 2 Lin 2	Col 3 Lin 2



Elemento	Descrição	Atributos
<TABLE> ... </TABLE>	Tabela	BORDER, CELLSPACING, CELLPADDING, BACKGROUND, WIDTH, BGCOLOR, ALIGN
<TR> ... </TR>	Linha da tabela	VALIGN , ALIGN
<TH> ... </TH>	Coluna (título)	VALIGN, ALIGN, ROWSPAN, COLSPAN, WIDTH, BGCOLOR, BACKGROUND, NOWRAP
<TD> ... </TD>	Coluna da tabela	

### Atributos de <TABLE>

A borda da tabela só é visível se o atributo BORDER estiver definido. Tabelas sem borda são muito úteis para posicionar elementos dentro de uma página. O atributo CELLSPACING define o espaço entre as células da tabela e seu valor default é 2. O atributo CELLPADDING o espaço entre o dado contido numa célula e a borda (o valor default é 1).

O atributo WIDTH define a largura da tabela na janela. A largura pode ser definida em termos absolutos (número de pixels) ou em termos da porcentagem da janela ocupada pela tabela. Se este atributo não for especificado, a largura é definida pelo conteúdo da tabela. O atributo ALIGN define o alinhamento da tabela em relação a janela (pode receber os valores CENTER, RIGHT, LEFT).

É possível atribuir cor (atributo BGCOLOR) ou uma imagem (atributo BACKGROUND) para o fundo de uma tabela. O funcionamento destes atributos é semelhante aos destes atributos na tag <BODY> ... </BODY>..

### Atributos de <TR>

Os atributos VALIGN e ALIGN servem para definir o alinhamento dos elementos dentro das células de uma determinada linha da tabela. O atributo VALIGN (alinhamento vertical) pode receber o valor TOP, MIDDLE ou

BOTTOM (topo, meio ou fundo) e o valor default é MIDDLE. O atributo ALIGN (alinhamento horizontal) pode receber o valor LEFT, CENTER ou RIGHT (esquerdo, centro ou direito) e o valor default é LEFT.

#### Atributos de <TD> e <TH>

Estas tags também tem VALIGN e ALIGN como atributos, com o mesmo significado da tag anterior, mas o alinhamento vale apenas para a célula em que foi definido (suplantando o alinhamento da linha, se houver). O atributo WIDTH permite que se indique, pelo número de pixels ou por uma porcentagem da largura da tabela, o tamanho horizontal de uma célula. Caso várias células de uma mesma coluna definam este atributo vale o maior valor. O atributo NOWRAP exibe o texto do interior de uma coluna como uma linha continua, sem quebras.

O atributo ROWSPAN permite que se indique o número de linhas que uma célula deve ocupar. O atributo COLSPAN permite que se indique o número de colunas que uma célula deve ocupar. O exemplo a seguir ilustra a utilização destes atributos

```
<TABLE BORDER>
  <TR>
    <TD ROWSPAN=2>Col 1 Lin 1</TD>
    <TD>Col 2 Lin 1</TD>
    <TD>Col 3 Lin 1</TD>
  </TR>
  <TR>
    <TD COLSPAN=2>Col 2 Lin 2</TD>
  </TR>
</TABLE>
```

Este trecho HTML vai produzir como saída a seguinte tabela:

Col 1 Lin 1	Col 2 Lin 1	Col 3 Lin 1
	Col 2 Lin 2	

## 13. Frames

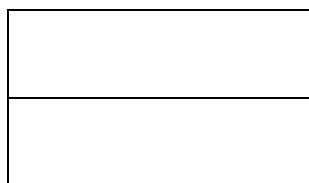
Uma outra forma de forçar o posicionamento de elementos e dividir o espaço de uma janela é através dos Frames. Este é um recurso relativamente recente em HTML e a demora de alguns navegadores em incorporar esta nova facilidade fez surgir uma certa má vontade em algumas pessoas.

Este recurso permite dividir a janela do navegador em várias áreas independentes (chamadas Frames). Em cada uma destas áreas pode ser carregado e visualizado um arquivo HTML. A independência de cada Frame, que permite que o conteúdo de um seja rolado enquanto outro permanece estático, faz com que este recurso seja ideal para a criação de menus.

A construção de uma página utilizando Frames torna necessário a criação de um arquivo de layout e de, pelo menos, dois arquivos HTML comuns. O arquivo de layout também é um documento HTML, onde a tag <BODY> ... </BODY> é substituída pela tag <FRAMESET> ... </FRAMESET>, e serve para definir a estrutura da janela principal (número, tamanho e localização dos Frames). O arquivo de layout tem, portanto, o seguinte formato:

```
<HTML>
  <HEAD>
    ...
  </HEAD>
  <FRAMESET ROWS="50%, 50%">
    <FRAME>
    <FRAME>
  </FRAMESET>
  <NOFRAMES>
  </NOFRAMES>
</HTML>
```

Este trecho HTML dividiria a tela da seguinte forma:



## Atributos de <FRAMESET>

Atributos de <FRAMESET>	
Atributos	Descrição
ROWS	Número e altura de cada linha de Frames
COLS	Número e largura de cada coluna de Frames
FRAMEBORDER	Indica se o Frame terá ou não borda (yes ou no)
BORDER	Largura da borda do Frame
BORDERCOLOR	Cor da borda do Frame

Os Frames são dispostos na janela do navegador na forma de linhas e colunas, como uma tabela. Os atributos ROWS e COLS servem para definir o número e tamanho ocupado por cada linha e coluna de Frames. O formato deste atributo é ROWS="v, v, ..." ou COLS="v, v, ...", onde v pode ser:

- Um número indicando a altura (no caso de ROWS) ou a largura (COLS) em pixels de um Frame. Ex: ROWS="100, 50, 200" define 3 linhas de Frames, onde a primeira tem a altura de 100 pontos, a segunda de 50 e a terceira de 200.
- Um percentual indicando a altura ou largura em função do tamanho da janela do navegador. Ex: COLS="30%, 20%, 50%" define 3 colunas de Frames, onde a primeira ocupa 30% da largura da janela, a segunda 20% e a terceira 50%.
- O caracter \*, indicando altura ou largura relativa aos demais valores definidos. Ex: COLS="30%, 20%, \*, \*" define 4 colunas de Frames, onde a primeira ocupa 30% da largura da janela, a segunda 20%, a terceira e a quarta dividem os 50% restante, ficando cada uma com 25%.

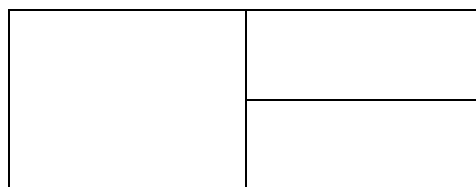
- Um número n seguido do caracter \*, indica que um Frame ocupará n vezes o espaço ocupado pelos demais Frames definidos relativamente. Ex: ROWS="40%, 2\*, \*" define 3 linhas de Frames, onde a primeira ocupa 40% da altura da janela, a segunda 40% (2/3 da altura restante) e a terceira 20% (1/3 da altura restante).

Por definição, os Frames são separados por bordas da cor da borda da janela do navegador. Através da atribuição do valor "no" ao atributo FRAMEBORDER é possível inibir a exibição da borda. O atributo BORDER permite atribuir um valor para a largura da borda e o atributo BORDERCOLOR permite mudar a sua cor.

É possível incluir FrameSets dentro de um FrameSet de modo a subdividir um Frame. Por exemplo, o seguinte arquivo de layout produziria:

```
<HTML>
  <HEAD>
    ...
  </HEAD>
  <FRAMESET COLS="50%, 50%">
    <FRAME>
      <FRAMESET ROWS="50%, 50%">
        <FRAME>
        <FRAME>
      </FRAMESET>
    </FRAMESET>
  <NOFRAMES>
  </NOFRAMES>
</HTML>
```

Este trecho HTML dividiria a tela da seguinte forma:



## Atributos de FRAME

Atributos de <FRAME>	
Atributos	Descrição
SRC	Arquivo inicialmente mostrado no Frame
NAME	Nome associado ao Frame
SCROLLING	Indica se o Frame será rolável (yes ou no)
NORESIZE	Usuário não pode mudar de tamanho do Frame
MARGINHEIGHT	Altura da margem do Frame
MARGINWIDTH	Largura da margem do Frame

O atributo SRC indica qual documento HTML será exibido no Frame após o carregamento do arquivo de layout. Para exibir um novo documento num Frame é necessário que isso seja resultado da seleção de um link. É possível, portanto, indicar que a URL referenciada num link seja exibida em um determinado Frame. Isto é feito através da definição do atributo TARGET da tag <A>...</A>. No exemplo abaixo a seleção de “primeiro link” faz com que o documento f1.html seja exibido no Frame “esquerdo”; a seleção de “segundo link” faz com que o documento f2.html seja exibido no Frame “direito”:

```
<HTML>
  <HEAD>
    ...
  </HEAD>
  <BODY>
    <A SRC="f1.html" TARGET="esquerdo">primeiro link</A>
    <A SRC="f2.html" TARGET="direito">segundo link</A>
  </BODY>
</HTML>
```

Para que o navegador saiba quais são os Frames “direito” e “esquerdo” é necessário atribuir nomes a eles, o que é feito através da inclusão do atributo

NAME da tag <FRAME>. No exemplo a seguir é mostrado o arquivo de layout onde foram definidos os Frames “direito” e “esquerdo”:

```
<HTML>
  <HEAD>
    ...
  </HEAD>
  <FRAMESET COLS="50%, 50%">
    <FRAME NAME="esquerdo">
    <FRAME NAME="direito">
  </FRAMESET>
</HTML>
```

Existem alguns nomes especiais que podem ser utilizados no atributo TARGET. O nome \_blank faz com que o documento HTML seja carregado numa nova janela do navegador. Isto é particularmente desejável quando se faz referências a um documento de um outro site. Desta forma a página onde está a referência continua sendo visível. O nome \_self indica que o novo documento será carregado no mesmo Frame onde está a página com o link que causou sua exibição. O nome \_parent faz com que o novo documento seja aberto na janela onde está o arquivo de layout, em substituição a este. Para entender o nome \_top é necessário recorrer a um exemplo:

```
<HTML>
  <HEAD>
    ...
  </HEAD>
  <FRAMESET COLS="50%, 50%">
    <FRAME NAME="esquerdo">
      <FRAMESET ROWS="50%, 50%">
        <FRAME SRC="s.htm" NAME="superior">
          <FRAMESET COLS="50%, 50%">
            <FRAME SRC="ie.htm" NAME="InfEsquerdo">
            <FRAME NAME="InfDireito">
          </FRAMESET>
        </FRAMESET>
      </FRAMESET>
    </FRAMESET>
  <NOFRAMES>
  </NOFRAMES>
</HTML>
```

Se a página ie.htm contiver um link com o TARGET=\_top, a seleção deste link fará com que uma página seja exibida abaixo do Frame “superior”, no lugar

dos Frames “InfEsquerdo” e “InfDireito”. Já se a página s.htm contiver um link com o TARGET=\_top, a seleção deste link fará com que uma página seja exibida ao lado do Frame “Esquerdo”, no lugar dos Frames “Superior”, “InfEsquerdo” e “InfDireito”.

Como nas células de uma tabela, há atributos para definir uma margem dentro da qual o documento vai ser exibido. O atributo MARGINHEIGHT serve para definir a altura da margem. O atributo MARGINWIDTH serve para definir a largura da margem. O atributo SCROLLING permite controlar a exibição da barra de rolagem vertical do Frame: o valor **yes** faz com que a barra de rolagem esteja sempre visível, o valor **no** faz com que ela nunca seja exibida e o valor **auto** (default) faz com que ela seja exibida apenas se necessário. O usuário pode, a qualquer momento, modificar o tamanho inicial do Frame, a menos que o atributo NORESIZE seja definido.

Como foi dito anteriormente, Frames é um recurso relativamente recente em HTML. Como é possível que ainda estejam sendo utilizados navegadores que não suportam este recurso existe a tag <NOFRAMES> ... </NOFRAMES>. Os navegadores mais antigos só vão interpretar o que há dentro desta tag e os mais novos vão ignorá-la. Ela deve ser incluída no fim do arquivo de layout com um código HTML alternativo ao código contendo Frames, como no exemplo a seguir:

```
<HTML>
  <HEAD> ... </HEAD>
  <FRAMESET>
    <FRAME NAME="esquerdo">
    <FRAME NAME="direito">
  </FRAMESET>
  <NOFRAMES>
    Utilize um browser que suporte Frames nesta página.
  </NOFRAMES>
</HTML>
```



## 14. Formulários

A principal forma de trocar informações entre o usuário de uma página e o responsável por um site é através de formulários. A linguagem HTML oferece uma série de elementos de interação com usuário que podem ser agrupados em um formulário. O resultado da interação do usuário é posteriormente enviada para um programa sendo executado no servidor ou enviado por e-mail para o responsável pelo site.

A tag `<FORM> ... </FORM>` delimita a área ocupada pelo formulário e possui dois atributos que indicam quem vai receber os dados do formulário (atributo ACTION) e a forma como eles serão enviados pela rede (atributo METHOD).

Quando METHOD recebe o valor GET o conteúdo do formulário é enviado para o servidor anexado à URL definida em ACTION. Este método é adequado a formulários pequenos, com pouca informação. Também não é conveniente enviar informações sigilosas. O outro valor que pode ser atribuído a METHOD é POST. Neste caso as informações do formulário são enviadas separadas da URL.

O atributo ACTION serve para informar a URL do programa CGI que irá processar o formulário. É possível utilizar este atributo para informar que os dados do formulário serão enviados por e-mail para o administrador do site. Neste caso deve-se definir o atributo ENCTYPE com o valor "text/plain" e o método de envio deve ser definido como POST. O exemplo a seguir mostra um formulário que será enviado para o endereço joao@nce.ufrj.br:

```
<FORM
  METHOD=POST
  ACTION="mailto:joao@nce.ufrj.br"
  ENCTYPE="text/plain">
...
</FORM>
```

O interior da tag <FORM>...</FORM> pode conter quatro tipos de tag que servem para definir os diversos elementos de interação. Estas tags são:

- <INPUT>
- <SELECT> ... </SELECT>
- <OPTION>
- <TEXTAREA> ... </TEXTAREA>

A tag <INPUT>

A tag <INPUT> é bastante versátil e permite criar campos para entrada de texto, campos para entrada de senhas, botões comuns, botões estilo rádio e botões de seleção múltipla.

Atributos de <INPUT>		
Atributo	Valor	Descrição
TYPE	TEXT	Campo de entrada de uma linha de texto.
	PASSWORD	Campo de entrada de senha.
	CHECKBOX	Botões de rádio e botões de seleção múltipla.
	BUTTON	Botão comum.
	SUBMIT	Botão que envia o formulário.
	RESET	Botão que limpa os campos do formulário.
NAME	String	Nome da variável associada ao campo.
VALUE	String	Valor default do campo.
SIZE	Tamanho	Tamanho do campo TEXT ou PASSWORD.
MAXLENGTH	Tamanho	Número máximo de caracteres a ser lido.
CHECKED		Define um campo CHECKBOX como selecionado.

## Campo para entrada de uma linha de texto

A tag `<INPUT>` com o atributo `TYPE=TEXT` definido serve para criar um campo para entrada de uma linha de texto. Deve-se definir o atributo `NAME` com um nome que estará associado ao valor inserido no campo de entrada. O exemplo abaixo faz com que o navegador exiba na página um campo de entrada de texto; caso o usuário digite “joao” e envie o formulário, a informação será recebida pelo programa na forma de “conta=joao”:

```
Conta:<INPUT TYPE=TEXT NAME=conta>
```

A tag `<INPUT>` com o atributo `TYPE=PASSWORD` definido serve para entrar informações sigilosas, como uma senha por exemplo. Neste caso, o que é digitado pelo usuário não é exibido na janela do navegador. Para tirar proveito do sigilo, este tipo de campo deve ser enviado pelo método POST. No exemplo abaixo se o usuário digitar “pcpw99” e enviar o formulário, a informação recebida pelo programa será “senha=pcpw99”:

```
Senha:<INPUT TYPE=PASSWORD NAME=senha>
```

É possível definir o tamanho da área para entrada de texto através do atributo `SIZE`. Este atributo não limita o número de caracteres que o usuário pode digitar, apenas o tamanho do campo. Para se indicar o número máximo de caracteres que pode ser digitado pelo usuário deve-se definir o atributo `MAXLENGTH`. No exemplo abaixo a área de entrada de texto tem o tamanho de 7 caracteres mas o usuário pode digitar até 11 caracteres:

```
Telefone:<INPUT TYPE=TEXT NAME=telef SIZE=7 MAXLENGTH=11>
```

É possível preencher o campo com um valor padrão que é exibido antes do usuário digitar qualquer coisa. Isso é feito através do atributo `VALUE`. A omissão do atributo `TYPE` faz com que o navegador assuma o tipo `TEXT`.

## Botões de rádio

Um tipo de elemento de interação muito comum são os botões de rádio (Radio Buttons). Estes botões são utilizados quando o usuário deve escolher

uma única resposta entre várias. Quando um dos botões é selecionado, automaticamente todos os outros são desselecionados. Estes elementos são criados pela tag `<INPUT>` com o atributo `TYPE=RADIO`. Para indicar ao navegador quais botões são mutuamente exclusivos é preciso que eles tenham o atributo `NAME` definido com o mesmo valor. O atributo `VALUE` deve ser definido e indica o valor que será enviado. Através do atributo `CHECKED` é possível indicar que uma das alternativas estará previamente selecionada. No exemplo abaixo, a opção “Superior” está previamente selecionada; se o usuário selecionar “Analfabeto” a informação enviada ao programa será “instruc=A”:

```
Nível de Instrução:<BR>
<INPUT TYPE=RADIO NAME=instruc VALUE=A>Analfabeto<br>
<INPUT TYPE=RADIO NAME=instruc VALUE=1>1º grau<br>
<INPUT TYPE=RADIO NAME=instruc VALUE=2>2º grau<br>
<INPUT TYPE=RADIO NAME=instruc VALUE=S CHECKED>superior<br>
```

### Botões de seleção múltipla

Quando é desejável escolher mais de uma resposta utiliza-se botões de seleção múltipla (toggle buttons). A criação destes botões é feita através da tag `<INPUT>` com o atributo `TYPE=CHECKBOX`. Diferente dos botões de rádio, cada um dos botões deve definir um valor diferente ao atributo `VALUE`. Como vários botões podem estar selecionados, vários deles podem ter o atributo `CHECKED` definido. No exemplo abaixo as opções “Netscape” e “Explorer” estavam previamente definidas; se o usuário enviar o formulário sem fazer nenhuma modificação o programa vai receber “nav1=N” e “nav3=E”:

```
Navegador utilizado:<BR>
<INPUT TYPE=CHECKBOX NAME=nav1 VALUE=N CHECKED>Netscape<br>
<INPUT TYPE=CHECKBOX NAME=nav2 VALUE=M>Mosaic<br>
<INPUT TYPE=CHECKBOX NAME=nav3 VALUE=E CHECKED>Explorer<br>
<INPUT TYPE=CHECKBOX NAME=nav4 VALUE=H>HotJava<br>
```

### Botão de envio de dados e limpeza do formulário

A tag `<INPUT>` permite a criação de dois tipos de botões que agem sobre todo o formulário. A definição do atributo `TYPE=SUBMIT` inclui um botão que faz o envio do formulário. A definição do atributo `TYPE=RESET` inclui um botão

que limpa os campos do formulário. Em ambos os casos, o atributo VALUE define um texto para ser escrito no interior do botão. A omissão deste atributo faz com que o navegador exiba um texto default. Normalmente estes dois botões são colocados no fim do formulário. O exemplo a seguir mostra a utilização destes botões:

```
<FORM>
    ...
    <INPUT TYPE=SUBMIT NAME="Envia">
    <INPUT TYPE=RESET NAME="Limpa">
</FORM>
```

## Área de texto

Quando se deseja enviar mais de uma linha de texto deve-se utilizar a tag <TEXTAREA> ... </TEXTAREA>. O atributo NAME deve ser definido para indicar a variável que será associada ao texto inserido. O tamanho da área de texto é definido pelo número de linhas (atributo ROWS) e pelo número de colunas (atributo COLS). No interior da tag é possível colocar um texto a ser exibido antes do usuário digitar qualquer coisa. O exemplo a seguir mostra a utilização desta tag:

```
Comentários:<BR>
    <TEXTAREA NAME=coment ROWS=3 COLS=40>
    Faça aqui seus comentários
</TEXTAREA>
```

## Listas de seleção

Quando há muitas opções a escolher pode não ser muito prático utilizar botões de rádio ou botões de seleção múltipla pelo espaço que eles ocupam. Um alternativa são as listas de seleção criadas através de tag <SELECT> ... </SELECT>. O atributo NAME deve ser utilizado para definir o nome da variável que será enviada ao programa. O atributo SIZE indica o número de opções mostradas de uma só vez pelo navegador (as demais opções podem ser acessadas através de uma barra de rolagem exibida à direita da lista). O atributo MULTIPLE indica que mais de uma opção pode ser selecionada.

No interior do <SELECT> devem ser colocadas cada uma das opções da lista, indicadas pela tag <OPTION>. O atributo SELECTED pode ser utilizado para definir qual opção está previamente selecionada. Se o atributo MULTIPLE foi definido na tag <SELECT> é possível definir o SELECTED para mais de uma opção.

O exemplo a seguir mostra a utilização destas tags:

Sexo:

```
<SELECT NAME=sexo>  
<OPTION SELECTED>Masculino  
<OPTION>Feminino  
</SELECT>
```

## 15. Folhas de estilo

Até a versão 4 da linguagem HTML, as tags tinham uma definição e comportamento padrão, que uns poucos atributos podiam modificar. Caso se desejasse criar páginas cuja aparência fosse ligeiramente diferente do comportamento padrão era necessário incluir os atributos e tags que definissem essa aparência nos pontos do documento onde fosse necessário, misturados a própria estrutura do documento.

Por exemplo, num documento cujo texto fosse de cor preta (definida na tag <BODY> através do atributo TEXT), caso se desejasse cabeçalhos de cor azul marinho seria necessário envolver cada ocorrência da tag <H1></H1> com a tag <FONT color=navy></FONT>. Qualquer modificação a ser realizada nesta definição tornava necessário buscar por todo o documento o que deveria ser modificado.

### Cascade Style Sheets (CSS)

A versão 4 do HTML introduziu o conceito de folhas de estilo (CSS – cascade style sheets). Através dos estilos é possível definir informações de tipografia, cor, layout e alinhamento de forma independente da estrutura do documento. Estas informações podem inclusive estar contidas em um arquivo a parte que pode ser incluído nos diversos documentos que compõem o site, garantindo assim uma apresentação comum em todas as páginas. Além das facilidades de formatação já existente foram acrescentadas novas facilidades, o que permite um controle muito maior do layout.

A folha de estilos é definida por meio da tag <STYLE> ... </STYLE> que deve ser colocada na seção <HEAD> do documento. Esta tag tem um atributo TYPE que indica a sintaxe dos estilos contidos na tag. Há dois tipos de sintaxe possível: a sintaxe CSS (normalmente o padrão, valor "text/css" do atributo TYPE) e a sintaxe javascript (reconhecida apenas pelo navegador netscape, valor "text/javascript" do atributo TYPE).

O exemplo a seguir faz com que todos os cabeçalhos do tipo 1 (<H1>...</H1>) sejam escritos utilizando o fonte arial e a cor azul; e todos os cabeçalhos do tipo 2 (<H2>...</H2>) sejam escritos utilizando o fonte arial e a cor verde:

```
<STYLE TYPE="text/css">
<!--
H1  {
    font-family: arial;
    color: blue;
}
H2  {
    font-family: arial;
    color: green;
}
-->
</STYLE>
```

A colocação dos estilos entre os símbolos <!-- e --> não é obrigatória. A utilização deste símbolos é ignorada pelos navegadores que conhecem a tag <STYLE> (explorer e netscape de versão igual ou superior a 4), mas não pelos navegadores de versões anteriores que desta forma tomam a definição dos estilos como comentário. O texto do interior da tag <STYLE> não é HTML e segue a seguinte sintaxe:

```
<STYLE TYPE="text/css">
/* comentário */
seletor {
    propriedade: valor;
    propriedade: valor valor valor ...;
    ...
}
seletor, seletor {
    propriedade: valor;
    ...
}
</style>
```

## Propriedades dos Estilos

Propriedades são atributos que definem a forma de visualização de uma determinada tag ou de um grupo de tags. Elas podem ser divididas em 7 grupos, de acordo com sua função e nem sempre são aplicáveis a qualquer



tag. Há pequenas incompatibilidades entre a forma como os dois principais tipos de navegadores (explorer e netscape) são afetados por algumas propriedades, que podem, inclusive, ser ignoradas. A implementação de estilos por parte do Netscape é mais recente e nem tudo que foi implementado no explorer é reconhecido por esse navegador. Os 7 tipos de propriedades são as que modificam: fontes de caracteres, cor e imagem de fundo, espaçamento, características de listas, alinhamento de texto, bordas e posicionamento de camadas. Os quadros a seguir mostram as propriedades de cada tipo, seus valores possíveis e o navegador(es) que as implementam (apenas excluído posicionamento de camadas que está fora do programa do curso).

## Valores

Há 5 tipos distintos de valores de propriedades:

- palavras chave - são palavras prédefinidas que podem expressar uma dimensão (small - pequeno), uma relação (bolder - mais escuro) ou o nome de um fonte de caracteres (arial, times, etc). Estas palavras não são case-sensitive.
- tamanho - é um valor numérico, precedido ou não de sinal e seguido da indicação da unidade. As unidades válidas são: pixel (px), polegadas (in), centímetros (cm), milímetros (mm), pontos (pt - 1 / 72 de uma polegada) ou picas (pc - 12 pontos). Por exemplo: 1in, 1.5cm, -3pt, +0,25mm.
- percentuais - um valor numérico seguido do símbolo %. Este valor não é dinâmico e diz respeito apenas ao momento em que o navegador for aplicar o estilo. Por exemplo: 120%.
- URLs - a forma de indicar uma URL é diferente da utilizada em HTML e segue a seguinte sintaxe: url(endereço). São, portanto, valores válidos de URL: url(http://equipe.nce.ufrj.br/joao/home.html) e url(foto.jpg).
- cores - há 3 formas de especificar uma cor: seu nome em inglês, a definição RGB do HTML (#RRGGBB - não pode estar entre aspas) ou através da

sintaxe `rgb(vermelho, verde, azul)`. São, portanto, valores válidos de cor: `yellow`, `#FFFF00`, `rgb(125, 100, 70)` ou `rgb(50%, 50%, 20%)`.

## Fonte

Propriedades	Valores Possíveis	browser
<b>font-family</b>	<i>lista de nome de fontes</i>	ambos
<b>font-size</b>	<i>tamanho, percentual, xx-small, x-small, small, medium, large, x-large, xx-large, larger, smaller</i>	ambos
<b>font-style</b>	<b>normal, oblique, italic</b>	ambos
<b>font-weight</b>	<b>normal, bold, bolder, lighter, 100, 200, 300, 400 (normal), 500, 600, 700, 800, 900</b>	ambos
<b>font-variant</b>	<b>normal, smallcaps</b>	explorer
<b>font</b>	<i>family size style weight variant</i>	explorer
<b>color</b>	<i>nome da cor, #RRGGBB</i>	ambos
<b>text-decoration</b>	<b>none, overline, underline, overline, blink, line-through</b>	ambos
<b>text-transform</b>	<b>capitalize, lowercase, none, uppercase</b>	ambos

## Cor e imagem de fundo

Propriedades	Valores Possíveis	browser
<b>background-position</b>	<i>tamanho (x y), percentual (x y), top, center, bottom, left, right</i>	explorer
<b>background-image</b>	<i>url da imagem</i>	ambos
<b>background-color</b>	<b>transparent, nome da cor, #RRGGBB</b>	ambos
<b>background-attachment</b>	<b>scroll, fixed</b>	explorer
<b>background-repeat</b>	<b>repeat, repeat-x, repeat-y, no-repeat</b>	ambos
<b>background</b>	<i>attachment color image position repeat</i>	explorer

## Espaçamento

Propriedades	Valores Possíveis	browser
<b>letter-spacing</b>	<i>tamanho</i>	explorer
<b>white-space</b>	<b>normal, nowrap, pre</b>	netscape
<b>line-height</b>	<i>tamanho, percentual, normal</i>	ambos
<b>margin-top</b>	<i>tamanho, percentual, auto</i>	ambos
<b>margin-bottom</b>	<i>tamanho, percentual, auto</i>	ambos
<b>margin-right</b>	<i>tamanho, percentual, auto</i>	ambos
<b>margin-left</b>	<i>tamanho, percentual, auto</i>	ambos
<b>margin</b>	<i>tamanho, percentual, auto</i>	explorer
<b>padding-top</b>	<i>tamanho, percentual, auto</i>	ambos
<b>padding-bottom</b>	<i>tamanho, percentual, auto</i>	ambos
<b>padding-right</b>	<i>tamanho, percentual, auto</i>	ambos
<b>padding-left</b>	<i>tamanho, percentual, auto</i>	ambos
<b>padding</b>	<i>tamanho, percentual, auto</i>	explorer
<b>word-spacing</b>	<i>tamanho</i>	ambos
<b>float</b>	<b>none, left, right</b>	ambos
<b>clear</b>	<b>none, both, left, right</b>	ambos

## Listas

Propriedades	Valores Possíveis	browser
<b>list-style-image</b>	<i>URL, none</i>	explorer
<b>list-style-type</b>	<b>circle, square, disc, lower-alpha, upper-alpha, lower-roman, upper-roman, decimal, none</b>	ambos
<b>list-style</b>	<i>image type</i>	ambos

## Alinhamento de texto

Propriedades	Valores Possíveis	browser
<b>text-indent</b>	<i>tamanho, percentual</i>	ambos
<b>text-align</b>	<b>left, center, right, justify</b>	ambos
<b>vertical-align</b>	<b>baseline, middle, sub, super, text-bottom, text-top, percentual</b>	ambos

## Bordas

Propriedades	Valores Possíveis	browser
<b>border-top-width</b>	<i>tamanho, thin, thick, medium</i>	ambos
<b>border-bottom-width</b>	<i>tamanho, thin, thick, medium</i>	ambos
<b>border-right-width</b>	<i>tamanho, thin, thick, medium</i>	ambos
<b>border-left-width</b>	<i>tamanho, thin, thick, medium</i>	ambos
<b>border-top</b>	<i>cor width style</i>	explorer
<b>border-bottom</b>	<i>cor width style</i>	explorer
<b>border-right</b>	<i>cor width style</i>	explorer
<b>border-left</b>	<i>cor width style</i>	explorer
<b>border-color</b>	<i>cor, #RRGGBB</i>	ambos
<b>border-width</b>	<i>tamanho, thin, thick, medium</i>	ambos
<b>border-style</b>	<b>none, dashed, dotted, double, groove, inset, outset, ridge, solid</b>	ambos
<b>border</b>	<i>cor width style</i>	ambos

## Seletor múltiplo

É possível atribuir o mesmo valor às propriedades de várias tags de uma só vez, colocando-as separadas por vírgulas antes da definição das propriedades. No exemplo a seguir todos os cabeçalhos H1, H2 e H3 e todos os textos em negrito serão escritos com a cor azul:

```
<STYLE>
H1, H2, H3, B {
    color: blue;
}
</STYLE>
```

## Seletor contextual

É possível indicar que as propriedades de uma determinada tag só serão modificadas quando ela se encontrar no interior de uma ou mais tags container. No exemplo a seguir e todo textos em itálico dentro de cabeçalhos H1 (apenas estes) serão escritos com a cor verde, os demais textos em itálico terão a cor padrão :

```
<STYLE>
H1 i {
    color: green;
}
</STYLE>
```

## Compartilhando estilos

É possível compartilhar estilos entre vários documentos HTML. Os estilos devem estar contidos num arquivo com extensão css, que pode ser incluído num documento por meio da tag <LINK> colocada em sua seção <HEAD>. A utilização de um arquivo de estilo permite criar uma verdadeira uniformização das páginas de um site. Qualquer modificação neste arquivo modifica todas as páginas de uma vez. O exemplo a seguir mostra a sintaxe da tag <LINK> para incluir o arquivo “estilo.css” num documento HTML:

```
<link rel=stylesheet type="text/css" href="estilo.css">
```

O valor do atributo HREF indica o nome do arquivo de estilos e além dele é necessário definir também os atributos REL e TYPE com os valores mostrados no exemplo. Um arquivo de estilos não é um arquivo HTML, ou seja, não é necessário nem se deve colocar os estilos dentro de uma tag <STYLE>.

## Classes de estilo

Com o que foi visto até o momento, ao definir o estilo de uma tag, sempre que ela for utilizada no documento terá o aspecto determinado por este estilo. Muitas vezes porém é desejável que uma tag tenha vários aspectos de acordo com o local onde é utilizada. É possível, portanto, criar diferentes estilos para uma tag criando várias classes. Para criar uma classe basta acrescentar ao nome da tag no seletor o caracter ponto (‘.’) seguido de um nome para a classe. No exemplo a seguir mostra uma definição aplicada a todos os cabeçalhos do <H1> (tipo e tamanho de fonte) e as definições de cor aplicáveis apenas aos cabeçalhos <H1> pertencentes as classes vermelho (cor de texto vermelha) e verde (cor de texto verde):

```
<style>
h1 {
    font-family: arial;
    font-size: 20pt;
}
h1.vermelho {
    color: red;
}
h1.verde {
    color: green;
}
</style>
```

Para aplicar o estilo definido numa classe a uma tag basta utilizar o atributo class, definindo como valor o nome da classe desejado. No exemplo a seguir o primeiro cabeçalho estará sujeito a definição de tamanho e tipo de fonte e será escrito na cor padrão; o segundo e o quarto cabeçalho também estarão sujeitos a definição de tamanho e tipo de fonte mas serão escritos na cor vermelha; o terceiro, por sua vez, estará sujeito a definição de tamanho e tipo de fonte mas será escrito na cor verde:

```
<h1>Cabeçalho normal</h1>
<h1 class=vermelho>Cabeçalho Vermelho</h1>
<h1 class=verde>Cabeçalho Verde</h1>
<h1 class=vermelho>Outro Cabeçalho Vermelho</h1>
```

## Classes genéricas

As classes definidas no exemplo da seção anterior só podem ser aplicadas a tags do tipo <H1> pois esta tag foi incluída no seletor. É possível criar classes aplicáveis a qualquer tipo de tag, bastando para isso omitir o nome da tag como mostrado no exemplo a seguir:

```
<STYLE>
.verde {
    color: green;
}
</STYLE>
```

A utilização de uma classe genérica também é feita por intermédio do atributo CLASS como mostra o exemplo a seguir:

```
<h1 class=verde>Este é um cabeçalho verde</h1>
<p class=verde>Este é um texto verde
```

## Pseudo-classes

A tag <A> tem algumas classes com nomes pré-definidos, que são associadas a estados assumidos por esta tag. Estes estados são:

Pseudo-classe	Descrição	browser
<b>link</b>	Link que ainda não foi visitado.	ambos
<b>active</b>	Link no momento que o usuário tem o botão do mouse pressionado.	ambos
<b>visited</b>	Link que já foi visitado.	ambos
<b>hover</b>	Link no momento em que o usuário passa o cursor do mouse sobre ele.	Explorer

No caso da pseudo-classe, diferente de uma classe comum, o nome da classe é separado do nome da tag no seletor pelo caracter ‘:’. Não existem pseudo-classes genéricas e, portanto, sempre é necessário indicar o nome da tag. O exemplo a seguir define a cor azul para os links não visitados, a cor vermelha e negrito para os links no momento em que o botão do mouse está pressionado sobre o link, a cor verde para os links que já foram visitados e a

cor vermelho escura para quando o cursor do mouse passar sobre o link (isto só funcionará para o explorer, sendo ignorado se o navegador for netscape):

```
<STYLE>
A:link      { color: blue; }
A:active    { color: red; font-weight: bold; }
A:visited   { color: green; }
A:hover     { color: darkred; }
</STYLE>
```

## Identificadores

Da mesma forma que uma classe modifica as definições feitas para uma tag, é possível redefinir, em um determinado elemento do documento, as definições de uma classe. Isso é feito através da criação de um identificador no interior da tag <STYLE>. O seletor de um identificador é um nome qualquer antecedido do caracter '#'. Ao definir um atributo ID com o nome de um identificador numa tag qualquer, esta tag tem seu aspecto modificado conforme o estilo definido no identificador. Se nesta tag também estiver definido o atributo classe, as propriedades definidas no identificador tem prioridade sobre as definidas na classe. Muito embora os navegadores normalmente não imponham restrições, um identificador deve ser utilizado em apenas uma tag, criando uma instância particular desta tag. No exemplo a seguir, os três cabeçalhos serão desenhados com o fonte arial e o tamanho grande, porém o primeiro será desenhado em verde (definição da classe), o segundo em azul e o terceiro em amarelo (definições dos identificadores):

```
<STYLE>
H1.grande {
    font-family: arial;
    font-size: large;
    color: green;
}
#azul      { color: blue; }
#amarelo   { color: yellow; }
</STYLE>
...
<H1 class=grande>Grande e verde</h1>
<H1 class=grande id=azul>Grande e azul</h1>
<H1 class=grande id=amarelo>Grande e amarelo</h1>
```



## Estilos inline

Uma outra forma de redefinir o estilo de uma tag é através do atributo STYLE. Este atributo pode ser colocado em qualquer tag e tem precedência sobre os estilos definidos através dos atributos ID e CLASS. O valor do atributo STYLE é uma string contendo propriedades e valores separados por vírgulas, da mesma forma que é utilizada associado a um seletor na tag <STYLE>. No exemplo a seguir, o segundo cabeçalho herda da classe o tipo do fonte, do identificador o tamanho e assume a cor definida no atributo STYLE:

```
<STYLE>
H1.grande {
    font-family: arial;
    font-size: xx-large;
    color: green;
}
#azul {
    font-size: x-large;
    color: blue;
}
</STYLE>
...
<H1 class=grande>Muito grande e verde</h1>
<H1 class=grande id=azul STYLE="color: red;">
    Grande, verde e com fonte times</h1>
```

## Tags div e span

Algumas vezes pode ser interessante modificar a forma de apresentação de um trecho do texto que não está associado a nenhuma tag em especial. Nestes casos a linguagem HTML fornece duas tags que não tem nenhum comportamento padrão especial e cujo aspecto pode ser determinado por meio de estilos: a tag <DIV> ... </DIV> e a tag <SPAN> ... </SPAN>.

A única diferença entre estas duas tags é que a tag <DIV> causa necessariamente uma mudança de linha antes e depois de sua utilização, ao passo que a tag <SPAN> pode ser utilizada no meio de um parágrafo sem causar nenhuma quebra de linha. A tag <DIV> pode ser utilizada para definir o estilo de vários parágrafos sem que seja necessário definir um a um o estilo das tag <P> destes parágrafos.

No exemplo a seguir a tag <DIV> define um parágrafo de cor vermelha, no meio do qual foi incluída uma palavra na cor verde:

```
<div style="color: red;">Este é um parágrafo vermelho com  
um texto <span style="color: green;">verde</span> no  
meio</div>
```

Estas duas tags tem uma grande importância em HTML 4 pois servem para definir camadas.

## Apêndice A - Caracteres Especiais

Car. Número	Código	Significado
&#00~&#08		Não é utilizado
&#09		Tabulação horizontal
&#10		Nova linha
&#11 -&#31		Não é utilizado
&#32		Espaço em branco
&#33		Ponto de exclamação
&#34	&quot;	Aspas
# &#35		Sinal numérico
\$ &#36		Cifrão
% &#37		Sinal de porcentagem
& &#38	&amp;	E comercial
&#39		Apóstrofo
&#40		Parêntese esquerdo
&#41		Parêntese direito
&#42		Asterisco
+ &#43		Sinal de mais
&#44		Vírgula
&#45		Hífen
&#46		Ponto final
&#47		Barra normal
0-9 &#48 - &#57		Dígitos de 0 a 9
&#58		Dois pontos
&#59		Ponto e vírgula
< &#60	&lt;	Símbolo menor do que
&#61		Sinal de igual
> &#62	&gt;	Símbolo maior do que
? &#63		Ponto de interrogação
@ &#64		Arroba
A-Z &#65 - &#90		Letras de A até Z
&#91		Colchete esquerdo
&#92		Barra invertida
&#93		Colchete direito
&#94		Acento circunflexo
&#95		Sublinhado
&#96		Acento grave
a-z &#97 - &#112		Letras de a até z
&#123		Chave esquerda
1 &#124		Barra vertical
&#125		Chave direita
~ &#126		Til
&#127 - &#159		Não é utilizado
&#160	&nbsp;	Espaço sem quebra
¡ &#161	&iexcl;	Ponto de exclamação invertido
¢ &#162	&cent;	Sinal de centavo
£ &#163	&pound;	Libra esterlina
¤ &#164	&curren;	Símbolo monetário geral
¥ &#165	&yen;	Iene
&#166	&brvbar;	Barra vertical quebrada

Car. Número	Código	Significado
§    #167	&sect;	Parágrafo (legal)
«    #168	&uml;	Trema
©    #169	&copy;	Copyright
a    #170	&ordf;	Ordinal feminino
«    #171	&laquo;	Aspas angulares de abertura
¬    #172	&not;	Sinal de não
-    #173	&shy;	Hífen longo
®    #174	&reg;	Marca registrada
-    #175	&macr;	Macron
°    #176	&deg;	Grau
±    #177	&plusmn;	Mais ou menos
²    #178	&sup2;	Ao quadrado
³    #179	&sup3;	Ao cubo
´    #180	&acute;	Acento agudo
μ    #181	&micro;	Mícron
¶    #182	&para;	Sinal de parágrafo
·    #183	&middot;	Ponto no meio
¸    #184	&cedil;	Cedilha
¹    #185	&supl;	Elevado a um
º    #186	&ordm;	Ordinal masculino
»    #187	&raquo;	Aspas angulares de fechamento
¼    #188	&frac14;	Fração de um/quarto
½    #189	&frac12;	Fração de meio
¾    #190	&frac34;	Fração de três/quartos
¿    #191	&iquest;	Interrogação invertida
À    #192	&Agrave;	A maiúsculo com crase
Á    #193	&Aacute;	A maiúsculo com acento agudo
Â    #194	&Acirc;	A maiúsculo com acento circunflexo
Ã    #195	&Atilde;	A maiúsculo com til
Ä    #196	&Aring;	A maiúsculo com trema
Å    #197	&Auml;	A maiúsculo com anel
Æ    #198	&Aelig;	Ligatura AE maiúscula
Ç    #199	&Ccedil;	C cedilha maiúsculo
È    #200	&Egrave;	E maiúsculo com crase
É    #201	&Eacute;	E maiúsculo agudo
Ê    #202	&Ecirc;	E maiúsculo com circunflexo
Ë    #203	&Euml;	E maiúsculo com trema
Ì    #204	&Igrave;	I maiúsculo com crase
Í    #205	&Iacute;	I maiúsculo com acento agudo
Î    #206	&Icirc;	I maiúsculo com acento circunflexo
Ï    #207	&Iuml;	I maiúsculo com trema
Ð    #208	&ETH;	Eth maiúsculo islandês
Ñ    #209	&Ntilde;	N maiúsculo com til
Ò    #210	&Ograve;	O maiúsculo com crase
Ó    #211	&Oacute;	O maiúsculo com acento agudo
Ô    #212	&Ocirc;	O maiúsculo com acento circunflexo
Õ    #213	&Otilde;	O maiúsculo com til
Ö    #214	&Ouml;	O maiúsculo com trema
×	&#215	Sinal de multiplicação

Car. Número	Código	Significado
Ø &#216	&Oslash;	O maiúsculo com uma barra
ú &#217	&Ugrave;	U maiúsculo com crase
ú &#218	&Uacute;	U maiúsculo com acento agudo
ú &#219	&Ucirc;	U maiúsculo com acento circunflexo
0 &#220	&Uuml;	U maiúsculo com trema
Ý &#221	&Yacute;	Y maiúsculo com acento agudo
Þ &#222	&THORN;	Thom maiúsculo islandês
ß &#223	&szlig;	Ligatura sz
à &#224	&agrave;	a minúsculo com crase
á &#225	&aacute;	a minúsculo com acento agudo
â &#226	&acirc;	a minúsculo com acento circunflexo
ã &#227	&atilde;	a minúsculo com til
ä &#228	&auml;	a minúsculo com trema
â &#229	&aring;	a minúsculo com anel
æ &#230	&aelig;	ligatura ae minúscula
ç &#231	&ccedil;	c cedilha minúsculo
è &#232	&egrave;	e minúsculo com crase
é &#233	&eacute;	e minúsculo agudo
ê &#234	&ecirc;	e minúsculo com acento circunflexo
ë &#235	&euml;	e minúsculo com trema
ì &#236	&igrave;	i minúsculo com crase
í &#237	&iacute;	i minúsculo com acento agudo
î &#238	&icirc;	i minúsculo com acento circunflexo
ï &#239	&iuml;	i minúsculo com trema
ð &#240	&eth;	eth minúsculo islandês
ñ &#241	&ntilde;	n minúsculo com til
ò &#242	&ograve;	o minúsculo com crase
ó &#243	&oacute;	o minúsculo com acento agudo
ô &#244	&ocirc;	o minúsculo com acento circunflexo
õ &#245	&otilde;	o minúsculo com til
ö &#246	&ouml;	o minúsculo com trema
÷ &#247		Sinal de divisão
ø &#248	&oslash;	o minúsculo com barra
ù &#249	&ugrave;	u minúsculo com crase
ú &#250	&uacute;	u minúsculo com acento agudo
û &#251	&ucirc;	u minúsculo com acento circunflexo
ü &#252	&uuml;	u minúsculo com trema
ý &#253	&yacute;	y minúsculo com acento agudo
þ &#254	&thorn;	thorn minúsculo islandês
ÿ &#255	&yuml;	y minúsculo com trema

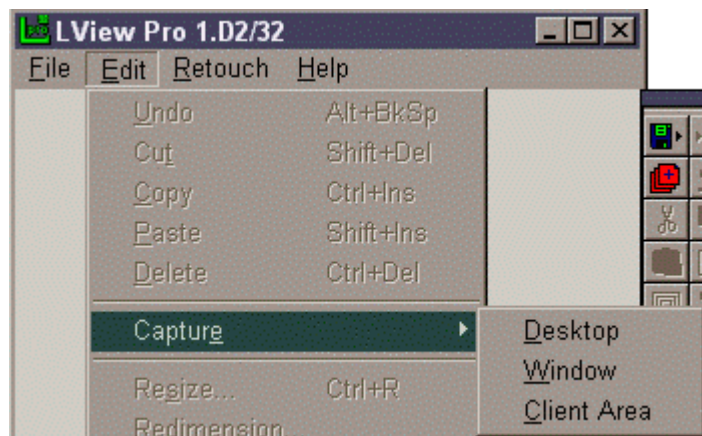
## Apêndice B – Manipulação de imagens

### Como capturar uma imagem

Através do Lview é possível capturar a imagem de uma janela ou de todo o conteúdo da tela:

1. Selecionar **Edit > Capture > Window**
2. Quando o Lview desaparecer clicar o mouse sobre a janela a capturar.

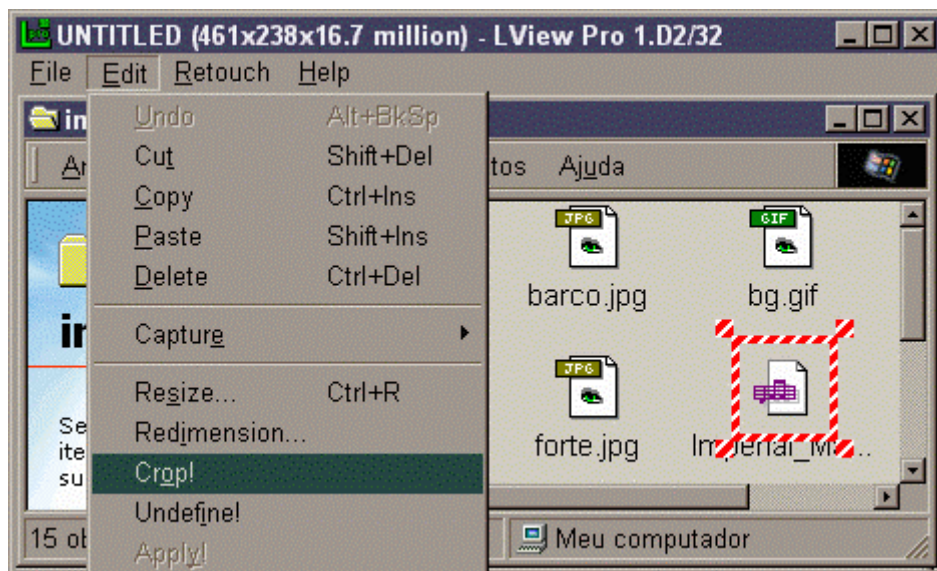
Além de Window há duas outras opções: Desktop (permite capturar toda a área do vídeo) e Client Area (permite capturar apenas a área útil de uma janela – sem a barra da títulos).



## Recorte de uma imagem

Ao se capturar uma janela, em muitos casos o que se deseja é apenas uma parte dela. É necessário então selecionar a parte da imagem que interessa e descartar o resto.

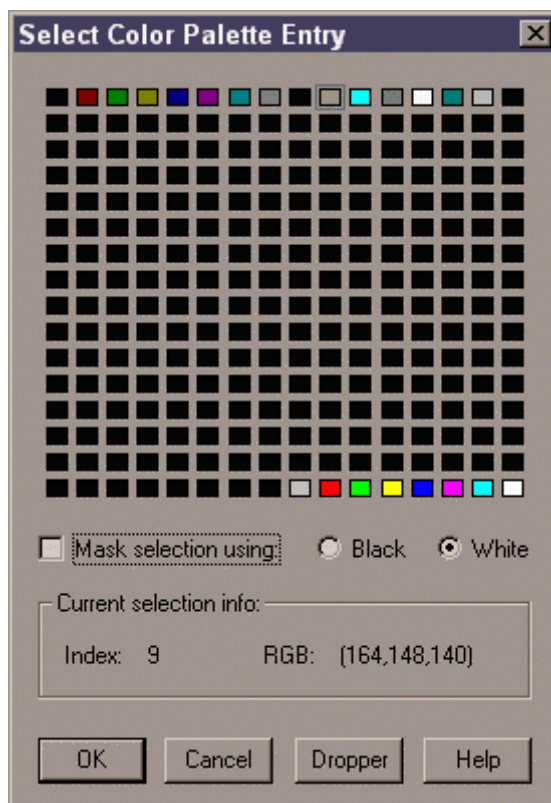
1. Pressionar o botão esquerdo do mouse no canto superior esquerdo da área de interesse
2. Com o botão pressionado, arrastar o mouse até o canto inferior direito da área de interesse e soltar o botão. Aparecerá um quadrado vermelho mostrando a área selecionada.
3. Selecionar **Edit > Crop!** e a figura ficará reduzida a área marcada.



## Figura com fundo transparente

No Lview:

1. Abrir o arquivo: **File > Open**.
2. Selecionar **Retouch > Color Depth...**.
3. Na janela que abrir selecionar Palette image e Custom number of colors. Se não souber o número de cores deixe 256 mesmo.
4. Selecionar **Retouch > Background Color**.
5. Será aberta uma janela que permite escolher qual cor ficará transparente na imagem.
6. Ao selecionar o botão **Dropper**, é possível selecionar a cor transparente clicando sobre um ponto da imagem.
7. Após fazer a seleção, é preciso salvar o arquivo no formato GIF: **File > SaveAs**. Na janela de seleção do nome de arquivo, mudar o campo "Salvar com o tipo" para Gif89a (.gif).





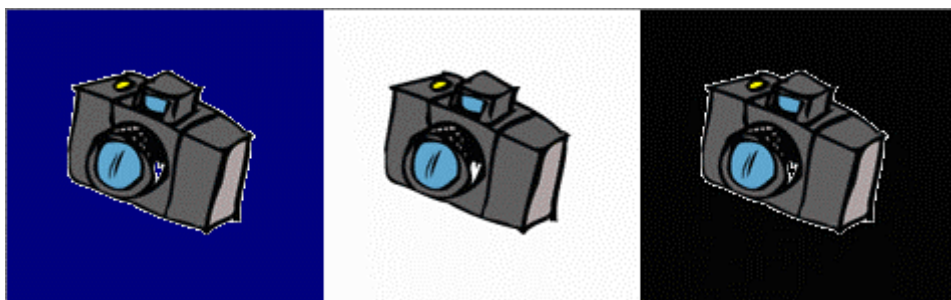
## Figuras transparentes bem acabadas

É comum encontrar páginas na Internet onde as figuras de fundo transparente apresentam linhas foscas residuais, normalmente brancas ou pretas, destoando da própria figura e do fundo da página. Estas linhas aparecem porque na criação da figura foi utilizado o recurso de suavização de serrilhado.

A suavização de serrilhado existe para criar uma área de transição entre a borda da figura e o fundo, disfarçando o fato das imagens na tela do computador serem formadas por pequenos pontos quadrados. Linhas inclinadas, sem a suavização de serrilhado, ficam com o aspecto de escada.

A figura a seguir mostra um mesmo desenho de máquina fotográfica em contraste com três diferentes cor de fundo. Nos dois fundos de cor escura a figura apresenta bordas residuais que estão disfarçadas no fundo claro. Isso ocorre pois a figura original tinha fundo de cor branca, que foi escolhida como cor de fundo e portanto não sendo desenhada.

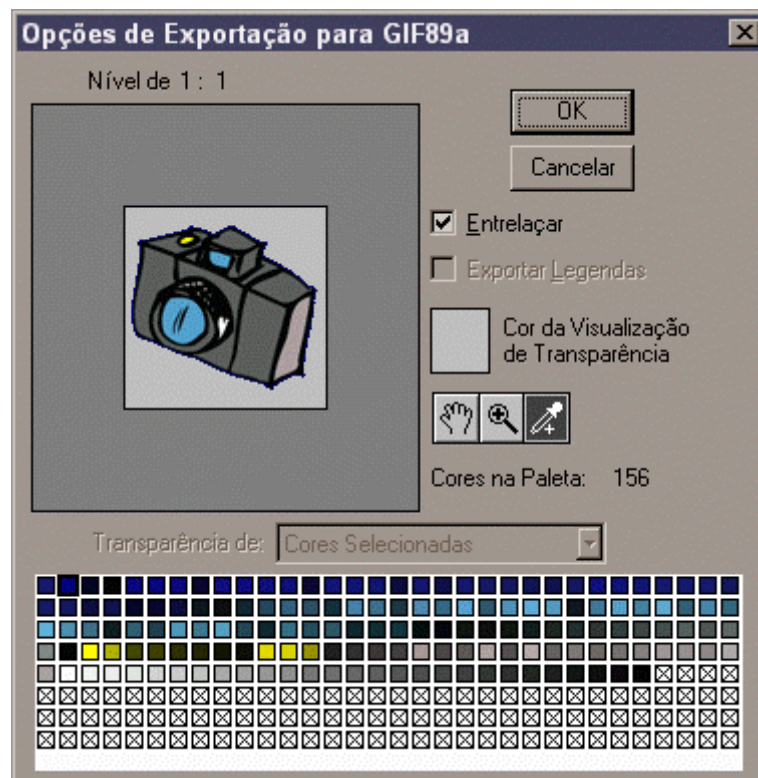
Por causa da suavização de serrilhado, há uma zona de transição entre o fundo branco e a cor preta da borda da máquina formada por pontos de cor intermediária entre o branco e o preto. Como no Lview só é possível definir uma cor como transparente, estes pontos da zona de transição continuam visíveis, contrastando com um fundo escuro.



O Photoshop oferece uma possível solução para este problema ao permitir definir várias cores como transparentes. Basta definir cada uma das cores que destoa da borda da figura como transparente.

Para se criar uma GIF transparente no Photoshop os seguintes passos devem ser seguidos:

1. Se estiver disponível selecionar: **Camada > Achatar imagem**
2. Se já não estiver selecionado: **Imagem > Modo > Cores indexadas**
3. Selecionar: **Arquivo > Exportar > Exportação para Gif89a**, o que causa a abertura da janela de seleção da cor de fundo.



Escolher várias cores de fundo só resolve o problema satisfatoriamente se nenhuma destas cores for utilizadas no interior da figura, o que faria com que parte do interior também ficasse transparente (este risco também ocorre quando escolhemos apenas uma cor – como se pode notar pelo detalhe da objetiva na figura da máquina fotográfica).

Uma outra opção para resolver o problema das linhas residuais é, antes de salvar a figura como GIF, pintar a área que vai ficar transparente com a cor de fundo da página onde se irá colocar a figura. Isto é feito no Photoshop da seguinte maneira:

1. Se a janela de ferramentas não estiver visível selecionar: **Janela > Mostrar ferramentas**
2. Selecionar na janela de ferramentas o balde de tinta.
3. Definir a cor do primeiro plano na janela de ferramentas
4. Se a janela de opções não estiver visível selecionar: **Janela > Mostrar opções.**
5. Selecionar **suavização de serrilhado.**
6. Modificar a tolerância de forma a desaparecer com a linha residual sem estragar a figura (tentativa e erro).
7. Clicar com o mouse sobre a área a pintar.



O resultado pode ser conferido acima. Note que com fundo branco a figura ficou serrilhada e com uma leve borda azul.

## Criação de um ladrilho

Utilizando os comandos de edição do Photoshop é possível criar um ladrilho a partir de uma figura qualquer. O exemplo abaixo mostra como isso pode ser feito. Obviamente os passos podem variar conforme a figura inicial:

1. Abrir logoHF.jpg: **Arquivo > Abrir** (fig. 1).
2. Selecionar um quadrado em volta do leão utilizando a seleção retangular da janela de ferramentas.
3. **Selecionar > inverter**
4. Apagar com a tecla del.
5. **Selecionar > desfazer**
6. Usar a borracha (janela de ferramentas) para apagar o escudo em volta do leão.
7. Selecionar a área de interesse da figura com a seleção retangular e eliminar o resto utilizando: **Imagem > corte demarcado** (fig. 2).
8. Tirar as cores: **Imagem > ajustar > Remover saturação** (fig. 3).
9. Modificar brilho e contraste até a figura ficar branca e preta: **Imagem > Ajustar > Brilho/Contraste** (fig. 4).
10. Inverter as cores da imagem: **Imagem > Ajustar > Inverter**.
11. Diminuir o contraste: **Imagem > Ajustar > Brilho/Contraste** (fig. 5).
12. Definir cor do primeiro plano: p. ex. 40, 100, 200.
13. Pintar a imagem selecionando: **Editar > Preencher**.
14. Na janela que se abrir mudar o modo para divisão (fig. 6).
15. Reduzir a imagem: **Imagem > tamanho da imagem**
16. **Selecionar > Tudo**
17. **Editar > Copiar**
18. Selecionar: **Arquivo > Novo** e definir tamanho como 200 x 200.
19. Preencher a cor de fundo.
20. Colar as figurinhas a gosto.



## Criação de um botão

No Photoshop é possível criar botões para incluir numa página. A seguir são descritos os passos para a criação de um botão:

1. Criar um arquivo novo de tamanho 80 x 40
2. Selecionar: **Camada > Nova > Camada.**
3. Usar a seleção para definir o área do botão.
4. Definir a cor do botão (p. ex. 255, 200, 10).
5. Pintar a área selecionada.
6. Selecionar: **Camada > Efeitos > Chanfro & entalhe.**
7. Na janela aberta definir (experimente valores a gosto):
  - a) Modo: Divisão; Modo: 85%; cor: 255,230,140
  - b) Modo: Multiplicação; Modo: 85%; cor: 175,124,7
  - c) Estilo: Chanfro interno
  - d) Ângulo:140
  - e) Profundidade: 15; abaixo
  - f) Desfoque: 5
8. **Selecionar > Desfazer seleção.**
9. Selecionar ferramenta de texto na janela de ferramentas.
10. Escrever o texto ajustando a cor e o tamanho do fonte forma a caber no botão e a ficar visível.
11. Mover o texto de forma a ficar centralizado no botão.
12. Salvar como GIF transparente.