Introdução à Linguagem HTML

HTML *(HyperText Markup Language - Linguagem de Formatação de Hipertexto)* é fruto do "casamento" dos padrões HyTime e SGML.;

HyTime - Hypermedia/Time-based Document Structuring Language

> Hy Time (ISO 10744:1992) - padrão para representação estruturada de hipermídia e informação baseada em tempo. Um documento é visto como um conjunto de eventos concorrentes dependentes de tempo (áudio, vídeo, etc.), conectados por webs ou hiperlinks. O padrão HyTime é independente dos padrões de processamento de texto em geral. Ele fornece a base para a construção de sistemas hipertexto padronizados, consistindo de documentos que alicam os padrões de maneira particular

SGML - Standard Generalized Markup Language

> Padrão ISO 8879 de formatação de textos: não foi desenvolvido para hipertexto, mas torna-se conveniente para transformar documentos em hiper-objetos e para descrever as ligações. SGML não é padrão aplicado de maneira padronizada: todos os produtos SGML têm seu próprio sistema para traduzir as etiquetas para um particular formatador de texto.

- **DTD - Document Type Definition** - define as regras de formatação para uma dada classe de documentos. Um DTD ou uma referência para um DTD deve estar contido em qualquer documento conforme o padrão SGML.

Portanto, HTML é definido segundo um DTD de SGML.
Todo documento HTML apresenta elementos entre parênteses angulares (**<** e **>**); esses elementos são as *etiquetas* (*tags*) de HTML, que são os comandos de formatação da linguagem. A maioria das etiquetas tem sua correspondente de fechamento:

```
<etiqueta>...</etiqueta>
```

Isso é necessário porque as etiquetas servem para definir a formatação de uma porção de texto, e assim marcamos onde começa e termina o texto com a formatação especificada por ela.
Alguns elementos são chamados “vazios”, pois não marcam uma região de texto, apenas inserem alguma coisa no documento:

```
<etiqueta>
```

Todos os elementos podem ter atributos:

```
<etiqueta atributo1=valor1 atributo2=valor2>...</etiqueta>
```

HTML é um recurso muito simples e acessível para a produção de documentos. Nestes “capítulos”, será possível aprender grande parte de seus elementos

Edição de documentos HTML

Os documentos em HTML são como arquivos ASCII comuns, que podem ser editados em **vi**, **emacs** (que já tem versão para editar arquivos HTML), **textedit**, ou qualquer editor simples.
Para facilitar a produção de documentos, existem editores HTML específicos:

- Editores de texto fonte

  - inserem automaticamente as etiquetas, orientando a inserção de atributos e marcações.

- Editores WYSIWYG

  - oferecem ambiente de edição com “um” resultado final das marcações.

O documento produzido terá sempre extensão `.html` (para servidores Web em ambiente UNIX).

Publicação de documentos

Para ter sua homepage é necessário ter uma área na rede; o primeiro passo é criar um diretório **WWW** (em letras maiúsculas) em sua área de rede. A partir do momento da criação desse diretório WWW, o URL:
`http://www.icmc.sc.usp.br/~seu login/`
passará a ser acessível pelo servidor. Nesse diretório deve haver um arquivo **index.html**.
(Veja mais sobre o arquivo index.html na coluna ao lado.)
A seguir, certifique-se de que sua área e o diretório WWW dentro dela estejam com permissão de leitura para "todo mundo".
Feito isso, basta contactar os Webmasters para que sua página seja referenciada através da lista de "pessoas com página", ou da página do seu grupo de pesquisa.

Obs.: Estas diretivas se aplicam em particular à rede do ICMC; outros sistemas podem ter outras configurações, bem diferentes.

Documento básico e seus componentes

A estrutura de um documento HTML apresenta os seguintes componentes:
<HTML>
<HEAD><TITLE>**Titulo do Documento**</TITLE></HEAD>
<BODY>
texto,
imagem,
links,
...
</BODY>
</HTML>
As etiquetas HTML **não são sensíveis à caixa**. Traduzindo: tanto faz escrever `<HTML>`, `<Html>`, `<html>`, `<HtMl>`, ...
Os documentos se dividem em duas seções principais, que veremos a seguir.

A Seção <HEAD>

<HEAD> contém informações sobre o documento; o elemento **<TITLE>**, por exemplo, define um título, que é mostrado no alto da janela do browser. Nesta página, por exemplo, está definido assim:
`<HEAD><TITLE>Documento basico em HTML</TITLE></HEAD>`
Todo documento WWW deve ter um título; esse título é referenciado em buscas pela rede, dando uma identidade ao documento.
É sugerido que os títulos dos documentos sejam sugestivos, evitando-se, portanto, títulos como ***"Introducao"***. De preferência, os títulos não devem conter acentos ou outros caracteres especiais (eles não serão mostrados corretamente pelos sistemas de janela em algumas plataformas).
Além do título, `HEAD` contém outras informações que podem ser recuperadas por robôs de pesquisa na Internet; esses campos de informação facilitam a classificação do documento em catálogos de busca, entre outras aplicações.

A Seção <BODY>

Tudo que estiver contido em **<BODY>** será mostrado na janela principal do browser, sendo apresentado ao leitor.
<BODY> pode conter cabeçalhos, parágrafos, listas, tabelas, links para outros documentos, e imagens.
Veja um documento básico em HTML.
<BODY> tem alguns atributos de apresentação que são aplicados ao documento

Atributos gerais de um documento

Como visto anteriormente (documento básico e seus componentes), <HEAD> contém informações sobre o documento. Além de <TITLE>, existem diversos outros campos de informação, sendo os campos <META> os mais usados.

Campos <META>

Os campos <META> têm dois atributos principais:

- `NAME`, indicando um nome para a informação
- `HTTP-EQUIV`, que faz uma correspondência com campos de cabeçalho do protocolo HTTP; a informação desse campo pode ser lida pelos browsers, e provocar algumas ações.

```
<HEAD>

<TITLE>Titulo do Documento</TITLE>

<META NAME="nome" CONTENT="valor">

<META HTTP-EQUIV="nome" CONTENT="valor">

</HEAD>
```

Este documento, por exemplo, tem as seguintes informações:

```
<HEAD>

<TITLE>Atributos de documentos em HTML</TITLE>

<meta name="Author" content="Maria Alice Soares de Castro
(masc@icmc.sc.usp.br)">

<META NAME="Description" CONTENT="Tutorial basico-avancado para suporte `a
editoracao de documentos Web.">

<META NAME="KeyWords" CONTENT="HTML, WWW, Webpublishing">

<META NAME="Editor" CONTENT="W3e - 5.22c">

</HEAD>
```

Alguns desses atributos são inseridos automaticamente pelos editores.
Um exemplo de uso do atributo `HTTP-EQUIV` é promover a mudança automática de páginas. Veja este exemplo (funciona com Netscape, Internet Explorer 2.0 em diante, e Mosaic 2.0 em diante - o Mosaic avisa que um novo documento estará sendo carregado automaticamente e pede sua permissão para fazê-lo)

Agora que você voltou do exemplo, veja como esse efeito é conseguido:

```
<HEAD>

<TITLE> ... </TITLE>

<META HTTP-EQUIV="Refresh" CONTENT="segundos; URL= pagina.html">

</HEAD>
```

onde:

pagina.html

é a página a ser carregada automaticamente

segundos

é o número de segundos passados até que a página indicada seja carregada.

Como foi comentado no exemplo, o efeito é interessante, mas para que serve? Se não pensamos em uma finalidade útil para esse efeito, caímos na tentação de usá-lo "à toa".
A aplicação mais utilizada é a atualização automática de um documento que, por exemplo, tenha uma foto produzida por uma câmara de vídeo: pode-se forçar, com o refresh, a atualização dessa página, mostrando para o leitor sempre uma imagem mais atual de algum evento sendo focalizado pela câmara. Outra utilização é em "chats", ou em páginas que desviem a navegação por documentos desenvolvidos para browsers avançados.

Atributos de <BODY>

Através de atributos de <BODY>, podemos definir cores para os textos, links e para o fundo das páginas, bem como uma imagem de fundo (marca d'água):

```
<BODY BGCOLOR="#rrggbb" TEXT="#rrggbb" LINK="#rrggbb" ALINK="#rrggbb"
VLINK="#rrggbb" BACKGROUND="URL">
```

onde:

BGCOLOR

cor de fundo (padrão: cinza ou branco)

TEXT

cor dos textos da página (padrão: preto)

LINK

a cor dos links (padrão: azul)

ALINK

cor dos links, quando acionados (padrão: vermelho)

VLINK

cor dos links, depois de visitados (padrão: azul escuro ou roxo)

Seus valores são dados em valores hexadecimais, equivalentes a cores no padrão RGB (`Red`, `Green`, `Blue`). Existem tabelas de cores com esses valores, mas grande parte dos editores já oferece uma interface bem amigável através da qual escolhemos as cores desejadas, sem nos preocuparmos com números esdrúxulos tais como `#FF80A0`.

Browsers que seguem a definição de HTML 3.2 também aceitam 16 nomes de cores, tirados da paleta VGA do Windows - por exemplo, podemos escrever **BGCOLOR**=`"BLUE"`. Porém, browsers mais antigos não apresentarão as cores indicadas.

Este documento tem a seguinte definição de atributos gerais:

```
<BODY BGCOLOR="#FFFFFF" LINK="#008000" VLINK="#000000" ALINK="#FFFF00">
```

BACKGROUND

indica o URL da imagem a ser replicada no fundo da página, como uma marca d'água. Veja o exemplo de uma página cuja imagem de fundo é .

Cabeçalhos

Há seis níveis de cabeçalhos em HTML, de **<H1>** a **<H6>**:

```
<H1>Este é um cabeçalho de nível 1</H1>
<H2>Este é um cabeçalho de nível 2</H2>
<H3>Este é um cabeçalho de nível 3</H3>
<H4>Este é um cabeçalho de nível 4</H4>
<H5>Este é um cabeçalho de nível 5</H5>
<H6>Este é um cabeçalho de nível 6</H6>
```

Esses cabeçalhos são mostrados da seguinte forma:

Este é um cabeçalho de nível 1

Este é um cabeçalho de nível 2

Este é um cabeçalho de nível 3

Este é um cabeçalho de nível 4

Este é um cabeçalho de nível 5

Este é um cabeçalho de nível 6

Os cabeçalhos não podem ser aninhados, isto é:

```
<H2>Este é <H1>um cabeçalho de nível 1</H1> dentro de um cabeçalho de nível 2</H2>
```

Embora em alguns browsers esse aninhamento possa dar um resultado (por exemplo,

Este é

um cabeçalho de nível 1

dentro de um cabeçalho de nível 2

), o aninhamento de cabeçalhos não é previsto pela definição de HTML.
É bom notar que o título do documento **não precisa ter necessariamente** o mesmo texto do cabeçalho principal.

Os cabeçalhos têm atributos de alinhamento:

```
<H2 ALIGN=CENTER>Cabeçalho centralizado</H2>
```

Cabeçalho centralizado

```
<H3 ALIGN=RIGHT>Cabeçalho alinhado à direita</H3>
```

Cabeçalho alinhado à direita

```
<H4 ALIGN=LEFT>Cabeçalho alinhado à esquerda (default)</H4>
```

Cabeçalho alinhado à esquerda (default)

Separadores

Como vimos no primeiro exemplo, as quebras de linha do texto fonte não são significativas na apresentação de documentos em HTML. Para organizar os textos, precisamos de separadores, apresentados aqui.

Quebra de linha

Quando queremos mudar de linha, usamos o elemento **
**. Isso só é necessário quando queremos uma quebra de linha em determinado ponto, pois os browsers já quebram as linhas automaticamente para apresentar os textos.
Com sucessivos **
**, podemos inserir diversas linhas em branco nos documentos. Esse elemento tem um atributo especial, que será apresentado no item sobre inserção de imagens.

Parágrafos

Para separar blocos de texto, usamos o elemento **<P>**:

```
Parágrafo 1;<P>Parágrafo 2.
```

que produz:
Parágrafo1;
Parágrafo2.
Combinando parágrafos e quebras de linha, temos:

```
Parágrafo 1;<br> linha 1 do parágrafo 1, <br>linha 2 do parágrafo
1.<P>Parágrafo 2;<br> linha 1 do parágrafo 2, <br>linha 2 do parágrafo 2.
```

O resultado da marcação acima é:
Parágrafo 1;
linha 1 do parágrafo 1,
linha 2 do parágrafo 1.
Parágrafo 2;
linha 1 do parágrafo 2,
linha 2 do parágrafo 2.

<P> tem atributo de alinhamento, como os cabeçalhos:

```
<P ALIGN=CENTER>Assim como os trens, as boas idéias às vezes chegam com
atraso. <BR>(Giovani Guareschi)</P>
```

Assim como os trens, as boas idéias às vezes chegam com atraso.
(Giovani Guareschi)

```
<P ALIGN=RIGHT>Como diz o provérbio chinês: "É melhor passar por ignorante
uma vez do que permanecer ignorante para sempre".</P>
```

Como diz o provérbio chinês: "É melhor passar por ignorante uma vez do que permanecer ignorante para sempre".

```
<P ALIGN=LEFT>Este é o alinhamento padrão (default), e por isso não vou
colocar nenhuma frase especial.</P>
```

Este é o alinhamento padrão (default), e por isso não vou colocar nenhuma frase especial.

Linha Horizontal

<HR> insere uma linha horizontal:

Essa linha tem diversos atributos, oferecendo resultados diversos.
<HR **SIZE**=7> insere uma linha de largura 7 (pixels):

<HR **WIDTH**=50%> insere uma linha que ocupa 50% do espaço disponível:

<HR **WIDTH**=30% **ALIGN**=RIGHT **NOSHADE**> insere uma linha de comprimento 30% (do espaço disponível), alinhada à direita, sem efeito tridimensional:

<HR **SIZE**=30 **WIDTH**=2 **ALIGN**=LEFT> insere uma linha de largura 30, comprimento 2, alinhada à esquerda:

Listas em HTML

Há vários tipos de listas em HTML, sendo estas as mais usadas e corretamente apresentadas pelos browsers:

Listas de Definição

Estas listas são chamadas também "Listas de Glossário", uma vez que têm o formato:

```
<DL>
<DT>termo a ser definido
<DD>definição
<DT>termo a ser definido
<DD>definição
</DL>
```

Que produz:

termo a ser definido

definição

termo a ser definido

definição

Este tipo de lista é muito utilizado para diversos efeitos de organização de páginas, por permitir a tabulação do texto. Um exemplo são os índices de navegação presentes nas pági nas deste tutorial; outro exemplo é a lista composta abaixo:

```
<DL>
```

```
<DT>Imperadores do Brasil:

<DD>D. Pedro I

   <DL>

   <DD>Nome completo: Pedro de Alcântara Francisco Antônio João Carlos Xavier
   de Paula Miguel Rafael Joaquim José Gonzaga Pascoal Cipriano Serafim de
   Bragança e Bourbon

   </DL>

<DD>D. Pedro II

   <DL>

   <DD>Nome completo: Pedro de Alcântara João Carlos Leopoldo Salvador Bibiano
   Francisco Xavier de Paula Leocádio Miguel Gabriel Rafael Gonzaga

   </DL>

</DL>
```

Imperadores do Brasil:

D. Pedro I

Nome completo: Pedro de Alcântara Francisco Antônio João Carlos Xavier de Paula Miguel Rafael Joaquim José Gonzaga Pascoal Cipriano Serafim de Bragança e Bourbon

D. Pedro II

Nome completo: Pedro de Alcântara João Carlos Leopoldo Salvador Bibiano Francisco Xavier de Paula Leocádio Miguel Gabriel Rafael Gonzaga

http://www.icmc.sc.usp.br/
manuals/HTML/
arquivo: listas.html

Listas não-numeradas

São equivalentes às listas com marcadores do MS Word:

```
<UL>

<LI>item de uma lista

<LI>item de uma lista, que pode ser tão grande quanto se queira, sem que seja
necessário se preocupar com a formatação das margens de texto

<LI>item

</UL>
```

- item de uma lista

- item de uma lista, que pode ser tão grande quanto se queira, sem que seja necessário se preocupar com a formatação das margens de texto
- item

A diferença está na mudança dos marcadores, assinalando os diversos níveis de listas compostas:

```
<UL>

<LI>Documentos básicos

<LI>Documentos avançados

   <UL>

   <LI>formulários

   <UL>

   <LI>CGI

   </UL>

   <LI>contadores

   <LI>relógios

   </UL>

<LI>Detalhes sobre imagens

</UL>
```

- Documentos básicos
- Documentos avançados
    - formulários
        - CGI
    - contadores
    - relógios
- Detalhes sobre imagens

Segundo o HTML 3.2, essa lista pode ter marcadores diferentes, indicados através do atributo `TYPE`, que assume os valores `CIRCLE`, `SQUARE` e `DISC` (default):

```
<UL TYPE=CIRCLE>

<LI>um item

<LI>mais um item

</UL>
```

- ☐ um item
- ☐ mais um item

Cada item também pode ter seu atributo específico:

```
<UL>

<LI TYPE=DISC>um item

<LI TYPE=CIRCLE>mais um item

<LI TYPE=SQUARE>último item
```

```
</UL>
```

- um item
- mais um item
- último item

Observação 1: Boa parte dos editores HTML (WYSIWYG ou não), insere marcações que não existem em listas. Exemplos típicos são `</DD>`, `</DT>` ; e `</LI>`. Porém, como essas etiquetas não são reconhecidas pelos browsers, não causam efeito colateral algum nos documentos.

Observação 2: Nestes exemplos, o texto fonte aparece tabulado apenas para efeito de melhor visualização, uma vez que já foi visto que os espaços em branco e tabulações originais não têm efeito no documento final.

http://www.icmc.sc.usp.br/
manuals/HTML/
arquivo: listas.html

http://www.icmc.sc.usp.br/
manuals/HTML/
arquivo: listas.html

Observação 3: Se você não está vendo diferença alguma entre as listas comuns e as que têm atributos de HTML 3.2, isso se deve ao fato de seu browse r não estar reconhecendo esses atributos como

válidos. Trata-se de um browser de versão antiga. Isso deve ser pensado quando usamos atributos mais recentes: *nem todo usuário poderá ver o resultado das novas marca&cced il;ões*.

Listas Numeradas

```
<OL>

<LI>item de uma lista numerada

<LI>item de uma lista numerada, que pode ser tão grande quanto se queira, sem
que seja necessário se preocupar com a formatação das margens de texto

<LI>item de lista numerada

</OL>
```

1. item de uma lista numerada
2. item de uma lista numerada, que pode ser tão grande quanto se queira, sem que seja necessário se preocupar com a formatação das margens de texto
3. item de lista numerada

Estas listas não apresentam numeração em formato `1.1`, `1.2` etc., quando compostas:

1. Documentos básicos
2. Documentos avançados
    1. formulários
        1. CGI
        2. contadores
        3. relógios
    2. Detalhes sobre imagens

Porém, através do atributo `TYPE` (HTML 3.2), pode-se lidar com a numeração dos itens:

```
<OL TYPE=I>

<LI>Documentos básicos

<LI>Documentos avançados

   <OL TYPE=a>

   <LI >formulários

   <OL TYPE=i>

   <LI>CGI

   </OL>

   <LI>contadores

   <LI>relógios

   </OL>

<LI>Detalhes sobre imagens

</OL>
```

I. Documentos básicos
II. Documentos avançados
    a. formulários

i. CGI
b. contadores
c. relógios

III. Detalhes sobre imagens

Ainda segundo HTML 3.2, o atributo `START` pode indicar o início da numeração da lista:

```
<OL START=4 TYPE=A>

<LI>um item

<LI>outro item

<LI>mais um item

</OL>
```

D. um item
E. outro item
F. mais um item

http://www.icmc.sc.usp.br/
manuals/HTML/
arquivo: listas.html

Listas e "sub-listas"

As listas podem ser aninhadas. Por exemplo:

```
<DL>
<DT>termo a ser definido
<DD>definição
   <OL>
   <LI>item de uma lista numerada
   <LI>item de uma lista numerada
      <UL>
      <LI>item de uma lista
      </UL>
   <LI>item de uma lista numerada
   </OL>
<DT>termo a ser definido
<DD>definição
</DL>
```

termo a ser definido

definição

1. item de uma lista numerada
2. item de uma lista numerada
   - item de uma lista
3. item de uma lista numerada

termo a ser definido

definição

Formatação de textos e caracteres

Há dois tipos de formatação em HTML: ***lógico*** e ***físico***. Os efeitos de apresentação na tela são os mesmos: o motivo da distinção entre eles se deve à idéia básica de independência entre especificação e apresentação.
Quando formatamos um trecho de texto como cabeçalho de nível 1, não explicitamos se esse tipo de cabeçalho deve ser em alguma fonte determinada, em um tamanho determinado, justificado à esquerda ou à direita, ou centralizado. Esses detalhes de apresentação são deixados para o browser - o dispositivo de apresentação do documento - que pode ser configurado de acordo com o leitor (usuário final).
Desse modo, além de facilitar enormemente o trabalho de quem escreve os documentos, a linguagem garante a uniformidade de apresentação de cabeçalhos, parágrafos, listas, etc.
A formatação lógica segue o significado lógico do texto marcado: um endereço de e-mail, uma citação etc. Sua apresentação final varia conforme o browser, podendo oferecer resultados mais ricos.
A formatação física especifica explicitamente o estilo que se quer para o texto: itálico, grifado etc. Sua apresentação final não sofre grandes variações.

Blocos de texto

HTML oferece as seguintes formatações de blocos de texto:

<PRE>

Apresenta o texto na mesma maneira em que foi digitado, mantendo quebras de linha e tabulações:

```
<pre>uma   linha        aqui,
outra ali,
  etc.</pre>
```

Resulta em:

```
uma       linha    aqui,
outra ali,
   etc.
```

Uma vez que `<PRE>` mantém o texto original, não se deve forçar espaços com essa marcação dentro de outra marcação que já apresente tabulações e espaços específicos.
Até o momento, somente como uso de `<PRE>` é possível gerar parágrafos com texto justificado à esquerda e à direita.
Veja este exemplo, que apresenta também alguns problemas com o uso de `<PRE>`.

<BLOCKQUOTE>

É usado para citações longas:

```
<blockquote>A massa do Sol arqueia o espaço-tempo de tal maneira que, ainda que a Terra siga uma trajetória reta no espaço-tempo quadridimensional, parece-nos que se desloca em órbita circular no espaço tridimensional.</blockquote> (Stephen W. Hawking, "Uma Breve História do Tempo")
```

> A massa do Sol arqueia o espaço-tempo de tal maneira que, ainda que a Terra siga uma trajetória reta no espaço-tempo quadridimensional, parece-nos que se desloca em órbita circular no espaço tridimensional.

(Stephen W. Hawking, "Uma Breve História do Tempo")

<ADDRESS>

Usado para formatar endereços E-mail e referências a autores de documentos:

Envie críticas e sugestões para <address>masc@icmc.sc.usp.br </address>
Envie críticas e sugestões para

masc@icmc.sc.usp.br

Formatação de frases

Como visto anteriormente (em Formatação de Textos e Caracteres), HTML permite dois tipos de formatação: ***lógico*** e ***físico***; aqui veremos as formatações mais utilizadas:

Estilos Lógicos

<CITE>

Para títulos de livros, filmes, e citações curtas. Exemplo:

Assisti *Guerra nas Estrelas* umas oito vezes!

<CODE>

Para indicar trechos de código de programas. Exemplo:

```
for (x=0); cl &&(!feof(stdin)); x++));
```

<DFN>

Indica definição de uma palavra, em geral apresenta o texto em itálico. Exemplo:

CERN: *Centre d'Éstudes et Recherches Nucleaires*

<EM>

Ênfase, também normalmente apresentado em itálico. Exemplo:

É preciso pesquisar *muito* para encontrar o termo exato.

<KBD>

Indica uma entrada via teclado. Exemplo:

Para ler mensagens recebidas, digite **`pine -i`**

<SAMP>

Indica uma seqüência de caracteres, por exemplo uma mensagem de erro ou um resultado. Exemplo:

O resultado do primeiro applet é: `Hello, World!`

<STRONG>

Forte ênfase, mostrado normalmente em negrito. Exemplo:

Antes de enviar um e-mail, **confira o campo "Subject:"**!

<VAR>

Indica variáveis, ou valores que o usuário deverá escrever; geralmente mostrado em itálico. Exemplo:

> No campo `Login`, escreva *guest*.

Estilos Físicos

<B>

Quando disponível no browser, é mostrado em **negrito** (em alguns browsers, pode aparecer sublinhado)

<I>

Itálico (em alguns casos, caracteres apenas inclinados)

<TT>

Tipo `teletype` - fonte de espaçamento fixo.

<U>

Sublinhado; deve ser usado com cuidado, pois confunde-se com a apresentação de *links*.

<STRIKE> ou **<S>**

Frase ~~riscada~~.

<BIG>

Fonte um pouco maior.

<SMALL>

Fonte um pouco menor.

<SUB>

Frase em estilo $_{\text{índice}}$, como em H_2O.

<SUP>

Frase em estilo $^{\text{expoente}}$, como em Km^2.

Caracteres Especiais

HTML permite que caracteres especiais sejam representados por sequências de escape, indicadas por três partes: um `&` inicial, um número ou cadeia de caracteres correspondente ao caracter desejado, e um `;` final.
Quatro caracteres ASCII - <, >, e & têm significados especiais em HTML, e são usados dentro de documentos seguindo a correspondência:

Entidade	Caracter
<	<
>	>
&	&

Outras sequências de escape suportam caracteres *ISO Latin1*. Temos aqui uma tabela com os caracteres mais utilizados em Português:

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

Como vemos, as sequências de escape são sensíveis à caixa.

Ao usar caracteres acentuados, pode-se inserir uma indicação do esquema de codificação *ISO Latin1*, escrevendo:

```
<HTML>

<HEAD>

<TITLE>...</TITLE>

<META HTTP-EQUIV="Content-Type"

   CONTENT="text/html; charset=ISO-8859-1">

</HEAD>

...
```

Existem alguns símbolos que vêm sendo incorporados ao conjunto de caracteres reconhecidos em HTML 3.2. Por exemplo, `&copy;`, que é o símbolo ©, `&reg;` para ®, e `&sect;` para §.
Também se pode usar seqüências com códigos ASCII, por exemplo:

```
&#191;Qué pasa, señor?
```

¿Qué pasa, señor?

Cores e Fontes

Cores

As cores são introduzidas através do elemento <FONT>, usando o sistema RGB para cores (da mesma forma que vimos para cores de documentos):
<FONT **COLOR**="#*rrggbb*">Texto</FONT>
Assim, um trecho de texto pode ter uma cor diferente da definição geral de cores, feita através dos atributos de <BODY>.

Tamanho

A formatação
<FONT **SIZE**=*tamanho_da_letra*>Texto</FONT>
permite que o autor do documento altere o tamanho das letras em trechos específicos de texto. O tamanho básico dos textos é 3. Podemos indicar tamanhos relativos a esse, por exemplo:

```
<FONT SIZE=+2>Letra maior</FONT>
Letra normal
<FONT SIZE=-2>Letra menor</FONT>
```

Letra maior Letra normal Letra menor

Fontes

Uma evolução que permite a escolha da fonte para os textos, é o atributo FACE:
<FONT **FACE**=*fonte_da_letra*>Texto</FONT>
Por exemplo:

```
<FONT FACE=Times COLOR="#0000AA">Fonte Times azul</FONT>
```

Fonte Times azul

```
<FONT FACE=Arial COLOR="#00AA00">Fonte Arial
verde</FONT>
```

Fonte Arial verde

```
<FONT FACE=Courier COLOR="#AA0000">Fonte Courier
vermelha</FONT>
Fonte Courier vermelha
```

Blink

Blinks são um perigo!

A formatação <BLINK>frase</BLINK> foi uma das primeiras inovações introduzidas pelo Netscape. O **perigo** de se usar o BLINK é que, se sua página já apresenta cores, desenhos, cabeçalhos, todos efeitos que chamam a atenção do leitor, o BLINK será ainda mais um fator chamativo, o que causa um efeito final cansativo e confuso.
Evite usar o BLINK; ao usá-lo, aplique somente em pequenos detalhes (palavras ou flechinhas), nunca em grande número, muito menos em frases inteiras ou cabeçalhos, como aqui!
E, se você já está ficando **neurótico** com o pisca-pisca dos exemplos de BLINK desta página, melhor mudar para outro assunto! ;-)

Marquee

É possível obter o efeito de animação de texto, através da formatação <MARQUEE>.

<MARQUEE **BEHAVIOR**=*efeito*>Texto</MARQUEE>
Atributos de largura e direção do efeito permitem diversas apresentações diferentes. Por exemplo (o efeito só é visto através do Internet Explorer):

<MARQUEE **BEHAVIOR**=SCROLL **WIDTH**=30%>Texto</MARQUEE>

<MARQUEE **BEHAVIOR**=SLIDE **DIRECTION**=RIGHT>Texto</MARQUEE>

<MARQUEE **BEHAVIOR**=SLIDE **DIRECTION**=LEFT>Texto</MARQUEE>

Ligações (uso de links)

Com HTML é possível fazermos ligações de uma região de texto (ou imagem) a um outro documento. Nestas páginas, temos visto exemplos dessas ligações: o browser destaca essas regiões e imagens do texto, indicando que são ligações de hipertexto - também chamadas *hypertext links* ou *hiperlinks* ou simplesmente *links*.
Para inserir um link em um documento, utilizamos a etiqueta **<A>**, da seguinte forma:
<A **HREF** = "*arq_dest*">âncora</A>
onde:

arq_dest

é o URL do documento de destino;

âncora

é o texto ou imagem que servirá de ligação hipertexto do documento sendo apresentado para o documento de destino.

Caminhos para o documento de destino

Caminho relativo

O caminho relativo pode ser usado sempre que queremos fazer referência a um documento que esteja no mesmo servidor do documento atual.
Através do campo *Location* do browser, vemos que este documento está localizado em um diretório `/manuals/HTML/` do servidor `www.icmc.sc.usp.br`. Para escrevermos um link deste documento para o documento `doc2.html` no diretório `/manuals/HTML/exemplos`, tudo que precisamos fazer é escrever:
`Veja o <A HREF="exemplos/doc2.html">exemplo de caminho relativo</A>.`
que é apresentado como:
Veja o exemplo de caminho relativo.
Da mesma forma, se quisermos um link deste documento para um outro que esteja em diretório diferente neste mesmo servidor, escrevemos, por exemplo:
`<A HREF="/Portugues/ICMC/">Instituto de Ciências Matemáticas e de Computação</A>`

que produz o link: Instituto de Ciências Matemáticas e de Computação
Para usar links com caminhos relativos é preciso, portanto, conhecer a estrutura do diretório do servidor no qual estamos trabalhando.
O esquema do diretório de nosso servidor está disponível no **Relatório nº. 35** e no relatório do servidor Web (em final de preparação).

Caminho absoluto

Utilizamos caminho absoluto quando desejamos referenciar um documento que esteja em outro servidor, por exemplo:

```
<A HREF="http://www.intermidia.icmc.sc.usp.br/">Grupo Intermídia</A>
```

que oferece um link para um documento no servidor WWW do Grupo de Pesquisa Intermídia:
Grupo Intermídia
Com a mesma sintaxe, é possível escrever links para qualquer servidor de informações da Internet.

Ligações para trechos de documentos

Além do atributo `href`, que indica um documento destino de uma ligação hipertexto, o elemento `A` possui um atributo `NAME` que permite indicar um trecho de documento como *ponto de chegada* de uma ligação hipertexto.
Neste documento temos diversos parágrafos marcados como chegada de um link, por exemplo:

```
<h3><A NAME="relativo">Caminho relativo</A></h3>
```

que faz com que a âncora **Caminho relativo** seja o destino de um link. Se escrevermos:

```
Leia sobre <A HREF="#relativo">caminhos relativos</A>.
```

teremos uma ligação hipertexto para um trecho deste mesmo documento:
Leia sobre caminhos relativos.
Da mesma forma, construímos links para trechos determinados de outros documentos, desde que saibamos quais trechos do documento destino estão marcados para ponto de chegada de um link. Por exemplo:

```
São Carlos é um <A HREF= "/Portugues/Sao_Carlos/histprog.html#polo">pólo de alta tecnologia</A>.
```

que produz um link para um parágrafo marcado no arquivo `histprogr.html` sobre a cidade de São Carlos, no diretório **/Portugues/Sao_Carlos/**:
São Carlos é um pólo de alta tecnologia.

Inserção de Imagens

O elemento **IMG** insere imagens que são apresentadas junto com os textos. Um atributo `SRC` deve estar presente, da seguinte forma:

```
<IMG SRC="URL_imagem">
```

onde **URL_imagem** é o URL do arquivo que contém a imagem que se quer inserir; pode ser referenciada uma imagem que esteja em um outro servidor (o que, logicamente, não é conveniente).

Assim, escrevendo:

```
<IMG SRC = "/icones/newred.gif">
```

inserimos a figura no documento.

As imagens usadas na Web são armazenadas em arquivos com extensão `*.gif`, `*.xbm`, `*.jpg` (ou `*.jpeg`).

Atributos básicos

ALT

Indica um texto alternativo, descrevendo brevemente a imagem, que é apresentado no lugar da imagem nos browsers texto, ou quando se desabilita o carregamento de imagens em browsers gráficos. É recomendável que esteja sempre presente.

```
<IMG SRC="URL_imagem" ALT="descrição_da_imagem">
```

`<IMG SRC="/icones/newred.gif" ALT="Novo!">` é apresentado nos browsers gráficos assim: - e, nos browsers texto, assim: `[Novo!]`

WIDTH e HEIGHT

Atributos de dimensão da imagem, em pixels. Grande parte dos editores HTML coloca automaticamente os valores destes atributos, quando indicamos a inserção de uma imagem.

```
<IMG SRC="imagem" ALT="descrição" WIDTH="largura" HEIGHT="altura">
```

Uma das vantagens de se usar esses atributos é que o browser pode montar mais rapidamente as páginas, por saber de antemão o espaço que deverá ser reservado a elas.

BORDER

Quando uma frase é marcada como âncora de um link, ela se apresenta sublinhada; quando uma imagem faz as vezes de âncora, ganha uma borda que indica sua condição. Por exemplo: bordaborda

Porém, por questões de apresentação, nem sempre interessa termos essa borda ao redor da imagem. Assim, com o atributo `BORDER`, podemos controlar esse detalhe.

Se quisermos uma borda maior... bordaborda

```
<A HREF="URL"><IMG SRC="imagem" ALT="descrição" BORDER=4></A>
```

Se quisermos uma imagem sem borda... bordaborda

```
<A HREF="URL"><IMG SRC="imagem" ALT="descrição" BORDER=0></A>
```

ALIGN

<IMG **SRC**="*imagem*" **ALT**="*descrição*" **ALIGN**=*alinhamento*>
Existem também atributos de alinhamento, que produzem os seguintes resultados:
ALIGN=TOP Alinha o texto adjacente com o topo da imagem, embora com linhas compridas o resultado não seja muito bom.
ALIGN=MIDDLE Alinha o texto adjacente com o meio da imagem, embora com linhas compridas o resultado não seja muito bom
ALIGN=BOTTOM Alinha o texto adjacente com a parte de baixo da imagem *(default)*

ALIGN=RIGHT Alinha imagem à direita, e tudo o que houver ao redor (texto, outras imagens) a partir do topo da imagem.

ALIGN=LEFT Alinha imagem à esquerda, e tudo o que houver ao redor (texto, outras imagens) a partir do topo da imagem.

Outros alinhamentos procuram posicionar as imagens com maior precisão com relação ao texto circundante: TOPTEXT, ABSMIDDLE e ABSBOTTOM.
Para ter duas imagens, uma em cada margem, numa mesma linha, escreva:

```
<IMG align=left SRC="imagem.gif" alt="imagem"><IMG align=right
SRC="imagem.gif" alt="imagem">...e se pode escrever à vontade entre as
imagens!
```

Isso resulta em:

...e se pode escrever à vontade entre as imagens!

Um detalhe surgido com o alinhamento de imagens foi a necessidade de se liberar o texto desse alinhamento. Ou seja:

Suponhamos um texto mais ou menos curto, que desejamos colocar aqui, com a imagem ilustrativa...
...mas gostaríamos que **este** trecho já estivesse abaixo da imagem! De acordo com o comprimento da primeira frase, não seria possível usar o alinhamento TOP.
Para conseguir isso, seria necessário incluir diversos <BR> consecutivos, inserindo linhas em branco; mesmo assim, o resultado final poderia ser bem pouco elegante. Surgiu, então, o atributo CLEAR para <BR>.
Com esse atributo, podemos, por exemplo...

...ter um texto posicionado no ponto em que a margem direita fica livre, com
<BR **CLEAR**=RIGHT>
ou no ponto em que a margem esquerda fica livre, com
<BR **CLEAR**=LEFT>
Dessa maneira, podemos controlar bem a posição relativa dos textos.

Também se pode posicionar o texto no ponto em que ambas as margens estão livres.
Isso é conseguido com
<BR **CLEAR**=ALL>
E, assim, vimos tudo sobre quebras de linha depois de imagens!

ISMAP

Qualquer imagem pode funcionar como uma âncora de link, como vimos no item sobre bordas. `ISMAP` indica quando uma imagem deve ser tratada como um mapa clicável, isto é, quando cada pixel de uma imagem pode ser considerado uma âncora para algum arquivo específico.
Os mapas serão apresentados com detalhes na seção de assuntos avançados, sob o item Interação.

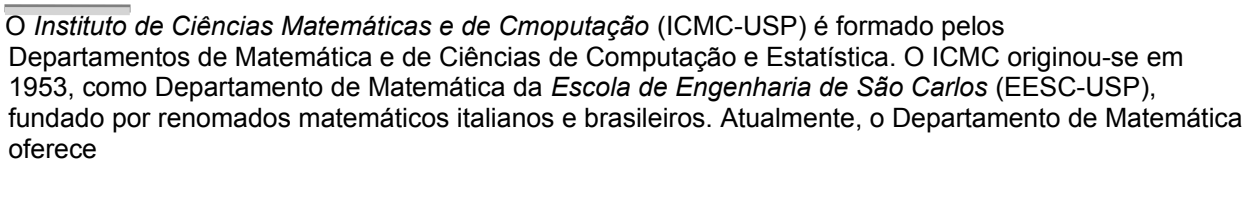
Molduras de imagem

Para melhorar ainda mais a apresentação das imagens junto com os textos, foram desenvolvidos atributos de moldura. Estes atributos definem o espaço - vertical e horizontal - deixado entre as imagens e os textos circundantes:

```
<IMG SRC="imagem" VSPACE=esp_vertical>

<IMG SRC="imagem" HSPACE=esp_horizontal>
```

O efeito desses atributos pode ser percebido nos textos abaixo. No primeiro texto, as imagens não têm atributos de moldura (é fácil notar como o texto fica "grudado" na imagem)


O *Instituto de Ciências Matemáticas e de Cmoputação* (ICMC-USP) é formado pelos Departamentos de Matemática e de Ciências de Computação e Estatística. O ICMC originou-se em 1953, como Departamento de Matemática da *Escola de Engenharia de São Carlos* (EESC-USP), fundado por renomados matemáticos italianos e brasileiros. Atualmente, o Departamento de Matemática oferece

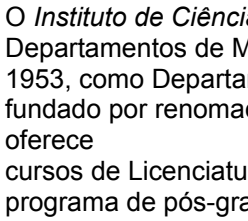
cursos de Licenciatura e Bacharelado em Matemática em nível de graduação, além de um programa de pós-graduação que inclui mestrado e doutorado na área de Matemática. O Departamento de Computação e Estatística é responsável pelo curso de Bacharelado em Ciência de Computação, no qual ingressam 40 alunos por ano. Em nível de pós-graduação oferece, desde 1975, o programa de mestrado em Ciências de Computação e Matemática Computacional e, a partir de agosto de 1995, o programa de doutorado na mesma área.

Neste segundo texto são usadas, respectivamente, as formatações:

```
<IMG SRC="icones/fotoicm.gif" WIDTH="148" HEIGHT="95" ALIGN=left
VSPACE="30">
```

e

```
<IMG SRC="icones/smallpos.gif" WIDTH="160" HEIGHT="71" ALIGN=right
HSPACE="30">
```



O *Instituto de Ciências Matemáticas e de Computação* (ICMC-USP) é formado pelos Departamentos de Matemática e de Ciências de Computação e Estatística. O ICMC originou-se em 1953, como Departamento de Matemática da *Escola de Engenharia de São Carlos* (EESC-USP), fundado por renomados matemáticos italianos e brasileiros. Atualmente, o Departamento de Matemática oferece
cursos de Licenciatura e Bacharelado em Matemática em nível de graduação, além de um programa de pós-graduação que inclui mestrado e doutorado na área de Matemática. O Departamento de Computação e Estatística é responsável pelo curso de Bacharelado em Ciência de Computação, no qual ingressam 40 alunos por ano. Em nível de pós-graduação oferece, desde 1975, o programa de mestrado em Ciências de Computação e Matemática Computacional e, a partir de agosto de 1995, o programa de doutorado na mesma área.

Os dois atributos de moldura podem estar presentes ao mesmo tempo. Vejamos primeiro o texto com a imagem sem moldura:

"A cultura UNIX começou a ser apreciada por usuários brasileiros ainda na década de 70, pelos contatos de pesquisadores brasileiros em cursos de aperfeiçoamento no exterior - notadamente na América do Norte. O contingente era, contudo, pequeno e restrito a acadêmicos. A disseminação da cutura UNIX no mercado comercial só teve início com o advento da década de 80."

(Citação de texto encontrado à página 18 do livro *UNIX - Guia do Usuário* - Autores: Marcus C. Sampaio, Jacques P. Sauvé e J. Antão B. Moura - McGraw-Hill, 1988)

Abaixo, vemos a aplicação dos dois atributos, através da formatação:

```
<IMG SRC="icones/earth.gif" ALIGN="LEFT" WIDTH="63" HEIGHT="68"
HSPACE="20" VSPACE="20">
```

"A cultura UNIX começou a ser apreciada por usuários

brasileiros ainda na década de 70, pelos contatos de pesquisadores brasileiros em cursos de aperfeiçoamento no exterior - notadamente na América do Norte. O contingente era, contudo, pequeno e restrito a acadêmicos. A disseminação da cutura UNIX no mercado comercial só teve início com o advento da década de 80."

(Citação de texto encontrado à página 18 do livro *UNIX - Guia do Usuário* - Autores: Marcus C. Sampaio, Jacques P. Sauvé e J. Antão B. Moura - McGraw-Hill, 1988)

Imagem Alternativa

Catedral de São Carlos (São Carlos-SP)

O efeito de sobrepor imagens, como visto aqui, é conseguido com o atributo `LOWSRC`.

```
<IMG LOWSRC="imagem1" SRC="imagem2">
```

Para conseguir o efeito, você precisará trabalhar com duas imagens:

1. a primeira (`LOWSRC="imagem1"`), que é carregada rapidamente na página. Ela deve estar em um arquivo pequeno (por exemplo, a imagem em preto-e-branco);
2. a segunda imagem a ser carregada (`SRC="imagem2"`), que permanecerá na página. Deve estar em um arquivo grande (por exemplo, a imagem colorida).

As duas imagens devem ter as mesmas dimensões! Isso significa que os atributos `WIDTH` e `HEIGTH` serão necessariamente utilizados.
É bom lembrar que o efeito pode ser realizado também com duas imagens coloridas: o importante é que a primeira imagem a ser carregada esteja em um arquivo bem menor que o da segunda imagem

Tabelas

A formatação de tabelas foi adotada bem antes de sua inclusão na definição de HTML. A manipulação de tabelas, mesmo em editores, é trabalhosa; a maior diferença entre tabelas em HTML e em edit ores como o MS Word, entretanto, é o fato das tabelas em HTML serem definidas apenas *em termos de linhas* e não de colunas. Mas isso será percebido no decorrer destas páginas.
As tabelas foram uma grande conquista para os autores de documentos para a Web. Com elas é possível, por exemplo, termos estas páginas do tutorial organizadas em colunas, sendo uma delas voltada exclusivamente aos links de navega&cced il;ão e observações. Tabelas implementam um conceito importante de *layout*: as “grades”, segundo as quais organizamos textos e ilustrações de maneira harmoniosa.
Como já foi possível perceber, as tabelas contêm textos, listas, parágrafos, imagens, diversas outras formatações - inclusive outras tabelas. Novas versões de HTML e de browsers populares vêm acrescentando diversos atributos às tabelas, e nosso objetivo aqui é saber lidar com a maioria desses recursos disponíveis.

Elementos básicos de tabelas

`<TABLE>...</TABLE>` delimita uma tabela. Um atributo básico é **BORDER**, que indica a apresentação da borda.

```
<TABLE BORDER="borda">

...

</TABLE>
```

Títulos, linhas e elementos

```
<CAPTION>...</CAPTION>
```

define o título da tabela

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

delimita uma linha

`<TH>...</TH>`

define um cabeçalho para colunas ou linhas (dentro de `<TR>`)

`<TD>...</TD>`

delimita um elemento ou célula (dentro de `<TR>`)

Uma tabela simples:

```
<TABLE BORDER=4>
<CAPTION>Primeiro exemplo</CAPTION>
<TR><TH>Coluna 1</TH><TH>Coluna 2</TH></TR>
<TR><TD>linha1, coluna 1</TD><TD> linha 1, coluna 2</TD></TR>
<TR><TD>linha 2, coluna 1</TD><TD>linha 2, coluna 2</TD></TR>
</TABLE>
```

Primeiro exemplo

Coluna 1	Coluna 2
linha1, coluna 1	linha 1, coluna 2
linha 2, coluna 1	linha 2, coluna 2

<HREF=".HTML"Folhas de Estilo

<HREF=".HTML"HTML dinâmico

Títulos compreendendo mais de uma coluna ou linha

É possível englobar colunas e linhas, através dos atributos **COLSPAN** (para colunas) e **ROWSPAN** (para linhas):

```
<TABLE BORDER=1>
<TR><TH COLSPAN=2>Colunas 1 e 2</TH></TR>
<TR><TD>linha1, coluna 1</TD><TD> linha 1, coluna 2</TD></TR>
<TR><TD>linha 2, coluna 1</TD><TD>linha 2, coluna 2</TD></TR>
<TR><TH ROWSPAN=3>3 linhas</TH><TD>uma linha</TD></TR>
<TR><TD>duas linhas</TD></TR>
<TR><TD>tres linhas</TD></TR>
</TABLE>
```

Colunas 1 e 2	
linha1, coluna 1	linha 1, coluna 2
linha 2, coluna 1	linha 2, coluna 2
3 linhas	uma linha
	duas linhas
	tres linhas

Neste exemplo, vemos que o cabeçalho **Colunas 1 e 2** compreende duas colunas (`COLSPAN=2`); o cabeçalho **3 linhas** compreende, por sua vez, 3 linhas (`ROWSPAN=3`).

Tabelas sem borda

As páginas deste tutorial foram construídas com tabelas sem borda. Para tanto, foi empregada a seguinte delaração:

```
<TABLE BORDER="0">

...

</TABLE>
```

Alinhamentos em tabelas

Este exemplo servirá para estudarmos alinhamentos, controle de larguras e espaçamento em tabelas:

Prédio principal do ICMSC-USP	O *Instituto de Ciências Matemáticas de São Carlos* (ICMSC-USP) é formado pelos Departamentos de Matemática e de Ciências de Computação e Estatística. O ICMSC originou-se em 1953, como Departamento de Matemática da *Escola de Engenharia de São Carlos* (EESC-USP), fundado por renomados matemáticos italianos e brasileiros.
Departamento de Matemática (SMA)	Atualmente, o Departamento de Matemática oferece cursos de Licenciatura e Bacharelado em Matemática em nível de graduação, além de um programa de pós-graduação que inclui mestrado e doutorado na área de Matemática.
Departamento de Computação e Estatística (SCE)	O Departamento de Computação e Estatística é responsável pelo curso de Bacharelado em Ciência de Computação, no qual ingressam 40 alunos por ano. Em nível de pós-graduação oferece, desde 1975, o programa de mestrado em Ciências de Computação e Matemática Computacional e, a partir de agosto de 1995, o programa de doutorado na mesma área.
Para maiores informações: Cursos de Graduação: "grad@icmsc.sc.usp.br Cursos de Pós-Graduação: "posgrad@icmsc.sc.usp.br	

O conteúdo é informativo, porém a apresentação não é agradável devido à disposição do texto na tabela.
Primeiro, vamos mexer com os alinhamentos.

Alinhamentos simples

Os alinhamentos padrão em tabelas, como podemos ver no exemplo acima, são:

no sentido horizontal: alinhamento à esquerda

no sentido vertical: alinhamento no centro da célula

As linhas e células podem ter alinhamentos definidos através dos atributos:

ALIGN = alin_horizontal

VALIGN = alin_vertical

<HREF=".HTML"Folhas de Estilo

<HREF=".HTML"HTML dinâmico

Vejamos como esses alinhamentos funcionam nas células:

`<TD ALIGN=alin_horizontal>`Texto da célula`</TD>`

`<TD VALIGN=alin_vertical>`Texto da célula`</TD>`

Padrão	**ALIGN**=LEFT	**ALIGN**=CENTER	**ALIGN**=RIGHT
Padrão	**VALIGN**=TOP	**VALIGN**=MIDDLE	**VALIGN**=BOTTOM

Obs.: a tabela acima foi feita especialmente para testar os alinhamentos. Uma tabela comum ajusta o tamanho de suas células ao conteúdo:

Padrão	align=left	align=center	align=right
Padrão	valign=top	valign=middle	valign=bottom

Alinhamentos combinados

Uma mesma célula pode ter atributos `ALIGN` e `VALIGN`.

`<TD ALIGN=alin_horizontal VALIGN=alin_vertical>`Texto da célula`</TD>`

Por exemplo:

Padrão	**ALIGN**=LEFT, **VALIGN**=BOTTOM	**ALIGN**=CENTER, **VALIGN**=TOP	**ALIGN**=RIGHT, **VALIGN**=MIDDLE

Alinhamentos de linhas

O alinhamento pode ser aplicado a linhas inteiras, com:

`<TR ALIGN=alin_horizontal VALIGN=alin_vertical>`Texto da célula`</TR>`

Porém, o alinhamento declarado em uma célula prevalece sobre o alinhamento da linha, como se vê no exemplo:

center	center	center	TD **ALIGN**=RIGHT
	TD **VALIGN**=	bottom	bottom

bottom	TOP		

Isso pode ser interessante para algumas aplicações.

Já conseguimos mexer um pouco na tabela inicial, inserindo alinhamentos combinados; serão necessários mais alguns passos para que a tabela fique realmente "apresentável" - o exemplo continua nas seções sobre larguras e espaçamentos.

Prédio principal do ICMSC-USP	O *Instituto de Ciências Matemáticas de São Carlos* (ICMSC-USP) é formado pelos Departamentos de Matemática e de Ciências de Computação e Estatística. O ICMSC originou-se em 1953, como Departamento de Matemática da *Escola de Engenharia de São Carlos* (EESC-USP), fundado por renomados matemáticos italianos e brasileiros.
Departamento de Matemática (SMA)	Atualmente, o Departamento de Matemática oferece cursos de Licenciatura e Bacharelado em Matemática em nível de graduação, além de um programa de pós-graduação que inclui mestrado e doutorado na área de Matemática.
Departamento de Computação e Estatística (SCE)	O Departamento de Computação e Estatística é responsável pelo curso de Bacharelado em Ciência de Computação, no qual ingressam 40 alunos por ano. Em nível de pós-graduação oferece, desde 1975, o programa de mestrado em Ciências de Computação e Matemática Computacional e, a partir de agosto de 1995, o programa de doutorado na mesma área.
Para maiores informações:	

Atributos de largura

Na seção anterior, foi comentado que uma tabela comum ajusta o tamanho de suas células ao conteúdo. Por exemplo:

janeiro	fevereiro	março
abril	maio	junho

Para apresentar uma tabela ocupando determinado espaço disponível na linha, usamos o atributo `WIDTH`. Esse atributo pode ser aplicado também a linhas e células.

Essa largura pode ser definida em porcentagem (do espaço disponível):

WIDTH=x%

ou em pixels:

WIDTH=x

Ex.1: Tabela ocupando 50% do espaço disponível

```
<TABLE BORDER=1 width=50%>
```

janeiro	fevereiro	março
abril	maio	junho

Ex.2: Tabela ocupando 50% do espaço disponível, com uma coluna de 60% do espaço disponível na tabela

```
<TABLE BORDER=1 width=50%>
<TR>
   <TD>janeiro</TD><TD width=60%>fevereiro</TD><TD>março</TD>
</TR>
<TR>
   <TD>abril</TD><TD width=60%>maio</TD><TD>junho</TD>
</TR>
</TABLE>
```

janeiro	fevereiro	março
abril	maio	junho

Ex3.: O controle da largura da tabela está limitado à dimensão de seu conteúdo:

```
<TABLE BORDER=1 width=50%>
<TR>
   <TD>janeiro</TD><TD width=1%>fevereiro</TD><TD>março</TD>
</TR>
<TR>
   <TD>abril</TD><TD width=1%>maio</TD><TD>junho</TD>
</TR>
</TABLE>
```

janeiro	fevereiro	março
abril	maio	junho

<HREF=".HTML"Folhas de Estilo

<HREF=".HTML"HTML dinâmico

Oh-oh...

Alguns editores WYSIWYG não trabalham com atributos de largura. Nestes casos, é preciso editar o arquivo fonte.

De volta ao exemplo inicial, já podemos melhorar um pouco mais nossa tabela. Mantendo os alinhamentos definidos na seção anterior, aplicaremos atributos de largura:

Prédio principal do ICMSC-USP	O *Instituto de Ciências Matemáticas de São Carlos* (ICMSC-USP) é formado pelos Departamentos de Matemática e de Ciências de Computação e Estatística. O ICMSC originou-se em 1953, como Departamento de Matemática da *Escola de Engenharia de São Carlos* (EESC-USP), fundado por renomados matemáticos italianos e brasileiros.
Departamento de Matemática (SMA)	Atualmente, o Departamento de Matemática oferece cursos de Licenciatura e Bacharelado em Matemática em nível de graduação, além de um programa de pós-graduação que inclui mestrado e doutorado na área de Matemática.
Departamento de Computação e Estatística (SCE)	O Departamento de Computação e Estatística é responsável pelo curso de Bacharelado em Ciência de Computação, no qual ingressam 40 alunos por ano. Em nível de pós-graduação oferece, desde 1975, o programa de mestrado em Ciências de Computação e Matemática Computacional e, a partir de agosto de 1995, o programa de doutorado na mesma área.
Para maiores informações: Cursos de Graduação: "grad@icmsc.sc.usp.br Cursos de Pós-Graduação: "posgrad@icmsc.sc.usp.br	

Ainda faltam detalhes. Um deles é evitar que o texto fique *grudado* nas bordas da tabela; veremos na próxima seção, sobre espaçamentos

Atributos de espaçamento

Dois atributos permitem o controle de espaçamento em tabelas:

CELLPADDING - espaço entre o texto e as bordas da célula

CELLSPACING - espaço entre células

Tomemos a mesma tabela simples da seção anterior:

janeiro	fevereiro	março
abril	maio	junho

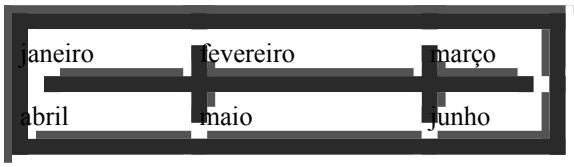
Ex.1: Espaço entre o texto e as bordas

`<TABLE BORDER=1 CELLPADDING=20>`

janeiro	fevereiro	março
abril	maio	junho

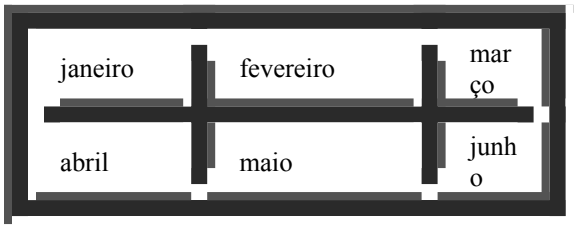
Ex.2: Espaço entre células

<TABLE BORDER=1 **CELLSPACING**=20>



Ex3.: Espaço entre texto e bordas, e espaço entre células

<TABLE BORDER=1 **CELLPADDING**=20 **CELLSPACING**=20>



<HREF=".HTML"Folhas de Estilo

<HREF=".HTML"HTML dinâmico

Assim, damos mais uma mexida na tabela inicial:

Prédio principal do ICMSC-USP	O *Instituto de Ciências Matemáticas de São Carlos* (ICMSC-USP) é formado pelos Departamentos de Matemática e de Ciências de Computação e Estatística. O ICMSC originou-se em 1953, como Departamento de Matemática da *Escola de Engenharia de São Carlos* (EESC-USP), fundado por renomados matemáticos italianos e brasileiros.
Departamento de Matemática (SMA)	Atualmente, o Departamento de Matemática oferece cursos de Licenciatura e Bacharelado em Matemática em nível de graduação, além de um programa de pós-graduação que inclui mestrado e doutorado na área de Matemática.
Departamento de Computação e Estatística (SCE)	O Departamento de Computação e Estatística é responsável pelo curso de Bacharelado em Ciência de Computação, no qual ingressam 40 alunos por ano. Em nível de pós-graduação oferece, desde 1975, o programa de mestrado em Ciências de Computação e Matemática Computacional e, a partir de agosto de 1995, o programa de doutorado na mesma área.
Para maiores informações: Cursos de Graduação: "grad@icmsc.sc.usp.br Cursos de Pós-Graduação: "posgrad@icmsc.sc.usp.br	

Como toque final, retiramos a borda:

Prédio principal do ICMSC-USP	O *Instituto de Ciências Matemáticas de São Carlos* (ICMSC-USP) é formado pelos Departamentos de Matemática e de Ciências de Computação e Estatística. O ICMSC originou-se em 1953, como Departamento de Matemática da *Escola de Engenharia de São Carlos* (EESC-USP), fundado por renomados matemáticos italianos e brasileiros.
Departamento de Matemática (SMA)	Atualmente, o Departamento de Matemática oferece cursos de Licenciatura e Bacharelado em Matemática em nível de graduação, além de um programa de pós-graduação que inclui mestrado e doutorado na área de Matemática.
Departamento de Computação e Estatística (SCE)	O Departamento de Computação e Estatística é responsável pelo curso de Bacharelado em Ciência de Computação, no qual ingressam 40 alunos por ano. Em nível de pós-graduação oferece, desde 1975, o programa de mestrado em Ciências de Computação e Matemática Computacional e, a partir de agosto de 1995, o programa de doutorado na mesma área.

Para maiores informações:

Cursos de Graduação: "grad@icmsc.sc.usp.br

Cursos de Pós-Graduação: "posgrad@icmsc.sc.usp.br

Agora já vimos grande parte dos recursos disponíveis para manipular tabelas, que permitem produzir bons efeitos de apresentação.

Extensões de Tabelas

Diversas extensões de tabelas possibilitam a apresentação de efeitos muito bons nas páginas.

Cor de fundo

```
<TABLE BORDER=5 CELLSPACING=5 CELLPADDING=10 BGCOLOR="#E1FFD9">
```

janeiro	fevereiro	março
abril	maio	junho

Cor de borda

```
<TABLE BORDER=5 CELLSPACING=5 CELLPADDING=10 BGCOLOR="#E1FFD9"
BORDERCOLOR="#00FF00">
```

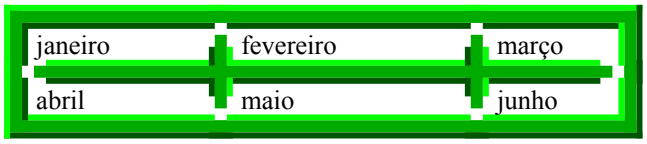
janeiro	fevereiro	março
abril	maio	junho

<HREF=".HTML"Folhas de Estilo

<HREF=".HTML"HTML dinâmico

Imagem de fundo

```
<TABLE BORDER=5 BACKGROUND="imagem">
```



Frames

Os frames são divisões da tela do browser em diversas telas (ou "quadros"). Com isso, torna-se possível apresentar mais de uma página por vez: por exemplo, um índice principal em uma parte pequena da tela, e os textos relacionados ao índice em outra parte.
É muito fácil colocar frames em páginas; porém, nem todos os usuários gostam deles. (*Eu gosto! Mas refiz o tutorial usando o recurso tabela, que é a alternativ a natural de frames*)
Uma página com frames tem um texto fonte semelhante a:

```
<HTML>

<HEAD><TITLE>Assunto X</TITLE></HEAD>

<FRAMESET COLS="20%, 80%">

   <FRAME SRC="indice1.html">

   <FRAME SRC="apresenta.html" NAME=principal>

   <NOFRAME>

   <BODY>

   <H2>Bem-vindo à página do assunto X!</h2>

   <P>

   Blá blá blá blá blá

   blá blá blá blá blá

   </BODY>

   </NOFRAME>

</FRAMESET>

</HTML>
```

<HREF=".HTML"Folhas de Estilo

<HREF=".HTML"HTML dinâmico

A parte **FRAMESET** define a divisão da página em "quadros". Neste exemplo, a página será dividida em duas colunas, sendo a primeira com 20% do tamanho da tela, e a segunda coluna com os restantes 80% da tela.

Dentro da formatação de **FRAMESET**, temos os **FRAME SRC**, que são referências às páginas que serão mostradas nos frames definidos
Assim, aqui vemos que a página `indice1.html` será mostrada na primeira coluna (que ocupará 20% da tela), e `apresenta.html` será mostrada na segunda (ocupando 80% da tela).
[Veja o exemplo]
A formatação de frames inclui também uma parte **NOFRAME**, que é mostrada normalmente pelos browsers que não suportam a apresentação de frames.

Links com Frames

Sempre que se aciona um link dentro de uma página, o *default* (isto é, o comportamento padrão) é que a página referente a esse link seja carregada na mesma janela da página anterior. No exemplo visto com fra mes, seguir um link dentro da janela à direita faz com que a página apontada seja mostrada ocupando a janela da direita (coluna de 80%):

```
<HTML>
<HEAD><TITLE>Assunto X</TITLE></HEAD>
<FRAMESET COLS="20%, 80%">
            <FRAME SRC="indice1.html">
            <FRAME SRC="apresenta.html" NAME=principal>

  <NOFRAME>
  <BODY>
  <H2>Bem-vindo à página do assunto X!</h2>
  <P>
  Blá blá blá blá blá
    blá blá blá blá blá
  </BODY>
  </NOFRAME>

</FRAMESET>
</HTML>
```

<HREF=".HTML"Folhas de Estilo

<HREF=".HTML"HTML dinâmico

Veja no código fonte acima que o frame associado a `apresenta.html` tem um atributo **NAME**. Isso faz com que se possa "direcionar" o frame em que queremos mostrar determinada página ao acionarmos um lin k.
No exemplo visto, o arquivo `indice1.html` tem um link da seguinte forma:

```
<a href="apresenta2.html" target=principal>Exemplo no.2</a>.
```

Quando se acionar esse link, a página `apresenta2.html` será mostrada no frame que denominamos `principal`, ou seja: em vez de carregar `apresenta2.html` na mesma coluna em que está `indice1.html`, ela será mostrada na coluna em que está `apresenta.html`.
Se você não entendeu como funciona esse mecanismo de direcionamento, veja o exemplo e teste todos os links para verificar o que foi dito neste parágrafo, pois a descrição fica muito confusa!! `%-/`

Composições com Frames

Como já foi possível observar, `FRAMESET` tem atributos que definem a divisão da janela do browser em colunas; essa divião também pode ser feita em linhas, e utilizando uma combinação de "framesets" para variadas apresentações:

```
<FRAMESET COLS="x, y">
   <FRAME SRC="col1.html">
   <FRAME SRC="col2.html">
</FRAMESET>
<FRAMESET ROWS="x, y">
   <FRAME SRC="lin1.html">
   <FRAME SRC="lin2.html">
</FRAMESET>
<FRAMESET COLS="x, y">
   <FRAME SRC="col1.html">
<FRAMESET ROWS="x, y">
   <FRAME SRC="lin1.html">
   <FRAME SRC="lin2.html">
</FRAMESET>
</FRAMESET>
<FRAMESET ROWS="x, y">
   <FRAME SRC="lin1.html">
<FRAMESET COLS="x, y">
   <FRAME SRC="col1.html">
   <FRAME SRC="col2.html">
</FRAMESET>
</FRAMESET>
```

Lembre-se de que os frames fixos não precisam ter nomes, mas os frames que receberão textos, sim!

O exemplo simples visto até agora segue o modelo da primeira composição (em colunas). As composições com mais de um `frameset` precisam ser bem planejadas para funcionarem bem.

Atributos de Frames

Até este ponto, vimos os atributos **COLS** e **ROWS** (para `FRAMESET`), **SRC** e **NAME** (para `FRAME`).
Outros atributos permitem um maior controle sobre a apresentação:

Eliminação das bordas dos frames:

`FRAMESET` **FRAMEBORDER**`="NO"`

[Exemplo]

Eliminação do espaço entre os frames

`FRAMESET FRAMEBORDER="NO"` **BORDER**`="0"`

[Exemplo]

Frame sem barra de rolagem

`FRAME` **SCROLLING**`="NO"`

[Exemplo]
É bom lembrar que a barra de rolagem de um frame fica sempre à direita; não é possível, atualmente, mudar essa característica

Limpando a Tela

Item em construção

Há basicamente dois efeitos possíveis para limpar a apresentação de frames, e isso é feito com "targets" especiais (lembre-se como usar o atributo `TARGET` no item Links com frames):

`TARGET="_top"` limpa os frames, apresentando a próxima página em tela inteira

[Exemplo de aplicação]

`TARGET="_blank"` abre uma nova janela do browser, sem frames, para apresentar a próxima página

Interação

A interação é realizada de duas formas diferentes:

1. através de programas executados/interpretados pelo browser;
2. através de programas executados pelo servidor HTTP.

Conforme a aplicação, apenas um destes tipos de interação pode ou deve ser utilizado.
Pelo momento, maiores detalhes constam apenas do manual em edição ("World-Wide Web: Manual do usuário"). Veremos aqui algumas aplicações básicas

Mapas

"Server-Side Imagemap"

Um primeiro método para criação de mapas é usando a comunicação com o servidor HTTP.
Os servidores WWW têm um programa que lida com a relação entre coordenadas de imagens e URLs. No servidor NCSA, esse programa é o **imagemap**, no CERN é o **htimage**.
Escolhida a imagem que servirá de base de partida de diversos links para outros documentos, é preciso gerar um arquivo relacionando coordenadas da imagem a determinados links. Essas coordenadas são encontradas, através de progr amas de manipulação de imagens - *XView*, *LView*, entre outros.
Um arquivo **.map** (do servidor NCSA), tem um conteúdo semelhante a:

```
default    /nada.html
rect       http://www.usp.br/                                  15,8 135,39
circle     http://www.intermidia.icmsc.sc.usp.br/              306,204 7
```

Um arquivo **.conf** (do servidor CERN), tem um conteúdo semelhante a:

```
default                        /nada.html
rect      (15,8)  (135,39)     http://www.usp.br/
```

```
circ      306,204 7          http://www.intermidia.icmsc.sc.usp.br/
```
Ambos arquivos significam a mesma coisa:

1. a região da figura, compreendida pelo retângulo (`rect`) de coordenadas `(15,8)` e `(135,39)`, funciona como um link para o URL `http://www.usp.br/`;
2. a região da figura, compreendida pelo círculo (`circle` ou `circ`) de centro `(306,204)` e raio `7` é um link para o URL `http://www.intermidia.icmsc.sc.usp.br/`;
3. se o mouse não for acionado em nenhuma dessas duas regiões previstas, o link será dirigido para o arquivo default - neste exemplo, o `nada.html`

O formato desses arquivos para figuras clicáveis pode variar, mas basicamente contém esses mesmos elementos:

- **default** - indica um endereço padrão para quando o mouse for acionado em uma área da figura além das previstas pelo autor;
- **circle** ou **circ** - declara um círculo na figura; os pontos indicam o centro e um ponto de fronteira do círculo;
- **poly** - um polígono; cada coordenada declarada é um vértice;
- **rect** - um retângulo; as coordenadas declaradas são, respectivamente, do vértice esquerdo superior e vértice direito inferior.

Tão logo seu arquivo **fazclic.map** esteja pronto, seu mapa sensível deve ser declarado da seguinte maneira:

```
<A HREF="http://www.host.br/clic.map"><IMG SRC="imagem.gif" ISMAP></A>
```

onde

- **http://www.host.br/clic.map** - é o endereço para o arquivo que contém as diretivas que associam regiões da figura a referências WWW
- **imagem.gif** - é a figura que irá "funcionar" como imagem clicável

Como exemplo, temos um arquivo de nome mapa.map, cujo conteúdo é:

```
#Pagina da Graduacao

rect docs/grad.html 3,2 130,42

#Pagina do Mestrado

rect docs/mestr.html 133,3 230,44

#Pagina do Doutorado

rect docs/dout.html 234,3 364,45
```

relacionado a uma imagem da seguinte forma:

```
<A HREF="mapa.map"><IMG SRC="icones/cursos.gif" ISMAP BORDER=0></A>
```

mapa.mapmapa.map

Obs.: Como estamos sem o programa que faz funcionar esse imagemap, que executa a correspondência entre as coordenadas e os arquivos, cada vez que for acionado o mouse sobre a imagem, o resultado será o código fonte do arquivo `mapa.map` (como visto acima), sendo que o URL desse resultado será alguma coisa parecida com:
`http://www.icmsc.sc.usp.br/manuals/HTML/mapa.map?183,24`
sendo que os números após o ponto de interrogação serão as coordenadas do pixel selecionado.

<HREF=".HTML"Folhas de Estilo

<HREF=".HTML"HTML dinâmico

"Client-Side Imagemap"

Com o *client-side imagemap*, os mapas sensíveis são rastreados pelo browser no momento em que o usuário escolhe um ponto na imagem; dessa forma, o servidor já recebe uma requisição de um documento, pois as c oordenadas do ponto escolhido e o arquivo correspondente já foram identificados pelo browser.
Na verdade, a relação coordenadas-documentos continua existindo, mas agora em vez de estar em um *script* separado, está dentro do próprio documento:

```
<MAP NAME="nomemapa">

   <AREA SHAPE="forma" HREF="arq.html" COORDS="x1,y1,x2,y2">

</MAP>

<IMG SRC="imagem.gif" USEMAP="#nomemapa">
```

Neste exemplo, transformamos a barra abaixo em mapa sensível:

```
<map name="mapname">

<area shape="rect" href="docs/grad.html" COORDS="0,0,130,45">

<area shape="rect" href="docs/mestr.html" COORDS="133,0,230,45">

<area chape="rect" href="docs/dout.html" COORDS="234,0,364,45">

</map>

<img src="icones/cursos.gif" USEMAP="#mapname" border=0>
```

Formulários

Um formulário é um modelo para a entrada de um conjunto de dados. O primeiro passo para fazer formulários é aprender as etiquetas que desenham as janelinhas de entrada de dados, para depois trabalharmos com os *scripts*, que são os programas que tratam esses dados, oferecendo os serviços desejados (acesso a banco de dados, envio de e-mail, etc.).
O elemento <FORM> delimita um formulário e contém uma seqüência de elementos de entrada e de formatação do documento.
<FORM **ACTION**="URL_de_script" **METHOD**=método>...</FORM>
Os atributos de FORM que nos interessam agora são:

ACTION

Especifica o URL do *script* ao qual serão enviados os dados do formulário.

METHOD

Seleciona um método para acessar o URL de ação. Os métodos usados atualmente são GET e POST. Ambos os métodos transferem dados do browser para o servidor, com a seguinte diferença básica:

- POST

  - os dados entrados fazem parte do corpo da mensagem enviada para o servidor;

  - transfere grande quantidade de dados.

- GET

  - os dados entrados fazem parte do URL associado à consulta enviada para o servidor;

  - suporta até 128 caracteres.

Veremos maiores detalhes sobre métodos no item CGI.
FORM também pode apresentar o atributo:

ENCTYPE

Indica o tipo de codificação dos dados enviados através do formulário. O tipo *default* é application/x-www-form-urlencoded. Outro tipo aceito por alguns browsers é text/plain, m as sua utilização ainda não está padronizada.

Os formulários podem conter qualquer formatação - parágrafos, listas, tabelas, imagens - exceto outros formulários. Em especial, colocamos dentro da marcação de <FORM> as formatações para campos de entrada de dados, que são três: <INPUT>, <SELECT> e <TEXTAREA>.
Todos os campos de entrada de dados têm um atributo NAME, ao qual associamos um nome, utilizado posteriormente pelo *script*. São os *scripts* que organizam esses dados de entrada em um conjunto de informações significativas para determinado propósito. Primeiro vamos ver os tipos de campos para montar um formulário, e depois passaremos aos *scripts*.

INPUT

O campo **<INPUT>** tem um atributo `TYPE`, ao qual atribuímos seis valores diferentes para gerar seis tipos diferentes de entrada de dados.

Campo de dados texto

Quando `INPUT` não apresenta atributos, é assumido `TYPE=TEXT` (*default*); a formatação:

```
<FORM>

Nome: <INPUT TYPE=TEXT NAME="Nome">

</FORM>
```

ou apenas:

```
<FORM>

Nome: <INPUT NAME="Nome">

</FORM>
```

produz o resultado:

Nome:

Campo de dados senha

Entrada de texto na qual os caracteres são escondidos por asteriscos, como se pode ver no exemplo.

```
<FORM>

Login: <INPUT TYPE=TEXT NAME=login><br>

Password: <INPUT TYPE=PASSWORD NAME="senha">

</FORM>
```

Login: masc

Password:

Alguns atributos

VALUE pode ser usado para dar um valor inicial a um campo de tipo texto ou senha. Desse modo, se o usuário não preencher este campo, será adotado este valor padrão. Se o usuário quiser entra r dados, ele somente precisará apagar o que já estiver escrito.

```
Nome: <INPUT TYPE=TEXT NAME="nome" VALUE="Seu nome">
```

Nome: Seu nome

SIZE especifica o tamanho do espaço no display para o campo do formulário. Só é válido para campos `TEXT` e `PASSWORD`. O valor *default* (padrão) é 20.

```
<FORM>

Endereço: <INPUT TYPE=TEXT SIZE=35>
```

```
</FORM>
```

Endereço:

MAXLENGHT é o número de caracteres aceitos em um campo de dados; este atributo só é válido para campos de entrada `TEXT` e `PASSWORD`.

```
<FORM>

Dia do mês: <INPUT TYPE=TEXT NAME="ex" MAXLENGHT=2>

</FORM>
```

produz o resultado:

Dia do mês:

Apenas 2 caracteres serão lidos pelo formulário, não importa o quanto se escreva neste campo.

<HREF=".HTML"Folhas de Estilo

<HREF=".HTML"HTML dinâmico

Múltipla escolha

CHECKBOX insere um botão para escolha de opções. A cada alternativa corresponde um valor a ser manipulado pelo script que processa os dados. Pode ser escolhida mais de uma alternativa.

```
<INPUT TYPE=CHECKBOX NAME="esporte" VALUE="basquete">Basquete <br >

<INPUT TYPE=CHECKBOX NAME=esporte VALUE=bocha>Bocha
```

Uma diretiva `CHECKED` marca uma escolha inicial, isto é, se o usuário não escolher nenhuma opção, o formulário irá considerar a alternativa "pré-escolhida":

```
<INPUT TYPE=CHECKBOX NAME="esporte" VALUE="volei" CHECKED& gt;Vôlei
<br>

<INPUT TYPE=CHECKBOX NAME="esporte" VALUE="futebol">Futebol
```

Esportes que você pratica:

- [ ] futebol — Futebol
- [ ] volei — Vôlei
- [ ] natacao — Natação
- [ ] basquete — Basquete
- [ ] tenis — Tênis
- [ ] bocha — Bocha

Escolha única

RADIO insere um botão de escolha de valores para uma opção, isto é, somente uma alternativa pode ser escolhida

```
<INPUT TYPE=RADIO NAME="time" VALUE="palm">Palmeiras <br>
```

```
<INPUT TYPE=RADIO NAME="time" VALUE="inte">Internacional
```

Uma diretiva `CHECKED` marca uma escolha inicial - se o usuário não escolher nenhuma opção, o formulário irá considerar a alternativa "pré-escolhida":

```
<INPUT TYPE=RADIO NAME="time" VALUE="aea" CHECKED>AEA <br>

<INPUT TYPE=RADIO NAME="time" VALUE="naut">Náutico
```

Seu time do coração:
palm Palmeiras; naut Náutico; flam Flamengo; grem Grêmio;
sant Santos; atle Atlético; cori Corinthians; flum Fluminense;
inte Internacional; cruz Cruzeiro; bota Botafogo; sant Santa Cruz; saop São Paulo; aea AEA.

Botões de ação

`SUBMIT` apresenta o botão que causa o envio dos dados de entrada para o servidor;

```
<FORM>

<INPUT TYPE=SUBMIT>

</FORM>
```

Enviar consult

É possível modificar a mensagem desse botão através do atributo `VALUE`

```
<FORM>

<INPUT TYPE=SUBMIT VALUE="Envia mensagem">

</FORM>
```

Envia mensag

`RESET` restaura os valores iniciais das entradas de dados.

```
<FORM>

<INPUT TYPE=RESET>

</FORM>
```

Redefinir

É possível modificar a mensagem desse botão através do atributo `VALUE`

```
<FORM>

<INPUT TYPE=RESET VALUE="Apaga tudo!">

</FORM>
```

Apaga tudo!

SELECT e TEXTAREA

<SELECT> apresenta uma lista de valores, através de campos OPTION.

```
<SELECT NAME="sabor">
<OPTION>Abacaxi
<OPTION>Creme
<OPTION>Morango
<OPTION>Chocolate
</SELECT>
```

Também é possível estabelecer uma escolha-padrão, através do atributo SELECTED

```
<SELECT NAME="sabor">
<OPTION>Abacaxi
<OPTION SELECTED>Creme
<OPTION>Morango
<OPTION>Chocolate
</SELECT>
```

Com o atributo SIZE, escolhe-se quantos itens da lista serão mostrados (no exemplo, **SIZE**=4):

<HREF=".HTML"Folhas de Estilo

<HREF=".HTML"HTML dinâmico

<TEXTAREA> abre uma área para entrada de texto, de acordo com atributos para número de colunas, linhas, e - se for o caso - um valor inicial.

```
<TEXTAREA COLS=40 ROWS=5 NAME="comentario"> Deixe seu comentário
< /TEXTAREA>
```

CGI Scripts (básico)

CGI, ou *Common Gateway Interface*, é uma interface definida de maneira a possibilitar a execução de programas ("gateways") sob um servidor de informações. Até o momento, CGI suporta apenas servidores HTTP.

Neste contexto, os "gateways" são programas ou scripts (também chamados "cgi-bin") que recebem requisições de informação, retornando um documento com os resultados correspondentes. Esse documento pode existir previamente, ou pode ser gerado pelo script especialmente para atender à requisição (usa-se muito a expressão *"on the fly"*).
Exemplos de aplicação de CGI incluem:

- processamento de dados submetidos através de formulários - consulta a banco de dados, cadastramento em listas, livros de visita, pesquisas de opinião;
- criação de documentos personalizados *on the fly*;
- gerenciamento de contadores de acesso;
- processamento de mapas.

Tais scripts podem ser escritos em qualquer linguagem que possa ler variáveis, processar dados e retornar respostas - ou seja, qualquer linguagem de programação. Os programadores de scripts costumam utilizar determinadas linguagens, d e acordo com a plataforma do servidor:

C, Perl, ou shell de UNIX, em redes de ambiente UNIX;

C, Perl, ou VB Script, em ambiente Windows.

Visão Geral

Os scripts têm uma forma geral comum:

1. leitura de dados (e/ou campos de informação de um pacote HTTP);
2. processamento dos dados (gravar dados em arquivo, realizar cálculos, recuperar dados etc.);
3. montagem de uma página Web ou uma imagem com os resultados produzidos.

Submissão de Formulários

Suponhamos um formulário cuja marcação principal seja:

```
<FORM ACTION="/cgi-bin/masc/manual" METHOD=método>

...

</FORM>
```

onde `ACTION` indica o `URL` do script que receberá os dados (ainda não vamos nos preocupar com o campo `METHOD`)
Vejamos o que ocorre quando acionamos o botão de `SUBMIT` desse formulário: o browser prepara a requisição, organizando os dados do formulário em *pares* **nome-valor**, onde:

nome foi definido nas etiquetas do formulário (pelo atributo `NAME`);

valor é a entrada oferecida pelo usuário.

No script, esses valores serão armazenados em uma estrutura deste tipo:

```
typedef struct {

   char *name

   char *val;

} entry;
```

O único detalhe diferente na manipulação da entrada dos dados no script é a necessidade de decodificá-los.
Por exemplo, um campo do tipo...

```
<INPUT TYPE=TEXT NAME="nome">
```

cuja entrada tenha sido...

será organizado pelo browser da seguinte forma:

```
nome=Prof.+Achille+Bassi
```

Se o formulário tiver mais campos, por exemplo:

```
<INPUT TYPE=TEXT NAME="email">
```

com uma entrada

bassi@icmsc.s

o browser estará gerando uma linha semelhante a:

```
nome=Prof.+Achille+Bassi& email=bassi@icmsc.sc.usp.br
```

Note que os campos de informação são separados por `&`. Também todos os acentos e símbolos especiais são codificados em hexadecimal para o envio dos dados. Esta codificação dos dados de um formulário é padrão.

<HREF=".HTML"Folhas de Estilo

<HREF=".HTML"HTML dinâmico

Para testar como é feita a codificação dos dados de um formulário com vários campos, entre seu e-mail no campo abaixo; ele será usado para enviar uma mensagem para você mesmo (se o e-mail usado no teste for inválido, não será possível realizar o teste corretamente):

Testa exemplo | Apaga

O browser envia a requisição para o `URL` indicado em `ACTION`, que está associado a um script. Uma das funções do script será separar os campos de informação (retirar os `&`) e "limpar" os dados dos sinais de + e símbolos hexadecimais. No exemplo (se você testou), foi possível ver uma entrada "suja", pois em vez de script foi usado o `mailto` para `ACTION`.

Uma forma de ter uma mensagem "limpa" com a entrada de formulários é declarar a codificação `text/plain`. O detalhe é que essa codificação não é ainda padronizada. Você pode testar se seu browser aceita esse formato ao submeter formulários, através de um segundo exemplo, bastando entrar no campo abaixo o seu e-mail:

Testa exemplo | Apaga

Ao usar scripts para formulários, aplicamos funções que fazem o tratamento da linha de entrada; essas funções já estão prontas, disponíveis no arquivo `util.c`, que acompanha o pacote do servidor HTTP.

Essas funções são:

```
char *makeword (char *line, char stop);

char *fmakeword (FILE *f, char stop, int *len);

char x2c (char *what);

void unescape_url (char *url);

void plustospace (char *str);
```

Variáveis de ambiente

Variáveis de ambiente são variáveis estabelecidas pelo servidor HTTP no momento de atender requisições de documentos. Os scripts têm acesso a essas variáveis através da interface CGI.

Variáveis comumente acessadas são:

`SERVER_SOFTWARE`

nome e versão do software do servidor

`SERVER_NAME`

nome do servidor

`SERVER_PROTOCOL`

versão do protocolo HTTP usado pelo servidor

`SERVER_PORT`

porta TCP pela qual o servidor atende as requisições

`QUERY_STRING`

cadeia de caracteres que contém uma consulta ou entrada de formulário (presente de acordo com o método e o script utilizado para tratar os dados do formulário)

`REMOTE_HOST`

máquina que solicita a execução do script (máquina do usuário)

`CONTENT_TYPE`

tipo MIME do conteúdo da mensagem enviada

`CONTENT_LENGHT`

comprimento da mensagem enviada

`HTTP_USER_AGENT`

nome e versão do browser utilizado pelo usuário

Métodos de HTTP

O protocolo HTTP utiliza vários métodos de manipulação e organização dos dados. Atualmente, dois métodos são mais utilizados para submeter dados de formulários: `POST` e `GET`. A diferença entre estes métodos é a seguinte:

`POST`

- com `POST`, os dados entrados pelo formulário fazem parte do corpo da mensagem enviada para o servidor;
- a cadeia de caracteres de entrada é lida como entrada padrão de comprimento `CONTENT_LENGTH`;
- é possível transferir grande quantidade de dados.

`GET`

- com `GET`, os dados de entrada do script fazem parte do URL associado à consulta enviada para o servidor (por exemplo, nas consultas a catálogos do tipo Yahoo e Surf);
- a cadeia de entrada é colocada na variável de ambiente `QUERY_STRING`;
- suporta somente até 128 caracteres;
- como os dados da consulta fazem parte do URL, ela pode ser encapsulada em um URL de link ou bookmark;
- um detalhe é que os dados de entrada são copiados no regsitro de acessos ao site; portanto, informações que exigem segurança não devem ser manipuladas por este método.

Métodos de HTTP

O protocolo HTTP utiliza vários métodos de manipulação e organização dos dados. Atualmente, dois métodos são mais utilizados para submeter dados de formulários: `POST` e `GET`. A diferença entre estes métodos é a seguinte:

`POST`

- com `POST`, os dados entrados pelo formulário fazem parte do corpo da mensagem enviada para o servidor;
- a cadeia de caracteres de entrada é lida como entrada padrão de comprimento `CONTENT_LENGTH`;

- é possível transferir grande quantidade de dados.

GET

- com GET, os dados de entrada do script fazem parte do URL associado à consulta enviada para o servidor (por exemplo, nas consultas a catálogos do tipo Yahoo e Surf);
- a cadeia de entrada é colocada na variável de ambiente QUERY_STRING;
- suporta somente até 128 caracteres;
- como os dados da consulta fazem parte do URL, ela pode ser encapsulada em um URL de link ou bookmark;
- um detalhe é que os dados de entrada são copiados no regsitro de acessos ao site; portanto, informações que exigem segurança não devem ser manipuladas por este método.

Áudio e Vídeo

O uso de áudio e vídeo na Internet vem ganhando muito destaque nos últimos dois anos, e é bom saber como usar bem estas mídias.

Áudio e vídeo são classificados como "mídias contínuas", pois são geradas segundo determinadas taxas, devendo ser reproduzidas nessa mesma taxa, para evitar distorções. Quanto mais informações de uma seqüência de áudio ou vídeo digital são armazenados, melhor a qualidade do áudio ou vídeo reproduzido. Isso implica, logicamente, no fato de arquivos de áudio e vídeo serem geralmente muito grandes, o que torna inviável o uso mais freqüente dessas mídias em páginas Web.

Além de procurarmos lidar sempre com pequenos trechos de áudio e vídeo, podemos explorar tecnologias recentes que permitem a transmissão em tempo real

Áudio

< < Seção em construção! > >

Há duas maneiras de inserir som em uma página:

- para browsers Netscape:

```
<EMBED SRC="audio.som">
```

- para browsers Internet Explorer:

```
<BGSOUND SRC="audio.som">
```

Essas formatações, porém, não farão efeito algum quando o browser não estiver configurado para "tocar" música, ou se o computador que receber a página não tiver uma placa de som.
É possível configurar a apresentação do arquivo de som, como veremos nos próximos passos...

Vídeo

< < Seção em construção! > >

Em geral, embutimos um arquivo de video em um documento desta maneira

```
<EMBED SRC="video.vid">
```

Essa formatação, porém, não fará efeito algum quando o browser não estiver configurado para "tocar" o vídeo.

Quando usar GIF e quando usar JPEG

Para alguns tipos de imagens, o formato GIF é superior em qualidade de imagem, tamanho de arquivo, ou ambos. O que precisamos saber é para que tipo de imagens devemos aplicar JPEG.
De maneira geral, JPEG é melhor aplicado a *imagens com variações de cor* - fotografias digitalizadas, por exemplo - coloridas ou com no mínimo 16 níveis em tons de cinza. Quanto mais complexa a imagem, melhor o resultado com JPEG.

GIF é melhor para *imagens com cores lisas* (bordas, áreas com cores "puras") realizando, nesses casos, uma compressão muito melhor do que JPEG faria. Nestes casos, JPEG produz imagens desfocadas. Por isso, imagens em preto-e-branco puros (sem tons de cinza) não devem ser processados em JPEG.

Sobre Gif

Chamamos GIF (em geral, lê-se "guif") as imagens (ícones, ilustrações, e outras) apresentadas nas páginas WWW. GIF é abreviação de *"Graphics Interchange Format"*, desenvolvido pela CompuServe para transmissão de imagens por linhas discadas. Este formato usa codificação LZW (Lempel-Ziv and Welsh) para comprimir imagens e assim reduzir tempos de transmissão. Imagens que seguem este formato são armazenadas em arquivos de extensão .gif. Os formatos GIF mais recentes são os GIF87a e GIF89a

Gif transparente

As imagens que parecem estar embutidas na própria página são *"Gifs transparentes"* - isto é, imagens armazenadas em arquivos .gif, com a aplicação de uma extensão gráfica que esconde uma determinada cor na apresentação da imagem.
O procedimento para fazer imagens com esse efeito é simples. Para quem tem Windows, uma das alternativas é usar o **LView Pro**.

Atendendo a pedidos, aqui estão os passos para obter uma imagem com fundo transparente: vamos tornar transparente o fundo desta imagem à esquerda.

1. Abrir o arquivo da imagem no LView:

2. Na barra de ferramentas, escolher o item **Retouch** e, depois, **Background Color**:

Outro caminho é através da barra de ferramentas, no item **Pallete Entry**.

3. Abre-se uma janelinha na qual escolhemos a cor que queremos deixar "transparente" na imagem:

Animação

http://www.icmsc.sc.usp.br/
manuals/HTML/
arquivo: trans.html

Note que foi selecionado o item **Mask selection using**, que faz com que uma máscara em branco ou preto seja ativada; essa máscara nos mostra quando a cor escolhida é adequada para o efeito que pretendemos. Neste caso foi escolhida uma máscara branca, que cobre a região da figura que não ficará transparente.
Se toda a figura na janela do LView estivesse branca (no caso da máscara em preto, se a figura estivesse totalmente em preto), a cor alaranjada escolhida não ficaria "transparente" no resultado final.
Uma maneira rápida de selecionar a cor, quando a paleta é muito grande, é usar a opção **Dropper**, com a qual selecionamos a cor através de uma caneta.
Escolhido o fundo, acionar **ok**.

4. Finalmente, gravar a imagem como GIF89a:

Este é o resultado obtido:

Se não tiver sucesso, provavelmente será porque o fundo original da imagem não tem uma cor lisa. Abrindo o arquivo em um aplicativo de edição de imagem, e aplicando uma função de lupa ou zoom, é possível ver que o fundo é xadrez, como o exemplo à direita. Pinte o fundo com uma cor lisa, e repita o processo de "transparentização".

Conforme o tipo de arquivo da imagem original, o que temos em vez da janelinha de paleta para escolher a cor do fundo é este seletor de cores de 24 bits:

Dessa maneira fica mais difícil escolher a cor exata. Uma opção é gravar a imagem como GIF e abrir o arquivo novamente; desta vez a janela quadriculada deverá aparecer.
Obs.: Maiores explicações sobre arquivos de imagem, por enquanto, constam apenas do manual em edição ("World-Wide Web: Manual do usuário").

Sobre JPEG

JPEG é um padrão internacional, proposto pelo comitê ISO *"Joint Photographers Expert Group"*.
Esse formato permite a transferência de arquivos por uma grande variedade de plataformas. JPEG é baseado em codificação da imagem por transformação matemática, o que oferece altas taxas de compressão, embora haja perda de informações.
JPEG pode comprimir imagens até um quinto (1/5) do tamanho original, sem perda de qualidade perceptível.
Arquivos de imagens que seguem este formato têm extensão `.jpeg` ou `.jpg`.

Animação

Quem já encontrou uma página com animação, ficou logo "louquinho" para colocar algo parecido na própria homepage.
Grande parte das animações é produzida de maneira semelhante aos desenhos animados: primeiro é preciso ter os desenhos - geralmente, arquivos `*.gif` - compondo a seqüência completa da animação desejada; depois, um *script*, um *applet* Java, ou um tipo especial de arquivo, ficará encarregado de mostrar os desenhos na ordem especificada para gerar a animação.

Algumas animações são produzidas por **scripts CGI**, outras por **Java**, e outras por **blocked GIFs** (GIFs animadas).
Esta animação simples é resultado desta última "técnica".
Outra possibilidade é realizar animações em programas próprios para apresentação, gerando arquivos de vídeo ou outros formatos que possam ser apresentados através de *plug-ins*.

Gif animado

Um GIF animado é o resultado do armazenamento da seqüência de animação em um só arquivo. Softwares comuns para fazer GIFs animadas são o GifCon e o Paint Shop para Windows, e o whirlgif em UNIX. Para este exemplo usaremos o GifCon, que está disponível para cópia aqui.

O GifCon tem um *wizard* que torna o processo de criação de animações muito rápido e simples.

Todas as opções *default* para a criação da imagem animada já estão selecionadas no *wizard* e, mesmo que não se entenda muito bem o significado dos passos seguidos, a animação será composta sem problemas.

As opções oferecidas são estas:

- uso da imagem - para documentos Web ou em outro ambiente;
- *loop* contínuo ou não - quantas vezes a animação será apresentada (indefinidamente ou uma vez só);
- qualidade da imagem animada;

- tempo entre cada imagem da animação

Depois de selecionar os itens desejados (ou simplesmente seguir a configuração padrão), selecionamos os arquivos de imagem que irão compor a animação, completando os passos do *wizard*.
Para o exemplo, escolhemos as seguintes imagens:

http://www.icmsc.sc.usp.br/
manuals/HTML/
arquivo: gifanim.html

Ao completar os passos do *wizard*, temos a janela do GifCon que mostra os controles do arquivo de animação e a seqüência de imagens.

Pode-se editar várias características da animação mesmo depois de se gravar o arquivo final (tempo entre imagens, loop, etc).
Na mesma janela do GifCon, podemos ver cada uma das imagens que formam a animação:

A animação resultante da seqüência escolhida foi esta:
Entre as facilidades do GifCon está a possibilidade de transformar um arquivo `AVI` em arquivo de `GIF` animado.
O whirlgif funciona de maneira semelhante quanto à montagem da animação, porém sua interface é muito pobre, sendo necessário trabalhar em linha de comando

Os diversos recursos interativos

< < Seção em construção! > >

Como visto na seção sobre Interação, podemos desenvolver scripts ou programas que atuem do lado do servidor ou do cliente Web, promovendo a interação do leitor com as informações apresentadas em uma página.
Os recursos geralmente utilizados do lado do servidor são:

- CGI Scripts
- SSI (*Server-Side Includes*)

Com tais recursos, podemos trabalhar com acesso a bases de dados, tratar dados entrados por formulários, realizar contagem de acessos, oferecer diversas informações personalizadas.
Do lado do cliente, temos os seguintes recursos:

- Java
- JavaScript

Tais recursos podem ser aplicados juntamente com scripts do lado do servidor; quando sozinhos, têm acesso somente a informações oferecidas localmente.
É preciso lembrar que existem os *servlets* Java e o LiveScript, "versões" e "extensões" de Java e JavaScript executados do lado do servidor

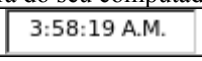
JavaScript

Com JavaScript implementamos diversos recursos interessantes.

Assim, é possível dizer: seu browser é o

Microsoft Internet Explorer (Mozilla) Versão 4.0 (compatible; MSIE 5.0; Windows 95; DigExt).

Outro exemplo é este reloginho, que mostra a hora do seu computador:



Podemos também cumprimentar o leitor:
(de acordo com a hora local!)

Credo, 3 da manhã! Você é um vampiro ou o que?!?!

Os roteiros JavaScript permitem acesso a:

- atributos relacionados ao documento apresentado
- informações disponíveis do lado do cliente (browser).

... mas não serve para gerarmos scripts de formulários, etc.

Problema: Só o Netscape apresenta bem todos os "JavaScripts".

Ao fazer páginas, teste os resultados em outros browsers