HTML

Prof. Marcos Alexandruk

UNINOVE

SUMÁRIO

INTERNET

Rede mundial de informação, a *rede das redes*. Criada nos EUA, tornou-se uma associação mundial de redes interligadas.

Utiliza a arquitetura de protocolos de comunicação TCP/IP.

INTRANET

Redes corporativas que utilizam da tecnologia de comunicação utilizada pela INTERNET.

Utilizadas na comunicação interna da própria empresa.

EXTRANET

Conjunto de INTRANETS interligadas através da INTERNET.

HIPERTEXTO

Um método de acoplamento de informações relacionadas sem a necessidade de um sistema hierárquico ou de menus. O usuário pode *navegar* através de um documento seguindo as *ligações (links)* pré-definidas.

MULTIMÍDIA

Combinação interativa de textos, gráficos, animações, imagens, áudio e vídeo exibidos por e sob o controle de um PC.

HIPERMÍDIA

Combinação de HIPERTEXTO e MULTIMÍDIA. O usuário pode ler um texto, visualizar uma imagem, animação ou vídeo e ouvir um áudio.

BROWSER

Software que fica do lado *cliente* utilizado para apresentar os documentos na tela do PC. Através do BROWSER é possível ler, imprimir, visualizar imagens e ouvir áudio.

Protocolo de rede é um conjunto de regras utilizadas pelos computadores de uma rede para estabelecer a comunicação entre eles.

Assim como na lingugem falada, onde duas pessoas somente se comunicam se falarem a mesma língua, dois computadores só podem se comunicar se utilizarem o mesmo protocolo.

TCP/IP (Transmition Control Protocol/Internet Protocol)

É um conjunto de protocolos de comunicação utilizado para troca de dados entre computadores em ambientes de redes locais e/ou remotas.

A arquitetura TCP/IP surgiu em 1975 na rede Arpanet (criada em 1969 pela ARPA - Advanced Research Projects Agency – órgão do departamento de defesa dos EUA).

As especificações dos protocolos TCP/IP são públicas e genéricas, o que permite sua implementação por diversos fabricantes.

Todos os sistemas operacionais atuais (Windows, Linux, Unix, MacOS, Netware, etc.) fornecem uma implementação do protocolo TCP/IP

HTTP (HiperText Transfer Protocol)

HTTP é o protocolo usado pela WWW – World Wide Web – para transmitir recursos. Um recurso é algum tipo de informação que pode ser identificada por um URL – Uniform Resource Locator.

O tipo mais comum de recurso é um arquivo no formato HTML, mas um recurso pode ser o resultado de uma consulta gerada dinamicamente, a saída de um script CGI, uma imagem ou qualquer outra forma de informação.

FTP (File Tranfer Protocol)

O protocolo FTP permite a transferência de arquivos entre um computador local e um servidor remoto, sendo muito utilizado para upload e download de arquivos na Internet.

Permite a nagevação em uma parte da estrutura de diretórios do servidor remoto para a localização do arquivo desejado.

O FTP diferencia arquivos de texto (ASCII) e arquivos binários (imagens, aplicativos, etc.)

TELNET

O protocolo Telnet permite o acesso tipo terminal a um computador remoto.

A conexão a um sistema utilizando o protocolo Telnet permite interagir com o servidor como se estivesse utilizando a console pó próprio servidor. É possível executar comandos do sistema operacional a partir de linhas de comandos, executar scripts e utilizar sistemas instalados no host remoto.

No processo de conexão é solicitado o nome do usuário e a senha para autenticação do terminal remoto.

DNS (Domain Name System)

O serviço DNS tem como objetivo facilitar o modo de especificar um host numa rede TCP/IP, utilizando um nome “amigável” em vez do seu endereço IP.

No servidor DNS são cadastrados os hosts e seus endereços IP. Quando um computador (cliente DNS) precisa saber a qual endereço IP um nome se refere, solicita a resolução deste nome ao servidor DNS.

Para distribuir essa tarefa entre diversos servidores, criou-se uma estrutura hierárquica, agrupando computadores em grupos chamados domínios e instalando um ou mais servidores de DNS para cada domínio.

CONSIDERAÇÕES INICIAIS

Para desenvolver um website é fundamental conhecer HTML (Hipertext Markup Language ou Linguagem de Hipertexto baseada em Marcas).

Produzir um documento HTML é muito parecido com o que fazemos ao formatar um texto num editor como o *Word*.

Definiu-se que os documentos HTML fossem gerados num formato muito simples, o txt. Por isso, os documentos HTML podem ser produzidos à partir de qualquer editor de texto. No UNIX o **vi** é muito utilizado e no Windows temos o **NotePad** (Bloco de Notas).

MARCAS (TAGS)

Para que o conteúdo de uma página Web seja exibido devidamente formatado, utiliza-se as marcas-padrão (ou tags, em inglês).

Por definição, as marcas são delimitadas pelos sinais **<** e **>**.

Há um conjunto de palavras reservadas que são interpretadas pelo browser, responsável por apresentar o documento devidamente formatado.

Por exemplo, a tag centraliza o texto na página,
 insere uma "quebra" de linha, <b> altera o texto para negrito, etc.

A maioria das marcas trabalham em pares, indicando o início e o fim de um trecho do documento. A marca para finalizar o bloco é idêntica a do início, porém é precedida por uma barra **/**. Exemplo: ** ** ... ** **

Algumas marcas não precisam de complemento ou finalização. Ex: **
**

EDITORES

Desde o surgimento do padrão HTML surgiram vários editores específicos para esta linguagem.

Há dois tipos básicos:

- WYSIWYG (What You See IS What You Get – *o que você vê é o que você obtém*). Exemplos: FrontPage e DreamWeaver.
- Editores com menus específicos da linguagem que apresentam o conjunto de tags conforme padrão HTML. Exemplos: FirstPage, HomeSite, HotDog.

Embora os editores WYSIWYG estejam cada dia mais sofisticados e ajudem muito na produtividade, é importante para o aprendizado da linguagem HTML esquecer um pouco esses produtos e utilizar um editor de textos simples para desenvolver páginas web.

Há pelo menos dois motivos importantes para conhecer a linguagem apesar das facilidades oferecidas por esses produtos:

- Os editores WYSIWYG muitas vezes inserem marcações desnecessárias que são indesejáveis para uma página "enxuta", que possa ser carregada mais rapidamente.

Às vezes é necessário fazer um "ajuste fino" nas páginas o que pode ser difícil ou impossível se contarmos exclusivamente com os recursos de um editor WYSIWYG.

BLOCOS BÁSICOS

```
<html>
<head>
<title>Título da página</title>
</head>
<body>
Conteúdo da página
</body>
</html>
```

TAGS HTML

<html> ... </html>

Indicam o início e o fim de um documento. Todo o resto deve estar entre estas marcas.

<head> ... </head>

Delimitam a seção de cabeçalho do documento. Trata-se da primeira seção do documento.

<title> ... </title>

Indicam o título do documento, que será apresentado na barra superior do browser. Estas marcas são internas à seção <head> ... </head>

```
<head>
<title>Título da página</title>
</head>
```

<meta> ... </meta>

Utilizada para indicar instruções especiais ao browser do cliente ou servidor, realizando uma operação de análise gramatical. Estas tags também são internas ao cabeçalho <head> de um documento.

```
<meta name="nome" http-equiv="nome" content="valor">
```

name	authors, keywords, description
http-equiv	creation-date, content type, expires, refresh
content	depende do valor de name ou http-equiv

Exemplo:

```
<html>
<head>
<title>Exemplo – marcador meta</title>
<meta name="authors" content="Nome do Autor">
<meta name="keywords" content="palavra_chave1, palavra_chave2">
<meta name="description" content="Descricao da pagina ou site">
<meta http-equiv="creation-date" content="01-fev-2004 14:00:00 GMT">
<meta http-equiv="content-type" content="text/html; charset=isso-8859-1">
<meta http-equiv="expires” content="31-jan-2005 14:00:00 GMT">
<meta http-equiv="refresh" content="10; URL=refresh.htm">
</head>
<body>
Dentro de dez segundos a página refresh.htm será exibida.
</body>
</html>
```

<body> ... </body>

Marca o início e o final do *corpo* da página e determina algumas propriedades. Entre estas marcas estará contida a maior parte do conteúdo a ser apresentado, textos, imagens, etc.

```
<body topmargin="n" leftmargin="n" background="arquivo"
      bgcolor="cor" text="cor" link="cor" alink="cor" vlink="cor">
...
</body>
```

topmargin	0, 1, 2 ...	margem superior da página
leftmargin	0, 1, 2 ...	margem esquerda da página
background	*arquivo.ext*	imagem para o fundo da página
bgcolor	#FFFFFF	cor de fundo da página
text	#000000	cor padrão do texto da página
link	#0000FF	cor dos links da página (v. LINKS)
alink	#00FF00	cor do link ativo (i.e. “clicado”)
vlink	#FF0000	cor dos links já visitados

<h*n*> ... </h*n*>

Utilizada para formatar títulos e subtítulos. Podemos utilizar até seis níveis de títulos ou "headings", que são numerados de 1 (o maior) a 6 (o menor).

```
<h1>Título nível 1</h1>
...
<h6>Título nível 6</h6>
```

Os elementos de título podem ter o parâmetro opcional align:

```
<h1 align="alinhamento">Título nível 1 alinhado</h1>
```

left	alinhamento à esquerda
right	alinhamento a direita
center	alinhamento centralizado

 ...

Determina o tamanho, o tipo e a cor da fonte.

```
<font size=n face="tipo da fonte" color="cor"> Texto </font>
```

size	1 a 7		Configura o tamanho da fonte
face	"Times", "Arial", etc.		Configura o tipo da fonte
color	#FFFF00	yellow	Configura a cor da fonte
	...		
	#00FF00	green	

Além das propriedades já citadas, às vezes é necessário destacar palavras e frases. Para isso podemos utilizar as seguintes tags:

negrito	**negrito**
itálico	*itálico*
<u>sublinhado</u>	sublinhado
<s>riscado</s>	~~riscado~~
<big>maior</big>	maior
menor	menor
_{subscrito}	texto subscrito
^{sobrescrito}	texto sobrescrito

...

Indica o início e o final (opcional) de um parágrafo.

```
<p align="alinhamento">
Primeiro parágrafo.
</p>
<p align="alinhamento">
Segundo parágrafo.
</p>
```

left	esquerda
direita	direita
center	central
justify	justificado

Insere uma quebra de linha no texto.

```
Neste ponto <br>
a linha foi "quebrada"
```

<pre> ... </pre>

Conserva o texto da mesma forma como foi digitado no editor de textos.

```
<pre>
Este texto terá a
formatação que foi
aplicada pelo editor.
</pre>
```

<hr>

Desenha uma linha (horizontal) na página.

```
<hr size="altura" width="comprimento" align="alinhamento" noshade>
```

size	número inteiro (1, 2, 3 ...)
witdh	comprimento (em pixels ou porcentagem)
align	left, right, center
noshade	preenche a linha

<blockquote> ... </blockquote>

Utilizado para deslocar um bloco de texto.

```
<blockquote>
  Texto deslocado
</blockquote>
```

...

Utilizado para posicionar um elemento (texto, imagem, etc.) na página.

```
<div align="alinhamento">Texto ou imagem</div>
```

left	esquerda
direita	direita
center	central
justify	justificado

<script> ... </script>

Utilizado para inserir o código de uma linguagem de programação voltada à Internet (JavaScript, VBScript, etc.). Pode ser inserida dentro do cabeçalho <head> ou fazer parte do corpo <body> da página.

```
<script language="linguagem">
... Inserir o código da linguagem.
</script>
```

<!-- ... -->

Utilizado para inserir um comentário na página html. O comentário não será visível, destinando-se basicamente para documentar o código.

LISTAS

LISTAS ORDENADAS

Insere uma lista ordenada, com marcadores do tipo: 1, 2, 3, ..., a, b, c, ...

```
<ol type="a">
  <li> Item 1</li>
  <li> Item 2</li>
</ol>
```

type: 1, a, I, i, A, ...

LISTAS NÃO ORDENADAS

Insere uma lista com marcadores diferenciados, sem uma ordem lógica.

```
<ul type="square">
  <li>Item 1</li>
  <li>Item 2</li>
</ul>
```

type: disk, square e circle

LISTAS DE DEFINIÇÃO <dl>

Insere uma lista onde cada item possui uma definição ou descrição do item.

```
<dl>
  <dt>Item 1
    <dd> Descrição 1
  <dt>Item 2
    <dd> Descrição 2
</dl>
```

LINKS

```
<a href="arquivo"> ... </a>
```

Um link liga uma página a outras páginas, e-mails e a pontos específicos da própria página ou de outras.

LOCAIS

Link com páginas ou arquivos de um mesmo site (inclusive localizados em outros diretórios ou pastas).

```
<a href="pagina2.htm" title="Texto explicativo">LINK LOCAL</a>
```

ou

```
<a href="./outro_diretorio/pagina2.htm">LINK LOCAL</a>
```

O parâmetro title faz com que um texto explicativo (referente ao link) seja exibido próximo ao mouse quando este encontrar-se sobre o link.

REMOTOS

Link com páginas ou arquivos de outro site.

```
<a href="http://www.outrosite.com.br/pagina2.htm">LINK REMOTO</a>
```

NA PRÓPRIA PÁGINA

Link para outras partes da própria página.

```
<a href="#marcador1">LINK NA PRÓPRIA PÁGINA</a>

<a name="marcador1"></a>Texto ...
```

PARA UM E-MAIL

Este link abre o programa cliente de e-mail do usuário e preenche o campo de cabeçalho “para:” com o mesmo valor indicado em “mailto:”

```
<a href="mailto:destinatario@empresa.com.br">LINK PARA E-MAIL</a>
```

TARGET

```
<a href= ... target= ...> ... </a>
```

Atributo que direciona o browser para exibir o conteúdo de uma URL em uma janela específica.

```
<a href="arquivo.ext" target="nome">LINK</a>
```

_blank	carrega o conteúdo em uma nova janela
_parent	exibe o conteúdo na "janela-pai"
_self	exibe o conteúdo na mesma janela
_top	carrega o conteúdo numa janela que preenche o espaço de visualização do browser

CORES

```
<link ... alink ... vlink>
```

Define uma cor para cada um dos estados que um link pode assumir:

link	link não visitado
alink	link ativo (“clicado”)
vlink	link já visitado

São utilizados com a tag `<body>` e valem para toda a página.

```
<body link="cor" alink="cor" vlink="cor">
```

IMAGENS

TIPOS

JPEG (Joint Photografic Experts Group)

O formato JPEG foi criado para que imagens “true-color” (16 milhões de cores) pudessem ser armazenadas em arquivos pequenos.

Imagens JPEG não podem ser transparentes, portanto sempre aparecem retangulares na tela.

Diferente do GIF, que tem uma compressão padrão, o JPEG tem uma compressão váriável. Porém, quanto mais comprimido o arquivo, maior será a perda da qualidade da imagem. Numa escala de 1 a 100, a compressão padrão situa-se em torno de 33, mas é recomendável testar a qualidade da imagem antes de publicá-la.

GIF (Graphics Interchange Format)

Os arquivos GIF podem ter no máximo 256 cores (8 bits por pixel).

O padrão permite o entrelaçamento e a transparência de imagens. Uma imagem entrelaçada (interlaced) pode ser vista à medida que vai sendo carregada (melhorando a definição aos poucos) e uma imagem transparente permite visualizar, através de certas partes, o que está por trás. O recurso de transparência oferece a impressão de que as imagens não são retangulares.

PNG (Portable Network Graphics)

Outro formato aceito pelos navegadores.

INSERÇÃO

```
<img src= ...>
```

Insere uma imagem na página.

```
<img src="arquivo" border="n" alt="texto" width="n" height="n">
```

src	nome do arquivo de imagem
border	tamanho da borda (em pixels)
alt	texto associado à imagem
width	largura (tamanho ou percentual)
height	altura (tamanho ou percentual)

MAPEAMENTO

```
<map name="mapa">
```

Divide a imagem em vários *setores* ou *pedaços*. Cada *setor* corresponde a um link diferente.

```
<img src="arquivo" usemap="#mapa">

<map name="mapa">
  <area shape="tipo" coords="valor" href="URL">
  <area shape="tipo" coords="valor" href="URL">
  ...
  <area shape="tipo" coords="valor" href="URL">
</map>
```

O "valor" é dado em pixels, de acordo com o "tipo":

```
tipo="circ"      valor="x,y,r"
tipo="rect"      valor="x1,y1,x2,y2"
tipo="poly"      valor="x1,y1,x2,y2,x3,y3, ..."
```

SONS

INSERÇÃO

`<embed src= ...>`

Insere um arquivo de som na página (formatos aceitos: WAV, AU, MIDI).

Internet Explorer:

```
<bgsound src="arquivo" loop="quantidade">
```

src	nome do arquivo de música
loop	"infinite" ou qualquer valor inteiro

Internet Explorer e Netscape:

```
<embed src="arquivo" autostart="true" hidden="true" width="1" height="1">
```

src	nome do arquivo de música
loop	"infinite" ou qualquer valor inteiro
autostart	a música será iniciada quando a página for carregada
hidden	esconde o "plugin" (programa necessário para tocar a música)
width=1 e height=1	assegura que o objeto incorporado não utilizará nenhum espaço

`<noembed> ... </noembed>`

Utilizado para compatibilidade com browsers que não suportam <embed>.

TABELAS

<table> ... </table>

Cria uma tabela, formata bordas, tamanho e alinhamento.

```
<table border="n" cellspacing="n" cellpadding="n" width="n"
       height="n" align="alinhamento">
...
</table>
```

border	largura da borda	0, 1, 2, ...
cellspacing	espaço entre o conteúdo e o lado da célula	0, 1, 2, ...
cellppading	espaço entre as células adjacentes	0, 1, 2, ...
width	largura da tabela (em pixels ou percentual)	250, 60% ...
height	altura da tabela (em pixels ou percentual)	100, 30% ...
align	alinhamento da tabela	left, right, center

<tr>

Especifica o início e o fim de uma linha da tabela.

```
<tr align="valor" valign="valor">
...
</tr>
```

align	alinhamento horizontal	left, center, right
valign	alinhamento vertical	top, center, bottom

<td> ... </td>

Especifica o conteúdo de uma célula da tabela.

```
<td colspan="n" rowspan="n" width="n" align="valor">
...
</td>
```

colspan	número de colunas ocupadas pela célula	2, 3, ...
rowspan	número de linhas ocupadas pela célula	2, 3, ...
width	largura da célula (em pixels ou percentual)	80, 15% ...
align	alinhamento horizontal	left, center, right

<th> ... </th>

Define as células de cabeçalho. Apresenta características idênticas às células de dados <td>, exceto pelo alinhamento horizontal padrão, que é centralizado, e pela utilização de fontes em negrito.

TABELA – EXEMPLO COMPLETO

<table>
<tr><td></td><td></td><td colspan="2"></td></tr>
<tr><td rowspan="2"></td><td></td><td></td><td></td></tr>
<tr><td></td><td colspan="2" rowspan="2"></td></tr>
<tr><td></td><td></td></tr>
</table>

```
<html>
<head>
<title>TABELA</title>
</head>
<body>
<table border="1" cellpadding="0" cellspacing="0" width="100%">
  <tr>
    <td> </td>
    <td> </td>
    <td colspan="2"> </td>
  </tr>
  <tr>
    <td rowspan="2"> </td>
    <td> </td>
    <td> </td>
    <td> </td>
  </tr>
  <tr>
    <td> </td>
    <td colspan="2" rowspan="2"> </td>
  </tr>
  <tr>
    <td> </td>
    <td> </td>
  </tr>
</table>
</body>
</html>
```

FORMULÁRIOS

<form> ... </form>

Elementos que delimitam um formulário. Todos os outros elementos devem ficar entre eles.

```
<form action="script" method="método" target="valor">
...
</form>
```

action	endereço do programa que vai receber os dados do formulário
method	"get" ou "post"
target	é opcional, necessário se a resposta será exibida em um frame diferente do formulário

<input type= ... >

Determina como será a entrada de dados nos campos de um formulário.

```
<input type="tipo" name="nome" value="valor" size="n" maxlength="n">
```

type	define o tipo do elemento (caixa de texto, botão, etc.)
name	nome ou identificador do campo
value	valor do campo (opcionalmente pode ser predefinido)

TEXT

O valor "text" no parâmetro "type" indica que o campo será de texto.

```
<input type="text" name="nome" value="valor" size="n" maxlength="n">
```

size	define o tamanho do campo que será mostrado na página
maxlenght	número máximo de caracteres que podem ser digitados

PASSWORD

Funciona da mesma forma que o valor "text", exceto que todas as letras digitadas aparecem como um asterisco *.

```
<input type="password" name="nome" value="valor" size="n" maxlength="n">
```

RADIO

O valor "radio" no parâmetro "type" define botões de escolha. Eles são usados onde somente UMA opção pode ser selecionada.

```
Coloque sua pergunta aqui ...
<input type="radio" name="pesquisa1" value="opção1">
<input type="radio" name="pesquisa1" value="opção2" checked>
<input type="radio" name=vpesquisa1" value="opção3">
```

type	define o tipo do elemento, neste caso "radio" (botão de escolha)
name	deve ser o mesmo para todos os campos no grupo, pois está identificando a questão formulada
value	valor do campo que será passado ao ao programa interpretador do formulário
checked	deve ser usado quando você desejar que uma das opções esteja selecionada como padrão (Ex: Deseja receber nossos e-mails?)

CHECKBOX

Deve ser usado quando UMA ou MAIS opções são válidas ou aceitáveis.

```
Coloque sua pergunta aqui ...
<input type="checkbox" name="pesquisa2" value="opção1">
<input type="checkbox" name="pesquisa2" value="opção2" checked>
<input type="checkbox" name="pesquisa2" value="opção3">
```

type	define o tipo do elemento, neste caso "checkbox"
name	deve ser diferente para cada campo
value	valor do campo que será passado ao programa interpretador do formulário
checked	deve ser usado quando você desejar que uma das opções esteja selecionada como padrão

RESET

Define um botão que limpa todos dos campos, colocando os mesmos valores de quando a página foi carregada.

No parâmetro "value" pode-se definir o que estará escrito no botão. Caso nenhum valor seja definido, aparecerá escrito somente "Reset".

```
<input type="reset" value="limpar">
```

SUBMIT

Define um botão que aciona o envio das informações preenchidas no formulário ao programa interpretador (definido no campo "action" do elemento "form".

Caso nenhum valor seja definido no parâmetro "value", aparecerá escrito "Submit" no botão correspondente.

```
<input type="submit" value="enviar">
```

IMAGE

Alternativamente ao botão "submit", pode-se utilizar uma imagem. Para isso, existe o tipo "image".

```
<input type="image" name="enviar" src="URL" alt="texto">
```

src	nome ou endereço (URL) do arquivo de imagem
alt	texto que será mostrado caso a imagem não seja carregada

HIDDEN

O valor "hidden" (oculto) no parâmetro "type" define dados que devem ser passados ao programa interpretador, mas que não são visíveis ao usuário.

```
<input type="hidden" name="nome" value="valor">
```

name	identifica o dado
value	define o valor que deve ser passado

<select name= ... > ... </select>

O elemento "select" possibilita definir uma lista de opções.

```
<select name="nome" size="tamanho">
<option value="opção1">opção1</option>
<option value="opção2">opção2</option>
<option value="opção3">opção3</option>
</select>
```

<textarea name= ... > ... </textarea>

O elemento "textarea" (área de texto) permite definir um campo de texto com várias linhas.

```
<textarea name="nome" rows="n" cols="n">
</textarea>
```

name	nome da caixa de texto
rows	número de linhas da caixa de texto
cols	número de caracteres (colunas) cada linha possui

FORMULÁRIO - EXEMPLO COMPLETO

```
<html>
<head>
<title>FORMULARIO</title>
</head>
<body>
<p>
<form action="AnyForm.cgi" method="post">
<input type="hidden" name="AnyFormMode" value="mail">
<input type="hidden" name="AnyFormDisplay" value="MeuFormulario">
<input type="hidden" name="AnyFormTo" value="usuario@empresa.com.br">
<input type="hidden" name="AnyFormSubject" value="MensagemDoSite">
<input type="text" name="AnyFormFrom" size="40">
<p>
Entre com o seu nome:
<input type="text" name="AnyFormMode" value="mail">
<p>
Você gostou da página?
<input type="radio" name="gostou" value="sim">sim
<input type="radio" name="gostou" value="mais ou menos">mais ou menos
<input type="radio" name="gostou" value="não">não
<p>
Qual a página que você mais gostou?
<select name="Melhor Página">
<option value="Links">Links</option>
<option value="Curriculum">Curriculum</option>
<option value="Fotos">Fotos</option>
<option value="Biblioteca">Biblioteca</option>
</select>
<p>
Deixe aqui seus comentários: <br>
<textarea name="Comentarios" rows="5" cols="40">
</textarea>
<p>
Deseja receber nossos e-mails diariamente?
<input type="checkbox" name="Deseja receber e-mails" value="sim"> sim
<p>
<input type="submit" value="Enviar">
<input type="reset" value="Limpar">
</form>
</body>
</html>
```

FRAMES

Dividem a janela do browser em diferentes áreas, cada uma comportando-se como mini-páginas. Cada frame pode exibir, separadamente, um arquivo HTML.

Um uso comum para os frames é a navegação. A janela do browser é dividida em duas áreas, e os botões de navegação ou links são carregados em um dos frames, com o resto do site sendo exibido pelo outro. Clicando em um link no frame de nagevação, o frame principal altera a exibição da página selecionada e mantém o frame de navegação imutável.

Isso permite ter uma única página de navegação para o site inteiro, que é carregada somente uma vez e permanece ali enquanto o visitante estiver acessando o site.

<frameset>

```
<frameset frameborder="yes/no" border="largura" rows="n" cols="n">
```

frameborder	yes/no
border	largura da borda
rows	altura da "linha" (em % ou pixels)
cols	largura da "coluna" (em % ou pixels)

Desenvolver um site utilizando frames envolve criar o documento de frameset para dividir a janela do browser em várias partes.

O frameset é um documento HTML padrão, conforme o exemplo a seguir:

```
<html>
<title>Dois frames horizontais</title>
<frameset rows="20%,80%">
<frame src="superior.htm">
<frame src="inferior.htm">
</frameset>
</html>
```

No exemplo acima, um frame ocupará 150 px da altura da página e o outro 80%.

Não há o elemento <body> no frameset, e as páginas que serão carregadas nos frames são especificadas na declaração do próprio frameset (superior.htm e inferior.htm).

Para criar frames verticais:

```
<html>
<title>Dois frames verticais</title>
<frameset cols="20%,80%">
<frame src="esquerdo.htm">
<frame src="direito.htm">
</frameset>
</html>
```

Para criar três frames, com a área superior cobrindo toda a largura da página e abaixo dois frames dividindo verticalmente a área restante.

```
<html>
<title>Um frame horizontal e dois frames verticais</title>
<frameset rows="20%,80%">
  <frame src="superior.htm">
  <frameset cols="20%,80%">
    <frame src="inferior_esquerdo.htm">
    <frame src="inferior_direito.htm">
  </frameset>
</frameset>
</html>
```

Nos exemplos apresentados, cada frame ocupa um percentual da página (20%, 80%). No entanto, é possível utilizar medidas baseadas em pixels, conforme exemplo a seguir:

```
<frameset rows="100,*">
```

O primeiro frame ocupará 100 pixels e o segundo ocupará o restante do espaço inferior da página (janela do browser).

<frame>

```
<frame src="URL_ pagina" name="nome_ frame" scrolling="yes/no/auto nosesize">
```

src	endereço do arquivo HTML a ser apresentado nesta parte
name	nome dado a uma parte específica, será utilizado pelo atributo target
scrolling	indica a existência da barra de rolagem
noresize	informa que a parte especificada não pode ser redimensionada

Esse marcador especifica que página estará em uma determinada parte.

<noframes>

```
<noframes>
  ... tags HTML e página a ser exibida
</noframes>
```

Utilizado para compatibilidade com browsers antigos que não reconhecem frames, este marcador evita que ocorram erros.

FRAMES - EXEMPLO COMPLETO

index.htm

```
<html>
<head>
<title>FRAMESET</title>
</head>
<frameset cols="150,*" frameborder="no" border="0">
  <frame src="menu.htm" name="esquerda" scrolling="no" noresize>
  <frame src="pagina1.htm" name="principal">
</frameset>
</html>
```

menu.htm

```
<html>
<head>
<title>MENU</title>
</head>
<body bgcolor=“#CCCCCC”>
<a href="pagina1.htm" target="principal">PÁGINA 1</a>
<a href="pagina2.htm" target="principal">PÁGINA 2</a>
</html>
```

pagina1.htm

```
<html>
<head>
<title>PAGINA 1</title>
</head>
<body>
Esta é a página 1.
</body>
</html>
```

pagina2.htm

```
<html>
<head>
<title>PAGINA 2</title>
</head>
<body>
Esta é a página 2.
</body>
</html>
```

FRAMES FLUTUANTES <iframe>

<iframe width=*n* height=*n* src="*arquivo*" name="*nome*">

width	largura do frame
height	altura do frame
src	endereço do arquivo HTML a ser apresentado nesta parte
name	nome dado a uma parte específica, será utilizado pelo atributo target

EXEMPLO:

index.htm

```
<html>
<head>
<title>FRAMES FLUTUANTES</title>
</head>
<body>
<iframe width=250 height=200 src="quadro1.htm" name="quadro">
</body>
</html>
```

quadro1.htm

```
<html>
<head>
<title>QUADRO 1</title>
</head>
<body bgcolor="#003366">
FRAME FLUTUANTE (IFRAME)
</body>
</html>
```

EXEMPLO – ÁLBUM DE FOTOS:

index.htm

```
<html>
<head>
<title>ALBUM DE FOTOS</title>
</head>
<body>
<a href="foto1.jpg" target="fotos">FOTO 1</a><br>
<a href="foto2.jpg" target="fotos">FOTO 2</a><br>
<a href="foto3.jpg" target="fotos">FOTO 3</a><br>
<iframe width=250 height=200 src="foto1.jpg" name="fotos">
</body>
</html>
```

CARACTERES ESPECIAIS E ACENTUAÇÃO

¡	¡	¡
¢	¢	¢
£	£	£
¤	¤	¤
¥	¥	¥
¦	¦	¦
§	§	§
¨	¨	¨
©	©	©
ª	ª	ª
«	«	«
¬	¬	¬
®	®	®
¯	¯	¯
°	°	°
±	±	±
²	²	²
³	³	³
´	´	´
µ	µ	µ
¶	¶	¶
·	·	·
¸	¸	¸
¹	¹	¹
º	º	º
»	»	»
¼	¼	¼
½	½	½
¾	¾	¾
¿	¿	¿
À	À	À
Á	Á	Á
Â	Â	Â
Ã	Ã	Ã
Ä	Ä	Ä
Å	Å	Å
Æ	Æ	Æ
Ç	Ç	Ç
È	È	È
É	É	É
Ê	Ê	Ê
Ë	Ë	Ë
Ì	Ì	Ì
Í	Í	Í
Î	Î	Î
Ï	Ï	Ï
<	<	
>	>	

Ð	Ð	Ð
Ñ	Ñ	Ñ
Ò	Ò	Ò
Ó	Ó	Ó
Ô	Ô	Ô
Õ	Õ	Õ
Ö	Ö	Ö
×	×	×
Ø	Ø	Ø
Ù	Ù	Ù
Ú	Ú	Ú
Û	Û	Û
Ü	Ü	Ü
Ý	Ý	Ý
Þ	Þ	Þ
ß	ß	ß
à	à	à
á	á	á
â	â	â
ã	ã	ã
ä	ä	ä
å	å	å
æ	æ	æ
ç	ç	ç
è	è	è
é	é	é
ê	ê	ê
ë	ë	ë
ì	ì	ì
í	í	í
î	î	î
ï	ï	ï
ð	ð	ð
ñ	ñ	ñ
ò	ò	ò
ó	ó	ó
ô	ô	ô
õ	õ	õ
ö	ö	ö
÷	÷	÷
ø	ø	ø
ù	ù	ù
ú	ú	ú
û	û	û
ü	ü	ü
ý	ý	ý
þ	þ	þ
ÿ	ÿ	ÿ
&	&	
"	"	

TABELAS DE CORES

HTML STANDARD COLORSET

aqua	#00FFFF	navy	#000080
black	#000000	olive	#808000
blue	#0000FF	purple	#800080
fuchsia	#FF00FF	red	#FF0000
gray	#808080	silver	#C0C0C0
green	#008000	teal	#008080
lime	#00FF00	yellow	#FFFF00
maroon	#800000	white	#FFFFFF

HTML EXTENDED COLORSET

aliceblue	#F0F8FF	gainsboro	#DCDCDC	mistyrose	#FFE4E1
antiquewhite	#FAEBD7	ghostwhite	#F8F8FF	moccasin	#FFE4B5
aquamarine	#7FFFD4	gold	#FFD700	navajowhite	#FFDEAD
azure	#F0FFFF	goldenrod	#DAA520	oldlace	#FDF5E6
beige	#F5F5DC	greenyellow	#ADFF2F	olivedrab	#6B8E23
bisque	#FFE4C4	honeydew	#F0FFF0	orange	#FFA500
blachedalmond	#FFEBCD	hotpink	#FF69B4	orangered	#FF4500
blueviolet	#8A2BE2	indianred	#CD5C5C	orchid	#DA70D6
brown	#A52A2A	indigo	#4B0082	palegoldenrod	#EEE8AA
burlywood	#DEB887	ivory	#FFFFF0	palegreen	#98FB98
cadetblue	#5F9EA0	khaki	#F0E68C	paleturquoise	#AFEEEE
chartreuse	#7FFF00	lavender	#E6E6FA	palevioletred	#DB7093
chocolate	#D2691E	lavenderblush	#FFF0F5	papayawhip	#FFEFD5
coral	#FF7F50	lawngreen	#7CFC00	peachpuff	#FFDAB9
cornflowerblue	#6495ED	lemonchiffon	#FFFACD	peru	#CD853F
cornsilk	#FFF8DC	lightblue	#ADD8E6	pink	#FFC0CB
crimson	#DC1436	lightcoral	#F08080	plum	#DDA0DD
cyan	#00FFFF	lightcyan	#E0FFFF	powderblue	#B0E0E6
darkblue	#00008B	lightgoldenrodyellow	#FAFAD2	rosybrown	#BC8F8F
darkcyan	#008B8B	lightgreen	#90EE90	royalblue	#4169E1
darkgoldenrod	#B8860B	lightgrey	#D3D3D3	saddlebrown	#8B4513
darkgray	#A9A9A9	lightpink	#FFB6C1	salmon	#FA8072
darkgreen	#006400	lightsalmon	#FFA07A	sandybrown	#F4A460
darkkhaki	#BDB76B	lightseagreen	#20B2AA	seagreen	#2E8B57
darkmagenta	#8B008B	lightskyblue	#87CEFA	seashell	#FFF5EE
darkolivegreen	#556B2F	lightslategray	#778899	sienna	#A0522D
darkorange	#FF8C00	lightsteelblue	#B0C4DE	skyblue	#87CEEB
darkorchid	#9932CC	lightyellow	#FFFFE0	slateblue	#6A5ACD
darkred	#8B0000	limegreen	#32CD32	slategray	#708090
darksalmon	#E9967A	linen	#FAF0E6	snow	#FFFAFA
darkseagreen	#8FBC8F	magenta	#FF00FF	springgreen	#00FF7F
darkslateblue	#483D8B	mediumaquamarine	#66CDAA	steelblue	#4682B4
drakslategray	#2F4F4F	mediumblue	#0000CD	tan	#D2B48C
darkturquoise	#00CED1	mediumorchid	#BA55D3	thistle	#D8BFD8
darkviolet	#9400D3	mediumpurple	#9370DB	tomato	#FF6347
deeppink	#FF1493	mediumseagreen	#3CB371	turquoise	#40E0D0
deepskyblue	#00BFFF	mediumslateblue	#7B68EE	violet	#EE82EE
dimgray	#696969	mediumspringgreen	#00FA9A	wheat	#F5DEB3
dodgerblue	#1E90FF	mediumturquoise	#48D1CC	whitesmoke	#F5F5F5
firebrick	#B22222	mediumvioletred	#C71585	yellowgreen	#9ACD32
floralwhite	#FFFAF0	midnightblue	#191970		
forestgreen	#228B22	mintcream	#F5FFFA		

CSS
Cascading Style Sheet

Prof. Marcos Alexandruk

UNINOVE

SUMÁRIO

INTRODUÇÃO

Com o objetivo de tratar das características de design de maneira mais efetiva e melhorar o gerenciamento das páginas de um site foi criada uma extensão do HTML, padronizada pelo W3C (World Wide Web Consortium) em 1996, chamada CSS (Cascading Style Sheets) ou folhas de estilo: CSS1 (1996) e CSS2 (1998).

Algumas características de design:

- Espaçamento entre linhas;
- Posicionamento de imagens e textos;
- Cores e imagens de fundo;
- Tratamento de fontes;
- Margem e identação;
- Tratamento de características dos links.

Benefícios:

- Separar a definição do layout das páginas do conteúdo a ser exibido;
- Padronização das páginas de um site;
- Código HTML mais simples e fácil de ser manipulado;
- Otimização para qualquer resolução de monitor de vídeo;
- Controle mais preciso e previsível do formato da apresentação;
- Impressão melhor das páginas.

Sintaxe geral:

```
marcador {
   atributo1: valor1;
   atributo2: valor2;
   ...
   atributoN: valorN
}
```

Exemplos:

```
h1 { font-family: arial; color: yellow }
```
O texto dentro do marcador **h1** terá fonte Arial e cor Amarela.

```
p { font-size: 20pt; margin-left: 1.0in }
```
O texto dentro do marcador **p** terá fonte tamanho 20 pontos e margem esquerda de uma polegada.

```
a { text-decoration: none }
```
O texto dentro do marcador **a** não será sublinhado.

FORMAS DE ESTILO

Quatro tipos:

- Estilo INLINE;
- Estilo INCORPORADO;
- Estilo IMPORTADO;
- Estilo EXTERNO.

Nota: O estilo INLINE prevalece sobre todos os outros.

Estilo inline

Utilizado quando deseja-se modificar somente **um marcador específico** dentro da página.

Aplicado nos marcadores (tags) dentro de <body> ... </body>.

Sintaxe geral:

```
<marcador style="atributo1: valor1;
   atributo2: valor2;
   ...
   atributoN: valorN">
```

Estilo incorporado

Utilizado quando deseja-se especificar alguma característica para os **marcadores internos à página**.

A especificação de um estilo é válida para todos os marcadores dentro do arquivo HTML.

Aplicado dentro de <head> ... </head>.

Sintaxe geral:

```
<style type="text/css">
marcador1 {atributo1: valor1; atributo2: valor2; ... atributoN: valorN }
marcador2 {atributo1: valor1; atributo2: valor2; ... atributoN: valorN }
...
marcador3 {atributo1: valor1; atributo2: valor2; ... atributoN: valorN }
</style>
```

Estilo importado

Utilizado quando deseja-se especificar ou acrescentar alguma característica para os marcadores de uma ou mais páginas.

Deve-se criar um arquivo separado com a extensão .css que será importado à página na qual deseja-se aplicar o estilo.

Aplicado dentro de <head> ... </head>.

Sintaxe geral:

```
<style type="text/css">
@import url(nome_url/arquivo.css);
marcador1 {atributo1: valor1; atributo2: valor2; ... atributoN: valorN }
marcador2 {atributo1: valor1; atributo2: valor2; ... atributoN: valorN }
...
marcador3 {atributo1: valor1; atributo2: valor2; ... atributoN: valorN }
</style>
```

Estilo externo

Utilizado quando deseja-se especificar alguma característica para os marcadores que serão utilizados em mais de uma página e deseja-se que este seja o padrão para todas elas.

Qualquer mudança realizada no arquivo externo, que também recebe a extensão .css, é refletido em todas as páginas do site que utilizam este arquivo.

Aplicado dentro de <head> ... </head>.

Sintaxe geral:

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

EXERCÍCIO:

Elaborar quatro páginas utilizando os quatro tipos de estilo (inline, incorporado, importado e externo). Altere as características dos elementos elementos: <h1>, <p> e <a ...>. (Exemplos na página 33)

CLASSES DE ESTILO

Classes de estilo podem ser úteis quando deseja-se que um marcador assuma estilos diferentes ou para aplicar estilos iguais a marcadores diferentes.

Pode-se fazer uso de classes de estilo utilizando-se o atributo **class** nos marcadores.

Exemplo:

```
<style type="text/css">
h1.naoprioritario {
   color: blue;
   font-family: georgia; }
h1.prioritario {
   color: red;
   font-family: verdana;
   text-align: center;
   font-size: 30pt; }
.importante {
   color: black;
   font-size: 40pt;
   background-color: yellow; }
</style>

...

<h1 class="naoprioritario">h1 class="naoprioritario"</h1>
<h1 class="prioritario">h1 class="prioritario"</h1>
<h1 class="importante">h1 class="importante"</h1>
<h2 class="importante">h2 class="importante"</h2>
```

ESTILOS INDIVIDUAIS

Utilizados para criar estilos exclusivos, independentes de qualquer tag HTML, tal como o seletor de classes.

O acesso ao estilo individual é obtido utilizando-se o atributo **id** dentro dos marcadores.

Exemplo:

```
<style type="text/css">
#fundo {
   background-color: yellow; }
#individual {
   color: red; }
</style>

...

<h1 id="fundo">h1 id="fundo"</h1>
<h1 id="individual">id="individual"</h1>
```

Notas:

- Pode-se combinar um id a uma classe e/ou estilo inline dentro de uma tag HTML.
- Um nome de id não pode ser uma palavra JavaScript.
- A diferença entre os id e as classes está no seu uso efetivo. Os ids são normalmente utilizados para dar a cada elemento do código HTML um nome e uma identidade exclusivos. Por esse motivo, um id é usado apenas uma vez dentro do documento HTML, permitindo a sua manipulação com o JavaScript.

MARCADORES E

São utilizados quando deseja-se criar suas próprias tags HTML.

A diferença entre os dois é que com o marcador **div** ocorre uma quebra de linha anterior e posterior a ele e com o **span** não ocorre esta quebra de linha.

div

Sintaxe geral:

```
<div class="classeestilo">... texto ...</div>

ou

<div id="nomeestilo">... texto ...</div>
```

Exemplo:

```
<style type="text/css">
.importante {
   color: blue;
   background-color: yellow;
   text-decoration: underline; }
</style>

...

<p>Este parágrafo utiliza o marcador <div class="importante">div com
   classes de estilo.</div> Verifica-se que onde o div é utilizado
   ocorre uma quebra de linha no texto anterior e posterior a ele.</p>
```

span

Sintaxe geral:

```
<span class="classeestilo">... texto ...</span>

ou

<span id="nomeestilo">... texto ...</span>
```

Exemplo:

```
<style type="text/css">
.importante {
   color: blue;
   background-color: yellow;
   text-decoration: underline; }
</style>

...

<p>Este parágrafo utiliza o marcador <span class="importante">span com
   classes de estilo.</span> Verifica-se que onde o span é utilizado
   não ocorre quebra de linha no texto.</p>
```

HERANÇA DE ESTILOS

Exemplo:

```
<style type="text/css">
h1 {
   color: red;
   font-family: verdana; }
u {
   font-size: 30pt; }
</style>

...

<h1>Texto <u> cor <i> vermelha </i> e fonte tipo <i> verdana</i></u></h1>
```

Hierarquia dos marcadores

<h1>

<u>

<i>

ELEMENTOS EM CONTEXTO

Somente ocorre o estilo se o marcador estiver dentro do contexto definido.

Exemplo:

```
<style type="text/css">
h1 u i {
   color: blue;
   background-color: yellow;
   text-decoration: underline; }
</style>

...

<h1>Título <u> com <i> contexto </i></u></h1>
<h1>Título sem <i> contexto </i></h1>
```

Fontes

Os atributos disponíveis para formatação de fontes são:

font-family
Família de fontes disponíveis
Abadi, Lúcida Sans, Arial, Tahoma, Times New Roman, Courier New ...

font-size
Tamanho de fontes
pt, px, in, cm, mm, pc, em, ex, palavras-chave

font-weight
Caracteres normais ou negrito
normal, bold, lighter, bolder, 100, 200, 300, 400, 500, 600, 700, 800 e 900

font-style
Caracteres normais ou itálico
normal, italic, oblique (inclina o texto para direita)

font-variant
Caracteres normais ou maiúsculas
normal, small-caps

font
Define diversos atributos simutaneamente em uma regra de estilo
font: font-style font-variant font-weight font-size font-family

Exemplos:

```
<html>
<head>
<style type = "text/css">
p {
   font-family: courier new;
   font-size: 20px;
   font-weight: bold;
   font-style: italic;
   font-variant: small-caps; }
</style>
</head>
<body>
<p<Texto formatado</p>
</body>
</html>
```

```
<html>
<head>
<style type = "text/css">
p { font: italic small-caps bold 20px courier new }
</style>
</head>
<body>
<p>Texto formatado</p>
</body>
</html>
```

Texto e Cor

Alguns atributos para a formatação de texto e cor são:

color
Especifica a cor do texto
color: nomedacor ou color: #RRGGBB (ver página 30)

letter-spacing
Espaçamento entre letras
normal, pt, px, in, cm, mm, pc, em, ex, palavras-chave

line-height
Altura de uma linha
normal, (número a ser multiplicado pelo tamanho da fonte*), %, px
* neste caso 2 equivale a espaço duplo

text-transform
Letras maiúsculas e minúsculas
normal, capitalize, uppercase, lowercase

text-align
Alinhamento horizontal do texto
left, right, center, justify

vertical-align
Alinhamento vertical do texto
baseline, super, sub

text-indent
Recuo do texto para a primeira linha de um parágrafo
pt, px, in, %

text-decoration
Presença de linhas sobrescritas em um texto
underline, overline, line-through, none

Exemplo:

```
<html>
<head>
<style type = "text/css">
p { color: #FF0000;
    letter-spacing: 10px;
    line-height: 2;
    text-transform: capitalize;
    text-align: justify;
    vertical-align: baseline;
    text-indent: 20px;
    text-decoration: underline; }
</style>
</head>
<body>
<p>Os Sistemas Gerenciadores de Banco de Dados tornaram-se
   peça fundamental na implantação de sistemas.</p>
</body>
</html>
```

Listas

list-style-type

Tipo de símbolo a ser utilizado como marcador de item

disc, circle, square, decimal, upper-roman, lower-roman, upper-alpha, lower-alpha

list-style-position

Posicionamento dos itens da lista

inside (alinhamento interno ao item anterior), outside (alinhamento externo ao item anterior)

list-style-image

Imagem a ser exibida no lugar do símbolo utilizado em cada item da lista

url(arquivodeimagem.gif)

list-style

Junção dos três atributos (type, position e image)

{ list-style: url(arquivodeimagem.gif) simbolopadrao alinhamento; }

Exemplo:

```
<html>
<head>
<style type="text/css">
li.outside { list-style-position: outside; list-style-type: disc;
}
li.inside { list-style-position: inside; list-style-type: circle;
}
</style>
</head>
```

```
<body>
<ul>
<li class="outside">Item 1</li>
<li class="inside">Item 1.1</li>
<li class="inside">Item 1.2</li>
<li class="outside">Item 2</li>
<li class="inside">Item 2.1</li>
</ul>
</body>
</html>
```

Valores permitidos para list-style-type:

- disc
- circle
- square
- decimal
- upper-roman
- lower-roman
- upper-alpha
- lower-alpha

Mouse

Utilizado para modificar a aparência do cursor do mouse quando o mesmo está sobre algum elemento da página.

style= "cursor:valor"

default, crosshair, hand, pointer, help, move, n-resize, ne-resize, e-resize, se-resize, s-resize, sw-resize, w-resize, nw-resize, text, wait

Exemplo:

```
<html>
<table width="400" border="1">
   <tr>
      <td style="cursor:default">default</td>
      <td style="cursor:crosshair">crosshair</td>
      <td style="cursor:hand">hand</td>
      <td style="cursor:pointer">pointer</td>
   </tr>
   <tr>
      <td style="cursor:help">help</td>
      <td style="cursor:move">move</td>
      <td style="cursor:n-resize">n-resize</td>
      <td style="cursor:ne-resize">ne-resize</td>
   </tr>
   <tr>
      <td style="cursor:e-resize">e-resize</td>
      <td style="cursor:se-resize">se-resize</td>
      <td style="cursor:s-resize">>s-resize</td>
      <td style="cursor:sw-resize">sw-resize</td>
   </tr>
```

```
   <tr>
      <td style="cursor:w-resize">w-resize</td>
      <td style="cursor:nw-resize">nw-resize</td>>
      <td style="cursor:text">text</td>
      <td style="cursor:wait">wait</td>
   </tr>
</html>
```

Background

Os atributos possíveis para formatação de fundo são:

background-color
Definição da cor de fundo
background-color: nomedacor ou background-color: #RRGGBB (ver página 30)

background-image
Definição da imagem de fundo
background-image:url(arquivodeimagem.gif)

background-repeat
Define se haverá repetição da imagem de fundo ou não

repeat	Repete a imagem tanto na horizontal quanto na vertical
repeat-x	Repete a imagem na horizontal
repeat-y	Repete a imagem na vertical
no-repeat	Exibe a imagem somente uma vez

background-attachment
Define se ocorrerá rolagem da imagem em relação aos outros elementos HTML

scroll	Faz a rolagem do texto e da imagem
fixed	Fixa a imagem permitindo a rolagem somente do texto

background-position
Posiciona a imagem de fundo em algum ponto específico da tela
background-position: x y ou background-position: x% y%
(em relação ao elemento HTML considerado "pai" na hierarquia)

top	Posiciona a imagem no topo da página
center	Posiciona a imagem no centro da página
bottom	Posiciona a imagem no rodapé da página
left	Posiciona a imagem à esquerda da página
right	Posiciona a imagem à direita da página

background
Todos os atributos juntos
style= "background: background-color background-image background-repeat background-attachment background-position"

Exemplo:

```
<html>
<head>
<style type="text/css">
body { background-color: yellow;
       background-image: url(logo.jpg);
       background-repeat: no-repeat;
       background-attachment: fixed;
       background-position: 50 80; }
p { font-size: 25px;
    margin-left: 200px;
    line-height: 3; }
</style>
</head>
<body>
<p>Linha1</p>
<p>Linha2</p>
<p>Linha3</p>
<p>Linha4</p>
<p>Linha5</p>
<p>Linha6</p>
<p>Linha7</p>
<p>Linha8</p>
</body>
</html>
```

Bordas e Margens

Os atributos disponíveis para formatação das bordas e das margens são:

width e height
Definem a largura e altura dos elementos

margin
Permite definir o espaço de um elemento em relação ao outro
margin, margin-top, margin-bottom, margin-left, margin-right

border
Define o estilo, largura e cor da borda que envolve o elemento.
border, border-style, border-color, border width, border-top, border-bottom, border-left, border-right

padding
Define o espaço entre a borda do elemento e o seu conteúdo

Outros três atributos relacionados a bordas e margens são:

float
Permite que um elemento "flutue" sobre um outro elemento

clear
Evita que um elemento "flutue" sobre um outro elemento

display
Define como um elemento deve ser exibido (ou não)

Exemplo:

```
<html>
<head>
<style type="text/css">
.frame { width: 250px;
         margin-top: 150px;
         margin-left: 100px;
         border-top: 1mm dotted #990000;
         border-bottom: 3px dashed #009900;
         border-left: 3pt solid #000099;
         border-right: 20pt inset #990000; }
</style>
</head>
<body>
<div class="frame"><img width="100" height="100" src="logo.jpg"></div>
</body>
</html>
```

Posicionamento

O posicionamento dos elementos podem ser definidos através do atributo **position**. Este atributo pode assumir um dos três valores de posicionamento:

static
Faz o conteúdo "fluir in line", mas não pode ser alterado pelos atributos top e left ou por linguagem JavaScript.

relative
Posiciona o elemento de forma relativa ao elemento pai.

absolute
Posiciona o elemento de forma independente com relação a origem (0,0).

O posicionamento pode ser definido de duas formas:

1. top e left (superior e esquerda)
2. bottom e right (inferior e direita)

Exemplo:

```
<style type="text/css">
.stat { position: static; font: bold 30pt verdana; color: green; }
.abs { position: absolute; top: 25px; left: 380px; witdh: 100 px;
       font: bold 20pt arial; color: red; }
.rel { position: relative; top: 80px; left: 25px; font: bold 30pt
       georgia; color: blue; }
</style>
</head>
```

```
<body>
<div class="stat">Uma das principais vantagens da TV Digital</div>
<div class="abs">no país é a possibilidade do aumento da interatividade
entre o telespectador e a televisão.</div>
<div class="rel">Esse é um novo mercado de <span class="rel">software
</span>, uma grande oportunidade para desenvolvedores.</div>
</body>
</html>
```

O posicionamento 3D é realizado através do atributo z-index. Os elementos posicionados recebem automaticamente números de empilhamento: 0, 1, 2 e assim por diante, na ordem em que aparecem. O número do índice z é um valor que mostra sua relação 3D com outros elementos da janela.

```
<img style="position:absolute; top:50px; left:50px; z-index:0"
 src="azul.jpg">
<img style="position:absolute; top:100px; left:100px; z-index:2"
 src="vermelho.jpg">
<img style="position:absolute; top:150px; left:150px; z-index:1"
 src="verde.jpg">
```

Opções: position:absolute e position:relative

Notas:

- Quando um elemento posicionado relativamente é colocado dentro de um elemento posicionado absolutamente, ele se move com o elemento absolute.
- Quando um elemento posicionado absolutamente é alinhado dentro de outro elemento que tem posicionamento relativo, o elemento absoluto flui a partir da origem do canto superior esquerdo do elemento relativo, em vez de fluir a partir da posição (0,0) da origem da janela.
- A ordem dos elementos empilhados de forma 3D pode ser alterada utilizando-se uma linguagem Script.
- O uso de um número negativo para o índice z faz com que um elemento seja empilhado aquele número de níveis abaixo de seu elemento pai em vez de ser empilhado acima.

Visibilidade

Os atributos de visibilidade são:

- **visibility**: define a visibilidade de um elemento
- **clip**: define a área visível de um elemento
- **overflow**: descreve o tratamento da área que ultrapassa o limite da área de exibição

visibility

Sintaxe geral:

```
style="position: valor; visibility: valores"
```

valor: relative ou absolute

valores: hidden, visible ou inherit

NOTA: Se a visibilidade estiver definida como **hidden** o elemento ficará invisível, mas ainda ocupará espaço no documento e um retângulo vazio aparecerá no lugar do elemento. Utilizando a opção **display: none** o espaço vazio desaparece do documento.

clip

Sintaxe geral:

```
style="position: valor; clip: valores"
```

valor: relative ou absolute

valores: rect(<toplength>,<rightlength>,<bottomlength>,
<leftlength>) ou auto

overflow

Sintaxe geral:

```
style="width: valorunidade; height: valorunidade; overflow: valor"
```

valorunidade: valor e unidade correspondente para a largura e a altura da região a ser realizado o overflow

valor:

- scroll: Barras de rolagem ao redor da área visível
- hidden: Oculta o overflow e evita que a barra de rolagem apareça
- visible: A parte recortada é exibida
- auto: O browser resolve como o material extra será tratado

Exemplo:

```
<html>
<head>
</head>
<body>
<img style="position:absolute; visibility: visible; top:10px; left:20px;
 z-index:0" src="logo.jpg">
<img style="position:absolute; visibility: hidden; top:10px; left:250px;
 z-index:1" src="logo.jpg">
</table>
</body>
</html>
```

JAVA SCRIPT

Prof. Marcos Alexandruk

UNINOVE

SUMÁRIO

1. INTRODUÇÃO

- Colocando JavaScript em uma página web

```
<html>
<head>
<title>Exemplo 01</title>
<body>

<SCRIPT LANGUAGE="JavaScript">
<!--
   document.write('Este é o meu primeiro JavaScript');
//-->
</SCRIPT>

</body>
</html>
```

2. SINTAXE E CONVENÇÕES

- JavaScript é sensível a maiúsculas e minúsculas (case sensitive). Ex: **a** é diferente de **A**

- Todas as instruções em JavaScript devem terminar com **;** (ponto e vírgula)

- Aspas duplas (" ") e aspas simples (' '): As aspas simples são usadas dentro de uma instrução com aspas duplas ou vice-versa. Exemplo:

```
<FORM>
<INPUT TYPE="Button" VALUE="Clique aqui" onClick="alert('Até breve');">
</FORM>
```

- Barra invertida **** e linhas de texto (strings)

\b backspace
\f form feed
\n inserir linha
\r retorno (sem inserção de linha)
\t tabulação
\' aspas simples (apóstrofo)
\" aspas duplas

Exemplos:

```
<SCRIPT LANGUAGE="JavaScript">
<!--
   alert('Fulano de Tal\nGerente Comercial');
//-->
</SCRIPT>

<SCRIPT LANGUAGE="JavaScript">
<!--
   alert('She said \"I don\'t like you\"');
//-->
</SCRIPT>
```

- Comentários

Comentários são úteis para mostrar informações sobre direitos autorais, datas de atualizações, funções, etc. Ajudam a entender o código mais tarde.

Para comentar uma linha:

```
<SCRIPT LANGUAGE="JavaScript">
<!--
   // A linha abaixo escreve o texto na página
   document.write('Este código está comentado');
//-->
</SCRIPT>
```

Para comentar mais de uma linha:

```
<SCRIPT LANGUAGE="JavaScript">
<!--
   /* A linha abaixo escreve o texto na página
   e eu estou usando mais de uma linha no meu
   comentário */
   document.write('Este código está comentado');
//-->
</SCRIPT>
```

- Nomes de variáveis e de funções: Devem seguir algumas regras simples, conforme mostrado a seguir:

  1. O primeiro caractere deve ser uma letra do alfabeto (maiúscula ou minúscula), uma barra _ (underscore) ou um sinal de dólar $. O sinal de dólar não é recomendado, já que algumas versões mais antigas do JavaScript não o reconhecem.
  2. Você não deve usar um número para iniciar o nome de uma variável ou função.
  3. Nomes de variáveis ou de funções não devem conter espaço. No caso de precisar de um espaço para identificar melhor uma variável ou uma função, utilize _ (underscore).

- Palavras reservadas: O JavaScript possui uma série de palavras que são usadas internamente pela linguagem e pelo interpretador. Estas não podem ser usadas como nomes de variáveis ou funções.

3. VARIÁVEIS

Variáveis são nomes dados aos locais na memória do computador onde os dados são armazenados durante a execução de nossos scripts. Quando declaramos uma variável, ela fica guardada na memória até que decidamos fechar a janela em que o script está sendo executado.

Variáveis são declaradas usando a palavra-chave **var**. Observe o exemplo:

```
<SCRIPT LANGUAGE="JavaScript">
<!--
   var Texto = 'Esta é a minha primeira variável';
   document.write(Texto);
//-->
</SCRIPT>
```

- Valores de variáveis alterados em tempo de execução:

```
<SCRIPT LANGUAGE="JavaScript">
<!--
   var Frase = 'O valor da variável será alterado em seguida';
   alert(Frase);
   Frase = 'O valor da variável Frase foi alterado aqui';
   document.write(Frase);
//-->
</SCRIPT>
```

- Cálculo matemático usando variáveis:

```
<form name="form1">
Primeiro número:
<input type="text" name="text1"><br>
Segundo número:
<input type="text" name="text2"><br>
Clique aqui para calcular:
<input type="button" value="Calcular" onClick="Calcular()">
</form>

<SCRIPT LANGUAGE="JavaScript">
<!--
   function Calcular() {
     var primeiroNum = eval(document.form1.text1.value);
     var segundoNum = eval(document.form1.text1.value);
     alert(primeiroNum+segundoNum);
   }
//-->
</SCRIPT>
```

- Armazenando dados do usuário usando variáveis:

```
<SCRIPT LANGUAGE="JavaScript">
<!--
   var Nome;
   Nome = prompt('Por favor, digite o seu nome', 'Digite o seu nome aqui');
   document.write('Olá '+Nome+'! Bem vindo ao meu site.');
//-->
</SCRIPT>
```

4. OPERADORES

- Operadores Aritiméticos

Operador	Usado para	Exemplo	Resultado
+	Adição	1+2	3
-	Subtração	12-10	2
*	Multiplicação	2*3	6
/	Divisão	10/3	3.333333
%	Módulo	10%3	1
++	Incremento	x=5 x++	x=6
--	Decremento	x=5 x--	x=4
-	Número negativo	-20	20 negativo

```
<SCRIPT LANGUAGE="JavaScript">
<!--
   var Numero = 10;
   Numero++;
   document.write(Numero);
//-->
</SCRIPT>

<SCRIPT LANGUAGE="JavaScript">
<!--
   var Numero1 = 10;
   var Numero2 = 20;
   document.write(Numero1+Numero2);
//-->
</SCRIPT>
```

- Operadores de Atribuição

Operador	Exemplo	Significado
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y
%=	x%=y	x=x%y

```
<SCRIPT LANGUAGE="JavaScript">
<!--
   var Numero1 = 10;
   var Numero2 = 20;
   Numero2 += Numero1;
   documento.write(Numero2);
//-->
</SCRIPT>
```

- Operadores de Comparação

Operador	Significado	Exemplo
==	É igual a	15==13 é **false**
!=	É diferente	15!=13 é **true**
>	É maior que	15>13 é **true**
>=	É maior ou igual a	15>=13 é **true**
<	É menor que	15<13 é **false**
<=	É menor ou igual a	15<=13 é **false**

```
<SCRIPT LANGUAGE="JavaScript">
<!--
   var Numero = 12;
   if(Numero==10) {
      alert('A variável Numero é igual a 10');
   }
//-->
</SCRIPT>
```

- Operadores Lógicos

Operador	Significado	Se x=10 e y=5 então ... x
&&	E	(x=10) && (y<10) é true
||	OU	(x=10) || (y=10) é true
!	NÃO	x!=y é true

```
<SCRIPT LANGUAGE= "JavaScript">
<!--
   var Numero1 = 20;
   var Numero2 = 30;
   if ((Numero1==20)&&(Numero2==30)){
      alert('Os valores estão corretos');
   }
//-->
</SCRIPT>
```

5. ESTRUTURAS CONDICIONAIS

IF

```
<html>
<SCRIPT LANGUAGE="JavaScript">
<!--
function Checar() {
  var Cor = document.form1.text1.value;
  if (Cor == 'Verde') {
    alert('Você escolheu VERDE');
  }
  if (Cor == 'Amarelo') {
    alert('Você escolheu AMARELO');
  }
  if (Cor == 'Azul') {
    alert('Você escolheu AZUL');
  }
}
//-->
</SCRIPT>

<form name="form1">
Escolha uma cor (Verde, Amarelo, Azul)<br><br>
<input type="text" name="text1">
<input type="button" value="OK" onClick="Checar()">
</form>
```

IF ... ELSE

```
<html>
<SCRIPT LANGUAGE="JavaScript">
<!--
function Checar() {
  var Cor = document.form1.text1.value;
  var Voltar = true;
  if (Cor == 'Verde') {
    alert('Você escolheu VERDE');
    Voltar = false;
  }
  if (Cor == 'Amarelo') {
    alert('Você escolheu AMARELO');
    Voltar = false;
  }
  if (Cor == 'Azul') {
    alert('Você escolheu AZUL');
    Voltar = false;
  }
  else {
    if (Voltar == true) {
    alert ('Você precisa digitar uma opção válida');
    }
  }
}
//-->
</SCRIPT>

<form name="form1">
Escolha uma cor (Verde, Amarelo, Azul)<br><br>
<input type="text" name="text1">
<input type="button" value="OK" onClick="Checar()">
</form>
```

6. COMANDOS DE REPETIÇÃO

WHILE

```
<html>
<head>
</head>
<body>

<SCRIPT LANGUAGE="JavaScript">

i=1;
while (i<10) {
  document.write("Esta é a linha ", i,"<br>");
i++;
}

</SCRIPT>
</body>
</html>
```

DO ... WHILE

```
<SCRIPT LANGUAGE="JavaScript">

i=1;
do {
  document.write("Esta é a linha ", i,"<br>");
  i++;
}
while (i<10);

</SCRIPT>
```

FOR

```
<SCRIPT LANGUAGE="JavaScript">

for (i=1; i<10; i++) {
  document.write("Esta é a linha " ,i, "<br>");
}

</SCRIPT>
```

FOR (2)

```
<SCRIPT LANGUAGE="JavaScript">

for (i=1; i<=3; i++) {
  for (j=1; j<=4; j++)
  document.write("Esta é a linha " ,i, "." ,j, "<br>");
}

</SCRIPT>
```

Ex1: Quadrado dos números 1 a 100

```
<SCRIPT LANGUAGE="JavaScript">

i=1;
while (i<=100) {
  document.write("O quadrado de ", i, " é igual a " ,i*i, "<br>");
i++;
}

</SCRIPT>
```

Ex2: Tabuada

```
<SCRIPT LANGUAGE="JavaScript">

n=2;
i=1;
while (i<=10) {
  document.write(n, " multiplicado por ", i, " é igual a " ,n*i, "<br>");
i++;
}

</SCRIPT>
```

Ex3: Soma 20 primeiros números (apresenta somente o resultado)

```
<SCRIPT LANGUAGE="JavaScript">

i=0;
j=1;
while (j<=20) {
i=i+j;
j++;
}
document.write(i);

</SCRIPT>
```

Ex3: Soma 20 primeiros números (apresenta passo-a-passo e resultado)

```
<SCRIPT LANGUAGE="JavaScript">

i=0;
j=1;
while (j<=20) {
document.write(i=i+j, "<br>");
j++;
}

</SCRIPT>
```

Ex3: Soma 20 primeiros números (apresenta passo-a-passo e resultado)

```
<SCRIPT LANGUAGE="JavaScript">

i=0;
for (j=1; j<=20; j++){
document.write(i=i+j, "<br>");
}

</SCRIPT>
```

7. ARRAYS (MATRIZES)

Um array é um conjunto de variáveis do mesmo tipo que é referenciado por um nome comum. Um elemento específico de um array é acessado por meio de um índice.

Temos dois tipos de índices:

- Ordenado: baseado em números (começa com ZERO)
- Associativos: formados por caracteres alfa-númericos

Ex1:

```
<SCRIPT LANGUAGE="JavaScript">
var Nomes = new Array();
  Nomes[0] = "Augusto";
  Nomes[1] = "Roberto";
  Nomes[2] = "Fernando";
  Nomes[3] = "Márcio";
  Nomes[4] = "Carlos";
  Nomes[5] = "Antonio";

window.alert("O funcionário do dia é: "+Nomes[3]);
</SCRIPT>
```

Ex2:

```
<html>
<head>
<SCRIPT language="JavaScript">
function mostrar()
{
  var nome= new Array(5)
    nome[0]="Um";
    nome[1]="Dois";
    nome[2]="Três";
    nome[3]="Quatro";
    nome[4]="Cinco";
  var x=0;
  for (x=0; x<5; x++)
  {
    alert(nome[x]);
  }
}
</SCRIPT>
</head>
<body>
<a href="javascript:mostrar()">Mostrar elementos do Array!</a>
</body>
</html>
```

Ex3:

```
<html>
<head>
<SCRIPT language="JavaScript">
function mostrar2()
{
  var nome= new Array(5);
  var y=0;
  for (y=0; y<5; y++)
  {
    nome[y]=prompt('Entre com um nome!',' ');
```

```
  }
  var x=0;
  for (x=0; x<5; x++)
  {
    alert(nome[x]);
  }
}
</SCRIPT>
</head>
<body>
<A HREF="javascript:mostrar2()">Entre com os elementos do Array</A>
</body>
</html>
```

Ex4: Array Associativo

```
<html>
<head>
</head>
<body>
<SCRIPT language="JavaScript">
function escolher_carro()
{
var meu_carro= new Array()
  meu_carro["esporte"]="Mustang";
  meu_carro["compacto"]="Ka";
  meu_carro["antigo"]="Brasília";
var carro_tipo=prompt("Qual o tipo de carro que você prefere?","");
if ((carro_tipo=="esporte") || (carro_tipo=="compacto") || (carro_tipo=="antigo"))
  alert("Para esta opção oferecemos: "+meu_carro[carro_tipo]+".");
else
  alert("Entre com um tipo válido: esporte, compacto ou antigo");
}
</SCRIPT>
<FORM>
<INPUT TYPE="button" onClick="escolher_carro()" value="Escolha seu carro!">
</FORM>
</body>
</html>
```

8. FUNÇÕES

Uma função é um grupo de linhas de código de programação identificado por um nome, por meio do qual pode ser referenciado em qualquer parte do programa principal, destinado uma tarefa bem específica e que podemos se necessário, utilizar várias vezes.

Ex1: Chamando a função ao carregar a página

```
<html>
<head>
<SCRIPT LANGUAGE="Javascript">
function Mensagem() {
document.write("Bem vindo a minha página");
}
</SCRIPT>
</head>
<body onLoad="Mensagem()">
</body>
</html>
```

Ex2: Chamando a função através de um botão

```
<html>
<head>
<title>Função 1</title>
<SCRIPT LANGUAGE="JavaScript">
function Mensagem(){
  alert('Mensagem chamada através de uma função');
}
</SCRIPT>
</head>
<body>
<form name="teste">
<input type="button" value="Exibir Mensagem" onClick="Mensagem()">
</form>
</body>
</html>
```

Ex3: Chamando a função através de um link

```
<html>
<head>
<title>Função 1</title>
<SCRIPT LANGUAGE="JavaScript">
function Mensagem(){
  alert('Mensagem chamada através de uma função');
}
</SCRIPT>
<head>
<body>
<a href="JavaScript:Mensagem()">Exibir Mensagem</a>
</body>
</html>
```

Ex4: Função utilizando um array

```
<html>
<head>
<title>Função 1</title>
<SCRIPT LANGUAGE="JavaScript">
matFigura = new Array(2)
matFigura[0] = "foto1.gif"
matFigura[1] = "foto2.gif"
numFigura = 0
totalFigura = 2
function mudaBanner(){
  document.imgBanner.src = matFigura[numFigura]
numFigura +=1
if (numFigura == totalFigura) {numFigura = 0}
setTimeout("mudaBanner()", 2000)
}
</SCRIPT>
</head>
<body onLoad="mudaBanner()">
<center>
<img name="imgBanner"><br><br>
</center>
</body>
</html>
```

Ex5: Função utilizando o comando de repetição for e array

```
<SCRIPT LANGUAGE="JavaScript">
function semana(hoje)
{
   document.write("<ul>")
   for (i=0;i<7;i++)
   {
      document.write("<li>" + hoje[i])
   }
   document.write("</ul>")
}
</SCRIPT>

<SCRIPT LANGUAGE="JavaScript">
dias = new Array("Domingo","Segunda","Terça","Quarta","Quinta","Sexta","Sábado")
semana(dias)
</SCRIPT>
```

Ex6: Função recursiva (recursivas: funções que chamam a elas mesmas)

```
<SCRIPT LANGUAGE="JavaScript">
function fatorial(n)
{
  if (n<0) {
    return -1;
  }
  if (n<=1) {
    return 1;
  }
  return n*fatorial(n-1);
}
document.write("O fatorial de 3 é: "+fatorial(3));
</SCRIPT>
```