
P R O G R A M A Ç Ã O

HTML

RESUMO DAS AULAS

Prof. Celso Gallão



RESUMO DAS AULAS DE HTML

Linguagem Gráfica de Programação I

PARTE 01 - Introdução	1
1 - Página Web	1
2 - Estrutura de uma Página HTML	1
3 - Primeiros Comandos	1
4 - Parágrafos	2
5 - Estilos de Textos	2
PARTE 02 - Listas	3
6 - Listas	3
PARTE 03 - Link e URL	4
7 - Documentos com Hipertextos	4
PARTE 04 - Melhorando a Aparência	5
8 - Cores	5
9 - Fundo com Imagem (Background)	5
10 - Alinhando Parágrafos	5
11 - Definindo Tamanhos	5
12 - Configurando as Fontes	5
13 - Texto em Movimento	6
PARTE 05 - Âncoras	6
14 - Âncoras	6
PARTE 06 - Tabelas	7
15 - Utilizando Tabelas	7
PARTE 07 - Frames	8
16 - Utilizando Frames	8
PARTE 08 - Formulários	9
17 - Utilizando Formulários	9
PARTE 09 - O Comando META	11
18 - Introdução	11
18.1 - Informações Específicas	11
18.2 - Mecanismos de Busca da Internet	11
PARTE 10 - Folhas de Estilo	12
19 - Introdução	12
19.1 - Regras de Utilização	12
19.2 - Aplicação da Folha de Estilo	12
19.2.1 - O Método LINKING	12
19.2.2 - O Método EMBEDDING	12
19.2.3 - O Método INLINE	13
19.3 - Unidades de Medida	13
19.4 - Definindo Estilo de Cores	13
19.5 - Fazendo Referência a URL	13
19.6 - Tabela de Seletores	13

Hiper Text Markup Language

Este material é apenas um resumo para acompanhamento dinâmico das aulas ministradas, sendo necessário o acompanhamento, e as orientações, do professor..

PARTE 01 - Introdução

1 - Página Web

Também conhecida como *Home Page* ou *Site*, é um documento composto basicamente de textos e códigos especiais chamados **TAGS (etiquetas)** que possibilitam a exibição do documento na *World Wide Web*. Pode conter também imagens, sons, animações, vídeo e links (ligações para outras *Páginas Web*). A *World Wide Web* (também conhecida como *Web*, *WWW* e *W₃*) é apenas um conjunto de documentos multimídia conectados através de *hyperlinks* que permitem o deslocamento entre documentos.

1.1 - Acesso às Páginas Web

O acesso é feito pelos **NAVEGADORES**, um tipo de **aplicativo** originalmente chamado de **BROWSER**. Existem vários **NAVEGADORES** no mercado, os principais são o *Netscape* e o *Internet Explorer*. O **BROWSER** deve estar instalado em seu micro, pois é ele quem vai identificar os comandos **HTML** para executá-los e exibir *Páginas Web*.

1.2 - Edição de uma Página Web

Vamos nos utilizar aqui da linguagem **HTML** para a edição de uma *Página Web*. O programador deve digitar o **CÓDIGO-FONTE** de sua *Página Web*, em um editor de textos, seguindo as regras de sintaxe da linguagem **HTML**. Depois de digitados, os programas devem ser gravados com a extensão **.HTM** ou **.HTML**, para que sejam lidos pelo Browser.

1.3 - Utilizando um Navegador (Browser)

Não é preciso estar conectado à internet para se utilizar um Browser, portanto, é muito simples testar as *Páginas Web* que estiverem sendo editadas; bastando para isso executar o Browser em seu micro e carregar o programa editado. Às vezes acontece de um comando ser reconhecido por um determinado Browser e não ser por outro. Para que isso seja testado, é interessante que se tenha instalado em seu micro pelo menos os dois principais Navegadores do mercado, e testar sua *Página Web* nos dois. Em caso de incompatibilidade, o melhor é escrever os comandos duas vezes (uma vez com a sintaxe de um navegador e a outra vez com a sintaxe do outro navegador).

2 - Estrutura de uma Página HTML

O código-fonte de uma *Página HTML* pode ser digitado com caracteres maiúsculos ou minúsculos, pois não há diferenciação entre eles. Os TAGs podem ser digitados na mesma linha ou em linhas subsequentes. O programador deve sempre estar atento com abertura e fechamento de TAGs e Aspas. A Linguagem de Programação HTML possui estrutura linear e não compilada, portanto não haverá lista de erros.

2.1 - Sintaxe dos Tags

Podem ser especificadas em pares ou individuais e estão sempre envolvidas pelos sinais **< >** ou **</>**:

<tag> texto </tag>

Exemplo:

** texto em negrito **

2.2 - Estrutura Básica da Página Web

Nem todos os TAGs são obrigatórios, mas basicamente a estrutura se dá desta forma:

```
<HTML>
<HEAD>
    <TITLE>Título da Página</TITLE>
</HEAD>
<BODY>
    <!--Comandos HTML-->
</BODY>
</HTML>
```

3 - Primeiros Comandos

a) **<HTML> ... </HTML>**

Especifica que dentro desses TAGs há comandos HTML.

b) **<HEAD> ... </HEAD>**

Área destinada aos TAGs relacionados com a identificação da página. Sua utilização não é obrigatória, entretanto, há alguns TAGs que necessitam ser declarados nesta área, como o **<TITLE>** por exemplo.

c) **<TITLE> ... </TITLE>**

Especifica o título da página, que aparecerá no cabeçalho do Navegador. A digitação incorreta deste TAG acarretará problemas na exibição da página.

- d) **Textos em Cabeçalhos:** É um tipo de formatação de texto que pode ser utilizada em qualquer lugar da página, com 6 tamanhos de fonte.

<H1> Texto em Cabeçalho Tamanho 1 - Maior **</H1>**
até

<H6> Texto em Cabeçalho Tamanho 6 - Menor **</H6>**

O maior é o **<H1>** e o menor é o **<H6>**. Automaticamente será adicionada uma linha em branco antes e outra linha depois. Possui o seguinte parâmetro opcional:

☐ **ALIGN** = alinhamento (LEFT, CENTER, RIGHT)

Exemplo:

```
<H1 ALIGN = "right">Subtítulo 1</H1>
```

- e) **<HR>**

Traça uma linha horizontal na tela. Possui os seguintes parâmetros opcionais:

☐ **SIZE** = espessura da linha

☐ **WIDTH** = comprimento da linha

☐ **ALIGN** = alinhamento (LEFT, CENTER, RIGHT)

☐ **COLOR** = cor da linha.

Exemplo:

```
<HR SIZE = "2" WIDTH = "50" ALIGN = "left"
COLOR = "red">
```

- f) **<!-- comentários -->**

Permite digitar comentários em qualquer parte do programa. Esses comentários não serão exibidos na página.

Exemplo:

```
<!-- Elaborado pelo Prof. Celso Gallão -->
```

4 - Parágrafos

Uma Página Web não reconhece o fim de um parágrafo. Mesmo que pressione a tecla **[ENTER]** na edição em HTML, o Navegador não reconhecerá a quebra de linha ou de parágrafo. É necessário utilizar TAGs específicas.

- a) **<P>**

Ao ser utilizado sozinho, especifica a quebra de parágrafo e insere uma linha em branco após.

Exemplo:

```
O HTML não reconhece a tecla ENTER como
sendo o final de uma linha.<P>
```

<P> ... </P>

Pode ser utilizado com o seu par, marcando o início e o fim de um parágrafo, neste caso possui o seguinte parâmetro opcional:

☐ **ALIGN** = alinhamento (LEFT, CENTER, RIGHT)

Exemplo:

```
<P ALIGN="right">O HTML não reconhece a
tecla ENTER como sendo o final de uma
linha.</P>
```

- b) **
**

Especifica a quebra de linha sem inserir uma linha em branco após.

- c) **<PRE> ... </PRE>**

Permite que um texto seja reconhecido pelo Navegador com sua formatação original, reconhecendo a tecla TAB para tabulação e ENTER para final de linha.

Exemplo:

```
<PRE>Agora sim, o HTML irá reconhecer a
tecla ENTER como sendo o final de uma
linha.</PRE>
```

- d) **<DIV> ... </DIV>**

Marca o início e o fim de um bloco de texto, sem adicionar uma linha em branco após. Possui o seguinte parâmetro:

☐ **ALIGN** = alinhamento (LEFT, CENTER, RIGHT)

Exemplo:

```
<DIV ALIGN="right">O HTML não reconhece a
tecla ENTER como sendo o final de uma
linha.</DIV>
```

5 - Estilos de Textos

Os TAGs que alteram os estilos de textos devem ser especificados aos pares.

a) **** texto em negrito ****

b) **<I>** texto em itálico **</I>**

c) **<U>** texto sublinhado **</U>**

d) **<TT>** texto em typewriter **</TT>**

e) **<BIG>** aumenta a fonte + negrito **</BIG>**

f) **<SMALL>** reduz a fonte **</SMALL>**

g) **^{** texto sobrescrito **}**

h) **_{** texto subscripto **}**

Editar o EXERCÍCIO 01 do Caderno de Exercícios

PARTE 02 - Listas

6 - Listas

As listas facilitam a criação de itens seriais. Podem ser **Ordenadas** (com itens numéricos) ou **Não Ordenadas** (com símbolos). Ao definir sub-listas teremos então **Listas Aninhadas**, podendo ser **Mistas ou não**, conforme os exemplos abaixo:

6.1 - Listas Não Ordenadas

```
<ul>
  <li> Curso Técnico em Turismo
  <li> Curso Técnico em Informática
</ul>
```

Será exibido:

- Curso Técnico em Turismo
- Curso Técnico em Informática

6.2 - Listas Ordenadas

```
<ol>
  <li> Curso Técnico em Turismo
  <li> Curso Técnico em Informática
</ol>
```

Será exibido:

1. Curso Técnico em Turismo
2. Curso Técnico em Informática

6.3 - Listas Aninhadas Mistas

```
<ol>
  <li> Curso Técnico em Turismo
  <li> Curso Técnico em Informática
    <ol>
      <li> TCG
      <li> LGP
        <ul>
          <li> Linguagem HTML
          <li> Linguagem Javascript
        </ul>
      <li> TP
    </ol>
  <li> Curso Técnico em Eletrônica
</ol>
```

Será exibido:

1. Curso Técnico em Turismo
2. Curso Técnico em Informática
 1. TCG
 2. LGP
 - Linguagem HTML
 - Linguagem Javascript
 3. TP
3. Curso Técnico em Eletrônica

Também podemos definir o tipo de estilo utilizado pela lista como números ou até mesmo utilizar letras (A, B, C) para isso utilizamos mais um atributo que é o "type" que deve ser incluído dentro da etiqueta sendo que o mesmo ficaria assim:

```
<ol type="a">
```

```
<ul type="a">
  <li> Internet</li>
  <li> Computador</li>
  <li> Site</li>
</ul>
```

Será exibido:

- a. Internet
- b. Computador
- c. Site

Agora iremos aprender a criar lista de definição o processo é semelhante ao que acabamos de aprender, portanto, para criar a lista utilizamos as etiquetas <dl></dl> sendo que utilizaremos duas etiquetas para definição <dt></dt> e <dd></dd>, então ficaria assim:

```
<dl>
  <dt>Html</dt>
  <dd>Html Básico</dd>
  <dt>Aprender</dt>
</dl>
```

Será exibido:

Html
 Html Básico
Aprender

[Editar o EXERCÍCIO 02 do Caderno de Exercícios](#)

[Editar o EXERCÍCIO 03 do Caderno de Exercícios](#)

PARTE 03 - Link e URL

7 - Documentos com Hipertextos

São documentos que se vinculam à outros documentos. São os chamados **LINKS**, ou seja, ligações de uma Página Web com outra que esteja no próprio micro ou em qualquer provedor existente no planeta. Para acessar uma Página Web em outro micro é necessário estar conectado com uma rede, como estamos falando de acesso em nível mundial, é necessário que esteja conectado à internet.

7.1 - Link Para Página Local

Pode ser texto ou figura que, ao ser clicada, fará ligação com um documento que for mencionado, bastando para isso saber o endereço completo do documento. O TAG usada é **<A HREF>**:

** hipertexto **

Exemplo:

** Página Principal **

No momento em que este Link for clicado, será exibido o documento chamado **index.htm**, que deverá estar no mesmo diretório da página Web que o acessou, caso contrário deverá ser indicado o *path (caminho)* completo do documento mas com as barras invertidas "\". Exemplos:

**Voltar à Página Principal **

**Voltar à Página Principal **

De modo geral, o nome **index.htm** é utilizado como nome da página inicial de um Site.

7.2 - Link Para Página Externa

Pode ser texto ou figura que, ao ser clicada, fará ligação com um documento que for mencionado. Para isso é necessário estar conectado à internet e conhecer o endereço virtual da Página Web. Este esquema de endereçamento virtual é chamado de **URL**. O TAG usado é **<A HREF>**.

** hipertexto **

Exemplo:

** Portal Singular**

O texto **Portal Singular** é chamado de Hipertexto ou simplesmente Link.

7.3 - URL (Uniform Resource Locator)

É composto por três partes principais:

- Protocolo Internet do documento:** é o tipo de servidor que está sendo acessado. Podem ser:
 - **http:** servidor que contém documentos no formato *HiperText Transfer Protocol*.
 - **gopher:** servidor composto por menus e diretórios c/ informações sobre arquivos e dados;
 - **ftp:** servidor que contém documentos no formato *File Transfer Protocol*.
 - **telnet:** inicia uma seção para se conectar remotamente a outro micro;
 - **waits:** servidor que contém documentos no formato *Wide Area Indexed Server*.

b) **Endereço do servidor:** é o computador que contém a Página Web.

c) **Endereço da Página Web:** é o *path* do arquivo acessado, dentro do servidor.

protocolo://servidor/página

Exemplo de URL:

http://www.uol.com.br/index.htm

7.4 - Exibição de Imagens

Para exibir uma imagem em sua Página Web, é necessário que ela esteja em um dos formatos aceitos pelo Browser:

- **GIF (Graphics Interchange Format):** É o formato padrão, o mais aceito pela Web;
- **JPEG (Joint Photographic Engineering Group):** Está sendo o formato mais utilizado pela sua maior capacidade de compactação;
- **BMP (Bit Map):** Deve ser evitado pela sua baixa capacidade de compactação.

O TAG utilizado é **** juntamente com o parâmetro **SRC="nome"**, conforme abaixo:

Exemplo:

Uma opção de alinhamento pode ser utilizada, para exibir a imagem ao lado de um texto. O parâmetro usado é o **ALIGN**.

Exemplos:

(texto no alto) ****

(texto ao centro) ****

(texto em baixo) ****

7.5 - Imagens Utilizadas Como Links

Para exibir uma imagem usando-a como um Link, basta utilizar o TAG **<A HREF>** em conjunto com o TAG ****.

Exemplo:

** **

(qdo. o usuário clicar na foto será exibida a página01)

7.6 - Link Para Enviar E-Mail

Para que o usuário envie e-mail apenas clicando em um hipertexto, o TAG **<A HREF>** também deve ser utilizado, mas especificamos no endereço o parâmetro **MAILTO:**

Exemplo:

hipertexto

(qdo. o usuário clicar no hipertexto será exibido o aplicativo para envio de e-mail que estiver instalado em seu micro)

Editar o EXERCÍCIO 04 do Caderno de Exercícios

Editar o EXERCÍCIO 05 do Caderno de Exercícios

PARTE 04 - Melhorando a Aparência

8 - Cores

Os navegadores possibilitam até 256 tons para cada uma das cores padrão *RGB*. Portanto, cada cor é representada por 3 dígitos, de 000 a 255. O HTML só reconhecerá esses dígitos se estiverem convertidos na base numérica *Hexadecimal*, ou seja, de 00 a FF. Também podemos declarar as cores especificando seus nomes, em inglês (tabela *RGB*). Os comandos abaixo são parâmetros do TAG <BODY>, portanto devem ser declarados dentro dele:

8.1 - Cor do Fundo

O parâmetro usado é **BGCOLOR**. O padrão é branco:

Exemplos:

<body bgcolor="#FFFFFF"> ou <body bgcolor="white">

8.2 - Cor do Texto

O parâmetro usado é **TEXT**. O padrão é preto:

Exemplos:

<body text="#000000"> ou <body text="black">

8.3 - Cor do Link (Hipertexto)

O parâmetro usado é **LINK**. O padrão é azul:

Exemplos:

<body link="#0000FF"> ou <body link="blue">

8.4 - Cor do Link Visitado

O parâmetro usado é **VLINK**. O padrão é marrom:

Exemplos:

<body vlink="#FF0000"> ou <body vlink="yellow">

8.5 - Cor do Link em Acesso

O parâmetro usado é **ALINK**. O padrão é azul:

Exemplos:

<body alink="#00FF00"> ou <body alink="green">

Exemplo utilizando todos os parâmetros juntos:

<body bgcolor="#000000" text="#EEEEBB"
link="#33CCFF" vlink="#CC0000" alink="#FFFFFF">

9 - Fundo com Imagem (Background)

Você pode carregar em sua página Web uma imagem de fundo, ou mesmo repetir várias vezes uma pequena imagem (.gif ou .jpg). O parâmetro é **BACKGROUND**, e também pertence ao TAG <BODY>.

Exemplo:

<body background="foto01.jpg">

ATENÇÃO: Quando for utilizar **BACKGROUND** você não deve declarar o parâmetro **BGCOLOR**, e vice-versa, por motivos óbvios!

10 - Alinhando Parágrafos

O padrão de alinhamento dos parágrafos editados em *HTML* é à esquerda, mas podemos modificá-los utilizando alguns parâmetros:

10.1 - O parâmetro ALIGN

É utilizado nos TAGs <p>, <div> ou <h1> até <h6>:

ALIGN = "right" (alinhamento à direita)

ALIGN = "center" (alinhamento ao centro)

ALIGN = "left" (alinhamento à esquerda - padrão)

Exemplos:

a) <p align = "center">

Alinha o parágrafo ao centro, até que outro comando <p> o modifique ou </p> seja encontrado.

b) <div align = "right">

Alinha o parágrafo à direita, até que o comando </div> seja encontrado.

c) <h1 align = "right">

Alinha o cabeçalho à direita, até que o comando </h1> seja encontrado.

d) <h6 align = "center">

Alinha o cabeçalho ao centro, até que o comando </h6> seja encontrado.

10.2 - O TAG <CENTER>

Além do parâmetro **ALIGN**, podemos centralizar textos, imagens e links com o TAG <CENTER>.

Exemplo:

<center>

Alinha os elementos ao centro, até que o comando </center> seja encontrado.

11 - Definindo Tamanhos

Em alguns elementos é possível configurar o tamanho e o alinhamento de sua exibição, com os parâmetros **SIZE**, **WIDTH** e **ALIGN**:

Exemplos:

a) <hr size=10 width=400 align=left>

exibe uma linha na tela com 10 pixels de espessura, 400 pixels de comprimento, alinhada à esquerda.

b) <hr size=20 width=50% align=right>

exibe uma linha na tela com 20 pixels de espessura, 50% do comprimento da tela exibida, alinhada à direita.

c)

exibe uma imagem chamada *foto.jpg*, alinhada pela sua parte inferior.

d)

exibe uma imagem chamada *foto.jpg*, alinhada pela sua parte superior.

e)

exibe uma imagem chamada *foto.jpg*, alinhada pela sua parte central.

12 - Configurando as Fontes

É perfeitamente possível alterar o tamanho, o tipo e a cor das fontes utilizadas, através do TAG e seus parâmetros **SIZE**, **FACE** e **COLOR**.

Exemplo:

Exibe texto com a fonte tipo **ARIAL**, tamanho 5, na cor **vermelha**. Para voltar a configuração original utilize .

ATENÇÃO: O parâmetro **TEXT** utilizado com o TAG <BODY> determina a cor da fonte a ser utilizada na *Página Web* inteira, enquanto o parâmetro **COLOR** do TAG determina a cor do texto até que seja encontrado o TAG .

13 - Texto em Movimento

Existe um TAG que faz com que o texto fique em movimento na tela, seu nome é **<MARQUEE>**. O padrão de movimento é da direita para a esquerda, mas pode ser alterado com os seus parâmetros.

```
<marquee behavior=rolagem
            direction=direção
            loop=num_rolagens
            scrollamount=velocidade
            scrolldelay=pixels_a_percorrer>
    texto em movimento
</marquee>
```

Obs.: Não funciona no Netscape.

Vejamos cada um dos parâmetros da Tag MARQUEE:

Behavior: define a forma de rolagem do texto, podendo ter 3 valores possíveis.

SCROLL – rolagem continua em um mesmo sentido

SLIDE – o letreiro sai do ponto de partida e ao chegar ao seu final, pára.

ALTERNATE – o letreiro deslizará de um lado para o outro da tela.

Direction: Define a direção de rolagem do letreiro. Podem ser utilizados dois valores, LEFT (esquerda) e RIGHT (direita).

Loop: Este parâmetro define o número de vezes que o letreiro deslizará pela página, o valor INFINITE (o padrão), define uma rolagem constante.

Scrollamount: Este parâmetro define a velocidade de rolagem do letreiro, dada em milissegundos.

Scrolldelay: Define a quantidade de pixels que irá percorrer cada quadro.

Exemplo:

```
<marquee behavior="alternate" scrolldelay="número">
texto </marquee>
```

[Editar o EXERCÍCIO 06 do Caderno de Exercícios](#)

[Editar o EXERCÍCIO 07 do Caderno de Exercícios](#)

[Editar o EXERCÍCIO 08 do Caderno de Exercícios](#)

[Editar o EXERCÍCIO 09 do Caderno de Exercícios](#)

PARTE 05 - Âncoras

14 - Âncoras

As *Âncoras* são utilizadas para fazer ligações entre partes de uma *Página Web*, através de Links. Para isso, é preciso determinar um ponto de referência na *Página*, que será acessado por um *Link*. Esse ponto de referência é a *Âncora* e receberá um nome que será mencionado pelo *Link* que o acessará. É bastante útil para se movimentar por grandes textos nas *Páginas Web*.

14.1- Definindo Uma Âncora

A TAG usada é **<A NAME>**.

texto opcional

Exemplo:

Início da Página

14.2- Link Para Acessar Âncora

Pode ser um texto ou uma figura que, ao ser clicada, fará ligação direta com a *Âncora* que for mencionada, fazendo com que a *Página* "se mova" até a *Âncora*. O TAG usado é **<A HREF>**.

** hipertexto **

Exemplo:

**Voltar ao Início da Página **

No momento em que este *Link* for clicado, será exibido o texto que estiver a *Âncora* chamada *#inicio*. O símbolo *#* junto ao nome, informa que a *Âncora* está no mesmo documento, ou seja, na mesma *Página Web*.

14.3- Link Para Acessar Âncora Externa

Pode ser um texto ou uma figura que, ao ser clicada, fará ligação direta com uma *Âncora* que esteja em um outro documento, ou seja, em uma outra *Página Web*, fazendo com que a outra *Página* "se mova" até a *Âncora* declarada. O segredo está em declarar o nome da outra *Página* antes do nome da *Âncora*. O TAG usado é o **<A HREF>**.

hipertexto

Exemplo:

Página 02

[Editar o EXERCÍCIO 10 do Caderno de Exercícios](#)

[Editar o EXERCÍCIO 11 do Caderno de Exercícios](#)

[Editar o EXERCÍCIO 12 do Caderno de Exercícios](#)

[Editar o EXERCÍCIO 13 do Caderno de Exercícios](#)

PARTE 06 - Tabelas

15 - Utilizando Tabelas

A utilização de *Tabelas* melhora muito a aparência de uma *Página Web*, mas requer muita atenção, paciência e testes contínuos de utilização, para que se obtenha um bom resultado. Os TAGs são simples, mas a estruturação delas pode ser bastante complexa, dependendo da aplicação requerida:

<code><TABLE>...</TABLE></code>	=> cria uma tabela.
<code><TR>...</TR></code>	=> cria uma linha da tabela.
<code><TD>...</TD></code>	=> cria uma célula em uma linha.
<code><TH>...</TH></code>	=> cria uma célula com texto centralizado e em negrito.
<code><CAPTION>...</CAPTION></code>	=> (opcional) cria um título.

Parâmetros: Nem todos os parâmetros podem ser utilizados em todas as TAGs relativas às tabelas, portanto, os exercícios e a prática, farão com que você se habitue com as aplicações:

- a) **BORDER** = nº (tamanho da borda)
- b) **CELLSPACING** = nº (espaço entre as células)
- c) **CELLPADDING** = nº (espaço entre o conteúdo e a borda da célula)
- d) **WIDTH** = nº ou % (largura da tabela)
- e) **HEIGHT** = nº ou % (altura da célula)
- f) **ALIGN** = Left, Center, Right (alinhamento horizontal)
- g) **VALIGN** = Top, Middle, Bottom (alinhamento vertical)
- h) **BGCOLOR** = cor (cor do fundo de toda a tabela)
- i) **BORDERCOLOR** = cor (cor da borda da tabela)
- j) **NOWRAP** (não permite quebra de texto numa célula)
- k) **COLSPAN** = nº (qtde. de colunas que a célula ocupará)
- l) **ROWSPAN** = nº (qtde. de linhas que a célula ocupará)

15.1 - Tabela Com Células Simples

Exemplo: Para criar a tabela abaixo, a estrutura deve ser:

Clubes de Futebol		
Palmeiras	Corinthians	São Paulo
Santos	São Caetano	Santo André

```
<table border=1>
<caption>Clubes de Futebol</caption>
<!--Criando a 1ª Linha-->
<tr>
  <th>Palmeiras</th>
  <td>Corinthians</td>
  <td>São Paulo</td>
</tr>
<!--Criando a 2ª Linha -->
<tr>
  <td>Santos</td>
  <td>São Caetano</td>
  <th>Santo André</th>
</tr>
</table>
```

15.2 - Tabela Com Células Mescladas

Exemplo: Para criar a tabela abaixo, a estrutura deve ser:

Clubes de Futebol		
Palmeiras		São Paulo
Corinthians	Portuguesa	São Caetano
Santos	Santo André	

```
<table border=2 width=100% bgcolor="yellow">
<caption>Clubes de Futebol</caption>
```

<!--Criando a 1ª Linha-->

```
<tr>
  <th colspan=2 height=50>Palmeiras</th>
  <td>São Paulo</td>
</tr>
```

<!--Criando a 2ª Linha-->

```
<tr>
  <td valign=bottom height=50>Corinthians</td>
  <th align=right>Portuguesa </th>
  <th rowspan=2>São Caetano </th>
</tr>
```

<!--Criando a 3ª Linha -->

```
<tr>
  <td height=50 valign=top align=center>Santos</td>
  <td>Santo André</td>
</tr>
</table>
```

[Editar o EXERCÍCIO 14 do Caderno de Exercícios](#)

[Editar o EXERCÍCIO 15 do Caderno de Exercícios](#)

PARTE 07 - Frames

16 - Utilizando Frames

A utilização de *Frames* possibilita a divisão da janela do *browser* de tal forma que, em cada divisão possa ser executado um programa *HTML* diferente. É possível também interagirmos em um *Frame* e vermos o resultado em outro *Frame*. Para melhor entendimento, imagine que temos um programa *HTML* chamado **ALFA.htm** e outro chamado **BETA.htm**. Para podermos visualizar os dois programas em uma única *Página Web* vamos criar então um programa chamado **JANELAO.htm** que terá as informações necessárias para administrar os dois programas, cada um em uma respectiva *Frame* criada no programa.

16.1 - Definindo o Lay-Out dos FRAMES

O TAG **<FRAMESET>** do tipo *contêiner*, ou seja, deve ser aberto e fechado, e define a aparência da página principal dos *Frames*. A seguir estão seus parâmetros:

- a) **COLS** = cria frames dispostos em colunas. Deve especificar os tamanhos das colunas separados por vírgulas. Os valores podem ser em pixels, em percentagens ou relativos. Exemplos:
`<FRAMESET COLS =100,300,200>` ou
`<FRAMESET COLS =200,*>` ou
`<FRAMESET COLS =60%,40%>`
- b) **ROWS** = cria frames dispostos em linhas. Deve especificar os tamanhos das linhas separados por vírgulas. Os valores podem ser em pixels, em percentagens ou relativos. Exemplos:
`<FRAMESET ROWS =100,300,200>` ou
`<FRAMESET ROWS =200,*>` ou
`<FRAMESET ROWS =60%,40%>`

Obs.: Para podermos combinar a utilização de **ROWS** e **COLS** devemos criar um **FRAMESET** para cada um, de forma aninhada.

- c) **FRAMEBORDER** = elimina, ou não, as bordas dos *Frames*. O valor **1** é o padrão e exibe as bordas. O valor **0** elimina a exibição das bordas.
- d) **FRAMESPACING** = altera o espaço entre os *Frames*, dando a impressão de que as bordas foram alteradas. Valor do espaçamento em pixels.

16.2 - Definindo o Conteúdo dos FRAMES

O TAG **<FRAME>** atribui o conteúdo de cada *Frame* e suas características. Os atributos do comando **<FRAME>** se sobrepõem aos atributos do comando **<FRAMESET>**. A seguir estão seus parâmetros:

- a) **ALIGN** = alinhamento do *Frame* ou do texto. Os valores válidos para alinhamento são:
 - **TOP**: parte superior do *Frame*;
 - **MIDDLE**: parte central do *Frame*;
 - **BOTTOM**: parte inferior do *Frame*;
 - **LEFT**: parte esquerda do *Frame*;
 - **RIGHT**: parte direita do *Frame*.
- b) **FRAMEBORDER** = aplicação idêntica à utilização em **<FRAMESET>**.
- c) **MARGINHEIGHT** = altura das margens superior e inferior do *Frame*, em pixels.

d) **MARGINWIDTH** = largura das margens esquerda e direita do *Frame*, em pixels.

e) **NAME** = atribui um nome (apelido) ao *Frame*, de maneira que ele possa ser identificado e localizado posteriormente.

f) **NORESIZE** = evita que as bordas do *Frame* sejam redimensionadas pelo usuário.

g) **SCROLLING** = permite que o *Frame* possua barras de rolagem (valor YES) ou não (valor NO).

h) **SRC** = endereço (URL) do documento que será exibido no *Frame*.

16.3 - Interligação entre FRAMES

Veja o exemplo dado anteriormente. Imagine que o arquivo chamado **JANELAO.htm** seja uma *Página Web* que divida a tela em duas partes, uma à direita e outra à esquerda. Imagine que a página da esquerda ocupe 200 pixels da tela e a página da direita ocupe o restante. cada uma dessas partes chamamos de **FRAMES**, e chamaremos a parte da direita de **Frame_Direita** e a parte da esquerda de **Frame_Esquerda**.

No **Frame_Esquerda** vamos exibir um arquivo chamado **ALFA.htm** que será uma espécie de *Menu de Opções*, ou seja, nele haverá *Links* para diversas outras páginas, e queremos que essas páginas sejam exibidas no **Frame_Direita**, onde inicialmente será exibido o arquivo chamado **BETA.htm**.

Frame_Esquerda (ALFA.htm)	Frame_Direita (BETA.htm e demais páginas acessadas)
------------------------------	---

Vamos então escrever o programa **JANELAO.htm** que será a *Página Principal*, que irá administrar os **FRAMES**:

```
<html>
<head>
<title>Título do Página</title>
</head>
<frameset cols="200,*">
  <frame name="Frame_Esquerda" src="alfa.htm">
  <frame name="Frame_Direita" src="beta.htm">
</frameset>
</html>
```

A *Página Principal* dividiu a janela em 2 colunas, sendo que a coluna da esquerda possui 200px de comprimento. Agora basta carregar a *Página Principal* que as outras 2 páginas serão automaticamente exibidas nos respectivos **FRAMES**. Para fazer a ligação entre os **FRAMES** precisamos avisar ao arquivo **ALFA.htm** que as páginas que ele abrir serão carregadas no **Frame_Direita**, e podemos fazer isso de duas formas:

- a) No arquivo **ALFA.htm**, na seção **<head>** declaramos o TAG:
`<base target="destino">`
Exemplo:
`<head> <base target="Frame_Direita"> </head>`
- b) Ou, no arquivo **ALFA.htm**, junto ao TAG **<a href>** declaramos a parâmetro **target**:
``
Exemplo:
``

16.4 - Criando FRAMES Aninhados

É bastante comum encontrarmos *Páginas Web* utilizando *Frames Aninhados*, ou seja, *Frames* dentro de *Frames*. Imagine uma *Página Web* com o seguinte Lay-Out:

Frame_Superior (DELTA.htm)	
Frame_Esquerda (ALFA.htm)	Frame_Direita (BETA.htm e demais páginas acessadas)

Normalmente o **Frame_Superior** é utilizado para exibir propagandas (*banners*) e informações sobre o próprio *Site*, o **Frame_Esquerda** é utilizado para abrigar os *Links* e o **Frame_Direita** é utilizado para abrir as *Páginas* acessadas pelos *Links*.

Na montagem da nossa *Página Principal*, criamos duas linhas de *FRAMES*, sendo que a 1ª linha terá 100px de altura, e a 2ª linha será subdividida em 2 colunas de *FRAMES*. O *FRAME* da esquerda terá 200px de comprimento, conforme o exemplo:

```
<html>
<head>
<title>Título do Página</title>
</head>
<frameset rows =100,*>
<frame name = Frame_Superior src="delta.htm">
  <frameset cols = 200,*>
    <frame name = Frame_Esquerda src="alfa.htm">
    <frame name = Frame_Direita src="beta.htm">
  </frameset>
</frameset>
</html>
```

ATENÇÃO: Os TAGS **<FRAMESET>** e **<FRAME>** não podem estar na área do **<BODY>**, caso contrário não serão exibidos.

16.5 - Navegadores sem FRAMES

Não é comum encontrarmos navegadores antigos que não aceitam a utilização de *Frames*, mas devemos nos precaver. Existe um TAG bem simples que impede que erros ocorram por conta disso, é o **<NOFRAMES>**:

Exemplo:

```
<noframes>
<body>
  O seu Navegador não suporta FRAMES, esta página
  não pode ser mostrada!
</body>
</noframes>
```

[Editar o EXERCÍCIO 16 do Caderno de Exercícios](#)

[Editar o EXERCÍCIO 17 do Caderno de Exercícios](#)

PARTE 08 - Formulários

17 - Utilizando Formulários

Os formulários eletrônicos gerados pela *Linguagem HTML* são recursos dos mais importantes oferecidos pelas *Páginas Web*: a troca de informações entre o usuário e o servidor, enviando dados que serão processados no servidor e devolvidos ao cliente. Os exemplos aqui mostrados funcionarão para efeito de teste, pois para que um formulário troque informações com o servidor, você precisará verificar com o seu provedor se ele permite o uso de formulários, em caso afirmativo o provedor deverá orientá-lo como realizar o acesso, pois no provedor deverá haver programas especiais (CGI), escritos em linguagens de programação como Perl, Visual Basic, C, PHP, etc, para fazer o processamento dos dados.

17.1 - O TAG <FORM>...</FORM>

Possui duas funções importantes: especificar o local do programa que controlará o formulário e definir a forma como os dados serão enviados. O TAG **<FORM>** usa os seguintes parâmetros:

- a) **NAME = "nome"**: especifica o nome do formulário.
- b) **ACTION="URL"**: especifica o local (*URL*) do *Servidor* e do programa *CGI* que vai processar o formulário.
- c) **METHOD="GET" ou "POST"**: indica o método usado pelo *Servidor* para receber o formulário. **POST** é o mais usado e transmite toda a informação do formulário imediatamente após o *URL*. **GET** faz com que o conteúdo do formulário seja anexado ao endereço do *URL*.

Exemplo:

```
<form name="form1" action="XXX" method="get">
```

17.2 - O TAG <INPUT>

Define um campo para entrada de dados. O TAG **<INPUT>** usa os seguintes parâmetros:

- a) **TYPE = "tipo de variável"**: especifica o tipo de variável a ser utilizada naquele objeto, podem ser:

TEXT	caracteres.
PASSWORD	senha oculta.
CHECKBOX	caixa de verificação (sim/não).
RADIO	botão de rádio (exclusivo).
SUBMIT	enviar o conteúdo do formulário.
RESET	apaga o conteúdo do formulário.
BUTTON	botão genérico.

- b) **NAME = "nome_objeto"**: todo objeto precisa ter seu próprio nome, exceto para os tipos *CHECKBOX* e *RADIO* onde todos os itens possuem o mesmo nome em uma mesma seção.

- c) **VALUE = "valor"**: depende do tipo do objeto:

TEXT ou PASSWORD	conteúdo digitado.
CHECKBOX ou RADIO	valor a ser enviado.
SUBMIT, RESET, ou BUTTON	rótulo do botão.

- d) **CHECKED**: determina uma marca em um campo tipo *CHECKBOX* ou *RADIO*.

- e) **SIZE = "tamanho"**: especifica o tamanho visual do campo tipo *TEXT* ou *PASSWORD*.
- f) **MAXLENGTH = "valor"**: especifica a quantidade máxima de caracteres aceitos no campo *TEXT* ou *PASSWORD*.

Exemplos:

1. Caixa de Texto:
Digite o nome:
`<INPUT TYPE="TEXT" NAME="texto1" VALUE="" SIZE="20" MAXLENGTH="40">`
2. Caixa para Senha:
Digite a senha:
`<INPUT TYPE="PASSWORD" NAME="senha1" VALUE="" SIZE="6" MAXLENGTH="6">`
3. Caixa de Verificação:
Características:
`<INPUT TYPE="CHECKBOX" NAME="caixa1" VALUE="legal" CHECKED>Sou Legal`
`<p>`
`<INPUT TYPE="CHECKBOX" NAME="caixa1" VALUE="bonito>Sou Bonitão`
4. Botão de Rádio:
Estado Civil:
`<INPUT TYPE="RADIO" NAME="radio1" VALUE="S" CHECKED>Solteiro`
`<p>`
`<INPUT TYPE="RADIO" NAME="radio1" VALUE="C">Casado`
5. Botão para Apagar:
`<INPUT TYPE="RESET" NAME="apaga1" VALUE="Apagar">`
6. Botão para Enviar:
`<INPUT TYPE="SUBMIT" NAME="envia1" VALUE="Enviar">`

17.3 - O TAG <SELECT>...</SELECT>:

Define uma lista de itens. A TAG <SELECT> usa os seguintes parâmetros:

- a) **NAME = "nome"**: especifica o nome da lista.
- b) **SIZE = "tamanho"**: especifica a quantidade de itens que serão exibidos de uma vez.
- c) **MULTIPLE**: especifica que vários itens podem ser escolhidos.

Cada item da lista deve ser especificado através do TAG <OPTION>. O parâmetro **SELECTED** pré seleciona um item.

Exemplo:

```
<SELECT NAME="lista1" SIZE="2">  
  <OPTION SELECTED>Item 1  
  <OPTION>Item 2  
</SELECT>
```

17.4 - O TAG <TEXTAREA>...</TEXTAREA>

Define uma caixa de digitação de textos. O TAG <TEXTAREA> usa os seguintes parâmetros:

- a) **NAME = "nome"**: especifica o nome da área de texto.
- b) **ROWS = "linhas"**: especifica a quantidade de linhas que serão exibidas na área de texto.
- c) **COLS = "colunas"**: especifica quantidade de colunas que serão exibidas na área de texto.

Exemplo:

```
<TEXTAREA NAME="area1" COLS="50" ROWS="10">  
</TEXTAREA>
```

17.5 - O TAG <FIELDSET>...</FIELDSET>

Define uma moldura para agrupar determinados campos. O TAG <FIELDSET> usa os seguintes parâmetros:

- a) **<LEGEND>...</LEGEND> = "rótulo"**: especifica um rótulo para o grupo
- b) **ACCESSKEY = "tecla"**: especifica uma tecla de atalho para mover o cursor àquela área.

Exemplo:

```
<FIELDSET ACCESSKEY="X">  
<LEGEND> Pressione CTRL+X </LEGEND>  
--  
--  
--  
</FIELDSET>
```

[Editar o EXERCÍCIO 18 do Caderno de Exercícios](#)

[Editar o EXERCÍCIO 19 do Caderno de Exercícios](#)

18 - Introdução

Este comando é utilizado para criar uma *metainformação*, pouco explorado na linguagem *DHTML* (*HTML Dinâmico*), contudo, pode ser responsável por importantes aspectos como a criação de documentos dinâmicos ou informações específicas que serão usadas pelo servidor ou pelos mecanismos de busca da internet.

18.1- Informações Específicas

O comando *META* deve sempre ser declarado na área de cabeçalho, ou seja, dentro dos *TAGs* `<HEAD>` `</HEAD>`. A sintaxe básica é:

```
<META HTTP-EQUIV=resposta CONTENT=descrição  
NAME=descrição URL=endereço>
```

Onde:

HTTP-EQUIV=resposta

Permite inserir informações nos cabeçalhos dos documentos.

CONTENT=descrição

Define a informação associada ao nome da variável ou metainformação.

NAME=descrição

É o nome da metainformação ou variável que está sendo criada, e que deve ser atribuído a um conteúdo através da opção **CONTENT**.

URL=endereço

Indica o URL do documento.

Exemplo: Vamos forçar a página inicial a fazer um *recarregamento automático* de uma outra página, após um determinado tempo:

```
<HTML>  
<HEAD>  
<TITLE>Exercício com o comando META</TITLE>  
  
<META HTTP-EQUIV= "Refresh" CONTENT=" 5; URL=  
pagina2.htm">  
  
</HEAD>  
<BODY>  
Esta página será exibida por apenas 5 segundos.  
</BODY>  
</HTML>
```

Caso você queria apenas ficar atualizando periodicamente a mesma página inicial, basta não declarar o parâmetro URL, conforme abaixo:

```
<META HTTP-EQUIV= "Refresh" CONTENT=" 5">
```

18.2- Mecanismos de Busca da Internet

Para fazer com que sua página seja encontrada pelos mecanismos de busca da internet é preciso que você dê uma mãozinha, e o comando *META* pode ser muito útil. Geralmente, se o mecanismo não encontra nenhuma dica sobre o conteúdo da página, ele lê as primeiras palavras encontradas, outros pegam todas as palavras da página, mas a maioria se utiliza do comando *META* para obter palavras-chave realmente importantes em sua página. Veja a seguir uma lista de parâmetros utilizados pelos mecanismos de busca:

- **TITLE**

Utilizado no `<HEAD>` para declarar o título da página.

- **Keywords**

Deve conter uma lista de palavras-chave separadas por vírgula. Coloque as palavras mais importantes primeiro.

- **Description**

Essa descrição é exibida junto com o título da página, no resultado de uma busca.

Exemplo: Vamos dar informações aos mecanismos de busca da internet, fornecendo palavras-chave e uma descrição do conteúdo da página web.

```
<HTML>  
<HEAD>  
<TITLE>Comando META para Busca na Net</TITLE>  
  
<META HTTP-EQUIV= "Keywords" CONTENT="HTML,  
Javascript, Colégio Singular">  
  
<META HTTP-EQUIV= "description" CONTENT="Site  
do Professor Celso Gallão">  
  
</HEAD>  
<BODY>  
Esta é página principal de meu site  
</BODY>  
</HTML>
```

Editar o EXERCÍCIO 20 do Caderno de Exercícios

PARTE 10 - Folhas de Estilo

19 - Introdução

Este comando é utilizado para facilitar a apresentação visual da página web, e agilizar a manutenção de um site, através da centralização das alterações de diversas páginas em um só lugar. Também aumenta as possibilidades de formatação visual das páginas pois permite alterar atributos válidos para todo o site. Uma folha de estilo pode ser definida como um gabarito que formata os comandos HTML de uma página de acordo com as preferências do programador.

19.1- Regras de Utilização

A maneira mais simples de utilizar a folha de estilo é através da TAG **<STYLE>**, que é especificada dentro da seção **<HEAD>**. Cada regra é composta por um SELETOR, que é um elemento que receberá o estilo, por exemplo **<BODY>**, **<P>**, **<H1>**, etc. Cada TAG HTML possui várias propriedades que podem ter seu ESTILO alterado. Cada propriedade possui um VALOR que pode ser alterado. Veja a sintaxe básica:

**Seletor { propriedade1: valor1;
propriedade2: valor2 };**

Exemplos:

- a) **P { color: "green" };**
Todos os textos entre **<P>** e **</P>** serão verdes.
- b) **H1 { font: 20pt "arial"; font-weight: "bold"; color: "brown" };**
Todos os textos entre **<H1>** e **</H1>** terão a fonte arial, tamanho de 20pt, negrito e marrom.

Pode-se agrupar diversos seletores para que recebam a mesma mudança de estilo, conforme o exemplo:

- c) **H1, H2, H3 { color: "red"; font-family: "Verdana" };**
Todos os textos entre **<H1>** e **</H1>**, **<H2>** e **</H2>**, **<H3>** e **</H3>**, serão vermelhos e com fonte verdana.

19.2- Aplicação da Folha de Estilo

Dentre as maneiras de se aplicar folhas de estilo, cada uma delas apresenta vantagens e desvantagens. Vamos aqui conhecer as três formas mais utilizadas:

19.2.1 - O Método LINKING

Linking (**ligando**) consiste em carregar os comandos da folha de estilo que se encontram em um arquivo externo com extensão **CSS**. Essa técnica tem como vantagem a facilidade de alterar os estilos de todo o site de uma só vez, pois todas as páginas que sofrerão as alterações farão o carregamento dos estilos diretamente deste arquivo externo. Veja a sintaxe, que deve estar sempre na seção **<HEAD>**:

**<LINK REL=StyleSheet
HREF="arquivo.css"
TYPE="text/css">**

Exemplo: Vamos carregar os estilos que estão gravados no arquivo externo chamado **exemplo.css**:

```
<HTML>
<HEAD>
<TITLE>Folha de Estilo Externa</TITLE>

<LINK REL=StyleSheet HREF="exemplo.css"
TYPE="text/css">

</HEAD>
<BODY>
Esta é página principal de meu site
</BODY>
</HTML>
```

19.2.2 - O Método EMBEDDING

Embedding (**incorporando**) consiste em declarar os comandos da folha de estilo dentro do documento HTML, na seção **<HEAD>** da própria página web. Essa técnica tem como vantagem configurar apenas uma página web do site. A incorporação é feita através da TAG **<STYLE>**. É conveniente declarar os seletores de estilo dentro da TAG de comentário HTML para evitar que navegadores antigos acusem erros. Veja a sintaxe, que deve estar sempre na seção **<HEAD>**:

```
<STYLE TYPE="text/css">
<!--
declare aqui os seletores
-->
</STYLE>
```

Exemplo: Vamos incorporar nesta página web os estilos para **<BODY>** com fonte arial, 10, verde e cor do fundo verde-claro; para **<H1>** com fonte arial, 20, negrito e marrom; para os LINKS com fonte verdana, 10, verde-escuro e para os LINKS VISITADOS com fonte verdana 10, cor de laranja.

```
<HTML>
<HEAD>
<TITLE>Folha de Estilo Incorporada</TITLE>

<STYLE>
<!--
    BODY {font: 10pt "arial"; color: "green";
background: "lightgreen"};
    H1 {font: 20pt "arial"; font-weight: "bold";
color: "brown"};
    A:LINK {font: 10pt "verdana"; color: "darkgreen"};
    A:VISITED{font: 10pt "verdana"; color: "orange"};
-->
</STYLE>

</HEAD>
<BODY>
<H1>Esta é página principal de meu site</H1>
<A HREF="exemplo.htm">Exemplo</A>
</BODY>
</HTML>
```

19.2.3 - O Método INLINE

InLine (**na linha**) consiste em declarar os comandos da folha de estilo aplicando diretamente no elemento (seletor). Essa técnica tem como vantagem configurar apenas uma ocorrência do estilo, ou para anular somente naquele elemento um estilo pré configurado em um arquivo **CSS**. A declaração é feita através do comando style dentro da TAG do seletor.

<TAG style = seletor: valor >

Exemplo: Imagine que uma página HTML carregue um arquivo de Folha de Estilo (CSS) externo. Nas declarações de estilo, imagine que foi declarado um estilo para a TAG <P>, conforme abaixo:

```
<STYLE>
<!-- Este é o Arquivo Externo CSS -->
P {border:"double"};
</STYLE>
```

Agora imagine que na página HTML tem uma ocorrência da TAG <P> que o programador quer anular o estilo pré formatado. Para isso basta declarar na ocorrência um novo estilo, que será válido apenas naquela ocorrência.

```
<!-- Esta é a página HTML -->
<P style=border: "none">
```

19.3- Unidades de Medida

As unidades de medida utilizadas, dependem do seletor, e podem ser:

in (inches, polegadas; 1pol = 25,4mm)
cm (centímetros; 1cm = 10mm)
mm (milímetros)
pt (pontos; 1pt = 1/72pol)
px (pontos por polegadas)

Também podem ser indicados em porcentagem.

19.4- Definindo Estilo de Cores

Um valor de cor pode ser especificado através das 16 cores da paleta VGA:

aqua; black; blue; fuchsia; gray; green; lime; maroon; navy; olive; purple; red; silver; teal; white e yellow.

Ou através de cor do Sistema RGB, da seguinte forma:

#rrggbb ou #rgb ou rgb(x,x,x) ou rgb(y%, y%, y%)

Também podem ser indicados em porcentagem.

19.5- Fazendo Referência a URL

Um URL pode ser informado para carregar uma figura, por exemplo. As aspas são opcionais:

BODY { BACKGROUND: URL("logosing.gif") };

19.6- Tabela de Seletores

A seguir, uma tabela com os principais seletores, seus atributos, valores e exemplos:

FONT-SIZE	Ajusta o tamanho da fonte	pt, in, cm, px
{ font-size: 12pt };		

FONT-FAMILY	Seleciona uma família de fontes. Se uma falhar carrega a próxima.	nomes das fontes do windows, separadas por vírgulas
{ font-family: arial, verdana, script };		

FONT-WEIGHT	Ajusta a espessura da fonte (negrito).	extra-light;light; demi-light; medium; demi-bold; bold; extra-bold;
{ font-weight: bold };		

FONT-STYLE	Aplica, ou não, o efeito itálico.	normal; italic;
{ font-style: italic };		

LINE-HEIGHT	Especifica a distância entre as linhas.	pt, in, cm, px, %
{ line-height: 24pt };		

COLOR	Especifica a cor do texto.	nome da cor VGA ou código RGB.
{ color: blue };		

TEXT-DECORATION	Aplica, ou não, efeitos no texto.	none; underline; italic; line-through;
{ text-decoration: underline };		

MARGIN-LEFT	Ajusta a distância da margem esquerda da página.	pt, in, cm, px
{ margin-left: 2in };		

MARGIN-RIGHT	Ajusta a distância da margem direita da página.	pt, in, cm, px
{ margin-right: 1cm };		

MARGIN-TOP	Ajusta a distância da margem superior da página.	pt, in, cm, px
{ margin-top: 20px };		

TEXT-ALIGN	Especifica o alinhamento do texto	left; center; right;
{ text-align: right };		

TEXT-INDENT	Especifica a distância entre o texto e a margem esquerda	pt, in, cm, px
{ text-indent: 0.5in };		

BACKGROUND	Especifica uma figura de fundo ou uma cor.	URL; código da cor;
{ background: #33CC00 }; ou { background: URL("figura.gif") };		

O Seletor **BACKGROUND** possui também os seguintes parâmetros adicionais:

REPEAT: Repete a imagem (default);

NO-REPEAT: A imagem é inserida apenas uma vez;

FIXED: A imagem não rola com a tela;

SCROLL: A imagem rola com a tela (default);

POSITION: Define a posição inicial da imagem. Pode ser especificado em percentual ou tamanho, tendo como referência o canto superior esquerdo (0%,0%).

Exemplo:

{ **BACKGROUND: URL("figura.gif") RIGHT BOTTOM NO-REPEAT** };

BORDER-COLOR	Especifica a cor da borda.	nome da cor VGA ou código RGB.
{ text-align: right };		

BORDER-STYLE	Especifica o estilo da borda.	dotted; dashed; solid; double; none; grove; ridge; inset; outset;
{ border-style: grove };		

BORDER-WIDTH	Especifica a espessura da borda.	thin; thick; medium;
{ border-width: thin };		

BORDER	Especifica a lateral da borda que sofre o efeito.	top; left; right; center;
{ border: right };		

SCROLLBAR-BASE-COLOR	Especifica a cor da Barra de Rolagem.	nome da cor VGA ou código RGB.
{ scrollbar-base-color: lightblue };		

Editar o EXERCÍCIO 21 do Caderno de Exercícios