

# *Apostila de C.S.S.*

*(Curso Introdutório)*

*Versão 0.2a*  
*6 de abril de 2005*

Prof. Luís Rodrigo de O. Gonçalves

E-mail: [luisrodrigoog@yahoo.com.br](mailto:luisrodrigoog@yahoo.com.br)

Site: <http://www.lrodrigo.cjb.net>

OBS: partes deste documento foram retiradas  
de sites, cujas referências se perderam

# Introdução

O uso de estilos nas páginas *WEB* permite que diferentes atributos possam ser agrupados, nomeados e aplicados a vários elementos, deste modo, a partir de um ponto central, arquivo de estilos, é possível padronizar a aparência de todas as páginas de um site.

Uma folha de estilo *CSS* (folha de estilo em cascata) não é *XHTML*, mas sim, um conjunto de informações sobre a formatação e exibição dos componentes do *layout* de uma página *WEB*. Ela é um código separado que altera as características da página. O *CSS* deve ser utilizado para liberar do *XHTML* ou do *HTML* o peso da responsabilidade da apresentação.

Existem quatro versões de *CSS*: *CSS Level 1 (CSS1)* lançada pelo *W3C Consortium* em 1996, a *CSS-Positioning (CSS-P)*, que é a segunda versão já permitia o posicionamento dos elementos, *CSS Level 2 (CSS2)* foi lançada em 1998 e *CCS Level 3 (CSS3)* que ainda está sendo desenvolvido.

O *CSS* possui três tipos de regras: há os *seletores XHTML*, que especificam como a *tag* será exibida; as *classes*, que podem possuir qualquer nome e podem ser aplicados a diversos elementos diferentes; e finalmente o *ID* que pode ser aplicado a qualquer *tag* do *XHTML*, mas somente a um elemento da página, normalmente utilizado junto com linguagens de script (*javascript* e *vbscript*).

Uma regra é formada por três partes: o *seletor*, que identifica a regra; a *propriedade* que informa o que está sendo definido; e os *valores* que são os modificadores das propriedades. Por exemplo:

```
seletor {propriedade01:valor01;propriedade02:valor02;}  
#objeto1 { position:absolute;top:10px; }  
.MinhaClasse { font:bold 12pt times; }  
p { font:bold 10pt arial; }
```

**OBS: Um erro frequente nas regras *CSS* é a ausência do último ponto e vírgula (;).**

O *CCS* pode ser incluído em três lugares de um documento:

- ***Diretamente na tag***: o uso da *CSS* nas *tags* individuais é útil quando se deseja substituir os estilos definidos universalmente. Estas declarações devem estar entre aspas (" "). Por Exemplo:

```
<h1 style = "color:blue;" > .... </h1>
```

- **Na página WEB:** este uso do CSS permite a inserção de informações que serão aplicadas a todo um documento. Por Exemplo:

```
<html>
<head>
  <style type="text/css">
    body { background: white url(papagaio.jpg) no-repeat; }
    #content { position: relative;
               top: 190px;
               left 165px;
               width 480px; }
    h1 { font: small-caps bold italic 2.5em arial }
  </style>
</head>
<body>
  ....
</body>
```

- **Arquivo externo:** este deve ser um arquivo texto composto somente de regras CSS e comentários. Podendo ser vinculado ao documento com *XHTML* através das tags *<link>* ou *@import*. Por exemplo:

```
Arquivo Externo: modelo.css

body { background: white url(papagaio.jpg) no-repeat; }
content { position: relative;
          top: 190px;
          left 165px;
          width 480px;
        }

1 { font: small-caps bold italic 2.5em arial }
```

```
Arquivo com XHTML
<html>
<head>
  <link rel="stylesheet" type="text/css" media="print" href="modelo.css" />
  <style media="screen" type="text/css">
    @import url(modelo2.css)
  </style>
</head>
<body>
</body>
</html>
```

**OBS:** A tag *@import* não funciona muito bem no Nestcape, logo devemos preferencialmente utilizar a tag *<link>*.

## Definindo o comportamento dos elementos da página.

Para se definir o comportamento dos elementos da página deve-se incluir no arquivo externo o nome do elemento a ser personalizado seguido de um sinal de abertura chave ( { ); dos pares de atributos e valores separados por um ponto e vírgula; e finalmente um sinal de fechamento chave ( } ). Por exemplo:

```
p { color:#666666;  
font-size:10pt;  
line-height:18px;  
font-family: Arial; }
```

Além de alterar o comportamento de todos os elementos de um determinado tipo, podemos criar classes que podem ser aplicadas a vários elementos diferentes. Por exemplo:

```
.minhaClasse { font-size:12pt; font-family:Arial }
```

Para utilizar esta classe no documento pode-se utilizar um código parecido com:

```
<p class="minhaClasse"> Teste de CSS </p>
```

É possível criar uma classe para um elemento específico com a seguinte regra de CSS:

```
elemento.classe { atributos:valor;}
```

Por exemplo:

```
p.arial10 { font-size:10pt; font-family:Arial; };
```

Outra maneira de formatar um elemento é baseada no seu identificador único (ID), neste caso devemos declarar a ID no CCS e depois aplica-lo a um elemento do documento. Por exemplo:

```
Na declaração do CCS:  
#titulo1 { font-size:26px }
```

```
No corpo da página:  
<p id="titulo1"> Primeiro Título </p>
```

Quando mais de um elemento possui as mesmas características podemos realizar a declaração dos mesmos em uma única linha, como mostra o exemplo abaixo:

```
h1,h2,h3,p {font-size:12pt;}
```

## Criando nossos elementos.

Do mesmo modo como é possível alterar o comportamento de elementos preexistentes no XHTML, existe a possibilidade de se criar os próprios elementos através das tags **<span>** e **<div>**. A tag **<span>** não herda as propriedades do elemento pai, servindo como uma folha em branco para a criação de novas tags inline. Já o **<div>** é útil na criação de elementos de estilo de parágrafos.

Exemplo de uso para **<span>**:

Na declaração do CSS coloque:

```
span.highlight { background-color:#ffcccc}
```

No corpo do documento coloque:

```
Este texto está em <span class="highlight"> em destaque </span>
```

Como podemos ver o elemento **<span>** pode ser incluído dentro de um parágrafo, pois ele não possui as quebras superior e inferior, já o mesmo não ocorre com o elemento **<div>** como podemos ver no exemplo abaixo:

Na declaração do CCS coloque:

```
div.highlight { background-color:#ffcccc}
```

No corpo do documento coloque:

```
Este texto está em <div class="highlight"> em destaque </span>
```

Assim como outras “línguas” o CSS também aceita comentários, básica mente existem três tipos de comentários:

- **“//”** : comentários em uma única linha, tudo que aparecer após estes caracteres serão considerados comentários.
- **/\* .... \*/** : comentários de várias linhas, onde o comentário deve começar com os caracteres **“/\*”** e ser encerrado pelos caracteres **“\*/”**.

## Elementos Aninhados

Existem casos onde elementos do *XHTML* aparecerão dentro de outros elementos, por exemplo uma ligação hipertexto geralmente aparece dentro de um elemento de parágrafo, nestes casos o elemento parágrafo é o elemento pai e a ligação é o elemento filho.

Assim como ocorre os elementos individuais os elementos aninhados também podem ser tratados pela CCS. Por exemplo:

No arquivo do CSS coloque:

```
p a.link { color:red;
           text-decoration:underline;
        }
div.menu a.link {
           color:#900;
           font-weight:bold;
           text-decoration:none;}
div.menu { font: bold 16px Arial, Helvetica, sans-serif; }
p { font: 12px Arial }
```

No corpo do documento coloque:

```
<div class=" menu">
    <a href="#"> Link I </a> | <a href="#"> Link II </a>
</div>
<hr />  <br />  <br />
<p>
    Este é o texto do parágrafo e este é o <a href="#"> link do parágrafo </a>
</p>
```

## Trabalhando com a mídia de visualização diferentes:

Além de ser exibido em um navegador, um documento pode ser manipulado por outros meios de visualização, como um leitor braile ou por uma impressora, deste modo o **CCS** deve permitir a adaptação da visualização de acordo com o tipo do meio de exibição/manipulação. Para isto, é possível fazer uso de duas ou mais chamadas ao elemento "link". Por exemplo:

No cabeçalho do documento coloque:

```
<link rel="stylesheet" href="print.css" media="print">
<link rel="stylesheet" href="screen.css" media="screen">
```

Deste modo, dependendo da mídia utilizada para visualizar o documento, um arquivo de CSS diferente será carregado para realizar a formatação dos elementos do texto.

# Manipulando as Fontes dos Documentos

O estilo/formato de uma fonte pode ser definido através da escolha de uma das várias famílias de fontes existentes. Em CSS voltado para *XHTML* há cinco famílias (**font-family**) distintas:

- **Serif ( Times New Roman )** : que devem ser utilizada quando se deseja uma fonte com *serifas*, as quais facilitam a leitura, uma vez que destaca melhor as letras de forma individual;
- **Sans serif ( Helvetica e Arial )** : são fontes sem serifa, usadas normalmente em texto menores exibidos na tela;
- **Nonospace ( Courier New )** : são fontes nas quais as letras ocupam a mesma quantidade de espaço, geralmente utilizadas em textos que devem ser lidos com perfeição, mesmo que de uma maneira mais lenta, como por exemplo o código fonte de programas;
- **Cursive ( Apple Chancery )** : são altamente estilizadas e geralmente utilizadas em texto decorativo;
- **Fantasy ( webdings - ¨ ¨ ¨ ¨ ¨ ¨ ¨ ? )** : são fontes extremamente ornamentais, geralmente reservadas para textos decorativos.

Para que definir qual será a fonte utilizada em um documento se faz uso da propriedades **“font-family”**, dentre os possíveis valores desta propriedade destacam-se: **times, “Times New Roman”, palatino, serif, Arial, Helvetica, Geneva, sans-serif, “Courier New”, Courier, Monaco, nonospace**. Quando desejamos oferecer uma lista de opções, cada nome de fonte deve estar separado por uma vírgula (,). Por exemplo:

```
p { font-family: Arial, Helvetica, Geneva, sans-serif; }
```

Todavia, é possível realizar o *download* das fontes diretamente da internet. Com isto, independente do sistema operacional dos clientes a fonte sempre estará disponível. Mas as fontes devem ser previamente processadas antes de serem liberadas para *download* e este processamento é diferente para o Internet Explorer e para o Netscape.

Para carregar as fontes diretamente da *internet*, deve-se fazer uso do elemento “**@font-face**”, que deve aparecer dentro do elemento “**style**” da página que acessa a fonte. Por exemplo:

```
<html>
  <head>
    <style type="text/css">
      @font-face { font-family: Garamond; src: url(Garamond.eot); }
      h2 { font-family: Garamond, "Times New Roman", serif; }
    </style>
  </head>
  <body>
    ...
    ...
  </body>
</html>
```

Uma forma mais segura para se exibir um texto utilizando uma formatação desejada, e através do uso das fontes “**browser-safe**”, ou seja, aquelas que estão disponíveis na maioria dos sistemas. Destas fontes podemos destacar:

- Andale Mono;
- Arial;
- Comic Sans;
- Georgia;
- Impact;
- Minion Web;
- Trebuchet;
- Verdana;
- Times.

Assim como podemos mudar as fontes do texto para melhorar a exibição ou para dar um destaque ao texto, também podemos alterar seu tamanho através da propriedade “**font-size**”, que aceita como valor as unidades de comprimento, as quais geralmente são especificadas em pontos (**pt**); as expressões absolutas (**xx-small**, **x-small**, **small**, **medium**, **large**, **x-large** e **xx-large**), sendo que as propriedades **smaller** e **larger** podem ser utilizadas para especificar o tamanho da fonte com relação ao tamanho da fonte do elemento pai; e finalmente podemos utilizar a *porcentagem* como unidade de medida baseada no tamanho da fonte do elemento pai.

Caso se deseje fazer a definição do tamanho da fontes em pontos (**pt**), um bom tamanho é o 12pt sendo de fácil leitura na maioria dos navegadores. Mas devemos lembrar que o tamanho de um ponto varia de sistema operacional para sistema operacional. Uma forma mais precisa de especificar o tamanho dos elementos é através dos pixels (**px**), que são mais precisos que os pontos, sendo que geralmente 12px é equivalente à 12pt.



Outro fator importante na formatação de um texto é se o mesmo será exibido em itálico ou não, para tal informação devemos fazer uso da propriedade “**font-style**” que pode assumir os valores:

- **normal** : remove o estilo da fonte, ou seja impede que o mesmo herde as características do pai;
- **italic** : utiliza uma versão em itálico da fonte;
- **oblique** : inclina o texto a direita.

Muitas vezes **italic** e **oblique** são confundidos, mas o primeiro faz uso de outra versão da mesma fonte e o segundo apenas inclina a fonte. Quando o efeito **italic** é solicitado mas não existe uma versão disponível o efeito **oblique** é aplicado.

Assim como podemos definir o uso do itálico em uma fonte podemos determinar se ela será apresentada em negrito, bem como o seu grau através da propriedade “**font-weight**” que aceita os valores:

- **lighter** : mais claro que o elemento pai;
- **normal** : o texto não será apresentado em negrito;
- **bold** : negrito normal;
- **bolder** : mais escuro que o elemento pai;
- **boldest** : mais escuro que os demais elementos;
- **100 à 900** : valores graduais múltiplos de 100 no intervalo especificado, este valor informa o qual escuro será o elemento.

Muitas vezes uso de “**minicaps**” melhora a visualmente os títulos, para atribuir este efeito ao texto fazemos uso da propriedade “**font-variant**”, que pode assumir os valores:

- **normal** : texto sem capitulação;
- **small-caps** : as letras minúsculas são convertidas em letras maiúsculas com um tamanho menor do que as maiúsculas convencionais.

Caso se deseje definir todas estas propriedades de uma vez só, devemos fazer uso da propriedade “**font**”, neste caso devemos especificar os valores das propriedades na seguintes ordem: **font-style**, **font-variant**, **font-weight**, **font-size** e **font-family**. Por exemplo:

```
h1 { font: italic small-caps bold 26px/32px Arial;}
```

**OBS: Podemos definir o valor da propriedade “line-height”, colocando uma barra e um valor após o tamanho da fonte, como visto no exemplo anterior. Se desejarmos não especificar algum valor basta remove-lo da lista da propriedade “font”.**

## Atributos de Texto

Além das opções de formatação da fonte, existem várias outras relacionadas com o texto de um documento. As próximas propriedades devem ser utilizadas quando se necessita alterar o espaçamento do texto.

A propriedade “**letter-space**”, por exemplo, especifica o espaço que deve haver entre as letras de uma palavra, para alterar este comportamento podemos utilizar:

- **valores positivos**;
- **nomal** : que substitui os valores herdados, ou
- **zero (0)** : que não inclui nem subtrai espaço, mas que evita que o texto seja justificado.

A propriedade “**word-spacing**”, por sua vez, determina como deve ser o espaçamento entre as palavras do texto, a alteração desta propriedade pode tanto atrapalhar quando facilitar a leitura do mesmo, um pequeno espaço entre as palavras pode facilitar a leitura, enquanto muito espaço pode acabar atrapalhando. Esta propriedade também aceita o valor “**normal**”.

Exemplo de uso:

No arquivo de CSS coloque:

```
.titulo { word-spacing: 8px; letter-spacing: 4px }  
p.citacao { word-spacing: 4px; letter-spacing: 1px }
```

No arquivo com o corpo do texto coloque:

...

Capítulo <span class=titulo> coleta de dados pela rede </span>

<p class=citacao> Como citado no capítulo anterior, a varredura de portas é uma das principais metodologias para coleta ativa de informações sobre os nos ativos na rede em um dado instante, mas ... </p>

E para alterar a espaçamento entre as linhas devemos utilizar a propriedade “**line-height**”, os valores que ela pode assumir são:

- **número** : indica o número pelo qual o tamanho da fonte deve ser multiplicado para que seja calculado o espaçamento;
- **valor de comprimento** : que é um tamanho independente do tamanho da fonte;
- **porcentagem** : a altura da linha é uma função do tamanho da fonte;
- **normal** : remove qualquer valor herdado.

O espaçamento das linhas podem ser aumentado para facilitar a leitura de parte do texto, principalmente quando ele ocupa uma grande região do documento. Com o mesmo objetivo partes dos texto podem ser colocadas em letras maiúscula, alterando a propriedade “**text-transform**”. Esta propriedade assume um dos seguintes valores:

- **capitalize** : toda primeira letras das palavras ficarão em maiúscula;
- **uppercase** : todas as letras ficarão em maiúscula;
- **lowercase** : todas as letras ficarão em minúsculas;
- **nome** : remove toda efeito herdado.

*OBS: Esta propriedade é muito utilizada na formatação de textos vindos de algum banco de dados para formar o corpo do documento.*

O alinhamento do texto é outro fator que podemos controlar. A propriedade “**text-align**” define as características de alinhamento horizontal do texto e seus possíveis valores são:

- **left** : alinha a esquerda o texto;
- **right** : alinha a direita o texto;
- **center** : centraliza o texto;
- **justify** : justifica o texto;

A propriedade “**vertical-align**”, por sua vez, determina como deve ser o alinhamento vertical de um determinado texto dentro de um objeto ou região do documento, esta propriedade aceita as seguintes opções:

- **super** : coloca o texto em sobrescrito;
- **sub** : coloca o texto em sub escrito;
- **baseline** : estado natural, coloca o texto na sua linha de base;
- **top** : eleva a linha de base com relação a linha de base do elemento pai 25%;
- **middle** : centraliza a linha de base com relação a linha de base do elemento pai;
- **bottom** : abaixa a linha de base com relação a linha de base do elemento pai 25%;
- **text-top** : coloca a parte superior do texto com a parte superior do elemento pai;
- **text-bottom** : coloca a parte inferior do texto com a parte inferior do elemento pai;

Exemplo de uso:

No arquivo do CSS coloque:

```
.superscript { vertical-align:super; font-size=12px; }  
.baseline   { vertical-align:baseline; font-size=12px; }  
.subscript  { vertical-align:sub; font-size=12px; }  
.top        { vertical-align:top; font-size=12px; }  
.middle     { vertical-align:middle; font-size=12px; }  
.bottom     { vertical-align:bottom; font-size=12px; }  
.texttop    { vertical-align:text-top; font-size=12px; }  
.textbottom { vertical-align:text-bottom; font-size=12px; }  
.normal     { font-weight:bold; font-size=24; }
```

No corpo do documento inclua:

```
...
<p class=normal> Texto <span class=superscript> sobrescrito </span> </p>
<p class=normal> Texto <span class=baseline> normal </span> </p>
<p class=normal> Texto <span class=subscript> subscrito </span> </p>
<p class=normal> Texto <span class=top> acima </span> </p>
<p class=normal> Texto <span class=middle> no meio da linha </span> </p>
<p class=normal> Texto <span class=bottom> abaixo </span> </p>
<p class=normal> Texto <span class=texttop> no topo </span> </p>
<p class=normal> Texto <span class=textbottom> em baixo </span> </p>
...
```

Um efeito muito comum em livros e apostilas é a indentação da primeira linha de um parágrafo, este estilo pode ser alcançado alterando a propriedade “**text-indent**”, que aceita como valor:

- um comprimento ou;
- um valor percentual em relação a largura do parágrafo e o;
- valor zero (0) pode ser utilizado para remover a tendência natural do elemento <p> de incluir espaço entre os parágrafos.

Alguns efeitos de “**decoração**” do texto podem ser conseguidos com o atributo “**text-decoration**”, que pode assumir os valores:

- **underline** (linha abaixo do texto – sublinhado);
- **overline** (linha cima do texto);
- **line-through** (linha no meio do texto);
- **blink** ( texto piscando) e;
- **none** ( remove declarações herdadas). Para aplicar mais de um efeito ao mesmo tempo e na mesma declaração, escreva seus nomes com um espaço entre eles.

Por exemplo:

```
.efeitos { text-decoration: underline overline blink; }
```

Muitas vezes desejamos manter os espaços extras que foram incluídos no arquivo *XHTML*, na visualização do documento, mas todos os *browsers* removem estes espaços, mas utilizando a atributo “**white-space**” é possível controlar este comportamento, este atributo aceita os valores:

- **pre** ( que preserva os espaços extras);
- **nowrap** (evita a quebra automática de linha que não possui a tag <break>) e;
- **normal** ( permite ao browser definir como será o tratamento).

Para forçar que uma página seja quebrada quando for impressa, devemos incluir no elemento que definirá a quebra (**style=“page-break-before: always”**).

**Obs:** este atributo não funciona quando é incluído como parte de uma regra CSS.

## Atributos de Listas e Mouse:

Dentre as alterações que podemos aplicar em um documento com CCS está a mudança do marcador utilizados pelas listas e dos indicadores do mouse.

O tipo do marcador da lista deve ser alterado utilizando-se a propriedade “**list-style-type**”, que pode assumir os valores:

- **disc** ( círculo preenchido);
- **circle** (circulo vazio);
- **square** (quadrado preenchido);
- **decimal** (número);
- **upper-roman** (numerais romanos em maiúscula);
- **lower-roman** (numerais romanos em minúsculas);
- **upper-alpha** (caracteres alfanuméricos em maiúscula) e
- **lower-alpha** (caracteres alfanuméricos em minúscula).

A definição do recuo do texto com relação aos marcadores é determinada pelo atributo “**list-style-position**”, que pode assumir os valores:

- **inside** (alinha com o marcador as linhas após a quebra) ou
- **outside** (alinha com a primeira letra as linhas após a quebra).

**OBS:** As listas que possuem um recuo normalmente se destacam mais do que aquelas não o possuem.

Caso o desenvolvedor não esteja satisfeito com os marcadores disponíveis é possível especificar uma figura para ser utilizada como marcador, esta figura deve ser do tipo: *GIF*, *JPEG* e/ou *PNG*. O atributo responsável por esta definição é “**list-style-image**”, cujo valor deve ser o parâmetro **url()**, que entre os parênteses deve possuir a localização da figura a ser utilizada como marcador. Por exemplo:

```
li { list-style-image: url (pinguim.jpg); margin-left 20px; }
```

Já quanto a aparência do mouse o único atributo que pode ser utilizado é o “**cursor**” que pode assumir um dos valores abaixo para indicar a figura a ser utilizada como ponteiro do mouse.:

- **crosshair;**
- **hand;**
- **pointer;**
- **move;**
- **n-resize;**
- **ne-resize;**
- **e-resize;**
- **se-resize;**
- **s-resize;**
- **sw-resize;**
- **w-resize;**
- **nw-resize;**
- **text;**
- **wait;**
- **help** ou
- **url()**.

## Atributos de Cor e Segundo Plano:

As propriedades do segundo plano podem ser definidas de uma única vez através do atributo **“background”**, os valores que este atributo pode assumir são: *cor do fundo, url da figura de fundo, como a figura será repetida, se a figura deve ficar fixa ou não, posicionamento da figura*. Ou podemos alterar as propriedade de forma individual através dos atributos correspondentes.

O atributo **“background-color”**, define a cor de fundo do elemento, esta cor pode ser representada através do seu valor RGB, ou através de seu nome.

**OBS:** Vários elementos aceitam o atributo **“color”** para a definição de sua cor de segundo plano.

Para controlar a imagem de fundo existem as propriedades: a **“background-image”** define a imagem a ser utilizada, esta deve ser atribuída através da sintaxe **url(localização.da.figura)**; e a **“background-repeat”** determina como a imagem será repetida para preencher todo o espaço disponível, esta ultima e pode assumir os valores:

- **repeat** (repete em todo o segundo plano);
- **repeat-x** (repete na horizontal);
- **repeat-y** (repete na vertical) ou
- **no-repeat** (gráfico será apresentado somente uma vez).

O atributo **“background-attachment”**, define como o segundo plano será tratado, os possíveis valores que podem ser utilizados nesta propriedade são:

- **fixed**: o background não rola com o restante do texto, permanecendo fixo enquanto o texto desliza pela tela;
- **scroll**: o fundo da tela acompanha o movimento do texto.

Finalmente existe o atributo **“background-position”**, que define o local onde a imagem deve aparecer, este atributo é formado por dois valores (x y) que devem ser separados por um espaço e especificam a coordenada na qual será posicionado o canto superior esquerdo da imagem. X e Y podem assumir valores percentuais, em relação ao elemento pai, o valor de comprimento e/ou as palavras: **right**, **left**, **top**, **center** e **bottom**. Sendo possível combinar os valores percentuais com os de comprimento, mas não com as palavras chaves.

## Atributos de Borda e Margem:

Todos os elementos do *XHTML* possuem quatro lados, com as seguintes propriedades:

- **Largura** e **Altura**, que são os comprimentos da base e do lado do elemento;
- **Margem** é o espaço entre a borda e os outros elementos da tela;
- **Borda**, é a linha que cerca o objeto e
- **Preenchimento**, que é o espaço entre as borda e o conteúdo do elemento.

A largura e a altura podem ser definidas através dos atributos “**width**” e “**height**” que aceitam valores de comprimento ou percentuais, quando aplicados à uma figura estes atributos alteram seu tamanho de exibição.

A propriedade “**margin**” define o espaço entre o elemento atual e os demais, ela também pode assumir valores percentuais ou de comprimento. Dependendo da resolução da tela do usuário que acessa o site, os resultados obtidos com o uso desta propriedade podem variar largamente.

É possível determinar as características individuais de cada margem alterando o valor das propriedades:

- **margin-top;**
- **margin-bottom;**
- **margin-left e**
- **margin-right.**

Para definir o formato da borda de um elemento pode-se utilizar o atributo “**border**”, que aceita dois valores, o primeiro indica a espessura da borda e o segundo seu estilo, também é possível manipular as bordas de forma individual através das propriedades:

- **border-top;**
- **border-bottom;**
- **border-left e**
- **border-right.**

O estilo da borda pode ser definido de forma individual através da propriedade “**border-style**”, que determina o desenho da borda e pode assumir os valores:

- **none** (sem borda),;
- **dotted** (pontilhado);
- **dashed** (tracejado),;
- **solid** (sólida);
- **double** (dupla);
- **groove** ou **ridge** (grossa);
- **inset** (dupla interna) e
- **outset** (dupla externa).



A cor da borda deve ser definida pela propriedade “**border-color**”, que aceita como valor tanto as cores nomeadas como seus valores *RGB*. E a espessura de uma borda pode ser definida através do uso da propriedade “**border-width**”, que aceita valores de comprimento e as palavras chaves: **thin**, **medium** e/ou **thick**, é possível determinar a espessura de cada lado da borda de forma individual através dos atributos:

- **border-top-width**;
- **border-bottom-width**;
- **border-left-width** e
- **border-right-width**.

O preenchimento dos elementos é controlado pelo atributo “**padding**”, que define o espaço entre a borda e seu conteúdo, esta propriedade aceita valores percentuais ou de comprimento, sendo possível determinar o espaçamento de forma individual através das propriedades:

- **padding-top**;
- **padding-bottom**;
- **padding-left** e
- **padding-right**.

Para que blocos de texto possam flutuar ao redor de outros elementos de texto, assim como de figuras, deve-se utilizar a propriedade “**float**”, que determina ao lado da tela no qual o elemento deve flutuar, este atributo aceita os valores:

- **right** (demais elementos ficam do lado esquerdo);
- **left** (demais elementos ficam do lado direito);
- **none** ( assume como alinhamento o alinhamento do elemento pai).

Para evitar que os elementos flutuem podemos utilizar a propriedade “**clear**”, que pode assumir os mesmos valores da anterior, assim como “**both**”.

Para a definição de como o elemento deve ser tratado quando ocorrem as quebras de linhas, deve ser utilizado o atributo “**display**”, que assume os valores:

- **list-item** (marcador de item na primeira linha e quebras de linha acima e abaixo);
- **block** ( quebra de linha acima e abaixo);
- **inline** (sem quebra de linha) e
- **none** ( impede que o elemento seja exibido).

## Atributos de Posicionamento:

Até o surgimento do CSS a única ferramenta que existia para realizar o posicionamento de elementos no navegador eram as tabelas, mas dependendo de sua complexidade elas levam muito tempo para serem carregadas, pois demoram muito para serem convertidas. Para solucionar este problema o CSS possui atributos de posicionamento que permitem um melhor controle e um carregamento mais rápido.

Para posicionar um elemento devemos lembrar que todas as janelas possuem uma largura, uma altura, uma parte superior, uma parte inferior, o lado da esquerda e da direita.

Um *browser* possui duas alturas e larguras, o primeiro conjunto é referente ao tamanho da janela do *browser*, e o segundo é referente à área útil (largura e altura vivas) do mesmo.

A origem de um elemento é sempre o canto superior esquerdo do elemento pai ou da janela. Um elemento posicionado “*absolutamente*” deve levar em consideração a área vida da janela, já o elemento posicionado relativamente deve levar em consideração o canto superior esquerdo da área na qual foi inserido.

O tipo do posicionamento deve ser definido através do atributo “***position***”, que aceita os valores:

- ***static***: um elemento com posicionamento estático (*static*) não pode ser re-posicionado explicitamente (por exemplo por scripts em javascript);
- ***relative***: os elementos que foram posicionados de forma relativa (*relative*) podem ser alinhados dentro da janela ou dentro do elemento pai e podem ser re-posicionados explicitamente;
- ***absolute***: cria um elemento independente, inserido em um ponto exato da tela, ele pode receber qualquer tipo de conteúdo HTML. O ponto de origem é sempre o canto superior esquerdo da janela ou do container de origem;
- ***fixed***: este posicionado não funciona em vários navegadores, assim como o posicionamento absoluto, cria-se um elemento independente que durante a rolagem da janela permanece na mesma posição

Para a definição da posição do elemento deve-se utilizar quatro propriedades:

- ***left*** : determina a distância do elemento à borda esquerda do container;
- ***right***: determina a distância do elemento à borda direita do container;
- ***top***: determina a distância do elemento à borda superior do container;
- ***bottom***: determina a distância do elemento à borda inferior do container;

A estas propriedades podem ser atribuídos valores de comprimento (**px, em, pt e etc**) e valores percentuais. Quando um valor negativo é utilizado o elemento será movido no sentido contrário ao especificado pela propriedade. Se o elemento for posicionado de forma relativa, a definição das margens podem fazer com que o elemento saia do seu container e sobreponha outros elementos. Caso um elemento este aninhado dentro de outro ele será movido junto com o elemento pai.

Exemplo de uso:

No código do CCS inclua:

```
.stat {position:static; font: bold 28pt courier; color: #cccccc;}  
.abs {position:absolute; top: 25px; left: 375px; width: 100px; font: bold 35pt helvetica;}  
.rel { position: relative; top: 70px; left: 25px; font: bold 12pt times;}
```

**OBS:** Para a alteração do posicionamento de um elemento de forma dinâmica pode-se utilizar linguagens de script como o **javascript** e o **vbscript**.

Sabe-se que todos os elementos de *XHTML*, por padrão, possuem apenas duas dimensões (altura e largura), mas como o *CCS* é possível incluir uma terceira, que será utilizada para empilhar elementos e criar os menus de contexto. A propriedade responsável por este efeito é a “**z-index**”, a qual pode ser atribuído um valor inteiro. A primeira camada, ou seja a camada que fica abaixo de todas, é a camada “0”, e todo o elemento que possuir o maior valor nesta propriedade deve sobrepor os demais que possuem um valor menor.

Exemplo de uso:

No código do CSS inclua:

```
#camada3 { position: absolute; z-index: 3; top: 175px; left: 255px;}  
#camada2 { position: absolute; z-index: 2; top: 100px; left: 170px;}  
#camada1 { position: absolute; z-index: 1; top: 65px; left: 85px;}  
#camada0 { position: absolute; z-index:0; top: 5px; left: 5px;}
```

# Atributos de Visibilidade

Através da CSS é possível determinar a visibilidade dos elementos do *XHTML*. A propriedade **“visibility”**, por exemplo, determina se o objeto pode ser visualizado ou não, contudo mesmo não estando visível ele ocupa seu espaço na janela, ou seja, haverá um espaço vazio onde o objeto estaria normalmente. Esta propriedade é radicalmente diferente da propriedade **“display”**, que ao ser definida como **“none”** seu elemento é removido da janela e nenhum buraco é apresentado.

A propriedade **“visibility”** pode assumir os valores:

- **hidden**: oculta o elemento, tornando-o invisível;
- **visible**: torna o elemento visível;
- **inherit**: faz com que o elemento herde a visibilidade do elemento pai.

Além de determinar se o elemento deve ser visualizado ou não, pode-se determinar qual parte do elemento deve ser visualizado, para determinar o formato e o tamanho do recorte (parte visível), deve-se fazer uso de um atributo chamado **“clip”**. Atualmente o único formato de recorte aceito é o retângulo (**rect**), que deve ser especificado a partir de quatro valores separados por espaço, o primeiro indica a distância da parte superior, o segundo é a distância da parte direita, o terceiro é a distância da parte inferior e o último é a distância da parte esquerda da área.

Exemplo de uso:

```
.recorte { position: absolute; clip: rect( 15, 350, 195, 50); top 0px, left 0px;}  
.oculto { position: relative; visibility: hidden;}
```

Muitas vezes a imagem não cabe totalmente no recorte e para gerenciar as regiões que ficaram de fora deve-se utilizar a propriedade **“overflow”**, que permite realizar a visualização das regiões inicialmente invisíveis. Esta propriedade aceita os valores:

- **scroll**: permite a inserção de barras de rolagem quando a figura é maior que o recorte;
- **hidden**: oculta o overflow, impedindo a inserção da barra de rolagem;
- **visible**: faz com que o navegador ignore o recorte exibindo todo o conteúdo da figura, inclusive o overflow;
- **auto**: permite ao navegador definir como o conteúdo extra será tratado.

Exemplo de uso:

```
.figura { width: 200px; height: 200px; overflow:auto; float:right; margin: 5px}
```

# Pseudos elementos

Existem quatro pseudos elementos associados aos links das paginas da web, estes podem ser formatados utilizando-se a sintaxe

```
a:link : formatação de um link antes de ser visitado;  
a:visited : formatação de um link após ter ser visitado;  
a:active : formatação de um link que está sendo visitado;  
a:hover : formatação do efeito de "rollover";
```

Exemplo de uso:

```
a:link { color: #ff0000; font-style: italic; }  
a.externo:link { color : 333333; }
```

Além dos links, podemos formatar de forma independente a primeira linha de um parágrafo e sua primeira letra, com os seguintes pseudo elementos:

```
p:first-line : especifica o formato da primeira linha do parágrafo;  
p:first-letter : especifica o formato do primeiro caracteres do parágrafo;
```

Exemplo de uso:

```
p { font-size: 10px; line-height: 10px }  
p:first-line { font-style: bold; font-size: 120%; }  
p:first-letter { font-size: 200%; float: left; }
```

## Unidades de medida válidas em CSS

Existem três conjuntos de unidades de medidas disponíveis em CSS: as relativas, as absolutas e as dependentes dos dispositivos. Estas unidades são descritas a seguir:

Unidade	Descrição
Relativas	
em	
porcentagem	Determina o tamanho de um elemento em relação ao tamanho do elemento pai.
Absolutas	
cm	Especifica o tamanho do elemento em centímetros
in	Especifica o tamanho do elemento em polegadas ( 1in = 2,54cm )
mm	Especifica o tamanho do elemento em milímetros
pt	Especifica o tamanho de um elemento em pontos (1pt = 1/72 in )
Dependentes do Dispositivo	
px	Especifica o tamanho de um elemento em Pixels, que é uma unidade dependente da resolução da janela onde o elemento está sendo exibido.

## Tabelas de Cores utilizadas em XHTML e CSS

O código **RGB (Red - Green - Blue)** informa a quantidade de luz vermelha, verde e azul que compõe respectivamente a cor. Este valor é representado em número hexadecimal, onde os bytes variam de 00 (ausência da cor) à FF (maior intensidade da cor), estão divididos em três grupos. Cada grupo pode variar até 256 tons da cor que ele representa. Os tons podem ser misturados com os tons de outras cores e o total de combinações possíveis é de **256 x 256 x 256 = 16.777.216**. Por exemplo:

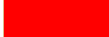













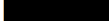

```
#FF0000 é vermelho
#00FF00 é verde
#0000FF é azul
#000000 é preto (ausência das cores)
#FFFFFF é branco (a soma de todas elas)
```

Além do código RGB, as cores também podem ser configuradas através dos 16 nomes da paleta **VGA** ou no formato da folha de estilo (CSS) em valores decimais ou porcentagens, conforme o modelo abaixo:



PADRÃO	FORMATO	SINTAXE
HTML	#RRGGBB	<FONT COLOR=#AA77CC>TEXTO</FONT>
HTML	paleta VGA	<FONT COLOR=yellow>TEXTO</FONT>
CSS	#RGB	<FONT STYLE="COLOR:#A7C">TEXTO</FONT>
CSS	RGB(x,x,x)	<FONT STYLE="COLOR:RGB(18,117,34)">TEXTO</FONT>
CSS	RGB(%,%,%)	<FONT STYLE="COLOR:RGB(20%,70%,10%)">TEXTO</FONT>

No formato *RGB(x,x,x)*, cada valor de "x" não pode ultrapassar a 256 (valor decimal) e no formato *RGB(%,%,%)* os três valores somados não podem ultrapassar os 100%, naturalmente. Nos exemplos acima foi usado como referência um texto, mas se pode aplicar os formatos em todas as propriedade e elementos relacionados à cores, tais como *background*, *tabelas*, *botões*, *links*, etc. O quadro a seguir apresenta as 16 cores da paleta VGA suportados por todos os *browsers*, com seus respectivos códigos RGB em hexadecimal.


































**OBS:** A exibição correta das cores depende da capacidade do vídeo onde serão vistas. Poucos sistemas têm capacidade de mostrar mais que 65.536 cores simultâneas. As cores podem variar de sistema para sistema, dependendo de diversos fatores de software e hardware.

PALETA VGA					
NOME PADRÃO	CÓDIGO	NOME PADRÃO	CÓDIGO		
	red	ff0000		maroon	800000
	lime	00ff00		green	008000
	blue	0000ff		navy	000080
	yellow	ffff00		olive	808000
	aqua	00ffff		teal	008080
	fuchsia	ff00ff		purple	800080
	white	ffffff		silver	c0c0c0
	black	000000		gray	808080

Os nomes das cores a seguir são definidas pelo sistema *X-Window (Unix)* e reconhecidas pelo Netscape. Qualquer outro *browser* dever ser capaz de interpretar o código hexadecimal.

NOME	CÓDIGO	NOME	CÓDIGO
 aliceblue	f0f8ff	 lightseagreen	20b2aa
 antiquewhite	faebd7	 lightskyblue	87cefa
 aquamarine	7fffd4	 lightslateblue	778899
 azure	f0ffff	 lightslategray	888888
 beige	f5f5dc	 lightsteelblue	b0c4de
 bisque	ffe4c4	 lightyellow	ffffe0
 black	000000	 limegreen	00ff00
 blanchedalmond	ffebcd	 linen	faf0e6
 blue	0000ff	 magenta	ff00ff
 blueviolet	8a2be2	 maroon	800000
 brown	a52a2a	 mediumaquamarine	66cdaa
 burlywood	deb887	 mediumblue	0000cd
 cadetblue	5f9ea0	 mediumorchid	ba55d3
 chartreuse	7fff00	 mediumpurple	9370db
 chocolate	d2691e	 mediumseagreen	3cb371
 coral	ff7f50	 mediumslateblue	7b68ee
 cornflowerblue	6495ed	 mediumspringgreen	00fa9a
 cornsilk	fff8dc	 mediumturquoise	48d1cc
 cyan	00ffff	 mediumvioletred	c71585
 darkblue	00008b	 midnightblue	191970
 darkcyan	008b8b	 mintcream	f5fffa
 darkgoldenrod	b8860b	 mistyrose	ffe4e1
 darkgray	a9a9a9	 moccasin	ffe4b5
 darkgreen	006400	 navajowhite	ffdead
 darkkhaki	bdb76b	 navy	000080
 darkmagenta	8b008b	 oldlace	fdf5e6
 darkolivegreen	556b2f	 olivedrab	6b8e23
 darkorange	ff8c00	 orange	ffa500
 darkorchid	9932cc	 orangered	ff4500
 darkred	8b0000	 orchid	da70d6
 darksalmon	e9967a	 palegoldenrod	eee8aa
 darkseagreen	8fbc8f	 palegreen	98fb98
 darkslateblue	483d8b	 paleturquoise	afeeee
 darkslategray	2f4f4f	 palevioletred	db7093
 darkturquoise	00ced1	 papayawhip	ffefd5
 darkviolet	9400d3	 peachpuff	ffdab9
 deeppink	ff1493	 peru	cd853f
 deepskyblue	00bfff	 pink	ffc0cb
 dimgray	696969	 plum	dda0dd
 dodgerblue	1e90ff	 powderblue	b0e0e6
 firebrick	b22222	 purple	800080
 floralwhite	fffaf0	 red	ff0000
 forestgreen	228b22	 rosybrown	bc8f8f
 gainsboro	dcdcdc	 royalblue	4169e1
 ghostwhite	f8f8ff	 saddlebrown	8b4513
gold	ffd700	salmon	fa8072
goldenrod	daa520	sandybrown	f4a460



	gray	7f7f7f		seagreen	2e8b57
	green	008000		seashell	fff5ee
	greenyellow	adff2f		sienna	a0522d
	honeydew	f0fff0		skyblue	87ceeb
	hotpink	ff69b4		slateblue	6a5acd
	indianred	cd5c5c		slategray	708090
	ivory	fffff0		snow	fffafa
	khaki	f0e68c		springgreen	00ff7f
	lavender	e6e6fa		steelblue	4682b4
	lavenderblush	fff0f5		tan	d2b48c
	lawngreen	7cfc00		thistle	d8bfd8
	lemonchiffon	fffacd		tomato	ff6347
	lightblue	add8e6		turquoise	40e0d0
	lightcoral	f08080		violet	ee82ee
	lightgoldenrodyellow	fafad2		wheat	f5deb3
	lightgray	d3d3d3		white	ffffff
	lightgreen	90ee90		whitesmoke	f5f5f5
	lightpink	ffb6c1		yellow	ffff00
	lightsalmon	ffa07a		yellowgreen	9acd32

# Lista de propriedades do CSS

<a href="#"><u>background</u></a>	Define a cor de fundo da página
<a href="#"><u>background-attachment</u></a>	Determina se a imagem será fixa ou não. fixed Fixa a imagem de fundo.
<a href="#"><u>background-image</u></a>	Define uma imagem de fundo na página, parágrafo ou texto. url Indica o diretório e o nome do arquivo.
<a href="#"><u>background-repeat</u></a>	Define a disposição que a imagem deverá ocupar no fundo. no-repeat É posicionada no canto superior esquerdo. repeat-x A imagem irá se repetir na horizontal. repeat-y A imagem irá se repetir na vertical.
<a href="#"><u>background-position</u></a>	Determina uma posição onde a imagem de fundo deve aparecer. bottom Alinha a imagem de background na base. center Centraliza a imagem de background. left Alinha de background à imagem a esquerda. right Alinha de background à imagem a direita. top Alinha a imagem de background no topo. centimeters (cm) Centímetros. inches (in) Polegadas. percentage (%) Porcentagem. pointd (pt) Pontos. pixels (px) Pixels.
<a href="#"><u>border</u></a>	Define uma borda que irá envolver o objeto.
<a href="#"><u>border-bottom</u></a>	Borda base.
<a href="#"><u>border-bottom-color</u></a>	Cor da borda base.
<a href="#"><u>border-bottom-style</u></a>	Estilo da borda base.
<a href="#"><u>border-bottom-width</u></a>	Largura da borda base.
<a href="#"><u>border-color</u></a>	Define a cor da borda.
<a href="#"><u>border-collapse</u></a>	Efeito Collapse. collapse Borda em alto relevo refinada.
<a href="#"><u>border-left</u></a>	Borda esquerda.
<a href="#"><u>border-left-color</u></a>	Cor da borda esquerda.
<a href="#"><u>border-left-style</u></a>	Estilo da borda esquerda.
<a href="#"><u>border-left-width</u></a>	Largura da borda esquerda.
<a href="#"><u>border-right</u></a>	Borda direita.
<a href="#"><u>border-right-color</u></a>	Cor da borda direita.
<a href="#"><u>border-right-style</u></a>	Estilo da borda direita.
<a href="#"><u>border-right-width</u></a>	Largura da borda direita.

border-style Define o estilo da borda.

solid Sólida.  
dashed Tracejada.  
dotted Pontilhada.  
double Dupla.  
groove Entalhada.  
ridge Eriçada.  
inset Baixo relevo.  
outset Alto relevo.  
none Sem borda.

border-top Borda topo.

border-top-color Cor da borda topo.

border-top-style Estilo da borda topo.

border-top-width Largura da borda topo.

border-width Especifica a espessura da borda.

thin Fina.  
medium Média.  
thick Grossa.  
px Determina a espessura em pixels.

color Especifica a cor do texto.

cursor Altera o formato do cursor.

help Interrogação.  
crosshair Cruz.  
default Cursor padrão.  
hand Mão padrão.  
move Cruz com setas.  
n-resize Seta para o norte.  
ne-resize Seta para o nordeste.  
e-resize Seta para o leste.  
se-resize Seta para o sudeste.  
s-resize Seta para o sul.  
sw-resize Seta para o sudoeste.  
w-resize Seta para o oeste.  
nw-resize Seta para o noroeste.  
text Inserção.  
wait Ampulheta.

direction Define um sentido de leitura jogando o texto à margem direita ou esquerda.

ltr Da esquerda para a direita.  
rtl Da direita para a esquerda.

filter Cria efeito visuais em objetos.

alpha Efeito de opacidade.

blur Efeito 3D.

chroma Filtra uma cor pré-definida do objeto.

dropshadow Efeito de sombreamento desfocado.

fliph Inverte a imagem do objeto horizontalmente.

flipv Inverte a imagem do objeto verticalmente.

glow Efeito de borrramento.

gray Apresenta o objeto em tons de cinza.

invert Inverte as cores do objeto.

mask Efeito transparente negativa.

motion-blur Cria um efeito de velocidade ao objeto.

shadow Efeito sombreado.

wave Efeito de ondulação.

xray Filtro de cores semelhante ao gray.

float Define o objeto como flutuante.

left Objeto à esquerda.

right Objeto à direita.

none Mostra o objeto sem alteração.

font-family Seleciona o tipo da fonte.

Arial Arial.

Bookman Old Style BookmanOldStyle.

Courier New Courier New.

MS Sans Serif MS Sans Serif.

Times New Roman Times New Roman.

Verdana Verdana.

etc, etc... E muitas outras...

font-size Muda o tamanho absoluto ou relativo da fonte.

xx-small Muito pequeno 2.

x-small Muito pequeno 1.

small Pequeno.

medium Médio.

large Grande.

x-large Muito grande 1.

xx-large Muito grande 2.

font-style Seleciona o estilo da fonte.

italic Itálico.

normal Normal.

oblique Obliquo.

font-weight Define a espessura da fonte.

extra-bold Extra negrito.

bold Negrito.

semi-bold Semi negrito.

medium Médio.

semi-light Semi leve.

light Leve.

extra-light Extra leve.

font-variant Variação de tamanho das fontes caixa alta.

normal Normal.

small-caps Caixa alta pequena.

height Define a altura do objeto.

points (pt) Pontos.

inches (in) Polegadas.

centimeters(cm) Centímetros.

pixels (px) Pixels.

letter-spacing Define um espaço entre as letras.

points (pt) Pontos.

inches (in) Polegadas.

centimeters(cm) Centímetros.

pixels (px) Pixels.

line-height Define a espessura da linha.

points (pt) Pontos.

inches (in) Polegadas.

centimeters(cm) Centímetros.

pixels (px) Pixels.

percentage (%) Porcentagem.

margin Define uma margem equidistante entre o objeto e seus quatro lados.

margin-bottom Define uma margem da base.

margin-left Define uma margem esquerda.

margin-right Define uma margem direita.

margin-top Define uma margem de topo.

points (pt) Pontos.

inches (in) Polegadas.

centimeters(cm) Centímetros.

pixels (px) Pixels.

padding Cria um espaçamento entre o objeto e as suas quatro bordas.

padding-bottom Espaço entre o objeto e a borda da base.

padding-left Espaço entre o objeto e a borda esquerda.

padding-right Espaço entre o objeto e a borda direita.

padding-top Espaço entre o objeto e a borda do topo.

points (pt) Pontos.

inches (in) Polegadas.

centimeters(cm) Centímetros.

pixels (px) Pixels.

position Posicionamento de objetos.

absolute Posicionamento absoluto.

relative Posicionamento relativo.

Static Posicionamento Estático.

text-align Especifica o alinhamento do texto.

left Pelo lado esquerdo.

center Pelo centro.

right Pelo lado direito.

justify Justificado.

text-decoration Altera o atributo do texto.

none Nenhum.

underline Sublinhado.

overline Sobrelinha.

line-through Riscado.

italic Itálico.

text-indent Cria um parágrafo no início do texto.

points (pt) Pontos.

inches (in) Polegadas.

centimeters(cm) Centímetros.

pixels (px) Pixels.

text-transform Transformação do texto.

capitalize Converte a primeira letra de cada palavra em maiúscula.

lowercase Converte todo o texto em letras minúsculas.

uppercase Converte todo o texto em letras maiúsculas.

vertical-align Alinhamento vertical de objetos.

baseline Alinhamento pela base.

bottom

middle Alinhamento pelo meio.

percentage Especifica uma porcentagem vertical no alinhamento.

sub Cria um texto sub-escrito.

super Cria um texto sobre-escrito.

text-bottom

text-top

top Alinhamento pelo topo.

white-space Gerencia como o browser deve tratar os espaços em branco.

normal Espaços normais, condição default.

pre Manterá todos os espaços inseridos.

nowrap Elimina a quebra de linha.

writing-mode Gerencia o modo da escrita.

lr-tb Escrita da esquerda para a direita, do topo para baixo.

tb-rl Escrita do topo para baixo, da direita para a esquerda.

width Define a largura do objeto.

points (pt) Pontos.

inches (in) Polegadas.

centimeters(cm) Centímetros.

pixels (px) Pixels.

word-spacing Altera o espaço entre as palavras.

points (pt) Pontos.

inches (in) Polegadas.

centimeters(cm) Centímetros.

pixels (px) Pixels.

z-index Define uma hierarquia na sobreposição dos objetos.

## Referências

- [1] Teague, Jason Craford Taegue; DHTML e CSS para World Wide Web; Editora Campus; 2001
- [2] Cascading Style Sheets – The Basic - CSS1 Recommendation; Texas Tech University; Junho de 2002
- [3] <http://orbita.starmedia.com/~edaurelio/>