

## Introdução

Se você se interessa pela criação de home pages já deve ter ouvido falar em *Cascading Style Sheets*, ou *CSS* apenas. Trata-se de um padrão de formatação para páginas web que vai além das limitações impostas pelo html. Proposto pelo *W3 Consortium* <<http://www.w3.org>> - uma espécie de comitê que define os padrões de programação para a WWW - o *CSS* foi introduzido pela primeira vez pela Microsoft, no lançamento do Internet Explorer 3.0.

O *Cascading Style Sheet* permite uma versatilidade maior na programação do layout de páginas web sem aumentar o seu tamanho em Kb, pois oferece várias possibilidades que antes só eram conseguidas com a utilização de gifs e jpgs. Basicamente, o *CSS* permite ao designer um controle maior sobre os atributos tipográficos de uma home page, como tamanho e cor das fontes, espaçamento entre linhas e caracteres, margem do texto, entre outros. Introduziu também às páginas a utilização de *layers*, permitindo a sobreposição de texto sobre texto ou texto sobre figuras.

Para visualizar inovações de layout programadas em *Cascading Style Sheet* é necessário um Internet Explorer versão 3.0 ou superior. O Netscape incorporou o *CSS* em seu código a partir da versão 4.0. Se você ainda não possui um browser 4.0, pode atualizá-lo na nossa seção de [download](#).

---

### O que Style Sheets podem fazer

Utilizando uma boa combinação de *tags* do código de programação html, como <B>, <I>, <H1> e <FONT>, podemos criar páginas interessantes sob o ponto de vista do design. Entretanto, as possibilidades que elas oferecem são bastante limitadas se comparadas às oferecidas pelo *CSS*.

Comandos de Style Sheets podem ser aplicados a **toda e qualquer tag**, modificando seus atributos. Até mesmo a tag <P> possui um </P> opcional e permite que você defina os atributos de tudo o que ficar compreendido entre as duas.

Você pode, por exemplo, trocar os atributos de uma tag <B>, que tradicionalmente adiciona negrito a uma palavra:

Este é o **negrito normal**.

Agora vamos trocar os atributos da tag para mostrar o negrito em uma cor diferente:

Este é o resultado final em *CSS*. Não esqueça de que você só vai visualizar este exemplo se estiver utilizando um browser que suporte *CSS*.

O *Style Sheets* permite a você, por exemplo, configurar todos os comandos <B> em uma página ou em um site inteiro de uma só vez.

---

### Adicionando Estilo a Sua Página

*Style Sheets* podem ser utilizados de três maneiras diferentes: local (modificando uma tag específica de uma página), geral (modificando determinados atributos para a toda a página) ou global (quando criamos um modelo que será aplicado a várias páginas simultaneamente).

#### • Adicionando estilo localmente:

*Style Sheet* pode modificar os atributos de uma única tag específica, em um determinado ponto de uma página.

Os comandos em *CSS* aplicados localmente seguem a seguinte sintaxe:

<tag STYLE="propriedade:valor; propriedade, valor;"...>

Pelo código html normal, o tamanho máximo de uma fonte que podemos obter é estipulado pela tag <FONT>. Utilizando o *CSS* podemos aumentar o tamanho com que as letras seriam tradicionalmente mostradas.

<FONT FACE="Trebuchet MS, Arial, Helvetica" SIZE="+7">PALAVRA</FONT> aparecerá assim:

**PALAVRA**

Introduzindo comandos de CSS na *tag* podemos modificar seus atributos para mostrar a frase com outra cor e em tamanho maior.

```
<FONT style="font-size:50pt; color:pink; line-height:30pt; font-family:Trebuchet MS, Arial, Helvetica;">PALAVRA</FONT>
```

, aparecerá desta maneira:

PALAVRA

Com o CSS não há limites para o tamanho da fonte. Ele pode ser definido em pontos, pixels e outras unidades. Veremos isso em detalhes mais adiante.

#### • Adicionando estilo geral:

Para criar um modelo padrão de CSS que será aplicado a toda uma página, introduzimos o comando STYLE no cabeçalho do documento, entre a *tag* HTML e a *tag* BODY. Dessa forma, tudo o que vier no conteúdo da página obedecerá aos comandos CSS estipulados uma única vez. Veja a seguir:

```
<HTML>
<TITLE> Título da página</TITLE>
<HEAD>
<STYLE type="text/css">

<!--
h1 {font: 20pt "Trebuchet MS, Arial, Helvetica"; color: red}
h2 {font: 15pt "Trebuchet MS, Arial, Helvetica"; color: blue}
p {font: 12pt "Corrier, Times"; color: black}
-- >
</STYLE>
</HEAD>
<BODY>
Conteúdo da página
</BODY>
</HTML>
```

Note que o que está dentro da *tag* STYLE aparece em comentário <!-- -->. Isso serve para evitar problemas com *browsers* que não suportam CSS. Estando entre comentários, os atributos funcionarão normalmente em browsers com capacidade para CSS e serão desprezados por *browsers* mais antigos.

#### Classes

Podemos também aplicar classes nas especificações gerais de CSS para definir diferentes atributos para uma mesma *tag*. Se por exemplo você quiser um parágrafo em vermelho para grifar textos importantes, outro em cinza para textos normais, deverá proceder desta maneira:

- Especificação geral:

```
<!--
P.importante {font: 20pt "Trebuchet MS, Arial, Helvetica"; color: red}
P.normal {font: 15pt "Trebuchet MS, Arial, Helvetica"; color: gray}
-- >
```

- No HTML os Códigos parecerão assim:

```
<P CLASS="importante"> Este texto será vermelho</P>
<P CLASS="normal">Este texto será cinza</P>
```

#### • Adicionando estilo global:

Podemos criar um único modelo de *Style Sheets* para ser aplicado a múltiplas páginas. Para tanto, devemos construir um modelo separado de *Style Sheets* e salvá-lo em um arquivo de terminação .css

Esse modelo pode ser aplicado a qualquer página, apenas referindo-se ao arquivo .css no seu cabeçalho.

A seguir um típico modelo global de CSS, definindo margens, cores, fontes e outros atributos:

```
<HTML>
<style type="text/css">
<--
body {font: 10pt "Verdana, sans-serif";
```

```

h2 {font: 50pt "Verdana, sans-serif"; font-weight: bold; color: #58F734}
h3 {font: 13pt/15pt "Verdana, sans-serif"; font-weight: bold; color: maroon; margin-left: 0.5in;
margin-top: -10px; line-height: 30px}
em {margin-top: -105px}
p {font: 10pt/12pt "Verdana, sans-serif"; color: black; margin-left: 0.5in; margin-top: -10px; line-
height: 20px}
-->
<BODY></BODY>
</HTML>

```

Para aplicar o modelo a uma página html específica, basta chamar o arquivo .css no seu cabeçalho:

```

<HEAD>

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

</HEAD>

```

Outra maneira de aplicar o modelo CSS é importá-lo. A diferença entre linká-lo a página como no exemplo acima e importá-lo é que a segunda maneira permite acrescentar estilos na própria página que está editando, sem precisar alterar o modelo global (o arquivo .css). Veja o exemplo a seguir:

```

<HTML>
<STYLE TYPE="text/css">
<!--
@import url (exemplo.css);
B { color:red; font-size:16pt }
H1 { font-family: Arial, Helvetica }
</STYLE>
<HEAD>
<TITLE>Exemplo de Style Sheets</TITLE>
</HEAD>

<BODY>
<H1>Títulos em Arial</H1>
<P>
O texto segue o modelo exemplo.css, mas os <B>negritos serão alterados</B>.
</P>
</BODY>
</HTML>

```

**IMPORTANTE:** No caso de uma mesma página ter mais de um estilo aplicado, para saber qual predominará deve-se pensar da seguinte forma: o estilo mais específico prevalece.

No caso, se uma página contém uma especificação de estilo geral ou global, você pode acrescentar novos atributos locais diferentes. Dessa forma:

- Especificações locais prevalecem sobre
- Especificações gerais que prevalecem sobre
- Especificações globais que, por sua vez, prevalecem sobre
- Especificações default dos comandos de html

## Aplicando CSS a fontes

Comandos:

- font-family
- font-size
- font-weight
- font-style

- [text-transform](#)
- [text-decoration](#)

**font-family** é o comando CSS utilizada para indicar qual a fonte que uma *tag* ou documento irá mostrar. Em uma aplicação geral de CSS a uma página, a linha de comando parecerá com esta:

```
P { font-family: Trebuchet MS, Arial, Helvetica }
```

No exemplo acima, tudo o que estiver entre as *tag* <P> e </P> no seu documento html será visualizado em *Trebuchet MS*. A razão de definir mais de uma fonte deve-se ao fato de que nem todos os computadores possuem as mesmas fontes instaladas. No caso, se o computador não possuir a *Trebuchet MS* instalada em seu hard disk, automaticamente passará para a segunda opção e mostrará o texto em *Arial*.

Deve-se observar que o nome da fonte especificadas na linha de comando do CSS deve ser exatamente igual ao nome da mesma no diretório "fontes" do computador. A fonte "*Courier*", por exemplo, leva este nome na plataforma *Mac*, mas se chama "*Courier New*" no *Windows*.

**font-size** é utilizado para definir o tamanho pelo qual as fontes serão apresentadas na tela. Há três formas básicas para definição de tamanho:

1. pontos, pixels, cm e outras unidades de medida;
2. palavras-chave;
3. percentagem

#### 1. Unidades de medida:

**Ponto** é uma medida familiar para designers, e pode ser usada para definir o tamanho de um elemento em uma página html. Tem sido a medida mais utilizada na programação em CSS pois funcionam bem em todos os *browsers* e plataformas operacionais. A única observação sobre esta medida é que as fontes costumam aparecer um pouco maiores na tela dos PCs do que nos monitores da linha *Mac*. Uma linha de comando geral em CSS parecerá assim:

```
P { font-size: 20pt } para pontos
```

ou:

```
B { font-size: 20px } para pixels, por exemplo.
```

As unidades de medida que podem ser utilizadas são:

- pt: pontos
- px: pixels
- in: inches
- cm: centímetros
- mm: milímetros
- pc: picas
- em: ems
- ex: x-weight

**Pixel** é uma medida familiar aos *web designers*. Pode dar uma boa noção do tamanho que a fonte aparecerá na tela, mas infelizmente a utilização dessa medida tem causado problemas para na impressão das páginas em papel.

Como já foi dito anteriormente, não há limites para a especificação do tamanho de uma fonte em CSS. Para quem estava acostumado a utilizar as *tag* <FONT> que permitia apenas sete tamanhos para a visualização de uma palavra, a possibilidade de aplicar qualquer tamanho através do atributo font-size será bastante útil. Compare:

- **HTML tradicional: *tag* <FONT>**

Tamanho 1

Tamanho 2

Tamanho 3

Tamanho 4

Tamanho 5

# Tamanho 6

# Tamanho 7

## • Comando Font-size:

5 Pontos (5 Pontos)

10 Pontos

15 Pontos

20 Pontos

30 Pontos

50 Pontos

100...

## 2. Palavras-chave:

Outra maneira de determinar tamanhos de fonte em CSS é utilizar palavras chave. A seguir, a lista de comandos em ordem crescente:

xx-small

x-small

small

medium

large

x-large

xx-large

Esses tamanho pré-configurados deixam o *browser* decidir qual o tamanho exato de fonte apropriado para cada palavra-chave. Por exemplo, x-large é mostrado em 28 pontos no *Netscape 4.0*, em 24 pontos no *IE 4* e em 18 pontos no *Internet Explorer 3* para Windows 95.

## 3. Percentagem:

O tamanho das fontes também pode ser determinado segundo regras de percentagem:

P { font-size: 12pt }

I { font-size: 200% }

No exemplo acima, todo o texto que estiver compreendido entre as *tags* <P> e </P> no corpo da página será mostrados em 12 pontos, assim como todas as palavras em itálico entre as duas *tags* serão apresentadas em um tamanho duas vezes maior.

**font-weight** é o comando CSS que controla a propriedade bold (negrito) de uma *tag*:

H1 { font-weight: bold }

No exemplo acima, todas as palavras da página que estiverem compreendidas entre as *tags* <H1> e </H1> serão mostradas em negrito. Da mesma forma você pode configurar "font-weight : normal" se quiser desativar todos os bolds de uma página. Podemos também especificar as características do negrito através de valores numéricos: 100, 200, 300...

400 é o valor normal de uma fonte sem negrito. 900 é o valor máximo de negrito disponível. Para uma fonte Arial, por exemplo, um browser com apenas a versão normal e a versão bold poderá mostrar a versão normal para valores de 100 à 600 e a versão bold para valores de 700 à 900.

**font-style** é o comando CSS que controla a propriedade italic de uma *tag*.

H1 { font-style: italic }

No exemplo acima o browser irá procurar uma versão itálico da fonte para apresentar o texto compreendido entre as *tags* <H1> e </H1>. Algumas fontes tem uma versão chamada "oblique" em lugar de "italic". Nesse caso, use a palavra oblique como valor para font-style.

**text-transform** serve para controlar os atributos maiúsculas e minúsculas de um texto:

P { text-transform: uppercase }

No exemplo, tudo o que estiver compreendido entre as *tags* <P> e </P> será mostrado em maiúsculas COMO NESTE EXEMPLO.

Os outros valores possíveis para o text transform são:

P { text-transform: lowercase }

Tudo entre as tags <P> e </P> será mostrado em maiúsculas como neste exemplo.

P { text-transform: capitalize }

Tudo entre as tags <P> e </P> será mostrado com as iniciais maiúsculas Como Neste Exemplo.

P { text-transform: none }

Tudo entre as tags <P> e </P> será mostrado normalmente.

Observação: O Internet Explorer 3.0 não aceita o comando text-transform.

**text-decoration** serve para controlar o sublinhado das palavras.

B { text-decoration: underline }

No exemplo acima todas as palavras em negrito da página aparecerão também sublinhadas. As opções são:

- underline: sublinha as palavras compreendidas e dentro de uma tag. Exemplo
- overline: coloca uma linha em cima da palavra compreendida entre as tags. Exemplo
- line-through: coloca uma linha em sobre o texto. Exemplo
- blink: sua palavra pisca (lembra da tag blink?). Exemplo
- none: tira o sublinhado de uma palavra (inclusive de um link). Exemplo

Observação: o atributo overline ainda não é mostrado no Netscape e o atributo blink não funciona em nenhuma das versões do Internet Explorer.

O mais interessante do comando **text-decoration** é que utilizando o atributo **none** finalmente podemos eliminar o sublinhado de um link! Se você inserir as linhas abaixo no cabeçalho de suas páginas, todos os sublinhados dos links desaparecerão. Mas não esqueça de configurá-los para serem de outra cor, ou ninguém saberá o que é e o que não é um link...

```
<style type="text/css">
```

```
<!--
```

```
A { text-decoration: none }
```

```
-->
```

```
</style>
```

## Aplicando CSS a parágrafos

Comandos:

- line-height
- text-align
- text-indent
- margin-left, top...
- padding-right, bottom...
- border-color, width...

**line-height** é o comando que controla o espaçamento entre as linhas de um texto (também conhecido pelo termo *leading*). O espaço é definido pela distância entre a base das linhas de um texto. Veja o exemplo a seguir:

P { line-height: 8pt }

"Acima da verdade estão os deuses.

A nossa ciência é uma falhada cópia

Da certeza com que eles

Sabem que há o Universo"

(...)

Os valores de line-height podem ser estipulados de três maneiras diferentes: por número, por unidade ou por porcentagem.

**Estipulando o valor em número**, o browser utilizará o font-size para definir o tamanho. Ele multiplicará o valor do font-size pelo número para definir o espaço entre as linhas. No exemplo a seguir teremos um espaçamento de 24 pontos:

```
P { font-size: 12pt; line-height: 2 }
```

(...)

"Tudo é tudo, e mais alto estão os deuses,  
Não pertence à ciência conhecê-los,  
Mas adorar devemos  
Seus vultos como às flores  
(...)"

**Estipulando o valor em unidade**, podemos definir diretamente o valor do espaçamento. As unidades que podem ser utilizadas são as mesmas explicadas para o comando **font-size**: pt, px, in, cm, mm, pc, ex, em.

**Estipulando o valor por percentagem**, o browser utilizará o valor do font-size como referência. No exemplo a seguir, ao estipularmos um valor de 60%, com uma fonte de 10 pontos, teremos um espaçamento de 6 pontos como resultado:

```
P { font-size: 10pt; line-height: 60% }
```

(...)

"Porque visíveis à nossa alta vista,  
São tão reais como reais as flores  
E no seu calmo Olimpo  
São outra Natureza."

Ricardo Reis  
(pseudônimo de Fernando Pessoa)

**text-align** é o comando utilizado para o alinhamento de parágrafos em *Style Sheets*. Ela só funciona em elementos que definam parágrafos, como <P>, <H1>, <H2>, <BLOCKQUOTE> e <UL>. Os valores que podem ser aplicados são: left (esquerda), right (direita), center (centro) e justify (justificado).

```
P { text-align: justify }
```

**Observação:** o valor justify (que está sendo utilizado neste parágrafo) funciona apenas para Netscape Communicator, em ambas as plataformas operacionais e no Internet Explorer 4 para Windows. Ele **não** funcionará no Internet Explorer 3 e 4 para Macintosh. Acrescentarei mais caracteres aqui para podermos observar melhor o efeito do text-align: nono onono non onononoo nononononon onn onononnononnon nonon onno nononon onon nononon nononon nonoon nononononon ononn ononon nononono nononon nonononononon nono.

**text-indent** é o comando que permite configurar o valor da indentação de um parágrafo. Também só funciona em elementos que definam parágrafos, como os descritos para o text-align.

```
P { text-indent: 40pt }
```

Aqui podemos verificar uma indentação de 40 pontos. Você pode definir os valores em unidades, as mesmas descritas anteriormente para o comando font-size.

```
P { text-indent: 1% }
```

Também podemos configurar a indentação em valores de percentagem, como você pode observar aplicado a este parágrafo. Uma indentação de 20% significa que o texto na primeira linha começará 20% distante de onde as outras linhas iniciam normalmente. Observação: o IE 4 assume que a percentagem refere-se ao valor total da janela do browser, não apenas a largura do parágrafo.

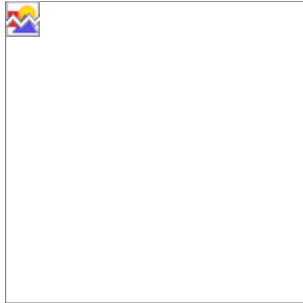
```
P { text-indent: -20px }
```

Por fim, se você configurar a indentação de um valor negativo, terá a primeira linha do parágrafo iniciando antes de onde as linhas normalmente começam, como neste exemplo. Algumas linhas a mais neste parágrafo auxiliarão a percepção do efeito ilustrado: nonono nonon nonon no no n non nonoon n nononno onnononon non noono nonnonnononoo n nooo n nonoo.

### margin-left, top...

Sabemos que em html normal só é possível acrescentar margens a um texto com a utilização de tabelas. *Style Sheets* por sua vez, permitem que tenhamos total controle das margens de um bloco de texto.

Antes de prosseguir vamos verificar um pouco das definições a respeito dos elementos de uma página:



Podemos definir a margem, borda ou *padding* de qualquer elemento do html que defina um parágrafo, como <P>, <BLOCKQUOTE>, <UL>, <H1>, <H2>...

Para controle da margem dispomos dos seguintes comandos:

margin-left (margem esquerda)  
margin-right (margem direita)  
margin-top (margem superior)  
margin-bottom (margem inferior)

Utilizando esses comandos podemos definir a margem de cada lado do elemento separadamente. As unidades de medida que podem ser utilizadas são as mesmas já discutidas anteriormente: px, pt... Também podem ser configuradas as margens utilizando valores de percentagem. Se estipularmos valores negativos para uma margem obtemos uma sobreposição, como verificamos no exemplo a seguir:

```
<P STYLE=" color:red; font-size: 24pt; font-family:'Trebuchet MS', Arial, Helvetica">PALAVRAS</P>
```

```
<P STYLE="margin-top: -45px; margin-left: 16pt; color:orange; font-size: 14pt; font-family:'Trebuchet MS', Arial, Helvetica">palavras</P>
```

PALAVRAS

palavras

Valores negativos para a margem não são a maneira mais indicada para sobrepor elementos. A diferença entre os browsers pode causar efeitos inesperados na visualização. Na última lição deste tutorial veremos como utilizar *layers* para conseguir esse efeito.

### padding-rigth, bottom...

Da mesma forma que a margem, o *padding* do elemento pode ser estipulado em unidades ou em valores de percentagem. Os comandos que permitem controlar o *padding* são:

- padding-top (*padding* superior)
- padding-bottom (*padding* inferior)
- padding-left (*padding* esquerdo)
- padding-right (*padding* direito)

Não é possível, entretanto, configurar valores negativos para o *padding* da mesma forma que fizemos com o comando que define a margem do elemento.

**Observação:** O comando *padding* não pode ser visualizado no Explorer 3.

### border-color, width....

Infelizmente os atributos de borda só podem ser visualizados no Netscape Communicator. Esperamos que nas próximas versões do Internet Explorer ele também incorpore os comando para o controle de borda.



**border-width:** define a largura da borda. Pode ser definido nas mesmas unidades explicadas para os comandos anteriores ou em palavras-chave:

: thin (borda fina)  
: medium (borda média)  
: thick (borda larga).

A borda de cada lado de um elemento pode ser definida individualmente, utilizando os comandos border-top-width (largura da borda superior), border-bottom-width (largura da borda inferior), border-left-width (largura da borda esquerda) e border-right-width (largura da borda direita)

```
<P STYLE="border-left-width: 10px; border-right-width: 5px; border-color: white">
```

Neste exemplo vemos a borda aplicada a direita e à esquerda do parágrafo (somente em Netscape Communicator).

**border-color:** define a cor da borda do elemento.

**border-style:** por fim, o comando border-style define que espécie de borda será mostrada. Aqui estão os tipos possíveis:

: solid  
: double  
: dotted  
: dashed  
: groove  
: ridge  
: inset  
: outset

## Aplicando CSS a imagens

**Comandos:**

- [color](#)
- [background-color](#)
- [background-image](#)
- [background-repeat](#)
- [background-attachment](#)
- [background-position](#)
- [background](#)

### color

Podemos aplicar cores a qualquer elemento de uma página html. Para tanto, utilizamos nomes ou números hexadecimais da mesma forma que na programação em html básico. A novidade em CSS é que podemos utilizar diretamente valores RGB para definir a cor.

Exemplo: itálicos em vermelho podem ser definidos por:

**Nome:**

```
I { color: red }
```

**Valor numérico:**

```
I { color: #FF0000 }
```

**Código RGB:**

```
I { color: rgb (255, 0, 0) }
```

O Internet Explorer 3 não é compatível com a definição em código rgb, mas ambos Netscape 4 e IE 4 aceitam.

**background-color** da mesma forma que no comando color, podemos aplicar valores numéricos, nomes ou códigos RGB para configurar uma cor de fundo de qualquer elemento da página html, incluindo imagens.

```
P.cinza { background-color: gray }
```

Neste exemplo você pode observar a cor cinza aplicada a todo o fundo deste parágrafo.

Observações:

No *IE3* e no *Netscape 4* a cor aparece apenas atrás de cada palavra, enquanto no *IE4* teremos um retângulo de cor envolvendo todo o parágrafo.

O *IE3* não aceita o comando **background-color**, mas aceita o comando **background**, que será discutido mais a frente.

### **background-image**

Com este comando podemos facilmente aplicar uma gif ou jpg atrás de qualquer elemento html.

```
<P STYLE="background-image: url (nome_da_imagem.gif)>
```

Como exemplo, aplicamos uma imagem atrás de todo este parágrafo. A figura se repete continuamente, da mesma forma que o background de uma página html.

Assim como podemos aplicar a imagem na tag <BODY> fazendo com que a figura se repita por toda a página, podemos aplicá-la a uma única palavra, somente. O valor : url (none) serve para desativar qualquer valor aplicado anteriormente ao background-image.

Observações:

O *IE3* só aceita o comando background, que será discutido posteriormente, mas não o comando background-image.

O Netscape Communicator às vezes não reage corretamente quando colocamos uma imagem em apenas parte de um parágrafo.

### **background-repeat**

Com este comando podemos definir se a imagem de fundo se repetirá continuamente ou não.

```
<P STYLE="background-repeat: no-repeat; background-image: url (nome_da_imagem.gif)>
```

O mesmo parágrafo que havíamos configurado anteriormente terá a imagem aplicada só ao canto superior esquerdo, em vez de se repetir continuamente.

Podemos também definir que as imagens se repitam só na horizontal, como executamos neste parágrafo para exemplificar, utilizando o valor **repeat-x**. Para repetições na vertical, aplicamos o valor **repeat-y**. O valor **repeat**, sozinho, serve para fazer o fundo repetir-se em ambas as direções, eliminando qualquer estilo estipulado anteriormente.

### **background-attachment**

Normalmente quando movimentamos a barra de scroll em uma página html o seu background se move junto com o conteúdo da página. Em Style Sheets podemos definir um fundo fixo para uma página. Para tal efeito, utilizamos o comando background-attachment.

```
BODY { background-attachment: fixed; background-image: url (nome_da_imagem.jpg) }
```

Infelizmente essa propriedade não funciona no Netscape Navigator. Há dois valores possíveis: fixed (fixo) e scroll (para garantir que o fundo role junto com a página).

### **background-position**

Por fim, podemos controlar exatamente onde a imagem de fundo aparecerá atrás do elemento ao qual a estamos aplicando.

```
P { background-position: right center; background-repeat: no-repeat; background-image: url (nome_da_imagem.jpg) }
```

Este comando também só pode ser visualizado no Internet Explorer. Neste parágrafo aplicamos a imagem ao centro direito do parágrafo. A primeira palavra na configuração da posição da imagem de fundo define o valor horizontal. A segunda palavra se refere ao posicionamento vertical da imagem.

Podemos definir o posicionamento da imagem de fundo em **palavras-chave**:

- right (alinhamento à direita)
- left (alinhamento à esquerda)
- center (centralizado)
- top (alinhamento superior)
- bottom (alinhamento inferior).

```
P { background-position: 50px 20px; background-repeat: no-repeat; background-image: url (nome_da_imagem.jpg) }
```

Para definir o posicionamento de maneira mais precisa podemos utilizar **unidades de medida**, como fizemos com este parágrafo. As unidades que podem ser utilizadas são as mesmas que vem sendo discutidas anteriormente: px, pt, em, cm...

Por último, também podemos utilizar **valores de percentagem** para definir o posicionamento de uma imagem atrás de um elemento do html:

```
P { background-position: 50% 80%; background-repeat: no-repeat; background-image: url (nome_da_imagem.jpg) }
```

Neste exemplo a imagem aparecerá 50% distante da margem direita, calculados pelo espaço horizontal ocupado pelo parágrafo, e a 80% do seu topo.

### background

A vantagem de utilizar este comando é podermos aplicar todos os atributos vistos até agora (cor de fundo, imagem, posicionamento) de uma única vez. Ele também permite que as páginas possam ser visualizadas no Internet Explorer 3. Para tanto, siga o seguinte modelo:

```
P { background: url (nome_da_imagem.gif) #FF0000 repeat-x 20px 30px}
```

## Utilização de Layers

### Comandos:

- [position, left, top](#)
- [width](#)
- [z-index](#)

Vamos verificar agora a maior dentre as inovações introduzidas em Style Sheets: a utilização de layers. Sabemos que posicionamento utilizando o tradicional código *html* é algo delicado. Mesmo com a utilização de tabelas, certas vezes alguns problemas de posicionamento e alinhamento podem quebrar a cabeça do mais experiente programador. Style Sheets chegou para acabar com essas limitações, como veremos a seguir.

Nota: Por terem sido criados posteriormente, os comandos *Style Sheets* para posicionamento e *layers* não podem ser visualizados com o Internet Explorer 3.

### position, left, top

Este são os comando CSS utilizados para posicionar os elementos na página html. O comando **left** determina a distância entre o elemento e a margem esquerda da página. O Comando **top** determina a distância até a parte superior da mesma.

Podemos utilizar valores de percentagem ou unidades de medida para determinar a posição do elemento através desses comandos. As unidades que podem ser utilizadas são as mesmas apresentadas anteriormente neste tutorial.

O comando **position** determina se o posicionamento dos elementos será **absoluto** ou **relativo**. Posicionamento absoluto significa que cada elemento terá sua posição determinada individualmente, através de coordenadas. Já no posicionamento relativo temos a posição de cada elemento determinada em relação aos outros.

Qualquer elemento pode ter sua posição determinada em *Style Sheets*: imagens, vídeos, párrafos ou uma única letra.

```
.exemplo1 {position: absolute; left: 200px; top: 100px}
```

Clique para ver um exemplo de [posicionamento absoluto](#).

```
<IMG STYLE="position: relative; left: 10px; top: 10px">
```

Clique para ver um exemplo de [posicionamento relativo](#).

### width

Determina largura. Esse comando só pode ser aplicado a elementos com posicionamento absoluto em uma página. O tamanho pode ser especificado em valores de percentagem ou unidades de medida.

```
.exemplo3 { position: absolute; left: 30%; top: 100px; width:200px }
```

Clique para ver um [exemplo](#) da aplicação do comando width.

### z-index

Através do posicionamento absoluto, podemos sobrepor vários elementos. O comando **z-index**, por sua vez, permite definir em que ordem eles serão visualizados.

Neste exemplo posicionamos os *layers* aparecem na ordem em que foram dispostos:

```
.layer1 { color: yellow; position: absolute; left: 210px; top: 110px }
```

```
.layer2 { color: blue; position: absolute; left: 220px; top: 120px }
```

```
.layer3 { color: green; position: absolute; left: 230px; top: 130px }
```

```
.layer4 { color: red; position: absolute; left: 240px; top: 140px }
```

Clique [aqui](#) para ver o resultado.

Com o comando **z-index** podemos alterar a ordem em que os *layers* aparecem dispostos:

```
.layer1 { color: yellow; position: absolute; left: 210px; top: 110px; z-index: 4 }
```

```
.layer2 { color: blue; position: absolute; left: 220px; top: 120px; z-index: 3 }
```

```
.layer3 { color: green; position: absolute; left: 230px; top: 130px; z-index: 1 }
```

```
.layer4 { color: red; position: absolute; left: 240px; top: 140px; z-index: 2 }
```

O comando **z-index** pode ser aplicado tanto a elementos com posicionamento absoluto quanto relativo. Quanto maior o valor, mais para cima o *layer* aparecerá. Clique [aqui](#) para ver o resultado

...

Os comandos apresentados neste tutorial fazem parte das especificações para CSS1 do [World Wide Web Consortium](#). Uma segunda versão já está aprovada. A próxima geração de browsers deverá estar capacitada a visualizar páginas elaboradas com CSS2.

Observação: Este tutorial é apenas introdutório. Alguns comandos foram excluídos em razão de serem aceitos apenas pelo Netscape Communicator, ou apenas pelo Internet Explorer 4.

Dúvidas, sugestões, críticas, escreva para [zaz@zaz.com.br](mailto:zaz@zaz.com.br)