

CSS - Artigos

Tutorial de CSS - Parte I

Por Marçal de Lima Hokama

Considerações Iniciais

Vamos mostrar aqui uma pequena noção do que é CSS, e como utilizá-lo em suas páginas. Mostramos apenas os conceitos básicos e alguns exemplos, para que o programador possa dar os primeiros passos. Depois, ele pode se aperfeiçoar indo nos sites mostrados nos nossos links. Para o pleno aproveitamento deste artigo, recomenda-se que o leitor já tenha algum conhecimento de programação em web e, principalmente, HTML.

O que é?

Cascading Style Sheets (CSS) ou **Folhas de Estilo Encadeadas**, é uma nova tecnologia (apareceu em 1996), padronizada pelo World Wide Web Consortium (a entidade que define os padrões da web), e que não é parte do HTML padrão, mas sim um conjunto de novas **tags** que ajudam a ter um melhor controle sobre o layout e o gerenciamento de nossas páginas.

Quem está usando CSS?

O CSS é suportado pelos browsers Microsoft Internet Explorer e Netscape Navigator, nas versões 4 ou posteriores; e pelo Opera, nas versões 3.50 ou posteriores. Devido às grandes vantagens advindas do uso do CSS, a maioria dos grandes sites estão usando-o em suas páginas. Se você visualizar página com CSS em browsers que não suportam esta tecnologia, você verá o conteúdo normalmente, mas sem as definições de estilo especificadas com o CSS.

Style Sheets

Um style sheet é um arquivo com comandos CSS que vão definir o estilo das suas páginas. Uma grande vantagem do uso de style sheets é que com uma mudança em um único arquivo, você pode alterar várias páginas de uma vez, que estiverem usando as definições de estilo daquele style sheet.

Primeiros passos

Para entender melhor os comandos CSS, vamos criar nosso primeiro style sheet. No seu editor HTML, crie a seguinte página.

```
<HTML>
```

```
<STYLE TYPE="text/css">
```

```
<!--
```

```
H1 {color: navy; font-size: 30px; font-family: impact}
```

```
P {text-indent: 1cm; background: yellow; font-family: arial}
```

```
-->
```

```
</STYLE>
```

```
<HEAD>
<TITLE> Meu primeiro Style sheet</TITLE>
</HEAD>
<BODY>
<H1>Meu primeiro Style Sheet</H1>
<P>Agora terei o controle total!</P>
</BODY>
</HTML>
```

Depois, abra a sua nova página no seu browser. Você deverá ver algo como abaixo:

Meu primeiro Style Sheet

Agora terei o controle total!

Se você não conseguir visualizar os estilos, é porque o seu browser não suporta CSS. Vamos entender agora um pouco do código CSS.

O núcleo do CSS são as **regras**. Um exemplo de regra seria o seguinte

```
H1{color:green}
```

Esta regra diz ao browser que todo o texto entre os tags <H1> e </H1> devem aparecer em verde.

Cada regra é composta de um **selecionador** e uma ou mais **declarações**. No exemplo acima, H1 é o *selecionador*. Ele é a tag HTML onde o estilo está sendo definido. A *declaração* define o estilo, é escrito entre chaves, e consiste de duas partes: a propriedade (no exemplo, *color*) e o valor (*green*).

Qualquer tag HTML pode ser usado como selecionador. Então, você pode adicionar informação de style sheet para qualquer tipo de elemento (texto normal, código, tabela, gráfico, etc.).

Podemos também agrupar selecionadores, para atribuí-los o mesmo estilo. Por exemplo:

```
H1, P, BLOCKQUOTE {font-family:arial}
```

Esta regra especifica que todo texto entre os tags <H1>, <P> e <BLOCKQUOTE> será exibido com a fonte Arial. Da mesma maneira, se agrupam declarações, como no exemplo a seguir:

```
P {text-indent: 1cm; background: yellow; font-family: arial}
```

As regras dos style sheets são "herdadas" de pai para filho. Veja o exemplo abaixo:

```
B: {color: navy}
```

O código acima diz ao browser que todo texto incluso com tag `` será exibido na cor "navy". E agora, veja a seguinte situação:

`Minhas páginas agora usarão <I> CSS </I> `

Não há regra definida para a tag `<I>`, mas se ela ocorrer dentro de ``, ela herdará as últimas declarações.

O que vem a seguir

Agora você já pode começar a brincar com CSS. Na próxima parte deste tutorial, mostraremos com mais detalhes como incluir style sheets nas suas páginas, e o que mais você pode fazer com CSS.

CSS - Artigos

Tutorial de CSS - Parte II

Por Marçal de Lima Hokama

Integrando os estilos às páginas

Existem 4 maneiras de incluirmos estilos CSS às páginas:

1. Incluir um style sheet no arquivo HTML.
2. Criar um link para um style sheet em outro arquivo.
3. Importar um style sheet de outro arquivo.
4. Adicionar estilos dentro dos comandos do arquivo HTML.

1- Incluir um style sheet no arquivo HTML

Neste método, o código do style sheet é incluído dentro do código da própria página, no começo do arquivo, antes do corpo (`<BODY>`) do código HTML. Veja o exemplo abaixo:

```
<HTML>
<STYLE TYPE="text/css">
<!--
H1 {font-family:impact; background: yellow;color:red}
P {color: green}
-->
</STYLE>
<HEAD>
<TITLE>Mais um Exemplo se Style Sheet</TITLE>
</HEAD>
<BODY>
<H1>Neste exemplo, estou começando a dominar</H1>
<P>O Cascading Style Sheets</P>
</BODY>
```

Quando o style sheet é incluído desta maneira, as definições colocadas ali, valem para toda a extensão daquele arquivo HTML. Este é o método mais apropriado quando queremos incluir style sheets em uma página de cada vez.

O atributo `TYPE="text/css"` quer dizer que o estilo se trata de um tipo MIME, para que os browsers que não suportam CSS ignorem o código.

Os tags de comentário (`<!--` e `-->`) também são muito importantes. Alguns browsers mais antigos (como IE 2.0 para Mac) não reconhecem o código do style sheet pelo atributo `TYPE="text/css"`, e irão mostrar os códigos style sheet como se fossem texto normal. Use comentários, e isto não irá acontecer.

2- Criar um link para um style sheet em outro arquivo

Com este método, você pode fazer com que várias páginas HTML usem style sheets definidos em um único arquivo central. A grande vantagem deste método é que, se você quiser fazer uma mudança nos estilos da sua página, você só precisa alterar em um único local. Imagine a comodidade para o webmaster de um site de + de 50 páginas?

Para criar um link, deve-se substituir a tag `<STYLE>` pela tag `<LINK>`, e colocar a declaração de estilo dentro da porção `<HEAD>` do documento. E neste método, não se precisa usar as tags de comentário. Vamos criar o arquivo abaixo:

```
<HTML>
<HEAD>
<TITLE> Mais um exemplo de style sheet </TITLE>
<LINK REL=stylesheet HREF="def_estilos.css" TYPE="text/css">
</HEAD>
<BODY>
<H1>Neste exemplo, estou começando a dominar</H1>
<P>O Cascading Style Sheets</P>
</BODY>
```

Agora vamos criar o arquivo de definição de estilos (no caso, "def_estilos.css"), com o qual o documento anterior possui um link:

```
H1 {font-family:impact; background: yellow;color:red}
P {color: green}
```

Na hora da visualização do arquivo HTML, as definições do arquivo no link serão carregadas e usadas no código da página.

3- Importar um style sheet de outro arquivo

A importação de um style sheet externo é parecida com o método anterior de criações de link. A diferença é que você não pode combinar o método de link com outros métodos de inserções de estilos, mas a importação pode ser combinada. Vejamos os exemplos dos arquivos HTML e o arquivo de definições.

```
<HTML>
<STYLE TYPE="text/css">
<!--
@import url(def_estilos.css)
P {color: red}
-->
</STYLE>
<HEAD>
<TITLE>Mais um Exemplo se Style Sheet</TITLE>
```

```
</HEAD>
<BODY>
<H1>Neste exemplo, estou começando a dominar</H1>
<P>O Cascading Style Sheets</P>
</BODY>
```

No exemplo acima, o browser importa primeiro as definições no arquivo .css, e adiciona as regras internas para serem usadas por toda a página.

No mesmo exemplo, veja que **P** tem uma regra tanto no arquivo externo como nos estilos embutidos. Neste caso, sempre será usado o estilo embutido, em detrimento da definição externa. O resultado seria que o texto que estiver dentro das tags **P**, seriam mostrada em vermelho.

Obs.: apenas o browser IE4 suporta importação.

4- Adicionar estilos dentro dos comandos HTML

Você também pode adicionar definições de estilo dentro do corpo de seu código HTML, através do atributo **STYLE**. Veja o exemplo:

```
<HTML>
<HEAD>
<TITLE>Mais um Exemplo se Style Sheet</TITLE>
</HEAD>
<BODY>
<H1 STYLE="font-family:impact; background: yellow;color:red">Neste exemplo, estou
começando a dominar</H1>
<P STYLE="color: green">O Cascading Style Sheets</P>
</BODY>
```

Uma observação a se fazer aqui é que a definição só vale na linha em que foi definida. Seguindo o exemplo acima, se tivéssemos mais um texto **H1**, sem o atributo **STYLE**, ele usaria as definições padrões do browser.

Ordem de prioridade de estilos

Ao utilizar style sheets, é importante saber a ordem que os browsers utilizam na definição de estilos:

1. Estilos incluídos na linha do comando
2. Estilos definidos no início da página
3. Estilos obtidos de um arquivo externo (via link)
4. Estilos importados de um arquivo externo
5. Estilos padrões do browser

Portanto, se tivéssemos o seguinte código

```
<HTML>
<STYLE TYPE="text/css">
<!--
P {color: red}
-->
</STYLE>
<HEAD>
```

```
<TITLE>Mais um Exemplo se Style Sheet</TITLE>
</HEAD>
<BODY>
<P STYLE="color: green">Cascading Style Sheets</P>
</BODY>
```

Observe que a tag **P** tem estilos definidos no início da página e na linha de comando. Com qual cor o texto irá aparecer? Verde ou vermelho? Seguindo a ordem de prioridades mostrada anteriormente, os estilos na linha de comando têm precedência. Logo, a cor do texto será verde.

Classes

Vamos agora complicar um pouco. Imagine que, por exemplo, você queira um texto de cor vermelha no primeiro parágrafo, verde no segundo parágrafo e cinza para o terceiro.

Esse é o uso típico de classes. Você pode criar três classes diferentes de P, cada uma com uma definição diferente de estilo. Veja o exemplo:

```
<HTML>
<STYLE TYPE="text/css">
<!--
P.first {color: red}
P.second {color: green}
P.third {color: gray}
-->
</STYLE>
<HEAD>
<TITLE>Mais um Exemplo se Style Sheet</TITLE>
</HEAD>
<BODY>
<P CLASS="first">Primeiro parágrafo</P>
<P CLASS="second">Segundo parágrafo</P>
<P CLASS="third">Terceiro parágrafo</P>
</BODY>
```

O nome da classe, pode ser qualquer nome, e identificada pelo "." também pode-se associar classes a mais de uma tag HTML, como:

```
P {font-family: Verdana}
H1 {font-family: Arial; font-size: 34 pt}
.cor_padrao {color: green}
```

Com esta definição, podemos usar **CLASS="cor_padrao"** dentro das tags **P** e **H1** no corpo da página. Exemplo:

```
<P CLASS="cor_padrao"> Este é um texto padrão .</P>
```

Também pode-se criar classes sem associá-las a uma tag HTML. Por exemplo:

```
.texto_padrao {color:green;background: yellow}
```

Agora, qualquer tag pode usar esta definição, bastando especificar **CLASS="texto_padrao"** no corpo da página.

Selecionadores de ID

Funcionam de maneira semelhante às classes. Primeiro, define-se a regra para um determinado id utilizando o "#":

```
#texto_padrao {font-family: verdana; font-size: 12 pt}
```

E depois usamos o estilo da seguinte maneira:

```
<P ID="texto_padrao">Este é um texto padrão .</P>
```

Selecionadores de Contexto

Se você quer que todo o texto em negrito seja vermelho, mas apenas se o texto em negrito ocorrer no corpo de texto. Selecionadores de contexto são selecionadores que dependem que uma certa situação seja verdadeira para que suas declarações sejam efetivadas:

A regra no exemplo diz ao browser que todo texto em negrito será azul, apenas se aparecer dentro da tag P.

```
P B {color:navy}
```

Para visualizar o resultado desta regra, digite o texto HTML abaixo e experimente no seu browser:

```
<P> Este é um texto <B>padrão</B></P>
```

Colocando comentários nas páginas

Dentro do código CSS, você também pode colocar comentários entre os símbolos /* e */. Veja o exemplo:

```
H1 {color:red} /* linhas vermelhas para a tag H1 */
```

Por Marçal de Lima Hokama

Nesta parte de nosso tutorial, veremos as pseudo-classes e pseudo-elementos, com atenção especial às pseudo-classes da tag [A](#), que irão nos permitir criar efeitos interessantes com links.

Pseudo-classes e Pseudo-elementos

Existem algumas classes e elementos especiais que são reconhecidos pelos browsers que suportam CSS, chamadas de "pseudo-classes" e "pseudo-elementos". Como exemplo de pseudo-classes podemos mostrar algumas da tag [A](#). Veja o código a seguir:

```
<HTML>
<STYLE TYPE="text/css">
<!--
A:link {color:red}
A:visited {color:navy}
A:hover {color:yellow}
-->
</STYLE>
<BODY>
<P ALIGN="CENTER"> <A HREF="TEST.HTM">TESTE</A></P>
</BODY>
</HTML>
```

O elemento [A](#) possui várias pseudo-classes definidas, como *:link*, *:visited* e *:hover*. O código acima quer dizer que qualquer link é mostrado inicialmente com o texto na cor vermelha (*red*), os links que já foram visitados na cor azul (*navy*), e quando o cursor do mouse estiver sobre o link, o texto é visualizado na cor amarela (*yellow*). Veremos mais sobre o elemento [A](#) e suas pseudo-classes mais adiante. As pseudo-classes se referem a diferentes estilos de um elemento. Regras com pseudo-classes têm a seguinte sintaxe:

selecionador: pseudo-classe {propriedade:valor}

Os pseudo-elementos fazem referência a sub-partes de um elemento, como por exemplo, a primeira letra de um parágrafo. Eles são referenciados da seguinte maneira:

selecionador: pseudo-elemento {propriedade:valor}

Não iremos tratar com profundidade os pseudo-elementos, pois eles não foram implementados nem no Internet Explorer nem no Netscape Navigator.

Pseudo-classes do elemento A

Usando a tag [A](#), e suas pseudo-classes, podemos criar muitos efeitos interessantes, principalmente o do link mudar de estilo quando o mouse passa sobre ele. Vamos ver as pseudo-classes da tag [A](#):

| Pseudo-classe | Uso |
|---------------|---|
| :link | é aplicada para links que ainda não foram visitadas. |
| :visited | é aplicada para links que já foram visitadas pelo usuário. |
| :hover | é aplicada quando o cursor do mouse está sobre o link. |
| :active | é aplicada quando um elemento está sendo ativado pelo usuário. Por exemplo, entre o tempo em que o usuário pressiona o botão do mouse e depois solta-o. |

[:link](#) e [:visited](#) são mutuamente exclusivas. [:hover](#) não funciona no Netscape Navigator. Vamos ver alguns exemplos das pseudo-classes.

```
<HTML>
<STYLE TYPE="text/css">
<!--
A:link {color:red}
A:visited {color:navy}
-->
</STYLE>
<BODY>
<P ALIGN="CENTER"> <A HREF="TEST.HTM">TESTE</A></P>
</BODY>
</HTML>
```

Este código irá mostrar um link com cor vermelha se não tiver sido visitado. Senão será mostrado com a cor azul.

Vamos ver outro código:

```
<HTML>
<STYLE TYPE="text/css">
<!--
A:link {color:red}
A:visited {color:yellow}
A:hover {color:navy;text-decoration:underline}
A:active {color:lime}
-->
</STYLE>
<BODY>
<P ALIGN="CENTER"> <A HREF="TEST.HTM">TESTE</A></P>
</BODY>
</HTML>
```

Se o link não tiver sido visitado, a palavra acima será mostrada na cor vermelha. Se tiver sido, será exibida na cor amarela. Quando o cursor do mouse passar sobre a palavra, esta ficará com a cor azul e em sublinhado. Se apertar o botão do mouse sobre o link, esta ficará com uma cor verde-limão até que a nova página seja aberta.

Você pode aplicar nas pseudo-classes qualquer formatação de fontes (as diretivas de fontes serão vistas com mais detalhes adiante): alterar tamanho, fonte, cor, fundo, etc.

CSS - Artigos

Tutorial de CSS - Parte IV

Por Marçal de Lima Hokama

Na quarta parte de nosso tutorial, veremos como trabalhar com as propriedades de background.

Background

Você já se perguntou como é que as páginas hospedadas no Geocities têm sempre aquele "G" no canto inferior direito da tela? Ou como se coloca uma única imagem de fundo no centro da tela? Esses são alguns dos efeitos que podemos conseguir com as propriedades de background.

background-attachment

Importante: esta propriedade não funciona no Netscape Navigator.

Esta propriedade determina se uma imagem de background será posicionada de maneira fixa no browser ou se essa imagem se movimentará na medida em que a janela é rolada. Existem dois valores possíveis para essa propriedade:

| Valor | Descrição |
|--------|---|
| fixed | A imagem é fixa na janela. |
| scroll | A imagem é movimentada na medida em que o usuário se movimenta na janela. |

Veja a seguir um exemplo com `background-attachment: fixed`.

```
<head>
<style TYPE="text/css">
<!--
BODY {background-attachment: fixed; background-image:
url(images/Marcal.gif);background-repeat: no-repeat;
background-position:center 150}
-->
</style>
</head>
```

O resultado deste código você verá na página de exemplo ([clique aqui](#)). As outras propriedades de background serão vistas adiante.

Na próxima página de exemplo veremos como funciona o `background-attachment: scroll` ([clique aqui](#)).

background-color

Aqui se escolhe a cor de fundo do elemento que está usando esta propriedade. Você pode usar um nome de cor válido (ex.: *blue*, *red*, etc.) ou um valor RGB como #808080 (branco). Veja o exemplo abaixo:

```
<html>
<head>
<style TYPE="text/css">
<!--
H2 {background-color: red}
-->
</style>
</head>
<body>
<h2>Teste de cor de fundo para um cabeçalho</h2>
</body>
```

Veja o resultado:

Teste de cor de fundo para um cabeçalho

background-image

Definimos nesta propriedade um arquivo gráfico que será usado como fundo para os elementos da página. O arquivo pode ser um gráfico .GIF ou .JPEG. Isto é definido da seguinte forma:

selecionador {background-image: URL(endereçodaimagem)}

O endereço pode ser um endereço completo ou relativo. Veja abaixo um exemplo de definição de uma imagem de fundo para um cabeçalho.

```
<html>
<head>
<style TYPE="text/css">
<!--
H1 {background-image: URL(images/Marcal.gif)}
-->
</style>
</head>
<body>
<h1>Teste de imagem de fundo para um cabeçalho</h1>
</body>
```

E agora um exemplo de definição de imagem de fundo para o corpo da página:

*BODY {background-image:
URL(<http://members.tripod.com/marcalhokama/images/Marcal.gif>)}*

Os browsers às vezes criam problemas quando usamos background-image em um parágrafo. Não mostram a imagem corretamente, ou podem adicionar uma linha antes e depois do texto que tem a imagem embutida.

background-position

Importante: esta propriedade não funciona no Netscape Navigator.

Com esta propriedade, podemos posicionar a imagem de background de maneira bem eficiente. A posição é relativa ao canto superior esquerdo do selecionador. A propriedade é definida da seguinte forma:

selecionador {background-position: x y}

Sendo x a posição horizontal e y a posição vertical da imagem. Existem três modos de especificar estas posições:

1-Valores-chave:

| Valor | Descrição |
|--------|---|
| top | Alinha a imagem com o topo do elemento selecionador. Só vale para substituir y. |
| left | Alinha a imagem com o canto esquerdo do elemento selecionador. Só vale para substituir x. |
| right | Alinha a imagem com o canto direito do selecionador. Só é válido na substituição de x. |
| bottom | Alinha a imagem com a parte inferior do selecionador. Só vale para y. |
| center | Centraliza a imagem dentro do selecionador; quando usado em x, centraliza horizontalmente, e em y, verticalmente. |

P {background-position: left top;background-repeat:repeat-y;background-image:url(background.gif)}

Nesta definição, a imagem será posicionada no canto superior esquerdo do parágrafo.

2-Valores numéricos:

Você coloca os valores com mais precisão. Por exemplo:

P {background-position: 70px 10px;background-repeat:repeat-y;background-image:url(background.gif)}

Isso quer dizer que a imagem será colocada 70 pixels depois do canto esquerdo e 10 pixels abaixo do canto superior do parágrafo.

3-Valores em porcentagem

Outra maneira de posicionar a imagem é usar valores em porcentagem. Exemplo:

P {background-position: 75% 50%;background-image: url(background.gif)}

Com esta declaração, a imagem de background terá seu ponto central desenhado a 75% da largura total do parágrafo e a 50% da altura do parágrafo. Veja o resultado abaixo:

Este é um teste de parágrafo com valores em porcentagem. Veja que a imagem está posicionada a 75% do total horizontal e a 50% do total vertical. Legal, não! Aproveite este momento para testar nas suas páginas.

background-repeat

Com esta propriedade, você pode controlar a taxa de preenchimento da tela com a imagem de background. Pode-se atribuir um dos quatro valores a esta propriedades:

| Valor | Descrição |
|-----------|--|
| repeat | Repete a imagem horizontalmente e verticalmente. |
| repeat-x | Repete a imagem apenas horizontalmente. |
| repeat-y | Repete a imagem apenas verticalmente. |
| no-repeat | Não repete a imagem. |

Vamos ver um código de exemplo:

```
<head>
<style TYPE="text/css">
<!--
BODY {background-attachment: fixed; background-image:
url(images/Marcal.gif);background-repeat: no-repeat;
background-position:center 150}
-->
</style>
</head>
```

O resultado deste código você verá na página de exemplo ([clique aqui](#)).Veja que a imagem de fundo só é mostrada uma única vez.

CSS - Artigos

Tutorial de CSS - Parte V

Por Marçal de Lima Hokama

Propriedades das Fontes

Veremos agora como alterar as propriedades de uma fonte usando CSS. Alguns exemplos você já deve ter visto nas partes anteriores do nosso tutorial. Mas aqui trataremos com mais profundidade sobre o assunto.

font-family

Nesta propriedades determinamos a lista das fontes (pode ser de uma ou mais fontes) a serem usadas para exibir o texto. Veja o exemplo abaixo:

```
P {font-family: impact, times, serif}
```

Isso quer dizer o seguinte para o browser: verifique se a primeira fonte na lista (impact) está instalada no seu computador. Se estiver, use-a. Se não, passe a verificação para a segunda fonte na lista, e assim por diante. Uma coisa a verificar quando você está usando CSS para a definição da fonte do seu texto é que, o texto só

vai aparecer do jeito que você quer se no computador da pessoa que estiver vendo a sua página também tiver a fonte instalada. Por isso, tente usar sempre as fontes mais usuais (Times New Roman, Arial, etc.), que garantirá a visualização correta na maioria dos browsers dos internautas.

Ao invés de se usar o nome da fonte, podemos usar nomes genéricos, o que aumenta a compatibilidade entre as plataformas. A tabela abaixo mostra os nomes de fontes genéricas e o nome da fonte com que se parecem.

| Nome genérico | Similar a |
|---------------|-----------------|
| serif | Times New Roman |
| sans-serif | Arial |
| cursive | Script |
| fantasy | Comic |
| monospace | Courier New |

Importante: O Netscape Communicator não suporta as fontes genéricas *cursive* e *fantasy*.

É interessante que, na sua lista de nomes de fontes, sempre use como última opção, um nome de fonte genérico, porque se não for encontrado no computador do navegador as fontes anteriores, essa fonte com certeza ele encontrará.

Se você usar um nome de fonte com mais de uma palavra, você deve colocar este nome entre aspas. Por exemplo:

`P{font-family:"courrier new", "new baskerville", serif}`

Quando você colocar estilos *inline*, você deve usar o nome da fonte entre apóstofos. Por exemplo:

`<P STYLE="font-family: 'gill sans', 'new baskerville', serif">Texto de exemplo.</P>`

font-size

Esta propriedade determina o tamanho da fonte em pixels (px), pontos (pt), polegadas (in), centímetros (cm), etc. Você também pode usar porcentagem ou um dos valores da tabela abaixo:

| Valor | Descrição |
|----------|--|
| xx-small | 50% menor que a fonte x-small |
| x-small | 50% menor que a fonte small |
| small | 50% menor que a fonte medium |
| medium | Um tamanho de fonte de aproximadamente 10 pontos |
| large | 50% maior que a fonte medium |
| x-large | 50% maior que a fonte large |
| xx-large | 50% maior que a fonte x-large |
| larger | 50% maior que a fonte do elemento pai |

| | |
|---------|---------------------------------------|
| smaller | 50% menor que a fonte do elemento pai |
|---------|---------------------------------------|

Algumas unidades de tamanho interessantes

1. Points

Veja o exemplo abaixo:

P {font-size: 16pt}

Este código diz ao browser para mostrar o texto da tag <P> num tamanho de 16 pontos. O tamanho de ponto é uma unidade que se refere a uma caixa imaginária que se estende do fundo de uma letra descendente (como um "p", que tem uma parte para baixo) até o topo de uma letra ascendente (como um "d", que tem uma parte para cima). Pontos são uma unidade excelente para se usar em vários browsers e plataformas. A única coisa que se deve cuidar é que as fontes aparecem maiores no monitor de PC do que no monitor de um Mac.

2. Em

Esta unidade é igual ao tamanho em pontos de uma fonte. No nosso caso, ela se refere ao tamanho do "elemento pai". Veja o exemplo a seguir.

P {font-size: 10pt}
B {font-size: 1.5em}

Neste caso, qualquer texto na tag inclusa numa tag <P> terá um tamanho de 15 pontos (1.5 x 10pt).

Valores em porcentagem

Veja o exemplo abaixo:

P {font-size: 10pt}
B {font-size: 100%}

O funcionamento aqui é parecido com o da unidade *em* (visto acima). Todo o texto incluso na tag que estiver dentro de uma tag <P> será mostrada num tamanho 100% maior que o tamanho definido em <P>, ou seja 20pt. Valores em porcentagem sempre são usados com referência ao tamanho de um "objeto pai".

font-style

Esta propriedade altera o estilo da fonte. A tabela abaixo mostra os valores possíveis:

| Valor | Descrição |
|---------|------------------|
| normal | Estilo normal |
| oblique | Fonte oblíqua |
| italic | Fonte em itálico |

Importante: O Netscape Communicator não suporta o valor *oblique*.

Veja o exemplo abaixo:

```
<html>
<head>
<style TYPE="text/css">
<!--
H3 {font-size:50pt;font-style:italic}
H2 {font-size:50pt;font-style:oblique}
-->
</style>
</head>
<body>
<H2>Exemplo de texto oblíquo</H2>
<H3>Exemplo de texto em itálico </H3>
</body>
</html>
```

Quando o browser encontra alguma fonte em itálico, ele verifica se há uma versão em itálico daquela fonte instalada na máquina do navegador. Se não encontrar, ele irá fazer uma inclinação na versão normal da fonte.

font-weight

Com esta propriedade determinamos a espessura da fonte. Você pode especificar os seguintes valores: normal, bold, bolder ou lighter nesta propriedade (o valor *lighter* não é suportado no Netscape Communicator). Também pode-se aplicar valores numéricos de 100, 200,...,900. O texto *normal* tem um valor de 400. Cada número maior que 400 representa um nível maior de espessura, sendo 900 o maior disponível. Para os números menores que 400, temos um nível menor de espessura, chegando até 100.

Veja o exemplo abaixo:

```
<html>
<head>
<style TYPE="text/css">
<!--
H3 {font-size:20pt;font-weight:900}
H2 {font-size:20pt;font-weight:100}
H1 {font-size:20pt;font-weight:400}
-->
</style>
</head>
<body>
<H1>Exemplo de fonte normal</H1>
<H2>Exemplo de fonte fina</H2>
<H3>Exemplo de fonte grossa </H3>
</body>
</html>
```

Os valores *lighter* e *bolder* só funcionam de você especificar a propriedade *font-weight* para um elemento que já tiver algum nível de espessura especificado. Por exemplo, se você aplicar *bolder* para um elemento que já está herdando uma espessura de outro elemento, então o browser irá tentar fazer com que a fonte fique com um negrito mais destacado.

Às vezes, se não existir uma versão em negrito da fonte, não irá dar para aplicar uma versão mais espessa ou menos espessa do negrito.

font-variant

Esta propriedade mostra o texto em letras minúsculas. Existem dois valores possíveis para ela: *normal* ou *small-caps*. Com *normal*, o texto é mostrado normalmente. Já com *small-caps*, o browser mostra todo o texto com letras minúsculas. A propriedade não é suportada no Netscape Communicator.

CSS - Artigos

Tutorial de CSS - Parte VI

Por Marçal de Lima Hokama

Propriedades de Texto

Nesta parte do tutorial aprenderemos como trabalhar com as propriedades de texto.

letter-spacing

Esta propriedade determina o espaço entre as letras de um texto. **Esta propriedade não funciona no Netscape Navigator.**

Veja o exemplo abaixo:

```
<html>
<head>
<style TYPE="text/css">
<!--
H3 { letter-spacing: 10px }
-->
</style>
</head>
<body>
<H3>Exemplo de texto espaçado </H3>
</body>
</html>
```

Veja o resultado abaixo:

E x e m p l o d e t e x t o e s p a ç a d o

Nesse caso cada caracter aparecerá com um espaço de 10 pixels entre eles. As unidades de medida usadas aqui são as mesmas vistas no [capítulo V](#) deste tutorial.

line-height

Você pode determinar o espaçamento entre linhas de um parágrafo com esta propriedade.

```
<html>
<head>
<style TYPE="text/css">
<!--
H2 {line-height: 30pt}
-->
</style>
</head>
<body>
<H3>Espaçamento normal</H3>
<H3>Espaçamento normal</H3>
<H2>Espaçamento maior</H3>
</body>
</html>
```

E o resultado segue abaixo:

Espaçamento normal

Espaçamento normal

Espaçamento maior

Existem 3 maneiras de dar um valor para **line-height**: número multiplicador, com unidade de comprimento ou porcentagem.

Número multiplicador

H3 {font-size: 12pt; line-height: 2}

Quando você usa um fator multiplicador, o browser se utiliza do parâmetro **font-size** para obter o valor final: ele multiplica font-size pelo número. Neste exemplo, line-height será de 24 pontos.

Com unidade de comprimento

H3 {font-size: 12pt; line-height: 12pt}

Como você viu anteriormente, será utilizado o valor com a unidade de medida.

Porcentagem

H3 {font-size: 12pt; line-height: 50%}

Desta maneira, o valor final de line-height será o valor da porcentagem sobre o valor de font-size. No exemplo, será 50 % de 12pt, ou seja, 6 pt.

text-align

Esta propriedade nos permite controlar o alinhamento horizontal dos parágrafos. Exemplo:

H2 {text-align: right}

Os valores possíveis para esta propriedade são:

| Valor | Resultado |
|---------|------------------------|
| left | alinhamento à esquerda |
| right | alinhamento à direita |
| center | texto centralizado |
| justify | texto justificado |

text-indent

Com esta propriedade, controlamos a indentação de um parágrafo. Veja o exemplo:

```
<html>
<head>
<style TYPE="text/css">
<!--
H3 {text-indent: 50pt}
-->
</style>
</head>
<body>
<H3>Veja a indentação deste parágrafo!</H3>
</body>
</html>
```

O parágrafo será mostrado da seguinte maneira:

Veja a indentação deste parágrafo!

Você também pode definir um recuo com valores negativos.

text-transform

Se você quiser que todo o seu texto fique em maiúsculas, minúsculas ou com a primeira letra de cada palavra em maiúscula, esta é a propriedade que você deve usar. Os valores possíveis são:

| Valor | Efeito |
|------------|---|
| uppercase | todas as letras do texto serão em maiúsculas |
| lowercase | todas as letras do texto serão em minúsculas |
| capitalize | a primeira letra de cada palavra estará em maiúscula |
| none | qualquer um dos valores acima herdados serão ignorados. |

Veja o exemplo abaixo:

P {text-transform:capitalize}

Isso quer dizer que cada palavra que estiver no parágrafo terá a sua primeira letra em maiúscula.

text-decoration

Mais alguns efeitos podem ser conseguidos com esta propriedade. Veja os valores possíveis na tabela a seguir:

| Valor | Efeito |
|--------------|-------------------------------------|
| underline | texto sublinhado |
| overline | adiciona uma linha sobre o texto |
| line-through | adiciona uma linha cortando o texto |
| blink | texto piscante |
| none | elimina todos os efeitos anteriores |

Observação importante: `overline` não funciona no Netscape Navigator e `blink` não funciona no Internet Explorer.

Veja a seguir um exemplo:

```
<html>
<head>
<style TYPE="text/css">
<!--
B {text-decoration: underline}
-->
</style>
</head>
<body>
<H3>Veja o efeito na palavra <B>"olá!" </B></H3>
</body>
</html>
```

CSS - Artigos

Tutorial de CSS - Parte VII

Por Marçal de Lima Hokama

Propriedades de Caixa

Usando CSS, podemos manipular com eficiência as bordas, margens e espaçamento interno ("*padding*") dos elementos da sua página. As três propriedades citadas anteriormente, junto com o elemento da página em si, estão contidas dentro de uma caixa ("*box*") imaginária.

Propriedades de borda

Com as propriedades "border", podemos definir as bordas esquerda, direita, superior e inferior de um elemento. Os atributos definíveis são a espessura, cor e estilo da borda.

- **border-top-width, border-bottom-width, border-left-width, border-right-width, border-width**

Atenção : Estas propriedades de borda só funcionam no Internet Explorer se também estiver definida a propriedade **border-style** (que veremos mais adiante), o resultado só será visto corretamente no Netscape Navigator.

Estas propriedades definem a espessura da borda, de acordo com a tabela abaixo:

| propriedade | descrição |
|---------------------|-----------------------------|
| border-top-width | espessura da borda superior |
| border-bottom-width | espessura da borda inferior |
| border-left-width | espessura da borda esquerda |
| border-right-width | espessura da borda direita |

E para estas propriedades, podem ser atribuídos os seguintes valores:

| valor | descrição |
|----------------|---|
| thin | Usa uma borda de espessura fina |
| medium | Usa uma borda de espessura média |
| thick | Usa uma borda de espessura grossa |
| valor numérico | Define a espessura exata da borda, usando como unidade points (pt), polegadas (in), centímetros (cm), ou pixels (px). |

Vamos ver um exemplo a seguir:

```
<html>
<head>
<style TYPE="text/css">
<!--
H3 { border-top-width: 2px; border-bottom-width: 5px; border-left-
width: 1px; border-right-width:1 px }
-->
</style>
</head>
<body>
<H3>Exemplo de texto com borda </H3>
</body>
</html>
```

(Este exemplo só será visualizado corretamente no Netscape Navigator)

Você não precisa definir uma borda para todos os lados do elemento. Pode-se definir, por exemplo, somente a borda inferior, ou só a borda superior, ou somente as duas bordas.

Atenção : Este modo de definir bordas para apenas alguns lados do elemento não serve para o Internet Explorer (somente para o Netscape Navigator). Para o Explorer, devemos definir para os lados que não deverão aparecer o valor `none` para a propriedade `border-style` (veja esta propriedade mais adiante).

Se quisermos que todas as bordas fiquem com a mesma espessura, podemos utilizar a propriedade `border-width`, para não ter que definir cada borda do elemento. Na definição de H3 no exemplo anterior, poderíamos modificá-lo para:

```
H3 { border-width: 2px }
```

E o resultado ficará:

Exemplo de texto com borda

(Este exemplo só será visualizado corretamente no Netscape Navigator)

- **border-color**

Esta propriedade define a cor da borda de um elemento. Você pode usar um nome de cor, como `YELLOW`, ou pode usar um valor RGB, como `#FF0000`.

Se você definir a tag H5 como segue:

```
H5 {border-color: green; border-width: 3px}
```

Teríamos um texto como abaixo:

Exemplo de texto com borda de cor verde

(Este exemplo só será visualizado corretamente no Netscape Navigator)

Se você não usar `border-color`, então a borda terá a cor do elemento. Se você quiser que cada borda tenha uma cor de borda diferente, então você poderá criar uma sequência de valores para a propriedade `border-color`, sendo estes valores na sequência: borda superior, borda direita, borda inferior, borda esquerda. Veja o exemplo:

```
H1 {border-color: green red blue orange; border-style: solid;border-width: 2px}
```

e o resultado:

Atenção : Bordas multicoloridas não funcionam no Netscape Navigator, somente no Internet Explorer.

- **border-style**

Com esta propriedade, podemos determinar o estilo de desenho da borda. Você pode setar de um a quatro valores para esta propriedade, nas quais o browser fará as seguintes ações:

| Número de Valores | Descrição da ação do browser |
|-------------------|--|
| Um valor | Todas as quatro bordas do elemento usarão o estilo definido |
| Dois valores | As bordas superior e inferior usam o primeiro valor, e as bordas esquerda e direita usam o estilo definido no segundo valor |
| Três valores | A borda superior usa o estilo definido no primeiro valor; as bordas esquerda e direita usam o estilo do segundo valor, e o estilo do terceiro valor é definido para a borda inferior |
| Quatro valores | Cada valor é definido para cada borda na sequência: borda superior, borda direita, borda inferior e borda esquerda |

Os valores possíveis para o estilo de borda são os seguintes:

| Valor | Descrição |
|--------|------------------|
| none | Sem borda |
| dotted | Linha pontilhada |
| dashed | Linha tracejada |
| solid | Linha sólida |
| double | Linha dupla |
| groove | Moldura 3-D |
| ridge | Moldura 3-D |
| inset | Moldura 3-D |
| outset | Moldura 3-D |

Se definirmos a tag H2 como se segue,

H2 {border-color: green; border-style: groove; border-width: 3px}

Teríamos o seguinte:

Exemplo de borda 3-D

Propriedades de margens

Com as propriedades de margens, podemos definir as margens em volta do elemento.

- **margin-top, margin-bottom, margin-left, margin-right**

As propriedades acima definem o seguinte:

| Propriedade | Descrição |
|---------------|-----------------|
| margin-top | Margem superior |
| margin-bottom | Margem inferior |
| margin-left | Margem esquerda |
| margin-right | Margem direita |

Veja o exemplo abaixo:

```
H4 {margin-top: 20px; margin-bottom: 5px; margin-left: 100px; margin-right: 55px}
```

Você pode definir cada margem separadamente, ou pode definir uma margem e deixar que o browser utilize o valor padrão para as outras margens que você não definiu.

Podemos também sobrepor elementos utilizando valores de margem negativos. Confira o seguinte exemplo:

```
H4 {margin-top: -30px; margin-left: 100px}
```

E observe o resultado final:

Exemplo de texto

Sobreposto

Propriedades de margens internas

Usamos essas propriedades para definir o espaço entre o elemento e suas bordas.

- **padding-top, padding-bottom, padding-left, padding right**

Estas propriedades definem:

| Propriedade | Descrição |
|----------------|-------------------------|
| padding-top | Margem interna superior |
| padding-bottom | Margem interna inferior |
| padding-left | Margem interna esquerda |
| padding-right | Margem interna direita |

```
H2 {border-color: green; border-style: groove; border-width: 3px; padding-top: 20px; padding-bottom: 5px; padding-left: 10px; padding-right: 55px}
```

Exemplo de texto de margem interna alterada

Outras propriedades

- **float**

Permite que um elemento seja posicionado a esquerda ou a direita, e com outros elementos em volta dele. Este elemento é definido como "flutuante".

A tabela seguinte mostra os valores que você pode atribuir-lhe:

| Valores | Descrição |
|---------|---|
| none | Mostra o elemento sem alteração |
| left | Move o elemento para a esquerda e posiciona o texto em volta dele |
| right | Move o elemento para a direita e posiciona o texto em volta dele |

Exemplificando:

```
<p>So this is
Xmas<br>
And what have you done?<br>
Another year over and a new one just begun</p>
```

- **clear**

Determina se o browser pode mostrar elementos flutuantes nos lados de um elemento.

| Valores | Descrição |
|---------|--|
| none | Elementos flutuantes são permitidos em todos os lados |
| left | Elementos flutuantes não são permitidos a esquerda |
| right | Elementos flutuantes não são permitidos a direita |
| both | Elementos flutuantes não são permitidos em nenhum lado |

Veja primeiramente que os dois parágrafos estão sem a propriedade clear definida.

```
<p>Parágrafo 1 - sem
clear</p>
<p>Parágrafo 2 - sem clear</p>
```

Agora veremos exemplo com o parágrafo 2 definido com a propriedade clear: left

```
<p>Parágrafo 1 - sem
clear</p>
<p style="clear: left">Parágrafo 2 - com clear</p>
```

Observe que, no segundo exemplo, o browser não mostrou o parágrafo 2 ao lado da imagem por causa da propriedade clear, que com valor left, não permite que objetos flutuantes fiquem a esquerda do elemento.

Fonte Tridimensional

Agora vamos entrar na área de **posicionamento de objetos** da folha de estilo para aprender um efeito novo e interessante. O que é um objeto? Objeto é todo elemento envolvido por uma tag, como por exemplo um texto, uma imagem, um link... O título **Fonte Tridimensional** nada mais é que três camadas de objetos sobrepostos, ou seja, três frases iguais porém com cores e alinhamentos diferentes.

Sua sobreposição é possível devido a propriedades de posicionamento **position, top e left**. Position está configurada com o valor **absolute** que define uma posição fixa no monitor. Top será a distância que o bloco terá da borda superior (de seu monitor) e left a distância da borda esquerda. Os valores top/left estão em **pixels(px)** Então a sintaxe ficará assim:

```
<DIV STYLE="position:absolute; top:80; left:196; font-family:arial; color:000000; font-size:45">Fonte Tridimensional</DIV>
<DIV STYLE="position:absolute; top:82; left:198; font-family:arial; color:daa520; font-size:45">Fonte Tridimensional</DIV>
<DIV STYLE="position:absolute; top:84; left:200; font-family:arial; color:228b22; font-size:45">Fonte Tridimensional</DIV>
```

Observe que os valores de top (**80, 82 e 84**) e left (**196, 198 e 200**), criam um ligeiro deslocamento de cada camada para baixo e para a direita. É esse deslocamento que irá criar o efeito tridimensional da fonte. Para alterar o ângulo de inclinação basta manipular esses valores. A ordem de sobreposição obedece a ordem da sintaxe, isso é, o objeto que vem primeiro será sobreposto pelos que vem depois, se você inverter as posições o mesmo ocorrerá no efeito visual.

Bom..., nem tudo é um mar de rosas para o webmaster. O problema do posicionamento de objetos com o *position:absolute* está nas diversas resoluções de monitores que existem por aí. Por isso aconselho usar esse recurso sempre no topo da página, assim ela permanecerá, mais ou menos, no local que você estipulou em monitores com resoluções diferentes. Como cerca de 90% dos monitores são de resolução 800 por 600 pixels aconselho usa-la como referência.

Uma vantagem é que se você for criar um arquivo gif com essas dimensões ele ocuparia, pelo menos, uns **8.000 bytes** de tamanho enquanto que pela forma de posicionamento de objetos ele ocuparia seus reais **400 bytes**, ou seja, **20 vezes** menor. Algo a se considerar.



NOTA: O browser **Netscape v4.51** não aceita o posicionamento de objetos. Já os browseres **IE v5.5** e o **Opera v5.02** aceitam esse recurso perfeitamente.

```
<span> ... </span>
```

Ênfase definida por folha de estilos. O texto marcado com este elemento se apresentará na tela de acordo com as propriedades de estilo (cor, fonte, tamanho, estilo, sublinhado, background, etc.) definidas em uma folha de estilos. No exemplo abaixo estou usando a tag **SPAN** para aplicar uma **cor de fundo às fontes**:

```
<SPAN style="background:#9acd32"><h2>Êxito</h2></SPAN>
```

Êxito

Todo aquele que pretende realizar um projeto, para obter o mais completo êxito, deve começar por convencer-se de que logrará o que deseja; deve ter a determinação de vencer todos os obstáculos, empregando o maior entusiasmo em tudo quanto fizer. Seja qual for o seu objetivo, dedica-lhe todas as energias do vosso cérebro; e, concentrando o vosso pensamento, dizeis: "Tereis bom êxito, nada haverá que seja capaz de me impedir".

Agora aplicando...

Agora aplicando uma textura no fundo do texto, usando o mesmo princípio

SPAN também possibilita criar diversos efeitos visuais na fonte através da propriedade **filter** ideais para destacar títulos ou dar maior ênfase a alguma frase. Aqui estão selecionados alguns deles:

EFEITO GLOW

No século 18...

"No século 18, a invenção do pára-raios, por **Benjamin Franklin**, foi condenada pela Igreja como invenção do diabo. Sendo o raio expressão da cólera do Senhor, só podia ser tentação do demônio impedir que o castigo divino caísse sobre o mundo. Na França, a "excomunhão" foi levada a sério: em Saint-Omer, Vissey de Bois foi processado por heresia, por instalar um pára-raios em sua casa".

Este efeito de borramento é conseguido através da propriedade de estilo **filter:glow** aplicado a tag **span**, conforme mostra a sintaxe acima. Ele também pode ser aplicado as linhas e figuras (arquivos gif, jpg...), basta envolver o elemento com span e suas propriedades. **height:1** destaca a visualização do efeito, **(color=#FF0000)** especifica a cor do borramento e finalmente **color:#FFDEAD** a cor do texto.

EFEITO BLUR

No século 18...

"No século 18, a invenção do pára-raios, por **Benjamin Franklin**, foi condenada pela Igreja como invenção do diabo. Sendo o raio expressão da cólera do Senhor, só podia ser tentação do demônio impedir que o castigo divino caísse sobre o mundo. Na França, a "excomunhão" foi levada a sério: em Saint-Omer, Vissey de Bois foi processado por heresia, por instalar um pára-raios em sua casa".

Este efeito de letras em 3D é conseguido através da propriedade de estilo **filter:blur** e configuração de seus atributos: **Direction=225** indica o ângulo de inclinação da sombra que vai de 0 a 360 graus (no exemplo coloquei uma inclinação de 225 graus), e **strength=4** indica o comprimento que a sombra deve ter.

DIRECTION indica o ângulo de inclinação do efeito

| | | | |
|---------------|---------------|---------------|---------------|
| Direction=0 | Direction=45 | Direction=90 | Direction=135 |
| Direction=180 | Direction=225 | Direction=270 | Direction=315 |

STRENGTH indica o comprimento do efeito

| | | | |
|------------|------------|------------|------------|
| Strength=2 | Strength=5 | Strength=7 | Strength=9 |
|------------|------------|------------|------------|

EFEITO DROPSHADOW

`No século 18...`

"No século 18, a invenção do pára-raios, por **Benjamin Franklin**, foi condenada pela Igreja como invenção do diabo. Sendo o raio expressão da cólera do Senhor, só podia ser tentação do demo impedir que o castigo divino caísse sobre o mundo. Na França, a "excomunhão" foi levada a sério: em Saint-Omer, Vissey de Bois foi processado por heresia, por instalar um pára-raios em sua casa".

Onde **color:#FFFFFF** define a cor da sombra, **offx=3** o deslocamento horizontal em pixel, **offy=3** o deslocamento vertical em px, **positive=1** a sombra terá um efeito externo ou **0** para um efeito interno.

Deslocamentos do efeito com **offx & offy**

| | | | |
|----------------|----------------|----------------|------------------|
| offx=0, offy=5 | offx=5, offy=0 | offx=5, offy=5 | offx=10, offy=10 |
|----------------|----------------|----------------|------------------|

EFEITO WAVE

`EFEITO WAVE`

Onde **Freq=2** define o número de ondulações, **Phase=5** determina a sua frequência e **Strength=50** a sua intensidade.

EFEITO MASK

`EFEITO MASK`

Este efeito cria uma máscara de fundo para a fonte que passa a ser transparente. Repare que o background da página também acontece no interior da fonte. Seu parâmetro **color=#800080** define a cor da máscara.

EFEITO SHADOW

Sintaxe:

```
<span style="height:10; color:#556B2F  
filter:shadow(color=ff0000, direction=320, enabled=1)">  
<h1>World<BR> Wide<BR> Web</h1>  
</span>
```

World
Wide
Web

Onde **height:10** ajusta a distância do sombreado com a próxima linha de baixo, **color:#556B2F** define a cor da letra, **color=ff0000** (entre aspas) define a cor da sombra, **direction=320** ajusta o ângulo de direção, e **enabled=1** habilita o efeito.

Os browsers **Netscape**, **Opera** e **Internet Explorer** inferior a **v5,5** não aceitam estes recursos.

FILTROS COMBINADOS

Agora que sabemos, um pouco, como eles funcionam já podemos criar alguns efeitos interessantes usando dois ou mais filtros sobre o mesmo objeto. No exemplo a seguir vou usar um combinado dos filtros **Wave** e **Glow** sobre o texto **CARTOON PICS**. Repare que alguns webmasters preferem simular este mesmo efeito criando arquivos.gif (por acharem esta maneira mais simples), que ficarem se envolvendo com parâmetros de DHTML/CSS, o que não é o nosso caso.

CARTOON PICS

Sintaxe:

```
<P ALIGN="center" STYLE="font-size:42px; font-family:'arial black';  
filter:wave(Add=1, Freq=2, LightStrength=100, Phase=16, Strength=1)  
filter:glow(color=black, Strength=1); height=1; font-weight:bold; color:purple">  
CARTOON PICS
```

Na sintaxe acima realcei os parâmetros dos dois filtros com cores distintas para que você possa observar onde começa e termina cada um com a formatação de suas respectivas propriedades (que está envolvida pelo HTML convencional). Um detalhe importante é que para obter o efeito degrade do texto, você não pode formatá-lo com o **código RGB (Red/Green/Blue)** mas somente com os **Nomes Padrão** suportados pelos browsers, Caso contrário o degrade não funcionará.

Assim, como usei a nome padrão da cor **purple**, no exemplo, então seu degrade correspondente será a cor **funchsia**, como consta na tabela acima. Ainda aproveitando o mesmo exemplo agora vou fazer outro combinado dos filtros **DropShadow** e **Glow** modificando algumas propriedades como fonte e cor:

CARTOON PICS

Sintaxe:

```
<P ALIGN="center" STYLE="font-size:42px; font-family:verdana; filter: DropShadow(offx=3, offy=3, color:#ff00ff, positive=1) filter: glow(color=black, Strength=2); height=1; font-weight:bold; color:#8a2be2"> CARTOON PICS
```

Uma pequena observação: Vai acontecer que quando você aplicar a sintaxe acima em um texto com letras minúsculas, as letras "p", "q", "y" e "j" aparecerão com suas bases cortadas.

COPY THE PICS

Para resolver isso basta aumentar a altura da linha inserindo a propriedade **line-height** e configurar seu valor para que o corte desapareça.

Sintaxe:

```
<P ALIGN="center" STYLE="font-size:35px; font-family:verdana; filter: DropShadow(offx=3, offy=3, color:#ff00ff, positive=1) filter: glow(color=black, Strength=2); height=1; font-weight:bold; color:#8a2be2; line-height:50px"> copy the pics
```

COPY THE PICS

Outro exemplo agora combinando 3 filtros: **Mask**, **Shadow** e **Chroma**. Este, porém, vou apresentar por etapas para que fique bem claro o efeito que cada filtro irá causar no objeto.

1) **filter:mask(color=#000000)** cria uma máscara de cor preta que irá envolver o texto e **width:280; height:70** define o tamanho que a máscara deve ter.

Olhe Isto!

2) O filtro **shadow(color=#228b22, direction=135)** cria um sombreado onde **color=#228b22** define a sua cor e **direction=135** o seu ângulo de inclinação.

3) E, finalmente o filtro **chroma(color=#000000)** anula a máscara preta realçando o efeito interno do texto.

Sintaxe:

```
<P ALIGN="center"> <SPAN STYLE="width:280; height:70; font-size:50px; font-family:verdana; font-weight:bold; filter:mask(color=#000000) shadow(color=#228b22, direction=135) chroma(color=#000000)">Olhe Isto! </SPAN>
```

Logicamente este efeito só irá funcionar em textos com letras grandes, algo acima de **font-size:30px** e espessura realçada com **font-weight:bold**. Abaixo configurei o mesmo efeito em um texto com tamanho normal. O efeito desapareceu!...

"No século 18, a invenção do pára-raios, por **Benjamin Franklin**, foi condenada pela Igreja como invenção do diabo. Sendo o raio expressão da cólera do Senhor, só podia ser tentação do demo impedir que o castigo divino caísse sobre o mundo. Na França, a "excomunhão" foi levada a sério: em Saint-Omer, Vissey de Bois foi processado por heresia, por instalar um pára-raios em sua casa".

Só para reforçar o entendimento agora vou pegar este mesmo exemplo e alterar alguns parâmetros para ver o que acontece:

```
<STYLE TYPE="text/css">
.efeito2a {color:#003366; font-size:35px; font-family:arial black;
margin:0px;}
.efeito2b {color:#999999; font-size:35px; font-family:arial black; margin:0px
5px -52px 15px;}
.efeito2c {color:#CCCCC; font-size:35px; font-family:arial black; margin:0px
10px -53px 30px;}
</STYLE>
```

```
<P ALIGN="center" CLASS=efeito2c>Cascading Style Sheets
<P ALIGN="center" CLASS=efeito2b>Cascading Style Sheets
<P ALIGN="center" CLASS=efeito2a>Cascading Style Sheets
```

```
<STYLE TYPE="text/css">
.efeito3a {color:#000000; font-size:40px; margin:0px;}
.efeito3b {color:#444444; font-size:40px; margin:0px 0px -35px 20px;}
.efeito3c {color:#888888; font-size:40px; margin:0px 0px -35px 40px;}
.efeito3d {color:#cccccc; font-size:40px; margin:0px 0px -35px 60px;}
</STYLE>
```

```
<STYLE TYPE="text/css">
.efeito4a {color:#a0522d; font-size:35px; font-family:courier new; margin:0;
letter-spacing:0px;}
.efeito4b {color:#daa520; font-size:40px; font-family:courier new;
margin:0px 0px -45px 20px; letter-spacing:-1px;}
.efeito4c {color:#ffd700; font-size:45px; font-family:courier new; margin:0px
0px -50px 35px; letter-spacing:-2px;}
.efeito4d {color:#ffed5; font-size:50px; font-family:courier new; margin:0px
0px -55px 50px; letter-spacing:-3px;}
</STYLE>
```

* FILTROS *

Observe a barra de comandos no topo desta página. Ela parece estranha, quase sumida... Agora repare o rótulo colorido "*Tutorial DHTML/CSS Edaurelio*" logo abaixo da barra à esquerda, está desfocado... Repare também no título desta seção **"*FILTROS*"** ele apresenta um sombreado e a sua letra **R** está invertida.

Estes efeitos visuais apareceram pela primeira vez como controles de **ActiveX** e posteriormente foram incorporados como propriedades de folha de estilo aceitas inicialmente pelo browser **Internet Explorer v4.0**. Efeitos anteriormente que só poderiam ser criados com aplicativos como o Photoshop em arquivos de imagem agora podem ser usados em qualquer objeto HTML como por exemplos bloco de texto, tabelas, links, bordas, etc e inclusive criar efeitos dinâmicos!! Eles são fáceis de aprender e basicamente sua sintaxe é a seguinte:

<DIV STYLE="filter:nome(parâmetro=valor, parâmetro=valor, etc...)">objeto</DIV>

Onde **name** define o tipo de filtro, **parâmetro** define suas propriedades de formatação e **valor** suas unidades que geralmente são graus, pixels, %, etc. A seguir apresento uma relação de filtros em ordem alfabética, bem como seus parâmetros e alguns exemplos interessantes. Você pode aplicar diversos filtros ao mesmo objeto criando uma variedade de efeitos conforme manda a sua imaginação.

ALPHA

O filtro **Alpha**, cujo efeito já foi apresentado na barra de comandos acima, cria um efeito de transparência total ou parcial no objeto por ele envolvido de forma que quanto menor for a sua graduação de porcentagem, definida por **opacity=%** maior será a sua transparência.

PARÂMETROS:

- ☐ **Opacity** - Apresenta o grau de transparência do objeto e seu valor refere-se a **%**. *Opacity=100* apresenta um objeto totalmente opaco e *opacity=0* um objeto totalmente invisível.
- ☐ **FinishOpacity** - Define o nível de opacidade e seus valores vão de **0** a **100%**.
- ☐ **Style** - Seus estilos podem ser **0** (uniforme), **1** (linear), **2** (radial) ou **3** Retangular.
- ☐ **StartX** - Coordenada horizontal para começo do gradiente de opacidade.
- ☐ **StartY** - Coordenada vertical para começo do gradiente de opacidade.
- ☐ **FinishX** - Coordenada horizontal para o fim do gradiente de opacidade.
- ☐ **FinishY** - Coordenada vertical para o fim do gradiente de opacidade.

No exemplo a seguir temos um bloco de texto com fonte preta e diversos graus de opacidade com estilos diferentes.

ALPHA UNIFORME

ALPHA UNIFORME

ALPHA LINEAR

ALPHA LINEAR

ALPHA RADIAL INTERNO

ALPHA RADIAL INTERNO

ALPHA RADIAL EXTERNO

ALPHA RADIAL EXTERNO

Agora o mesmo efeito alpha aplicado a uma tabela, primeiro apresentando a tabela normal, depois ela mesma envolvida pelo filtro:

A sintaxe simplificada da tabela é esta:

```
<DIV STYLE="filter:alpha(opacity=20, style=2); height:1">
<TABLE>
...
</TABLE>
</DIV>
```

BLUR

Este filtro cria um efeito difuso diminuindo a definição das bordas do objeto. Sua sombra apresenta a mesma cor do objeto porém com uma textura mais atenuada.

PARÂMETROS:

- ☐ **add** - Parâmetro booleano, **1** Sombreamento nítido, **0** Sombreamento difuso.
- ☐ **direction** - Define o ângulo em que o efeito deve acontecer. Seu valor vai de **0** (topo) em incrementos de 45 graus até **315** graus.
- ☐ **strength** - Define o comprimento que o efeito deve ter. Default = 5.

```
<SPAN STYLE="height:1; filter: Blur(Direction=45, add=0, Strength=3)"><FONT
COLOR="blue">
<IMG SRC="imagens/cs05g.gif">Figura e texto</FONT> <B>COM</B><FONT
COLOR="red"> o filtro.</FONT></SPAN>
```

No exemplo acima envolvi uma imagem e um texto de três cores. Repare que a difusão acompanha as cores do texto. Para aprender mais sobre este filtro vá para []

CHROMA

Este filtro torna uma cor específica do objeto transparente. No exemplo a seguir tenho a imagem **home1.gif** cujo fundo branco cria um contraste com o background da página. Porém com o filtro chroma posso neutralizar o fundo branco (**#FFFFFF**) da imagem simulando uma imagem GIF de fundo transparente.

PARÂMETROS:

☐ **color** - Define uma cor do código RGB que sofrerá o efeito de transparência.

NOTA: O filtro não funcionará se a cor selecionada para o efeito não existir na figura.

DROPSHADOW

Este filtro cria um efeito de sombra sólida no objeto que pode ser direcionada conforme o ângulo desejado. No exemplo a seguir temos a mesma tabela já apresentada no filtro Alpha, porém sem o BGCOLOR (para realçar o efeito deste filtro sobre ela).

PARÂMETROS:

☐ **color** - Define a cor RGB (**#rrggbb**) da sombra. Os valores são em hexadecimal.

☐ **offx** - Determina a distância de afastamento em "x" (**horizontal**), os valores em pixels (px) podem ser positivos ou negativos.

☐ **offy** - Determina a distância de afastamento em "y" (**vertical**), os valores em pixels (px) podem ser positivos ou negativos.

☐ **positive** - Parâmetro booleano: **1** para sombra positiva. **0** para sombra negativa.

Sintaxe simplificada da tabela:

```
<DIV STYLE="filter:dropshadow(color=#1E90FF, offx=-2,
offy=-2, positive=1); height:1">
<TABLE>
...
</TABLE>
</DIV>
```

NOTA: Embora esse filtro possa ser aplicado em qualquer objeto, ele apresentará um melhor resultado em fontes ou imagens com fundos transparentes ou vazados. Para aprender mais sobre este filtro vá para

FLIPH e FLIPV

Estes dois filtros criam o efeito de imagem invertida no objeto. **FlipH** inverte no sentido horizontal como um reflexo de espelho e **FlipV** inverte no sentido vertical como o reflexo na superfície de um lago.

```
</P ALIGN="center" STYLE="filter:flipH; height:1; font-size:20px; color:#8B008B">
<B>Socorram-me,...</B>
```

Socorram-me, subi no ônibus em Marrocos.

```
<P ALIGN="center" STYLE="filter:flipV; height:1; font-size:20px; color:#8A2BE2">
<B>Socorram-me,...</B>
```

Socorram-me, subi no ônibus em Marrocos.

GLOW

O filtro Glow cria um efeito de borramento nos contornos do objeto. Este borramento pode assumir uma cor diferente do objeto. Repare que no exemplo a seguir o objeto (uma tabela) possui bordas brancas com um texto preto, e no entanto seu borramento é azul (**color=#1E90FF**).

PARÂMETROS:

- ☐ **color** - Define a cor RGB (**#rrggbb**) do borramento.
- ☐ **strength** - Determina a distância de afastamento do efeito. Seu valor é em pixels (**px**) e pode ser valor positivo (borramento externo) ou negativo (borramento interno).

Sintaxe simplificada da tabela:

```
<DIV STYLE="filter:glow(color=#1E90FF strength=10); height:1">
<TABLE>
...
</TABLE>
</DIV>
```

GRAY

Este filtro anula os efeitos de cor do objeto apresentando-o em tons de cinza. Até onde sei este filtro não possui parâmetros e a sintaxe é bem simples.

INVERT

Este filtro tem a propriedade de inverter a matriz de cores do objeto criando um efeito semelhante ao negativo de uma fotografia.

MOTION BLUR

Este filtro cria um efeito de movimento no objeto.

PARÂMETROS:

- ☐ **strenght** - Define o comprimento do efeito, seu valor é em pixels.
- ☐ **direction** - Define o grau em que o efeito deve ocorrer. Ele pode ser de **0** a **360** graus.
- ☐ **progid:DXImageTransform.Microsoft** - Parâmetros proprietários da Microsoft.

No exemplo a seguir há uma imagem normal e mais duas com o efeito aplicado dentro de uma tabela. Todas elas foram centralizadas dentro de suas respectivas células. Repare que as imagens com efeito Motion Blur sofreram uma deslocação no sentido do movimento que o efeito causou. Isso acontece porque agora o efeito passa a fazer parte da imagem alterando o seu centro de equilíbrio.

Sintaxe da célula 3

```
<SPAN STYLE="HEIGHT:1;  
filter:progid:DXImageTransform.Microsoft.MotionBlur(strength=40,  
direction=30)">  
<IMG SRC="imagens/gal001.gif">  
</SPAN>
```

WAVE

Cria um efeito de ondulação no objeto.

PARÂMETROS:

- ☐ **add** - Parâmetro booleano: **1** cria o efeito wave, **0** anula o efeito.
- ☐ **freq** - Define o número de ondulações que o objeto deve sofrer.
- ☐ **lightstrength** - Define a intensidade de ondulação luminosa. Seu valor vai de **0** a **100%**.
- ☐ **phase** - Define a frequência da ondulação, seu valor vai de **0** a **360graus**.
- ☐ **strength** - Define a intensidade da ondulação da imagem.

XRAY

Este filtro cria um efeito semelhante ao filtro **Gray** transformando as cores do objeto em tons de cinza. Ainda não sei quais as diferenças entre esses dois filtros.