CSS

Cascading Style Sheets

Profº Elton Rodrigo

Esta apostila foi criada e editada com base em informações contidas nos livros **Guia CSS (Editora AltaBooks, 2011)**, **Construindo sites com CSS (Editora Novatec, 2011), Guia de Orientação e Desenvolvimento de Sites (Editora Erica 2011)** e nos sites **códigofonte.net**, **scriptbrasil.com.br, webdeveloper.com, internetbootcamp.net** e **webmasters.com.**

As imagens nela contidas foram capturadas com *PrintScreen* de computadores associados.

Sumário

Introdução

Cascading Style Sheets (ou simplesmente CSS) é uma linguagem de folhas de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação, como HTML ou XML. Seu principal benefício é prover a separação entre o formato e o conteúdo de um documento.

Em vez de colocar a formatação dentro do documento, o desenvolvedor cria um link (ligação) para uma página que contém os estilos, procedendo de forma idêntica para todas as páginas de um portal. Quando quiser alterar a aparência do portal basta portanto modificar apenas um arquivo.

Com a variação de atualizações dos navegadores (browsers) como Internet Explorer que ficou sem nova versão de 2001 a 2006, o suporte ao CSS pode variar. O Internet Explorer 6, por exemplo, tem suporte total a CSS1 e praticamente nulo a CSS2. Navegadores mais modernos como Google Chrome e Mozilla Firefox tem suporte maior, inclusive até a CSS3.

A interpretação dos navegadores pode ser avaliada com o teste Acid2, que se tornou uma forma base de revelar quão eficiente é o suporte de CSS, fazendo com que a nova versão em desenvolvimento do Firefox seja totalmente compatível a ele assim como o Opera já é. O Doctype informado ou a ausência dele determina o quirks mode ou o strict mode modificando o modo como o CSS é interpretado e a página desenhada.

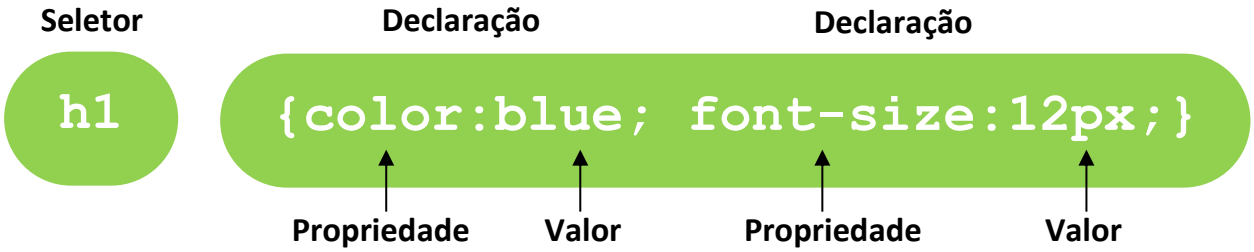
1. O que são folhas de estilo?

Uma folha de estilos é um conjunto de regras que informa a um programa, responsável pela formatação de um documento, como organizar a página, como posicionar e expor o texto e, dependendo de onde é aplicada, como organizar uma coleção de documentos.

A maior parte dos programas de editoração eletrônica e processadores de texto modernos trabalham com folhas de estilos. O processo consiste em definir um rótulo (nome do estilo) para um determinado parágrafo e em seguida alterar os seus atributos. Todo parágrafo que for rotulado com aquele estilo passará a exibir as características definidas anteriormente. Qualquer alteração nos atributos de um estilo afetará todos os parágrafos que estiverem rotulados com ele.

Esta descrição, que se aplica a estilos em processadores de texto e programas de editoração eletrônica, também vale para a Web. Na Web, os "parágrafos" são blocos marcados por descritores HTML como <h1>, <p>, etc. Para fazer com que todos os blocos de textos marcados com <h1> em um documento sejam exibidos em tamanho de 48 pontos, basta definir a regra abaixo dentro de uma "folha de estilos" aplicada ao documento.

h1 {color:blue; font-size: 12px}

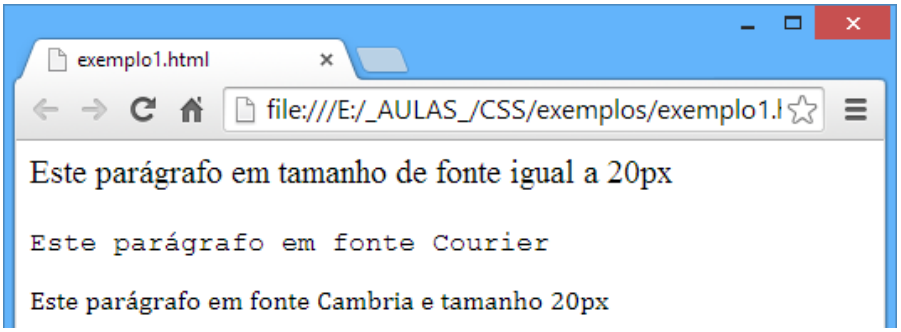


1.1 - Atribuindo à pagina

CSS pode ser adicionado com uso do atributo style. Por exemplo, pode ser definido o tipo e o tamanho da fonte em um parágrafo:

```
<p style="font-size:20px;">Este parágrafo em tamanho de fonte
igual a 20px</p>
<p style="font-family:courier;">Este parágrafo em fonte
Courier</p>
<p style="font-size:16px; font-family:cambria">Este parágrafo
em fonte Cambria e tamanho 20px</p>
```

Será renderizado no navegador assim:



No exemplo acima foi usado o atributo style para definir o tipo de fonte usado (com a propriedade font-family) e o tamanho da fonte (com a propriedade font-size). Notar que no último parágrafo do exemplo foi definido tanto o tipo como o tamanho da fonte separados por um ponto e vírgula.

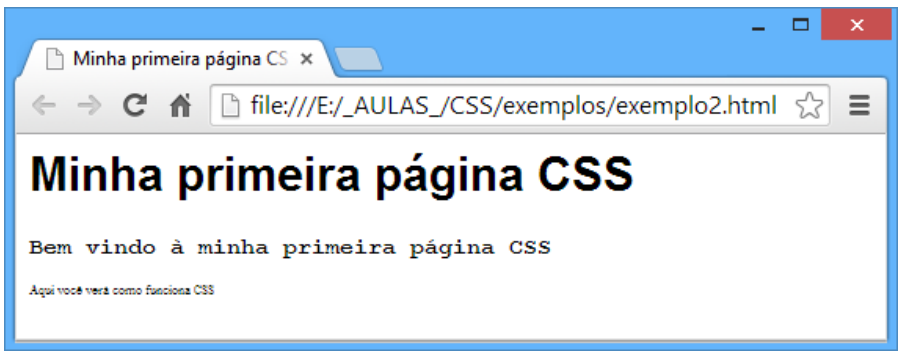
1.2 – Estilos em um bloco central

Uma das funcionalidades mais inteligentes das CSS é a possibilidade de controlar o layout de um bloco central. Em lugar de se usar o atributo style em cada tag, pode-se dizer ao navegador como deve ser o layout de todos os textos em uma página:

```
<html>
    <head>
    <title>Minha primeira página CSS</title>
<style type="text/css">
h1 {font-size: 30px; font-family: arial}
h2 {font-size: 15px; font-family: courier}
p {font-size: 8px; font-family: times new roman}
</style>
    </head>
<body>
    <h1>Minha primeira página CSS</h1>
    <h2>Bem vindo à minha primeira página CSS</h2>
    <p>Aqui você verá como funciona CSS</p>
</body>
</html>
```

No exemplo acima inserimos as CSS na seção **head** do documento, assim ela se aplica à página inteira. Para fazer isto é necessário usar a tag <style type="text/css"> que informa ao navegador que está digitando CSS.

No exemplo, todos os cabeçalhos da página serão em fonte Arial e tamanho 30px. Todos os subtítulos serão em fonte Courier tamanho 15. E, todos os textos dos parágrafos serão em fonte Times New Roman tamanho 8.



2. Como funciona as CSS

Muitas das propriedades usadas em Cascading Style Sheets (CSS) são semelhantes àquelas do HTML. Se você está acostumado a usar HTML para layout irá reconhecer muitos dos códigos a serem usados.

2.1 - A sintaxe básica das CSS

Para definir uma cor de fundo vermelha para a página web usando HTML pode-se fazer assim:

```
<body bgcolor="#FF0000">
```

Com CSS o mesmo resultado será obtido assim:

```
body {background-color: #FF0000;}
```

Os códigos HTML e CSS são mais ou menos parecidos. O exemplo acima serve também para demonstrar o fundamento do modelo CSS:

seletor {propriedade: valor}

seletor: Em qual tag será aplicada a propriedade. Por exemplo: body

propriedade: A propriedade pode ser como por exemplo: a cor do fundo

valor: O valor da propriedade cor do fundo por exemplo: vermelha("#FF0000")

2.2 - Aplicando CSS a um documento HTML

É possível aplicar CSS a um documento de três maneiras distintas. Os três métodos de aplicação estão exemplificados a seguir. É recomendado focar no terceiro método, ou seja o método externo. **O método externo além de ser menos confuso para trabalhar por não estar junto com o HTML não ficará amostra se alguém olhar o código fonte da página pelo browser, pois estará 'escondido' no servidor onde está hospedando o site.**

Método 1: In-line (o atributo style)

Uma maneira de aplicar CSS é pelo uso do atributo style do HTML. Tomando como base o exemplo mostrado anteriormente a cor vermelha para o fundo da página pode ser:

```
<html>
  <head>
   <title>Exemplo</title>
  </head>
  <body style="background-color:
#FF0000;">

<p>Esta é uma página com fundo
vermelho</p>
</body>
</html>
```

Método 2: Interno (a tag style)

Uma outra maneira de aplicar CSS e pelo uso da tag <style> do HTML. Como mostrado:

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

<style type="text/css">
 body {background-color: #FF0000;}
</style>

  </head>
<body>
  <p>Esta é uma página com fundo
vermelho</p>
</body>
</html>
```

Método 3: Externo (link para uma folha de estilos)

O método recomendado é o de lincar para uma folha de estilos externa.

Uma folha de estilos externa é um simples arquivo de texto com a extensão **.css**. Tal como com qualquer outro tipo de arquivo pode-se colocar uma folha de estilos tanto no servidor como no disco rígido.

Supondo, por exemplo, que uma folha de estilos tenha sido nomeada de “estilo.css” e está localizada no diretório **estilo (o que não é necessário, colocar em pasta separada do arquivo HTML)**. Tal situação está mostrada a seguir:

O que você tem a fazer é criar um link no documento HTML (index.html) para a folha de estilos (estilo.css). O link é criado em uma simples linha de código HTML como mostrado a seguir:

```
<link rel="stylesheet" type="text/css" href="c:\~documentos~/estilo.css" />
```

Nota-se que o caminho para a folha de estilos é indicado no atributo **href**.

Esta linha de código deve ser inserida na seção header do documento HTML, isto é, entre as tags <head> e </head>. Conforme mostrado abaixo:

```
<html>
<head>
<title>Meu documento</title>
<link rel="stylesheet" type="text/css" href="style/style.css" />
</head>
<body>
...
```

Este link informa ao navegador para usar o arquivo CSS na renderização e apresentação do layout do documento HTML.

A coisa realmente inteligente disto é que vários documentos HTML podem *linkar* para uma mesma folha de estilos. Em outras palavras isto significa que um simples arquivo será capaz de controlar a apresentação de muitos documentos HTML.

Esta técnica pode economizar uma grande quantidade de trabalho. Por exemplo, caso seja necessário trocar a cor do fundo de um site com 100 páginas, a folha de estilos evita ter que editar manualmente uma a uma as páginas para fazer a mudança nos 100 documentos HTML. Usando CSS a mudança se fará em uns poucos segundos trocando-se a cor em uma folha de estilos central.

Exercício para fixação

Crie dois arquivos — um arquivo HTML e um arquivo CSS — com os seguintes conteúdos:

index.html

```
<html>
<head>
<title>Meu documento</title>
<link rel="stylesheet" type="text/css" href="estilo.css" />
</head>
<body>
<h1>Minha primeira folha de estilos</h1>
</body>
</html>
```

estilo.css

```
body {
background-color: #FF0000;
}
```

Salve os dois arquivos no mesmo diretório. Lembre-se de salvar os arquivos com a extensão apropriada (".css" e ".html"(ou “htm”).

Abra index.html no seu navegador e veja uma página com o fundo vermelho.

3. Cores e fundos

É possível aplicar cores de primeiro plano e cores de fundo em um website. Será abordado ainda os métodos avançados de controle e posicionamento de imagens de fundo.

3.1 - Cor do primeiro plano: a propriedade 'color'

A propriedade color define a cor do primeiro plano de um elemento.

Considerando que seja desejado que todos os cabeçalhos de primeiro nível no documento sejam na cor vermelha. O elemento HTML que marca tais cabeçalhos é o elemento <h1>. O código a seguir define todos os <h1> na cor vermelha.

```
h1 {
      color: #ff0000;
}
```

As cores podem ser definidas pelo seu valor hexadecimal como no exemplo acima (#ff0000), com uso do nome da cor ("red") ou ainda pelo seu valor rgb (rgb(255,0,0)).

3.2 - A propriedade 'background-color'

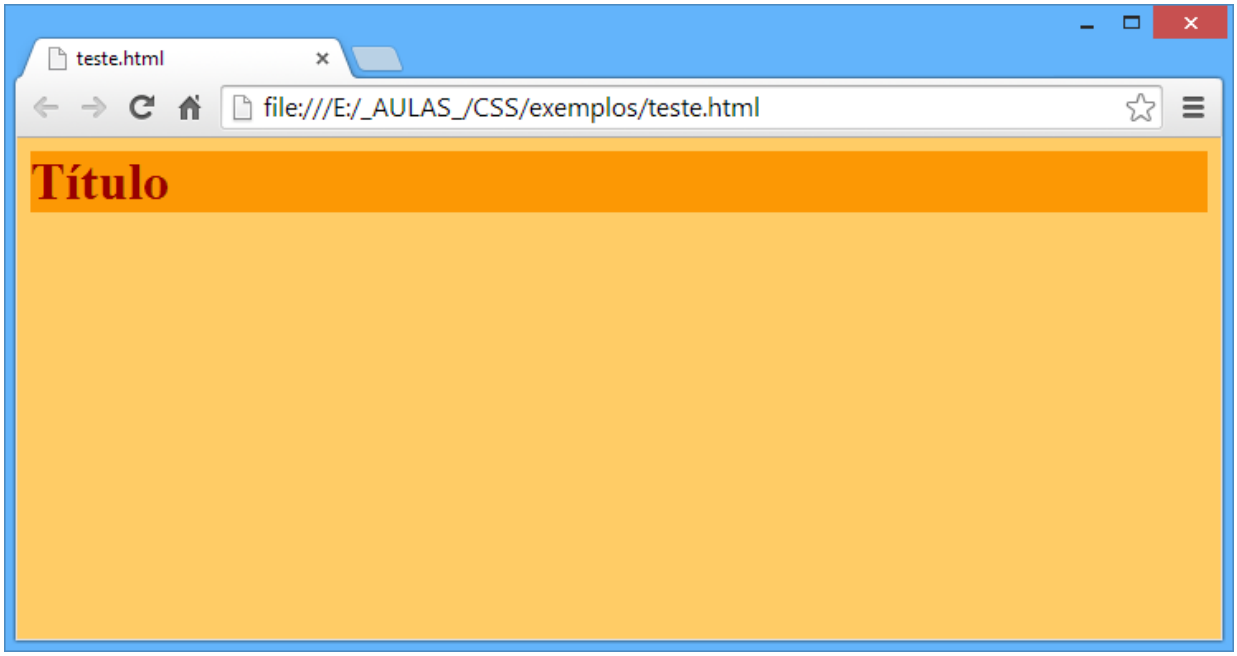
A propriedade background-color define a cor do fundo de um elemento.

O elemento <body> contém todo o conteúdo de um documento HTML. Assim, para mudar a cor de fundo da página, deve-se aplicar a propriedade **background-color** ao elemento <body>.

Pode-se aplicar cores de fundo para outros elementos, inclusive para cabeçalhos e textos. No exemplo abaixo foram aplicadas diferentes cores de fundo para os elementos <body> e <h1>.

```
body {
      background-color: #FFCC66;
}
h1 {
      color: #990000;
      background-color: #FC9804;
}
```

Obs.: Foram aplicadas duas propriedades ao elemento <h1> separadas por um ponto e vírgula.



3.3 - Images de fundo [background-image]

A propriedade CSS background-image é usada para definir uma imagem de fundo.

Para inserir uma imagem de fundo na página basta aplicar a propriedade **background-image** ao elemento <body> e especificar o caminho para onde está gravada a imagem.

```
body {
      background-color: #FFCC66;
      background-image: url("imagem.jpg");
}
h1 {
      color: #990000;
      background-color: #FC9804;
}
```

Obs.: Foi especificado o caminho para a imagem usando **url("imagem.jpg")**. Isto significa que a imagem está localizada no mesmo diretório da folha de estilos. Pode ser escolhido um outro diretório para gravar as imagens e o caminho seria **url("../images/imagem.jpg")** ou até mesmo hospedá-la na Internet: **url("http://www.html.net/imagem.jpg")**.

3.3 - Imagem de fundo repetida [background-repeat]

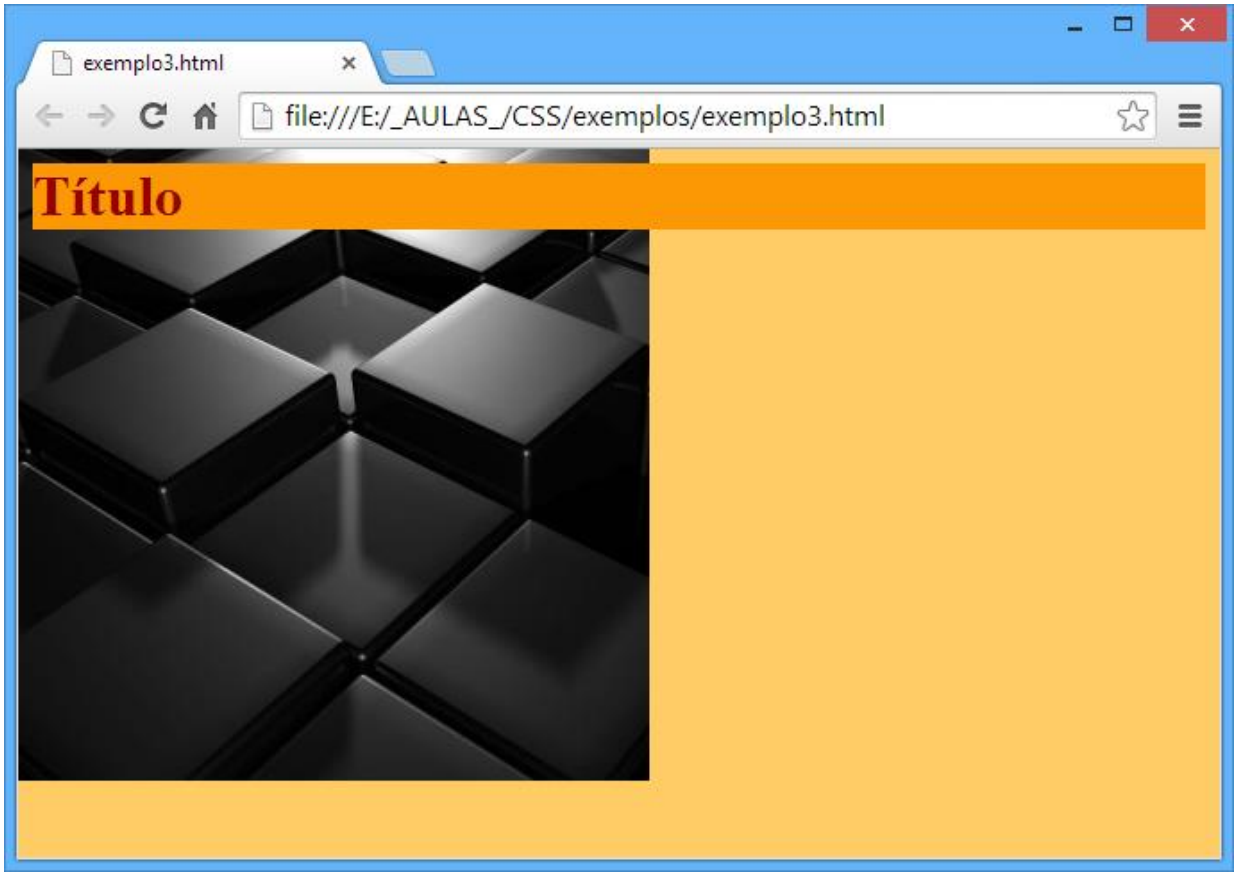
No exemplo anterior foi observado que a imagem se repetiu tanto na vertical como na horizontal cobrindo toda a tela? A propriedade background-repeat controla o comportamento de repetição da imagem de fundo.

A tabela a seguir mostra os quatro diferentes valores para background-repeat.

Valor	Descrição
Background-repeat: repeat-x	A imagem se repete na horizontal
Background-repeat: repeat-y	A imagem se repete na vertical
Background-repeat: repeat	A imagem se repete tanto na horizontal como na vertical
Background-repeat: no-repeat	A imagem não se repete

Por exemplo, o código mostrado a seguir é para que a imagem não se repita na tela:

```
body {
      background-color: #FFCC66;
      background-image: url("blocos.jpg");
      background-repeat: no-repeat;
}
h1 {
      color: #990000;
      background-color: #FC9804;
}
```



3.4 - Imagem de fundo fixa [background-attachment]

A propriedade background-attachment define se a imagem será fixa ou se irá rolar juntamente com o elemento que a contém.

Uma imagem de fundo fixa permanece no mesmo lugar e não rola com a tela ao contrário da imagem que não é fixa e rola acompanhando o conteúdo da tela.

A tabela a seguir mostra os dois diferentes valores para background-attachment. Veja os exemplos para constatar a diferença entre imagem fixa e imagem que rola.

Valor	Descrição
Background-attachment: scroll	A imagem rola com a página
Background-attachment: fixed	A imagem é fixa

Por exemplo, o código abaixo fixa a imagem na tela.

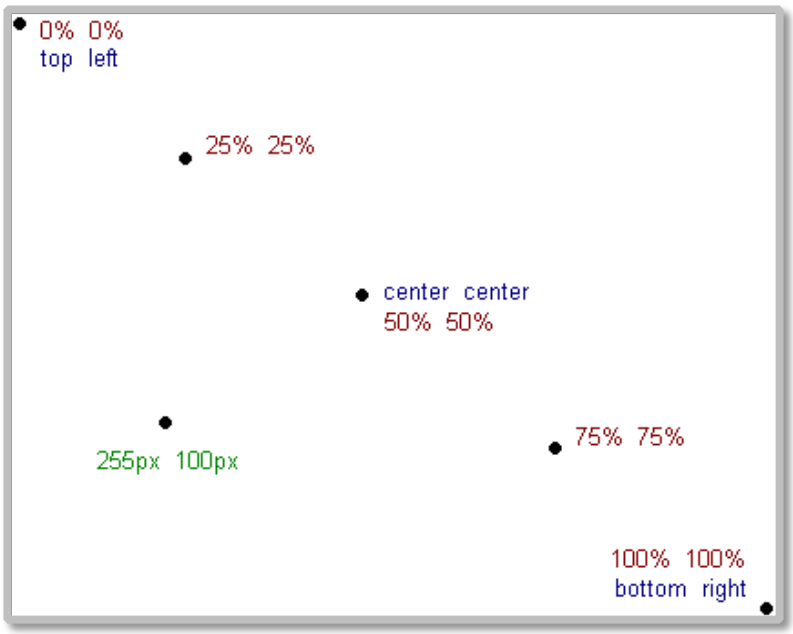
```
body {
      background-color: #FFCC66;
      background-image: url("blocos.jpg");
      background-repeat: no-repeat;
      background-attachment: fixed;
}
h1 {
      color: #990000;
      background-color: #FC9804;
}
```

3.5 - Posição da imagem de fundo [background-position]

Por padrão uma imagem de fundo é posicionada no canto superior esquerdo da tela. A propriedade background-position permite alterar este posicionamento padrão e colocar a imagem em qualquer lugar na tela.

Existem várias maneiras de definir o posicionamento da imagem na tela definindo valores para background-position. Todas elas se utilizam de um sistema de coordenadas. Por exemplo, os valores '100px 200px' posiciona a imagem a 100px do topo e a 200px do lado esquerdo da janela do navegador.

As coordenadas podem ser expressas em percentagem da largura da janela, em unidades fixas (pixels, centímetros, etc.) ou pode-se usar as palavras **top**, **bottom**, **center**, **left** e **right**. A figura a seguir ilustra o modelo de coordenadas:

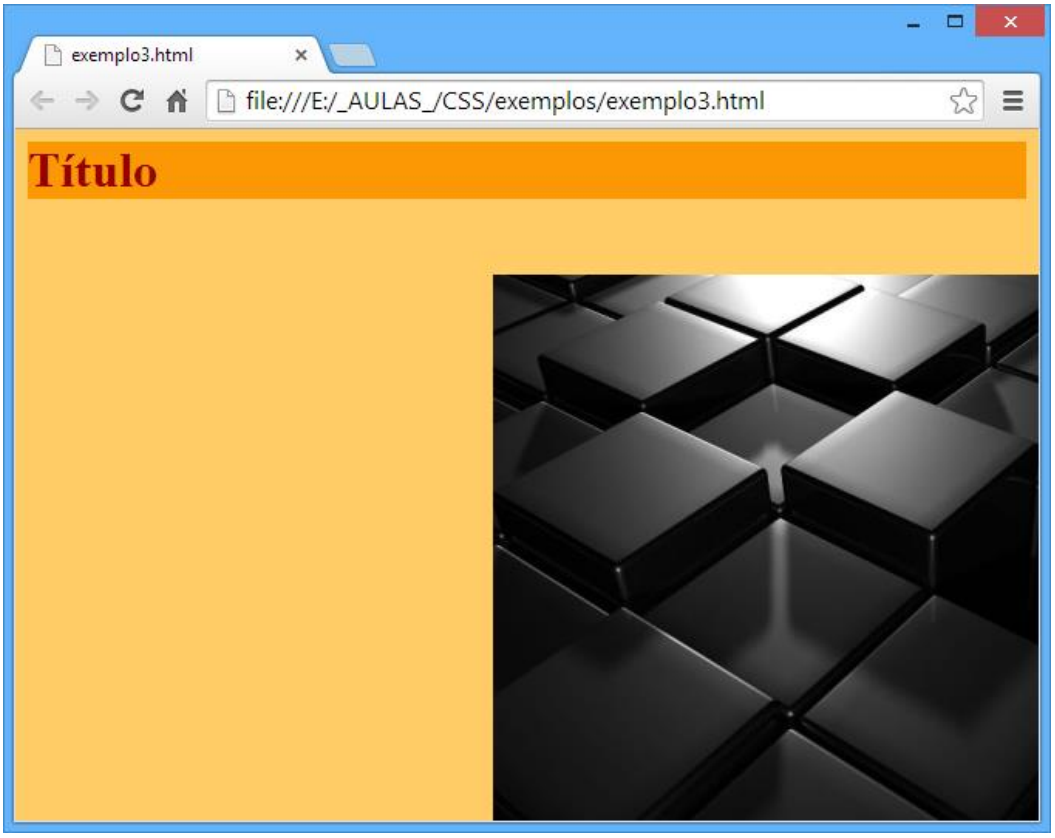


Na tabela a seguir são mostrados alguns exemplos.

Valor	Descrição
background-position: 2cm 2cm	A imagem é posicionada a 2cm da esquerda e 2cm para baixo na página
background-position: 50% 25%	A imagem é centrada na horizontal e a um quarto (25%) para baixo na página
Background-position: top right	A imagem é posicionada no canto superior direito da página

No exemplo de código a seguir a imagem é posicionada no canto inferior direito da página:

```
body {
      background-color: #FFCC66;
      background-image: url("blocos.jpg");
      background-repeat: no-repeat;
      background-attachment: fixed;
      background-position: right bottom;
}
h1 {
      color: #990000;
      background-color: #FC9804;
}
```



3.6 - Compilando [background]

A propriedade background é uma abreviação para todas as propriedades listadas anteriormente.

Com background você declara várias propriedades de modo abreviado, economizando digitação e alguns bites, além de tornar a folha de estilo mais fácil de se ler e entender.

Por exemplo, observe as cinco linhas a seguir:

```
background-color: #FFCC66;
background-image: url("blocos.jpg");
background-repeat: no-repeat;
background-attachment: fixed;
background-position: right bottom;
}
```

Usando background consegue-se o mesmo resultado, abreviando como mostrado abaixo:

```
background: #FFCC66 url("imagem.gif") no-repeat fixed right bottom;
```

A declaração abreviada deve seguir a seguinte ordem:

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

Se uma das propriedades não for declarada ela assume automaticamente o seu valor default. Por exemplo, a propriedade background-attachment e background-position não foram declaradas no código mostrado a seguir:

```
background: #FFCC66 url("imagem.gif") no-repeat;
```

As duas propriedades não declaradas assumirão o valor default que como já foi explicado: a imagem rola na tela e será posicionada no canto superior esquerdo (que são os valores default para as propriedades não declaradas).

4. Fontes

Será abordado as fontes e como aplicá-las usando CSS. Serão definidos como criar situações para que determinada fonte seja visualizada pelo usuário mesmo não estando instalada em seu sistema operacional.

4.1 - Família de fontes [font-family]

A propriedade font-family é usada para definir uma lista de fontes e sua prioridade para apresentação de um elemento em uma página. Se a primeira fonte da lista não estiver instalada na máquina do usuário, deverá ser usada a segunda e assim por diante até ser encontrada uma fonte instalada.

Existem dois tipos de nomes para definir fontes: nomes para famílias de fontes e nomes para famílias genéricas. Os dois são explicados a seguir:

nome para famílias de fontes
Exemplos para este tipo (normalmente conhecidas como "font") são "Arial", "Times New Roman" ou "Tahoma".

nome para famílias genéricas
Famílias genéricas são fontes que pertencem a um grupo com aparência uniforme. Um exemplo são as fontes sans-serif que englobam a coleção de fontes que "não têm pé".

Times New Roman
Garamond
Georgia

→ Estas três famílias de fontes pertencem à família genérica **serif**. Elas se caracterizam por terem "pé".

Trebuchet
Arial
Verdana

→ Estas três famílias de fontes pertencem à família genérica **sans-serif.** Elas se caracterizam por não terem "pé".

Courier
Courier New
Andele Mono

→ Estas três famílias de fontes pertencem à família genérica **monospace.** Elas se caracterizam por terem todos seus caracteres com uma largura fixa.

Ao listar fontes para o website, comece com aquela preferida, seguindo-se algumas alternativas para ela. É recomendável encerrar a listagem das fontes com uma fonte genérica. Assim fazendo, em último caso a página será renderizada com fonte da mesma família das que foram especificadas quando todas as demais estiverem indisponíveis na máquina do usuário.

A seguir é mostrado um exemplo de listagem de fontes:

```
h1 {font-family: arial, verdana, sans-serif;}
h2 {font-family: "Times New Roman", serif;}
```

Cabeçalhos <h1> serão renderizados com fonte "Arial". Se o usuário não tiver a font Arial instalada, será usada a fonte "Verdana". Se ambas estiverem indisponíveis na máquina do usuário será usada uma fonte da família sans-serif.

Nota-se que para especificar a fonte "Times New Roman" foram usadas aspas. Isto é necessário para fontes com nomes compostos e que contenham espaços entre os nomes.

4.2 - Estilo da fonte [font-style]

A propriedade font-style define a escolha da fonte em **normal**, **italic** ou **oblique**. No exemplo a seguir todos as cabeçalhos <h2> serão em itálico.

```
h1 {font-family: arial, verdana, sans-serif;}
h2 {font-family: "Times New Roman", serif; font-style: italic;}
```

4.3 - Fonte variante [font-variant]

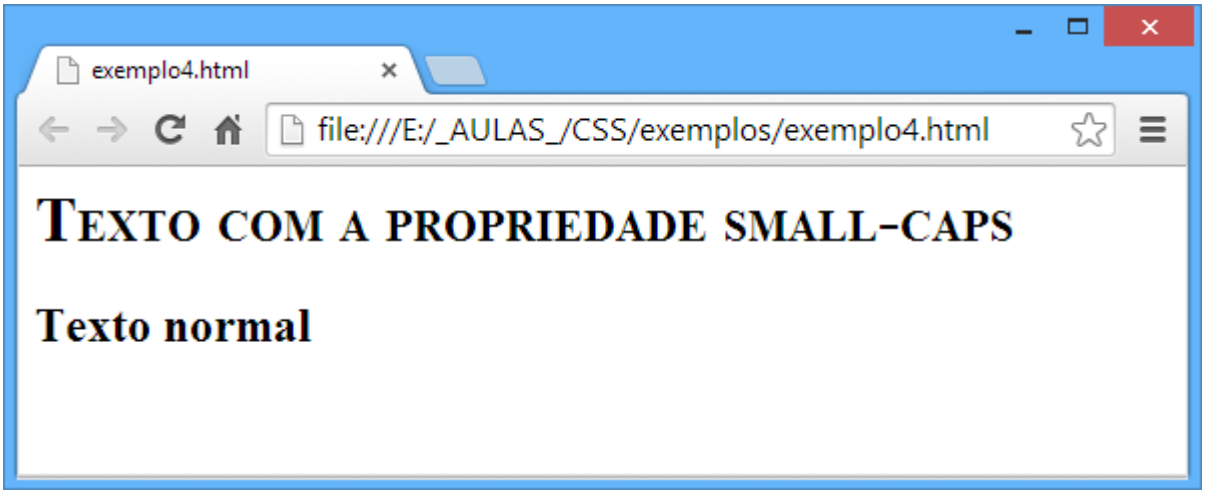
A propriedade font-variant é usada para escolher as variantes normal ou **smallcaps**.

Uma fonte small-caps é aquela que usa letras maiúsculas de tamanhos reduzidos, como são ilustrados nos exemplos a seguir:

Sans Book SC	Sans Bold SC	Serif Book SC	Serif Bold SC
ABCABC	**ABCABC**	ABCABC	**ABCABC**

Se a propriedade font-variant for definida para small-caps e não estiver disponível na máquina do usuário, será usada fonte em maiúscula.

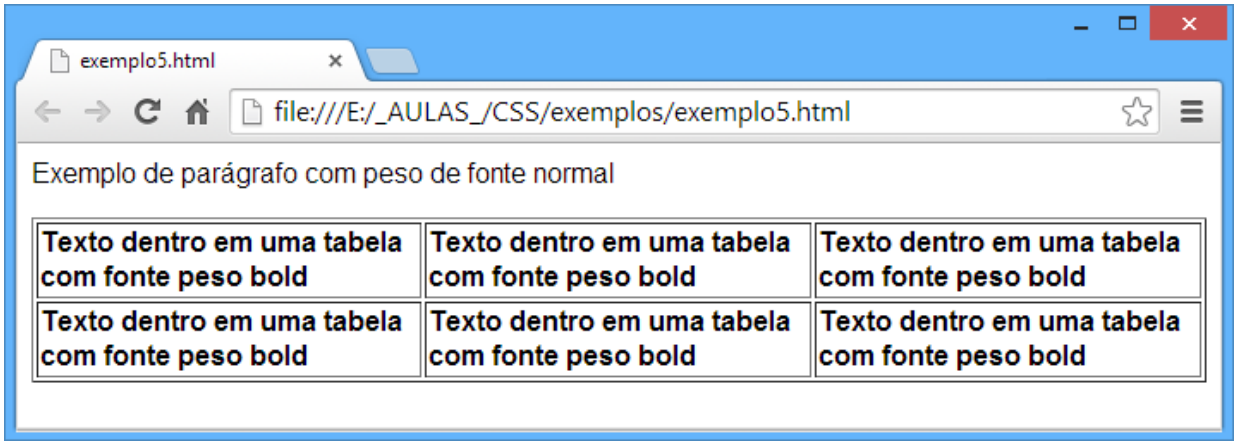
```
h1 {font-variant: small-caps;}
h2 {font-variant: normal;}
```



4.4 - Peso da fonte [font-weight]

A propriedade font-weight define se a fonte será o quão negrito. Uma fonte pode ser **normal** ou **bold**. Alguns navegadores suportam números de 100-900 (em intervalos de 100 em 100) para definir o peso da fonte.

```
p {font-family: arial, verdana, sans-serif;}
td {font-family: arial, verdana, sans-serif; font-weight: bold;}
```



4.5 - Tamanho da fonte [font-size]

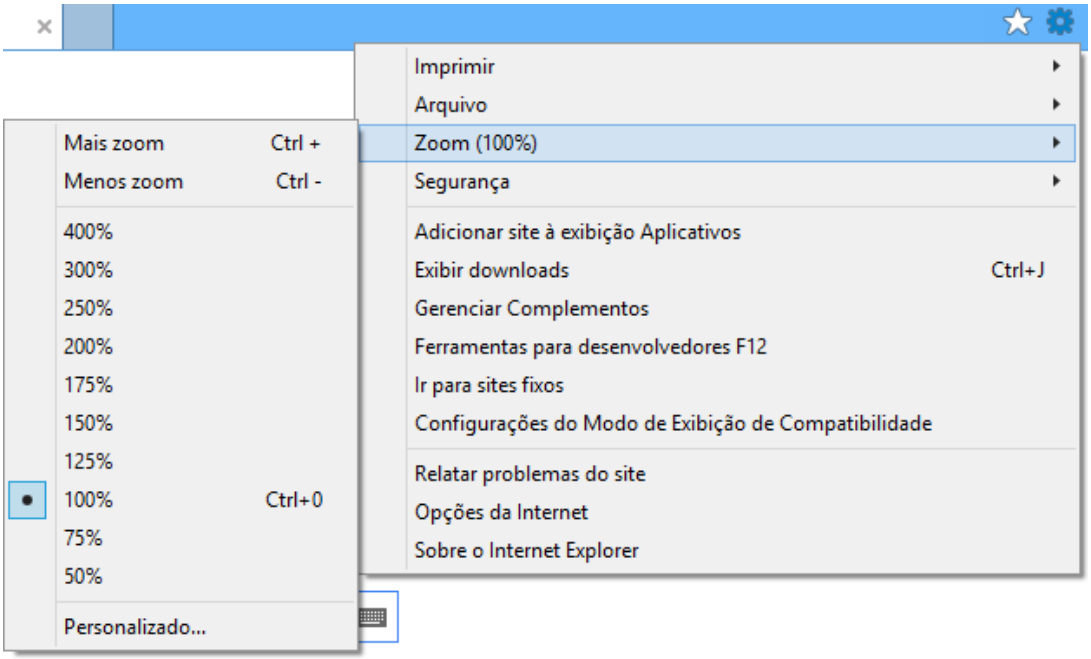
O tamanho da fonte é definido pela propriedade font-size.

Existem muitas unidades (pixels e porcentagens) que podem ser usadas para definir o tamanho da fonte. Vejamos exemplos a seguir:

```
h1 {font-size: 30px;}
h2 {font-size: 12pt;}
h3 {font-size: 120%;}
p {font-size: 1em;}
```

Existe uma diferença fundamental entre as quatro unidades adotadas no exemplo acima. As unidades 'px' e 'pt' são absolutas, enquanto '%' e 'em' permitem ao usuário ajustar o tamanho das fontes ao seu gosto e necessidade. Para fazer um site acessível a todos, deve-se usar unidades com '%' ou 'em'.

Abaixo uma figura mostrando como ajustar o tamanho das fontes no navegador Internet Explorer.



4.6 - Compilando [font]

Usar font é uma abreviação que permite definir várias propriedades em uma só. No exemplo a seguir quatro linhas de código usadas para definir propriedades de fonte para um parágrafo <p>:

```
p {
      font-style: italic;
      font-weight: bold;
      font-size: 30px;
      font-family: arial, sans-serif;
}
```

Usar a abreviação simplifica o código como mostrado abaixo:

```
p {
      font: italic bold 30px arial, sans-serif;
}
```

A ordem dos valores para font é a mostrada a seguir :

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

5. Textos

Formatar e estilizar textos é um item chave para qualquer web designer. Aqui será apresentado às interessantes oportunidades que as CSS proporcionam para adicionar layout aos textos. Serão discutidas as propriedades listadas abaixo:

5.1 - Indentação de texto [text-indent]

A propriedade text-indent permite que você aplique um recuo à primeira linha de um parágrafo. No exemplo a seguir um recuo de 30px é aplicado à todos os textos

```
p {
      text-indent: 30px;
}
```

5.2 - Alinhamento de textos [text-align]

A propriedade text-align corresponde ao atributo align das antigas versões do HTML.

Textos podem ser alinhados à esquerda (**left**), à direita (**right**) ou centrados (**center**). E temos ainda o valor **justify** que faz com o texto contido em uma linha se estenda tocando as margens esquerda e direita. Este tipo de alinhamento é usado em jornais e revistas.

No exemplo a seguir o texto contido na célula de cabeçalho <th> é alinhado à direita e os contidos nas células de dados <td> são centrados. E, os textos normais em parágrafos são justificados:

```
th {
text-align: right;
}
td {
text-align: center;
}
p {
text-align: justify;
}
```

5.3 - Decoração de textos [text-decoration]

A propriedade text-decoration possibilita adicionar "efeitos" ou "decoração" em textos. Você pode por exemplo, sublinhar textos, cortar o texto com uma linha, colocar uma linha sobre o

texto, etc. No exemplo a seguir os cabeçalhos <h1> são sublinhados, os cabeçalhos <h2> levam um linha em cima e os cabeçalhos <h3> são cortados por uma linha.

```
h1 {
text-decoration: underline;
}
h2 {
text-decoration: overline;
}
h3 {
text-decoration: line-through;
}
```

5.4 - Espaço entre letras [letter-spacing]

O espaçamento entre os caracteres de um texto é controlado pela propriedade letterspacing. O valor desta propriedade define o espaço entre os caracteres. Por exemplo, se você deseja um espaço de **3px** entre as letras do texto de um parágrafo <p> e de **6px** entre as letras do texto de um cabeçalho <h1> o código a seguir deverá ser usado.

```
h1 {
letter-spacing: 6px;
}
p {
letter-spacing: 3px;
}
```

5.5 - Transformação de textos [text-transform]

A propriedade text-transform controla a capitalização (tornar maiúscula) do texto. Você pode escolher **capitalize**, **uppercase** ou **lowercase** independentemente de como o texto foi escrito no código HTML.

Como exemplo tomamos a palavra "cabeçalho" que pode ser apresentada ao usuário como "CABEÇALHO" ou "Cabeçalho". São quatro os valores possíveis para texttransform:

capitalize

Capitaliza a primeira letra de cada palavra. Por exemplo: "john doe" transforma-se para "John Doe".

uppercase

Converte todas as letras para maiúscula. Por exemplo: "john doe" transforma-se para"JOHN DOE".

lowercase

Converte todas as letras para minúscula. Por exemplo: "JOHN DOE" transforma-se para"john doe".

none

Sem transformações - o texto é apresentado como foi escrito no código HTML. Para exemplificar vamos usar uma lista de nomes. Os nomes estão marcados com o elemento <li> (item de lista). Vamos supor que desejamos os nomes capitalizados e os cabeçalhos em letras maiúsculas.

Ao consultar o exemplo sugerido para este código dê uma olhada no HTML da página e observe que os textos no código foram escritos com todas as letras em minúsculas.

```
h1 {
text-transform: uppercase;
}
li {
text-transform: capitalize;
}
```

5.5 - Efeitos de textos

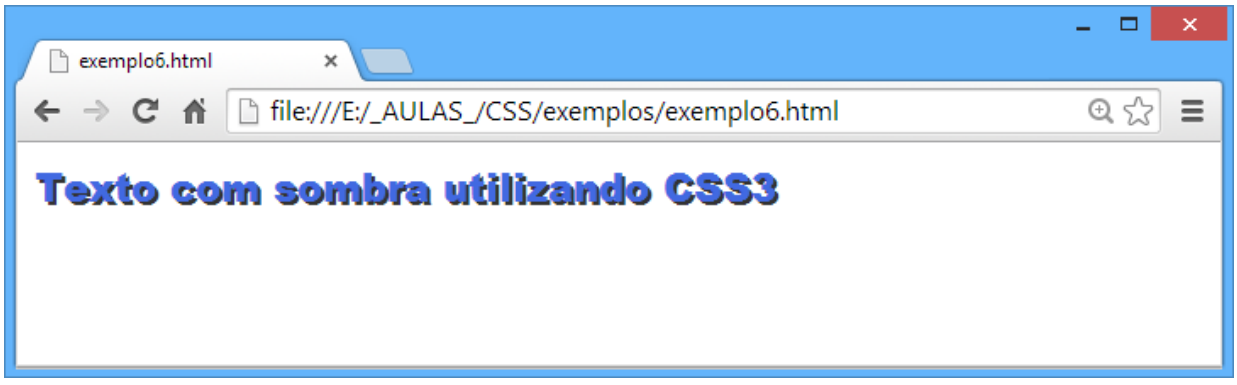
A versão 3 das CSS possuem novos recursos criativos que facilitam o desenvolvimento de uma página evitando a utilização de imagens para isso.

5.5.1 -Texto com sombras

O CSS nível 3 tem uma propriedade chamada 'texto com sombras' para adicionar uma sombra a cada letra do texto. Na sua forma mais simples, é uma algo como isto:

```
h2 {text-shadow: 0.1em 0.1em #333}
```

Foi adicionada uma sombra na cor cinza escuro(#333) um pouco à direita(0.1em) e baixo(0.1em) relativo ao texto normal: O resultado é este:

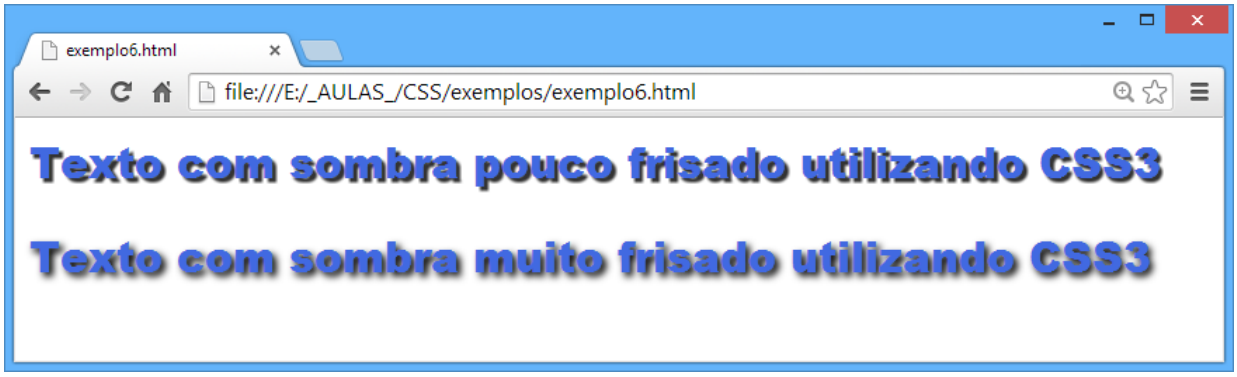


5.5.2 - Texto com sombras frisado

A forma simples da propriedade do 'texto com sombras' tem duas partes: a cor e uma compensação (0.1em 0.1em no exemplo anterior). Isto resulta numa sombra fina indicada na compensação. Mas a compensação também pode ser frisada, resultando em sombras mais ou menos turva.

A quantidade do frisado é dado como uma compensação. Aqui estão duas linhas, uma com pouco frisado(0.05em) outra com muito frisado(0.2em):

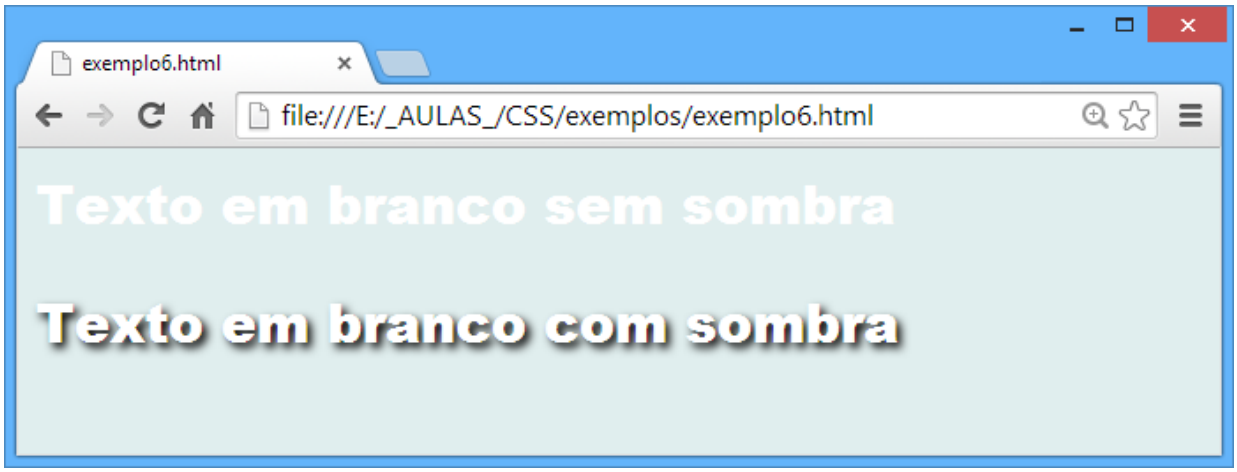
```
h2.a {text-shadow: 0.1em 0.1em 0.05em #333}

h2.b {text-shadow: 0.1em 0.1em 0.2em black}
```



5.5.3 - Texto com sombras frisado

As sombras podem tornar um texto mais legível se o contraste entre o primeiro plano e o fundo é pequeno. Aqui está um exemplo de texto branco em fundo azul claro, primeiro sem as sombras e depois com elas:

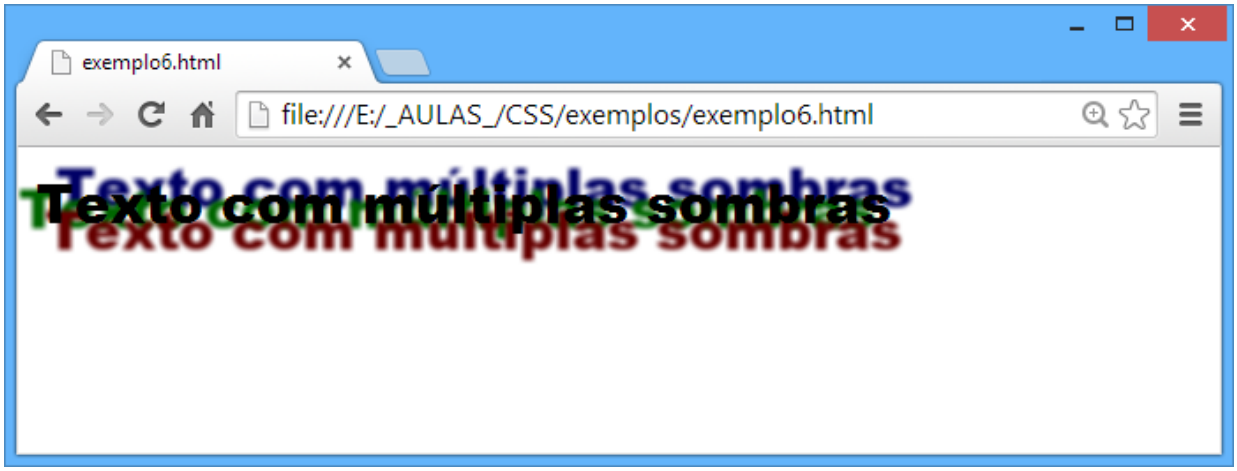
```
h2 {color: white}

h2.a {color: white; text-shadow: black 0.1em 0.1em
```



5.5.3 – Sombras múltiplas

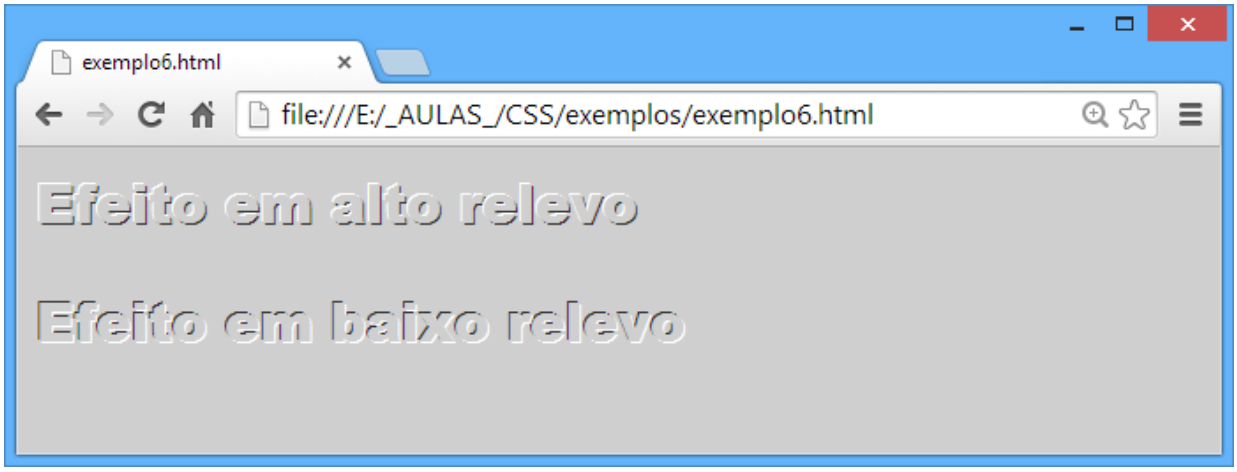
O texto pode ter também mais de uma sombra aplicada a ele.

```
h2 {
       text-shadow: 0.2em 0.5em 0.1em #600,
         -0.3em 0.1em 0.1em #060,
         0.4em -0.3em 0.1em #006
   }
```



Mas com duas sombras escuras e claras bem colocadas, o efeito pode ser útil:

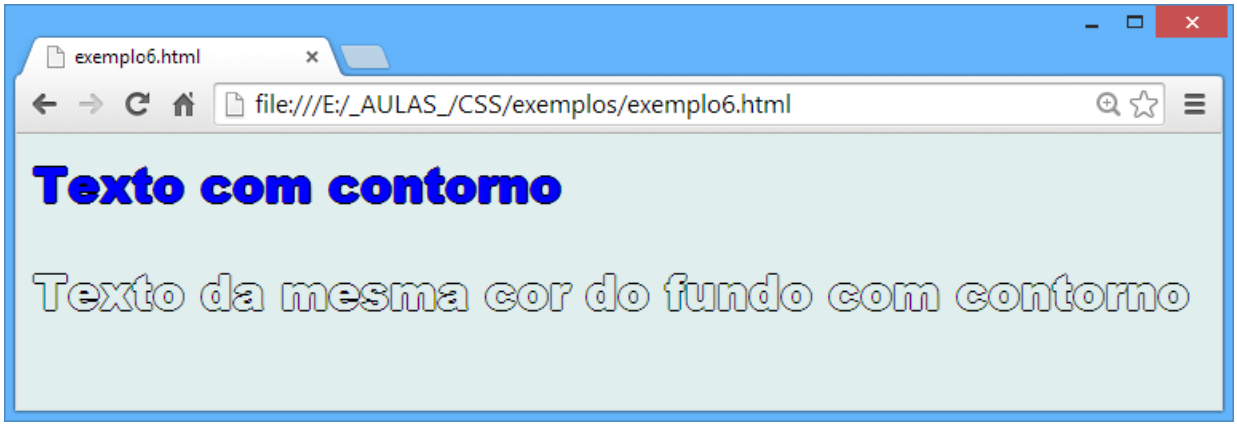
```
h2.a {text-shadow: -1px -1px white, 1px 1px #333}

h2.b {text-shadow: 1px 1px white, -1px -1px #333}
```



5.5.4 – Letras com contorno

O exemplo das duas sombras do exemplo anterior pode ser levada mais longe. Com quatro sombras, é possível fazer um contorno nas letras:

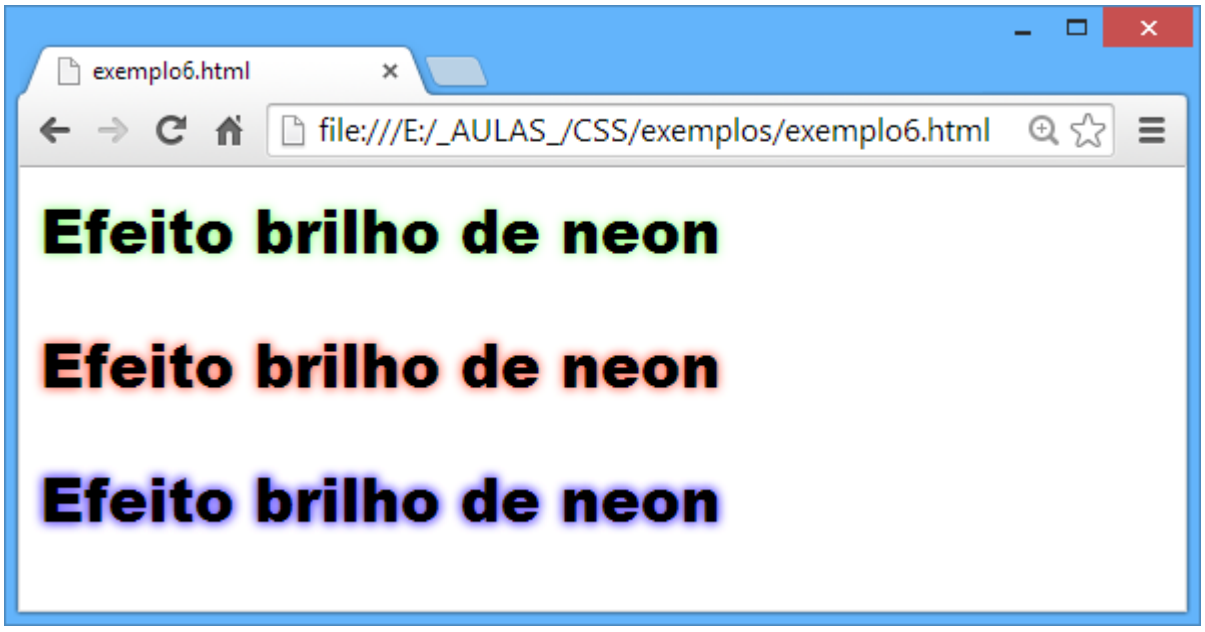
```
h2 {text-shadow: -1px 0 black, 0 1px black,
                 1px 0 black, 0 -1px black}
```



5.5.5 - Efeito de neon

Se puseres uma sombra frisada mesmo por trás do texto,i.e., com a compensação zero, o efeito é para criar um brilho em volta das letras. Se o brilho de uma sombra única nem é intenso o suficiente, repete a mesma sombra mais vezes:

```
h3.a {text-shadow: 0 0 0.2em #8F7}

h3.b {text-shadow: 0 0 0.2em #F87, 0 0 0.2em #F87}

h3.c {text-shadow: 0 0 0.2em #87F, 0 0 0.2em #87F,
           0 0 0.2em #87F}
```



6. Links

Você pode aplicar aos links tudo que aprendeu nas lições anteriores (mudar cores, fontes, sublinhados, etc). A novidade aqui é que você pode definir as propriedades de maneira diferenciada de acordo com o estado do link ou seja visitado, não visitado, ativo ou com o ponteiro do mouse sobre o link. Isto possibilita adicionar interessantes efeitos ao seu website. Para estilizar estes efeitos você usará as chamadas pseudoclasses.

6.1 - O que é pseudo-classe?

Uma pseudo-classe permite estilizar levando em conta condições diferentes ou eventos ao definir uma propriedade de estilo para uma tag HTML.

Vamos ver um exemplo. Como você já sabe, links são marcados no HTML com tags <a>. Podemos então usar a como um seletor CSS:

```
a {
      color: blue;
}
```

Um link pode ter diferentes estados. Por exemplo, pode ter sido visitado ou não visitado. Você usará pseudo-classes para estilizar links visitados e não visitados.

```
a:link {
      color: blue;
}
a:visited {
      color: red;
}
```

Use as pseudo-classes a:link e a:visited para estilizar links não visitados e visitados respectivamente. Links ativos são estilizados com a pseudo-classe a:active e a:hover, esta última é a pseudo-classe para links com o ponteiro do mouse sobre ele.

A seguir será abordado mais detalhes e exemplificação, as quatro pseudo-classes.

6.1.1 - Pseudo-classe: link

A pseudo-classe: link é usada para links não visitados.

No exemplo a seguir links não visitados serão na cor verde.

```
a:link {
      color: green;
}
```

6.1.2 - Pseudo-classe: visited

A pseudo-clases: visited é usada para links visitados. No exemplo a seguir links visitados serão na cor amarela:

```
a:visited {
      color: yellow;
}
```

6.1.3 - Pseudo-classe: active

A pseudo-classe: active é usada para links ativos. No exemplo a seguir links ativos terão seu fundo na cor vermelha:

```
a:active {
      background-color: red;
}
```

6.1.4 - Pseudo-classe: hover

A pseudo-classe: hover é usada para quando o ponteiro do mouse está sobre o link.

Isto pode ser usado para conseguir efeitos bem interessantes. Por exemplo, podemos mudar a cor do link para laranja e o texto para itálico quando o ponteiro do mouse passa sobre ele, o código CSS para estes efeitos é o mostrado a seguir:

```
a:hover {
      color: orange;
      font-style: italic;
}
```

6.2 - Exemplos de efeitos de links

Exemplo 1: Efeito quando o ponteiro está sobre o link

É comum a criação de efeitos diferentes quando o ponteiro está sobre o link. A seguir alguns exemplos extras de estilização da pseudo-classe: hover.

Exemplo 1a: Espaçamento entre as letras

Como você deve estar lembrado da lição anterior, o espaçamento entre as letras de um texto pode ser controlado pela propriedade letter-spacing. Isto pode ser aplicado aos links para obter um efeito interessante:

```
a:hover {
      letter-spacing: 10px;
      font-weight:bold;
      color:red;
}
```

Exemplo 1b: UPPERCASE e lowercase

Anteriormente foi abordado a propriedade text-transform, para estilizar com letras maiúsculas e minúsculas. Isto pode ser usado para estilizar links:

```
a:hover {
      text-transform: uppercase;
      font-weight:bold;
      color:blue;
      background-color:yellow;
}
```

Exemplo 2: Removendo sublinhado dos links

Uma pergunta comum: Como remover o sublinhado dos links?

Deve-se estudar com muito cuidado a necessidade de retirar o sublinhado dos links, pois isto poderá reduzir significativamente a usabilidade do website. As pessoas estão acostumadas com links na cor azul e sublinhados e sabem que ali há um texto a ser clicado. Se ao mudar a cor e retirar o sublinhado dos links, poderá confundir os visitantes e em consequência não retirar o máximo dos conteúdos do website.

Feita esta ressalva, é muito fácil retirar o sublinhado dos links. Conforme explicado anteriormente, a propriedade text-decoration pode ser usada para definir se o texto é ou não sublinhado. Para remover o sublinhado, basta definir o valor none para a propriedade text-decoration.

```
a {
      text-decoration:none;
}
```

Alternativamante, pode-se definir text-decoration juntamente com outras propriedades para as quatro pseudo-classes.

```
a:link {
      color: blue;
      text-decoration:none;
}

a:visited {
      color: purple;
      text-decoration:none;
}

a:active {
      background-color: yellow;
      text-decoration:none;
}

a:hover {
      color:red;
      text-decoration:none;
}
```

7. Identificando e agrupando elementos

Em alguns casos você deseja aplicar estilos a um elemento ou grupo de elementos em particular. Veremos como usar class e id para estilizar elementos.

Como definir uma cor para um determinado cabeçalho, diferente da cor usada para os demais cabeçalhos do website? Como agrupar links em diferentes categorias e estilizar cada categoria diferentemente? Estas são algumas das questões que vamos ver agora.

7.2 - Agrupando elementos com uso de classe

Vamos supor que temos duas listas de links para diferentes tipos de uvas usadas na produção de vinho branco e de vinho tinto. O código HTML conforme mostrado abaixo:

```
<p>Uvas para vinho branco:</p>
  <ul>
       <li><a href="ri.htm">Riesling</a></li>
       <li><a href="ch.htm">Chardonnay</a></li>
       <li><a href="pb.htm">Pinot Blanc</a></li>
  </ul>

<p>Uvas para vinho tinto:</p>
  <ul>
       <li><a href="cs.htm">Cabernet Sauvignon</a></li>
       <li><a href="me.htm">Merlot</a></li>
       <li><a href="pn.htm">Pinot Noir</a></li>
  </ul>
```

Queremos que os links para vinho branco sejam na cor amarela, para vinho tinto na cor vermelha e os demais links na página permaneçam na cor azul.

Para conseguir isto, dividimos os links em duas categorias. Isto é feito atribuindo uma classe para cada link, usando o atributo class.

Vamos especificar esta classe no exemplo a seguir:

```
<p>Uvas para vinho branco:</p>
  <ul>
    <li><a href="ri.htm" class="whitewine">Riesling</a></li>
    <li><a href="ch.htm" class="whitewine">Chardonnay</a></li>
    <li><a href="pb.htm" class="whitewine">Pinot Blanc</a></li>
  </ul>

<p>Uvas para vinho tinto:</p>
  <ul>
    <li><a href="cs.htm" class="redwine">Cabernet Sauvignon</a></li>
    <li><a href="me.htm" class="redwine">Merlot</a></li>
    <li><a href="pn.htm" class="redwine">Pinot Noir</a></li>
</ul>
```

Agora podemos definir propriedades específicas para links pertencentes as classes whitewine e redwine, respectivamente.

```
a {
      color: blue;
}

a.whitewine {
      color: #FFBB00;
}

a.redwine {
      color: #800000;
}

<li><a href="cs.htm" class="redwine">Cabernet Sauvignon</a></li>
<li><a href="me.htm" class="redwine">Merlot</a></li>
<li><a href="pn.htm" class="redwine">Pinot Noir</a></li>
</ul>
```

Como mostrado no exemplo acima, pode-se definir propriedades para estilização dos elementos pertencentes a uma determinada classe usando um **.nomedaclasse** na folha de estilos do documento.

7.3 - Identificando um elemento com uso de id

Além de agrupar elementos podemos querer atribuir identificação a um único elemento. Isto é feito usando o atributo id.

O que há de especial no atributo id é que não poderá existir dois ou mais elementos com a mesma id, ou seja em um documento apenas um e somente um elemento poderá ter uma determinada id. Cada id é única. Para casos em que haja necessidade de mais de um elemento com a mesma identificação usamos o atributo class. A seguir um exemplo de possível uso de id:

```
<h1>Capítulo 1</h1>
...
<h2>Capítulo 1.1</h2>
...
<h2>Capítulo 1.2</h2>
...
<h1>Capítulo 2</h1>
...
<h2>Capítulo 2.1</h2>
...
<h3>Capítulo 2.1.2</h3>
...
```

O exemplo acima simula os cabeçalhos de um documento estruturado em capítulos e parágrafos. É comum atribuir uma id para cada capítulo como mostrado a seguir:

```
<h1 id="c1">Capítulo 1</h1>
...
<h2 id="c1-1">Capítulo 1.1</h2>
...
<h2 id="c1-2">Capítulo 1.2</h2>
...
<h1 id="c2">Capítulo 2</h1>
...
<h2 id="c2-1">CCapítulo 2.1</h2>
...
<h3 id="c2-1-2">Capítulo 2.1.2</h3>
...
```

Vamos supor que o cabeçalho do capítulo 1.2 deva ser na cor vermelha. Isto pode ser feito conforme mostrado na folha de estilo a seguir:

```
#c1-2 {
      color: red;
}
```

Como mostrado no exemplo acima, podemos definir propriedades para um elemento específico usando um seletor #id na folha de estilos para o documento.

8. Agrupando elementos (span e div)

Os elementos e são usados para agrupar e estruturar um documento e são freqüentemente usados em conjunto com os atributos class e id.

Nesta lição veremos com detalhes o uso dos elementos HTML e no que se refere a sua vital importância para as CSS.

- Agrupando com
- Agrupando com

8.1 - Agrupando com

O elemento é um elemento neutro e que não adiciona qualquer tipo de semântica ao documento. Contudo, pode ser usado pelas CSS para adicionar efeitos visuais a partes específicas do texto no seu documento.

Um exemplo deste uso é mostrado na citação abaixo:

```
<p>Dormir cedo e acordar cedo faz o homemsaudável, rico e sábio.</p>
```

Vamos supor que queremos enfatizar na cor vermelha os benefícios apontados na frase. Para isto marcamos os benefícios com . A cada span atribuímos uma class, e estilizamos na folha de estilos:

```
<p>Dormir cedo e acordar cedo faz o homem
<span class="beneficio">saudável</span>,
<span class="beneficio">rico</span>
e <span class="beneficio">sábio</span>.</p>
```

A folha de estilos:

```
span.beneficio {
      color:red;
}
```

É claro que você pode usar id para estilizar o elemento . Mas, como você deve estar lembrado, deverá usar uma única id para cada um os três elementos , conforme foi explicado na lição anterior.

8.2 - Agrupando com

Enquanto é usado dentro de um elemento nível de bloco como no exemplo anterior, é usado para agrupar um ou mais elementos nível de bloco. Diferenças à parte, o agrupamento com funciona mais ou menos da mesma maneira. Abaixo um exemplo tomando duas listas de presidentes dos Estados Unidos agrupados segundo suas filiações políticas:

```
<div id="democrats">
 <ul>
   <li>Franklin D. Roosevelt</li>
   <li>Harry S. Truman</li>
   <li>John F. Kennedy</li>
   <li>Lyndon B. Johnson</li>
   <li>Jimmy Carter</li>
   <li>Bill Clinton</li>
 </ul>
</div>

<div id="republicans">
 <ul>
   <li>Dwight D. Eisenhower</li>
   <li>Richard Nixon</li>
   <li>Gerald Ford</li>
   <li>Ronald Reagan</li>
   <li>George Bush</li>
   <li>George W. Bush</li>
 </ul>
</div>
```

E na folha de estilos, podemos agrupar a estilização da mesma maneira como fizemos no exemplo acima:

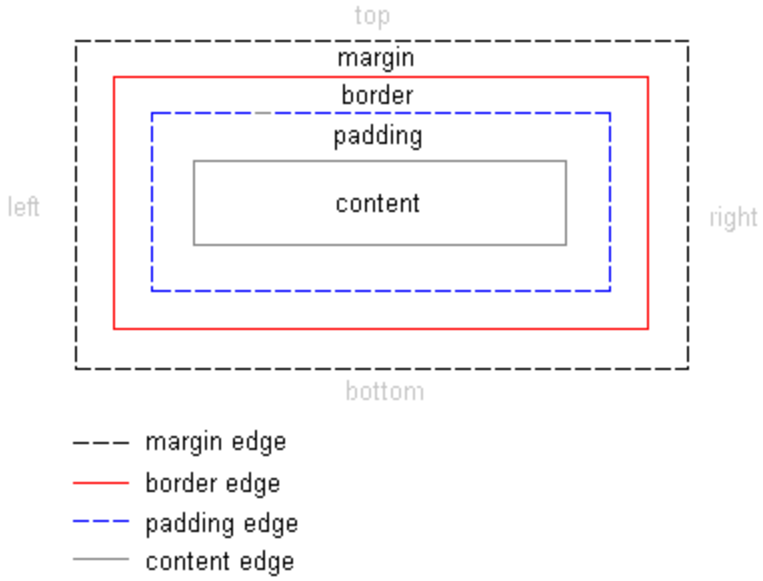
```
#democrats {
      background:blue;
}

#republicans {
      background:red;
}
```

Nos exemplos mostrados acima usamos somente e para simples estilizações, tais como cores de textos e de fundos. Contudos estes dois elementos possibilitam estilizações bem mais avançadas como veremos adiante nas lições deste tutorial.

9. Box Model

O box model (modelo das caixas) em CSS, descreve os boxes (as caixas) geradas pelos elementos HTML. O box model, detalha ainda, as opções de ajuste de margens, bordas, padding e conteúdo para cada elemento. Abaixo apresentamos um diagrama representando a estrutura de construção do box model:

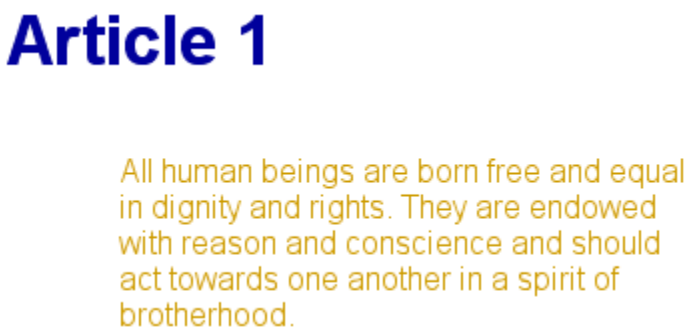
9.1 - O box model em CSS



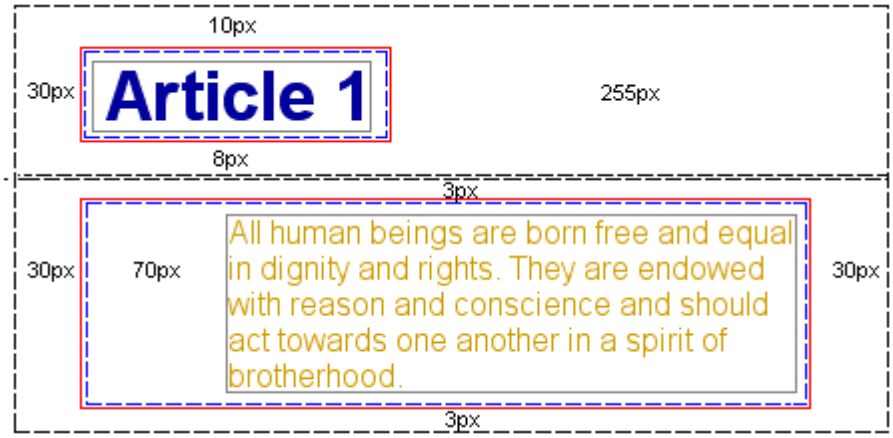
A ilustração acima é teórica. Vamos explicá-la na prática tomando como base um cabeçalho e um texto. O HTML para nosso exemplo (o texto foi retirado da Declaração Universal dos Direitos Humanos e está no original em inglês) é o mostrado abaixo:

```
<h1>Article 1:</h1>

<p>All human beings are born free
and equal in dignity and rights.
They are endowed with reason and conscience
and should act towards one another in a
spirit of brotherhood</p>
```

Definindo estilos para cores e fontes o exemplo pode ser apresentado como a seguir:



O exemplo contém dois elementos: <h1> e <p>. O box model para os dois elementos é mostrado a seguir:



Embora possa parecer um pouco complicado, a ilustração mostra como cada um dos elementos é contido em um box (uma caixa). Boxes que podem ser ajustados e controlados via CSS.

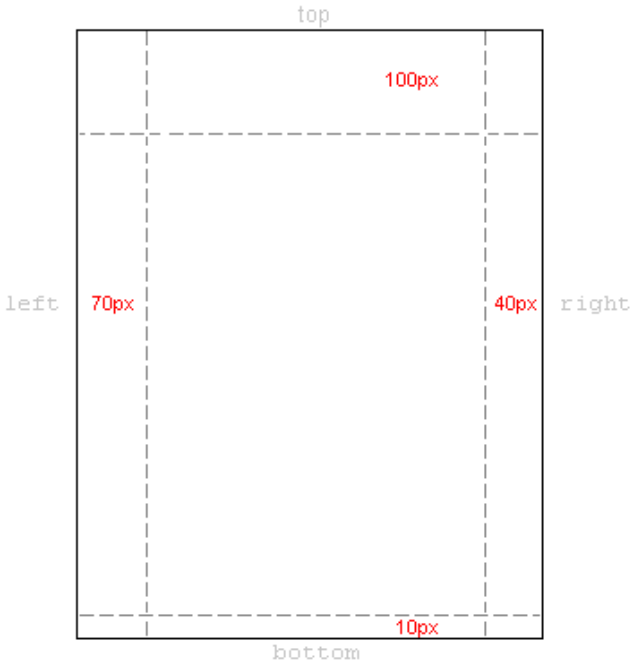
10. Margin e padding

Na lição anterior vimos o box model. Nesta lição veremos como controlar a apresentação de um elemento definindo as propriedades margin e padding.

10.1 - Definindo margin de um elemento

Um elemento tem quatro lados: right, left, top e bottom (direito, esquerdo, superior e inferior). A margin é a distância entre os lados de elementos vizinhos (ou às bordas do documento). Veja o diagrama da lição anterior.

Vamos começar com um exemplo mostrando como definir margins para o documento, ou seja, para o elemento <body>. A ilustração a seguir mostra como serão as margens da página.



As CSS são mostradas abaixo:

```
body {
      margin-top: 100px;
      margin-right: 40px;
      margin-bottom: 10px;
      margin-left: 70px;
}
```

Ou, adotando uma sintaxe mais elegante:

```
body {
     margin: 100px 40px 10px 70px;
}
```

As margens para a maioria dos elementos pode ser definida conforme o exemplo acima. Podemos então, por exemplo, definir margens para todos os parágrafos <p>:

```
body {
     margin: 100px 40px 10px 70px;
}

p {
     margin: 5px 50px 5px 50px;
}
```

10.2 - Definindo padding de um elemento

Padding pode também ser entendido como "enchimento". Isto faz sentido, porque padding não é computado na distância entre elementos, padding define simplesmente a distância entre a borda e o conteúdo do elemento.

Ilustramos o uso de padding através de um exemplo onde todos os cabeçalhos têm uma cor de fundo definida:

```
h1 {
     background: yellow;
}

h2 {
     background: orange;
}
```

Definindo padding para os cabeçalhos, alteramos a quantidade de enchimento existente ao redor de cada um deles:

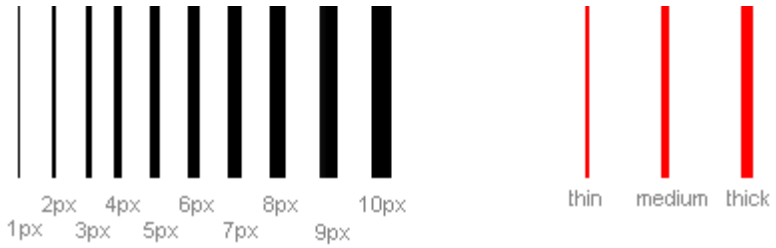
```
h1 {
     background: yellow;
     padding: 20px 20px 20px 80px;
}

h2 {
     background: orange;
     padding-left:120px;
}
```

11. Bordas

Bordas podem ser usadas para muitas coisas, por exemplo, como elemento decorativo ou para servir de linha de separação entre duas coisas. CSS proporciona infinitas possibilidades de uso de bordas na página.

11.1 - A espessura das bordas [border-width]

A espessura das bordas é definida pela propriedade border-width, que pode assumir os valores thin, medium, e thick (fina, média e grossa), ou um valor numérico em pixels. A figura a seguir ilustra algumas espessuras de bordas:



11.2 - As cores das bordas [border-color]

A propriedade border-color define as cores para as bordas. Os valores são expressos em código ou nome de cores, por exemplo, "#123456", "rgb(123,123,123)" ou"yellow".

11.3 - Tipos de bordas[border-style]

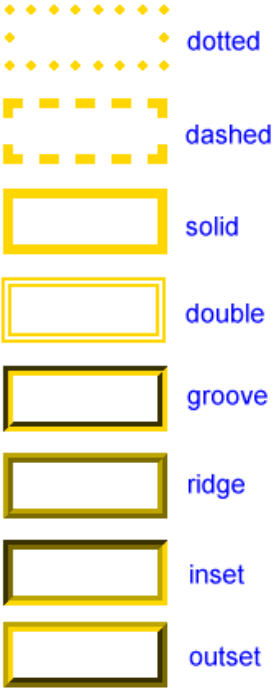
Existem vários tipos de bordas disponíveis para escolha. A seguir apresentamos 8 tipos diferentes de bordas e como elas são renderizadas Internet Explorer 5.5. Todos os exemplos são mostrados na cor "gold" e com espessura "thick", mas você pode usar qualquer cor e espessura ao seu gosto.

Os valores none ou hidden podem ser usados quando não se deseja a existência de bordas.

11.3.1 - Exemplos de definição de bordas

As três propriedades explicadas acima podem ser definidas juntas para cada elemento e resultam em diferentes bordas.

Para exemplificar, foram estilizadas diferentes bordas para os elementos <h1>, <h2>, <ul> e <p>. O resultado pode não ser uma obra prima, mas, ilustra bem algumas das inúmeras possibilidades de estilização de bordas:




```
h1 {
     border-width: thick;
     border-style: dotted;
     border-color: gold;
}

h2 {
     border-width: 20px;
     border-style: outset;
     border-color: red;
}

p {
     border-width: 1px;
     border-style: dashed;
     border-color: blue;
}

ul {
     border-width: thin;
     border-style: solid;
     border-color: orange;
}
```

É possível ainda definir propriedades especialmente para as bordas top, bottom, right ou left (superior, inferior, direita e esquerda). Veja o exemplo a seguir:

```
h1 {
     border-top-width: thick;
     border-top-style: solid;
     border-top-color: red;

     border-bottom-width: thick;
     border-bottom-style: solid;
     border-bottom-color: blue;

     border-right-width: thick;
     border-right-style: solid;
     border-right-color: green;

     border-left-width: thick;
     border-left-style: solid;
     border-left-color: orange;
}
```

11.4 - Compilando [border]

Assim como para muitas outras propriedades, você pode usar uma declaração abreviada para bordas. Vamos a um exemplo:

```
p {
     border-width: 1px;
     border-style: solid;
     border-color: blue;
}
```

Pode ser abreviada assim:

```
p {
     border: 1px solid blue;
}
```

12. Altura e largura

Até agora ainda não fizemos qualquer consideração sobre as dimensões dos elementos com que trabalhamos. Nesta lição veremos como é fácil atribuir uma altura e uma largura para um elemento.

12.1 - Atribuindo largura [width]

A propriedade width destina-se a definir a largura de um elemento. O exemplo a seguir constrói um box dentro do qual podemos digitar um texto:

```
div.box {
      width: 200px;
      border: 1px solid black;
      background: orange;
}
```

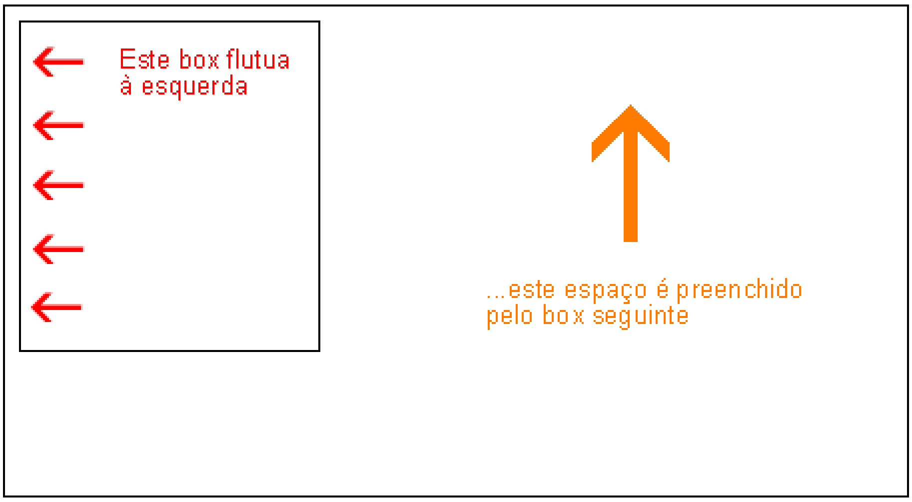
12.2 - Atribuindo altura [height]

No exemplo acima a altura será determinada pelo conteúdo inserido no box. Você pode definir a altura de um elemento com a propriedade height. Como exemplo, vamos fazer a altura do box anterior igual a 500px:

```
div.box {
      height: 500px;
      width: 200px;
      border: 1px solid black;
      background: orange;
}
```

13. Flutuando elementos (floats)

Um elemento pode ser 'flutuado' à esquerda ou à direita com uso da propriedade float. Isto significa que o box e seu conteúdo são deslocados para a direita ou para a esquerda do documento (ou do bloco container). A figura a seguir ilustra o princípio de float:



Se desejamos que um texto seja posicionado em volta de uma figura como mostrado abaixo, basta flutuarmos a imagem:

Uma imagem flutuada

textoxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxtexto

```
<div id="picture">
<img src="bandeiradobrasil.jpg" alt="Bandeira do Brasil">
</div>
<p>xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx,
xxxxxxxxxxxxxxxxxxxxxxxx...</p>
```

Para conseguir o efeito mostrado, basta definir uma largura para o box que o contém e declarar para ele float: left;

```
#picture {
      float:left;
      width: 100px;
}
```

13.1 - Outro exemplo : colunas

Floats podem ser usados para construir colunas em um documento. Para criar as colunas estruturamos as colunas no código HTML usando como mostrado a seguir:

```
<div id="column1">
      <p>xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx...</p>
</div>

<div id="column2">
      <p>xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx,
      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx...</p>
</div>

<div id="column3">
      <p>xxxxxxxxxxxxxxxxxxxxxxxxx
      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx...</p>
</div>
```

A seguir definimos a largura de cada coluna, por exemplo 33%, e declaramos float: left; para cada uma das colunas:

```
#column1 {
     float:left;
     width: 33%;
}

#column2 {
     float:left;
     width: 33%;
}

#column3 {
     float:left;
     width: 33%;
}
```

float pode ser declarado left, right ou none.

13.2 - A propriedade clear

A propriedade clear é usada para controlar o comportamento dos elementos que se seguem aos elementos floats no documento.

Por padrão, o elemento subsequente a um float, ocupa o espaço livre ao lado do elemento flutuado. Veja no exemplo acima que o texto deslocou-se automaticamente para o lado da foto de Bill Gates.

A propriedade clear pode assumir os valores left, right, both ou none. A regra geral é: se clear, for por exemplo definido both para um box, a margem superior deste box será posicionada sempre abaixo da margem inferior dos boxes flutuados que estejam antes dele no código.

```
<div id="picture">
     <img src="bandeiradobrasil.jpg" alt="Bandeira do Brasil">
</div>

<h1>Bandeira do Brasil</h1>

<p class="floatstop">xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx,
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx...</p>
```

Para evitar que o texto se posicione no espaço livre deixado pela foto do Bill Gates basta adicionar a seguinte regra CSS:

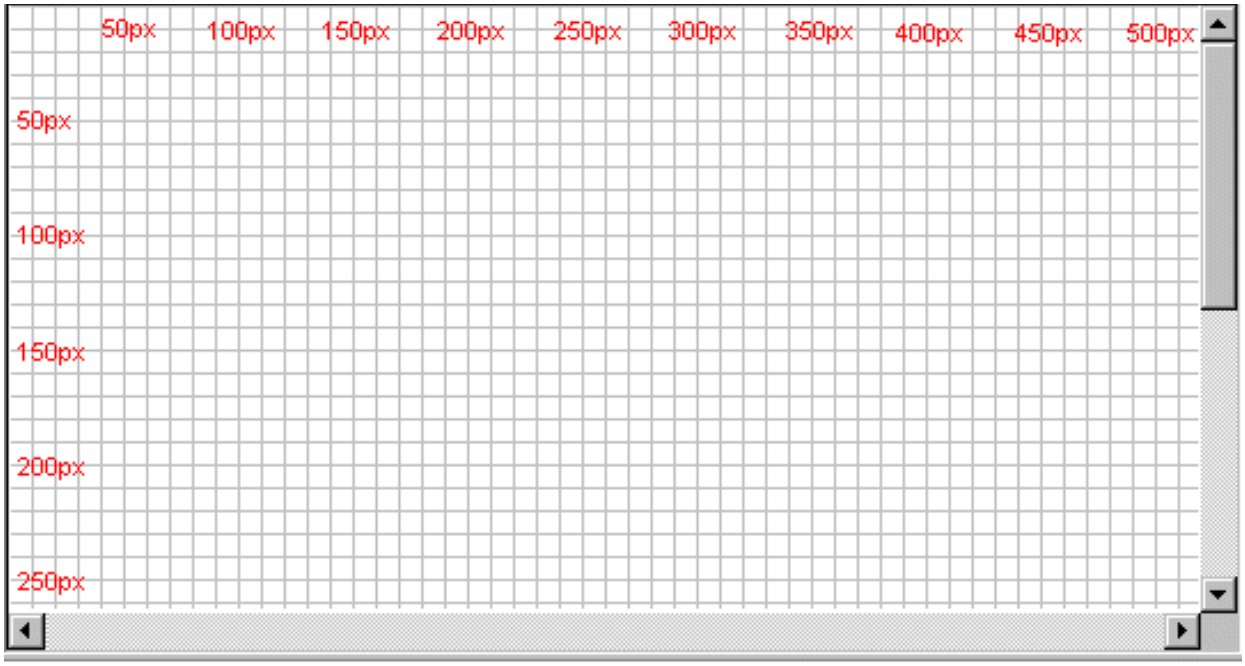
```
#picture {
      float:left;
      width: 100px;
}

.floatstop {
      clear:both;
}
```

14. Posicionando elementos

Com posicionamento CSS podemos colocar um elemento em uma posição exata na página. Combinado com floats (ver lição 13), o posicionamento abre muitas possibilidades para criação de layouts precisos e avançados.

14.1 - O princípio de posicionamento CSS

Considere a janela do navegador como um sistema de coordenadas:



O princípio de posicionamento CSS estabelece que você pode posicionar um elemento em qualquer lugar na tela usando um sistema de coordenadas.

Vamos supor que queremos posicionar um cabeçalho. Usando o box model, o cabeçalho pode ser estilizado para ser apresentado como mostrado abaixo:

Headline

```
h1 {
      position:absolute;
      top: 100px;
      left: 200px;
}
```

O resultado é mostrado a seguir:

Como você pode ver, posicionar com CSS é uma técnica precisa para colocar elementos. É muito mais fácil do que usar tabelas, imagens transparentes e tudo mais.

14.2 - Posicionamento absoluto

Um elemento posicionado absolutamente não cria nenhum espaço no documento. Isto significa que não deixa nenhum espaço vazio após ser posicionado.

Para posicionar um elemento de forma absoluta a propriedade position deve ser definida para **absolute**. Você pode então usar as propriedades **left**, **right**, **top**, e **bottom** para definir as coordenadas e posicionar o elemento.

Para exemplificar o posicionamento absoluto escolhemos colocar quatro boxes nos quatro cantos da página:

```
#box1 {
      position:absolute;
      top: 50px;
      left: 50px;
}

#box2 {
      position:absolute;
      top: 50px;
      right: 50px;
}

#box3 {
      position:absolute;
      bottom: 50px;
      right: 50px;
}

#box4 {
      position:absolute;
      bottom: 50px;
      left: 50px;
}
```

14.3 - Posicionamento relativo

Para posicionar um elemento de forma relativa a propriedade position deve ser definida para **relative**. A diferença entre os dois tipos de posicionamento é a maneira como o posicionamento é calculado.

O posicionamento para posição relativa é **calculado com base na posição original do elemento no documento**. Isto significa uma movimentação do elemento para a esquerda, para a direita, para cima ou para baixo. Assim fazendo o elemento ocupa um espaço após ser posicionado.

Como exemplo de posicionamento relativo vamos tentar posicionar três imagens relativamente as suas posições originais na página. Notar como as imagens deixam um espaço vazio nas suas posições originais no documento:

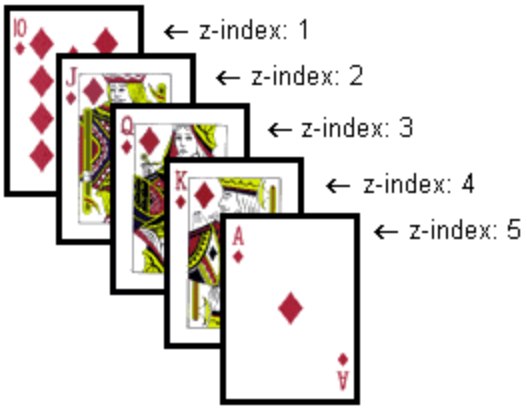
```
#bola1 {
     position:relative;
     left: 350px;
     bottom: 150px;
}

#bola2 {
     position:relative;
     left: 150px;
     bottom: 500px;
}

#bola3 {
     position:relative;
     left: 50px;
     bottom: 700px;
}
```

15. Usando o z-index (Layers)

CSS usa o espaço tri-dimensional - altura, largura e profundidade. Nas lições anteriores vimos as duas primeiras dimensões. Nesta lição aprenderemos como colocar elementos em layers (camadas). Resumindo, camadas significam como os elementos se sobrepõem uns aos outros.

Para fazer isto definimos para cada elemento um número índice (z-index). O comportamento é que elementos com número índice maior se sobrepõem àqueles com menor número.

Vamos supor um royal flush no jogo de poker. As cartas podem ser apresentadas como se cada uma delas tivesse um z-index:



No caso mostrado, os números índice estão em uma sequência direta (de 1-5), contudo o mesmo resultado poderia ser obtido com uso de 5 diferentes números, não em sequência. O que conta é a cronologia dos números (a ordem).

O código para a ilustração das cartas é mostrado a seguir:

```
#ten_of_diamonds {
     position: absolute;
     left: 100px;
     bottom: 100px;
     z-index: 1;
}

#jack_of_diamonds {
     position: absolute;
     left: 115px;
     bottom: 115px;
     z-index: 2;
}

#queen_of_diamonds {
     position: absolute;
     left: 130px;
     bottom: 130px;
     z-index: 3;
}

#king_of_diamonds {
     position: absolute;
     left: 145px;
     bottom: 145px;
     z-index: 4;
}

#ace_of_diamonds {
     position: absolute;
     left: 160px;
     bottom: 160px;
     z-index: 5;
}
```

O método é simples, mas as possibilidades são muitas. Você pode colocar imagens sobre textos, texto sobre texto, etc.