Introdução

Folhas de estilos em cascata — Cascading Style Sheets (CSS) — é uma ferramenta fantástica para construção do layout dos seus websites. Permite que você projete websites com uma técnica completamente diferente da convencional e possibilita uma considerável redução de tempo de trabalho. Conhecer CSS é uma necessidade para qualquer um envolvido com o projeto web.

Este tutorial iniciará você nas CSS em algumas poucas horas. É fácil de entender e ensinará a você todas as técnicas sofisticadas.

Aprender CSS é divertido. À medida em que você for progredindo no tututorial não esqueça de dedicar algum tempo para fazer seus experimentos com o que for aprendendo em cada lição.

Para usar CSS **é necessário um conhecimento básico de HTML**. Se você não está familiarizado com HTML, consulte nosso tutorial HTML antes de começar com CSS.

Quais softwares eu precisarei?

Por favor, evite usar softwares tais como FrontPage, DreamWeaver ou Word para acompanhar este tutorial. Softwares avançados não facilitarão seu aprendizado de CSS. Ao contrário, eles colocarão limites para você e atrasarão seu aprendizado.

Tudo o que você precisa é de um simples e gratuito editor de texto.

Por exemplo, Microsoft Windows vem com um programa chamado Notepad. Normalmente este programa está no diretório Accessories no menu Start => Programs. Alternativamente você pode usar um editor de texto similar, tal como, Pico para Linux ou Simple Text para Macintosh.

Um editor de texto simples é o ideal para o aprendizado de HTML e de CSS porque eles não afetam e nem modificam o código que você digita. Assim os acertos e erros de codificação devem-se exclusivamente a você — não ao software que você utiliza.

Você pode usar qualquer navegador parta acompanhar este tutorial. Nós aconselhamos que você mantenha seu navegador sempre atualizado e utilize a última versão disponível.

Um navegador e um editor de textos simples é tudo o que você precisa.

Vamos começar!

<< Tabela de conteúdos Lição 1: O que é CSS? >>

Lição 1: O que é CSS?

Talvez você já tenha ouvido falar em CSS, mas não saiba o que significa exatamente. Nesta lição você saberá o que é e para que serve CSS.

CSS é a abreviatura para **C**ascading **S**tyle **S**heets. — Folha de Estilos em Cascata

O que eu posso fazer com CSS?

CSS é uma linguagem para estilos que define o layout de documentos HTML. Por exemplo, CSS controla fontes, cores, margens, linhas, alturas, larguras, imagens de fundo, posicionamentos e muito mais. Aguarde e você verá!

HTML pode ser (in)devidamente usado para definir o layout de websites. Contudo CSS proporciona mais opções e é mais preciso e sofisticado. CSS é suportado por todos os navegadores atuais.

Depois de estudar algumas poucas lições deste tutorial, você estará em condições de projetar uma folha de estilos, usando CSS para dar um grande visual ao seu website.

Qual é a diferença entre CSS e HTML?

HTML é usado para estruturar conteúdos. CSS é usado para formatar conteúdos estruturados.

OK, isto soa um tanto técnico e confuso. Mas, por favor continue lendo. Tudo fará sentido em breve.

Em tempos passados quando a Madonna era virgem e um sujeito chamado Tim Berners Lee inventou a World Wide Web, a linguagem HTML era usada somente para estruturar textos. Um autor podia marcar seus textos definindo "isto é um cabeçalho " ou "isto é um parágrafo" usando tags HTML tais como `<h1>` e `<p>`.

À medida que a Web ganhava popularidade, os designers começavam a sentir a necessidade de encontrar meios de construir layout para os documentos online. Para suprir estas necessidades os fabricantes de navegadores (àquela época a Netscape e a Microsoft) inventaram novas tags HTML tais como, por exemplo a tag `<font>` que se diferenciava das tags originais do HTML pelo fato de destinar-se à layout — e não à estrutura.

Isto adicionalmente teve o efeito de disvirtuar o emprego de tags inicialmente projetadas para estrutura como por exemplo a tag <table> que passaram a ser empregadas para layout. Muitas destas novas tags para layout como a tag <blink> eram suportadas somente por um determinado tipo de navegador. A frase "Você precisa do navegador X para visualizar esta página" tornou-se comum nos websites.

CSS foi inventada para solucionar esta situação, colocando à disposição dos web designers meios sofisticados de projetar layouts suportados por todos os navegadores. E ao mesmo tempo a separação dos estilos de apresentação da marcação dos conteúdos torna a manutenção dos sites bem mais fácil.

Quais são os benefícios do uso de CSS?

CSS é uma revolução no mundo do web design. Os benefícios concretos do uso de CSS incluem:

- controle do layout de vários documentos a partir de uma simples folha de estilos;
- maior precisão no controle do layout;
- aplicação de diferentes layouts para servir diferentes mídias (tela, impressora, etc.);
- emprego de variadas, sofisticadas e avançadas técnicas de desenvolvimento.

Na próxima lição veremos com maiores detalhes como funciona CSS e como começar seu estudo.

<< Introdução

Lição 2: Como funciona CSS? >>

Lição 2: Como funciona CSS?

Nesta lição você aprenderá a desenvolver sua primeira folha de estilos. Você verá o básico sobre o modelo CSS e que código é necessário para usar CSS em um documento HTML.

Muitas das propriedades usadas em Cascading Style Sheets (CSS) são semelhantes àquelas do HTML. Assim, se você está acostumado a usar HTML para layout irá reconhecer muitos dos códigos que usaremos. Vamos dar uma olhada em um exemplo concreto.

A sintaxe básica das CSS

Suponha que desejamos uma cor de fundo vermelha para a página web:

Usando **HTML** podemos fazer assim:

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

Com **CSS** o mesmo resultado será obtido assim:

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

Como você pode notar os códigos HTML e CSS são mais ou menos parecidos. O exemplo acima serve também para demonstrar o fundamento do modelo CSS:

Mas, onde colocar o código CSS? É isto que veremos a seguir.

Aplicando CSS a um documento HTML

Você pode aplicar CSS a um documento de três maneiras distintas. Os três métodos de aplicação estão exemplificados a seguir. Recomendamos que você foque no terceiro método, ou seja o método externo.

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 aplicada conforme mostrado a seguir:

```
<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 a seguir:

```
<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. Usaremos este método nos exemplos deste tutorial.

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

Vamos supor, por exemplo, que sua folha de estilos tenha sido nomeada de **style.css** e está localizada no diretório **style**. Tal situação está mostrada a seguir:

O "truque" é criar um link no documento HTML (default.htm) para a folha de estilos (style.css). O link é criado em uma simples linha de código HTML como mostrado a seguir:

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

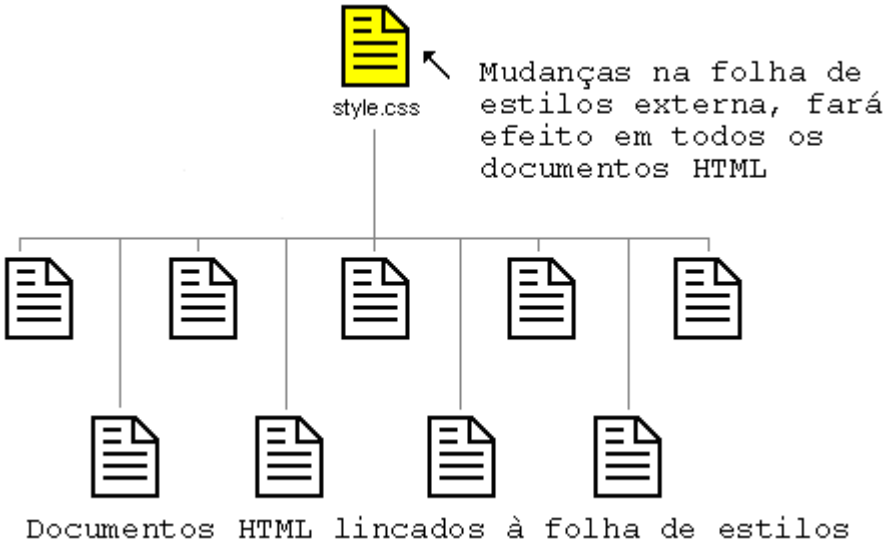
Notar que o caminho para a folha de estilos é indicado no atributo `href`.

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 lincar 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. Se por exemplo, você quiser trocar a cor do fundo de um site com 100 páginas, a folha de estilos evita que você edite 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.

Vamos praticar o que aprendemos.

Faça você mesmo

Abra o Notepad (ou qualquer outro editor de texto que queira usar) e crie dois arquivos — um arquivo HTML e um arquivo CSS — com os seguintes conteúdos:

default.htm

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

  </head>
  <body>
        <h1>Minha primeira folha de estilos</h1>
  </body>
</html>
```

style.css

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

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

Abra **default.htm** no seu navegador e veja uma página com o fundo vermelho. Parabéns! Você construiu sua primeira folha de estilos!

Prossiga para a próxima lição onde abordaremos algumas propriedades CSS.

<< Lição 1: O que é CSS? Lição 3: Cores e fundos >>

Lição 3: Cores e fundos

Nesta lição você aprenderá como aplicar cores de primeiro plano e cores de fundo no seu website. Abordaremos ainda os métodos avançados de controle e posicionamento de imagens de fundo. Serão explicadas as seguintes propriedades CSS:

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

Cor do primeiro plano: a propriedade 'color'

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

Considere, por exemplo, que desejamos 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;
}
```

Ver exemplo

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)).

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, devemos aplicar a propriedade `background-color` ao elemento `<body>`.

Você pode 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;

}
```

Ver exemplo

Notar que foram aplicadas duas propriedades ao elemento `<h1>` separadas por um ponto e vírgula.

Images de fundo [background-image]

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

Usaremos a imagem de uma borboleta para exemplificar a aplicação de imagens de fundo. Você pode fazer o download da imagem mostrada abaixo e usá-la nos seus experimentos (clique com o botão direito do mouse sobre a imagem e escolha "salvar imagem como") ou você poderá usar uma outra imagem qualquer ao seu gosto.

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("butterfly.gif");
}

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

Ver exemplo

NB: Notar como foi especificado o caminho para a imagem usando **url("butterfly.gif")**. 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/butterfly.gif")** ou até mesmo hospedá-la na Internet: **url("http://www.html.net/butterfly.gif")**.

Imagem de fundo repetida [background-repeat]

No exemplo anterior você observou que a imagem da borboleta 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`.

Value	Description	Example
`Background-repeat: repeat-x`	A imagem se repete na horizontal	Ver exemplo
`background-repeat: repeat-y`	A imagem se repete na vertical	Ver exemplo
`background-repeat: repeat`	A imagem se repete na tanto na horizontal como na vertical	Ver exemplo
`background-repeat: no-repeat`	A imagem não se repete	Ver exemplo

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

```
body {
        background-color: #FFCC66;
        background-image: url("butterfly.gif");
        background-repeat: no-repeat;
}

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

Ver exemplo

Image 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 quatro diferentes valores para `background-attachment`. Veja os exemplos para constatar a diferença entre imagem fixa e imegem que rola.

Value	Description	Example
`Background-attachment: scroll`	A imagem rola com a página	Ver exemplo
`Background-attachment: fixed`	A imagem é fixa	Ver exemplo

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

```
body {
        background-color: #FFCC66;
        background-image: url("butterfly.gif");
```

```
                background-repeat: no-repeat;
                background-attachment: fixed;
        }

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

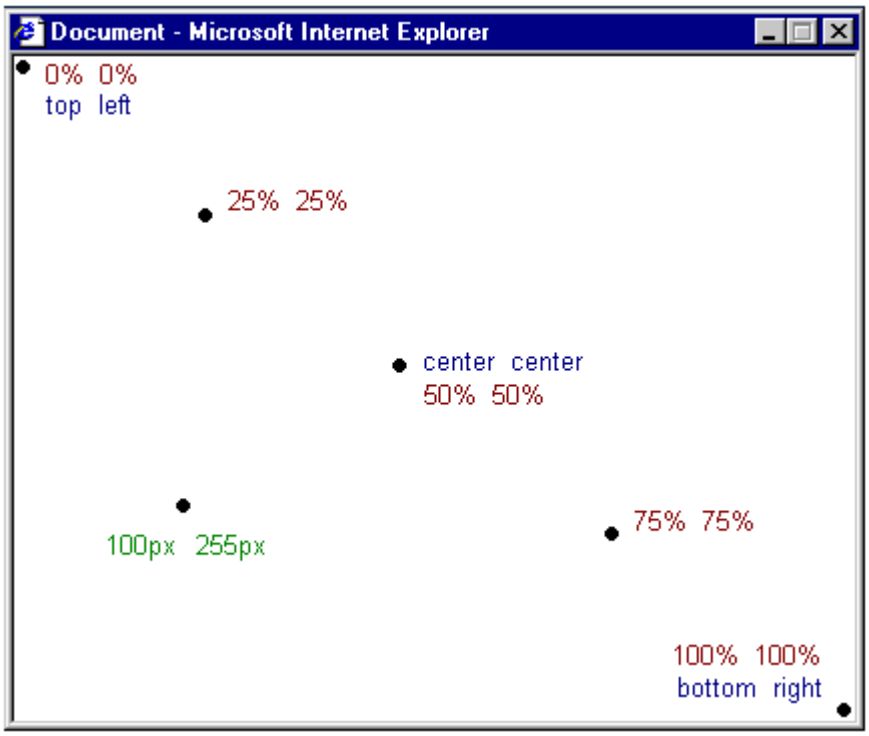
Ver exemplo

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 janeja, 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 .

Value	Description	Example
`background-position: 2cm 2cm`	A imagem é posicionada a 2 cm da esquerda e 2 cm para baixo na página	Ver exemplo
`background-position: 50% 25%`	A imagem é centrada na horizontal e a um quarto (25%) para baixo na página	Ver exemplo
`background-position: top right`	A imagem é posicionada no canto superior direito da página	Ver exemplo

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

```
body {
        background-color: #FFCC66;
        background-image: url("butterfly.gif");
        background-repeat: no-repeat;
        background-attachment: fixed;
        background-position: right bottom;
}

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

Ver exemplo

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("butterfly.gif");
background-repeat: no-repeat;
background-attachment: fixed;
background-position: right bottom;
```

Usando `background` você consegue o mesmo resultado, abreviando como mostrado abaixo:

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

As duas propriedades não declaradas assumirão o valor default que como você já sabe são: a imagem rola

na tela e será posicionada no canto superior esquerdo (que são os valores default para as propriedades não declaradas).

Sumário

Nesta lição você aprendeu técnicas que não são possíveis com uso de HTML. A brincadeira continua na próxima lição onde examinaremos as possibilidades de estilização das fontes.

<< Lição 2: Como funciona CSS? Lição 4: Fontes >>

Lição 4: Fontes

Nesta lição estudaremos as fontes e como aplicá-las usando CSS. Veremos como criar situações para que determinada fonte seja visualizada pelo usuário mesmo não estando instalada em seu sistema operacional. Descreveremos as seguintes propriedades CSS:

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

Família de fontes [font-family]

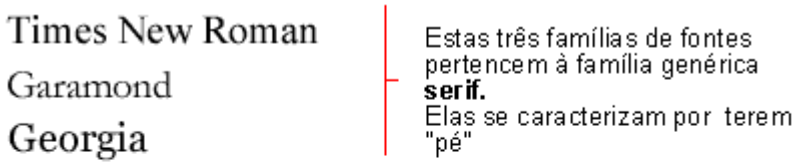
A propriedade `font-family` é usada para definir uma lista de fontes e sua prioridade para apresentação de um elemento em uma página. Se a primeira fonte da lista não estiver instalada na máquina do usuário, deverá ser usada a segunda e assim por diante até ser encontrada uma fonte instalada.

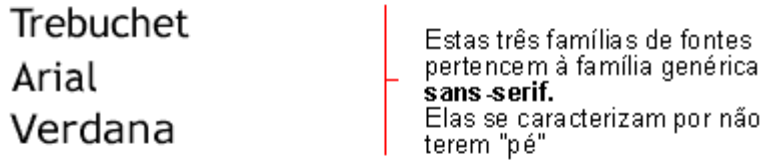
Existem dois tipos de nomes para definir fontes: nomes para famílias de fontes e nomes para famílias genéricas. Os dois são explicados a seguir:

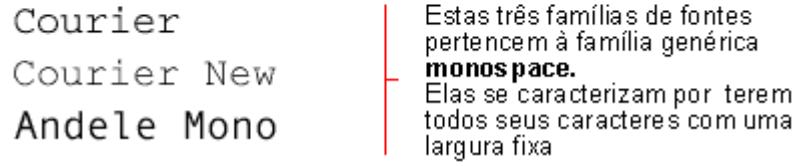
nome para famílias de fontes
: Exemplos para este tipo (normalmente conhecidas como "font") são "Arial", "Times New Roman" ou "Tahoma".

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".

A diferença está mostrada na figura a seguir:







Ao listar fontes para seu 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 mostramos um exemplo de listagem de fontes:

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

Ver exemplo

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**.

Notar 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.

Estilo da fonte [font-style]

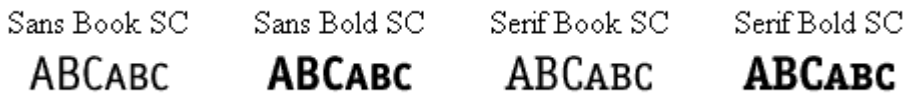
A propriedade `font-style` define a escolha da fonte em **normal**, **italic** ou **oblique**. No exemplo a seguir todos as cabeçalhos `<h2>` serão em itálico.

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

Ver exemplo

Fonte variante [font-variant]

A propriedade `font-variant` é usada para escolher as variantes **normal** ou **small-caps**. Uma fonte **small-caps** é aquela que usa letras maiúsculas de tamanhos reduzidos. Confundiu? Dê uma olhada nos exemplos a seguir:



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;}
```

Ver exemplo

Peso da fonte [font-weight]

A propriedade `font-weight` define quão negrito ou "pesada" deve ser renderizada a fonte. 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;}
```

Ver exemplo

Tamanho da fonte [font-size]

On tamanho da fonte é definido pela propriedade `font-size`.

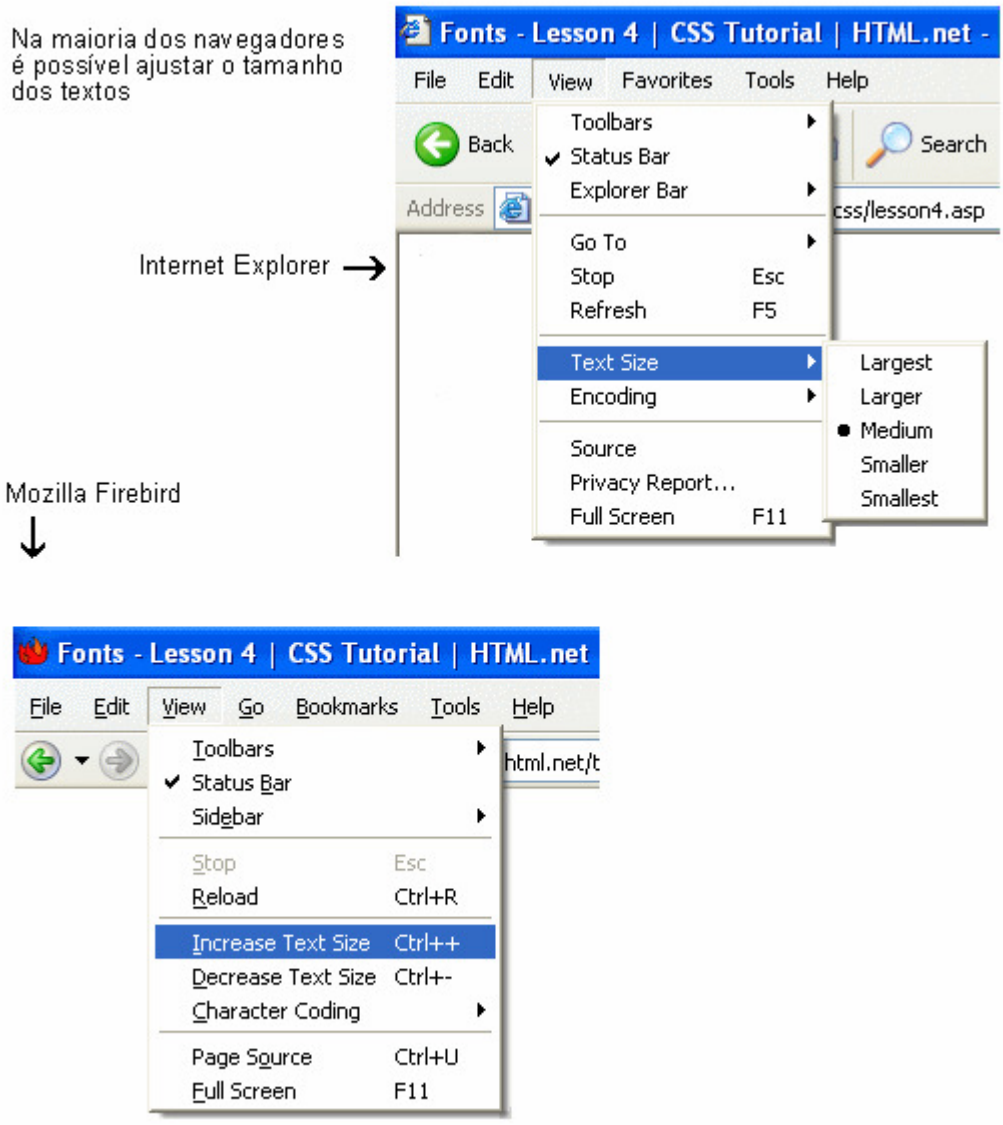
Existem muitas unidades (p. ex.:. pixels e percentagens) que podem ser usadas para definir o tamanho da fonte. Neste tutorial nós usaremos as unidades mais comuns e apropriadas. Ver exemplos a seguir:

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

Ver exemplo

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. Muitos usuários têm restrições, como por exemplo, pessoas idosas, pessoas com visão limitada ou as que usam um monitor de baixa qualidade. **Para fazer seu site acessível** a todos, você deverá usar unidades como '**%**' ou '**em**'.

Abaixo uma figura mostrando como ajustar o tamanho das fontes nos navegadores Mozilla Firefox e Internet Explorer. Tente você mesmo este ajuste — uma excelente funcionalidade do navegador, não é mesmo?



Compilando [font]

Usar `font` é uma abreviação que permite definir várias propriedades em uma só.

Veja 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`

Sumário

Você acaba de aprender algumas possibilidades relativas a fontes. Lembre-se que a grande vantagem de especificar fontes usando CSS é que você poderá alterar em alguns minutos, as fontes de um website inteiro. CSS economiza tempo e facilita sua vida. Na próxima lição veremos as propriedades para textos.

<< Lição 3: Cores e fundos Lição 5: Textos >>

Lição 5: Textos

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

- text-indent
- text-align
- text-decoration
- letter-spacing
- text-transform

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 marcados com `<p>`:

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

Ver exemplo

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 (**centred**). 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;
}
```

Ver exemplo

Decoração de textos [text-decoration]

A propriedade `text-decoration` possibilita adicionar "efeitos" ou "decoração" em textos. Você pode por rexemplo, 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;
}
```

Ver exemplo

Espaço entre letras [letter-spacing]

O espaçamento entre os caracteres de um texto é controlado pela propriedade `letter-spacing`. 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;
}
```

Ver exemplo

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 `text-transform`:

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 trasformações - 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;
}
```

Ver exemplo

Sumário

Nas três últimas lições aprendemos várias propriedades CSS, mas há muito mais ainda a aprender com CSS. Na próxima lição veremos os links.

<< Lição 4: Fontes Lição 6: Links >>

Lição 6: Links

Você pode aplicar aos links tudo que aprendeu nas lições anteriores (i.e. 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 pseudo-classes.

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 explicaremos com mais detalhes e exemplificação, as quatro pseudo-classes.

Pseudo-classe: link

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

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

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

Ver exemplo

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;
}
```

Ver exemplo

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;
}
```

Ver exemplo

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;
}
```

Ver exemplo

Exemplo 1: Efeito quando o ponteiro está sobre o link

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

Exemplo 1a: Espaçamento entre as letras

Como você deve estar lembrado da lição 5, 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;
}
```

Ver exemplo

Exemplo 1b: UPPERCASE e lowercase

Na lição 5 vimos 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;
}
```

Ver exemplo

Os exemplos mostrados dão uma idéia das inúmeras possibilidades de combinação de diferentes propriedades. Você pode criar seus próprios efeitos — faça uma tentativa!

Exemplo 2: Removendo sublinhado dos links

Uma pergunta comum: Como remover o sublinhado dos links?

Você deve estudar com muito cuidado a necessidade de retirar o sublinhado dos links, pois isto poderá reduzir signifcativamente 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. Até minha mãe sabe disto! Se você muda a cor e retira o sublinhado dos links, poderá confundir seus visitantes e em conseqüência não retirar o máximo dos conteúdos do seu website.

Feita esta ressalva, é muito fácil retirar o sublinhado dos links. Conforme explicado na lição 5, 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, você pode 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;
}
```

Ver exemplo

Sumário

Nesta lição você aprendeu pseudo-classes, usando algumas propriedades já estudadas nas lições anteriores. Isto deve ter proporcionado uma idéia das possibilidades das CSS.

Na próxima lição ensinaremos como definir propriedades para elementos específicos e grupamentos de elementos.

<< Lição 5: Textos Lição 7: Identificando e agrupando elementos (class and id) >>

Lição 7: Identificando e agrupando elementos (classes e id)

Em alguns casos você deseja aplicar estilos a um elemento ou grupo de elementos em particular. Nesta lição 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 iremos responder nesta lição.

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>
```

Ver exemplo

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;
}
```

Ver exemplo

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.

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;
}
```

Ver exemplo

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

Sumário

Nesta lição você aprendeu que com o uso dos seletores `class` e `id`, é possível definir propriedades CSS para elementos específicos.

Na próxima lição trataremos de dois elementos do HTML largamente usados com as CSS: `<span>` e `<div>`.

<< Lição 6: Links Lição 8: Agrupando elementos (span and div) >>

Lição 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 `<span>`
- Agrupando com`<div>`

Agrupando com

O elemento `<span>` é um elemento neutro e que não adiciona qualquer tipo de semântica ao documento. Contudo, `<span>` 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 de autoria de Benjamin Franklin:

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

Vamos supor que queremos enfatizar na cor vermelha os benefícios apontados por Mr. Franklin pelo fato de não se passar o dia dormindo. Para isto marcamos os benefícios com `<span>`. A cada `span` atribuímos uma `class`, e estilizamos na folha de estilos:

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

A folha de estilos:

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

Ver exemplo

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

Agrupando com

Enquanto `<span>` é usado dentro de um elemento nível de bloco como vimos no exemplo anterior, `<div>` é usado para agrupar um ou mais elementos nível de bloco.

Diferenças à parte, o agrupamento com `<div>` funciona mais ou menos da mesma maneira. Vamos ver 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;
}
```

Ver exemplo

Nos exemplos mostrados acima usamos somente `<div>` e `<span>` 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.

Sumário

Na lição7 e 8 você aprendeu seletores `id` e `class` e elementos `<span>` e `<div>`.

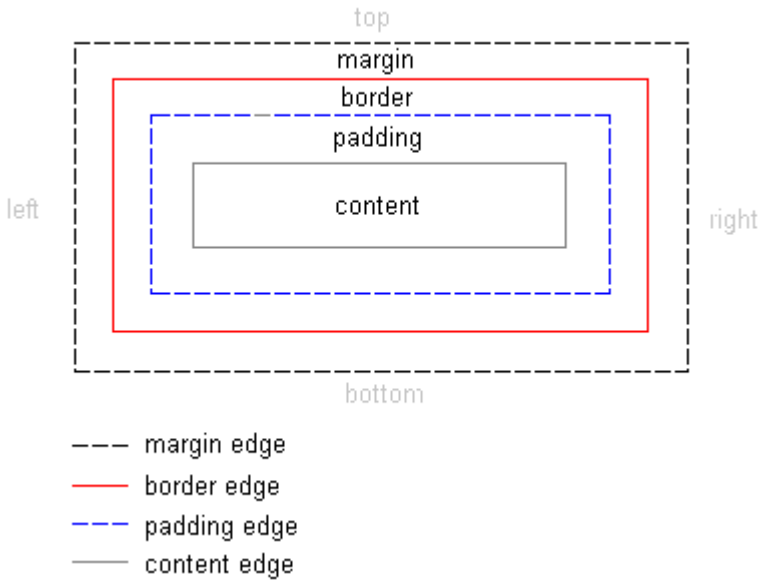
Você agora já deve estar apto a agrupar e identificar razoalvelmente todas as partes de um documento, o que é um grande passo na direção de tornar-se um mestre nas CSS. Na lição 9 veremos o box model.

<< Lição 7: Identificando e agrupando elementos (class and id) Lição 9: O box model >>

Lição 9: O 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:

O box model em CSS

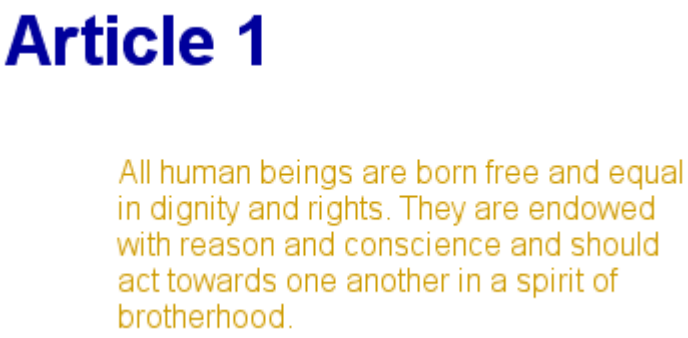


A ilustração acima é teórica. Vamos explicá-la na prática tomando como base um cabeçalho e um texto. O HTML para nosso exemplo (o texto foi retirado da Declaração Universal dos Direitos Humanos e está no original em inglês) é o mostrado abaixo:

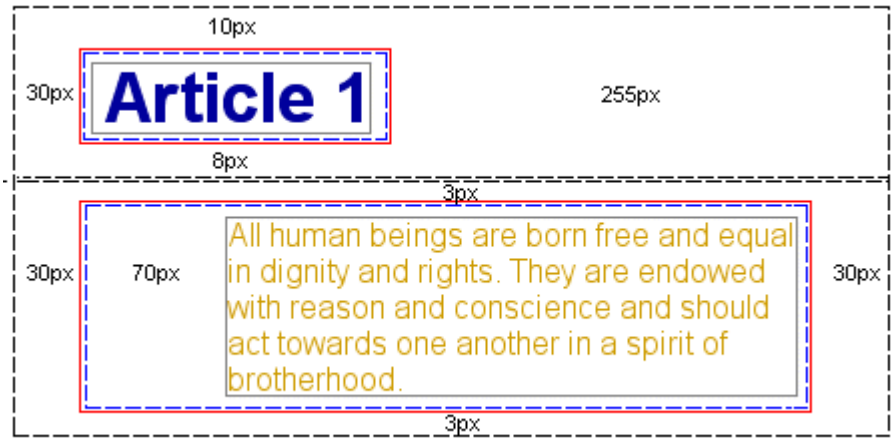
```
<h1>Article 1:</h1>

<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.

Sumário

Nesta lição você foi apresentado ao box model. Nas três lições seguintes iremos detalhar como criar e controlar elementos no box model.

<< Lição 8: Agrupando elementos (span and div) Lição 10: Margin e padding >>

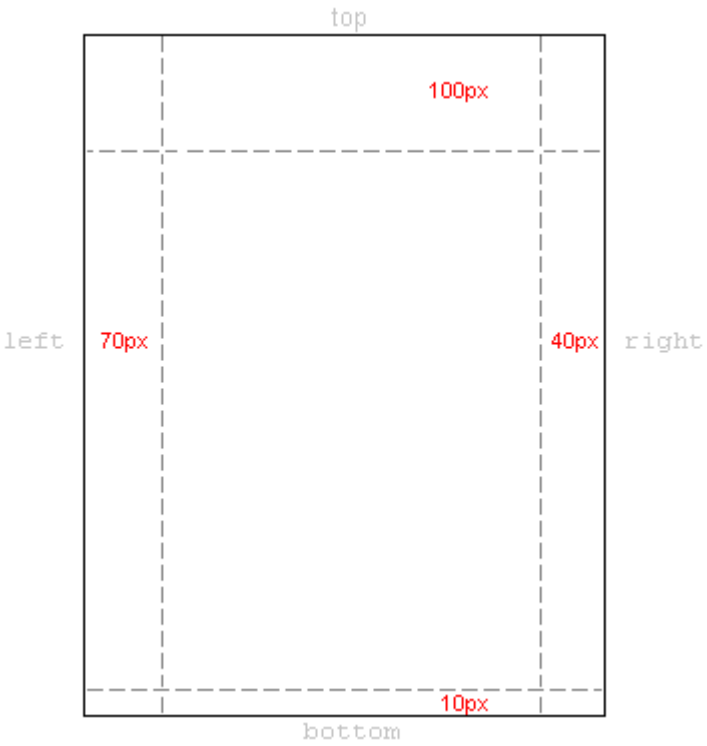
Lição 10: Margin e padding

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

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). Ver o diagrama mostrado na lição 9.

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;
```

```
}
```

Ver exemplo

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;
}
```

Ver exemplo

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;
}
```

Ver exemplo

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;
}
```

Ver exemplo

Sumário

Você está no caminho para se tornar um mestre em CSS box model. Na próxima lição veremos com detalhes como atribuir bordas coloridas e formas aos elementos.

<< Lição 9: O box model

Lição 11: Bordas >>

Lição 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.

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:

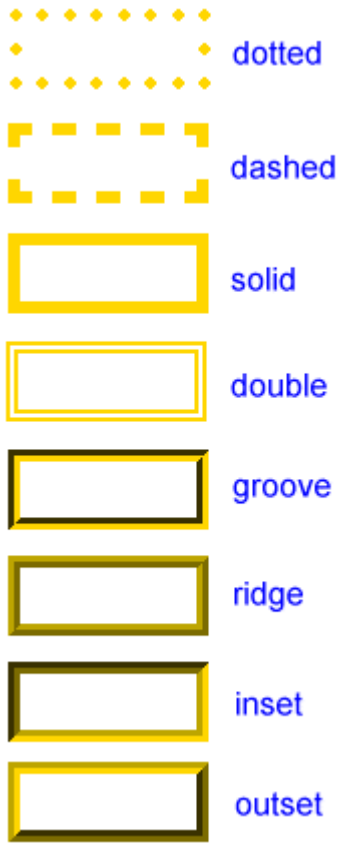
As cores das bordas [border-color]

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" .

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.



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;
}
```

Ver exemplo

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;
}
```

Ver exemplo

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;
}
```

Sumário

Nesta lição aprendemos que são infinitas as possibilidades de estilização de bordas com CSS.

Na próxima lição veremos como definir as dimensões do box model - height e width.

<< Lição 10: Margin e padding Lição 12: Altura e largura >>

Lição 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.

- width
- height

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;
}
```

Ver exemplo

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;
}
```

Ver exemplo

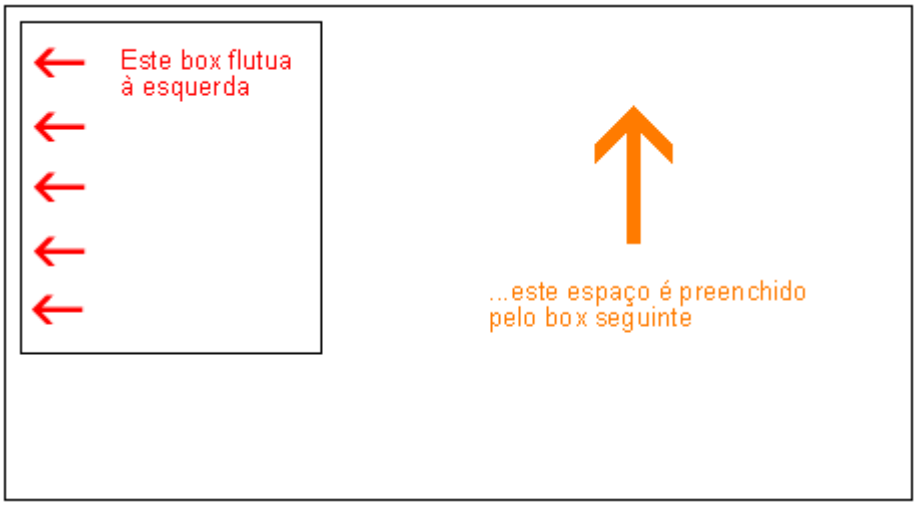
Sumário

Nas Lições9, 10, 11 e 12 você aprendeu os fundamentos do box model em CSS. Como podemos notar o box model fornece uma série de novas opções de estilização. Até agora, você deve ter usado tabelas para criar seus layouts, porém com CSS e o box model você, a partir de agora, estará capacitado a criar layouts mais elegantes, precisos e em conformidade com as recomendações do W3C.

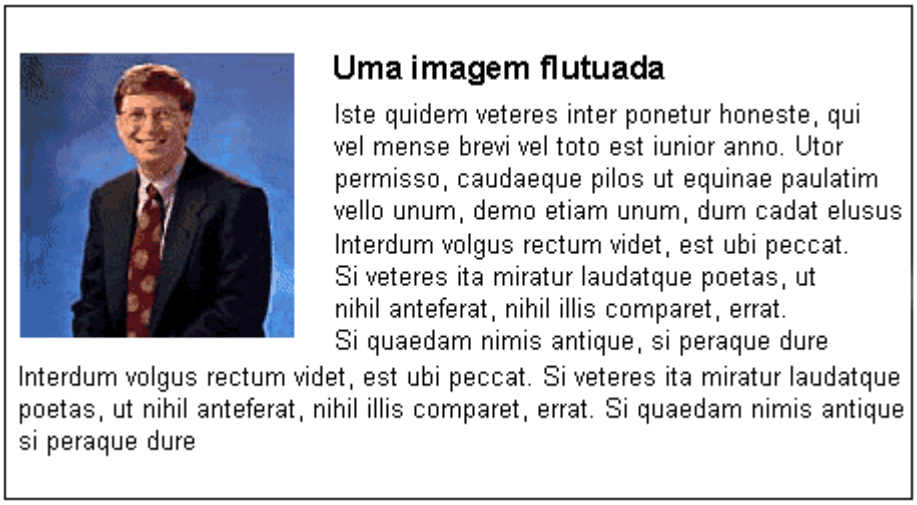
<< Lição 11: Bordas Lição 13: Flutuando elementos (floats) >>

Lição 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) (ver lição 9 para descrição do Box model). 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

Iste quidem veteres inter ponetur honeste, qui vel mense brevi vel toto est iunior anno. Utor permisso, caudaeque pilos ut equinae paulatim vello unum, demo etiam unum, dum cadat elusus Interdum volgus rectum videt, est ubi peccat. Si veteres ita miratur laudatque poetas, ut nihil anteferat, nihil illis comparet, errat. Si quaedam nimis antique, si peraque dure Interdum volgus rectum videt, est ubi peccat. Si veteres ita miratur laudatque poetas, ut nihil anteferat, nihil illis comparet, errat. Si quaedam nimis antique si peraque dure

Como isto é feito ?

O HTML para o exemplo acima é mostrado a seguir:

```
<div id="picture">
        <img src="bill.jpg" alt="Bill Gates">
</div>

<p>causas naturales et antecedentes,
idciro etiam nostrarum voluntatum...</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;
}
```

Ver exemplo

Outro exemplo : colunas

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

```
<div id="column1">
        <p>Haec disserens qua de re agatur
        et in quo causa consistat non videt...</p>
</div>

<div id="column2">
        <p>causas naturales et antecedentes,
        idciro etiam nostrarum voluntatum...</p>
</div>

<div id="column3">
        <p>nam nihil esset in nostra
        potestate si res ita se haberet...</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%;
}
```

Ver exemplo

`float` pode ser declarado **left**, **right** ou **none**.

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="bill.jpg" alt="Bill Gates">
</div>

<h1>Bill Gates</h1>

<p class="floatstop">causas naturales et antecedentes,
idciro etiam nostrarum voluntatum...</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;
}
```

Ver exemplo

Sumário

Floats são muito úteis em várias situações e frequentemente usados em conjunto com posicionamento. Na próxima lição veremos como posicionar um box tanto de maneira relativa como absoluta.

<< Lição 12: Altura e largura

Lição 14: Posicionando elementos >>

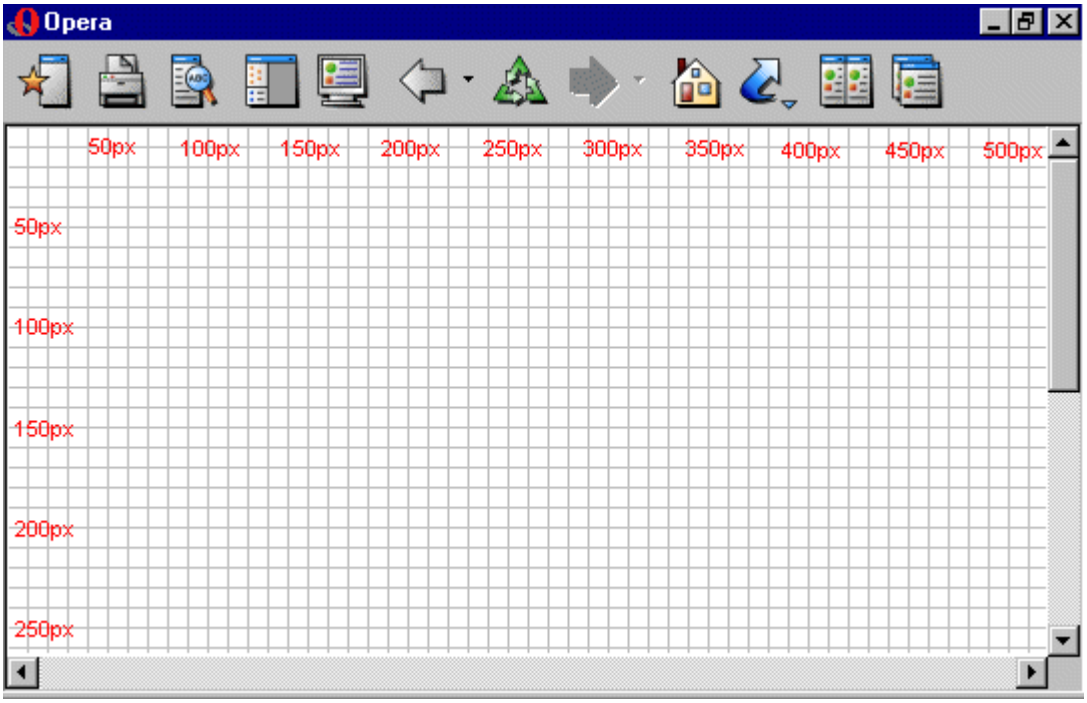
Lição 14: Posicionando elementos

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.

Nesta lição veremos os seguintes itens:

O princípio de posicionamento CSS

Considere a janela do navegador como um sistema de coordenadas:



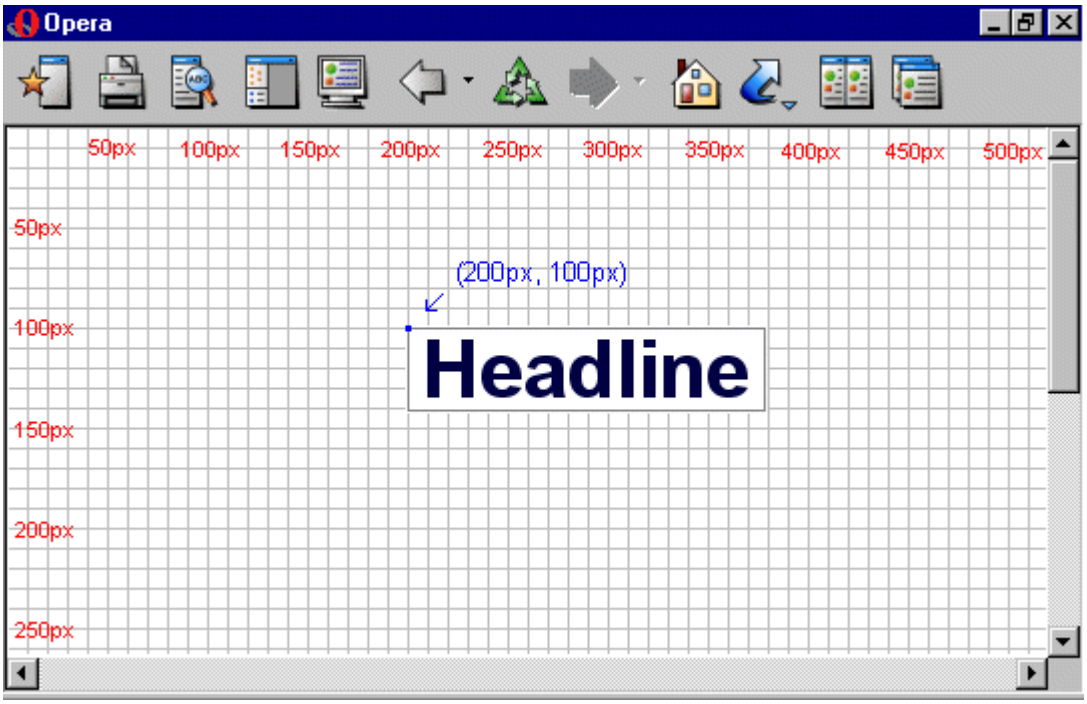
O princípio de posicionamento CSS estabelece que você pode posicionar um elemento em qualquer lugar na tela usando um sistema de coordenadas.

Vamos supor que queremos posicionar um cabeçalho. Usando o box model (ver lição 9) o cabeçalho pode ser estilizado para ser apresentado como mostrado abaixo:

Se quisermos o cabeçalho posicionado a 100px do topo do documento e a 200px à esquerda, podemos usar o seguinte CSS:

```
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.

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;
}
```

Ver exemplo

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:

```
#dog1 {
        position:relative;
        left: 350px;
        bottom: 150px;
}
#dog2 {
        position:relative;
        left: 150px;
        bottom: 500px;
}

#dog3 {
        position:relative;
        left: 50px;
        bottom: 700px;
}
```

Ver exemplo

Sumário

Nas duas lições anteriores você aprendeu como flutuar e posicionar elementos. Estes dois métodos posibilitam a você construir o layouy sem uso das ultrapassadas tabelas e imagens transparentes no HTML. Use CSS. É mais preciso, mais vantajoso e muito mais fácil de manutenir.

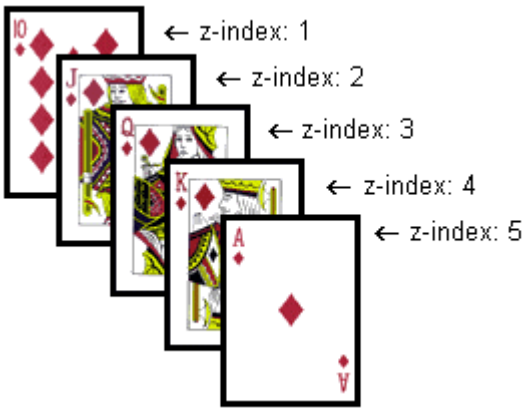
<< Lição 13: Flutuando elementos (floats) Lição 15: Usando z-index (Layers) >>

Lição 15: Usando 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 seqüência direta (de 1-5), contudo o mesmo resultado poderia ser obtido com uso de 5 diferentes números, não em seqüência. 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;
        top: 100px;
        z-index: 1;
}

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

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

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

#ace_of_diamonds {
        position: absolute;
        left: 160px;
```

```
            top: 160px;
            z-index: 5;
        }
```

Ver exemplo

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

Sumário

Layers podem ser usados em muitas situações. Tente, por exemplo, usar `z-index` para criar efeitos em cabeçalhos no lugar de usar imagens. Por um lado é mais rápido carregar texto na página e por outro, texto é mais amigável aos dispositivos de indexação.

<< Lição 14: Posicionando elementos Lição 16: Web-standards e validação >>

Lição 16: Web-standards e validação

W3C é a sigla para World Wide Web Consortium, uma organização independente que gerencia as normas para codificação na Internet (isto é, HTML, CSS, XML e outros). Microsoft, Fundação Mozilla e muitas outras organizações são membros do W3C e formam um consenso sobre o futuro desenvolvimento de normas.

Se você tem alguma experiência com web design, provavelmente sabe que há uma grande diferença na maneira como diferentes navegadores renderizam uma página. É frustante e requer um consumo de muito tempo, criar uma página que possa ser visualizada consistentemente no Mozilla, Internet Explorer, Opera e no restante dos navegadores existentes.

A idéia da normatização é criar um consenso e encontrar um denominador comum para uso de tecnologias para a Web. Isto significa que seguindo as normas, um desenvolvedor terá a certeza de que sua criação será tratada de maneira apropriada em diferentes plataformas. **Assim, nós recomendamos que você se beneficie do trabalho desenvolvido pelo W3C e valide sua CSS para estar em conformidade com as normas.**

Validador CSS

Para facilitar a verificação aos preceitos das normas CSS, o W3C desenvolveu um validador que faz uma verificação da folha de estilos e retorna um relatório com os eventuais erros e avisos caso sua CSS não valide.

Para facilitar a validação da sua folha de estilos você poderá submetê-la ao validador aqui mesmo nesta página. Na caixa de texto abaixo, substitua a URL existente pela URL da sua folha de estilos e clique no botão para validar. Você será informado pelo site do W3C se há erros na sua folha de estilos.

http://www.html.net/site/style/default.screen.css

Validar folha de estilos

Se o validador não encontrar erros; será mostrada uma imagem como a abaixo, que você poderá usar na sua página para anunciar que está usando um código válido:

O validador pode também ser encontrado neste link: http://jigsaw.w3.org/css-validator/

<< Lição 15: Usando z-index (Layers)