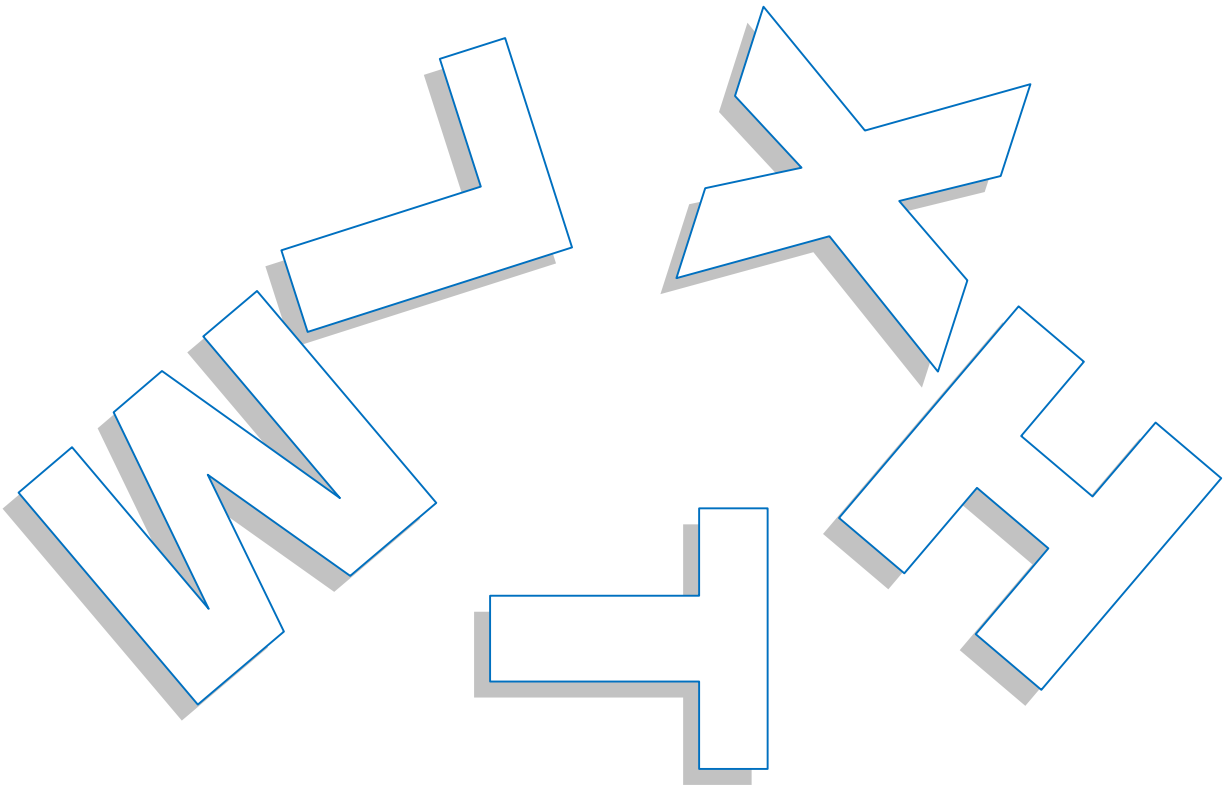
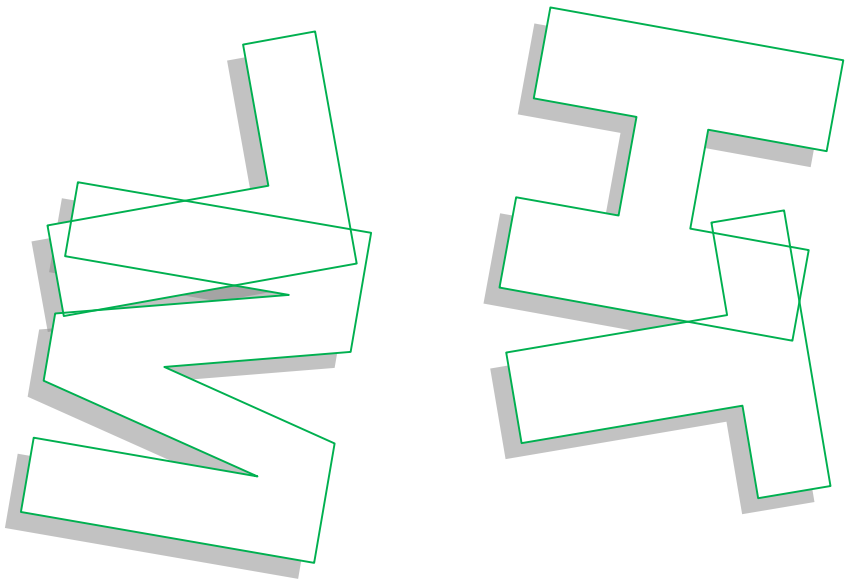
aPOstiLA DE



html, xhtml e cSs



Índice

HTML

CSS

XHTML

INTRODUÇÃO

Nesta primeira parte do tutorial vamos falar sobre a linguagem HTML. Podemos dizer que o HTML é a linguagem mãe. É importante aprendermos o HTML, para poder trabalhar com outras linguagens. O HTML, é responsável pela parte 'pesada' do site. Hoje, existe o XHTML, que veio para substituir o HTML.Você deve se perguntar, por que então eu vou aprender HTML em vez de aprender logo XHTML? É importante que você aprenda o básico primeiro, a linguagem XHTML não é tão diferente do HTML, mas é melhor que você conheça o HTML primeiro que tudo vai ficar mais fácil depois para aprender o XHTML. E é difícil achar algum tutorial ou apostila ensinando o XHTML, já que é praticamente o HTML modificado, com maneiras de fechar as tags diferente etc. Então, aprenda primeiro pelo HTML e depois faça a comparação e veja as diferenças. Será muito mais fácil entender o XHTML depois de ter visto o HTML.

Introdução ao HTML

O HTML, é a linguagem mãe usada para construir os sites. Apesar de ser coisa do passado, um site ser feito apenas com HTML, sem o HTML não é possível montar uma página de internet. O HTML foi ficando obsoleto, por ser uma linguagem limitada não podendo trabalhar com gráficos e animações por exemplo, apenas texto e imagens estáticas ou GIFs animados sem muita complexidade. Hoje é usado o HTML, junto com alguma outra linguagem, ou com outras linguagens. Sim, porque vamos supor que você queira colocar animações em seu site ou até mesmo fazer pequenas alterações no ponteiro do mouse, mensagem na barra de título, mensagens de aviso. Isso não seria possível somente com o HTML. São utilizadas linguagens como o Java Script ou DHTML(Dynamic HTML) juntamente com o HTML. Hoje são desenvolvidas páginas com banners em Flash, Fireworks, Photoshop, páginas feita toda em Flash(o que não é recomendável para que não tem banda larga), páginas feitas no Fireworks, mas introduzido e alinhado em uma site usando HTML e CSS.

1. O que são tags HTML?

AS TAGS HTML É A ESTRUTURA ONDE SERÁ MONTADA NOSSAS PÁGINAS HTML. AS TAGS SÃO RESPONSÁVEIS PELO NAVEGADOR INTERPRETAR O QUE ESTÃO ENTRE ELAS. SEM ELAS NÃO SERIA POSSÍVEL INTERPRETAR O SEU SITE.

2. VAMOS VER AS TAGS PRINCIPAIS PARA A CONSTRUÇÃO DE UMA PÁGINA HTML.

1. ESTRUTURA BÁSICA.

<html> é a tag de abertura do html.

<head> cabeçalho da página. onde ficam informações adicionais.

</head> tag de fechamento do cabeçalho.

<title> é a tag de abertura do título da janela.

</title> tag de fechamento do título da janela.

<body> corpo da página. aqui ficará a parte principal do seu site.

</body> tag de fechamento do corpo da página.

</html> tag de fechamento do html.

3. INICIANDO COM O HTML

.. TÍTULO NA JANELA DO SITE.

A tag utilizada para o título da janela é o <title>.

```
<html>
<head>
<title>Meu website</title>
</head>
<body>
</body>
</html>
```

```
<title> Aqui ficará o título da janela </title>
```

2. TÍTULO DO SITE

A tag utilizada para o título é a <h1> ao <h6>.

```
<html>
<head>
<title>Meu website</title>
</head>
<body>
 <h1>Um cabeçalho</h1>
</body>
</html>
```

```
<h1>Aqui ficará o título </h1>

<h2>Aqui ficará o título </h2>

<h3>Aqui ficará o título </h3>

<h4>Aqui ficará o título </h4>

<h5>Aqui ficará o título </h5>

<h6>Aqui ficará o título </h6>

*O tamanho do título pode ser definido por suas tags de h1 a h6, sendo que quanto maior o valor da tag menor será o tamanho do texto.
```

3.SUBTÍTULO DO SITE

A tag utilizada para o subtítulo é a <h1> ao <h6>.

```
<html>
<head>
<title>Meu website</title>
</head>
<body>
<h1>Um cabeçalho</h1>
<h2>Subtítulo</h2>
</body>
</html>
```

```
<h1>Aqui ficará o subtítulo </h1>

<h2>Aqui ficará o subtítulo </h2>

<h3>Aqui ficará o subtítulo </h3>

<h4>Aqui ficará o subtítulo </h4>

<h5>Aqui ficará o subtítulo </h5>

<h6>Aqui ficará o subtítulo </h6>
```

*O tamanho do subtítulo, pode ser definido assim como no título.

4.TEXTO PARÁGRAFO NO SITE

A tag para o texto parágrafo é o <p>.

```
<html>
<head>
<title>Meu website</title>
</head>
<body>
<h1>Um cabeçalho</h1>
<p>texto, texto texto, texto</p>
<h2>Subtítulo</h2>
 <p>texto, texto texto, texto</p>
</body>
</html>
```

```
<p>Aqui ficará o texto parágrafo</p>
```

No exemplo foi colocado um texto embaixo do título e em baixo do subtítulo.

5.Texto em negrito

Para colocarmos um texto em negrito devemos adicionar entre a tags e .

```
<b> Este texto deve ser negrito.</b>
```

Será renderizado no navegador assim:

Este texto deve ser negrito.

6.Texto em Itálico

Para colocarmos um texto em negrito devemos adicionar entre a tags e .

```
<i> Este texto deve ser itálico.</i>
```

Será renderizado no navegador assim:

Este texto deve ser itálico.

7.Texto com letras menores

Para fazermos textos usando letras menores podemos usar a tag <small></small>.

```
<small> Este texto deve ser com letras em tamanho small</small>
```

Será renderizado no navegador assim:

Este texto deve ser com letras em tamanho small.

8. Posso usar várias tags simultaneamente?

Sim você pode usar quantas tags queira, desde de que as aninhe convenientemente. Veja como fazer isto no exemplo abaixo:

Para escrever um texto em negrito e itálico faça como mostrado a seguir:

```
<b><i>Texto em negrito e itálico.</i></b>
```

E não assim:

```
<b><i>Texto em negrito e itálico.</b></i>
```

Observe que no primeiro exemplo a primeira tag de abertura corresponde a última tag de fechamento , e o aninhamento está certo. Isto evita confusão para quem escreve o código e para o navegador que lê o código. As últimas tags a serem abertas tem que ser as primeiras a serem fechadas, e as primeiras a serem abertas terão de ser as últimas a serem fechadas.

4. MAIS TAGS!

9. Existem tags que são abertas e fechadas em única tag. Estas tags são comandos isolados, ou seja, não contém nenhum texto dentro delas para poder funcionar. Um exemplo é a tag <br /> que serve para criar uma quebra de linha:

```
Um texto<br /> e mais texto em nova linha
```

Será renderizado no navegador assim:

Um texto
e mais texto em nova linha

Notar que a tag é escrita como se fosse uma mistura de tag de abertura e de fechamento com uma barra "/" no final: <br />. A princípio podemos escrever também
</br> (sem conteúdo), mas para que complicar?

Outra tag de comando é <hr /> que serve para definir uma linha horizontal ("hr" vem "horizontal rule" - régua horizontal):

```
<hr />
```

Será renderizado no navegador assim:

---

10. Lista

```
<ul>
  <li>Um item de lista</li>
  <li>Outro item de lista</li>
</ul>
```

Será renderizado no navegador assim:

- Um item de lista
- Outro item de lista

11. Lista ordenada:

```
<ol>
  <li>Primeiro item da lista</li>
  <li>Segundo item da lista</li>
</ol>
```

Será renderizado no navegador assim:

1. Primeiro item da lista
2. Segundo item da lista

Tente você mesmo!

```
<i>Itálico</i>
<small>Texto tamanho small</small>
<br /> Pula linha
<hr /> Linha Horizontal
<blockquote>Indentação</blockquote>
<ul>Lista<ul>
<ol>Lista ordenada<ol>
<li>Item de lista</li>
```

5. ATRIBUTOS

O que é atributo?

Como você deve estar lembrado, uma tag é um comando para o navegador (por exemplo,
 é um comando para mudar de linha). Em algumas tags você pode ser mais específico, acrescentando na tag, informações adicionais de comando. Isto é feito através dos atributos.

```
<h3 style="background-color:#ff0000;"> HTML (Hyper Text Markup Language)</h2>
```

Atributos são escritos dentro da tag, seguidos por um sinal de igual e depois entre aspas são declaradas as informações do atributo. As informações quando mais de uma, devem ser separadas por ponto e vírgula, tudo conforme mostrado no exemplo acima. Adiante voltaremos a este assunto.

Como é isto?

Existem vários atributos. O primeiro que você aprenderá é o atributo style. Com o atributo style você pode adicionar estilização e layout ao seu website. Por exemplo, uma cor de fundo:

```
<html>
<head>
</head>
<body style="background-color:#ff0000;">
</body>
</html>
```

O código acima renderiza uma página com cor de fundo vermelha. Experimente você mesmo, construa uma página vermelha. A seguir explicaremos como funcionam as cores.

Notar que algumas tags e atributos usam nomes do idioma inglês dos E.U.A. É muito importante que você use os nomes exatamente como mostrados neste tutorial - se você mudar uma letra que seja, o navegador não irá entender seu código. É importante também que você não se esqueça de fechar as aspas nas informações do atributo.

Como a página ficou vermelha?

No exemplo acima nós usamos o código "#ff0000" para fazer a página na cor vermelha. Eate é o código para a cor vermelha no sistema chamado de números hexadecimal (HEX). Cada cor é representada por um número hexadecimal. Você pode pesquisar na internet a tabela de cores, nela você encontrará todos os códigos hexadecimais para cada cor.

Um código hexadecimal para cores é formado por um sinal # seguido de seis dígitos e/ou letras. Existe mais de 1000 códigos HEX e não é fácil decorar o código para todas as cores.

Para algumas cores, você pode usar o nome das cores em inglês por exemplo (white, black, red, blue, green e yellow - branco, preto, vermelho, azul, verde, amarelo).

Voltando aos atributos:

```
<body style="background-color: red;">
```

Quais tags podem usar atributos?

Atributos podem ser aplicados à maioria das tags.

Você normalmente usará atributos com mais freqüência em algumas tags, tais como a tag body e raramente usará em outras, como por exemplo, a tag br que é um comando para pular de linha e não precisa de nenhuma informação adicional.

Assim como existem muitas tags, também existem muitos atributos. Alguns atributos são empregados em tags específicas enquanto outros servem para várias tags. E vice-versa: algumas tags podem conter somente um tipo de atributo, enquanto outras podem conter vários tipos.

Isto pode parecer um pouco confuso, mas à medida que você for praticando vai constatar que tudo é fácil e lógico, bem como vai verificar as inúmeras possibilidades que os atributos oferecem.

Então, quais são as partes que constituem uma tag?

A constituição básica de uma tag é denominada elemento (por exemplo em <p>). Assim, uma tag é constituida de um elemento (por exemplo <p>), ou por um elemento e um ou mais atributos (por exemplo <p style="background-color:#ff0000;">).

6. Links

Como construir links entre as páginas.

O que eu preciso para construir um link?

Para construir um link você usa o que tem usado até agora para codificar HTML: uma tag. Uma pequena tag com um elemento e um atributo é suficiente para você construir links para onde quiser. A seguir um exemplo de link para o site HTML.net:

```
<a href="http://www.html.com.br/">Aqui entra o nome do link</a>
```

Será renderizado assim no navegador:

Aqui entra o nome do link

O elemento ‘a’ refere-se a "anchor" - âncora . O atribuito href é abreviação para "hypertext reference" - referência a hypertexto - e especifica o destino do link - que normalmente é um endereço na Internet ou um arquivo.

No exemplo acima o atributo href tem o valor "http://www.html.net", que é o endereço completo do site HTML.net e é chamado de URL (Uniform Resource Locator). Notar que "http://" deve sempre ser incluido nas URLs. A frase "Aqui entra o nome do link" é o texto mostrado no navegador como link. Lembre-se de fechar a tag com um .

Como são os links entre minhas próprias páginas?

Se você quer construir links entre páginas de um mesmo website você não precisa escrever o endereço completo de cada página (URL). Por exemplo, se você tem duas páginas (vamos chamá-las de pagina1.htm e pagina2.htm) e salvou as duas em um mesmo diretório você constrói um link de uma para a outra usando somente o nome do arquivo no link. Nestas condições, um link da página1.htm para a pagina2.htm é como mostrado abaixo:

```
<a href="pagina2.htm">Aqui um link para a pagina 2</a>
```

Se a pagina2 for colocada em um subdiretório (chamado de "subdiretorio"), o link é como mostrado abaixo:

```
<a href="subdiretorio/pagina2.htm">Aqui um link para a pagina 2</a>
```

Por outro lado, um link da pagina2 no "subdiretorio" para a pagina1 é como mostrado a seguir:

```
<a href="../pagina1.htm">Aqui um link para a pagina 1</a>
```

"../" aponta para o diretório a um nível acima do arquivo onde foi feito o link. Seguindo o mesmo princípio você pode apontar para dois (ou mais) níveis acima, escrevendo "../../".

Como uma outra opção você pode usar sempre o endereço completo do arquivo (URL).

Como são os links dentro de uma mesma página?

Você pode criar links internos, dentro da própria página. Por exemplo, uma tabela de conteúdos ou índice com links para cada um dos capítulos em uma página. Tudo o que você precisa é usar o atributo id e o símbolo "#".

Use o atributo id para marcar o elemento que é o destino do link. Por exemplo:

```
<h1 id="heading1">Cabeçalho 1</h1>
```

Você agora pode criar um link que leve àquele elemento usando o símbolo "#" no atributo do link. O símbolo "#" informa ao navegador para ficar na mesma página que está. O símbolo "#" deve ser seguido pelo valor atribuido a id para onde o link vai. Por exemplo:

```
<a href="#heading1">Link para o cabeçalho 1</a>
```

Para ficar mais claro, vamos a um exemplo:

```
<html>
  <head>
  </head>
  <body>
    <p><a href="#heading1">Link para cabeçalho 1</a></p>
    <p><a href="#heading2">Link para cabeçalho 2</a></p>

    <h1 id="heading1">Cabeçalho 1</h1>
    <p>Texto texto texto texto</p>
    <h1 id="heading2">Cabeçalho 2</h1>
    <p>Texto texto texto texto</p>
  </body>
```

Será renderizado no navegador assim (clique nos dois links):

Link para cabeçalho 1

Link para cabeçalho 2

Cabeçalho 1

Texto texto texto texto

Cabeçalho 2

Texto texto texto texto

Obs.: O nome de um atributo id deve começar com uma letra

Links para um endereço de email

```
<a href="mailto:nobody@html.net">Enviar e-mail para nobody em HTML.net</a>
```

Será renderizado no navegador assim:

Enviar e-mail para nobody em HTML.net

A única diferença é que no lugar do endereço do documento você escreve mailto: seguido pelo endereço de e-mail. Quando o link é clicado o programa de e-mail padrão do usuário é aberto com o endereço do link já digitado na linha para destinatário. Mas, atenção, isto só irá funcionar se o usuário tiver um programa de e-mail instalado na sua máquina. Como por exemplo o Outlook.

Existem outros atributos que eu deva conhecer?

Par criar links você sempre usa o atributo href. Adicionalmente você pode usar um title (título) para seu link:

```
<a href="http://www.html.net/" title="Visite HTML.net e aprenda HTML">HTML.net</a>
```

Será renderizado no navegador assim:

HTML.net

O atributo title é usado para fornecer uma breve descrição do link. Se você - sem clicar no link - colocar o cursor do mouse sobre o link, vai aparecer o texto "Visite o site HTML.net e aprenda HTML".

7. Imagens

O que você acha de colocar uma imagem no centro da sua página?

Tudo o que você precisa é da nossa conhecida tag.

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

Será renderizado no navegador assim:

O que você tem a fazer é dizer ao navegador que quer inserir uma imagem (img) e depois informar onde a imagem esta localizada (src, abreviatura para "source" - local de armazenagem).

Notar que a tag imagem é do tipo comando isolado, isto é, uma só tag de abertura e fechamento. Semelhante a tag <br /> e <hr />que não precisa de um texto inserido nela.

"bandeiradobrasil.jpg" é o nome do arquivo da imagem que você quer inserir na página. ".jpg" é a extensão do tipo de imagem. Tal como a extensão “.html” ou".htm" para arquivos de documentos HTML, ".jpg" informa ao navegador que o arquivo é uma imagem. Veja abaixo os três os tipos de imagens que você pode inserir na sua página:

- GIF (Graphics Interchange Format)
- JPG / JPEG (Joint Photographic Experts Group)
- PNG (Portable Network Graphics)

Em geral imagens GIF são melhores para gráficos e desenhos e imagens JPEG são melhores para fotografia. Existem duas razões para isto: primeiro, imagens GIF são constituidas por 256 cores, e imagens JPEG por milhões de cores, segundo, imagens GIF são melhores otimizadas para imagens simples ao passo que imagens JPEG são melhores otimizadas para imagens complexas. Quanto maior a compressão tanto menor o tamanho do arquivo e tanto mais rápido a página é carregada no navegador.

Tradicionalmente os formatos GIF e JPEG tem sido os mais empregados, mas ultimamente o formato PNG tem se tornado cada vez mais popular. O formato PNG é em vários aspectos melhor que os formatos JPEG e GIF: milhões de cores e efetiva compressão.

Onde consigo minhas imagens?

Para criar suas próprias imagens você precisa de um programa de edição de imagens. O Adobe Photoshop e o Adobe Fireworks são excelentes para fazer um papel de parede para a sua página por exemplo. (No final da apostila vou ensinar como faz multiplicar um plano de fundo, de forma que ele cubra com as mesmas imagens todo o fundo do seu website). Um programa de edição de imagens é a ferramenta essencial para criação de websites com grande impacto e apelo visual.

Vamos aprender mais algumas coisas sobre imagens.

Primeiro, você pode inserir imagens que estão localizadas em outros diretórios ou até mesmo em outros websites:

```
<img src="images/caixa.png">
```

```
<img src="http://www.html.com.br/caixa.png">
```

Segundo, imagens podem ser links:

```
<a href="http://www.html.com.br">
<img src="caixa.png"></a>
```

Será renderizado no navegador assim:

Existem outros atributos que eu deva conhecer?

Você sempre terá que usar o atributo ‘src’, que diz ao navegador onde a imagem está localizada. Além dele existem alguns outros atributos que podem ser bastante úteis quando você insere imagens em uma página.

O atributo ‘alt’ é usado para fornecer uma descrição textual alternativa da imagem caso por alguma razão a imagem não seja renderizada para o usuário. Isto é particularmente importante para usuários com restrições visuais ou quando a imagem é carregada muito lentamente. Em conseqüência, sempre use o atributo alt:

```
<img src="logo.gif" alt="logotipo do BR Masters">
```

Alguns navegadores mostram uma caixa pop-up com o conteúdo do atributo alt quando o usuário passa o mouse sobre a imagem. Tenha em mente que a finalidade principal do atributo alt é a de fornecer uma alternativa textual para imagem. O atributo alt não deve ser usado para criar mensagens pop-up uma vez que os leitores de tela passarão uma mensagem que não descreve a imagem para os usuários com restrições visuais.

O atributo title pode ser usado para fornecer uma curta descrição da imagem:

```
<img src="bandeira.gif" title="Aprenda HTML no site HTML.com.br">
```

Será renderizado no navegador assim:

Coloque o ponteiro do mouse sobre a imagem, sem clicar e aparecerá uma caixa pop-up com o texto "Aprenda HTML no site HTML.net".

Dois outros atributos importantes são width e height:

```
<img src="logo.png" width="141" height="32">
```

Os atributos width e height podem ser usados para definir respectivamente, a largura e a altura da imagem. O valor adotado para medidas é o pixel. Pixel é a unidade de medida usada para medir a resolução da tela. (As resoluções de tela mais comuns são de 800x600 e 1024x768 pixels, apesar das resoluções 800x600 estarem ficando para trás e sendo cada vez mais usada as de 1024x768 e 1280x1024 ou superior). Ao contrário de centímetros, pixel é uma unidade de medida relativa que depende da resolução da tela. Usuários com grande resolução de tela terão 25 pixels em 1 centímetro de tela enquanto aqueles com baixa resolução de tela terão os mesmos 25 pixels em 1,5 cm de tela.

Se não forem definidos os valores para width e height, a imagem será inserida com seu tamanho real. Com width e height você pode alterar o tamanho da imagem:

```
<img src="logo.gif" width="32" height="32">
```

O tempo de descarga da imagem será sempre aquele requerido como se ela tivesse suas dimensões reais, mesmo que você diminua seu tamanho com uso destes atributos. Assim, você não deve diminuir o tamanho das imagens com o uso dos atributos width e height. Se você precisa diminuir a imagem diminua suas dimensões reais em um editor de imagem para tornar suas páginas mais leves e mais rápidas.

Dito isto, acrescentamos que é sempre uma boa idéia definir os atributos width e height para imagens, pois assim fazendo o navegador reservará o espaço para descarga da imagem previamente. Isto acaba por permitir ao navegador, saber com antecedência (antes de descarregar as imagens) como será o layout da página.

8. Tabelas

Tabelas são usadas para apresentar "dados tabulares", isto é, informação que deva ser apresentada em linhas e colunas, de forma lógica e organizada.

É difícil?

Criar tabelas em HTML pode parecer complicado, mas quando você acompanhar passo a passo a explicação, verá que é bem fácil.

```
<table>
  <tr>
    <td>Célula 1</td>
    <td>Célula 2</td>
  </tr>
  <tr>
    <td>Célula 3</td>
    <td>Célula 4</td>
  </tr>
</table>
```

Será renderizado no navegador assim:

Célula 1 Célula 2
Célula 3 Célula 4

Qual a diferença entre as tags <tr> e <td>?

Este com certeza é o código mais complicado até agora. Vamos analisar isto por partes e explicar as diferentes tags:

3 tags básicas são usadas para inserir tabelas:

- <table> começa e termina uma tabela.
- <tr> significa "table row" - linha de tabela - começa e termina e uma linha horizontal da tabela.
- <td> significa "table data" - dados da tabela. começa e termina cada célula contida nas linhas da tabela.

Eis o acontece no exemplo dado acima: a tabela começa com <table>, segue-se uma <tr>, que indica o início de uma nova linha. Duas células são então inseridas na linha: <td>Célula 1</td> e <td>Célula 2</td>. A linha termina com </tr> e uma nova linha <tr> começa imediatamente a seguir. A nova linha também contém duas células. A tabela termina com </table>.

Para esclarecer: linhas são horizontais e colunas são verticais, ambas contendo células:

Célula 1	Célula 2
Célula 3	Célula 4

Célula 1 e Célula 2 formam uma linha. Célula 1 e Célula 3 formam uma coluna.

No exemplo acima a tabela tem duas linhas e duas colunas. Uma tabela pode conter um número ilimitado de linhas e colunas.

Outro exemplo:

```
<table>
  <tr>
    <td>Célula 1</td>
    <td>Célula 2</td>
    <td>Célula 3</td>
    <td>Célula 4</td>
  </tr>
  <tr>
    <td>Célula 5</td>
    <td>Célula 6</td>
    <td>Célula 7</td>
    <td>Célula 8</td>
  </tr>
  <tr>
    <td>Célula 9</td>
    <td>Célula 10</td>
    <td>Célula 11</td>
    <td>Célula 12</td>
  </tr>
</table>
```

Será renderizado no navegador assim:

Célula 1	Célula 2	Célula 3	Célula 4
Célula 5	Célula 6	Célula 7	Célula 8
Célula 9	Célula 10	Célula 11	Célula 12

Existem atributos?

Claro! Por exemplo, o atributo border que é usado para definir a espessura de uma borda em volta da tabela:

```
<table border="1">
  <tr>
    <td>Célula 1</td>
    <td>Célula 2</td>
  </tr>
  <tr>
    <td>Célula 3</td>
    <td>Célula 4</td>
  </tr>
</table>
```

Será renderizado no navegador assim:

Célula 1	Célula 2
Célula 3	Célula 4

A borda da tabela é especificada em pixels.

Tal como fizemos com as imagens, podemos definir width - largura - em pixels, para uma tabela - ou alternativamente em percentagem da tela. Veja abaixo:

```
<table border="1" width="30%">
```

Este exemplo renderiza no navegador uma tabela com largura igual a 30% da largura da tela.

Existem mais atributos?

Existe uma grande quantidade de atributos para tabelas. A seguir mais dois:

- align: define o alinhamento horizontal do conteúdo da tabela, de uma linha ou de uma célula. Por exemplo, left, centre ou right (à esquerda, no centro ou à direita).
- valign: define o alinhamento vertical do conteúdo de uma célula. Por exemplo, top, middle ou bottom (em cima, no meio ou em baixo).

```
<td align="right" valign="top">Célula 1</td>
```

O que posso inserir em tabelas?

Teoricamente você pode inserir qualquer coisa em uma tabela: texto, links e imagens... MAS, tabelas tem por objetivo apresentar dados tabulares (isto é, dados que por sua natureza devam ser apresentados em linhas e colunas) então abstenha-se de colocar coisas dentro de tabela simplesmente porque você deseja que elas estejam próximas umas às outras.

Nos primórdios da Internet - isto é, há poucos anos atrás - tabelas eram usadas como ferramenta para construir layout. Se você quer controlar a apresentação de textos e imagens, existe uma maneira bem mais racional (dica:CSS ou tableless). Veremos isso mais a frente em CSS e tableless.

9. Mais Tabelas

Nesta segunda parte vamos dar continuação e aprender mais sobre as tabelas.

O que mais existe?

Os dois atributos colspan e rowspan são usados para criar tabelas singulares.

Colspan é a abreviação para "column span". Colspan é usada na tag <td> para indicar quantas colunas estarão contidas em uma célula.

```
<table border="1">
  <tr>
    <td colspan="3">Célula 1</td>
  </tr>
  <tr>
    <td>Célula 2</td>
    <td>Célula 3</td>
    <td>Célula 4</td>
  </tr>
</table>
```

Será renderizado no navegador assim:

Célula 1		Célula 2
Célula 3	Célula 4	Célula 5

E o rowspan?

Como você já deve ter concluido, rowspan especifica quantas linhas estarão contidas em uma célula:

```
<table border="1">
  <tr>
    <td rowspan="3">Célula 1</td>
    <td>Célula 2</td>
  </tr>
  <tr>
    <td>Célula 3</td>
  </tr>
  <tr>
    <td>Célula 4</td>
  </tr>
</table>
```

Será renderizado no navegador assim:

Célula 1	Célula 2
	Célula 3
	Célula 4

No exemplo acima rowspan é definido em "3" na Célula 1. Isto especifica que uma célula deve conter 3 linhas. Célula 1 e Célula 2 estão na mesma linha, enquanto Célula 3 e Célula 4 formam duas linhas independentes.

Isso não é difícil, é confuso. É bom desenhar a tabela em uma folha de papel antes de começar a codificação HTML.

10. Uploading do seu site

Até agora somente você conseguiu visualizar suas páginas. Chegou a hora de mostrá-las para o mundo todo.

Para publicar a sua página na Web você precisará apenas de um espaço em um servidor. Para facilitar o seu upload você poderá baixar um programa FTP(opcional) em qualquer site de download de programas. Ele faz o upload das páginas sem precisar acessar toda vez o site do servidor. Você acessará diretamente o servidor do seu computador.

Na internet você já deve ter visto vários servidores gratuitos para hospedagem. Seu endereço ficará cumprido algo parecido com http://home.servidor.com/nomedousuario. Mas, existe uma maneira de redirecionar o endereço para que fique mais curto. Como por exemplo: www.seusite.com.br. Porém, estes, com domínios: .com.br, .com, .net etc são pagos. Existem gratuitos que deixam parecido com isso: www.seusite.k10.com.br (k10 – é o nome da empresa que está hospedando). Bem melhor do que deixar aquele endereço enorme.

Depois de cadastrado, para acessar o seu endereço FTP, de um programa FTP, você precisará de apenas 3 coisas. O nome do servidor FTP (por exemplo: ftp.br.geocities.com.br), o seu nome de usuário (por exemplo: seunome@yahoo.com.br) e sua senha.

11. Webstandards e validação

Nesta lição você aprenderá mais alguns conceitos teóricos do HTML.

O que mais há para conhecer sobre HTML?

HTML pode ser escrito de várias maneiras. O navegador está apto a ler HTML escrito de várias maneiras. Podemos dizer que HTML tem muitos dialetos. Esta é a razão porquê alguns websites são apresentados de formas diversas em diferentes navegadores.

Desde o aparecimento da Internet tem sido feitas várias tentativas para se normatizar o HTML notadamente através do World Wide Web Consortium (W3C) fundado por Tim Berners-Lee (o grande inventor do HTML).

No passado - quando você tinha que comprar um navegador - Netscape dominava o mercado de navegadores. Àquela época as normas para o HTML estavam nas suas versões 2.0 e 3.2. Mas pelo fato de dominar 90% do mercado a Netscape não teria, e não teve que se preocupar muito com normas. Pelo contrário, a Netscape inventava seus próprios elementos de marcação que não funcionavam em outros navegadores.

Por muitos anos a Microsoft ignorou completamente a Internet. Em determinado momento, resolveu competir com a Netscape e lançou seu navegador próprio. A primeira versão do navegador da Microsoft, o Internet Explorer, não era melhor do que o Netscape no suporte às normas do HTML. Mas, a Microsoft resolveu distribuir seu navegador gratuitamente junto com o Sistema Operacional Windows e o Internet Explorer em pouco tempo tornou-se o navegador mais usado e mais popular.

A partir das versões 4 e 5 a Microsoft anunciava que seus navegadores ofereciam cada vez maior suporte às normas HTML do W3C. A Netscape não se movimentou para atualizar seu navegador e continuou a colocar no mercado a velha e desatualizada versão 4.

Quando você codifica HTML de acordo com as normas do W3C, você está construindo um website para ser visualizado em todos os navegadores, não só agora mas também no futuro. E felizmente, tudo o que você aprendeu neste tutorial está de acordo com a mais nova versão do HTML, que é o XHTML.

Devido a existência de diferentes tipos de HTML, você precisa informar ao navegador qual é o "dialeto" do HTML e no seu caso você aprendeu XHTML. Para informar ao navegador, você usa o Document Type Definition. O Document Type Definition deve ser escrito sempre no topo do documento:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" lang="pt-br">

<head>
<title>Título</title>
</head>

<body>
<p>texto texto</p>
</body>

</html>
```

Além do Document Type Definition (escrito na primeira linha no exemplo acima), que informa ao navegador que você está codificando XHTML, você precisa ainda adicionar informação extra na tag html, usando os atributos xmlns e lang.

xmlns é abreviação de "XML-Name-Space" e deve ter sempre o valor http://www.w3.org/1999/xhtml. Isto é tudo o que você precisa saber.

No atributo lang você específica em que língua (aqui trata-se de linguagem humana) o documento é escrito. As abreviaturas para as línguas existentes no mundo todo, estão nas ISO 639 standard . No exemplo acima a língua definida no atributo é o português do Brasil ("pt-br").

Validação? Porquê deveria eu fazer isto?

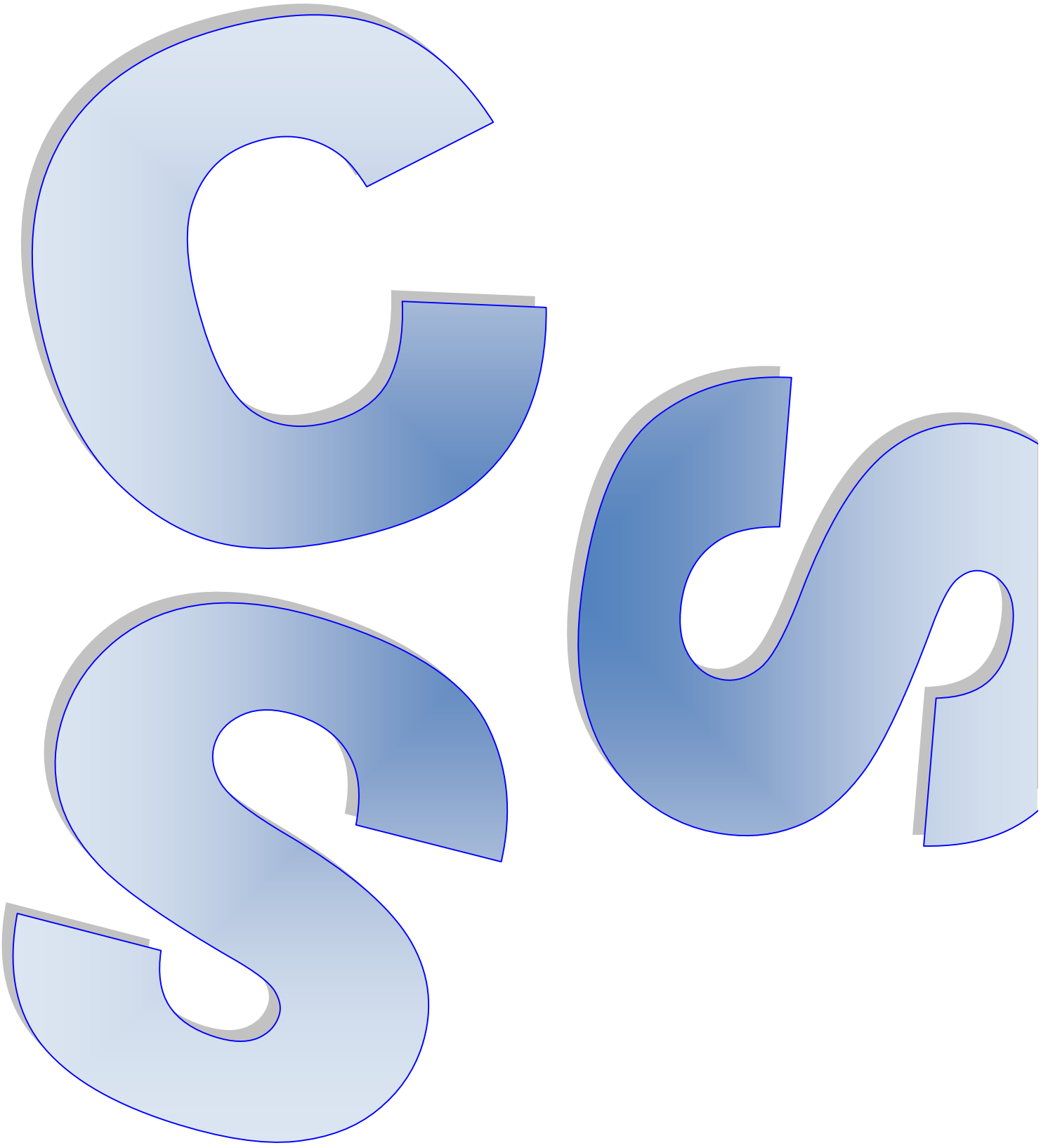
O DTD é importante também para a validação da página.

Insira o DTD nas suas páginas e poderá verificar erros no seu HTML, usando o validador gratuito do W3C.

Para testar o validador faça o seguinte: crie uma página e publique na Internet. A seguir entre em validator.w3.org e lá digite o endereço (a URL) da sua página, depois clique no botão validar. Se seu HTML estiver correto, vai aparecer uma mensagem de congratulações. Se não, será apresentada uma lista de erros, informando o quê está errado e onde. Cometa alguns erros propositais no seu código para verificar o que acontece.

O validador não é útil somente no encontro de erros. Alguns navegadores tentam interpretar os erros cometidos pelos desenvolvedores e consertar o código mostrando a página corretamente. Em navegadores assim você nunca encontrará erros no próprio navegador. Já outros navegadores não aceitam o erro e apresentam a página arruinada ou mesmo nem apresentam. O validador então ajuda você a encontrar erros que você não tenha nem idéia de que existiam

Sempre valide suas páginas, para ter certeza que elas serão mostradas corretamente em todos os navegadores.



1. Introdução ao CSS

CSS significa Cascading Style Sheetes (Folhas de Estilo em Cascata). Não seria ótimo fazer layouts na sua página sem ter que alterar o HTML toda vez que quisesse mudar?

Nesta introdução vamos ter uma breve introdução ao CSS para você ter uma idéia como o CSS é prático. Podemos dizer que o CSS é a melhor metade do HTML. Codificando, não há melhor parceria: HTML é responsável pelo trabalho pesado (a estrutura), enquanto CSS dá o toque de elegância (layout).

CSS pode ser adicionado com uso do atributo style. Por exemplo, você pode definir 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:

Este parágrafo em tamanho de fonte igual a 20px

Este parágrafo em fonte Courier

Este parágrafo em fonte Cambria e tamanho 16px

No exemplo acima usamos 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 definimos tanto o tipo como o tamanho da fonte separados por um ponto e vírgula.

Está parecendo que há uma grande quantidade de trabalho a executar

Uma das funcionalidades mais inteligentes das CSS é a possibilidade de controlar o layout de um arquivo central. Em lugar de se usar o atributo style em cada tag, você pode 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 use a tag <style type="text/css"> que informa ao navegador que você 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.

Uma outra opção é a de digitar as CSS em um documento separado. Com as CSS em um documento separado você pode gerenciar o layout de muitas páginas ao mesmo tempo. Muito inteligente, pois você pode mudar de uma só vez, o tipo ou o tamanho da fonte de todo o site, quer ele tenha centenas ou milhares de páginas. Nós não nos aprofundaremos em CSS agora, mas você pode aprender tudo, no futuro, em nosso tutorial CSS.

O que mais posso fazer com CSS?

CSS faz muito mais do que definir tipos e tamanhos de fontes. Por exemplo, você pode adicionar cores e fundos. A seguir alguns exemplos para você praticar:

```
<p style="color:green;">Texto verde</p>

<h1 style="background-color: blue;">Cabeçalho com fundo azul</h1>

<body style="background-image: url('http://www.html.net/logo.png');">
```

Além de cores, tipos de fontes, etc., CSS pode ser usado para controlar a configuração e a apresentação da página (margens, flutuações, alinhamentos, larguras, alturas, etc.) Controlando os diferentes elementos com CSS você será capaz de criar layouts elegantes e precisos.

```
<p style="padding:25px;border:1px solid red;">Eu amo CSS</p>
```

Será renderizado no navegador assim:

Eu amo CSS

Com a propriedade float um elemento poderá "flutuar" à esquerda ou à direita. O exemplo seguinte ilustra este princípio:

```
<img src="bandeiradobrasil.jpg" alt="Bandeira do Brasil" style= "float:left;" />

<p>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX,
        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX...</p>
```

Será renderizado no navegador assim:

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

No exemplo mostrado, um elemento (a imagem) flutua à esquerda e o outro elemento (o texto) preenche o espaço deixado à direita.

Com a propriedade position, você pode posicionar um elemento em qualquer lugar da página, com precisão:

```
<img src="bill.jpg" alt="Bill Gates"
style="position:absolute;bottom:50px;right:10px;" />
```

No exemplo mostrado a imagem foi posicionada a 50 pixels da borda inferior e a 10 pixels da borda direita do navegador. Você pode colocar em qualquer lugar na página.

Você não aprende CSS em 10 minutos. É preciso se dedicar um tempo para estudar. Mas não é nada difícil, vamos ver nas próximas páginas o que você precisa saber sobre as CSS.

2. Como funciona as 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. Se você está acostumado a usar HTML para layout irá reconhecer muitos dos códigos que usaremos.

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:

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”)

Mas, onde colocamos 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. Recomendo que você foque 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 ele não ficará amostra se alguém olhar o código fonte da sua página pelo browser, pois estará 'escondido' no servidor onde está hospedando o seu 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. 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 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 (índex.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" />
```

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!

Abra o bloco de notas (ou equivalente em outro sistema operacional) e 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="style.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. Parabéns! Você construiu sua primeira folha de estilos!

3. Cores e fundos

Você vai aprender agora 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.

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;
}
```

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;
}
```

Obs.: Note 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.

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.gif");
}

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

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

Imagem de fundo repetida [background-repeat]

No exemplo anterior você observou 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.

Value	Description
`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 na 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("butterfly.gif");
        background-repeat: no-repeat;
}

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

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 imagem que rola.

Value	Description
`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.

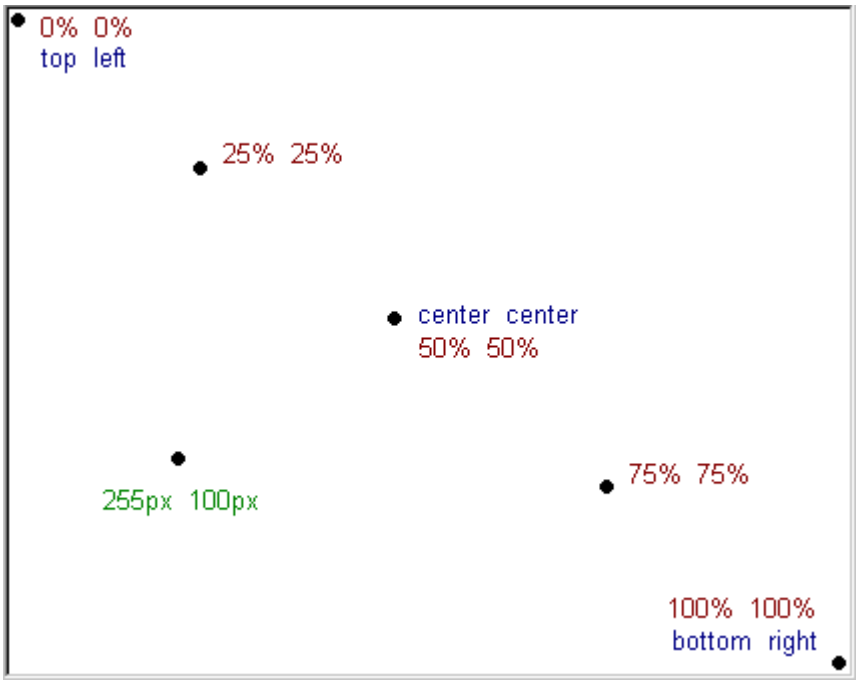
```
body {
        background-color: #FFCC66;
        background-image: url("butterfly.gif");
        background-repeat: no-repeat;
        background-attachment: fixed;
}

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

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
`background-position: 2cm 2cm`	A imagem é posicionada a 2 cm da esquerda e 2 cm 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("butterfly.gif");
        background-repeat: no-repeat;
        background-attachment: fixed;
        background-position: right bottom;
}

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

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("imagem.gif") no-repeat fixed right bottom;
```

A declaração abreviada deve seguir a seguinte ordem:

background-colo

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 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).

4. FONTES

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.

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.

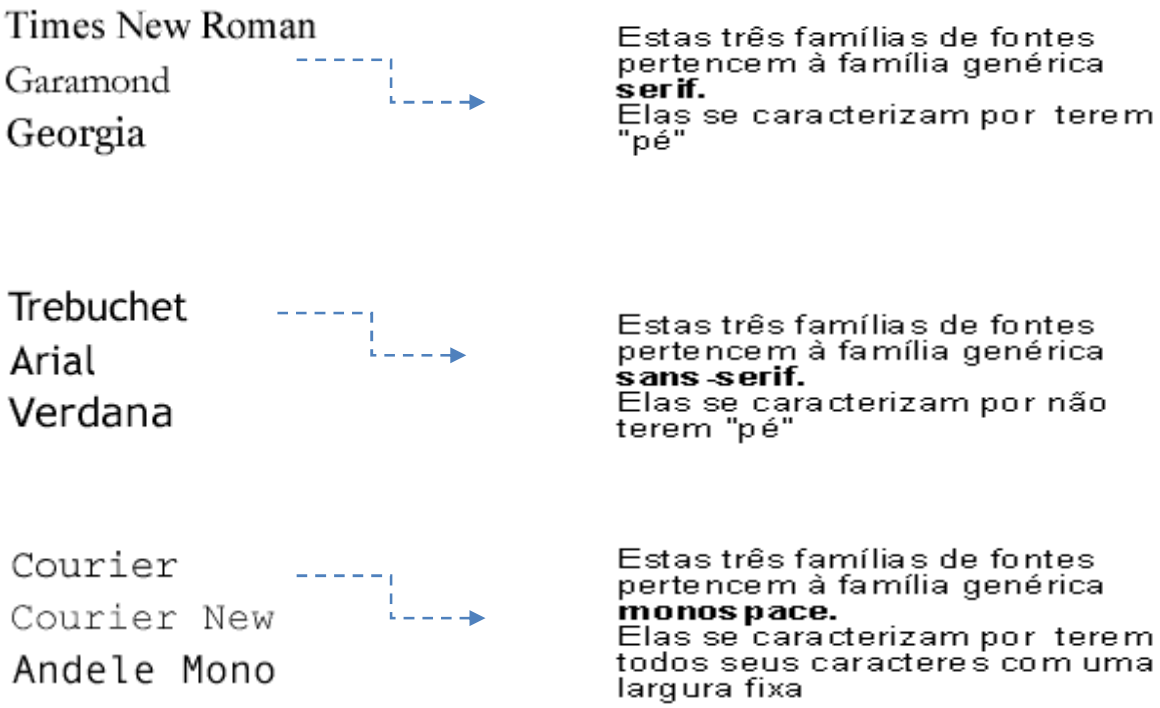
Existem dois tipos de nomes para definir fontes: nomes para famílias de fontes e nomes para famílias genéricas. Os dois são explicados a seguir:

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".



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;}
```

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.

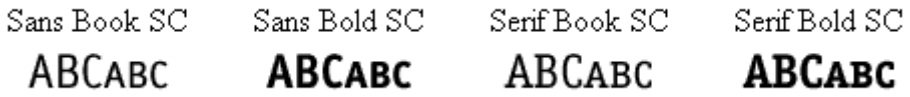
Estilo da fonte [font-style]

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;}
```

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;}
```

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;}
```

Tamanho da fonte [font-size]

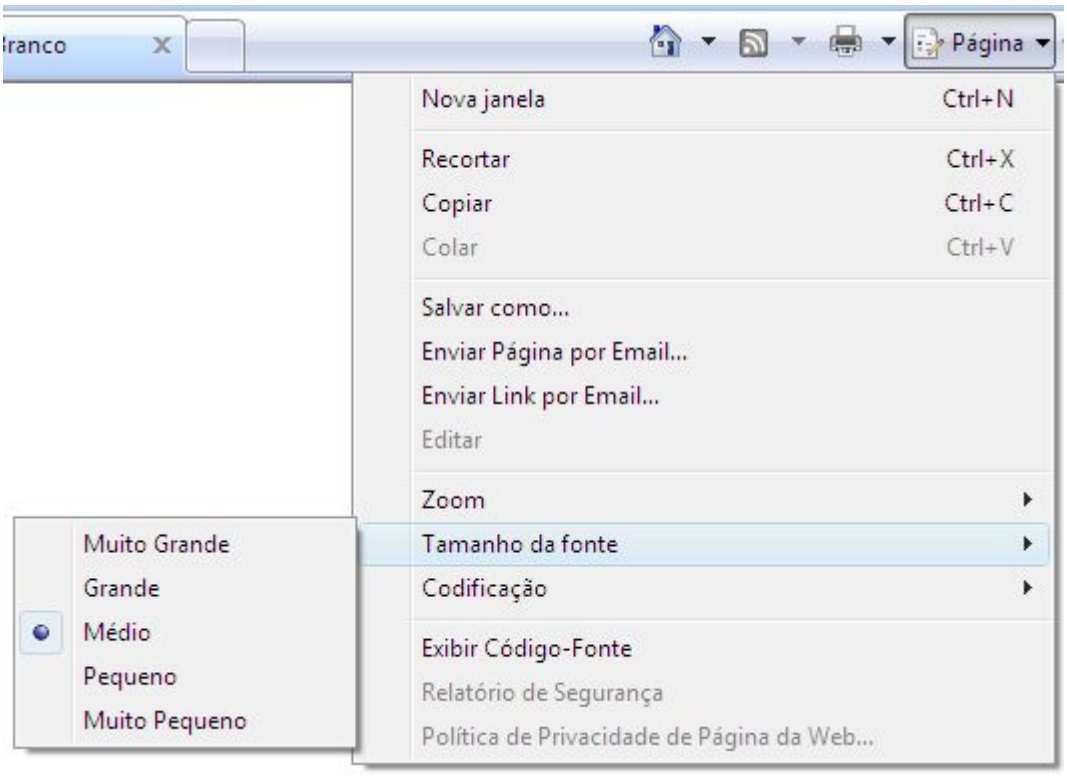
O tamanho da fonte é definido pela propriedade font-size.

Existem muitas unidades (p. ex.:. pixels e porcentagens) 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;}
```

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 seu site acessível a todos, você deverá usar unidades como '%' ou 'em'.

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

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:

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;
}
```

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;
}
```

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;
}
```

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;
```

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;
}
```

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 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;
}
```

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;
}
```

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;
}
```

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;
}
```

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 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

Na lição anterior 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;
}
```

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. 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 anterior, 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;
}
```

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 vamos ver agora.

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.

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

*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.

*Agrupando com *

Enquanto é usado dentro de um elemento nível de bloco como vimos 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. 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:

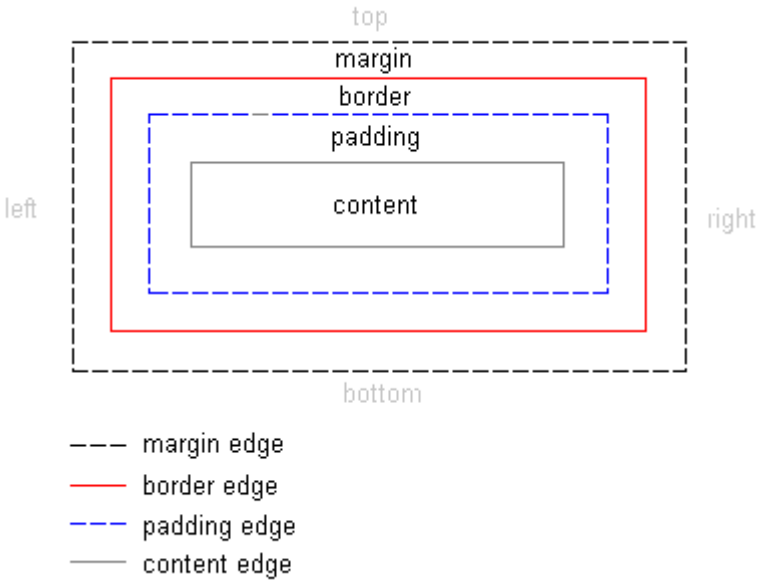
```
#democrats {
        background:blue;
}

#republicans {
        background:red;
}
```

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:

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:

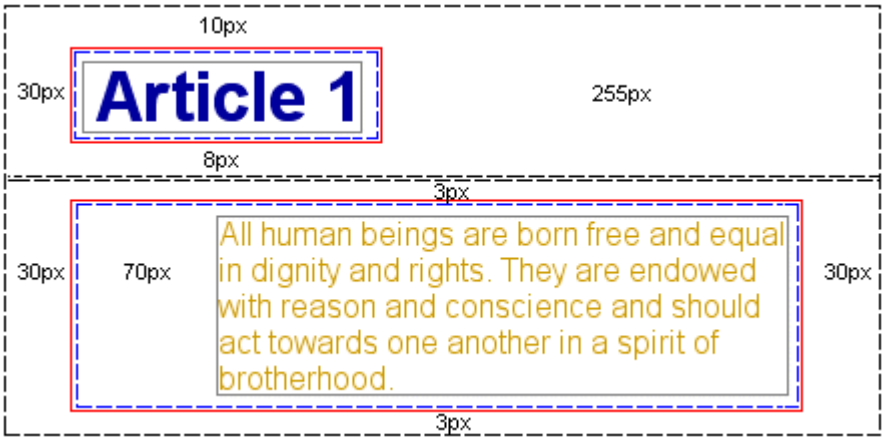
```
<h1>Article 1:</h1>

<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:

Article 1

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.

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.

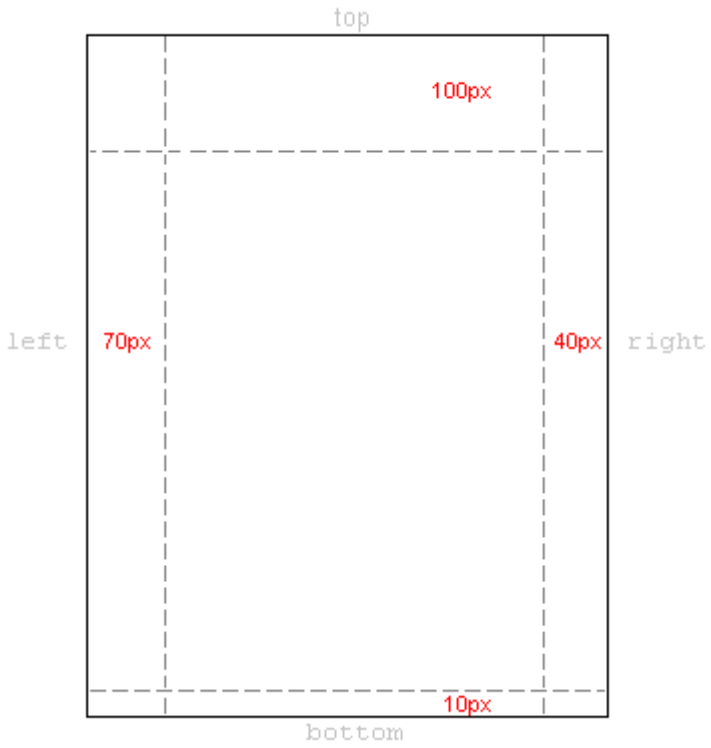
10. Margin e padding

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

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;
}
```

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:

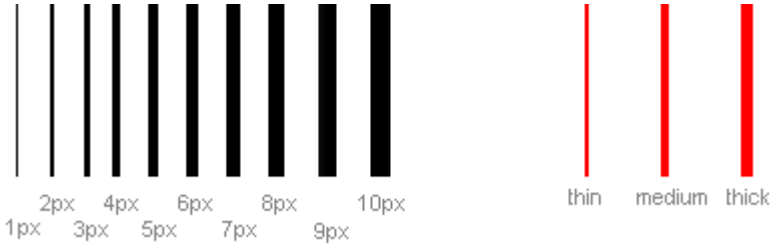
```
h1 {
        background: yellow;
        padding: 20px 20px 20px 80px;
}

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.

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:



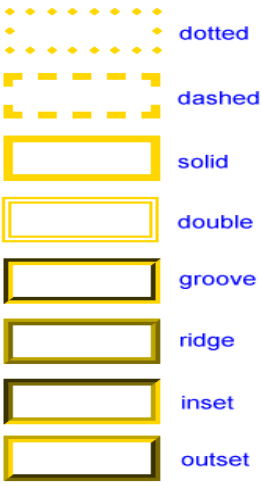
As cores das bordas [border-color]

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;
}
```

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;
}
```

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.

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;
}
```

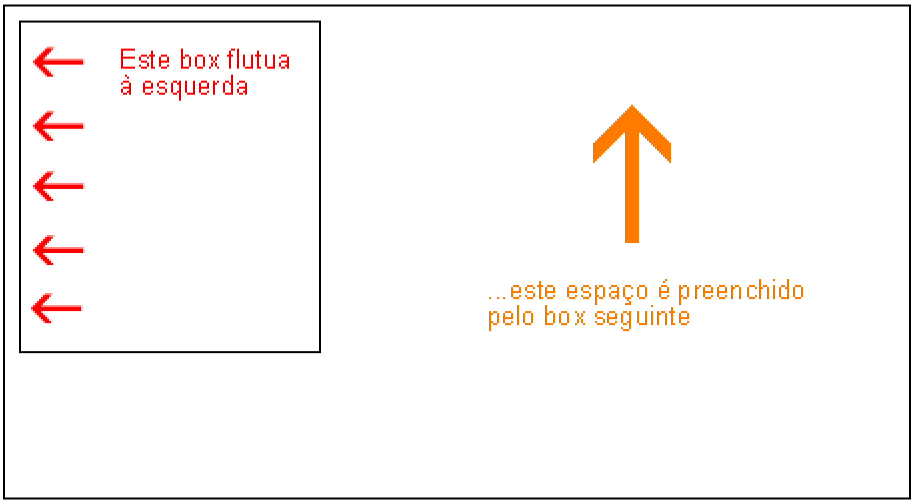
Atribuindo altura [height]

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) (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:


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

<p>xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx,
xxxxxxxxxxxxxxxxxxxxxxxxxx...</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;
}
```

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>xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
        xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx...</p>
</div>

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

<div id="column3">
        <p>xxxxxxxxxxxxxxxxxxxxxxxxx
        xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx...</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.

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:

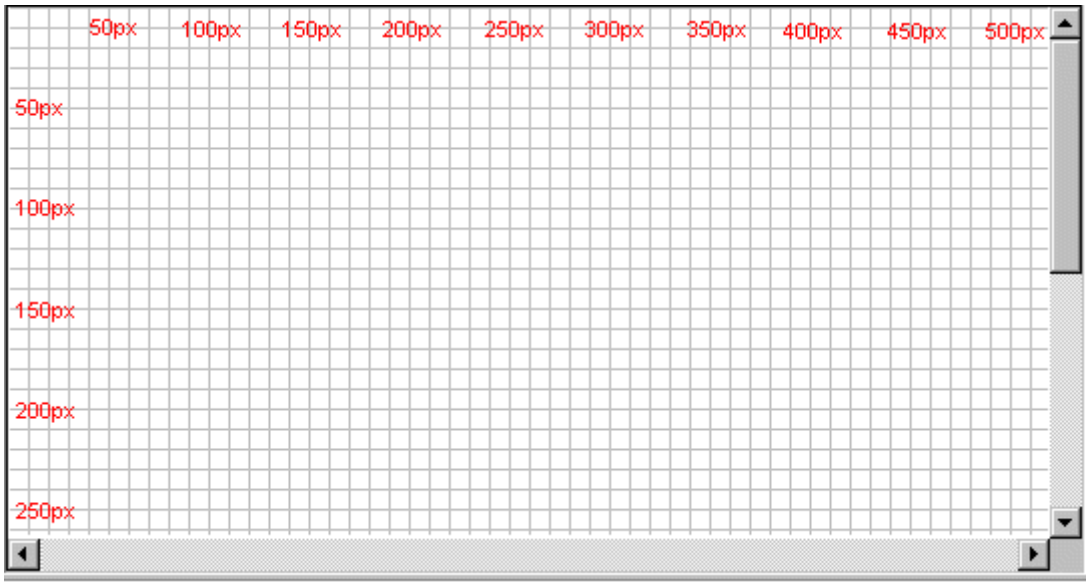
```
#picture {
        float:left;
        width: 100px;
}

.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.

O princípio de posicionamento CSS

Considere a janela do navegador como um sistema de coordenadas:



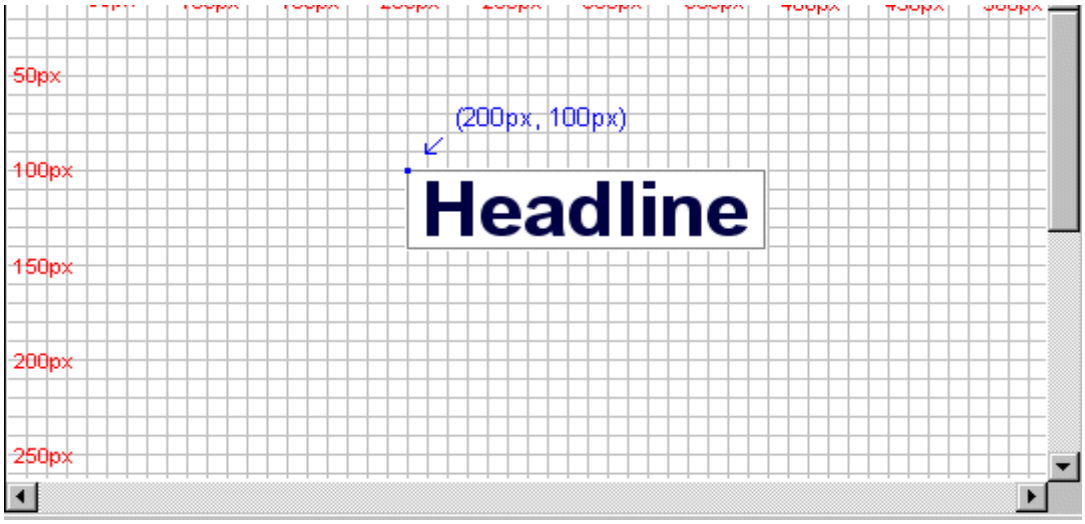
O princípio de posicionamento CSS estabelece que você pode posicionar um elemento em qualquer lugar na tela usando um sistema de coordenadas.

Vamos supor que queremos posicionar um cabeçalho. Usando o box model (veja na lição 9) 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.

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;
}
```

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:

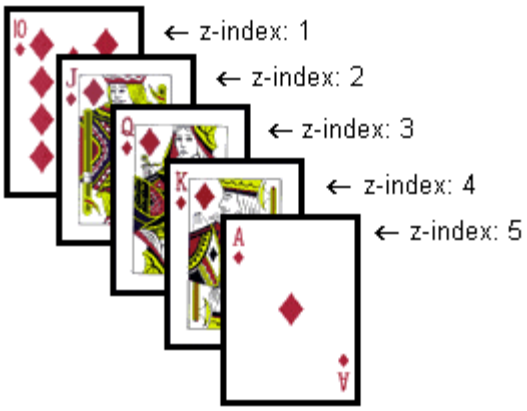
```
#bola1 {
	position:relative;
	left: 350px;
	bottom: 150px;
}
#bola2 {
	position:relative;
	left: 150px;
	bottom: 500px;
}

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

15. Usando 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:

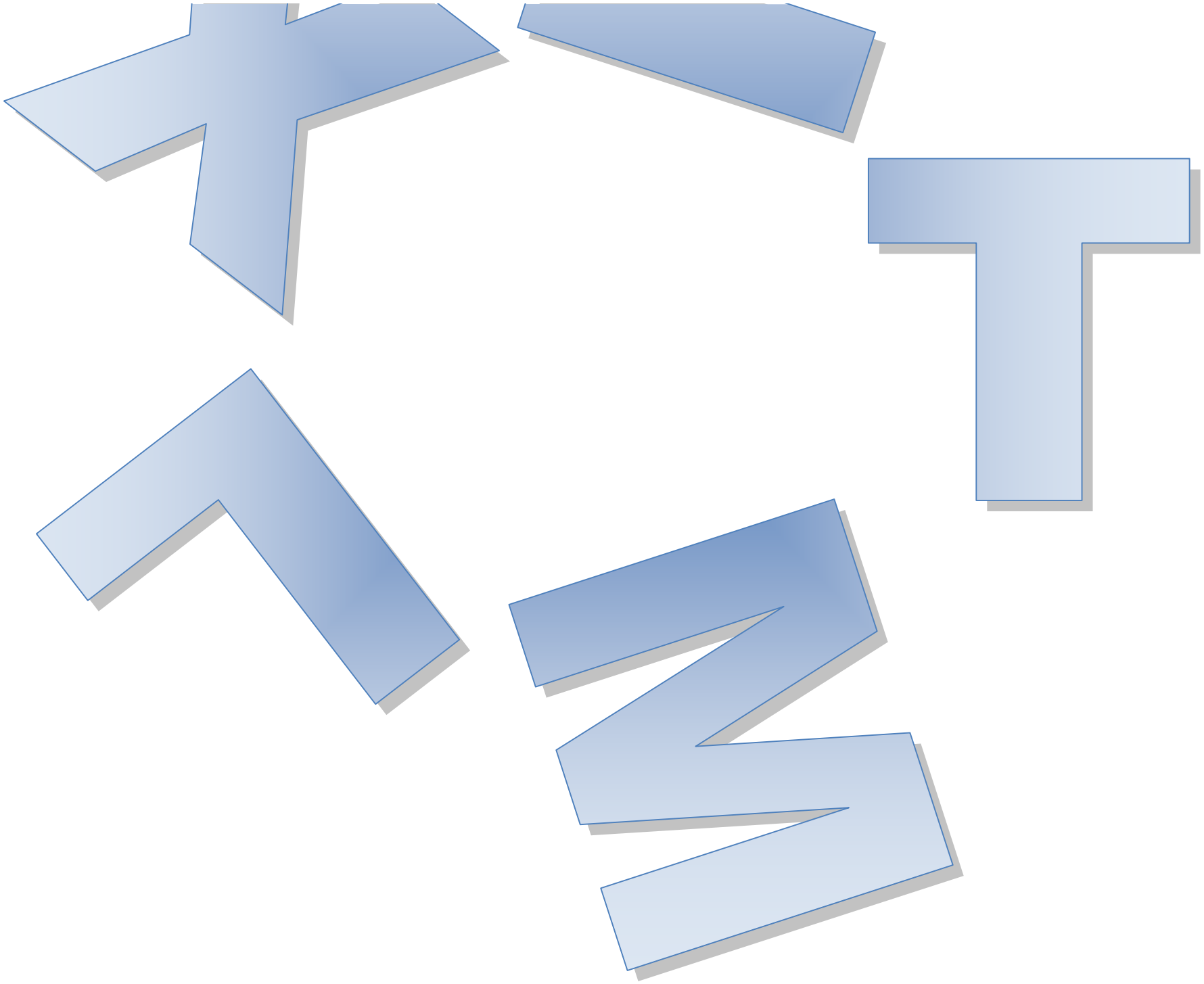


No caso mostrado, os números índice estão em uma seqüência direta (de 1-5), contudo o mesmo resultado poderia ser obtido com uso de 5 diferentes números, não em seqüência. O que conta é a cronologia dos números (a ordem).

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

```
#ten_of_diamonds {
        position: absolute;
        left: 100px;
        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.



INTRODUÇÃO AO XHTML

Para que serve o XHTML? Um arquivo XHTML é um arquivo HTML, que pode ser lido por qualquer browser. Não estamos falando de um novo HTML, com novas tags ou coisa assim. Pelo contrário, o XHTML foi feito para funcionar mesmo em navegadores antigos. Mas, ao mesmo tempo, Um arquivo XHTML é também um arquivo XML válido, que pode ser lido por qualquer interpretador de XML.

Meu primeiro conselho, nesse caso, é que você, se não trabalha com XML, deixe preocupação com o XHTML para depois de dominar bem o código semântico e o layout tableless. Não porque seja complicado, pelo contrário, transformar HTML em XHTML é bem simples, mas simplesmente porque você pode obter benefícios muito significativos, e muito mais rapidamente, aprendendo CSS do que XHTML.

O segundo conselho é que você comece a estudar o assunto. Depois de dominar bem layouts CSS, mergulhe no XML. A maioria dos bancos de dados hoje permite extrair dados diretamente em XML e todas as plataformas de aplicações web trabalham bem com XML. E com a poderosa linguagem XSLT você pode muito facilmente oferecer seus os dados em XHTML para o navegador.

Para que seu arquivo possa ser lido por máquinas além de humanos é muito importante que você escreva um XHTML válido, com isso você está fazendo com que as informações do seu site fique mais acessível para as buscas, contribuindo para o projeto e principalmente melhorando as visitas do seu site.

O QUE É O DOC TYPE?

O Doctype (Document Type Definition) é a primeira coisa que se deve escrever em um arquivo XHTML, ele vai na primeira linha do seu documento, se você quiser ter um XML válido, não devemos esquecê-lo, ele serve para informar ao browser que tipo de documento será visualizado.

Existem 3 tipos:

- Strict: Este tipo é usado quando você fez um código 100% XHTML, sem erros, deve ser escrito assim:
  <!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

- Transitional: Este é o modo mais usado, você o usa quando está começando a migrar do nosso amigo HTML para o poderoso XHTML, sua sintaxe é:
  <!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

- Frameset: É usado quando você está utilizando FRAMES em seu site, se escreve assim:
  <!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">

Veja o exemplo abaixo:

EXEMPLO:

```
<!DOCTYPE html
PUBLIC “-//W3C//DTD XHTML 1.0 Strict//EN”
“http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd”>
<html>
<head>
<title></title>
</head>
<body>
...
</body>
</html>
```

FECHE TODAS AS TAGS

Quem já escreveu algum XML sabe que ele não funciona até que TODAS as tags estiverem bem fechadas, no HTML era diferente, muitas vezes deixávamos tags abertas, e ele funcionava que era uma beleza.

Para se fazer um XHTML válido, devemos fechar TODAS as tags:

1. Não devemos esquecer de fechar as tags que estamos carecas de conhecer: , , etc...

2. E não devemos esquecer de forma alguma de fechar as tags “solitárias”, assim, ao invés de <br> escrevemos
</br>, ou na forma simplificada: <br />.

Descobriram que fechando tags desta forma <br/>, não se sabe porque estava causando um problema no Netscape, mas apenas colocando um espaço antes da / o problema é solucionado.

Use letras minúsculas

Quem nunca viu um código fonte de um documento HTML escrito assim: <A href=”http://tags.com.letras.minúsculas/” TARGET=”_BLANK”> </A>
Um documento XHTML deve ter TODAS as tags e seus respectivos atributos escritos com letra minúscula!

Não esqueça das “ASPAS”

Esta regra é bem simples. Todos os atributos XHTML devem conter as benditas “ASPAS”.

Atributo NAME

O antigo atributo NAME foi substituído pelo atributo ID. Se seus usuários, clientes, etc, utilizam ainda antigos browsers, você pode sem problema nenhum utilizar as duas formas juntas durante neste período em que estamos migrando:

Atributos sem valor

Não devemos esquecer também os atributos que escrevemos sem valor, por exemplo:

ERRADO:
<option selected>
<frame noresize>
<input checked>
<input readonly>

CERTO:
<option selected=”selected”>
<frame noresize=”noresize”>
<input checked=”checked”>
<input readonly=”readonly”>

E assim por diante.

Quer uma dica?

Se você está migrando do HTML para o XHTML, o TIDY pode te dar uma forcinha. O TIDY é uma ferramenta para validar e consertar códigos HTML, ele tem opções que você pode escolher qual a versão do HTML você quer validar, uma dessas opções é a XHTML. Se você já está escrevendo um XHTML e quer que seu código fique livre de todos os erros, o TIDY arruma para você.