

CSS 2

Para aplicar estilos a links, pode-se usar qualquer propriedade CSS (color, font-family, background etc). Além disso, os links podem ser estilizados dependendo do estado em que estão. Existem 4 estados

a:link	link ainda não visitado
a:visited	link já visitado
a:hover	quando o mouse passa sobre o link
a:active	link no momento em que é clicado

Veja no exemplo:

```
a:link { color: #FF0000; }
a:visited { color: #00FF00; }
a:hover { color: #FF00FF; }
a:active { color: #0000FF; }
```

Exercício 1

Aproveite os arquivos brasil.html e brasil.css e garanta que o html tenha pelo menos 5 links. Aplique no arquivo brasil.css os links acima descritos com as cores aqua, violet, sienna e chartreuse pela ordem. Acrescente nos 4 o amarelo como cor de fundo (background-color:yellow) Veja o funcionamento disso tudo num browser.

Para aplicar estilos a listas, pode-se usar a propriedade list-style-type: que pode ser circle, square, upper-roman, lower-alpha. Pode-se usar também list-style-image que indica qual imagem usar como o marcador de itens em listas não ordenadas. Eis o formato:

```
ul { list-style-image: url('spurple.gif'); }
```

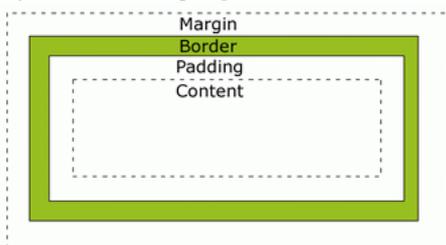
Para as tabelas, as propriedades são: border:, border-collapse:, width:, height:, text-align:, vertical-align:, padding:, color: e background-color:. Veja um exemplo de algumas dessas opções:

```
<style>
table, td, th { border: 1px solid green; }
th { background-color: green; color: white; }
</style>
...
<table> <tr> <th>Nome</th><th>Cidade</th>
<th>Contribuição</th> </tr>
<tr> <td>Pedro</td> <td>Curitiba</td>
<td>100</td> </tr>
<tr> <td>Maria</td> <td>Joinville</td>
<td>300</td> </tr> </table>
```

Exercício 2

Defina arquivos tabela.html e tabela.css e escreva o exemplo acima. Sobre ele, aplique pelo menos 3 propriedades de tabelas diferentes e veja em cada caso o resultado em um browser.

Agora vamos estudar o modelo caixa do CSS (CSS box model). Todos os elementos HTML podem ser considerados como uma caixa. Este conceito é muito importante quando se pensa em layout. Pense no modelo da caixa como uma cebola quadrada composta de 4 camadas. De fora para dentro tem-se: margin, border, padding e finalmente o conteúdo que é o elemento HTML. Veja o desenho a seguir para entender o conceito



Eis um exemplo disso tudo junto

```
div { width: 300px; padding: 25px;
border: 25px solid navy; margin: 25px; }
```

Note que quando você define largura e altura de alguma coisa, essas medidas se referem à área de conteúdo. Para saber o tamanho total do elementos as outras medidas (se existentes) devem ser somadas. Por exemplo, o elemento abaixo ocupa 350px:

```
div { width: 320px; padding: 10px;
border: 5px solid gray; margin: 0; }
```

As propriedades da borda são: border-style: com os valores none, dotted, dashed, solid, double, groove, ridge, inset e outset. Todas as demais propriedades só atuam se a propriedade border-style for assinalada.

Depois vem border-width:, border-color: etc. Para definir a borda de maneira abreviada, a sequência é border-width:, border-style: (obrigatória), border-color:.

A seguir existem diversas propriedades para tratar o outline, elas são muito parecidas com as propriedades da borda e ficam como dever de casa para serem pesquisadas.

A margem clareia completamente o entorno de um elemento (fora da borda). A margem não tem cor de background e sempre é completamente transparente. As margens superior, direita, inferior e esquerda podem ser estabelecidas separadamente, ou pode-se usar a maneira abreviada e fazer tudo junto. Os valores possíveis de margin: são: auto (o browser calcula), tamanho (medida em px, pt, cm, ... - default é 0px), % medida em porcentagem do tamanho do elemento contido e inerit quando a margem é herdada do elemento pai. Note que em tamanho pode-se usar valores negativos para sobrepujar algum conteúdo. As margens podem ser especificadas como margin-top:, margin-bottom:, margin-right: e margin-left: ou se preferir escrever apenas margin: e depois 4, 3 2 ou 1 valor de tamanho. A regra é: 3 tamanhos → top, laterais, bottom; 2 tamanhos → cima-baixo, laterais, e 1 tamanho → os 4 iguais.

O padding limpa o entorno do elemento (dentro da borda). Ele é afetado pela cor de fundo do elemento. De maneira muito parecida à margem, ele pode ser estabelecido com 4 propriedades ou de maneira abreviada com a única propriedade.

Os elementos HTML também podem ter suas dimensões estabelecidas pelas propriedades: height:, max-height:, min-height:, width:, max-width: e min-width:. Veja um exemplo:

```
p { max-width: 20%;
background-color: yellow; }
```

A propriedade display: determina se e como um elemento é mostrado e a propriedade visibility especifica se um elemento deve ser visível ou escondido. Um elemento pode ser apagado fazendo display: none ou visibility:hidden. Este último apaga o elemento mas preserva o espaço ocupado por ele (em branco). Ou seja o layout permanece inalterado. Já display: none apaga o elemento e consome com o espaço ocupado por ele. O layout é refeito.

Exercício 3

Escreva um html chamado image.html e um css chamado image.css. Fale sobre o litoral paranaense. Apresente 3 fotos de mesma dimensão. Use 3 classes diferentes. No CSS alterne as propriedades vistas acima. Veja o resultado no browser.

Relembrando elementos em bloco (como <h1>, <p>, e <div> por exemplo) são aqueles que forçam quebra de linha antes e depois dele. Já os elementos inline não quebram linha nem antes nem depois. Para transformar um elemento inline em bloco e vice-versa, usa-se a propriedade display:. Para inline o valor é inline e para bloco o valor é block. Veja no exemplo como transformar itens de lista em elementos inline (todos os itens serão mostrados na mesma linha).

```
li { display: inline; }
```

As propriedades de posicionamento permitem posicionar um elemento. Também permitem por um na frente do outro e especificam o que acontece quando o conteúdo de um elemento é muito grande. A primeira propriedade é position:. Ela tem 4 valores:

static é o valor default. O elemento é posicionado de acordo com o fluxo normal de preenchimento do browser. O elemento não é afetado pelas propriedades top, bottom, left e right.

fixed posição fixa em relação à janela do browser. Ele não é movido mesmo que a janela seja *scrolled*. Veja um exemplo

```
p.fixo { position: fixed;
top: 30px; right: 5px; }
```

Neste caso o elemento é removido do fluxo normal. Os outros elementos são posicionados como se o elemento fixo não existisse.

relative modifica a posição em relação à sua posição normal. Aqui por exemplo, os cabeçalhos h2, serão trazidos para fora da janela:

```
h2 { position: relative; left: -20px; }
```

absolute posiciona relativo ao primeiro elemento pai que tenha outra posição que não static. Se não existir, será o <html>.

Quando elementos são posicionados fora do fluxo normal eles podem sobrepujar outros elementos. A propriedade z-index: indica qual a ordem de apresentação. (podem ser atribuídos valores positivos ou negativos). A regra é que um número maior sempre vai aparecer na frente de um número menor. Veja o exemplo:

```
<style>
img { position: absolute; left: 0px;
top: 0px; z-index: -2; }
</style> </head> <body>
<h1>Um cabeçalho</h1>

</body>
```

Exercício 4

Copie o exemplo acima, usando uma figura qualquer. Varie a propriedade z-index e veja o que acontece.

Para você fazer

1. Imprima os arquivos pedidos (brasil.html e css, tabela.html e css e image.html e css) grameie-os nesta folha e devolva tudo ou então gere-os no micro na sala de aula e mostre-os ao professor, durante a aula.

2. Descreva o que faz o código a seguir

```
<!DOCTYPE html> <html> <head> <style>
table, td, th { border: 1px solid green; }
th { background-color: green; color: white; }
</style> </head> <body> <table> <tr>
<th>Firstname</th> <th>Lastname</th> <th>Savings</th> </tr>
<tr><td>Peter</td><td>Griffin</td><td>$100</td></tr>
<tr><td>Lois</td><td>Griffin</td><td>$150</td></tr>
<tr><td>Cleveland</td><td>Brown</td><td>$250</td></tr>
</table> </body> </html>
```

3. Descreva o que faz o código a seguir

```
<!DOCTYPE html> <html> <head> <style>
a.two:link {color:#ff0000;}
a.two:visited {color:#0000ff;}
a.two:hover {font-size:150%;}
</style> </head> <body>
<p>Mouse over the links and watch them change layout:</p>
<p><b>a class="two" href="default.asp"
target="blank">This link changes font-size</a></b></p>
</body> </html>
```

