

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA BA – IFBA

PROGRAMAÇÃO WEB - APOSTILA CSS
PROFESSOR: ADDSON ARAÚJO DA COSTA

VALENÇA - BA

Sumário

1. INTRODUÇÃO.....	4
1.1 DIV.....	4
1.2 CSS.....	4
1.2.1 Seletor.....	4
1.2.2 Regras CSS.....	5
1.2.3 Comentários.....	5
2. Identificadores e Classes.....	5
2.1 ID.....	6
2.2 Classe.....	6
3. CSS.....	7
3.1 Seletores.....	7
→ Seletor Universal.....	7
→ Seletor por Tag.....	8
→ Seletor por Classe.....	8
→ Seletor por ID.....	8
3.2 Cores e fundo.....	8
3.2.1 Color.....	9
→ Background-color.....	10
→ Background-image.....	10
→ Background-repeat.....	10
→ Background-attachment.....	10
3.3 Fontes.....	11
→ Font-family.....	11
→ Font-style.....	11
→ Font-weight.....	11
→ Font-size.....	11
3.4 Tamanhos.....	11
3.5 Borda.....	12
3.6 Modelo de caixa.....	12
→ Margin.....	12
→ Detalhes sobre Margin e padding.....	13
4. Google Fonts.....	13

Lista de Figuras

Figura 1: Visualização Exemplo 1.....	4
Figura 2: Visualização do Exemplo 2.....	5
Figura 3: Visualização Exemplo 3.....	6
Figura 4: Paleta de cor.....	10
Figura 5: Estilos de borda.....	12
Figura 6: Modelo de Caixa.....	12

Lista de Tabelas

Tabela 1: Primeiros Seletores.....	7
Tabela 2: Cores e fundo.....	9
Tabela 3: Fontes.....	11
Tabela 4: Tamanhos.....	12
Tabela 5: Bordas.....	12
Tabela 6: Margens.....	13
Tabela 7: Margem, Espaçamentos e Bordas.....	13

1. INTRODUÇÃO

Antes de iniciarmos o assunto CSS, é importante observar que em HTML não estudamos tags para modificar cor de texto, cor de fundo, tamanhos de elementos, entre outras características estéticas. Antigamente eram usadas tags para este fim, mas elas foram abolidas do HTML pois não é prático a manutenção de códigos usando estas tags. Nesse ponto, entra o CSS, mas antes vamos estudar mais uma tag do html, a DIV. Os exemplos nessa apostila não contêm todos os elementos obrigatórios do HTML 5 por não ser o foco aqui e, dessa forma, os exemplos ficam mais simples e didáticos por conter menos código.

1.1 DIV

A tag “DIV” é responsável por criar uma caixa e por padrão não tem cores, tamanhos nem bordas, é uma caixa invisível usada para agrupar elementos e facilitar posicionamento destes elementos na página, uma vez que ao modificar a posição da DIV, todos os elementos internos a acompanham por estarem dentro dessa caixa invisível. Veja o Exemplo 1 e note também que, por padrão, as divs ficam uma abaixo da outra, seguindo a idéia de que caixas ficam empilhadas.

1.2 CSS

Para utilizar o CSS, várias formas podem ser válidas, mas começaremos com um método exemplificado abaixo. Nesse método, é incluso uma tag chamada *style* dentro da tag HEAD. É importante dizer que dentro da tag *style* não pode ser incluso nenhum código HTML. Ao invés disso, deve conter apenas código CSS. Note a diferença no Exemplo 1 entre o padrão usado na escrita dentro da tag *style* e o resto do código.

```
<html>
  <head>
    <style>
      div {
        color: orange;           /* cor da fonte laranja */
        background-color: teal; /* fundo verde azulado */
        margin: 20px;           /* espaço entre divs de 20px */
        width: 200px;           /* largura da div de 200px */
        height: 200px;          /* altura da div de 200px */
        text-align: center;      /* texto centralizado */
      }
    </style>
  </head>
  <body>
    <div> Caixa 1 </div>
    <div> Caixa 2 </div>
  </body>
</html>
```

Exemplo 1: Utilização do CSS

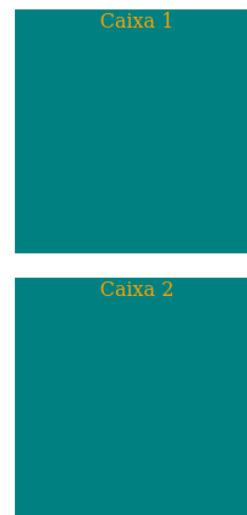


Figura 1: Visualização Exemplo 1

1.2.1 Seletor

Agora que temos um exemplo de como utilizar o CSS, vamos entender o que foi declarado dentro da tag *style*. Primeiro estudaremos a parte destacada em vermelho no Exemplo 1, o seletor.

Temos a palavra DIV seguida de chaves. Chamaremos todo o código que estiver imediatamente antes das chaves por “seletor”, isto é, o critério utilizado para selecionar quais elementos devem ser modificados por este código CSS incluso dentro das chaves.

No exemplo acima foi utilizado um seletor por tag, então não importa quantas divs (caixas) existam no código fonte, todas serão afetadas, uma vez que foi colocado uma seleção pela tag DIV.

1.2.2 Regras CSS

Dentro das chaves temos um “conjunto de regras”, destacadas em azul no Exemplo 1, que devem ser separadas por ponto e vírgula. No exemplo acima foi utilizado algumas regras dentro das chaves. As regras usadas foram explicadas nos comentários, destacados em verde no Exemplo 1. Essas regras serão detalhadas a frente, mas note o padrão de escrita.

Cada regra é colocada de forma semelhante a forma de definir atributos no HTML, mas note que ao invés do sinal de igualdade, é usado o sinal de dois pontos para separar o nome da regra e seu valor. Além disso, não é usado aspas no valor. A semelhança existe, mas o correto é chamar de regra CSS ao invés de atributo do CSS, uma vez que atributos são de objetos, mas aqui não existe a idéia de objetos.

1.2.3 Comentários

Note que o padrão de comentário de código em CSS, destacados no Exemplo 1 em verde, é diferente da forma usada em HTML. Aqui é usado “/*” para iniciar um comentário e “*/” para terminar o comentário que pode abranger uma ou várias linhas.

2. Identificadores e Classes

No HTML, em todas as tags, podemos usar os atributos ID e CLASS, mas eles têm objetivos bem distintos. Quando nosso objetivo é identificar um único elemento, devemos usar o atributo ID que deve ter um identificador único (daí o nome do atributo ID) para este elemento, já quando vários elementos deverão receber uma mesma aparência, devemos usar o atributo CLASS (uma classe é um conjunto de elementos, daí o nome do atributo). Veja o Exemplo 2, abaixo.

```
<html>
  <head>
    <style>
      #cor-verde {
        color: green; /* cor da fonte verde */
      }
      .cor-vermelho {
        color: red; /* cor da fonte vermelha */
      }
    </style>
  </head>
  <body>
    <p id='cor-verde'> Texto vai aqui </p>
    <p class='cor-vermelho'> Texto vai aqui </p>
    <p class='cor-vermelho'> Texto vai aqui </p>
    <p class='cor-vermelho'> Texto vai aqui </p>
  </body>
</html>
```

Texto vai aqui

Texto vai aqui

Texto vai aqui

Texto vai aqui

Exemplo 2: IDs e Classes

Figura 2: Visualização do Exemplo 2

Note nos parágrafos dentro do BODY que o uso do atributo ID chamado de “cor-verde” foi usado apenas uma vez pois o atributo ID não deve repetir com mesmo valor, já nos parágrafos seguintes foi usado o atributo CLASS que pode e deve se repetir.

Para que o CSS consiga selecionar os elementos pelo ID, note que foi iniciado um seletor chamado “#cor-verde”, onde a hashtag indica ao CSS que estamos buscando por ID, já quando desejamos selecionar elementos por classe, foi usado “.cor-vermelho”, onde o ponto no início significa que o CSS buscará elementos pela classe.

2.1 ID

Ao construir um site é comum que alguns itens sejam bem específicos, por exemplo, o tamanho da fonte para o logotipo, a largura ou altura do menu, entre outras características. Para esses usos pontuais, é interessante uso do atributo ID.

Note que no atributo ID não podemos definir mais de um elemento com mesmo ID pois ele representa um identificador, isto é, uma informação única, mas se não respeitarmos isto, o navegador continuará exibindo tudo corretamente pois o navegador não deseja exibir nada feio, então ele corrigirá seus erros, assim como o professor, porém, no segundo caso, retirando pontos =) uma vez que não estará validado em HTML 5.

Uma vez que o ID não pode se repetir no código, não seria muito inteligente usá-lo como seletor quando desejamos que vários elementos tenham uma mesma aparência, pois isto geraria muito código repetido. Para isso, é interessante fazer uso da classe, assim como nos demais parágrafos definidos no Exemplo 2.

2.2 Classe

As classes, ao contrário dos IDs, são atributos que podem e devem se repetir. Elas permitem agrupar elementos que devem ter mesma aparência, veja no Exemplo 2, acima, quando foi usado vários parágrafos com mesma classe “cor-vermelho”.

```
<html>
  <head>
    <style>
      /* Aqui foi definido que todos os elementos com classe “azul” */
      .azul {
        color: blue; /* Terão a cor de fonte azul */
      }
      /* Aqui foi definido que todos os elementos com classe “grande” */
      .grande {
        font-size: 30px; /* O tamanho da fonte será 30 pixels */
      }
      /* Aqui foi definido que todos os elementos com ID “elemento1” */
      #elemento1 {
        font-style: italic; /* Texto em itálico */
        font-weight: bolder; /* Texto em negrito */
      }
    </style>
  </head>
  <body>
    <!-- O significado da aparência é definido no CSS -->
    <div id="elemento1" class="azul grande"> div 1 </div>
    <div class="azul"> div 2 </div>
    <div class="grande"> div 3 </div>
    <div class="azul grande"> div 4 </div>
  </body>
</html>
```

div 1
div 2
div 3
div 4

Figura 3: Visualização Exemplo 3

Um detalhe é que um elemento HTML pode pertencer a várias classes ao mesmo tempo, o que não é possível com uso de IDs. É uma forma de dizer que ele contém a aparência combinada de várias classes de elementos. Veja o código a seguir no Exemplo 3, acima.

Note no exemplo acima que a primeira DIV ficou em negrito (letra mais espessa) e itálico (inclinação mais fácil de ver na parte reta da letra “d”). Isto porquê ela tem o ID “elemento1”, já nas demais ocorre alteração apenas de cor na segunda div, tamanho na terceira div e ambos na quarta div de acordo com as classes que foram usadas combinadas.

Uma observação importante aqui é que apesar do elemento poder pertencer a várias classes, não pode ser definido mais de uma vez o atributo class, isto é, uma vez definido, dentro deve conter o nome de todas as classes que o elemento pertence separados por espaço em branco, ao contrário de dois atributos class com valores diferentes. Isto vale para qualquer outro atributo do HTML.

3. CSS

A Sigla CSS refere-se a Cascading Style Sheets (do inglês, folhas de estilo em cascata) e é responsável por definir o estilo dos elementos HTML usados no site, isto é, definir a estética do site. Mais para frente entenderemos o porquê de se chamar folhas de estilo em cascata, mas note que ele está ligado unicamente a estilo, isto é, aparência.

Uma curiosidade é que nomes de classes e IDs não podem começar com número, ter acentos e/ou pontuação e, uma vez que, é uma boa prática deixar todo o código em minúsculo, note o uso de hífen no Exemplo 2, acima, para separar palavras diferentes em IDs e classes, contudo isto não é obrigatório, sendo apenas uma boa prática para facilitar a leitura do código. Esta prática facilitará também a compreensão no padrão usado em frameworks, estudados mais a frente.

3.1 Seletores

Seletores são usados como critério para selecionar elementos HTML. Existem várias opções e eles podem ser combinados de diversas formas para atender qualquer situação. Vamos esquecer por enquanto as formas de combiná-los e mostrar algumas opções.

Nome	Exemplo CSS	Significado
Seletor universal	<code>* {</code> <code>}</code>	Seleciona todos os elementos
Seletor por tag	<code>b {</code> <code>}</code>	Seleciona todos que têm a tag descrita antes das chaves.
Seletor por classe	<code>.azul{</code> <code>}</code>	Seleciona todos os elementos com a classe após o ponto.
Seletor por ID	<code>#elemento1 {</code> <code>}</code>	Seleciona o elemento que tem o ID descrito após o hashtag.

Tabela 1: Primeiros Seletores

O seletor por TAG pode ser visto no Exemplo 1, já os seletores por classe e ID, podem ser vistos no Exemplo 2 e Exemplo 3 que, ao contrário do Exemplo 1, o seletor usado não inicia com uma letra, ao invés disso, inicia com um ponto (“.”) ou hashtag (“#”). Além destes, foi introduzido mais um tipo de seletor, o asterisco.

→ Seletor Universal

Ele é responsável por selecionar todos os elementos do site de uma só vez. Isto é muito útil, por exemplo, para quando se quer definir a fonte de todo o site. Dessa forma, de uma só vez, se define algo para todo o projeto. Para isso, basta colocar um asterisco no lugar do seletor.

→ Seletor por Tag

Com ele pode-se selecionar qualquer elemento através de sua Tag, isto é muito útil por permitir ajustes como, por exemplo, para que todos os parágrafos tenham o recuo na primeira linha, para alterar a aparência de todas as âncoras (Tag A usada para indicar links) para que não fiquem com suas aparências padrão, estilizar todos os botões, etc.

Para usar o seletor por tag, basta digitar o nome da tag desejada no lugar do seletor, por exemplo, body, b, i, table, etc.

→ Seletor por Classe

Ao iniciar o seletor com o caracter de ponto final (“.”) estamos informando que desejamos buscar elementos que contenham aquela classe. Note que um elemento pode conter várias classes como observado no Exemplo 3.

Assim, a idéia é definir classes mais genéricas e combiná-las ao invés de definir uma classe específica para um elemento. Coisas específicas devem ser feitas via ID. Generalize criando muitas classes, mas com pouco código ou até uma única regra CSS em cada uma, tornando-as mais fácil de combinar.

→ Seletor por ID

Por último, quando iniciamos o seletor com hashtag (“#”), estamos informando que desejamos buscar um elemento com base em seu ID. Isto é útil no caso contrário ao seletor por classes. Aqui é usado quando desejamos definir algo específico que só existe para aquele elemento.

Nesse caso, se atribui um ID ao elemento, definimos as regras CSS específicas e jogamos as características gerais para classes. Note no Exemplo 3 que um elemento pode, ao mesmo tempo, ter um ID e várias classes.

3.2 Cores e fundo

Antes de entrarmos nas regras CSS que permitem definir cores e fundos, vamos entender como o computador faz para definir as cores na tela. É preciso saber que dentro do olho humano existem conjuntos de células (cones) capazes de reconhecer apenas três cores. São elas o vermelho, verde e azul que em inglês são, respectivamente, *Red*, *Green* e *Blue*, cuja sigla é *RGB*.

Caso estejamos usando o modelo de definição de cores RGB, estas serão chamadas de cores primárias e a mistura destas cores em igual proporção irá gerar as cores secundárias que são o ciano, magenta e amarelo. Nesta apostila será usado o modelo RGB como padrão de cores primárias, mas existe outro padrão de cores primárias que é o CMYK (Ciano, Magenta, Amarelo e Preto).

A partir de combinações destas 3 cores do RGB em diferentes intensidades, são formadas todas as cores visíveis ao olho humano, assim, em informática é usado a combinação destas cores como forma de definição de cor, ao contrário de outras possibilidades como, por exemplo, nos meios artísticos muitas vezes se usa como cores primárias a união de vermelho, azul e amarelo.

Em impressoras usa-se a união entre Ciano, Magenta, Amarelo e Preto (CMYK) ao invés do modelo RGB porque no RGB a união de todas as cores gera a cor branca, porém usa-se papel branco, então na impressão não faz sentido gastar todas as cores de tinta para obter o branco que já está no papel. Já no modelo CMYK, o preto não precisa ser gerado pela combinação de cores e é a cor mais usada em impressoras.

Abaixo temos uma tabela com as regras CSS possíveis para definir cores, detalhada nos tópicos seguintes.

Regra CSS	Significado	Valores possíveis
color	Cor do texto	Nome da cor em inglês rgb(255,0,0) rgba(255,0,0,0.5) #FF0000
background-color	Cor de background	Nome da cor em inglês rgb(255,0,0) rgba(255,0,0,0.5) #FF0000 transparent
background-image	Imagem de background	url("caminho/img.jpg")
background-repeat	Repetição do background	repeat repeat-x repeat-y no-repeat
background-attachment	Anexação de background	scroll fixed

Tabela 2: Cores e fundo

3.2.1 Color

A primeira regra da Tabela 2 se chama color, ela é responsável pela cor da fonte e pode ser definida de quatro formas diferentes, explicadas abaixo.

→ Nome da cor

Entre os valores possíveis, note que a primeira opção é colocar o nome da cor em inglês, para muitas cores isto funcionará, por exemplo, black, blue, red, purple, white, etc. Nos Exemplos 1, 2 e 3 foi usado esta forma para definir cores de texto por questões didáticas, embora esta seja uma opção muito limitada. Por exemplo, a regra "color: red;" significa cor do texto vermelho.

→ RGB

A segunda opção é colocar a palavra "rgb" seguido de parênteses com três números entre vírgulas, o primeiro valor refere-se a quantidade de vermelho, o segundo refere-se a quantidade de verde e o último, a quantidade de azul pois esta é a ordem na sigla RGB. É importante ressaltar que o valor para cada cor a ser usado deve ser um número inteiro entre 0 e 255, não sendo aceito outros valores. Por exemplo, a regra "color: rgb(255,0,0);" significa cor do texto em vermelho em sua cor mais forte.

→ RGBA

A terceira opção se chama RGBA e é idêntico ao modelo RGB, porém adiciona-se mais um valor que deve ser informado por último. Este último valor refere-se ao canal *Alpha* e é quem define a opacidade da cor caso não se deseje uma cor opaca como, por exemplo, deseja-se atribuir cor a um elemento, mas continuar mostrando um efeito de fundo que existe na página. Para o canal Alpha deve-se informar um valor entre 0 e 1. Para metade da opacidade, por exemplo, deve ser atribuído o valor "0.5". Note que usamos o ponto, e não a vírgula, nesse caso. Por exemplo, a regra "color: rgb(255,0,0,0.5)" significa vermelho em sua cor mais forte, porém com metade de transparência.

→ Hexadecimal

Antes de entendermos a quarta opção para definir cores, é preciso entender a diferença entre o sistema decimal e hexadecimal. No sistema decimal, cada dígito assume os valores de 0 a 9, o que totaliza dez opções de números em cada dígito dando a esse sistema o nome de decimal. Já no

sistema hexadecimal cada dígito pode assumir valores de 0 a 9 e também as letras de A a F que significam, respectivamente, 10 a 15.

Essa opção é ainda uma definição com base em RGB, mas a forma de escrever é ligeiramente diferente. O símbolo de hashtag aqui para definir cores significa que os valores para o RGB serão informados usando números hexadecimais. Nesse caso, os primeiros dois dígitos significam a quantidade de vermelho, os dois dígitos do meio significam a quantidade de verde e os últimos significam a quantidade de azul.

Assim, o número 255 em decimal, quando convertido para hexadecimal equivale a FF (confere!), então note que a cor #ff05f4 significa que terá “ff” de vermelho (que em decimal equivale a 255), “05” de verde (em decimal equivale a 5) e “f4” de azul (que em decimal equivale a 244). Ou seja, as regras “color: #ff05f4” e “color: rgb(255, 5, 244)”, são equivalentes usando formas diferentes de escrita.

A preferência entre os programadores costuma ser pelo sistema hexadecimal uma vez que é mais fácil achar paletas de cor usando esse sistema e requer menos texto a ser escrito. Busque no google por paletas de cores (conjunto de cores que combinam entre si). Será comum encontrar imagens como na Figura 4, ao lado.



Figura 4: Paleta de cor

→ Background-color

Para cor de fundo, pode-se usar o mesmo modo de valores que o usado para definir cor de fonte, porém pode-se utilizar também a palavra “transparent” que é a cor de fundo padrão de todos os elementos e significa transparente (obviamente). Por exemplo, a regra “background-color: transparent;” significa que a cor de fundo será transparente, mas poderia ser usado a definição em nome de cor, rgb ou rgba.

→ Background-image

Esta regra permite usar uma imagem de fundo e para isto deve-se especificar um caminho onde será encontrada a imagem. Este caminho pode ser um endereço na internet e nesse caso o endereço deve ser completo especificando a partir do “http://” ou “https://” que é o protocolo usado para obter a imagem no servidor. Caso a imagem esteja copiada na mesma pasta em que encontra-se seu código HTML, basta informar o nome da imagem seguido de sua extensão, assim como usamos na tag IMG do HTML.

A forma de indicar uma imagem de fundo, contudo, não é apenas colocar o nome do arquivo. Devemos usar, por exemplo, background-image: url(“caminho/img.jpg”). Note que o caminho para o nome do arquivo fica entre aspas e dentro de parênteses após a palavra url.

→ Background-repeat

Esta regra é útil quando se quer colocar uma imagem de fundo e permitir que ela se repita na tela. Uma possível aplicação é ao colocar uma textura como imagem de fundo.

Com esta regra, pode-se informar quatro opções de valores. O primeiro, *repeat*, permite que a imagem de fundo se repita horizontal e verticalmente. O segundo, *repeat-x*, permite repetições apenas no eixo horizontal. O terceiro, *repeat-y*, permite repetições apenas no eixo vertical. O último, *no-repeat*, não permite repetições. Por exemplo, background-repeat: *no-repeat*.

→ Background-attachment

Esta regra é responsável por definir como se dá a posição do background quando a página sofre o efeito de *scroll*, ou seja, o efeito de descer a tela pois existe mais conteúdo do que o que cabe sendo exibido.

Ao utilizar essa regra com o valor “*fixed*”, o *background* não rolará na tela mesmo que o conteúdo seja rotacionado pelo *scroll*. Já usando o valor “*scroll*” informará que o fundo deve também ser rotacionado junto ao conteúdo.

3.3 Fontes

Para fontes, temos um conjunto de regras CSS que podem ser vistas na Tabela 3, abaixo.

Regra CSS	Significado	Valores possíveis
font-family	Família(s) de fontes	Nome da fonte
font-style	Estilo utilizado na fonte	normal italic
font-weight	Peso da fonte	normal bold bolder lighter
font-size	Tamanho da fonte	Valor do tamanho

Tabela 3: Fontes

→ Font-family

Com a regra *font-family*, se define qual o nome da fonte usar. Apesar de ser semelhante ao que encontramos em editores de texto quando fazemos um relatório ou conjunto de slides, em web não podemos usar nomes de fontes disponíveis em nosso computador, uma vez que esse código será executado por qualquer computador na internet, assim, quem vai visualizar nosso site pode não ter a fonte que desejamos instalada em sua máquina.

Mais para frente, será mostrado como incluir uma fonte em nosso site de forma a ela estar disponível em qualquer computador que abrir nosso site. Por enquanto, usaremos uma entre as cinco fontes genéricas que existem em todos os browsers. São elas: serif, sans-serif, monospace, cursive e fantasy. Por exemplo, font-family: serif.

→ Font-style

Na regra font-style podemos definir se queremos a fonte normal ou em itálico. Note que para definir negrito é outra regra CSS, isto porque a fonte pode estar em itálico e negrito ao mesmo tempo. Por exemplo, font-style: italic.

→ Font-weight

Com a regra font-weight podemos definir se queremos a fonte normal, bold, bolder ou lighter que significam, respectivamente, normal, negrito, negrito escuro e mais claro que o normal, respectivamente. Por exemplo, font-weight: bolder.

→ Font-size

Esta regra serve para definir o tamanho da fonte. Esse valor é normalmente em pixels e representa a altura da fonte do seu ponto mais baixo até a sua parte mais alta. Recebe um valor numérico. Por exemplo, font-size: 12px.

3.4 Tamanhos

Todos os elementos podem ter seus tamanhos definidos. É possível definir também os tamanhos mínimo e máximo de cada elemento, mas deixaremos isso mais para frente.

Regra CSS	Significado	Valores possíveis
width	Largura	Valor numérico
height	Altura	

Tabela 4: Tamanhos

3.5 Borda

A DIV como vimos, por padrão, não tem cor, tamanho ou qualquer outra característica estética pré-definida. Já aprendemos a definir cores, tamanhos e estilos de fonte. Porém podemos desejar colocar bordas ao redor de elementos.

Regra CSS	Significado	Valores possíveis
border-width	Largura da borda	valor numérico
border-color	Cor da borda	uma cor (rgb, rgba, hexadecimal, ...)
border-style	Estilo da borda	none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset

Tabela 5: Bordas

Para borda, podemos definir sua largura, cor e estilo. Os estilos podem melhor ser entendidos observando a Figura 5, abaixo.

Para as definições de borda foram exemplificados na Tabela 6 apenas três regras, mas a realidade é que existem algumas muitas que encontram-se listadas abaixo. Note que são variações destas três, porém para aplicar a borda apenas de algum dos lados e seu uso é idêntico as regras acima. São elas:

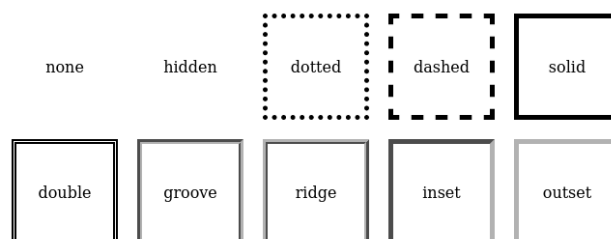


Figura 5: Estilos de borda

- border-left-width, border-right-width, border-top-width, border-bottom-width
- border-left-color, border-right-color, border-top-color, border-bottom-color
- border-left-style, border-right-style, border-top-style, border-bottom-style

3.6 Modelo de caixa

Para todos os elementos no HTML, podemos imaginá-los como em um modelo de várias caixas uma dentro da outra como na Figura 5, ao lado.

→ **Margin**

A região em amarelo, chamada de margem, é uma área invisível que separa a borda do elemento de outros elementos ao seu redor. Imagine, por exemplo, que você criou a borda em uma caixa (DIV), mas não gostaria que essa borda encoste em outros elementos. Para isso, você deverá usar as regras referentes a margin em CSS.

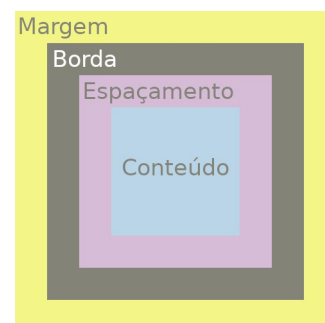


Figura 6: Modelo de Caixa

Regra CSS	Significado	Valores possíveis
margin-left	Margem esquerda	valor numérico
margin-right	Margem direita	
margin-top	Margem acima	
margin-bottom	Margem abaixo	
margin	Margem em todas as direções	

Tabela 6: Margens

→ **Padding**

Quando definimos uma cor de fundo ou imagem de fundo para um elemento, toda a área dentro da borda será afetada, ou seja, as áreas em rosa e azul na Figura 4. A diferença é que a área em rosa serve como um espaçador para que o conteúdo não toque na borda. Este é o padding (espaçamento).

Regra CSS	Significado	Valores possíveis
padding-left	Espaçamento esquerdo	Valor numérico
padding-right	Espaçamento direito	
padding-top	Espaçamento acima	
padding-bottom	Espaçamento abaixo	
padding	Espaçamento em todas as direções	

Tabela 7: Margem, Espaçamentos e Bordas

→ **Detalhes sobre Margin e padding**

Uma curiosidade é que ao usar a regra margin ou padding informando apenas um valor numérico, este valor é aplicado em todas as direções, como de se esperar.

Porém ao fornecer dois valores separados por um espaço em branco, o primeiro valor é aplicado para superior e inferior, enquanto o segundo valor é aplicado para esquerda e para direita.

Já se fornecermos quatro valores para margem ou espaçamento, o primeiro valor é aplicado para cima, o segundo para a direita, o terceiro para baixo e o quarto valor para a esquerda. Uma dica para memorizar é que começa de cima e depois vai em sentido horário.

4. Google Fonts

Acima, no tópico sobre a regra *font-family* foi dito que para usar uma fonte, ela precisa estar disponível no computador de todos os usuários que abrirem nosso site. Não adiantando ter a fonte em nosso próprio computador uma vez que o código será executado em outra máquina. Bom, chamamos os computadores que forem executar nosso site de computadores cliente, mas precisamos garantir que a fonte esteja disponível nele.

Para isso, precisamos linkar um arquivo de fonte disponível online, usando um link seguro, rápido e confiável. Assim, caso o computador do cliente não tenha a fonte que queremos, ele terá um link onde pode encontrá-la e fará todo o processo automaticamente.

Nesse sentido, surgiu um projeto da Google chamado Google Fonts <fonts.google.com>, onde pode-se escolher a fonte que se deseja vendo já sua aparência. Dentre as fontes, pode-se escolher os estilos entre os disponíveis para cada fonte. Os estilos são a espessura do traço e está diretamente relacionado ao peso da fonte que se deseja usar, isto é, intensidade para o que seria normal, bold (negrito), bolder (negrito escuro), etc. Os estilos de fonte são específicos para a espessura de traço e também para o estilo normal ou itálico.

Após selecionar as fontes desejadas e seus respectivos estilos, aparece um carrinho onde o serviço do google fornece uma tag do tipo LINK que deve ser copiada e colada na tag HEAD de seu

site. Ainda no Google Fonts ele mostra um exemplo de como usar a fonte, fornecendo a regra font family. O serviço é gratuito e pensado para ser utilizado por desenvolvedores.