

# Aula 5

## Desenvolvimento CSS

---

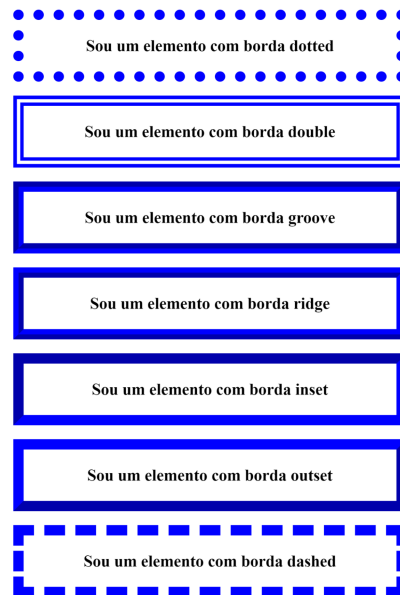
### Bordas, arredondamento, gradiente e botões

#### Bordas

O CSS, nos permite criar “box’s(caixas)” de texto. Porém com a grande capacidade estética, ainda nos permite escolher uma opção estética diferente da padrão, para a mesma.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <meta charset="utf-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1">
6    <title>Bordas </title>
7    <style type="text/css">
8
9      p.dashed {border-style: dashed;}
10
11    </style>
12  </head>
13  <body>
14
15
16
17    <p class="dashed">Exemplo de borda.</p>
18  </body>
19  </html>
```

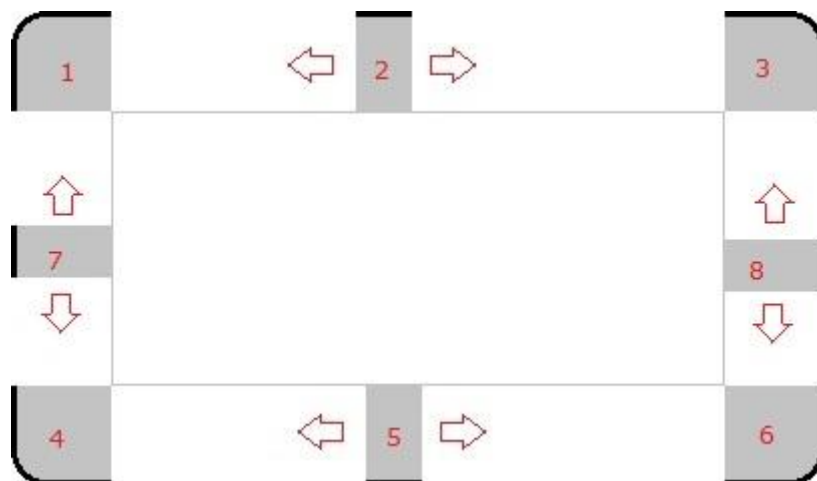
*Código com a implementação de borda estilo “dashed”*



*Imagem de exemplo de demais tipos de bordas*

## Arredondamento

O CSS conta com um recurso estético que possibilita a transformação de “box’s(caixas)” do HTML, que são totalmente retangulares ou quadradas, em uma forma geométrica com bordas arredondadas. A estética tem a possibilidade de ficar muito mais harmoniosa para o leitor da página.



*Teoria das bordas de uma “box” do HTML*

Para que possamos realizar a utilização de bordas do CSS em um projeto, é necessário que utilizemos alguns atributos para definir o modo como será definida a criação do nosso elemento.

O código a seguir, realiza a criação de uma “Box” com bordas arredondadas. Para que ela seja criada, é necessário definirmos alguns elementos que realizam a criação com base nas medidas informadas.

```
1  <html>
2  <head>
3  <title>Bordas</title>
4  <style type="text/css">
5  #div1{
6      width:100px;
7      height:100px;
8      border:solid 1px;
9      border-radius:20px;
10 }
11 </style>
12 </head>
13 <body>
14 <div id="div1"/>
15 </body>
16 </html>
```

*Exemplo de criação de “Box” com bordas em CSS*

- **width:**
  - Define a largura da “caixa” utilizada na nossa página web, utilizando a medida em pixels.
- **height:**
  - Define a altura da “caixa” utilizada em nossa página web, utilizando a medida em pixels.
- **border:**
  - Define a espessura da borda da “caixa” utilizada em nossa página web utilizando a medida em pixels.
- **border-radius:**
  - Define a medida do raio(trigonométrico) para realizar a criação da borda.

## Gradiente

As cores do CSS, são responsáveis por alterar a estética de outro elemento. Para definirmos elas, há possibilidade de utilização de mais de um tipo de valor diferente para definirmos, podendo ser RGB(“red, green, blue”), hexadecimal ou hsl(“hue, saturation, lightness”).

Tendo isto em vista, este conceito, o gradiente é um elemento responsável por utilizar as cores para criar efeitos estéticos. Podemos pensar no estilo “degradê”, onde uma cor vai perdendo sua saturação conforme avançamos a visão na imagem.

Para realizarmos a criação de um gradiente em nossa página web, basta utilizar o código “#grad” e definir os valores entre as chaves. Lembrando que um gradiente necessita “ir” de uma cor até outra.



*Exemplo de gradiente*



*Imagem comparando como cada valor é descrito.*

```

1  <html>
2    <head>
3      <style>
4
5      #grad1 {
6        height: 100px;
7        background-color: red;
8        background-image: linear-gradient(0deg, red, yellow);
9      }
10
11    </style>
12  </head>
13  <body>
14
15    <h1>Exemplo de Gradiente Linear</h1>
16
17    <div id="grad1" style="text-align:center;"></div><br>
18
19  </body>
20 </html>

```

*Imagem exemplificando o uso de gradiente no CSS.*

## Botões

O CSS permite a criação de botões para nossas páginas de internet criadas. Os botões abrangem funções como, selecionar alguma opção, enviar dados e algumas outras.



*Comparação entre botão padrão(esquerdo) e estilizado pelo CSS(direito).*

Para realizar a criação de um botão, basta utilizar o elemento “<button>”. Este elemento irá gerar um botão padrão, mas podemos estilizar ele por meio da criação de uma classe de CSS.

```
<html>
<head>
<style>
.button {
  background-color: #4CAF50;
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  cursor: pointer;
}
</style>
</head>
<body>

<h2>Botões CSS</h2>

<button>Default Button</button>

<button class="button">Button</button>

</body>
</html>
```

Muitos outros elementos podem ser utilizados para a criação de botões, como tamanho da fonte, alinhamento e até mesmo arredondamento, entre muitos outros.

## Colunas

Um artigo que contém muitas linhas, pode ser dividido em colunas para facilitar a leitura e organização, por meio do CSS. O CSS, também permite decidirmos todos os aspectos destas colunas, como por exemplo, número de colunas, largura, cor e muito mais.

```

1 <html>
2 <head>
3 <style>
4 .newspaper {
5   column-count: 3;
6   column-gap: 40px;
7   column-rule: 1px solid lightblue;
8 }
9 </style>
10 </head>
11 <body>
12
13 <h1>História do HTML</h1>
14
15 <div class="newspaper">
16 O HTML foi criado em 1991, por Tim Berners-Lee, no CERN (European Council for Nuclear Research) na Suíça. Inicialmente o HTML foi projetado
17 para interligar instituições de pesquisa próximas, e compartilhar documentos com facilidade. Em 1992, foi liberada a biblioteca de
18 desenvolvimento WWW (World Wide Web), uma rede de alcance mundial, que junto com o HTML proporcionou o uso em escala mundial da WEB.
19 </div>
20 </body>
21 </html>

```

*Exemplo de artigo dividido em colunas.*

Para que possamos utilizar as colunas em CSS, basta utilizar o elemento “column”. O elemento apresenta vários valores e definições diferentes para que seja possível a criação. Observação: A medida utilizada pelo elemento são pixels.

Propriedade	Descrição
column-count	Especifica o número de colunas em que um elemento deve ser dividido.
column-fill	Especifica como preencher as colunas.
column-gap	Especifica o intervalo entre as colunas.
column-rule	Uma propriedade abreviada para definir todas as propriedades do elemento “column-rule”
column-rule-color	Especifica a cor entre as células das tabelas.
column-rule-style	Especifica o estilo entre as células das tabelas.
column-rule-width	Especifica a largura entre as células das tabelas.
column-span	Especifica quantas colunas um elemento deve abranger.
column-width	Especifica uma largura entre as células das tabelas.

column	Uma propriedade abreviada para definir a largura da coluna e a contagem de células.
--------	---

*Tabela contendo os valores possíveis para o elemento de tabelas.*

# Transições, animações, Tooltip

## Transições

As transições CSS permitem a alteração das propriedades dos elementos, como altura, largura, cor, tamanho do texto, entre outros. Para que uma transição seja feita, é necessário que especifiquemos a propriedade no qual será alterada e o tempo de duração da transição.

```

1  <html>
2    <head>
3      <style>
4        div {
5          width: 100px;
6          height: 100px;
7          background: green;
8          transition: width 2s, height 4s;
9        }
10
11       div:hover {
12         width: 300px;
13         height: 300px;
14       }
15     </style>
16   </head>
17   <body>
18
19     <h1>Exemplo de transição</h1>
20
21     <p>Coloque o cursor em cima do elemento verde para ver a transição.</p>
22
23     <div></div>
24
25   </body>
26 </html>

```

*Exemplo de transição.*

Nas transições também há muito mais possibilidades, como por exemplo definir o tempo da transição, a largura e altura a serem alteradas, entre muitos



outros. Observação: As medidas utilizadas são pixels e segundos(quando remete a tempo).

Propriedade	Descrição
transition	Uma propriedade abreviada para definir as quatro propriedades de transição em uma única.
transition-delay	Especifica um delay/atraso (em segundos) para o efeito de transição.
transition-duration	Especifica quantos segundos(ou milissegundos) um efeito de transição leva para ser concluído.
transition-property	Especifica o nome da propriedade CSS para a qual se destina o efeito de transição.
transition-timing-function	Especifica a curva de velocidade do efeito de transição.

## Animações

O CSS permite a criação de animações para os elementos. Estas animações podem ser aplicadas em elementos como imagens, botões, etc. Para isso, é necessário que definamos o que e como a animação deve fazer.

```

1 <html>
2 <head>
3 <style>
4   div {
5     width: 100px;
6     height: 100px;
7     background-color: red;
8     position: relative;
9     animation-name: example;
10    animation-duration: 4s;
11  }
12
13  @keyframes example {
14    0% {background-color:red; left:0px; top:0px;}
15    25% {background-color:yellow; left:200px; top:0px;}
16    50% {background-color:blue; left:200px; top:200px;}
17    75% {background-color:green; left:0px; top:200px;}
18    100% {background-color:red; left:0px; top:0px;}
19  }
20 </style>
21 </head>
22 <body>
23   <h1>Animação CSS</h1>
24   <a>Nota: A animação ocorre apenas uma vez e volta ao estado original.
25   <div></div>
26 </body>
27 </html>

```

*Exemplo de animação.*

Propriedade	Descrição
@keyframes	Especifica o código da animação.
animation	Uma propriedade abreviada para definir todas as propriedades de animação.
animation-delay	Especifica um delay(atraso) para o início de uma animação.
animation-direction	Especifica se uma animação deve ser reproduzida para frente, para trás ou em ciclos alternados.
animation-duration	Especifica quanto tempo uma animação deve levar para completar um ciclo.
animation-fill-mode	Especifica um estilo para o

	elemento quando a animação não está sendo reproduzida (antes de começar, depois de terminar ou ambos).
animation-iteration-count	Especifique o número de vezes que uma animação deve ser reproduzida.
animation-name	Especifica o nome da animação @keyframe.
animation-play-state	Especifica se a animação está em execução ou pausada.
animation-timing-function	Especifica a curva de velocidade da animação.

## Tooltip

O “Tooltip”, é um recurso que permite a adição de uma informação extra em algum elemento. Ele serve para que quando o usuário posicionar o cursor em cima de um elemento da página web, uma informação seja exibida. Para a criação de um “tooltip”, é necessário utilizar o comando “.tooltip” e os argumentos dentro das chaves.

```

1  <html>
2  <style>
3  .tooltip {
4      position: relative;
5      display: inline-block;
6      border-bottom: 1px dotted black;
7  }
8
9  .tooltip .tooltiptext {
10     visibility: hidden;
11     width: 120px;
12     background-color: black;
13     color: #fff;
14     text-align: center;
15     border-radius: 6px;
16     padding: 5px 0;
17
18
19     position: absolute;
20     z-index: 1;
21 }
22
23 .tooltip:hover .tooltiptext {
24     visibility: visible;
25 }
26 </style>
27 <body style="text-align:center;">
28
29     <h2>Tooltip Básico</h2>
30
31     <div class="tooltip">Ponha o cursor aqui
32         <span class="tooltiptext">Informação Extra</span>
33     </div>
34
35     <p>Note que o posicionamento da informação extra pode ser alterado para uma melhor visibilidade.</p>
36
37 </body>
38 </html>

```

### *Criação de um Tooltip.*

O “Tooltip”, também pode ter seu posicionamento, cor e formato da borda alterados. Para isto é necessário que definamos os valores na sua declaração no código. Além disso, o CSS também nos permite incluir animações no “Tooltip”.

```

1  <html>
2    <style>
3      .tooltip {
4        position: relative;
5        display: inline-block;
6        border-bottom: 1px dotted black;
7      }
8
9      .tooltip .tooltiptext {
10       visibility: hidden;
11       width: 120px;
12       background-color: black;
13       color: #fff;
14       text-align: center;
15       border-radius: 6px;
16       padding: 5px 0;
17       position: absolute;
18       z-index: 1;
19       top: -5px;
20       right: 110%;
21     }
22
23     .tooltip .tooltiptext::after {
24       content: "";
25       position: absolute;
26       top: 50%;
27       left: 100%;
28       margin-top: -5px;
29       border-width: 5px;
30       border-style: solid;
31       border-color: transparent transparent transparent black;
32     }
33     .tooltip:hover .tooltiptext {
34       visibility: visible;
35     }
36   </style>
37   <body style="text-align:center;">
38     <h2>Tooltip esquerda com flecha direita</h2>
39     <div class="tooltip">Ponha o cursor aqui.
40       <span class="tooltiptext">Informação extra.</span>
41     </div>
42   </body>
43 </html>
44

```

*Tooltip à esquerda com flecha à direita.*

## Tarefa de Casa

Altere os botões do código a seguir para ficarem iguais à imagem de referência. **Dica:** Utilize o gradiente para definir a cor e lembre do uso de classes. Código hexadecimal da cor verde: #4CAF50. Código hexadecimal da cor azul: #008CBA.

```
1  <html>
2  <head>
3  <style>
4  .button {
5      border: none;
6      color: white;
7      padding: 15px 32px;
8      text-align: center;
9      text-decoration: none;
10     display: inline-block;
11     font-size: 16px;
12     margin: 4px 2px;
13     cursor: pointer;
14 }
15
16 </style>
17 </head>
18 <body>
19
20 <h1>Altere a cor dos botões</h1>
21
22 <button>Verde</button>
23 <button>Azul</button>
24
25 </body>
26 </html>
```

*Código para realizar o exercício*

## Altere a cor dos botões



*Resultado esperado*