

CSS



Artist Image :Descourtilz, Jean-Théodore | Dates:179?-1855

Guia iniciante:

CSS

# SUMMARY

---

## SELETORES CSS3

05

*Aqui você irá ver como seletores de atributo permite aplicar formatação específicas com base no valor do atributo.*

## CAIXAS BOX MODEL

06

*O que é caixas model? E por quê ela é tão fundamental?*

## BORDAS ARREDONDADAS

07

*Definindo uma borda arredondada com exemplos*

## BOTÕES ARREDONDADOS

08

*Criando botões arredondados e Box Sizing*

# ABOUT ME



*Brunna Croches*

***Developer Full Stack***

Brunna Croches é Dev FullStack, advogada e empreendedora.

Apaixonada por tech, vem adquirindo vasto conhecimento na área.

Desenvolveu projetos ricos em diversidade, buscando captar as próximas tendências e necessidades do mercado.

Neste e-book você aprenderá ou recapitulará de forma simplificada e otimizados conceitos de programação feito por ela.

*let's share*

## 5- Seletores CSS3

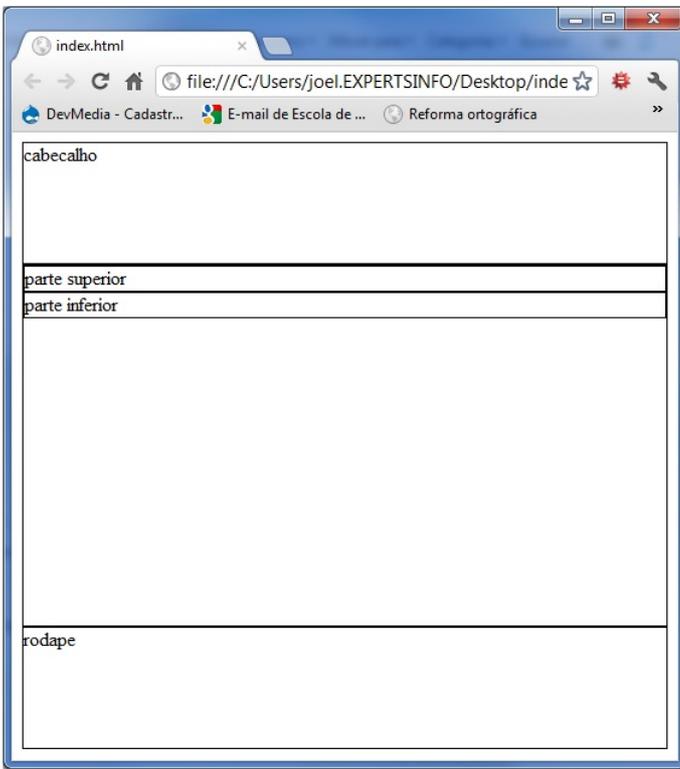
### SELETORES CSS POR ATRIBUTO

Estes seletores permitem aplicar uma determinada formatação a certos elementos com base no valor de um atributo.

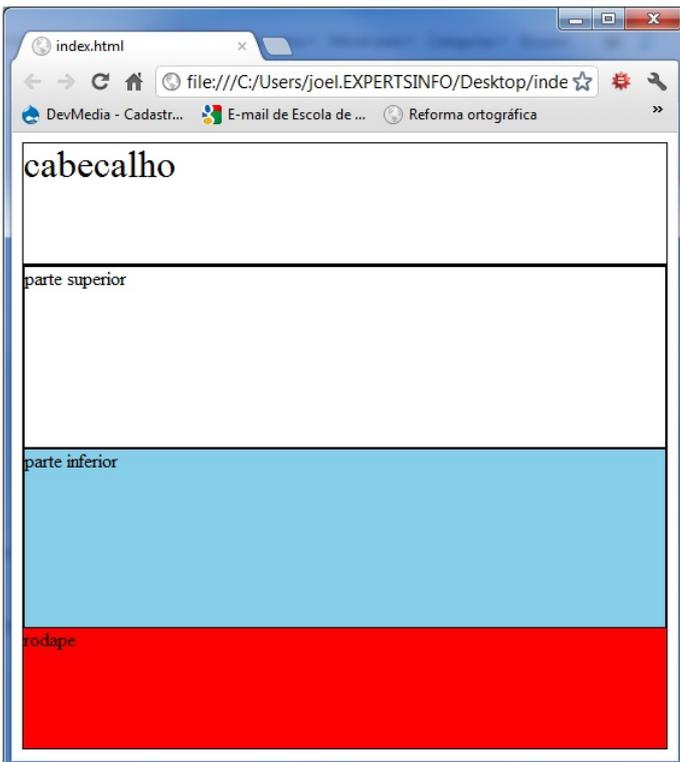
<b>IGUAL " VALOR "</b>	<b>E[atributo="valor"]:</b> elementos do tipo E cujo atributo <b>seja igual</b> a valor.	<b>E[atributo="valor"]</b>	div[id="cabecalho"]{ font-size:xx-large; }
<b>COMECE COMA PALAVRA " VALOR "</b>	<b>E[atributo^="valor"]:</b> elementos do tipo E cujo valor do atributo <b>comece com a palavra valor</b> .	<b>E[atributo^="valor"]</b>	div[id^="parte"]{ height:50%; }
<b>TERMINE COM A PALAVRA "VALOR"</b>	<b>E[atributo\$="valor"]:</b> elementos do tipo E cujo valor do atributo <b>termine com a palavra valor</b> .	<b>E[atributo\$="valor"]</b>	div[id\$="inferior"]{ background:#87CEEB; }
<b>CONTENHA A PALAVRA "VALOR"</b>	<b>E[atributo*="valor"]:</b> elementos do tipo E cujo valor do atributo <b>contenha a palavra valor</b> .	<b>E[atributo*="valor"]</b>	div[id*="roda"]{ background:red; }

### PRATICA

```
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="seletores.css"/>
    <style>
      div{border:solid 1px;}
      #cabecalho{height:100px;}
      #corpo{height:300px;}
      #rodape{height:100px;}
    </style>
  </head>
  <body>
    <div id="cabecalho">cabecalho</div>
    <div id="corpo">
      <div id="parte_superior">parte superior</div>
      <div id="parte_inferior">parte inferior</div>
    </div>
    <div id="rodape">rodape</div>
  </body>
</html>
```



```
div[id="cabecalho"]{  
    font-size:xx-large;  
}  
  
div[id^="parte"]{  
    height:50%;  
}  
  
div[id$="inferior"]{  
    background:#87CEEB;  
}  
  
div[id*="roda"]{  
    background:red;  
}
```



# SELETORES CSS POR POSIÇÃO

Estes seletores funcionam **com base na posição do elemento no documento ou no controle que o contém** (elemento pai). Por exemplo, podemos seleccionar o segundo elemento contido dentro de uma div, independente do seu tipo.

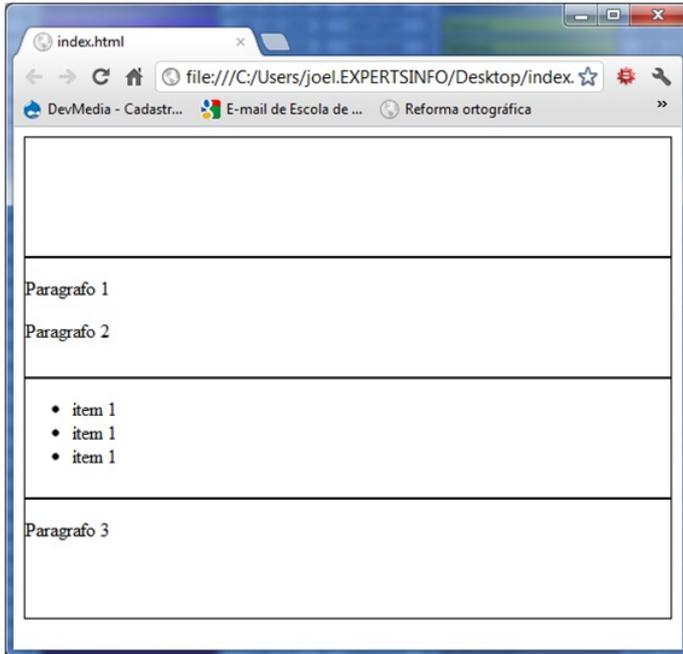
<b>Raiz do documento ( HTML 5)</b>	<b>:root:</b> seleciona o elemento <b>raiz</b> do documento, no caso, é a tag HTML.	<b>:root:</b>	
<b>Seja o N-ésimo</b>	<b>E:nth-child(N):</b> seleciona o elemento do tipo E que <b>seja o N-ésimo</b> filho do seu elemento pai.	<b>E:nth-child(N):</b>	div:nth-child(1){ background:red; }
<b>Começando a contagem de baixo para cima</b>	<b>E:nth-last-child(N):</b> semelhante ao anterior, mas <b>começando a contagem de baixo para cima</b> .	<b>E:nth-last-child(N):</b>	div:nth-last-child(2){  background:#AB82FF; }
<b>N-ésimo do seu tipo</b>	<b>E:nth-of-type(N):</b> seleciona o elemento que seja o <b>N-ésimo do seu tipo</b> .	<b>E:nth-of-type(N):</b>	li:nth-of-type(1){ color:green; }
<b>Contando de baixo para cima</b>	<b>E:nth-last-of-type(N):</b> semelhante ao anterior, mas <b>contando de baixo para cima</b> .	<b>E:nth-last-of-type(N):</b>	li:nth-last-of-type(1){ color:yellow; }
<b>Último filho do elemento pai</b>	<b>E:last-child:</b> seleciona o elemento E que seja o <b>último filho do elemento pai</b>	<b>E:last-child:</b>	div:last-child{ background:orange; }
<b>Primeiro do seu tipo dentro do elemento pai</b>	<b>E:first-of-type:</b> seleciona o elemento E que seja o <b>primeiro de seu tipo dentro do elemento pai</b> .	<b>E:first-of-type:</b>	div:first-of-type{ border:solid 5px green; }
<b>O último elemento</b>	<b>E:last-of-type:</b> semelhante ao anterior, mas com o <b>último elemento</b> .	<b>E:last-of-type:</b>	div:last-of-type{ border:solid 5px brown; }
<b>Único filho do seu elemento pai</b>	<b>E:only-child:</b> seleciona o elemento E que seja o <b>único filho do seu elemento pai</b> .	<b>E:only-child:</b>	p:only-child{ font-size:30px; }
<b>Único filho deste tipo dentro de seu elemento pai</b>	<b>E:only-of-type:</b> seleciona o elemento que <b>seja o único filho deste tipo dentro de seu elemento pai</b> .	<b>E:only-of-type:</b>	ul:only-of-type{ text-decoration:underline; }

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="seletores.css"/>
<style>
div{border:solid 1px; height:100px;}
</style>
</head>
<body>
<div id="barra_1"></div>
<div id="barra_2">
<p>Paragrafo 1</p>
```

```

    <p>Paragrafo 2</p>
  </div>
  <div id="barra_3">
    <ul>
      <li>item 1</li>
      <li>item 1</li>
      <li>item 1</li>
    </ul>
  </div>
  <div id="barra_4">
    <p>Paragrafo 3</p>
  </div>
</body>
</html>

```



```

div:nth-child(1){
  background:red;
}

div:nth-last-child(2){
  background:#AB82FF;
}

li:nth-of-type(1){
  color:green;
}

li:nth-last-of-type(1){
  color:yellow;
}

div:last-child{
  background:orange;
}

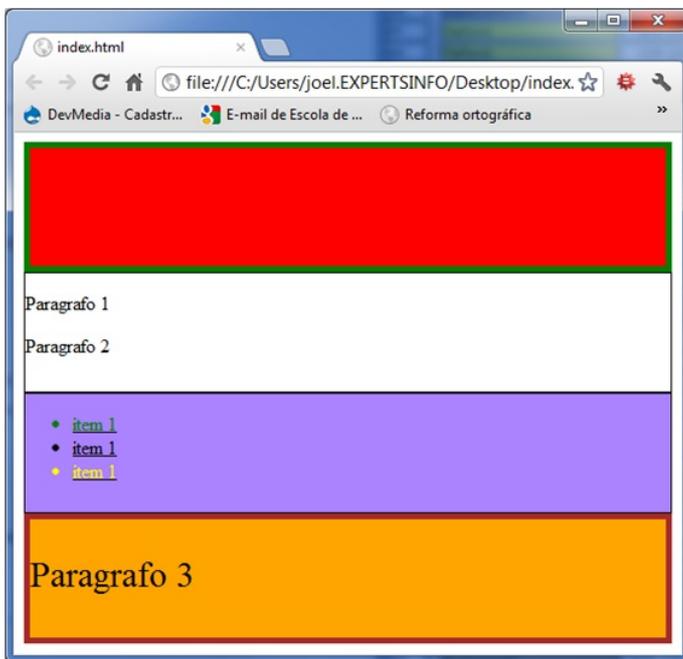
div:first-of-type{
  border:solid 5px green;
}

div:last-of-type{
  border:solid 5px brown;
}

p:only-child{
  font-size:30px;
}

ul:only-of-type{
  text-decoration:underline;
}

```



## SELETORES CSS POR ESTADO

Estes, por sua vez, funcionam **com base no estado atual do controle** (vazio, habilitado, desabilitado etc.).

Selecionar elemento vazio (sem elemento filho)	<b>E:empty:</b> seleciona o elemento E que esteja vazio, sem nenhum elemento filho.	<b>E:empty:</b>
Selecionar objeto alvo de um link interno	<b>:target:</b> seleciona o objeto alvo de um link interno (aqueles que são precedidos por # e que direcionam o usuário para um elemento na página).	<b>:target:</b>
Seleciona Elemento habilitado para o usuário	<b>E:enabled:</b> seleciona o elemento E que esteja habilitado para o usuário	<b>E:enabled:</b>
Seleciona Elementos Desabilitados	<b>E:disabled:</b> semelhante e ao anterior, mas para elementos desabilitados.	<b>E:disabled:</b>
Seleciona elementos que estejam marcados	<b>E:checked:</b> seleciona elementos que estejam marcados (radios e checkboxes).	<b>E:checked:</b>
Formata conteúdo selecionado pelo usuário	<b>::selection:</b> formata o conteúdo selecionado pelo usuário.	<b>::selection:</b>
Seleciona elementos que não atendem ao seletor simples	<b>:not(P):</b> seleciona os elementos que não atendem ao seletor simples P.	<b>:not(P):</b>
Seleciona todo elemento precedido por outro elemento	<b>E~F:</b> seleciona todo elemento F que seja precedido por um elemento E.	<b>E~F:</b>

Na pratica

```

<html>
  <head>
    <link rel="stylesheet" type="text/css" href="seletores.css"/>
    <style>
      div{border:solid 1px; height:100px;}
    </style>
  </head>
  <body>
    <h1>Selecione esse texto</h1>
    <p><a href="#Texto1">Link para o Texto 1</a></p>
    <p><a href="#Texto2">Link para o Texto 2</a></p>
    <div>
      <p id="Texto1">Esse e o TEXTO 1</p>
      <p id="Texto2">Esse e o TEXTO 2</p>
    </div>
    <div id="divVazia"></div>
    <div>
      <span>Span antes do input</span><br/>
      <input type="checkbox" checked="checked" value="Opcao 1">Opcao 1
      <input type="checkbox" value="Opcao 2">Opcao 2
      <input type="text" disabled="disabled" value="desabilitado"/>
      <input type="text" value="habilitado"/><br/>
      <span>Span apos o input</span>
    </div>
  </body>
</html>

```



```

:target{
  color:green;
  font-weight:bold;
}

::selection{
  color:red;
}

.empty{
  border:solid 1px;
  height:50px;
  background:gray;
}

input:checked{
  text-decoration:underline;
}

h1
{
  font-family:"Courier New";
}

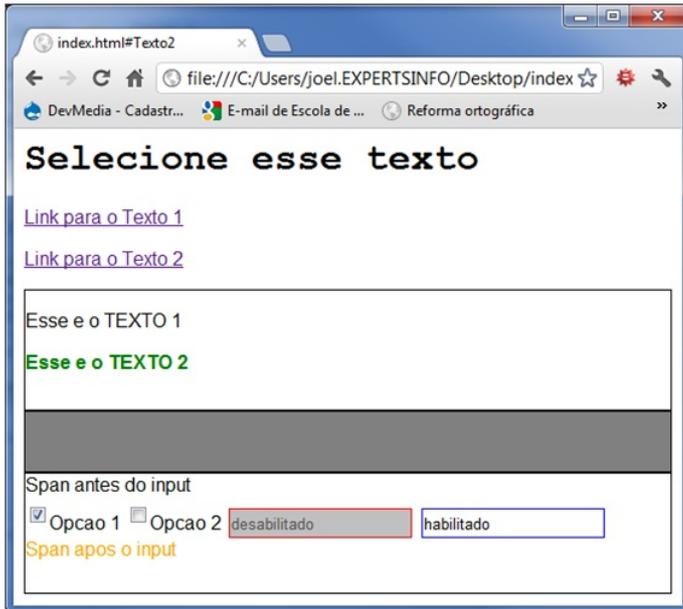
:not(h1)
{
  font-family:"Arial";
}

```

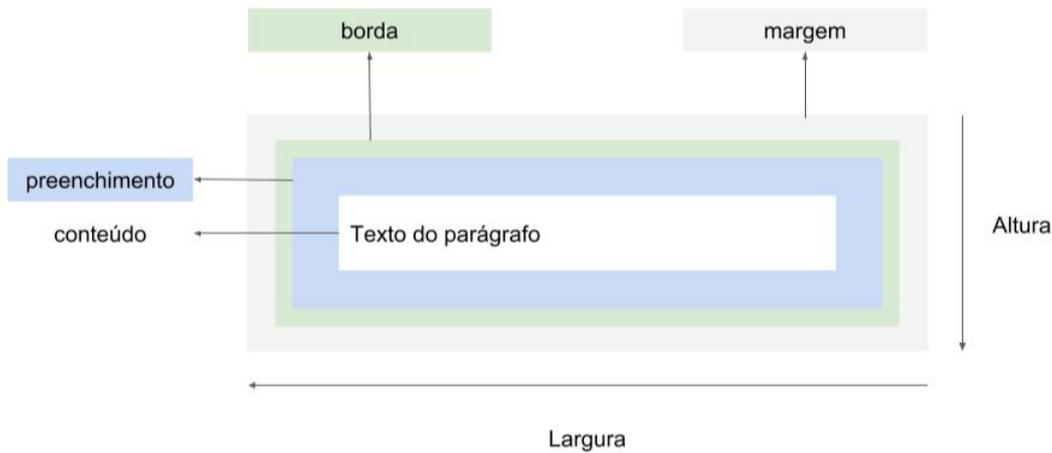
```
input-span{
  color: orange;
}

input:disabled{
  border:solid 1px red;
  background-color: silver;
  height:25px;
}

input:enabled{
  border:solid 1px blue;
  background-color: #fff;
  height:25px;
}
```



## 6-Caixas Box Model



### Box Model

<p><b>O que é?</b></p>	<p>O motor de renderização do navegador representa cada elemento como uma caixa retangular, de acordo com o padrão definido pelo CSS conhecido como box (caixa) model</p> <p><b>O conteúdo do elemento é uma das quatro partes que compõem o box, sendo as demais o seu preenchimento, borda e margem.</b></p>	
<p><b>Por que é fundamental ?</b></p>	<p><b>é o conteúdo somado a margem, borda e preenchimento do elemento que determinam o espaço que o mesmo ocupa na tela do navegador.</b> Por essa razão entender o box model é fundamental para a criação de layouts consistentes.</p>	
<p><b>Controle das áreas que compõem um box</b></p>	<p><b>margin</b> <b>padding</b> <b>width</b> <b>height</b></p>	

### Width e Height

<p><b>Width e Height</b></p>	<p>Propriedades <b>width e height</b> permitem <b>controlar a altura e largura</b>, da área de conteúdo de um elemento.</p>	<pre>div {   height: 16px; width: 20px; display: inline-block; }</pre> <p>Exemplo no qual um parágrafo recebe 16 pixels de <b>altura</b> e 20 de <b>largura</b></p>	<pre>blockquote {   height: 400px; width: 200px; }</pre>
------------------------------	---	---	--

Especificar o quanto ele poderá crescer ou se encolher na página

Definir como a altura e largura de um elemento são calculadas { content-box }

Definir como a altura e largura + borda e preenchimento { border-box }

Fazemos isso com as propriedades: **min-width, max-width, min-height, max-height**

Essa propriedade define como a altura e largura de um elemento são calculadas. O valor padrão de **boxsizing** é **content-box**, que define que a largura e altura de um elemento serão definidas pelos valores atribuídos para as propriedades height e width

O segundo valor possível para essa propriedade é **border-box**, que define que a altura e largura de um elemento deve **levar em consideração também a sua borda e preenchimento**

```
div { width: 100px; min-width: 80px; }
```

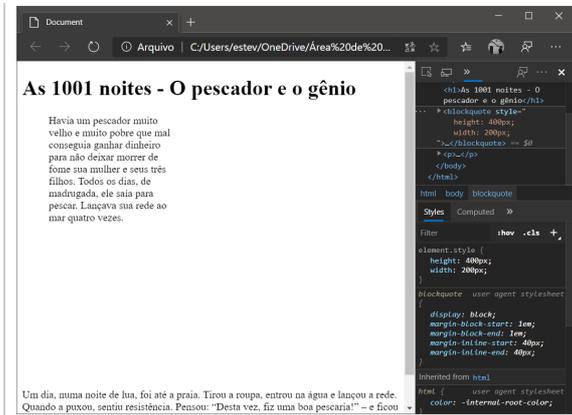
Algumas vezes pode ser necessário **ter maior controle sobre as dimensões de um elemento**, especificando o quanto ele poderá **crescer ou se encolher** na página.

```
div { height: 16px; width: 20px; padding: 16px; box-sizing: content-box; }
```

No caso acima, **as dimensões finais do elemento serão 48 pixels de altura** (height de 16px + padding de 2 \* 16 px = 32px) e 52 pixels de largura (width de 20px + padding de 2 \* 16px = 32px) pois os valores das propriedades (height e padding) e (width e padding) **serão somados** quando o motor de renderização do navegador os apresentar.

```
div { height: 16px; width: 20px; padding: 16px; box-sizing: border-box; }
```

Neste caso, uma vez que definimos o padding do elemento como 16 pixels, a altura final do elemento será de 32 pixels, considerando que este valor corresponde a área de preenchimento superior e inferior (16 + 16). Sendo a altura, height, de 16 pixels, ela será considerada dentro dos 32 pixels já calculados a partir da área de preenchimento. Isso **acontece pois no box-**



	<p>sizing o elemento <b>tende a se encolher para se ajustar ao espaço correspondente às suas dimensões.</b> Uma vez que a altura, height, do elemento supere os 32 pixels definidos pelo padding, por exemplo ao receber 33 pixels, a área de conteúdo do elemento passará a ter 1 pixel e a altura final do elemento passará a ser um pixel maior (<math>33 - 32 = 1</math>). O mesmo se aplica a largura, width.</p>
--	--

## Padding

<b>Padding</b>	<p>propriedade padding <b>controla a área de preenchimento de um elemento</b>, que envolve o seu conteúdo.</p>	<p>div { <b>padding</b>: 16px 10px 20px 30px; }</p> <p><b>16</b> pixels corresponde a área <b>superior</b>, top, <b>10</b> pixels a <b>direita</b>, <b>20</b> a área <b>inferior</b> e <b>30</b> a <b>esquerda</b></p>
<b>Padding RESUMIDO EM 2 VALORES</b>	<p>Podemos ainda escrever essas propriedades de <b>forma curta</b>, sendo isso útil em alguns casos nos quais <b>desejamos atribuir o mesmo valor para a parte superior e inferior de uma elemento</b>, tal qual para sua área de <b>preenchimento esquerda e direita</b></p>	<p>div { <b>padding</b>: 16px 10px; }</p> <p>a <b>altura</b> da área de preenchimento do elemento terá <b>32 pixels (16 + 16)</b>, enquanto sua <b>largura será de 20 pixels</b>.</p>

## Margin

<b>Margin</b>	<p>A propriedade margin <b>controla a área de margem de um elemento, que influencia no espaçamento</b> entre esse e os demais elementos ao seu redor.</p>	<p>div { <b>margin</b>: 16px 10px 20px 30px; }</p> <p>16 pixels corresponde a área <b>superior</b>, top, 10 pixels a <b>direita</b>, 20 a área <b>inferior</b> e 30 a <b>esquerda</b></p>
<b>Margin Escrevendo esses valores de forma curta</b>	<p>Podemos ainda <b>escrever esses valores de forma curta</b>, sendo isso útil em alguns casos nos quais desejamos <b>atribuir o mesmo valor para a parte superior e inferior</b> de uma elemento, tal qual para sua área de margem esquerda e direita</p>	<p>div { <b>margin</b>: 16px 10px; }</p> <p>o elemento terá uma <b>distância</b> de 32 pixels (16 + 16) <b>daqueles acima e abaixo dele</b>, enquanto os lados da área de margem criarão uma distância de 20 pixels.</p>

## Border

	<p>A propriedade border <b>define o estilo da borda de um elemento</b>. Ela pode receber</p>	<p>div { <b>border</b>: 1px solid #eee; }</p> <p>o primeiro valor define a <b>largura da borda</b>, que neste caso será de 1 pixel.</p>
--	--	---

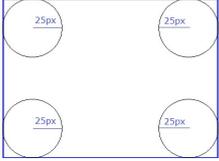
<p><b>Border</b></p> <p><b>Para determinar a largura da borda</b></p>	<p>uma lista de valores de diferentes tipos</p> <p>Para determinar a <b>largura da borda</b> podemos utilizar a propriedade <b>border-width</b>. Ao utilizar essa propriedade, podemos <b>definir individualmente as bordas superior, esquerda, inferior e direita</b> de um elemento:</p>	<p>O segundo valor <b>informa o estilo da borda</b>. Por fim podemos <b>atribuir a borda uma cor</b>.</p> <pre>div { border-width: 1px 2px 1px 2px; }</pre> <p>atribuímos 1 pixel para a parte <b>superior/inferior</b> e 2 pixels para a borda <b>esquerda/direita</b> do elemento.</p>
<p><b>Definir o valor dessa propriedade</b></p>	<p>Podemos ainda <b>utilizar palavras chave para definir o valor dessa propriedade</b>, sendo <b>elas thin, medium e thick</b>.</p>	<pre>div { border-width: thin; }</pre> <p>No exemplo acima, <b>o elemento terá uma borda fina aplicada</b>.</p>
<p><b>Cancelar os valores anteriormente</b></p>	<p>Essa propriedade pode receber ainda o valor <b>unset</b>, que <b>cancela os valores anteriormente aplicados</b>.</p>	<p>unset</p>
<p><b>Determinamos como a borda do elemento será apresentada</b></p>	<p>Com a propriedade <b>border-style</b> determinamos <b>como a borda do elemento será apresentada</b>. Ela pode ser <b>sólida</b> com o valor <b>solid</b>, <b>pontilhada</b> com o valor <b>dotted</b> ou mesmo <b>dupla</b>, se aplicado o valor <b>dashed</b>.</p>	<pre>div { border-style: dotted solid; }</pre> <p>Nesse exemplo, o elemento <b>receberá uma borda superior/inferior pontilhada e esquerda/direita sólida</b>. Podemos ainda utilizar o valor <b>none</b>, que <b>cancela</b> qualquer outro anteriormente aplicado.</p>

# 7- Bordas Arredondadas

## Borda Arredondada

### Border-Radius



<b>Border-Radius</b>	defina uma <b>borda arredondada</b> de acordo com o valor fornecido, que indica o raio de uma circunferência teoricamente posicionada nos vértices do elemento e a parti		
<b>Definir border-radius GERAL</b>	A propriedade <b>border-radius</b> pode ser definida de duas formas: definindo um <b>valor único</b> para todos os vértices (pontas)	<pre>&lt;html&gt; &lt;head&gt; &lt;title&gt;Trabalhando com Bordas em CSS3&lt;/title&gt; &lt;style type="text/css"&gt; #div1{ width:100px; height:100px; border:solid 1px; <b>border-radius:20px;</b> } &lt;/style&gt; &lt;/head&gt; &lt;body&gt; &lt;div id="div1"/&gt; &lt;/body&gt; &lt;/html&gt;</pre>	 geral
<b>Definir border-radius INDIVIDUALMENTE</b>	Especificando <b>individualmente</b> o valor de cada vértice.	<pre>&lt;html&gt; &lt;head&gt; &lt;title&gt;Trabalhando com Bordas em CSS3&lt;/title&gt; &lt;style type="text/css"&gt; #div1{ width:100px; height:100px; border:solid 1px; <b>border-radius:0px 20px 0px 20px;</b> } &lt;/style&gt; &lt;/head&gt; &lt;body&gt; &lt;div id="div1"/&gt; &lt;/body&gt; &lt;/html&gt;</pre>	 individual

- 01 border-radius: valor;
- 02 border-radius: valor1 valor2;
- 03 border-radius: valor1 valor2 valor3 valor4
- 04 border-top-left-radius: valor;
- 05 border-top-right-radius: valor;

**Linha 1:** Dessa forma **o mesmo valor é aplicado a todos os vértices (pontas) do objeto, obtendo assim uma aparência homogênea em todas as bordas;**

## Border Radius - Exemplo 1

06 border-bottom-left-radius: ;valor;  
07 border-bottom-right-radius: valor

**Linha 2:** Na segunda forma o valor1 é aplicado às **extremidades superior esquerda e inferior direita**, enquanto o valor2 é aplicado aos vértices superior direito e inferior esquerdo;

**Linha 3:** Nessa terceira forma o valor1 **corresponde ao ponto** superior esquerdo, o valor2 ao ponto superior direito, o valor3 ao ponto inferior direito e o valor4 ao ponto inferior esquerdo.

**Linhas 4 a 7:** Aqui estamos **atribuindo os valores específicos de cada vértice**

## Border Radius - Exemplo 2

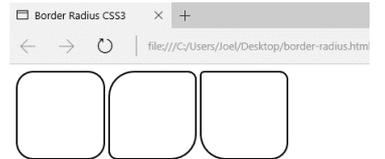
```
01 <!DOCTYPE html>
02 <html>
03 <head>
04 <meta charset="UTF-8">
05 <title>Border Radius CSS3</title>
06 <style>
07   div {
08     width: 100px;
09     height: 100px;
10     border-width: 2px;
11     border-color: black;
12     border-style: solid;
13     display: inline-block
14   }
15   #caixa1 {
16     border-radius: 20px;
17   }
18   #caixa2 {
19     border-radius: 30px 10px;
20   }
21   #caixa3 {
22     border-radius: 5px 10px 20px
30px;
23   }
24 </style>
25 </head>
26 <body>
27 <div id="caixa1"></div>
28 <div id="caixa2"></div>
29 <div id="caixa3"></div>
30 </body>
31 </html>
```

**Linhas 7 a 14:** configuramos a **aparência básica de todas as divs do exemplo** (dimensões e borda);

**Linhas 15 a 17:** configuramos a primeira div para apresentar a **borda arredondada com 20px em todas as extremidades**;

**Linhas 18 a 20:** configuramos a segunda div para apresentar a **borda arredondada com 30px nas extremidades superior esquerda e inferior direita**, e 10px **nos cantos superior direito e inferior esquerdo**;

**Linhas 21 a 23:** configuramos a terceira div para **apresentar borda arredondada** com 5px no **canto superior esquerdo**, 10px no canto superior direito, 20px no **canto inferior direito** e 30px no **canto inferior esquerdo**.



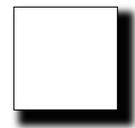
## Sombreamento Box-Shadow

### Box-Shadow

**sombreamento.** Esta propriedade é composta por alguns parâmetros e sua sintaxe de uso é a seguinte:

- **h-shadow:** posição **horizontal** da sombra.
- **v-shadow:** posição **vertical** da sombra.
- **blur:** define a **intensidade** do efeito blur (borrão) da **sombra** (opcional).
- **spread:** o **tamanho da sombra**

```
<html>
<head>
<title>Trabalhando com Bordas em
CSS3</title>
<style type="text/css">
#div1{
width:100px;
height:100px;
border:solid 1px;
box-shadow: 10px 10px 10px 5px
black;
}
</style>
</head>
<body>
<div id="div1"/>
</body>
```



	<p>(opcional).</p> <ul style="list-style-type: none"> <li><b>color:</b> a <b>cor da sombra</b>.</li> <li><b>inset:</b> define se a <b>sombra é externa (padrão)</b> ou <b>interna (inset)</b>.</li> </ul>	</html>	
--	---	---------	--

## Border Image

<h3>Border-Image</h3>	<p>Definir uma <b>imagem</b> para ser <b>usada no preenchimento da borda de um elemento</b>.</p> <p>Infelizmente esta propriedade não é suportada pelo Internet Explorer e requer o uso de prefixos para os diferentes browsers</p>	<pre>&lt;html&gt; &lt;head&gt; &lt;title&gt;Trabalhando com Bordas em CSS3&lt;/title&gt; &lt;style type="text/css"&gt; #div1{ width:100px; height:100px; border-width:10px; border-image:url(border.png) 30 30 repeat; -moz-border-image:url(border.png) 30 30 repeat; /* Firefox */ -webkit-border-image:url(border.png) 30 30 repeat; /* Safari e Chrome */ -o-border-image:url(border.png) 30 30 repeat; /* Opera */ } &lt;/style&gt; &lt;/head&gt; &lt;body&gt; &lt;div id="div1"/&gt; &lt;/body&gt; &lt;/html&gt;</pre>	<p>O primeiro estilo "estica" a imagem, preenchendo todo o elemento de uma única vez, enquanto o segundo repete a imagem até preencher toda a borda</p> 
-----------------------	---	--	--

## Elementos Circulares

<h3>Elementos Circulares</h3>	<p>Para isso, basta que o <b>border-radius seja preenchido com um valor igual à altura ou largura desse elemento</b> (que por ser quadrado, serão iguais).</p> <p>O formato circular tem sido muito utilizado, por exemplo, em fotos de perfil e galerias de imagens.</p>	<pre>&lt;img src="avatar.png"&gt; &lt;img src="avatar.png" id="img_circular"&gt;  img { width: 200px; height: 200px; }  #img_circular { border-radius: 200px; }</pre>	
-------------------------------	---	---	--

## 8- Botões arredondados

### CRIAR Botão CSS

.dvm-button

```
01 .dvm-button {
02   color: #FFF;
03   background-color: green;
04   border: none;
05   height: 60px;
06   line-height: 30px;
07   padding: 0 20px;
08   text-transform: uppercase;
09   font-weight: bold;
10   cursor: pointer;
11 }
```

Linha 01: Definição do nome da classe. Optamos pelo nome "**dvm-button**", com um prefixo que remete às iniciais da DevMedia. Esse tipo de nomenclatura é bastante comum em classes CSS;

Linha 02: Propriedade **color**. Aqui, indicamos que teremos um **texto branco (#FFF) no botão**;

Linha 03: Propriedade **background-color**. Especifica que a **cor de fundo do botão será verde**;

Linha 04: **Resetamos** a propriedade **border**, para **evitar os valores padrão do elemento <button>**;

Linha 05: A **altura** é definida para o **botão (height)** em 60 pixels;

Linha 06: Propriedade bastante importante. É ela que **centraliza verticalmente o texto do botão**. Para esse comportamento, **o valor dessa propriedade deve ser exatamente a metade da altura do elemento**;

Linha 07: Definição do **padding, a margem interna do botão**. Nesse caso, optamos por 0 (zero) no eixo y (distância do topo e base) e 20 pixels nas laterais direita e esquerda;

Linha 08: Propriedade **text-transform**, **informa que o texto do botão terá somente letras maiúsculas**, independentemente do valor inserido no HTML;

Linha 09: Propriedade **font-weight**, ou largura da fonte, informa, nesse caso, que **temos uma fonte em negrito**;

Linha 10: Por meio da propriedade **cursor**, garante que **o botão sempre terá o ponteiro como cursor**, ou seja, **no momento em que o mouse estiver no botão, teremos esse tipo de cursor**.

DevMedia - Elementos arredondados

CLIQUE AQUI!

### ARREDONDAR border-radius

```
01 .dvm-button {
02   ...
03   border-radius: 30px;
04 }
05
06 .dvm-button:hover {
07   transform: scale(1.05);
08 }
```

A parte de arredondamento do elemento, entretanto, ainda não foi criada. Para isso, existe uma propriedade específica do CSS: **border-radius**, ou raio da borda.

**O raio é uma medida utilizada no círculo** (e em outros elementos similares) **para indicar a distância da borda em relação ao centro**. Assim, ao setarmos o **border-radius**, **estamos definindo a distância que essa borda terá em relação a um centro imaginário, arredondando o elemento**.

Linha 03: Definição da propriedade **border-radius**, **responsável pelo arredondamento do elemento que utilizará essa classe** (<button>, no caso). Note que temos um raio de 30 pixels, garantindo que o elemento será totalmente arredondado nas laterais, uma vez que o mesmo possui altura de 60 pixels;

Linhas 06 e 07: Classe CSS **dvm-button no momento do hover**, **ação de manter o cursor sobre o elemento**. Nesse caso, **apenas utilizamos a propriedade transform para aumentar ligeiramente o botão**.



CRIAR  
ELEMENTO  
TOTALMENTE  
CIRCULAR

`dvm-circle`

para termos um elemento totalmente circular, basta configurar um `border-radius` de 50% da altura (`height`) e largura (`width`).

```
01 .dvm-circle {
02   margin: 5px;
03   background-color: black;
04   color: white;
05   padding: 5px;
06   height: 20px;
07   width: 20px;
08   border-radius: 15px;
09 }
```

deve ser evitada, pois é extremamente "enraizada" em um determinado tamanho; 30 pixels, no caso.

```
.dvm-circle {
  margin: 5px;
  background-color: black;
  color: white;
  padding: 5px;
  border-radius: 50%;
}
```

Linha 01: Definição da classe `dvm-circle`;

Linhas 02 a 05: Definição das propriedades básicas do elemento. Aqui, informamos que **teremos uma margem** (`margin`) de 5px em cada lado, bem como o **padding**. As **cores de fundo e frente** (`background-color` e `color`, respectivamente) também foram especificadas. Note que utilizamos as cores preta para o fundo e branca para os **elementos internos**, como textos;

Linhas 06 e 07: **Especificamos a altura e a largura do elemento circular**, ambas de 20px. **Somando-se** esses valores ao `padding` em cada direção, **temos um elemento com altura e largura** totais de 30 pixels;

Linha 08: Definição da **propriedade border-radius baseada na altura total**, somando-se o `padding`. Note que o valor é de exatamente 50% da altura e largura do elemento, gerando um visual circular.

DevMedia - Elementos arredondados

CLIQUE AQUI!

E-mail: teste@devmedia.com

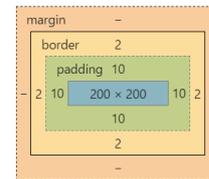
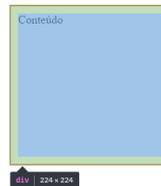
## Box Sizing

### Box Sizing

- altura total = `height` + `padding-top` + `padding-bottom` + `border-top-width` + `border-bottom-width`;
- largura total = `width` + `padding-left` + `padding-right` + `border-left-width` + `border-right-width`.

Considerando, por exemplo, um elemento com **largura e altura** iguais a 200px, **bordas** de 2px e **padding** igual a 10px, ao ser exibido na página suas **dimensões finais** seriam 224x224px (200+10+10+2+2px em ambas as direções)

```
div{
  width:200px;
  height:200px;
  border:2px solid black;
  padding:10px;
}
```

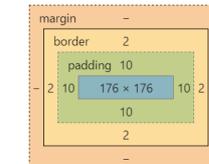
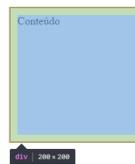


### Box Sizing

**box-sizing**, que, **uma vez definida, faz com que os valores de padding e border sejam incorporados às dimensões horizontal e vertical do elemento**. Com isso, `width` e `height` passam a representar o tamanho total, independentemente das demais propriedades.

```
div{
  width:200px;
  height:200px;
  border:2px solid black;
  padding:10px;
  box-sizing: border-box;
}
```

Observe que dessa vez a parte central foi reduzida para 176x176px (200 - 12 - 12). Para ver esses exemplos em execução,



# CONTACT



*Brunna Croches*

*Developer Full Stack*



*brunnacroches.dev*



*linkedin.com/brunnacroches*



*github.com/brunnacroches*



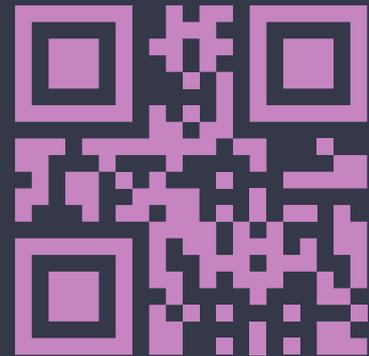
*@brunnacroches.dev*



*discord.com/brunnacroches*



*brunnacroches@gmail.com*



*let's share*