

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 adquirido 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

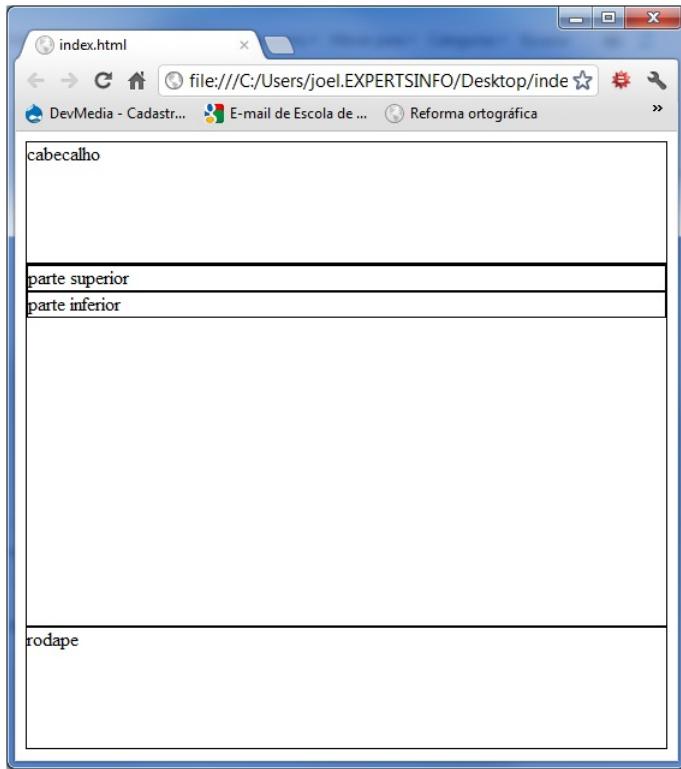
SELETORES CSS POR ATRIBUTO

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

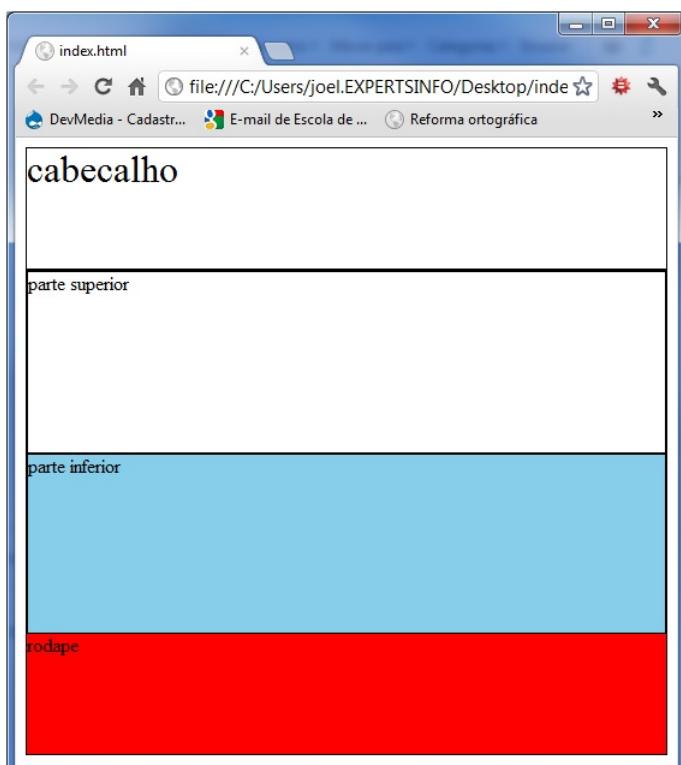
IGUAL " VALOR "	E[atributo="valor"]: elementos do tipo E cujo atributo seja igual a valor.	E[atributo="valor"]	div[id="cabecalho"]{ font-size:xx-large; }
COMECE COMA PALAVRA " VALOR "	E[atributo^="valor"]: elementos do tipo E cujo valor do atributo comece com a palavra valor .	E[atributo^="valor"]	div[id^="parte"]{ height:50%; }
TERMINE COM A PALAVRA "VALOR"	E[atributo\$=""valor"]: elementos do tipo E cujo valor do atributo termine com a palavra valor .	E[atributo\$=""valor"]	div[id\$="inferior"]{ background:#87CEEB; }
CONTENHA A PALAVRA "VALOR"	E[atributo*="valor"]: elementos do tipo E cujo valor do atributo contenha a palavra valor .	E[atributo*="valor"]	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*="rodape"]{  
    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 selecionar o segundo elemento contido dentro de uma div, independente do seu tipo.

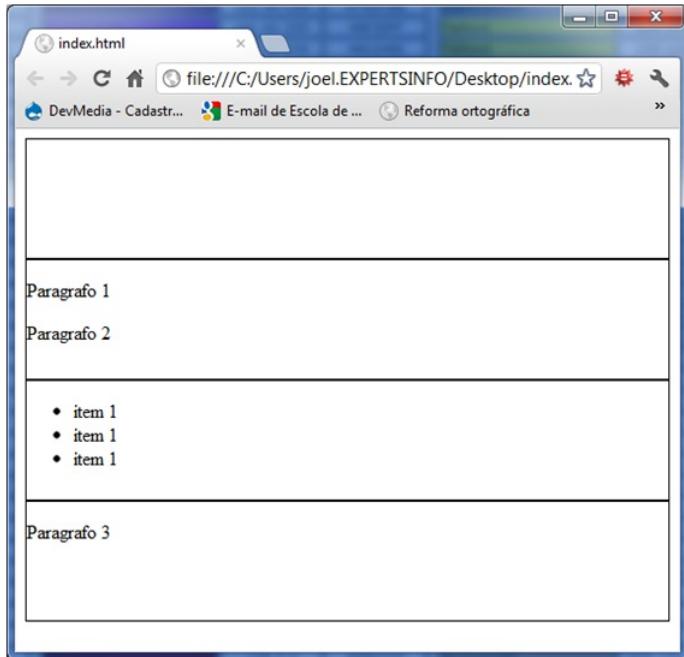
Raiz do documento (HTML 5)	:root: seleciona o elemento raiz do documento, no caso, é a tag HTML.	:root:	
Seja o N-esimo	E:nth-child(N): seleciona o elemento do tipo E que seja o N-ésimo filho do seu elemento pai.	E:nth-child(N):	div:nth-child(1){ background:red; }
Começando a contagem de baixo para cima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

```

<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;
}

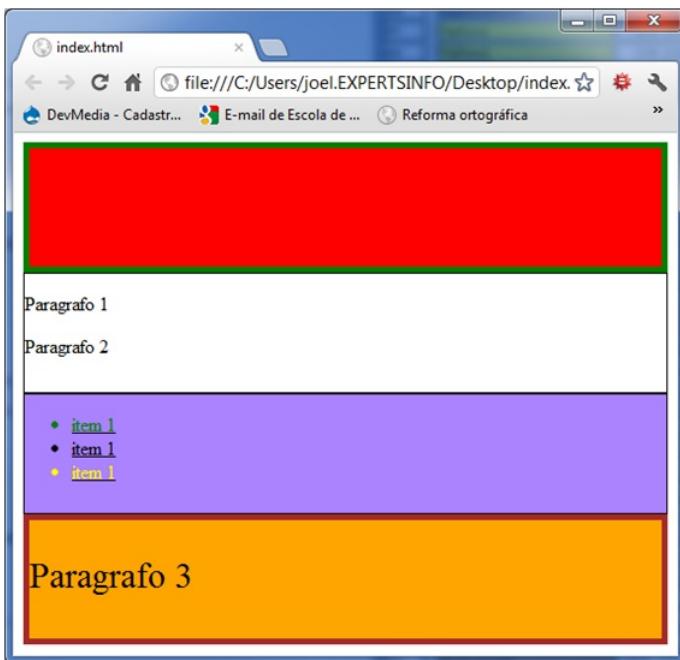
div:first-of-type{
    border:solid 5px green;
}

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

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="" value="Opcão 1">Opcão 1
      <input type="checkbox" value="Opcão 2">Opcão 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;
}

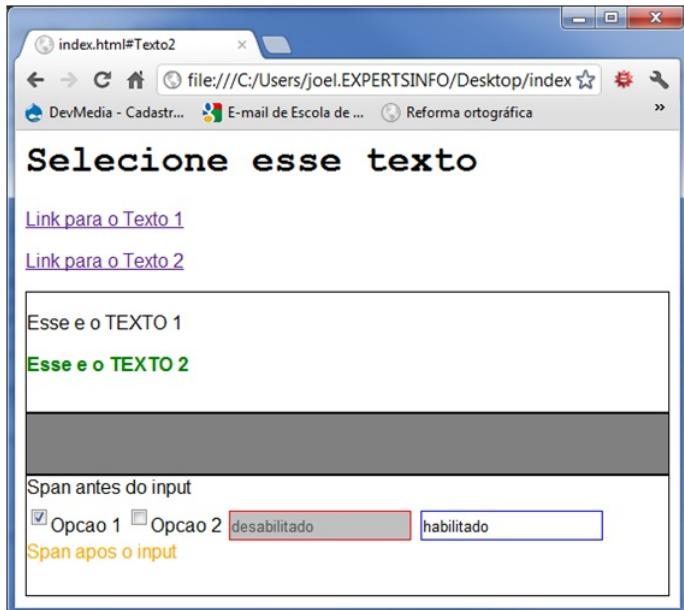
input:checked{
  text-decoration:underline;
}

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

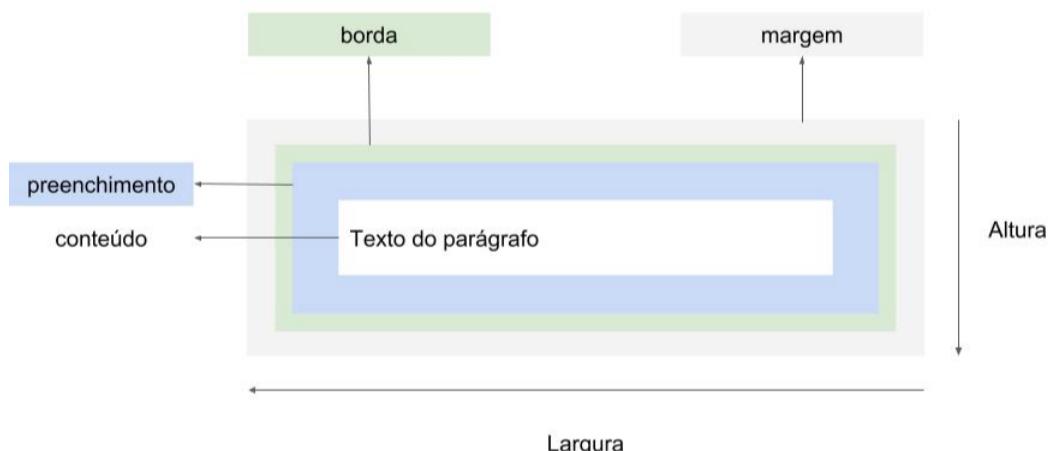
:not(h1){
  font-family:"Arial";
}

```

```
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

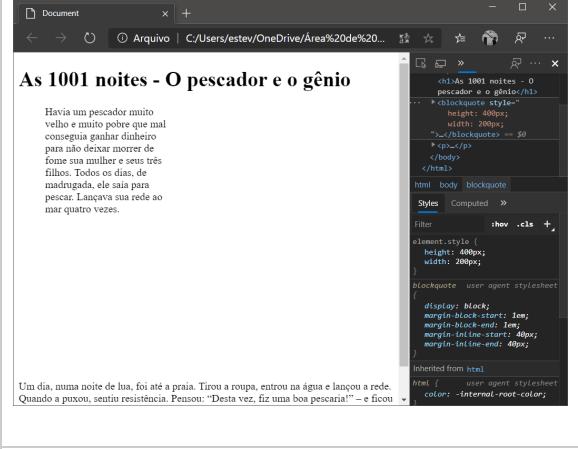


Box Model

O que é?	<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>O conteúdo do elemento é uma das quatro partes que compõem o box, sendo as demais o seu preenchimento, borda e margem.</p>	O diagrama ilustra a estrutura de um elemento HTML com o seguinte desdobramento: <ul style="list-style-type: none">Texto do parágrafo: O conteúdo principal.conteúdo: A área central, que inclui o texto e o preenchimento.preenchimento: A área entre a borda e o conteúdo.borda: A borda externa da caixa.margem: Espaço entre a borda e a borda das caixas adjacentes.Largura: A medida horizontal da caixa, composta por conteúdo, borda e margem.Altura: A medida vertical da caixa, composta por conteúdo, borda e margem.
Por que é fundamental?	<p>é o conteúdo somado a margem, borda e preenchimento do elemento que determinam o espaço que o mesmo ocupa na tela do navegador. Por essa razão entender o box model é fundamental para a criação de layouts consistentes.</p>	
Controle das áreas que compõem um box	margin padding width height	

Width e Height

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

		 <p>As 1001 noites - O pescador e o gênio</p> <p>Havia um pescador muito velho e muito pobre que mal conseguia ganhar dinheiro para养家. Um dia, ao sair de casa em busca de peixes, três filhos. Todos os dias, de madrugada, ele saía para pescar. Lançava sua rede ao mar quatro vezes.</p> <pre> <html> <head> <title>As 1001 noites - O pescador e o gênio</title> </head> <body> <p>Havia um pescador muito velho e muito pobre que mal conseguia ganhar dinheiro para养家. Um dia, ao sair de casa em busca de peixes, três filhos. Todos os dias, de madrugada, ele saía para pescar. Lançava sua rede ao mar quatro vezes.</p> </body> </html> </pre> <p>Um dia, numa noite de lua, foi até a praia. Tirou a roupa, entrou na água e lançou a rede. Quando o puxou, sentiu resistência. Pensou: "Desta vez, fiz uma boa pescaria!" — e ficou</p>
<p>Especificando quanto ele poderá crescer ou se encolher na página</p>	<p>Fazemos isso com as propriedades: min-width, max-width, min-height max-height</p>	<div data-bbox="541 540 742 601" style="background-color: #ffffcc; padding: 2px;"> <div style="border: 1px solid black; padding: 2px;">div { width: 100px; min-width: 80px; }</div> </div> <p>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.</p>
<p>Definir como a altura e largura de um elemento são calculadas { content-box }</p>	<p>Essa propriedade define como a altura e largura de um elemento são calculadas.</p> <p>O valor padrão de box-sizing é content-box, que define que a largura e altura de um elemento serão definidas pelos valores atribuídos para as propriedades height e width</p>	<div data-bbox="541 888 758 989" style="background-color: #ffffcc; padding: 2px;"> <div style="border: 1px solid black; padding: 2px;">div { height: 16px; width: 20px; padding: 16px; box-sizing: content-box; }</div> </div> <p>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.</p>
<p>Definir como a altura e largura + borda e preenchimento { border-box }</p>	<p>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.</p>	<div data-bbox="541 1538 758 1628" style="background-color: #ffffcc; padding: 2px;"> <div style="border: 1px solid black; padding: 2px;">div { height: 16px; width: 20px; padding: 16px; box-sizing: border-box; }</div> </div> <p>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>

	<p>sizing o elemento tende a se encolher para se ajustar ao espaço correspondente às suas dimensões. 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 ($33 - 32 = 1$). O mesmo se aplica a largura, width.</p>	
--	---	--

Padding

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

Margin

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

Border

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

<h2>Border</h2>	<p>uma lista de valores de diferentes tipos</p>	<p>O segundo valor informa o estilo da borda. Por fim podemos atribuir a borda uma cor.</p>
<h3>Para determinar a largura da borda</h3>	<p>Para determinar a largura da borda podemos utilizar a propriedade border-width. Ao utilizar essa propriedade, podemos definir individualmente as bordas superior, esquerda, inferior e direita de um elemento:</p>	<pre>div { border-width: 1px 2px 1px 2px; }</pre> <p>atribuímos 1 pixel para a parte superior/inferior e 2 pixels para a borda esquerda/direita do elemento.</p>
<h3>Definir o valor dessa propriedade</h3>	<p>Podemos ainda utilizar palavras chave para definir o valor dessa propriedade, sendo elas thin, medium e thick.</p>	<pre>div { border-width: thin; }</pre> <p>No exemplo acima, o elemento terá uma borda fina aplicada.</p>
<h3>Cancelar os valores anteriormente</h3>	<p>Essa propriedade pode receber ainda o valor unset, que cancela os valores anteriormente aplicados.</p>	<p>unset</p>
<h3>Determinamos como a borda do elemento será apresentada</h3>	<p>Com a propriedade border-style determinamos como a borda do elemento será apresentada. Ela pode ser sólida com o valor solid, pontilhada com o valor dotted ou mesmo dupla, se aplicado o valor dashed.</p>	<pre>div { border-style: dotted solid; }</pre> <p>Nesse exemplo, o elemento receberá uma borda superior/inferior pontilhada e esquerda/direita sólida. Podemos ainda utilizar o valor none, que cancela qualquer outro anteriormente aplicado.</p>

7- Bordas Arredondadas

Borda Arredondada

Border-Radius



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

<h3>Border Radius - Exemplo 1</h3>	<pre> 06 border-bottom-left-radius: ;valor; 07 border-bottom-right-radius: valor </pre>	<p>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;</p> <p>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.</p> <p>Linhas 4 a 7: Aqui estamos atribuindo os valores específicos de cada vértice</p>
<h3>Border Radius - Exemplo 2</h3>	<pre> 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 23 30px; 24 } 25 </style> 26 </head> 27 <div id="caixa1"></div> 28 <div id="caixa2"></div> 29 <div id="caixa3"></div> 30 </body> 31 </html> </pre>	<p>Linhas 7 a 14: configuramos a aparência básica de todas as divs do exemplo (dimensões e borda);</p> <p>Linhas 15 a 17: configuramos a primeira div para apresentar a borda arredondada com 20px em todas as extremidades;</p> <p>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;</p> <p>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.</p>

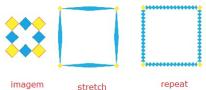


Sombreamento Box-Shadow

<h3>Box-Shadow</h3>	<p>sombreamento. Esta propriedade é composta por alguns parâmetros e sua sintaxe de uso é a seguinte:</p> <ul style="list-style-type: none"> • 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 	<pre> <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> </pre> 
---------------------	--	--

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

Border Image

<h3>Border-Image</h3>	<p>Definir uma imagem para ser usada no preenchimento da borda de um elemento. Infelizmente esta propriedade não é suportada pelo Internet Explorer e requer o uso de prefixos para os diferentes browsers</p>	<pre><html> <head> <title>Trabalhando com Bordas em CSS3</title> <style type="text/css"> #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 */ } </style> </head> <body> <div id="div1"/> </body> </html></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 border-radius seja preenchido com um valor igual à altura ou largura desse elemento (que por ser quadrado, serão iguais). O formato circular tem sido muito utilizado, por exemplo, em fotos de perfil e galerias de imagens.</p>	<pre> img { width: 200px; height: 200px; } #img_circular { border-radius: 200px; }</pre>	
-------------------------------	---	---	--

8- Botões arredondados

CRIAR Botão CSS

```
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 }
```

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;



Linhos 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**.

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**.

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

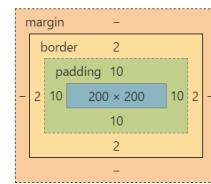
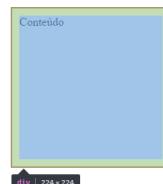
Box Sizing

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;  
}
```

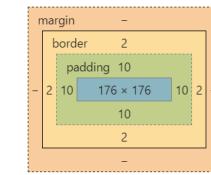
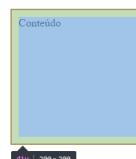


Box Sizing

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;  
}
```



CONTACT



Brunna Croches

Developer Full Stack



brunnacroches.dev



linkedin.com/brunnacroches



github.com/brunnacroches



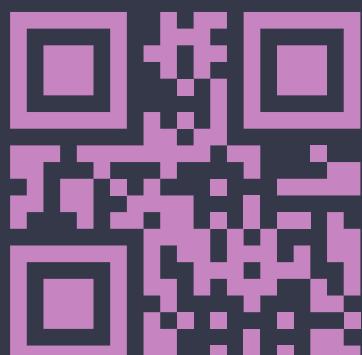
@brunnacroches.dev



discord.com/brunnacroches



brunnacroches@gmail.com



let's share