

# Modelo de Caixa

DigitalHouse>



**Certified Tech  
Developer**

The Ultimate Degree

# Índice

1. [Modelo de caixa](#)
2. [Width e Height](#)
3. [Padding](#)
4. [Border](#)
5. [Margin](#)
6. [Box-sizing](#)

# 1 | Modelo de Caixa

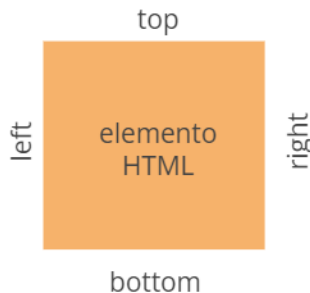
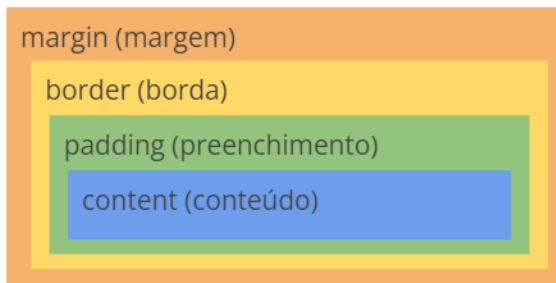


Em HTML, todos os elementos são representados por **caixas**. Cada caixa é composta por **conteúdo**, **preenchimento**, **borda** e **margem**.

# Modelo de caixa

Através do CSS, podemos manipular todas essas propriedades para mudar a aparência de cada elemento. Por sua vez, essas propriedades podem ser aplicadas de maneira diferente aos quatro lados de cada caixa (top, right, bottom e left).

As propriedades do modelo de caixa se aplicam apenas a elementos de bloco e semi-bloco.



# 2 | Width e Height



# Height

Se um elemento não tiver a propriedade height declarada no css, a altura será igual à altura do seu conteúdo interno. Seja um elemento de bloco ou de linha.

Para atribuir valor a esta propriedade, podemos fazer isso usando a medida de porcentagens (%) ou pixels (px).

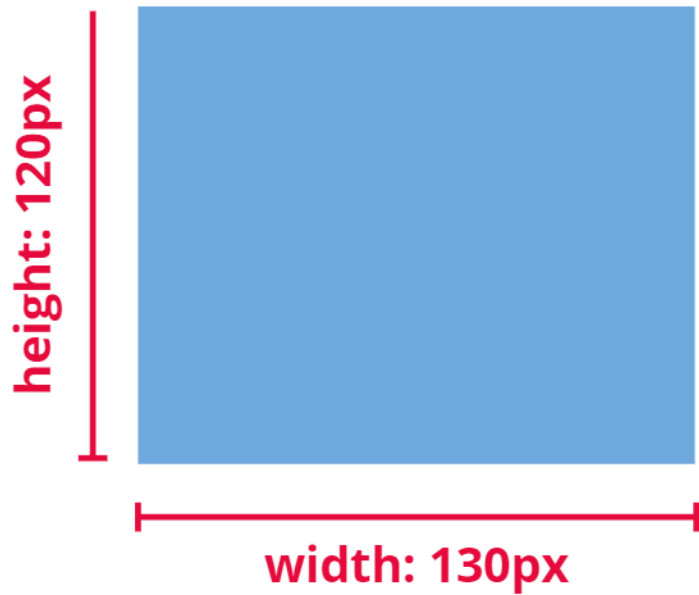
Para a altura dos elementos, é recomendado utilizar porcentagem.

css

```
div {  
    height: 130px;  
}
```



## Propriedades Height e Width no elemento:



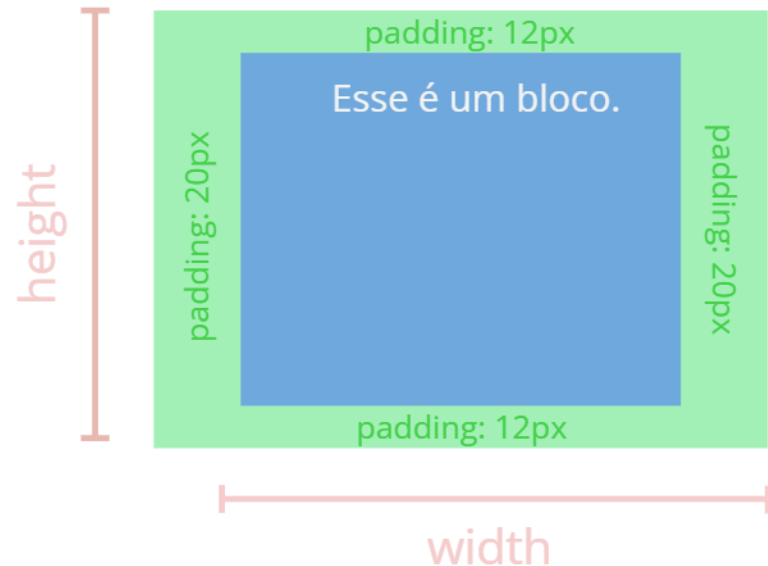
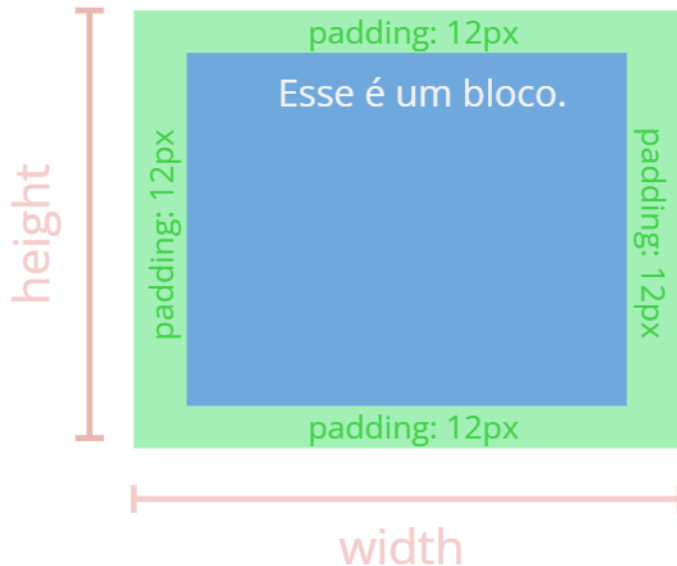
css

```
div {  
    background-color: blue;  
    width: 130px;  
    height: 120px;  
}
```

# 3 | Padding

# Padding

É o espaço de **preenchimento** que podemos agregar **entre** o conteúdo do elemento e sua **borda**. Em outras palavras, é a margem interna do elemento.



# Padding

Podemos atribuir valor a esta propriedade usando a medida em pixels (px), indicando 1 valor para os 4 lados da caixa.

css

```
div { padding: 12px; }
```



## Único valor

Isso quer dizer que a margem interna de 12px será aplicada em todos os lados.

# Padding

Também podemos fazer com **2 valores**. O primeiro valor indica o padding de **cima** e de **baixo**, e o segundo indicará o padding da **esquerda** e da **direita**.

css

```
div { padding: 12px 20px; }
```



## Dois valores

**12px** de padding para cima e para baixo.

**20px** de padding para esquerda e para direita.

# Padding

Também podemos fazer com **3 valores**. O primeiro indica o padding de **cima**, o segundo da **esquerda** e da **direita** e o terceiro será o de **baixo**.

css

```
div { padding: 12px 20px 18px; }
```



## Três valores

**12px** de padding para cima.

**20px** de padding para esquerda e direita.

**18px** de padding para baixo.

# Padding

Por último, podemos utilizar 4 valores que representarão os quatro lados individualmente. Começaremos pelo valor de cima e seguiremos em sentido horário (direita, embaixo, esquerda).

css

```
div { padding: 12px 15px 18px 21px; }
```

## Quatro valores

**12px** de padding para cima.

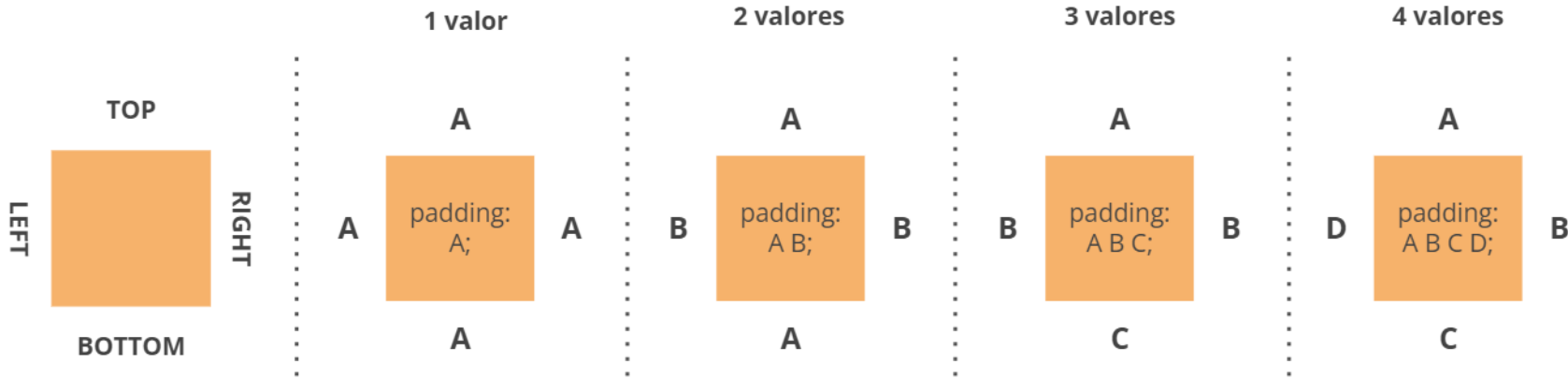
**15px** de padding para direita.

**18px** de padding para baixo.

**21px** de padding para esquerda.

# Valores para Propriedades de Estilo

Em todas as propriedades de estilo, como margin e padding, podemos escrever 1, 2, 3 ou 4 valores.

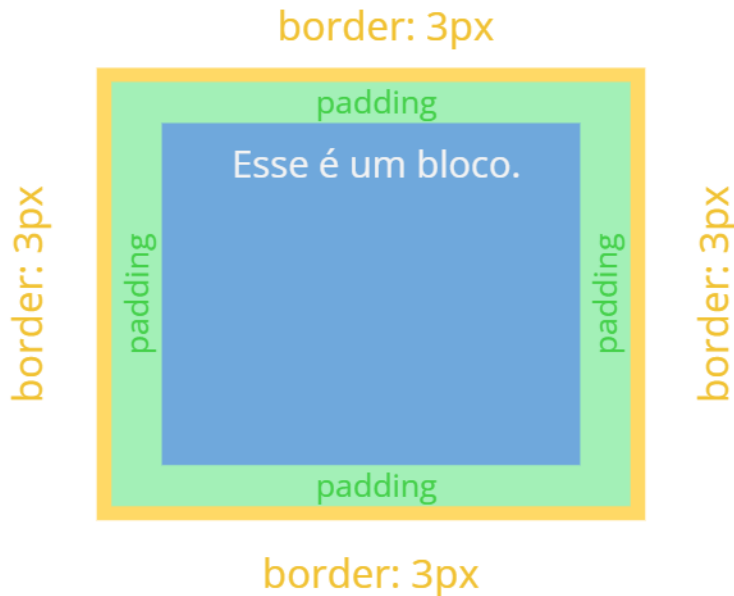




# 4 | Border

# Border

Refere-se à borda do elemento. Fica entre o conteúdo e a margem.



# Border

Para atribuir valor a essa propriedade, fazemos isso definindo o estilo da linha, seu tamanho e sua cor.

O estilo da linha pode ser `solid`, `dotted`, `dashed` ou `double`.

A espessura da linha pode ser qualquer unidade de medida de CSS.

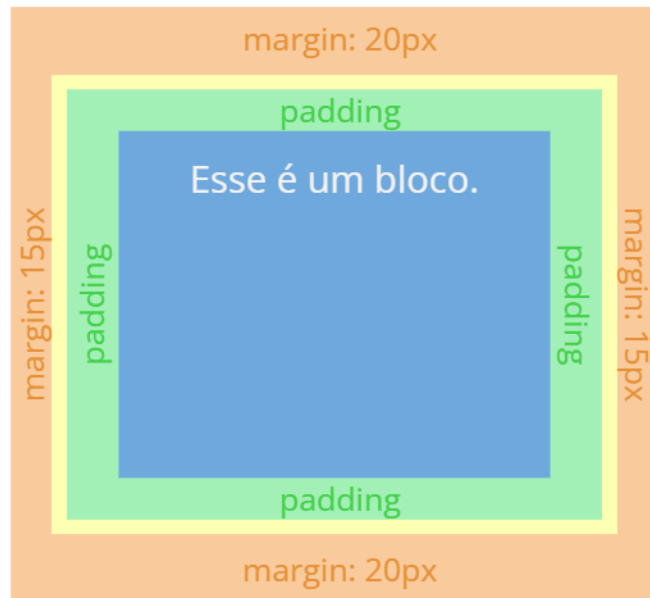
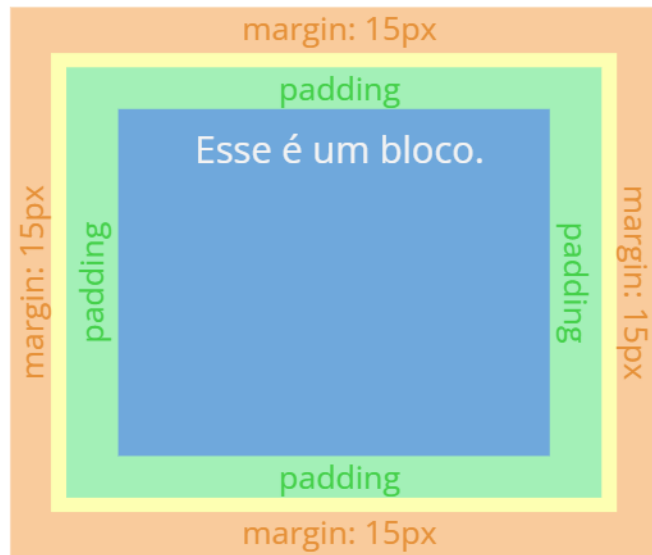
A cor pode ser qualquer cor válida de CSS.



# 5 | Margin

# Margin

Refere-se à **margem exterior** do elemento. Serve para separar uma caixa da outra.



# Margin

Para atribuir valor a essa propriedade, podemos usar a medida em pixels (px), indicando 1 valor para os 4 lados da caixa.

css

```
div { margin: 15px }
```



## Único valor

Isso quer dizer que a margem externa de **15px** será aplicada em todos os lados.

# Margin

Da mesma maneira que padding e border, podemos também declarar 2, 3 e até 4 valores para a propriedade margin.

```
css  div { margin: 20px 15px }
```

```
css  div { margin: 20px 15px 30px }
```

```
css  div { margin: 20px 15px 30px 25px }
```

# 6 | Box-Sizing



# Box-sizing

Para poder calcular a largura e altura total de um elemento, temos que somar todas as propriedades que vimos antes.

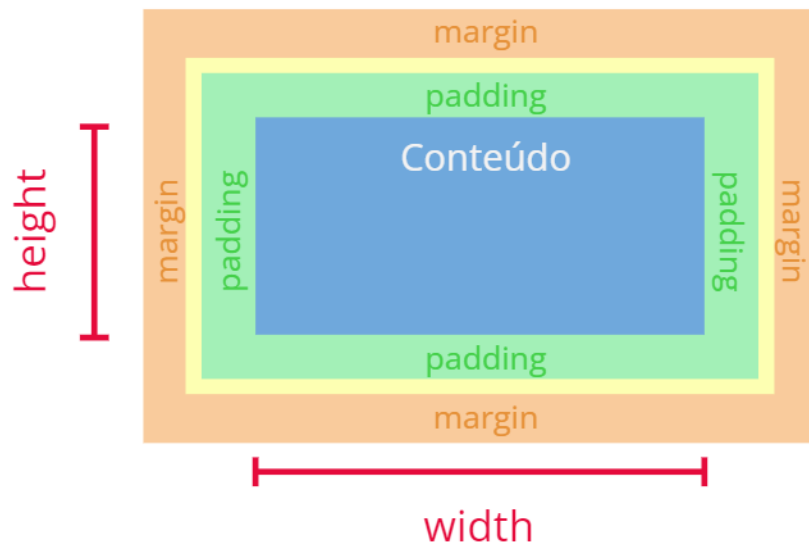
Como podem imaginar, fazer esse cálculo para saber quanto espaço um elemento vai ocupar no total é bastante trabalhoso.

Por isso, a propriedade `box-sizing` nos ajuda com esse cálculo.



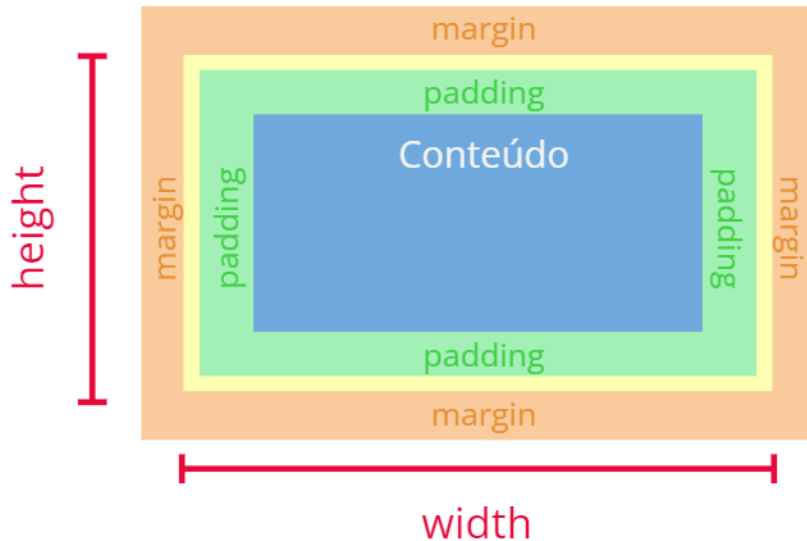
# Content-Box

Por padrão, seu valor será `content-box`, e o comportamento será de aplicar a largura e altura que definirmos ao conteúdo do elemento.

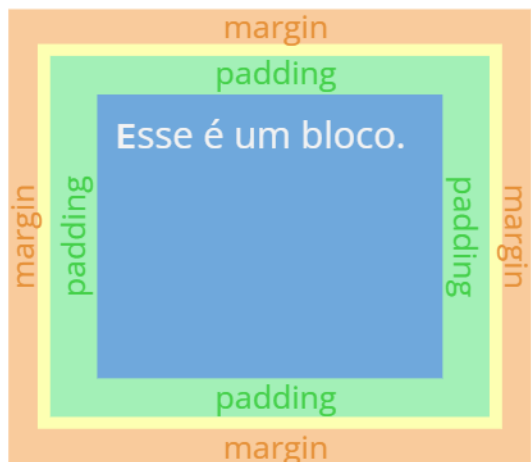


# Border-Box

Se usarmos o valor `border-box`, a altura e largura que indicarmos abrangerá o conteúdo, padding e borda do elemento, deixando apenas a margem de fora.



## Box-sizing: border-box no css



```
div {  
    width: 150px;  
    height: 130px;  
    padding: 12px;  
    border: solid 3px yellow;  
    margin: 15px;  
    box-sizing: border-box;  
}
```



É uma prática muito comum aplicar

**box-sizing: border-box**

a todos os elementos do site com a seguinte linha de código:

```
* { box-sizing: border-box }
```





**Digital**House>