

## 1 Box Model

Tudo em CSS tem um quadro em torno de si, e entender estes quadros é chave para ser capaz de criar layouts com CSS, ou para alinhar itens com outros itens.

### 1.1 Block x inline

Geralmente temos dois tipos de caixas nos quais os elementos são encaixado: *block* e *inline*. Cada uma possui suas características e afetam como os elementos se comportam.

Se uma caixa é do tipo *block*, ela terá o seguinte

- comportamento:
- A caixa irá gerar uma nova linha;
  - Na maioria dos casos irá ocupar todo o espaço disponível;
  - A largura e altura são respeitadas.

Elementos como `<h1>` e `<p>` são do tipo *block* por padrão.

Se uma caixa é do tipo *inline*, ela terá o seguinte

- comportamento:
- A caixa não irá gerar uma nova linha;
  - As propriedades de largura e altura não serão aplicadas.

Elementos como `<a>`, `<span>` e etc são do tipo *inline* por padrão. O tipo de caixa aplicado depende da propriedade ***display***.

### 1.2 Afinal, o que é o Box Model?

O modelo completo de Box Model se aplica principalmente em elementos do tipo *block*. Elementos do tipo *inline* usam apenas alguns dos comportamentos definidos pelo box model.

O box model define como diferentes partes de uma caixa (margem, borda, preenchimento e conteúdo) irão funcionar em conjunto para criar a caixa que vemos na tela.

#### 1.2.1 Partes de uma caixa

Dentro de uma caixa CSS nós temos os seguintes elementos:

- **Content box:** Área onde o conteúdo é exibido. É controlado pelas propriedades *height* e *width*;
- **Padding box:** Área ao redor do conteúdo como espaço em branco. É controlado pela propriedade *padding*;
- **Border box:** Envolve o conteúdo e o preenchimento. É controlado pela

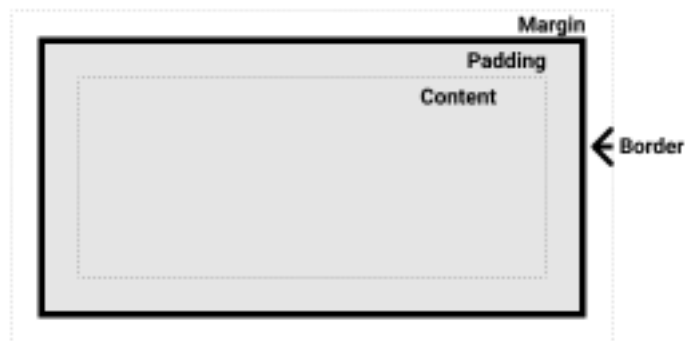
propriedade *border*;

1

Unidade III: Box Model soulcode.com

- **Margin box:** É a camada mais externa da caixa, envolve o conteúdo, o preenchimento e a borda. É um espaço em branco entre a caixa e outros elementos. É controlado pela propriedade *margin*.

Figura 1: Box Model



Fonte: MDN Web Docs.

### 1.3 Box model padrão

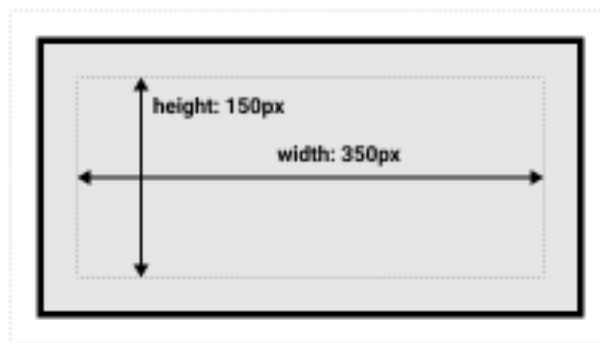
No modelo padrão de caixa, se você define altura e largura a uma caixa isso definirá as dimensões da content box. Qualquer preenchimento e borda aumentam a largura e altura final da caixa. Podemos ver isso a seguir:

Listagem 1: Uso das propriedades do box model

```
1 .box {  
2 width: 350px;  
3 height: 150px;  
4 margin: 10px;  
5 padding: 25px;  
6 border: 5px solid black;  
7 }
```

O espaço ocupado pela caixa usando o box model padrão será de fato 410px ( $350 + 25 + 25 + 5 + 5$ ) e a altura será 210px ( $150 + 25 + 25 + 5 + 5$ ). Já que a borda e preenchimento aumentam a largura e altura final. Podemos ver isso na imagem a seguir:

Figura 2: Box Model



Fonte: MDN Web Docs.

A área final da caixa é afetada pelo preenchimento e pela borda, mas não pela margem, pois a margem é algo externo a caixa, e afeta apenas os elementos em volta.

### 1.3.1 Alterando o dimensionamento da caixa

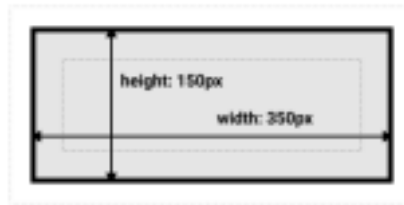
Pode ser inconveniente para seu layout ter dimensionamentos não-calculados. Você define uma altura e largura para uma <div>, aplica margens, preenchimentos e bordas e percebe ao final que a div agora possui outra dimensão diferente. Podemos mudar este comportamento, alterando como o box model calcula essa situação.

Listagem 2: Propriedade box-sizing

```
1 .box {  
2   box-sizing: border-box;  
3 }
```

Dessa forma, o preenchimento e borda não aumentam o tamanho da caixa, e passam a respeitar a largura e altura definidas para a content-box.

Figura 3: Box Model



Fonte: MDN Web Docs.

3

Unidade III: Box Model soulcode.com

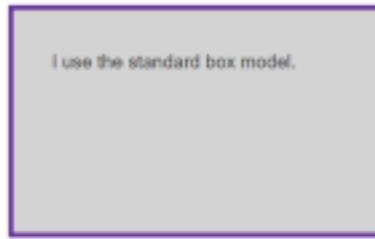
Caso queira que todos os elementos já tenham este comportamento por padrão, você pode aplicar essa propriedade em um seletor universal (\*), :root ou html.

Aqui neste exemplo abaixo, podemos ver claramente a diferença entre usar o comportamento padrão e usar o *box-sizing: border-box*.

Listagem 3: Diferença do box-sizing: border-box

```
1 .box {  
2 border: 5px solid rebeccapurple;  
3 background-color: lightgray;  
4 padding: 40px;  
5 margin: 40px;  
6 width: 300px;  
7 height: 150px;  
8 }  
9  
10 .alternate {  
11 box-sizing: border-box;  
12 }  
13  
14  
15 ...  
16  
17 <div class="box">I use the standard box model.</div>  
18 <div class="box alternate">I use the alternate box model.</div>
```

Figura 4: Diferença entre usar o padrão e usar o box-sizing:



`border-box;`

Fonte: MDN Web Docs.

**Dica:** Após ler as seções abaixo, recomendo voltar para esta seção e concretizar seu entendimento sobre o comportamento do Box Model.

#### 4

Unidade III: Box Model soulcode.com

## 1.4 Margens

As margens são espaços invisíveis ao redor da caixa. Empurra outros elementos ao redor da caixa. O valor da margem pode ser positivo ou negativo. Não afetam o tamanho da caixa visível do elemento. Podemos controlar a margem do elemento de diversas formas:

- `margin: 10px;` Todos os lados da caixa ficam com 10px de margem;
- `margin: 10px 20px;` Os lados da horizontal (`left`, `right`) recebem 20px e os lados da vertical (`top`, `bottom`) recebem 10px;
- `margin: 10px 20px 30px 40px;` Aplica margem separadamente em cada lado, seguindo o sentido horário (`top`, `right`, `bottom`, `left`);
- `margin-*`: Podemos usar uma propriedade separada para aplicar.
  - `margin-top`: Margem no topo;
  - `margin-right`: Margem na direita;
  - `margin-bottom`: Margem no fundo;
  - `margin-left`: Margem na esquerda.

## 1.5 Bordas

A borda é desenhada entre a margem e o preenchimento da caixa. Caso esteja sendo utilizado o comportamento padrão do box model, a dimensão da borda é adicionada à largura e altura da caixa.

Caso esteja usando box-sizing: border-box o tamanho da borda faz o conteúdo ficar menor e a altura e largura são mantidas.

Podemos controlar a borda de diversas formas:

- border: <width> <style> <color>: Esta propriedade é um atalho para as propriedades: *border-width*, *border-style* e *border-color*;
- border: 1px solid black: Todos os lados da caixa terão a borda descrita: 1px de largura, estilo sólido e cor black.
- border-\*: Podemos definir a borda de cada lado da caixa separadamente.
  - border-top: 1px solid red: Aplica esta borda apenas para o topo da caixa;
  - border-right: 1px solid orange: Aplica esta borda apenas para o lado direito da caixa;
  - border-bottom: 1px solid green: Aplica esta borda apenas para o fundo da caixa;

## 5

Unidade III: Box Model soulcode.com

- border-left: 1px solid yellow: Aplica esta borda apenas para o lado esquerdo da caixa;

Para saber mais possibilidades da propriedade *border-style*, leia esta [documentação](#) do MDN Web Docs.

## 1.6 Preenchimentos

O preenchimento fica entre a borda e o conteúdo da caixa. Diferente da margem, não é possível ter valores negativos para preenchimento. Podemos controlar o preenchimento de diversas formas:

- padding: 10px: Todos os lados da caixa ficam com 10px de preenchimento;
- padding: 10px 20px: Os lados da horizontal (left, right) recebem 20px e os lados da vertical (top, bottom) recebem 10px;
- padding: 10px 20px 30px 40px: Aplica preenchimento separadamente em cada lado, seguindo o sentido horário (top, right, bottom, left);
- padding-\*: Podemos usar uma propriedade separada para aplicar.

- padding-top: Preenchimento no topo;
- padding-right: Preenchimento na direita;
- padding-bottom: Preenchimento no fundo;
- padding-left: Preenchimento na esquerda.

## 2 Mais sobre Box Model

Caso queira aprender detalhes mais específicos sobre Box Model, recomendo ler a [documentação](#) do MDN Web Docs sobre CSS. Lá são abordados diversos tópicos mais avançados sobre este tema.