

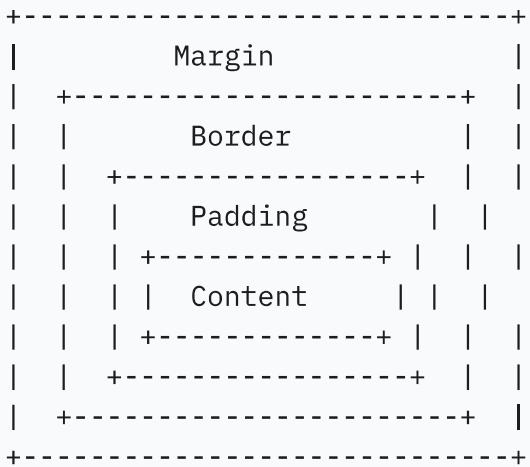
Box-Model

Box-Model

O **Box Model** é a maneira como o CSS **organiza e desenha** os elementos na página.

Imagine cada elemento HTML como uma **caixinha** com camadas que vão de dentro pra fora:

Estrutura do Box Model



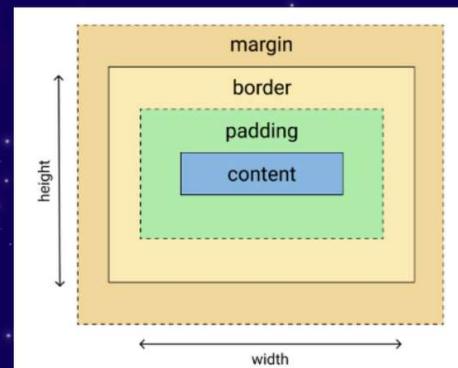
- **Margin** (margem)
- **Border** (borda)
- **Padding** (preenchimento)
- **Content** (conteúdo)



BOX MODEL

O Box Model é fundamental para fazer layouts para web porque ele vai te dar maior facilidade na hora de aplicar o CSS. Ao entender os conceitos do Box Model muitas questões do CSS começam a fazer sentido.

- CADA ELEMENTO É REPRESENTADO COMO UMA CAIXA RETANGULAR
- ESSA CAIXA POSSUI PROPRIEDADES DE UMA CAIXA EM 2 DIMENSÕES (LARGURA X ALTURA)



Extra

[Documentação box model ↗](#)

Entenda o Box Model - Curso de HTML e CSS - Aula 9



Content (Conteúdo)

É onde o texto ou outros elementos aparecem.

Exemplo: a área onde vemos um parágrafo, uma imagem ou botão.

É a parte visível e interativa do elemento.

```
<section>
    <p>Conteúdo</p>
</section>
```

Extra

[Documentação content ↗](#)

Padding (Preenchimento interno)

Espaço **interno** entre o conteúdo e a borda.

Ele "empurra" o conteúdo **pra dentro**, aumentando a área visível da caixa.

Serve para dar "respiro" visual dentro da caixa.

```
padding: 20px;
```

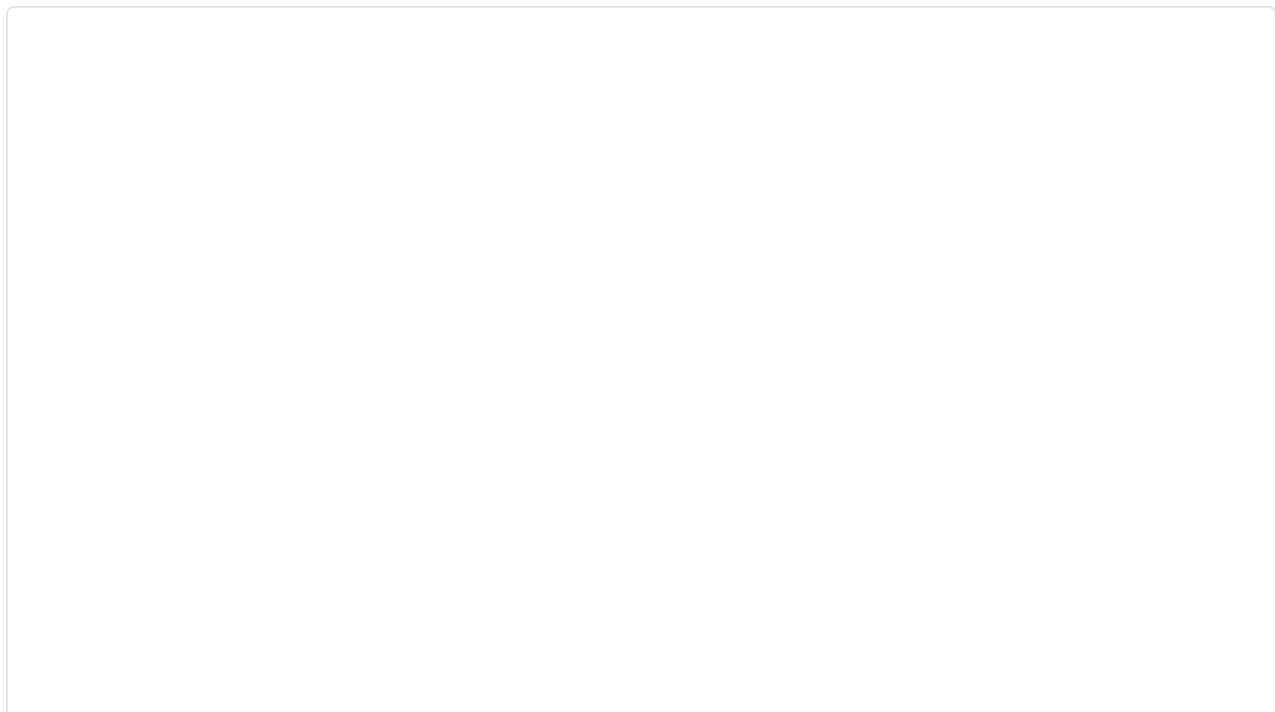
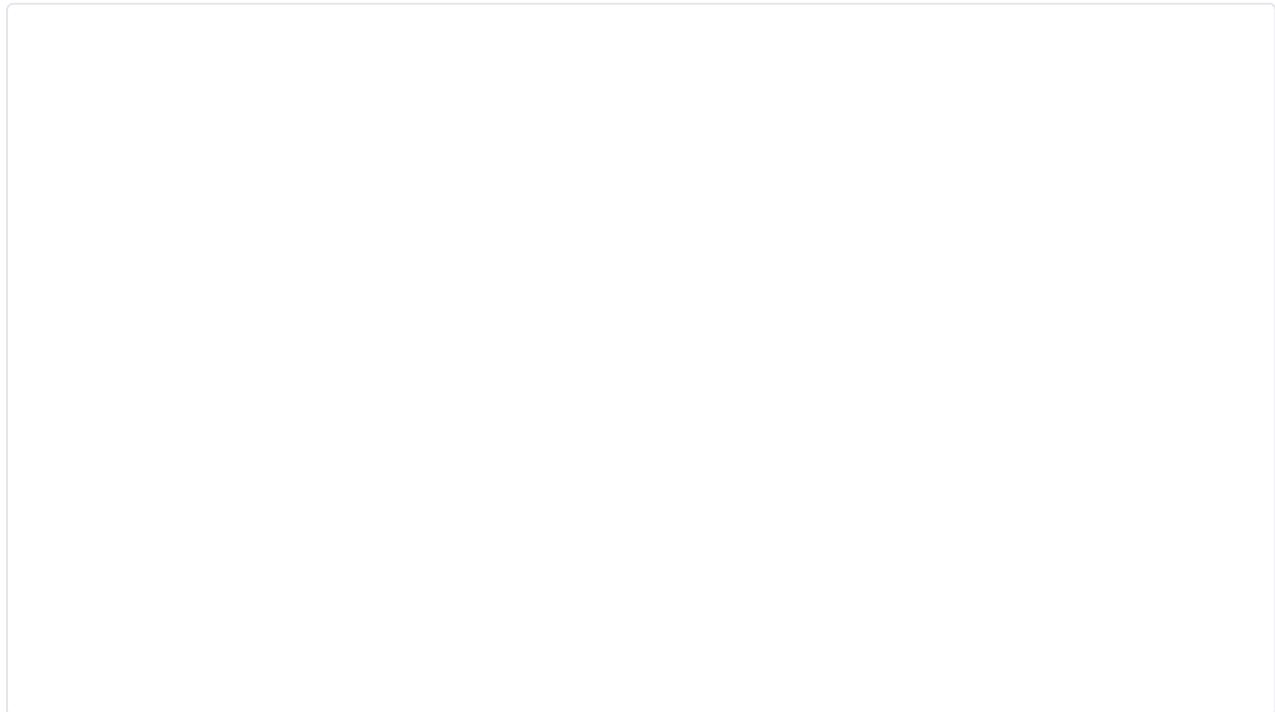
Atenção! O padding aumenta o tamanho total da caixa se estiver usando box-sizing: content-box.

Pode ser definido individualmente:

```
padding-top: 10px;  
padding-right: 15px;  
padding-bottom: 10px;  
padding-left: 15px;
```

Extra

Documentação padding ↗



Border (Borda)

Contorna o conteúdo e o padding.

Pode ter estilo, cor e espessura.

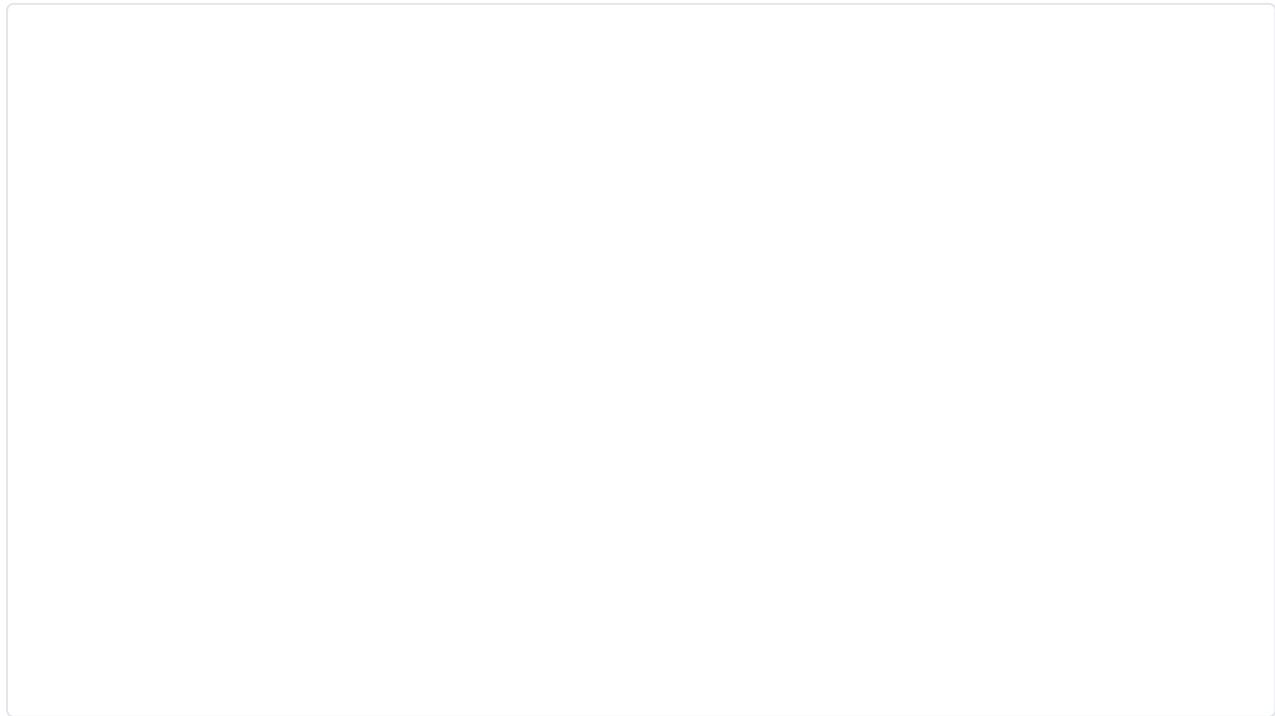
```
border: 2px solid #000;
```

Estilos possíveis:

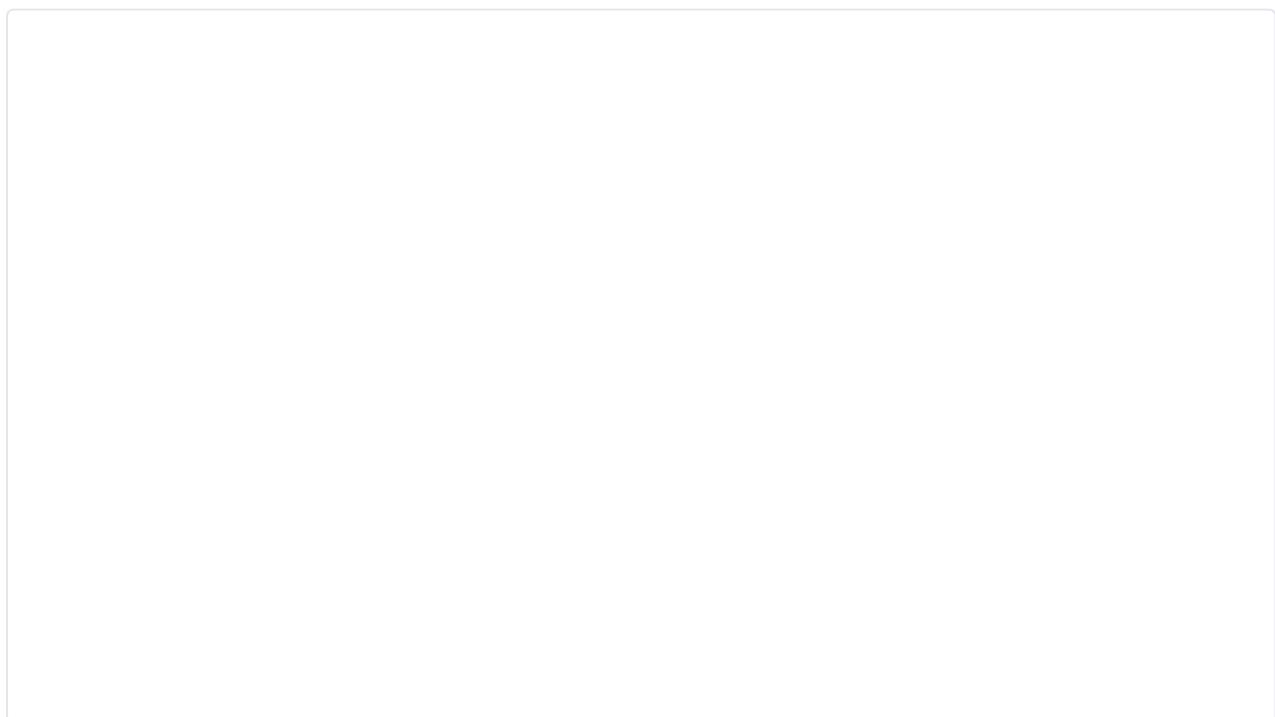
- solid – contínua
- dashed – tracejada
- dotted – pontilhada
- double - Exibe duas linhas retas
- groove - Exibe uma borda com aparência esculpida
- ridge - Exibe uma borda com aparência extrudada.
- inset - Exibe uma borda que faz o elemento parecer incorporado.
- outset - Exibe uma borda que faz o elemento parecer em relevo.

Extra

Tipos de bordas ↗



Bordas personalizadas



Margin (Margem)

Espaço entre a borda do elemento e os elementos ao redor.

Afasta o elemento de outros ao seu redor.

```
margin: 20px;
```

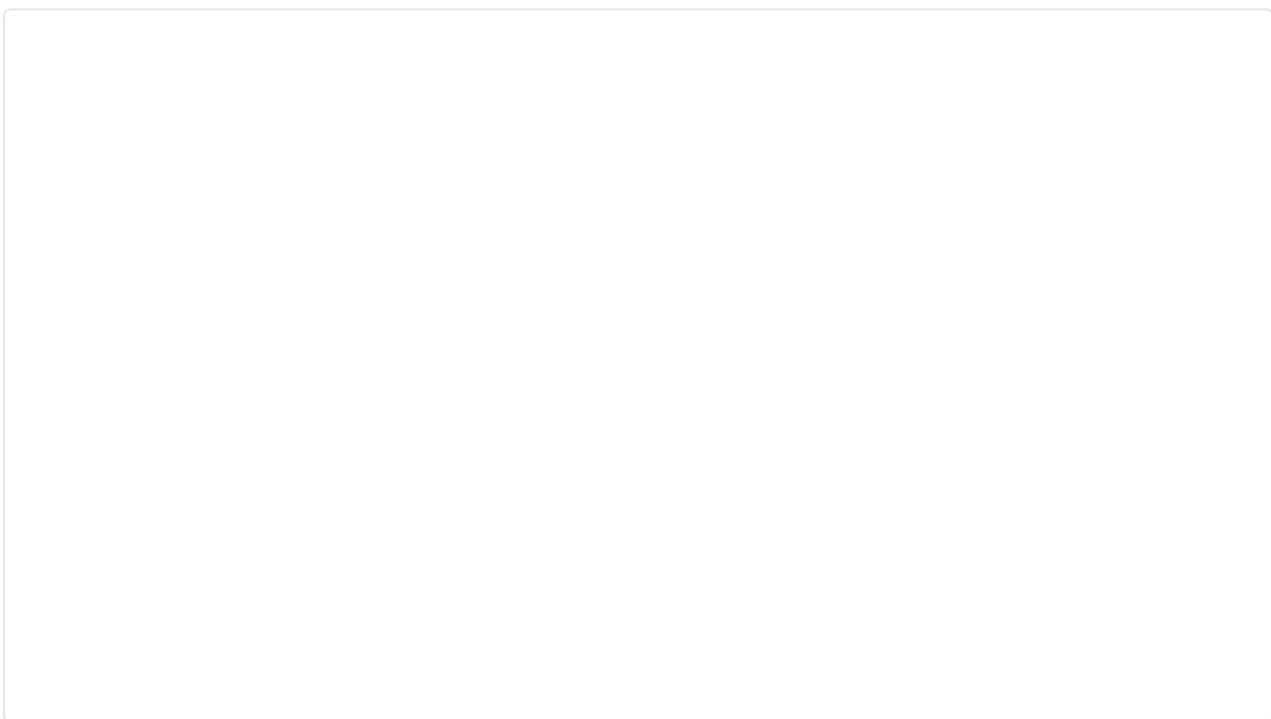
Também pode ser configurado individualmente:

```
margin-top: 10px;  
margin-bottom: 20px;
```

Dica:

margin: auto; é usado para centralizar elementos horizontalmente (com display: block e width definida).

Extra



Medidas no CSS

As medidas definem o **tamanho** da caixa (largura, altura, espaçamento etc.).

Elas podem ser:

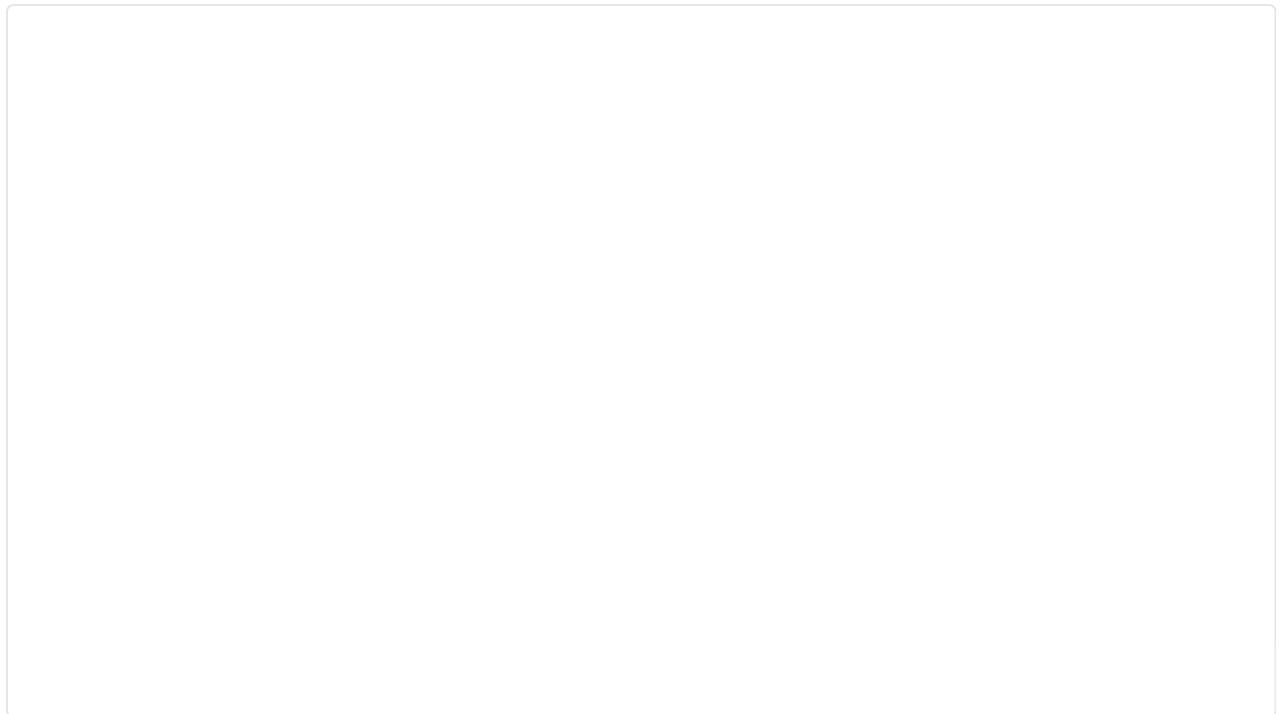
Medida Absolutas

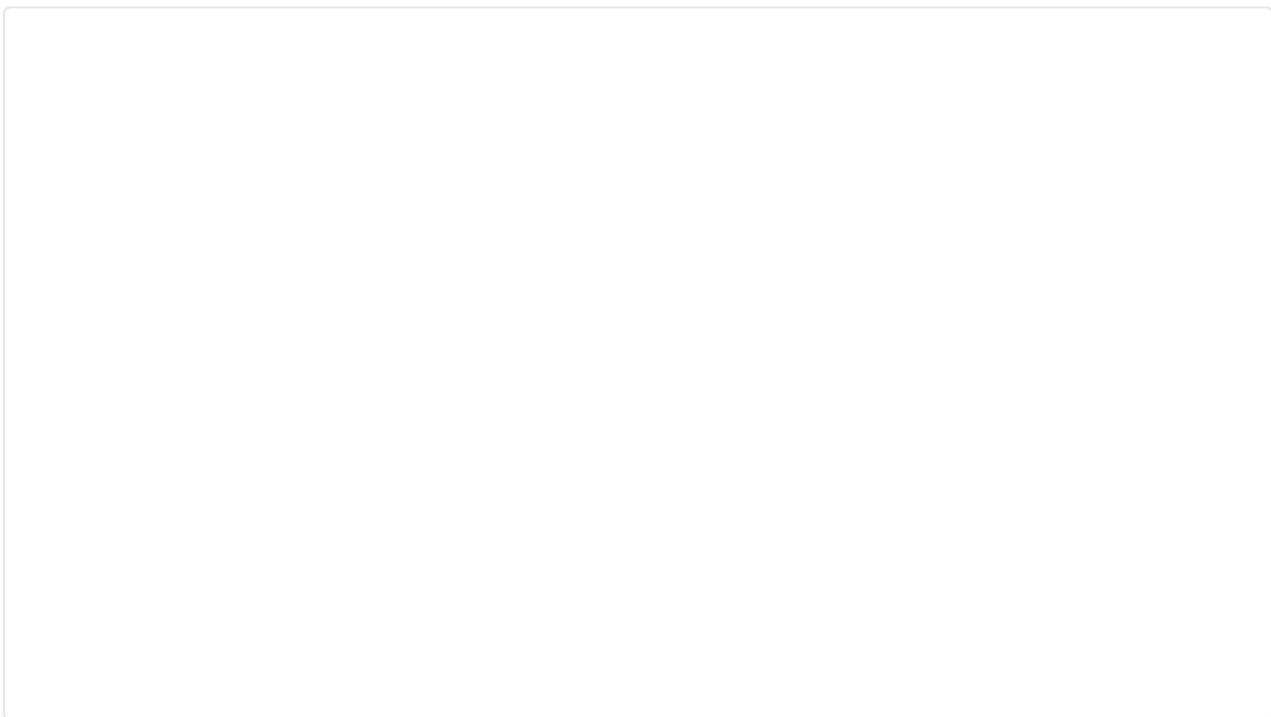
- px (pixel)

Medidas Relativas

- % – baseado no elemento pai
- em – baseado no tamanho da fonte do **elemento atual**
- rem – baseado na fonte do **elemento raiz (html)**
- vw – porcentagem da largura do viewport
- vh – porcentagem da altura do viewport

Extra





Box-Sizing

Essa propriedade define como o navegador deve calcular a **largura e altura** da caixa.

content-box (valor padrão)

A width e height consideram **apenas o conteúdo**.

O padding e border **são adicionados por fora**.

border-box

A width e height já **incluem o conteúdo, o padding e a border**.

Ajuda muito no layout, pois evita "surpresas" de tamanhos.

```
box-sizing: border-box;
```

Extra

Display inline

O que é display: inline?

O display: inline é o **comportamento padrão de elementos de texto**, como , <a>, , etc.

Ele não quebra a linha

Fica **alinhado com o texto ou outros elementos inline**

Não permite mudar altura, largura, nem aplicar margin e padding vertical de forma eficiente

Como pensar no inline?

Pense nele como palavras em uma frase:

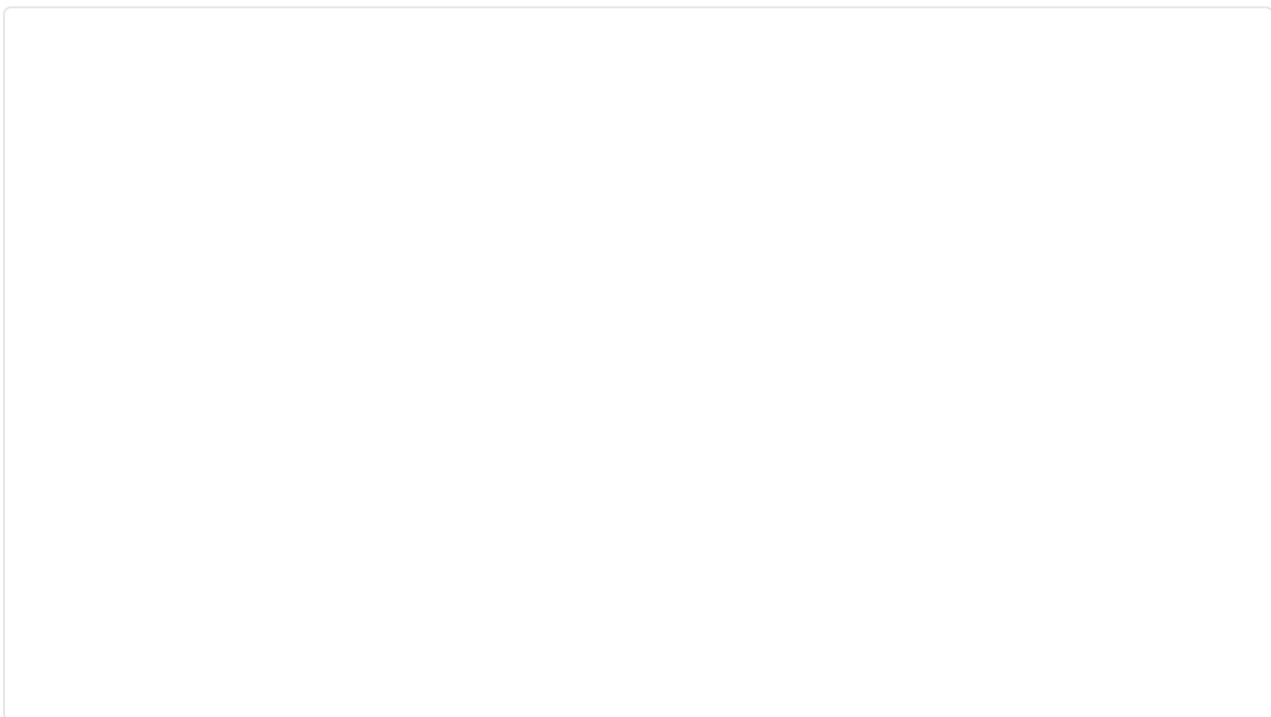
```
<p>O <span>HTML</span> é divertido!</p>
```

A palavra "HTML" com a tag está **na mesma linha**, junto com o restante do texto.

Limitações do inline:

| Característica | Comportamento no inline |
|-------------------------------|-------------------------|
| Quebra de linha | Não |
| Largura personalizada (width) | Não funciona |
| Altura personalizada (height) | Não funciona |
| margin e padding vertical | Limitado |
| margin e padding horizontal | Funciona |

Extra



Display Block

O que é display: block?

O display: block é o comportamento padrão de muitas **tags estruturais**, como:

- <div>
- <p>
- <h1> a <h6>
- <section>, <header>, <footer>, etc.

Como funciona?

```
<div class="bloco">Eu sou um bloco!</div>
```

```
.bloco {  
    background-color: lightcoral;  
    padding: 10px;  
    width: 300px;  
    height: 100px;  
}
```

Resultado: O elemento **começa em uma nova linha**, pode ter altura e largura, e você controla todo o espaço dele.

Características do block:

| Característica | Comportamento no block |
|---|------------------------|
| Ocupa toda a linha | Sim |
| Quebra de linha | Sempre |
| Define width/height | Sim |
| Aceita margin/padding vertical e horizontal | Sim |

Quando usar block?

Use quando quiser criar **áreas maiores e independentes** na sua página, como:

- Seções
- Blocos de texto
- Containers para outros elementos
- Cartões, banners, menus verticais

Comparando com outros displays:

| Tipo | Quebra de linha? | Pode definir tamanho? | Fica lado a lado? |
|--------------|------------------|-----------------------|-------------------|
| inline | Não | Não | Sim |
| inline-block | Não | Sim | Sim |
| block | Sim | Sim | Não |

Display inline-block

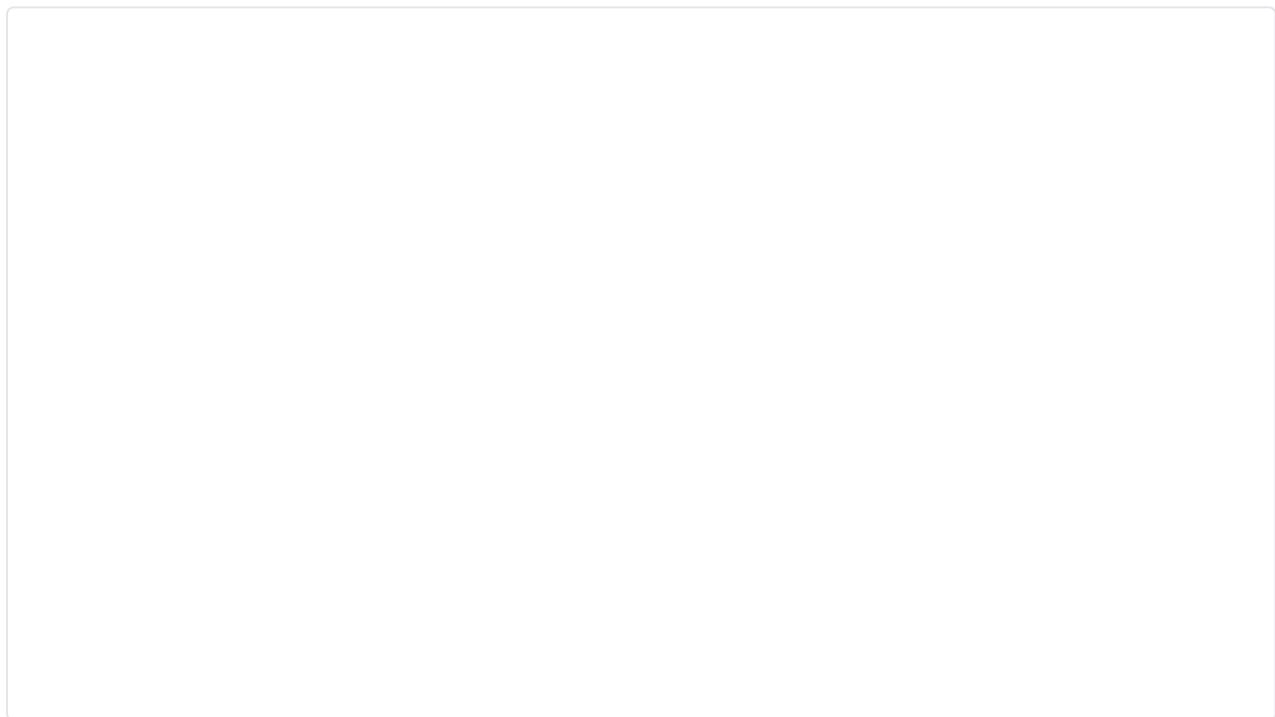
O que é display: inline-block?

O inline-block é uma mistura do melhor dos dois mundos:

- Ele **fica na mesma linha** dos outros elementos (como o inline)
- Mas **permite aplicar largura, altura, padding e margin vertical** (como o block)

Quando usar?

Use inline-block quando você quiser **colocar elementos lado a lado** (como botões, caixas ou imagens), **mas ainda controlar o tamanho e espaçamento deles**.



Diferença entre inline, block e inline-block

| Tipo | Alinha na mesma linha? | Pode definir largura e altura? |
|--------------|------------------------|--------------------------------|
| inline | Sim | Não |
| block | Não (quebra linha) | Sim |
| inline-block | Sim | Sim |

Curiosidade

Antigamente, antes do Flexbox e Grid, inline-block era a forma mais usada para **alinear elementos horizontalmente!**