



# Caixas Flexíveis (Flex/Flexbox)



**Certified  
Developer**  
The Ultimate Tech Degree

**DigitalHouse** >  
Coding School



# Temas

**1**

O que é

**2**

Aplicação

**1**

**0 que é Flexbox?**



**Objetivo**

**Comportamento  
utilizando Flex**



**O que é**

**Comportamento  
Padrão**

**Revisão e  
Conclusão**

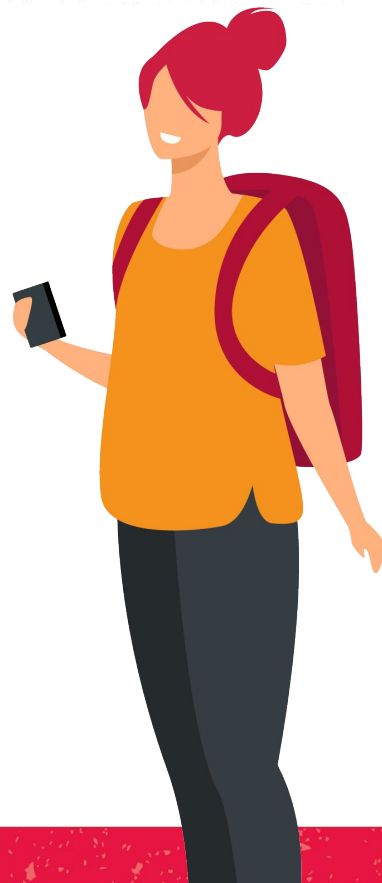
**1** | **0 que é**



## O que é?

Flexbox é um modelo de layout unidimensional que podemos utilizar para organizar, alinhar e agrupar os nossos elementos de uma melhor forma.

Dessa maneira podemos utilizar elementos uns aos lados dos outros como na maioria dos sites



## 2 | Objetivo



## Objetivo

Quando pensamos em flexbox sempre temos que ter em mente que iremos estilizar o elemento pai para que ele trate os seus elementos filhos de uma forma diferente.

## Requisitos

- Um elemento pai
- Alguns elementos filhos





**3**

**Comportamentos**



## Comportamento padrão

Elementos como `<div>` e `<section>` sempre serão exibidos um abaixo do outro.

Isso ocorre por conta desses elementos serem 'Elementos em Bloco' aonde independente do tamanho ele irá ocupar uma linha inteira.

Mas podemos mudar esse comportamento...

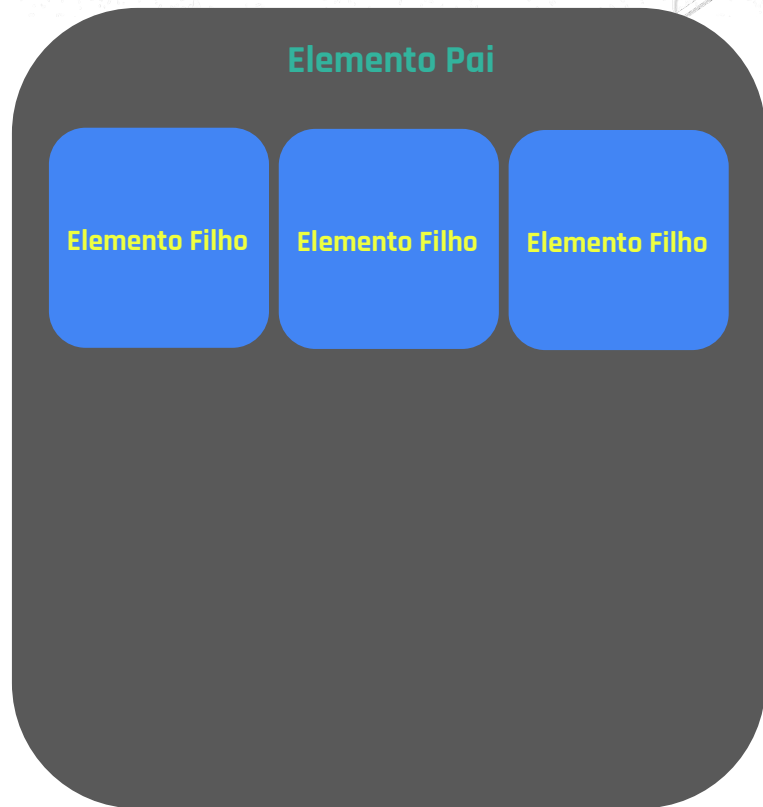




## Comportamento utilizando Flex

Com a utilização do Flex nós podemos organizar nossos elementos de uma maneira diferente que o HTML organiza por padrão.

É possível agora agruparmos em uma mesma linha os elementos filhos que antes estavam um a baixo dos outros.



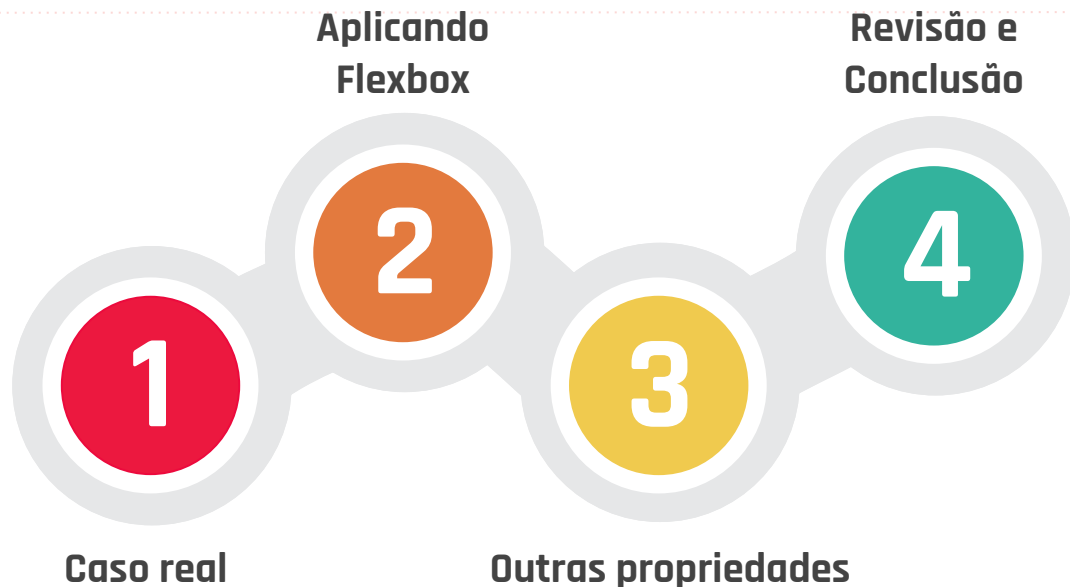
**Hora da revisão**



## Conclusão

- Vimos que é possível mudar a maneira em que o HTML geralmente agrupa os nossos elemento utilizando o modelo de layout Flexbox.

## **2 | Aplicando**



**1**

**Caso real**





## Mas em um caso real?

Agora que já entendemos o que o Flexbox é capaz de fazer, vamos ver como isso funciona na prática!

Para isso, vamos construir um menu utilizando uma lista e alinhar os seus itens um ao lado do outro.





## Exemplo do nosso caso

HTML

```
<ul class="main-menu"> Elemento pai
  <li> Elemento filho
    <a href="#">Home</a>
  </li>
  <li> Elemento filho
    <a href="#">Produtos</a>
  </li>
  <li> Elemento filho
    <a href="#">Sobre</a>
  </li>
</ul>
```

- Home
- Produtos
- Sobre

**2**

## **Aplicando Flexbox**



## Aplicando Flexbox

Agora que estruturamos semanticamente o nosso menu, podemos partir para a utilização do flexbox.

Para isso, basta adicionar `display: flex;` à nossa classe `main-menu` em nosso CSS.

CSS

```
.main-menu {  
  list-style: none;  
  display: flex;  
}
```

*list-style foi utilizado apenas para retirar as bolinhas da lista*



HomeProdutosSobre



## Aplicando Flexbox

Para nossos itens não ficarem grudados podemos utilizar a propriedade `gap: 12px;`

CSS

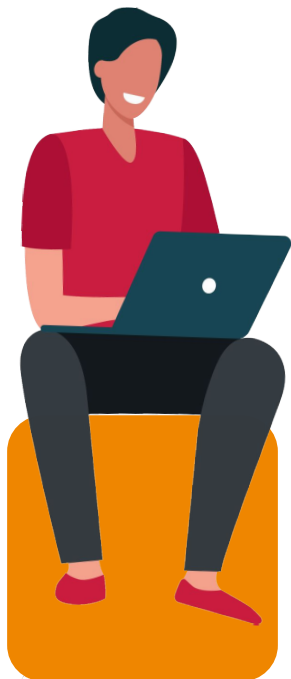
```
.main-menu {  
  list-style: none;  
  display: flex;  
  gap: 12px;  
}
```



*gap adiciona um espaço entre os itens*



## Só é possível fazer menus?



Claramente não! Utilizando Flex, podemos construir diversos layouts que com **position** seria muito difícil.

Mas para isso, precisamos das principais propriedades da Flex.

[Voce pode encontrar uma documentação mais detalhada aqui](#)

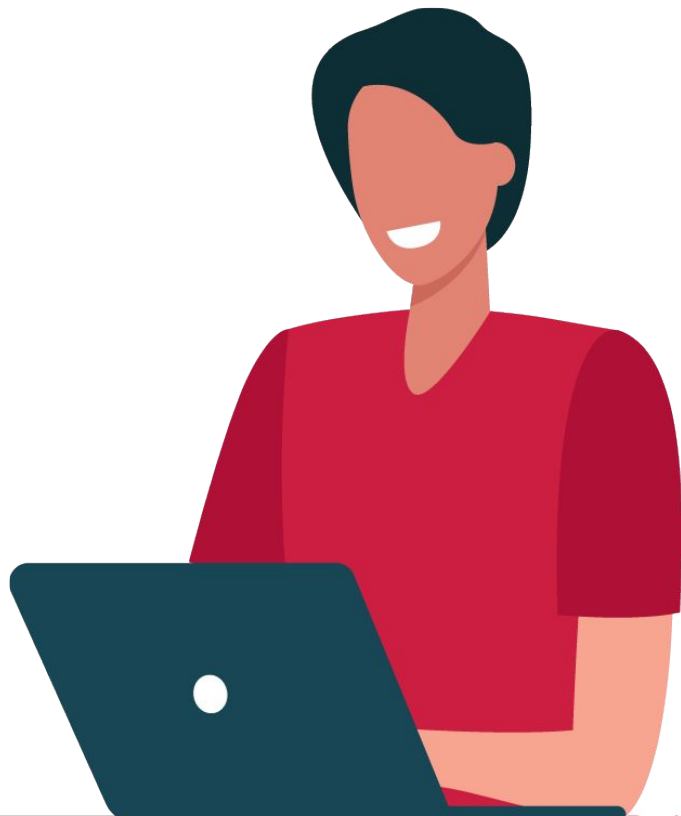
# 3 | Outras propriedades



## Aviso importante

As propriedades que vamos ver sempre serão aplicadas em nosso Elemento Pai, para que ele agrupe nossos Elementos Filhos.

Lembrando que a Flex do Elemento Pai irá agrupar apenas os Elementos que forem Filhos direto, os Filhos dos Elementos Filhos não irão sofrer alterações!

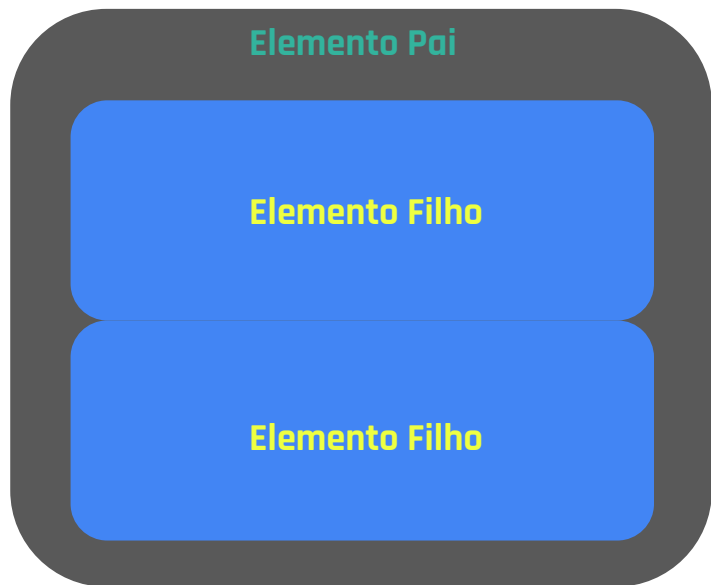




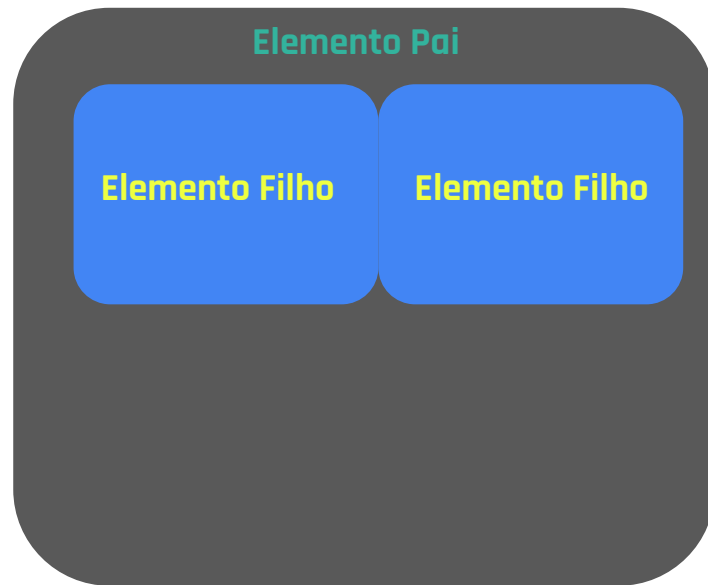


## flex-direction

Com flex-direction é possível alterarmos as direções em que nossos Elementos Filhos serão organizados.



**flex-direction: column** - Alinha um item por coluna

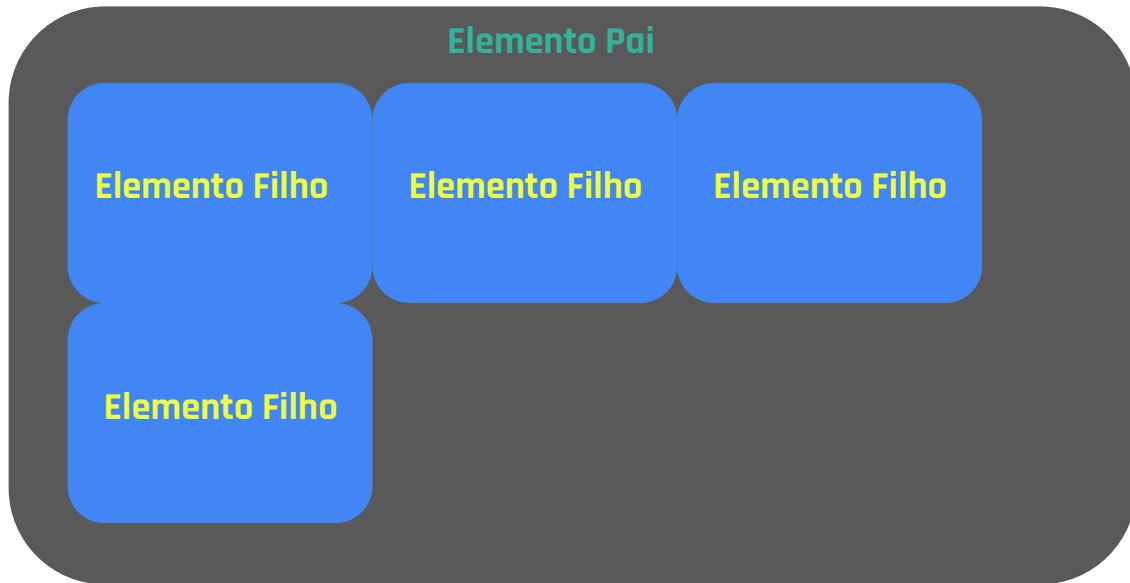


**flex-direction: row** - Alinha vários itens por linha



## flex-wrap: wrap

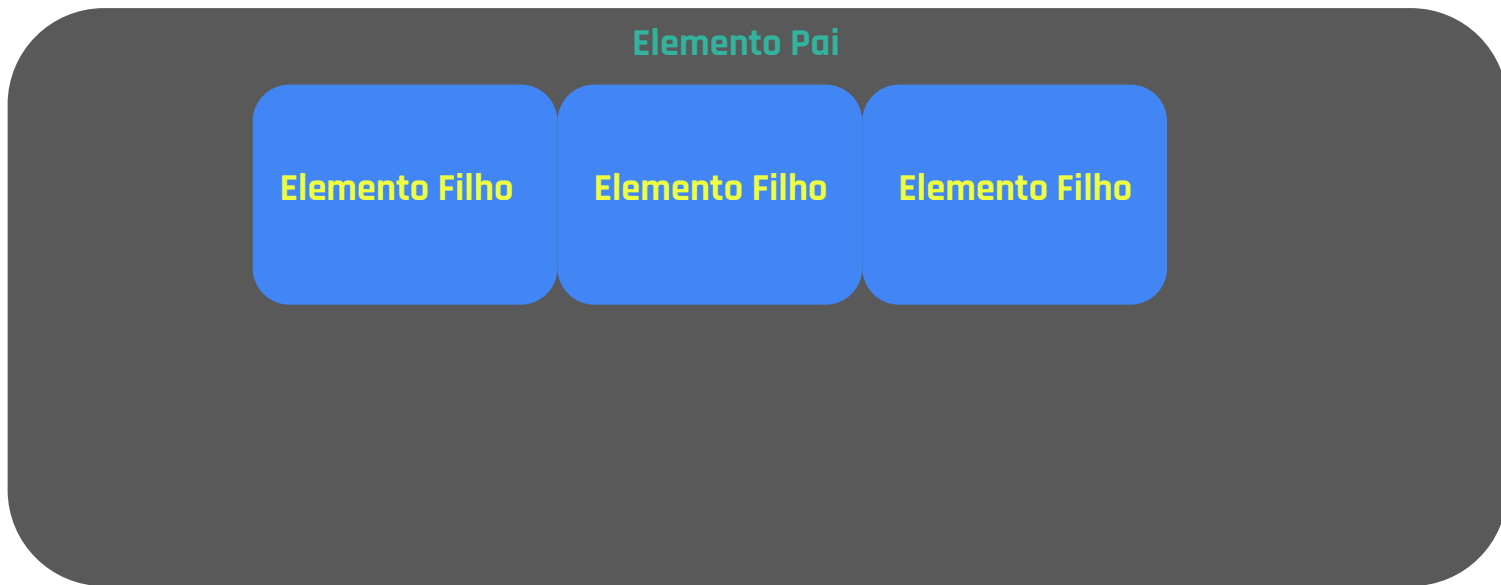
Com flex-wrap é possível dizermos que os nossos elementos podem pular para a linha de baixo caso não haja espaço para ele existir na linha de cima.





## justify-content

Com justify-content é possível alinharmos horizontalmente nossos elementos.

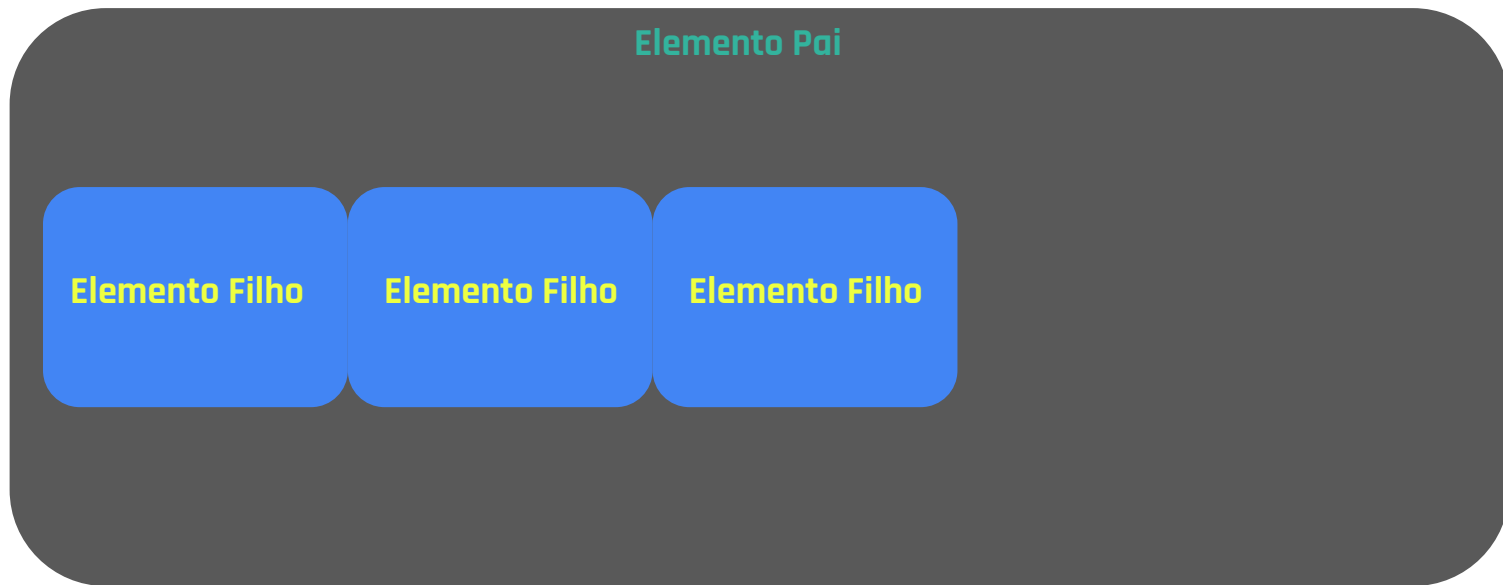


**justify-content: center** - Alinha todos os itens para o centro horizontal do elemento pai



## align-items

Com align-items é possível alinharmos verticalmente nossos elementos.



**align-items: center** - Alinha todos os itens para o centro vertical do elemento pai



## gap

Com gap é possível informarmos um valor para criarmos um espaço entre os itens.



**gap: 24px** - Cria um espaçamento entre os Elementos Filhos de nosso Elemento Pai

**Hora da revisão**



## Conclusão

- Vimos que é possível escolher a direção, centralizar verticalmente, horizontalmente e criar espaçamentos para os Elementos filhos de um Elemento pai que contenha a propriedade `display: flex`;

Dessa maneira, podemos criar layouts cada vez mais ricos e complexos.

DigitalHouse>  
Coding School