



# Display Flex

Existe no CSS a propriedade display com o valor flex, junto temos mais algumas propriedades que em conjunto com o display: flex tem um alto poder sobre os elementos HTML no sentido de posicionamentos e alinhamentos desses elementos em questão, nessa aula vamos aprender uma nova forma que vai mudar como você utiliza o CSS, para um iniciante existe a vida antes e a vida depois do display flex ;)

## Quando utilizar o display flex?

O cenário ideal para trabalhar com display flex é um elemento pai com 1 ou mais filhos dentro, no caso a propriedade display e seus auxiliares são colocados no elemento pai em questão, veremos exemplos a seguir.

O HTML usado será:

```
<!DOCTYPE html>
<html lang="pt_BR">

<head>
  <title>Flex</title>
  <link rel="stylesheet" href="style.css">
</head>

<body>
  <section>
    <div>Primeira Div</div>
    <div>Segunda Div</div>
    <div>Terceira Div</div>
    <div>Quarta Div</div>
  </section>
</body>

</html>
```

E no CSS inicialmente teremos:

```
div{
  margin: 10px;
  background: purple;
  border-radius: 5px;
  color: white;
  padding: 10px;
}

section{
  margin: 100px;
  display: flex;
  background-color: green;
  border-radius: 15px;
}
```

Observe que as divs ficaram uma abaixo da outra, isso acontece pois por padrão o elemento div possui seu display como block, então a div ocupa um bloco, a largura toda, mesmo se for menor que a largura da tela seu "resto" é preenchido com margin, porém se colocarmos display com flex eu seu pai (que também tem display block → section) ao invés de ficarem uma abaixo da outra vão ficar ao lado, porém ainda podemos deixá-las abaixo uma da outra se quisermos, isso tudo mostra o poder do display com flex.

```
section{
  margin: 100px;
  display: flex;
  background-color: green;
  border-radius: 15px;
}
```

```
div{
  margin: 10px;
  background: purple;
  border-radius: 5px;
  color: white;
  padding: 10px;
}
```

Com o display: flex ativado podemos começar a brincar com outras propriedades, veremos abaixo algumas possibilidades.

## flex-direction

Define a direção dos elementos, podemos deixá-los em linha, em coluna, linha reversa e coluna reversa:

```
section {
  display: flex;
  flex-direction: column;
}

/* ou */
section {
  display: flex;
  flex-direction: column-reverse;
}

/* ou */
section {
  display: flex;
  flex-direction: row;
}

/* ou */
section {
  display: flex;
  flex-direction: row-reverse;
}
```

Abaixo vamos ver diferentes tipos de `flex-direction`:

`flex-direction: column`



`flex-direction: row-reverse`



`flex-direction: column-reverse`



`flex-direction: row`



## justify-content

Alinha os itens na vertical, ou seja, no eixo y, observe abaixo como fazemos:

```
section {
  display: flex;

  justify-content: flex-start; /* os itens ficam alinhados no inicio da "linha" */
  justify-content: flex-end; /* os itens ficam alinhados no final da "linha" */
  justify-content: center; /* os itens ficam alinhados ao centro */
  justify-content: space-around; /* os itens ficam com espaçamento em volta */
  justify-content: space-between; /* os itens ficam com espaçamento entre si */
  justify-content: space-evenly; /* os itens ficam com espaçamento mais uniforme */
}
```

Vamos mostrar abaixo os exemplo de `justify-content` :

`justify-content: flex-start`



`justify-content: flex-end`



`justify-content: center`



`justify-content: space-around`



`justify-content: space-between`



`justify-content: space-around`



## align-items

Alinha os itens no eixo horizontal, ou seja, no eixo x, observe abaixo alguns exemplos:

```
section {
  display: flex;

  align-items: center; /* Os itens se alinham pelo centro */
  align-items: flex-end; /* Os itens se alinham por baixo */
  align-items: flex-start; /* Os itens se alinham por cima */
}
```

## flex-wrap

Permite a quebra de linha dos elementos, se um elemento não couber mais em uma "linha" ele é colocado na linha de baixo, veja como é simples utilizar:

```
section {
  display: flex;
  flex-wrap: wrap;
}

section{
  display: flex;
  flex-wrap: nowrap;
}
```

`flex-wrap: no-wrap`



`flex-wrap: wrap`



## A Complete Guide to Flexbox | CSS-Tricks

Our comprehensive guide to CSS flexbox layout. This complete guide explains everything about flexbox, focusing on all the different possible properties for the parent element (the flex container) and the child elements (the flex items). It also includes history, demos, patterns, and a browser support chart.

✳ <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

