

CSS: display (flex)

Autor: Martin Röpke

Código para testar propriedades:

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

<head>
  <style>
    .flex-container {
      display: flex;
    }

    div {
      outline: 3px solid red;
      width: fit-content;
      height: fit-content;
      margin: 10px;
      padding: 10px;
    }
  </style>
</head>

<body>
  <div class="flex-container">
    <div>1</div>
    <div>2</div>
    <div>3</div>
  </div>
</body>

</html>
```

Para que serve o display flex?

A ferramenta display flex serve para tornar mais simples e funcional as tarefas de criação de leiaute.

Ele ajuda em funcionalidades como alinhar e distribuir espaços entre itens em um container, mesmo quando as dimensões destes itens são desconhecidas.



Como funciona?

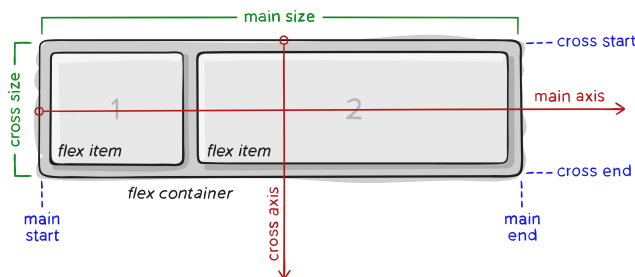
O display flex não é apenas uma única propriedade aplicada pelo CSS, mas sim um conjunto de propriedades CSS, sendo algumas aplicadas em elementos filhos de um elemento container e outras no próprio elemento pai container.

Você define um elemento pai container como flex com a propriedade **display** e o valor **flex**

```
.flex-container {
  display: flex;
}
```

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
```

O display flex é baseado em dois **eixos principais**, o **main axis** e o **cross axis**. Segue uma imagem de exemplo:



Os itens dentro do container serão dispostos no layout seguindo ou o **main axis** ou o **cross axis**, tudo depende de qual eixo foi escolhido como eixo principal.

Algumas propriedades para elementos pai container

Flex-direction

A propriedade flex-direction serve para definir qual eixo é usado como eixo principal, assim alterando a forma como os itens são dispostos.



Existem quatro valores possíveis para essa propriedade. **Row, row-reverse, column e column-reverse**.

As versões **reverse** de cada valor da propriedade apenas alteram se os itens são dispostos da esquerda para direita ou da direita para esquerda no eixo escolhido.

```
.flex-container {
  display: flex;
  flex-direction: row-reverse;
}
```

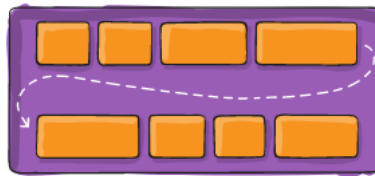


```
.flex-container {
  display: flex;
  flex-direction: column;
}
```



Flex-wrap

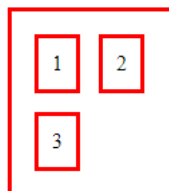
Por padrão, os *flex items* vão todos tentar se encaixar em uma só linha. Com esta propriedade você pode modificar esse comportamento e permitir que os itens quebrem para uma linha seguinte conforme for necessário.



Os valores possíveis para essa propriedade são **nowrap** (padrão), **wrap**, **wrap-reverse**.

```
.flex-container {
  flex-wrap: wrap;
}
```

```
<div class="flex-container" style="width: 100px;">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
```



Justify-content

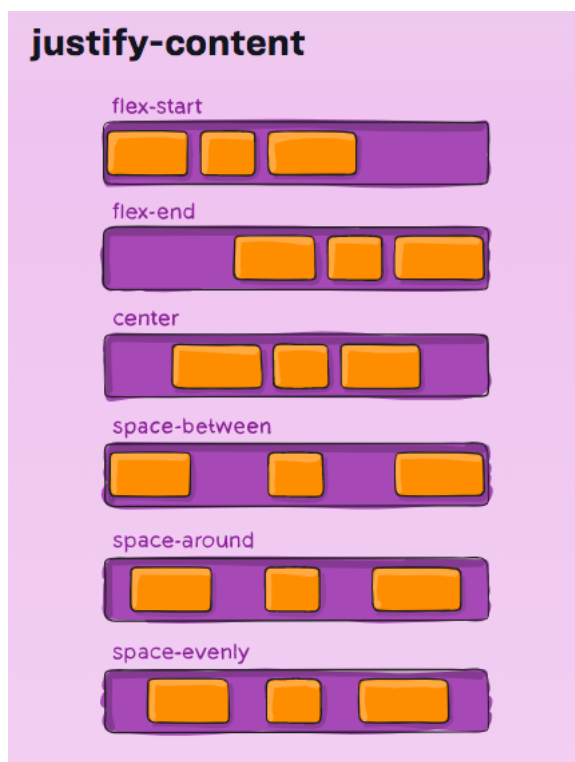
A propriedade **Justify-content** serve para alterar a forma como os itens se alinham em relação em eixo em que eles estão dispostos.

```
.flex-container {
  display: flex;
  justify-content: flex-start;
}
```

```
<div class="flex-container" style="width: 350px;">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
```



Segue abaixo os valores possíveis para essa propriedade:

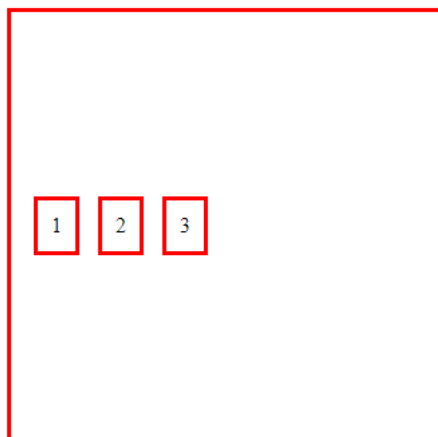


align-items

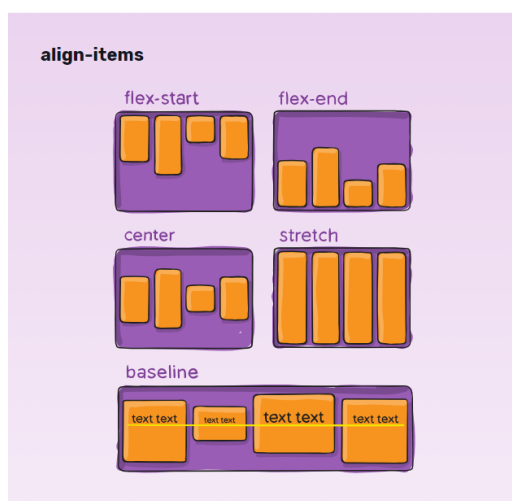
A propriedade **align-items** serve para alterar a forma como os itens se alinham em relação ao eixo perpendicular ao eixo principal no qual eles estão dispostos.

```
.flex-container {
  display: flex;
  align-items: center;
}
```

```
<div class="flex-container" style="width: 300px; height: 300px;">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
```



Segue abaixo os valores possíveis para essa propriedade:



Propriedades para elementos filhos

É possível aplicar propriedades CSS específicas para elementos filhos de um elemento pai container flex e assim mudar a forma como eles são exibidos individualmente ao longo do eixo.

```
<div class="flex-container">
  <div class="flex-item1">1</div>
  <div class="flex-item2">2</div>
  <div class="flex-item3">3</div>
</div>
```

Order

Determina a ordem em que os elementos aparecerão.

```
.flex-item1 {  
  order: 2;  
}  
  
.flex-item2 {  
  order: 3;  
}  
  
.flex-item3 {  
  order: 1;  
}
```



Referências

Se você quiser se aprofundar mais nas propriedades apresentadas e aprender algumas novas propriedades você pode acessar esses sites:

[https://www.alura.com.br/artigos/css-guia-do-flexbox?](https://www.alura.com.br/artigos/css-guia-do-flexbox?gclid=Cj0KCQjw4omaBhDqARIsADXULuXLXYmgnYgkuAQISw13hioZNR14IzD3Q0t6awwgp_6YFWmYSxn42t8aAvTvEALw_wcQ)

[gclid=Cj0KCQjw4omaBhDqARIsADXULuXLXYmgnYgkuAQISw13hioZNR14IzD3Q0t6awwgp_6YFWmYSxn42t8aAvTvEALw_wcQ](https://www.alura.com.br/artigos/css-guia-do-flexbox?gclid=Cj0KCQjw4omaBhDqARIsADXULuXLXYmgnYgkuAQISw13hioZNR14IzD3Q0t6awwgp_6YFWmYSxn42t8aAvTvEALw_wcQ)

<https://origamid.com/projetos/flexbox-guia-completo/>