



flexbox

Flexbox é o nome de um conjunto de regras de CSS para facilitar o posicionamento de múltiplos elementos, o que pode ser bem complicado usando outras técnicas. Ele serve basicamente para organizarmos mais facilmente os elementos filhos dentro do elemento pai, que chamaremos de container daqui para frente.

Para utilizar o *flexbox* devemos alterar o **display** do container, para **flex** usando o css: `display: flex;`



Flex Direction

Então podemos definir qual será a direção que o *flexbox* usará para posicionar os elementos usando **flex-direction**. Isso define como os elementos filhos serão organizados.

Os valores possíveis de **flex-direction** são:

- **row**: em linha (um ao lado do outro) da esquerda para direita
- **row-reverse**: em linha (um ao lado do outro) da direita para esquerda
- **column**: em coluna (um abaixo do outro) de cima para baixo
- **column-reverse**: em coluna (um abaixo do outro) de baixo para cima

`flex direction` determina se o eixo principal do *flexbox* será vertical ou horizontal, isso é importante porque algumas propriedades atuam no eixo principal ou no eixo perpendicular ao principal. Mudar o `direction` de linha para coluna, por exemplo, faz com que essas propriedades se apliquem invertidas.

Alinhamento do conteúdo

`justify-content` e `align-items` são regras de alinhamento dos elementos dentro de um *container flex* e dependem diretamente do `flex-direction` para atuar nesses elementos.

- `justify-content` define como os elementos são espaçados na orientação do eixo principal.
- `align-items` define como os elementos são espaçados na perpendicular ao eixo principal.

Sendo assim, se `flex-direction` é `row`, `justify-content` define o posicionamento horizontal e `align-items` define o posicionamento vertical.

No entanto, se `flex-direction` for alterado para `column`, `justify-content` define o posicionamento vertical e `align-items` define o posicionamento horizontal.

Os valores possíveis de `justify-content` são:

`flex-start` alinha os itens ao início do container.

```
/* CSS */
.container {
```

```
border: 1px solid blue;
display: flex;
flex-direction: row;
justify-content: flex-start;
}
```

```
.item {
width: 80px;
height: 80px;
background-color: #d1d1d1;
border: 1px solid silver;
}
```

```
<!-- HTML -->
<div class="container">
  <div class="item"></div>
  <div class="item"></div>
  <div class="item"></div>
  <div class="item"></div>
</div>
```

Resultado:



Imagem 1 - justify-content: flex-start (Fonte da imagem: do autor)

flex-end alinha os itens ao final do container.

```
/* CSS */  
.container {  
  border: 1px solid blue;  
  display: flex;  
  flex-direction: row;  
  justify-content: flex-end;  
}  
  
.item {  
  width: 80px;  
  height: 80px;  
  background-color: #d1d1d1;  
  border: 1px solid silver;  
}
```

```
<!-- HTML -->  
<div class="container">  
  <div class="item"></div>  
  <div class="item"></div>  
  <div class="item"></div>  
  <div class="item"></div>  
</div>
```

Resultado:



Imagem 2 - justify-content: flex-end (Fonte da imagem: do autor)

center alinha os itens ao centro do container.

```
/* CSS */  
.container {  
  border: 1px solid blue;  
  display: flex;  
  flex-direction: row;  
  justify-content: center;  
}  
  
.item {  
  width: 80px;  
  height: 80px;  
  background-color: #d1d1d1;  
  border: 1px solid silver;  
}
```

```
<!-- HTML -->  
<div class="container">  
  <div class="item"></div>  
  <div class="item"></div>  
  <div class="item"></div>  
  <div class="item"></div>  
</div>
```

Resultado:



Imagem 3 - *justify-content: center* (Fonte da imagem: do autor)

space-between espaça igualmente os elementos usando todo espaço extra entre os elementos.

```
/* CSS */
.container {
  border: 1px solid blue;
  display: flex;
  flex-direction: row;
  justify-content: space-between;
}

.item {
  width: 80px;
  height: 80px;
  background-color: #d1d1d1;
  border: 1px solid silver;
}

<!-- HTML -->
<div class="container">
  <div class="item"></div>
  <div class="item"></div>
  <div class="item"></div>
```

```
<div class="item"></div>  
</div>
```

Resultado:



Imagem 4 - *justify-content: space-between* (Fonte da imagem: do autor)

space-around espaça igualmente os elementos usando todo espaço no entorno dos elementos.

```
/* CSS */  
.container {  
  border: 1px solid blue;  
  display: flex;  
  flex-direction: row;  
  justify-content: space-around;  
}  
  
.item {  
  width: 80px;  
  height: 80px;  
  background-color: #d1d1d1;  
  border: 1px solid silver;  
}
```

```
<!-- HTML -->
<div class="container">
  <div class="item"></div>
  <div class="item"></div>
  <div class="item"></div>
  <div class="item"></div>
</div>
```

Resultado:



Imagem 5 - justify-content: space-around (Fonte da imagem: do autor)

Os valores possíveis de `align-items` são:

`stretch` faz com que os itens cresçam igualmente.

```
/* CSS */
.container {
  border: 1px solid blue;
  display: flex;
  flex-direction: row;
  justify-content: space-around;
  align-items: stretch;
```



```
}
```

```
.item {  
  width: 80px;  
  background-color: #d1d1d1;  
  border: 1px solid silver;  
}
```

```
<!-- HTML -->
```

```
<div class="container">  
  <div class="item"><br /></div>  
  <div class="item"><br /><br /></div>  
  <div class="item"><br /><br /><br /></div>  
  <div class="item"><br /><br /><br /><br /></div>  
</div>
```

Resultado:



Imagem 6 - align-items: stretch (Fonte da imagem: do autor)

flex-start alinha os itens ao início.

```
/* CSS */  
.container {  
  border: 1px solid blue;
```

```
display: flex;  
flex-direction: row;  
justify-content: space-around;  
align-items: flex-start;  
}
```

```
.item {  
width: 80px;  
background-color: #d1d1d1;  
border: 1px solid silver;  
}
```

```
<!-- HTML -->  
<div class="container">  
  <div class="item"><br /></div>  
  <div class="item"><br /><br /></div>  
  <div class="item"><br /><br /><br /></div>  
  <div class="item"><br /><br /><br /><br /></div>  
</div>
```

Resultado:



Imagem 7 - align-items: flex-start (Fonte da imagem: do autor)

flex-end alinha os itens ao final.

```
/* CSS */
.container {
  border: 1px solid blue;
  display: flex;
  flex-direction: row;
  justify-content: space-around;
  align-items: flex-end;
}

.item {
  width: 80px;
  background-color: #d1d1d1;
  border: 1px solid silver;
}

<!-- HTML -->
<div class="container">
  <div class="item"><br /></div>
  <div class="item"><br /><br /></div>
  <div class="item"><br /><br /><br /></div>
  <div class="item"><br /><br /><br /><br /></div>
</div>
```

Resultado:



Imagem 8 - *align-items: flex-end* (Fonte da imagem: do autor)

center alinha os itens ao centro.

```
/* CSS */
.container {
  border: 1px solid blue;
  display: flex;
  flex-direction: row;
  justify-content: space-around;
  align-items: center;
}

.item {
  width: 80px;
  background-color: #d1d1d1;
  border: 1px solid silver;
}

<!-- HTML -->
<div class="container">
  <div class="item"><br /></div>
  <div class="item"><br /><br /></div>
  <div class="item"><br /><br /><br /></div>
  <div class="item"><br /><br /><br /><br /></div>
</div>
```

Resultado:



Imagem 9 - *align-items: center* (Fonte da imagem: do autor)

baseline alinha os itens de acordo com a linha base da tipografia.

```
/* CSS */
.container {
  border: 1px solid blue;
  display: flex;
  flex-direction: row;
  justify-content: space-around;
  align-items: baseline;
}

.item {
  width: 80px;
  background-color: #d1d1d1;
  border: 1px solid silver;
}

<!-- HTML -->
<div class="container">
  <div class="item"><br /></div>
  <div class="item"><br /><br /></div>
  <div class="item"><br /><br /><br /></div>
```

```
<div class="item"><br /><br /><br /><br /></div>
</div>
```

Resultado:



Imagem 10 - *align-items: baseline* (Fonte da imagem: do autor)

Wrap

Quando tiver elementos demais para apenas uma linha, o *flex* espremerá todos eles.

Para impedir que isso aconteça, você pode determinar que o *flex* faça mais de uma linha, para isso você precisa usar a regra **flex-wrap**.

Os valores possíveis são o **no-wrap** (padrão), o **wrap** que quebra linha para baixo e **wrap-reverse** que quebra linha para cima.

E muito mais...

Flexbox tem várias outras regras que podem ser aplicadas aos itens de um *container flex*.

Não abordamos mais do que o essencial aqui, então vale dar uma olhada por conta própria, sugerimos dar uma aprofundada nos links abaixo.

Referências e Links úteis

- [CSS Position](#). MDN Web Docs, 26/03/2022. Acesso em: 27/06/2022.
- [Guia Flexbox](#). Origamid. Acesso em 27/06/2022.
- [A Complete Guide to Flexbox](#). Chris Coyier (CSS Tricks), 08/04/2013. Acesso em 27/06/2022
- [Flexbox - Game](#).

Próximo Tópico