

Flexbox

Através da propriedade flex podemos construir rapidamente layouts complexos e flexíveis com características que antes eram difíceis para o CSS básico.

Flexbox vem com a finalidade de remover as limitações que os floats e positions impunham para algumas propostas de UI.

Inicialmente precisamos definir a propriedade do elemento pai como display flex:

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

flex-direction

Esta propriedade define uma direção e sentido da exibição dos elementos filhos.

```
.container {  
  border: 1px solid red;  
  width: 400px;  
  height: 250px;  
  margin: 0 auto;  
  display: flex;  
  
  flex-direction: row;  
}
```

Valores possíveis

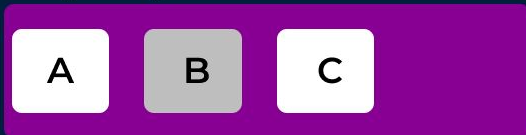
flex-direction: row | row-reverse | column | column-reverse;



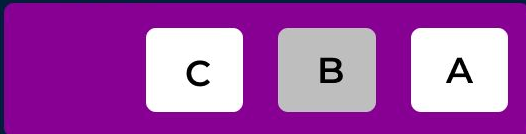
Valores possíveis

`flex-direction: row | row-reverse | column | column-reverse ;`

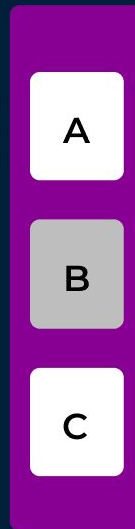
row (padrão)



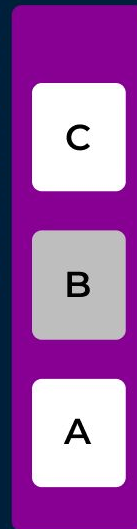
row-reverse



column



column-reverse





justify-content

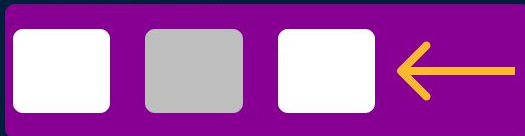
Define o alinhamento dos itens ao longo do eixo principal do container.

```
.container {  
  border: 1px solid red;  
  width: 400px;  
  height: 250px;  
  margin: 0 auto;  
  display: flex;  
  flex-direction: row;  
  
  justify-content: flex-end;  
}
```

Valores possíveis

justify-content: **flex-start** | **flex-end** | **center** | **space-between** | **space-around**;

flex-start (padrão)



center



flex-end



space-between



space-around



space-evenly



align-items

Permite o alinhamento central sob o eixo vertical do pai, antes possível só com workarounds no código.

```
.container {  
  border: 1px solid red;  
  width: 400px;  
  height: 250px;  
  margin: 0 auto;  
  
  display: flex;  
  flex-direction: row;  
  justify-content: center;  
  align-items: center;  
}
```

Valores possíveis

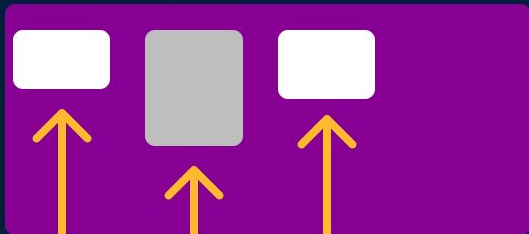
align-items: **flex-start** | **flex-end** | **center** | **stretch** | **baseline**;



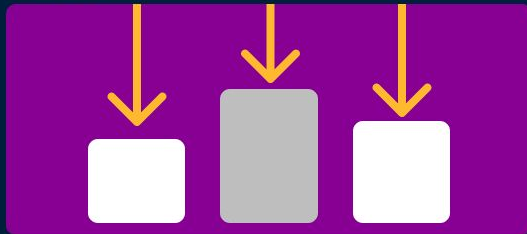
Valores possíveis

`align-items: flex-start | flex-end | center | stretch | baseline;`

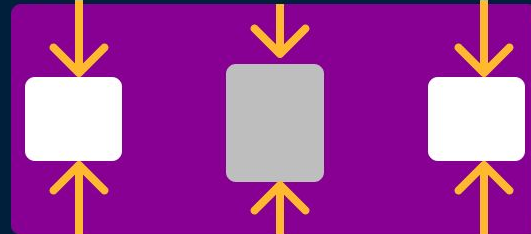
flex-start (padrão)



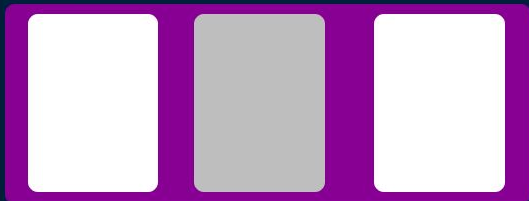
flex-end



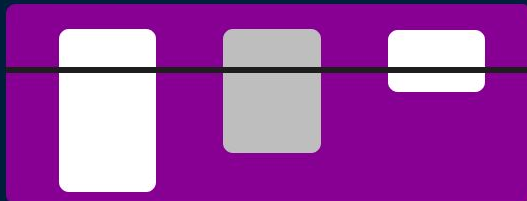
center



stretch



baseline

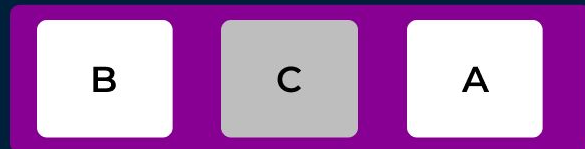


order

Permite ordenar elementos manualmente dentro de um container.

```
<div class="container">  
  <div class="box-a">A</div>  
  <div class="box-b">B</div>  
  <div class="box-c">C</div>  
</div>
```

```
.container .box-a {  
  order: 2;  
}  
.container .box-b {  
  order: 0;  
}  
.container .box-c {  
  order: 1;  
}
```



flex-grow

É a capacidade de um elemento crescer dentro de um pai que seja flex.

```
<div class="container">  
  <div class="box-a">A</div>  
  <div class="box-b">B</div>  
</div>
```

Valor padrão:

flex-grow: 0;

```
.container {  
  border: 1px solid purple;  
  width: 200px;  
  height: 400px;  
  
  display: flex;  
}  
.container div {  
  background: orange;  
  width: 50px;  
  border: 2px solid red;  
  height: 100px;  
  flex-grow: 0;  
}
```





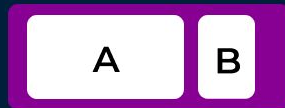
flex-shrink (encolhimento do elemento)

É a capacidade de um elemento se esticar dentro de um pai que seja flex.

Esta propriedade impede que mesmo que a largura dos elementos excedam o pai, eles evitem de “vazar” para fora do elemento.

```
<div class="container">  
  <div class="box-a">A</div>  
  <div class="box-b">B</div>  
</div>
```

```
.container {  
  border: 1px solid purple;  
  width: 200px;  
  height: 400px;  
  
  display: flex;  
}  
.container div {  
  background: orange;  
  width: 50px;  
  border: 2px solid red;  
  height: 100px;  
  flex-grow: 1;  
}
```



***Esta propriedade é sempre aplicada aos filhos de um container!**

flex-grow

O valor deve sempre ser numérico, sendo que quanto maior o valor MAIOR a capacidade de redução do elemento.

Irredutível:

```
flex-grow: 0;
```

Redutível:

```
flex-grow: 1;
```

Mais redutível:

```
flex-grow: 2;
```

As reduções são proporcionais aos níveis atribuídos à propriedade.

O valor deve ser sempre positivo!

flex-basis

Permite forçar manualmente que todos elementos tenham o mesmo tamanho dentro de um container.

```
<div class="container">  
  <div class="box-a">A</div>  
  <div class="box-b">Outro  
elemento</div>  
</div>
```

Valor padrão:

flex-basis: auto;

```
..container {  
  border: 1px solid purple;  
  width: 200px;  
  height: 400px;
```

```
  display: flex;  
}
```

```
.container div {  
  background: orange;  
  border: 2px solid red;  
  height: 100px;  
  flex-grow: 1;
```

```
  flex-basis: auto;  
}
```

A

Outro elemento

flex-basis

O valor deve sempre ser numérico, sendo que quanto maior o valor MAIOR a capacidade de redução do elemento.

Se ajusta automaticamente:

```
flex-basis: auto;
```

Todos elementos com tamanho em 80px:

```
flex-basis: 80px;
```

A expansão / ajuste dos elementos depende do tamanho do elemento pai, e podem ter limite.

Os valores podem ser em %, px ou auto.



flex-wrap

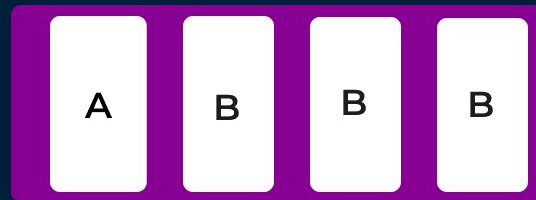
Define se os elementos devem ficar na mesma linha, ou se podem ser quebrados em várias linhas.

```
<div class="container">  
  <div class="box-a">A</div>  
  <div class="box-b">B</div>  
  <div class="box-c">C</div>  
  <div class="box-d">D</div>  
</div>
```

Valor padrão:

flex-wrap: nowrap;

```
.container {  
  background: orange;  
  color: white;  
  width: 600px;  
  text-align: center;  
  
  display: flex;  
  flex-wrap: nowrap;  
}  
  
.container div {  
  border: 1px solid white;  
  height: 120px;  
  width: 250px;  
  background: purple;  
}
```

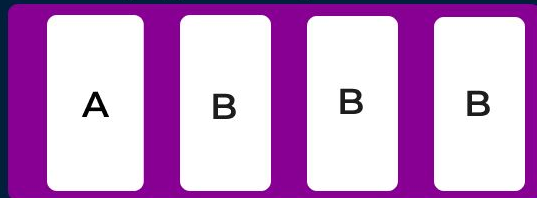




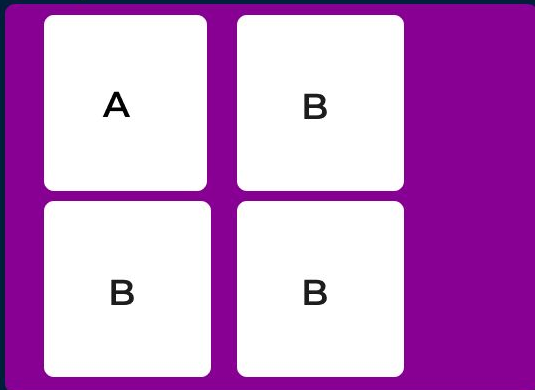
flex-wrap

Define se os elementos devem ficar na mesma linha, ou se podem ser quebrados em várias linhas.

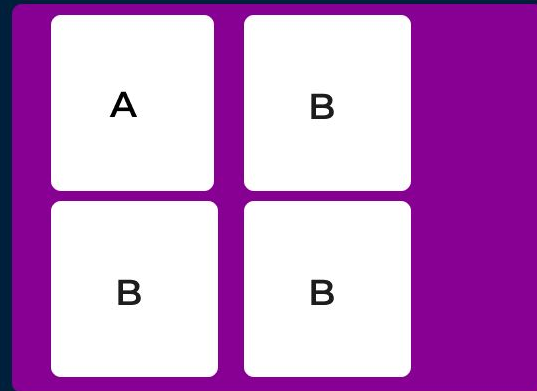
nowrap (padrão)



wrap



wrap-reverse





flex-wrap

Possui três variantes possíveis.

Evita a quebra:

```
flex-wrap: nowrap;
```

Os elementos se quebrar para a próxima linha quando necessário:

```
flex-wrap: wrap;
```

Os elementos se quebram em ordem inversa :

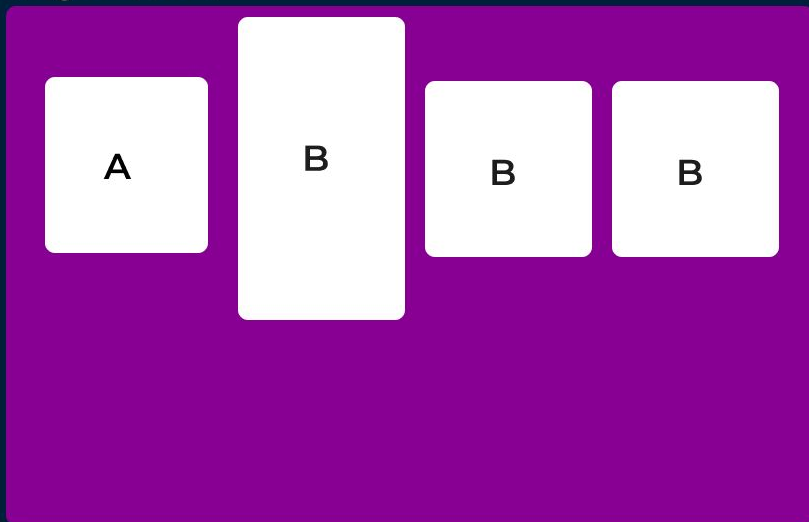
```
flex-wrap: wrap-reverse;
```



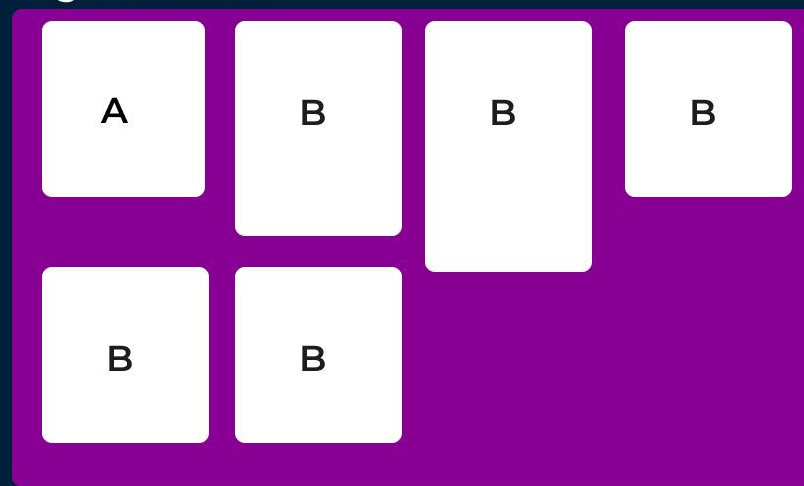

flex-wrap

Define se os elementos devem ficar na mesma linha, ou se podem ser quebrados em várias linhas.

align-items



align-content





O site desenvolvido deve ter todos os exemplos a baixo com as cores que achar melhor.

