

# FLEXBOX

Serve para você alinhar elementos dentro de uma caixa de forma dinâmica

O que seria essa caixa/container?

- Seria um elemento pai, podendo ser uma `<div>`, `<ul>`
- Esse container sempre vai ser colocado no elemento que vai estar por fora dos elementos que vamos querer, realmente, alinhar.
- Sempre vai ter o container no elemento pai.

## PROPRIEDADES PARA USAR NO ELEMENTO PAI

1. **Display: flex;** = Para identificar que a tag vai ser o elemento container do flexbox.
2. **flex-direction:** Para dizer qual vai ser a direção do eixo principal.



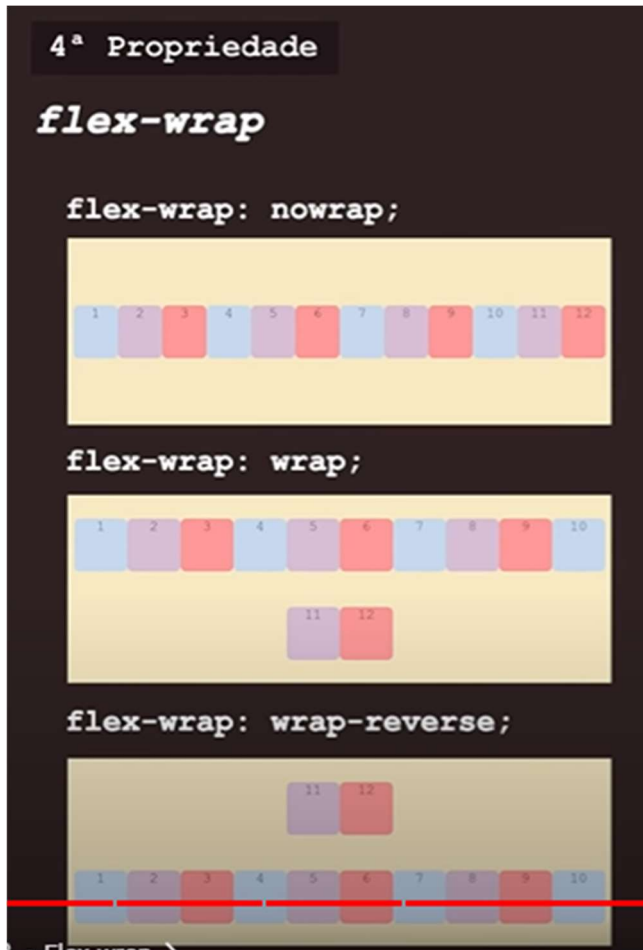
3. **Justify-content:** Para alinhar os elementos no eixo principal.



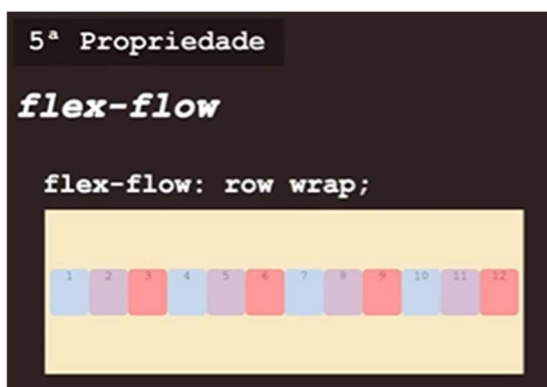
4. **Align-items:** Para alinhar os elementos no eixo perpendicular.



5. **Flex-wrap:** Para decidir se os elementos irão para a linha de baixo quando chegarem no final ou se ficarão espremidos.



6. **Flex-flow:** Para indicar o flex-direction e o flex-wrap de uma só vez.




7. **Align-content:** Para quando existe o flex-wrap e queremos alinhar as linhas no eixo perpendicular


6ª Propriedade

**align-content**

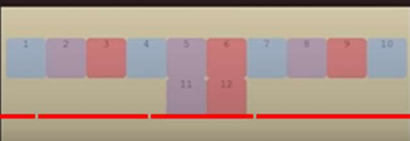
`align-content: flex-start;`



`align-content: flex-end;`



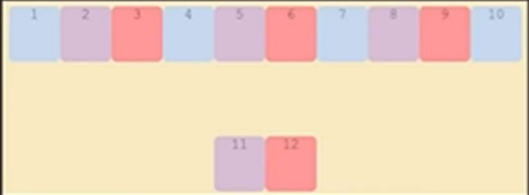
`align-content: center;`




6ª Propriedade

**align-content**

`align-content: space-between;`



`align-content: space-around;`



3 • Align-content >

8. **Gap, row-gap e column-gap:** Para fixar o tamanho do espaço entre os elementos. Sejam eles lado a lado ou entre eles em cada linha

7ª Propriedade

**gap, row-gap e column-gap:**

`gap: 20px;`



`gap: 70px;`



`gap: 10px 30px;`



7ª Propriedade

**gap, row-gap e column-gap:**

`row-gap: 10px;`



`column-gap: 30px;`



8 • Gap, row-gap e column-gap >

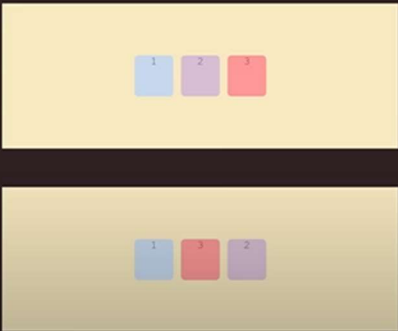
# PROPRIEDADES PARA USAR NO ELEMENTOS FILHOS

1) **Order:** Vai mudar a ordem dos elementos, colocando com a gente quiser.

1ª Propriedade

**order**

Exemplo:



1ª Propriedade

**order**

```
.bloco-1 {
  order: 0;

  /* indiferente: */
  background-color: #c6d8f0;
}


.bloco-2 {
  order: 0;

  /* indiferente: */
  background-color: #d8bfd8;
}

.bloco-3 {
  order: 0;

  /* indiferente: */
  background-color: #ff9999;
}
```

Representação dos blocos



1ª Propriedade

**order**

```
.bloco-1 {
  order: 0;


  /* indiferente: */
  background-color: #c6d8f0;
}

.bloco-2 {
  order: 1;

  /* indiferente: */
  background-color: #d8bfd8;
}

.bloco-3 {
  order: 0;
}
```

Representação dos blocos



1ª Propriedade

**order**

```
.bloco-1 {
  order: 0;


  /* indiferente: */
  background-color: #c6d8f0;
}

.bloco-2 {
  order: 1;

  /* indiferente: */
  background-color: #d8bfd8;
}

.bloco-3 {
  order: -1;
}
```

Representação dos blocos



- 2) **Flex-grow:** Define a possibilidade de um ou mais itens de cresce, caso seja necessário.

## 2ª Propriedade

### *flex-grow*

**Exemplo:**



## 2ª Propriedade

### *flex-grow*

```
.bloco-1 {  
  order: 0;  
  flex-grow: 0;  
  
  /* indiferente: */  
  background-color: #c6d8f0;  
}  
  
.bloco-2 {  
  order: 0;  
  flex-grow: 0;  
  
  /* indiferente: */  
  background-color: #d8bfd8;  
}  
  
.bloco-3 {  
  order: 0;  
  flex-grow: 0;  
  
  /* indiferente: */  
  background-color: #ff9999;  
}
```

#### Representação dos blocos



## 2ª Propriedade

### *flex-grow*

```
.bloco-1 {  
  order: 0;  
  flex-grow: 1;  
  
  /* indiferente: */  
  background-color: #c6d8f0;  
}  
  
.bloco-2 {  
  order: 0;  
  flex-grow: 0;  
  
  /* indiferente: */  
  background-color: #d8bfd8;  
}  
  
.bloco-3 {  
  order: 0;  
  flex-grow: 0;  
  
  /* indiferente: */  
  background-color: #ff9999;  
}
```

#### Representação dos blocos



## 2ª Propriedade

### *flex-grow*

```
.bloco-1 {  
  order: 0;  
  flex-grow: 1;  
  
  /* indiferente: */  
  background-color: #c6d8f0;  
}  
  
.bloco-2 {  
  order: 0;  
  flex-grow: 1;  
  
  /* indiferente: */  
  background-color: #d8bfd8;  
}  
  
.bloco-3 {  
  order: 0;  
  flex-grow: 0;  
  
  /* indiferente: */  
  background-color: #ff9999;  
}
```

#### Representação dos blocos



## 2ª Propriedade

### *flex-grow*

```
.bloco-1 {  
  order: 0;  
  flex-grow: 1;  
  
  /* indiferente: */  
  background-color: #c6d8f0;  
}  
  
.bloco-2 {  
  order: 0;  
  flex-grow: 1;  
  
  /* indiferente: */  
  background-color: #d8bfd8;  
}  
  
.bloco-3 {  
  order: 0;  
  flex-grow: 1;  
  
  /* indiferente: */  
  background-color: #ff9999;  
}
```

#### Representação dos blocos



## 2ª Propriedade

### *flex-grow*

```
.bloco-1 {  
  order: 0;  
  flex-grow: 2;  
  
  /* indiferente: */  
  background-color: #c6d8f0;  
}  
  
.bloco-2 {  
  order: 0;  
  flex-grow: 1;  
  
  /* indiferente: */  
  background-color: #d8bfd8;  
}  
  
.bloco-3 {  
  order: 0;  
  flex-grow: 1;  
  
  /* indiferente: */  
  background-color: #ff9999;  
}
```

#### Representação dos blocos





## 2ª Propriedade

### *flex-grow*



**1** = 456px

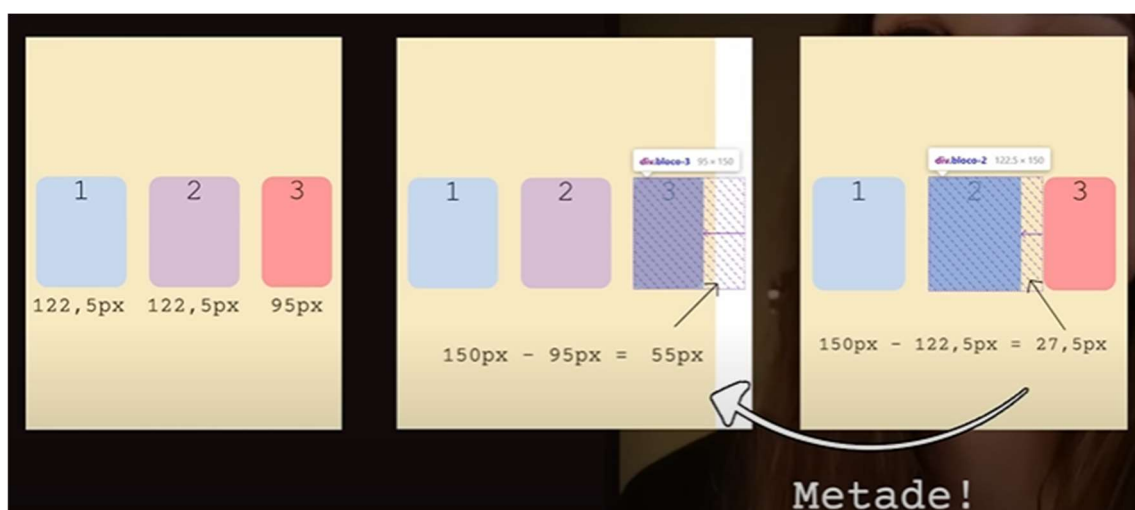
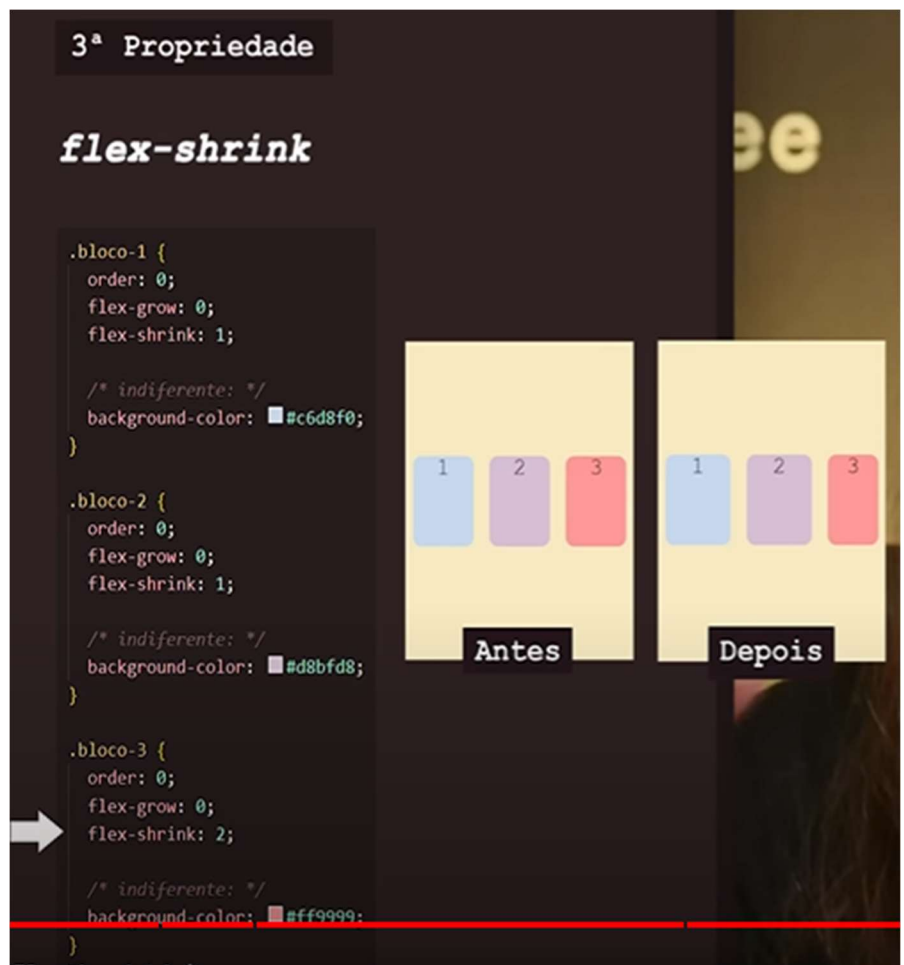
456px - 150px = 306px

**2** = 303px

303px - 150px = 153px

dobro  
proporcional

- 3) **Flex-shrink:** Caso seja necessário, os itens, vão escolher para caber dentro da classe contêiner, na tag pai. O container deve ter uma largura máxima estabelecida. Tendo como padrão o 1 (vai se ajustar)

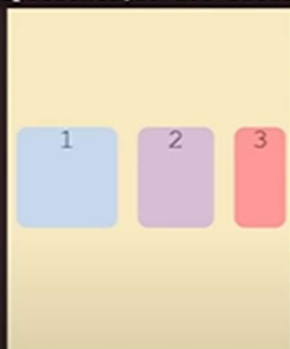


### 3ª Propriedade

## ***flex-shrink***

```
.bloco-1 {  
  order: 0;  
  flex-grow: 0;  
  flex-shrink: 0;  
  
  /* indiferente: */  
  background-color: #c6d8f0;  
}  
  
.bloco-2 {  
  order: 0;  
  flex-grow: 0;  
  flex-shrink: 1;  
  
  /* indiferente: */  
  background-color: #d8bfd8;  
}  
  
.bloco-3 {  
  order: 0;  
  flex-grow: 0;  
  flex-shrink: 2;  
  
  /* indiferente: */  
  background-color: #ff9999;  
}
```

Representação dos blocos



- 4) **Flex-basis:** Vai definir o tamanho dos nossos itens, tanto em largura e altura. Tendo como padrão seu valor auto.



5) **Flex:** É a junção de três propriedades (flex-grow, flex-shrink, flex-basis)



- 6) **Align-self:** Ele “desrespeita” a tag pai e impõe suas condições.

### ***align-self***

```
.bloco-1 {  
  order: 0;  
  /* flex-grow: 0;  
  flex-shrink: 1;  
  flex-basis: auto; */  
  flex: 0 1 auto;  
  align-self: flex-start;  
  
  /* indiferente: */  
  background-color: #c6d8f0;  
}
```



Representação dos blocos

### ***align-self***

```
.bloco-2 {  
  order: 0;  
  /* flex-grow: 0;  
  flex-shrink: 1;  
  flex-basis: auto; */  
  flex: 0 1 auto;  
  align-self: flex-end;  
  
  /* indiferente: */  
  background-color: #d8bfd8;  
}
```



Representação dos blocos