

Curso Técnico Superior Profissional em Tecnologias e Programação de Sistemas de Informação

Unidade Curricular: Design Hipermedia

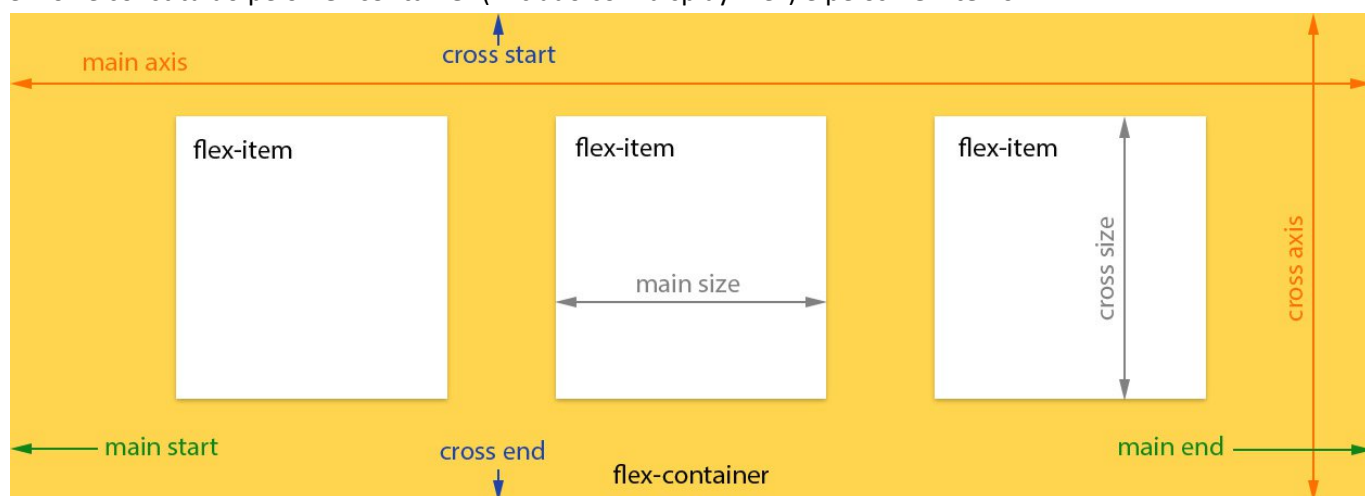
1º Ano/1º Semestre

Docente: Marco Miguel Olival Olim

Data 30/10/2017

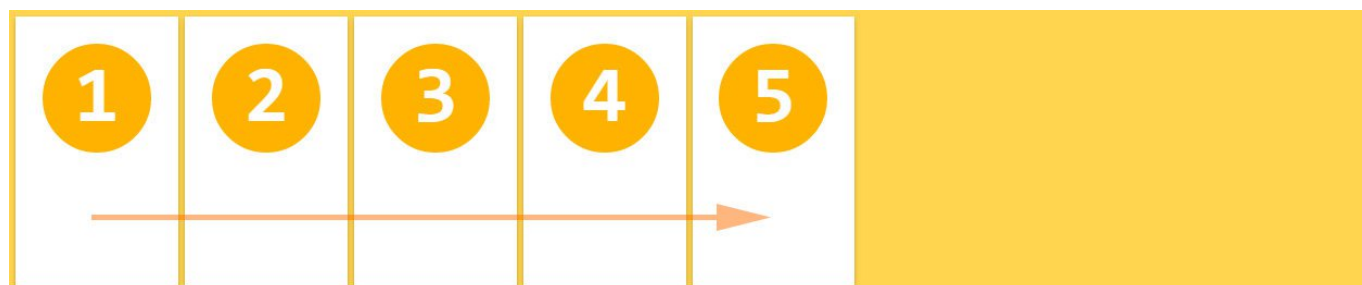
ESTE EXERCÍCIO PRETENDE EXPLORAR AS DIFERENTES PROPRIEDADES DO FLEXBOX

1. O FlexBox é constituído pelo **flex container** (iniciado com `display: flex`) e pelos **flex items**:

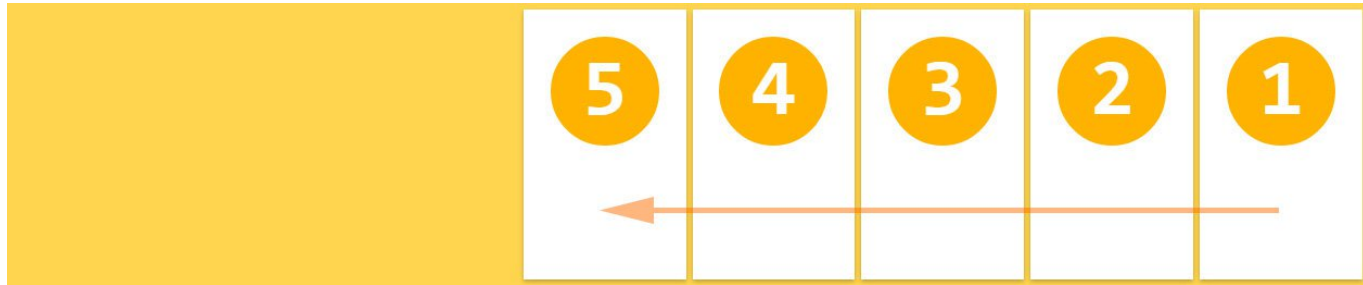


2. **flex-direction**: especifica como os flex items são posicionados no flex container, definindo a direção do eixo principal. Em suma, os flex items podem ser orientados em linhas (valor default) ou colunas:

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



**flex-direction:** row-reverse:



**flex-direction:** column:



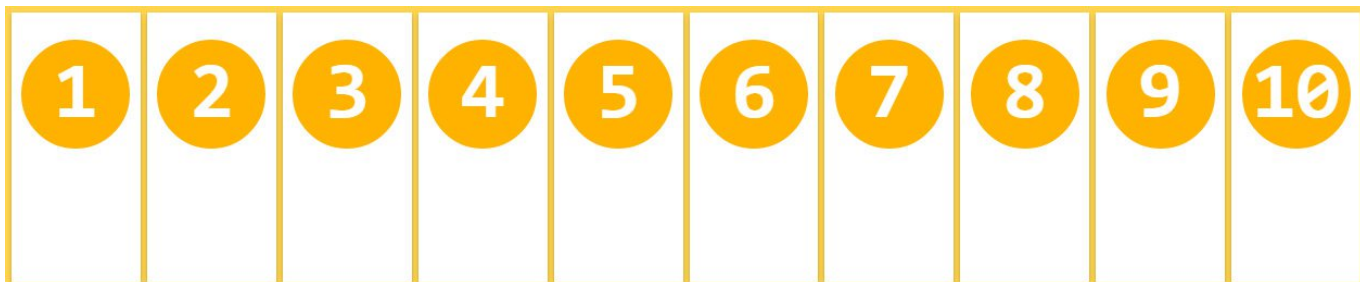
**flex-direction:** column-reverse:



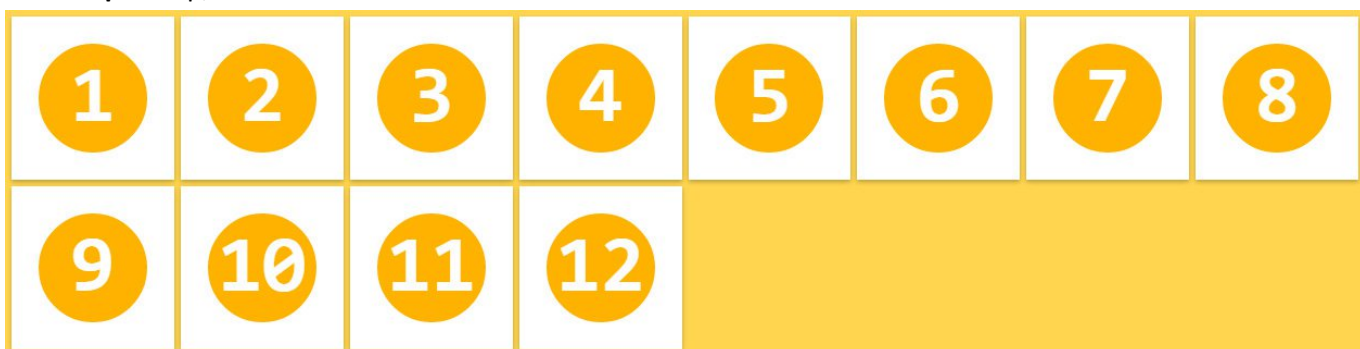
Cofinanciado por:

3. **flex-wrap**: os flex items são dispostos numa única linha (por default), adaptando as suas dimensões para se adaptar ao flex container. Usamos esta propriedade para permitir que os itens fluam em múltiplas linhas:

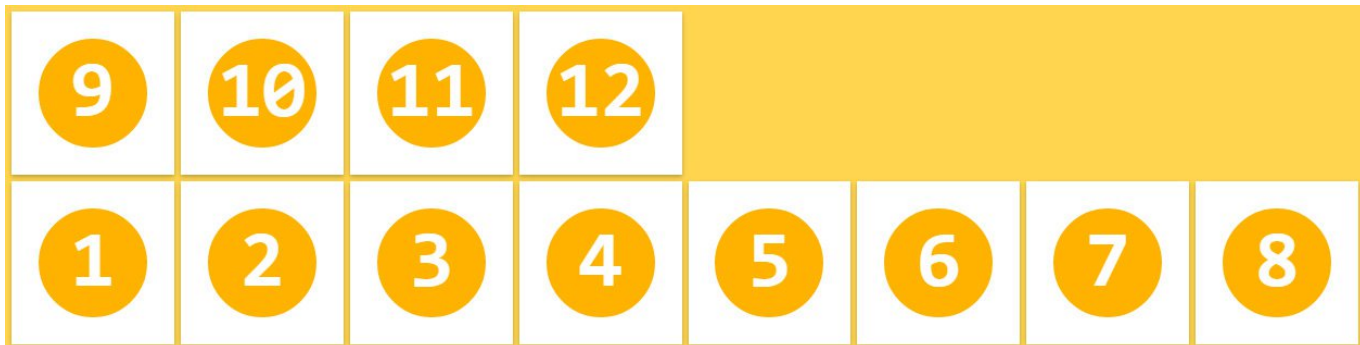
```
.flex-container {  
  display: flex;  
  flex-wrap: nowrap;  
}
```



**flex-wrap: wrap;**



**flex-wrap: wrap-reverse;**



4. **flex-flow**: Esta propriedade aglomera o flex-direction e o flex-wrap.

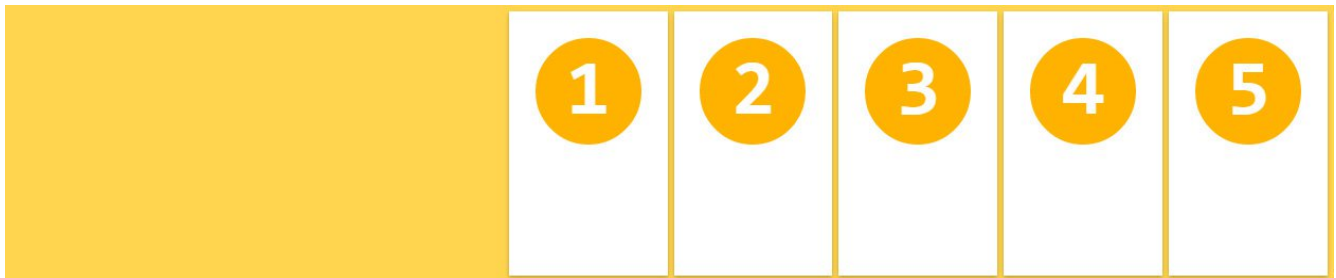
Cofinanciado por:

5. **justify-content**: Define o alinhamento dos flex items no eixo principal.

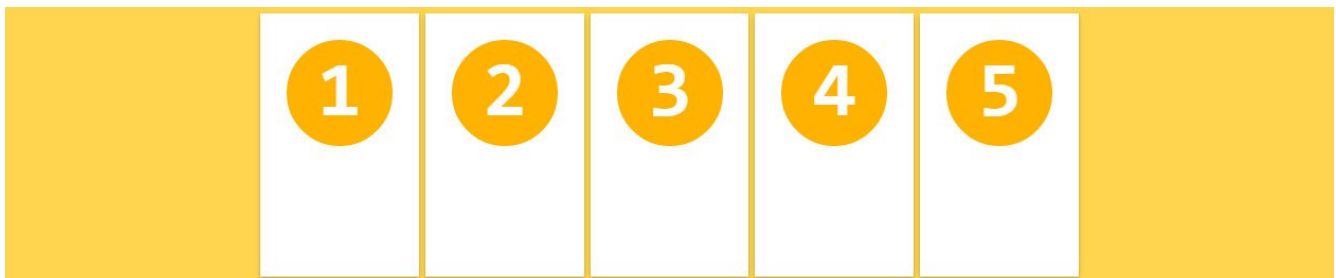
```
.flex-container {  
  display: flex;  
  justify-content: flex-start;  
}
```



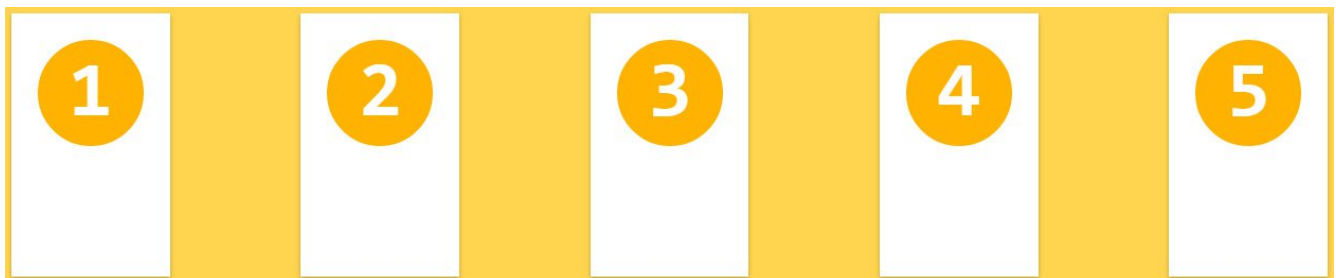
**justify-content: flex-end;**



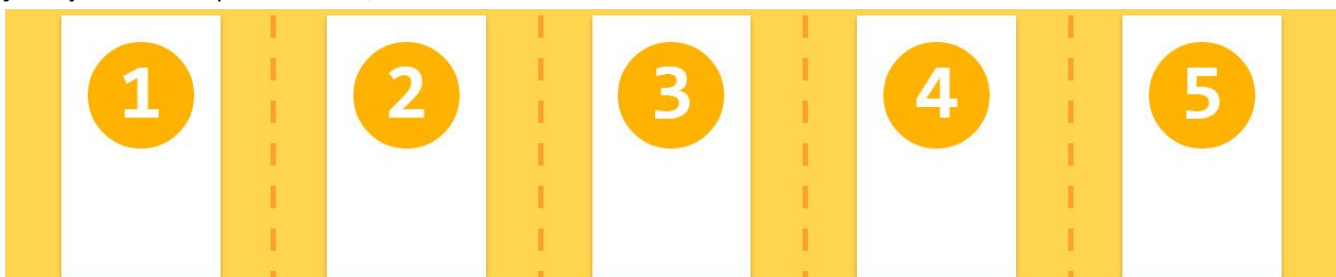
**justify-content: center;**



**justify-content: space-between;**



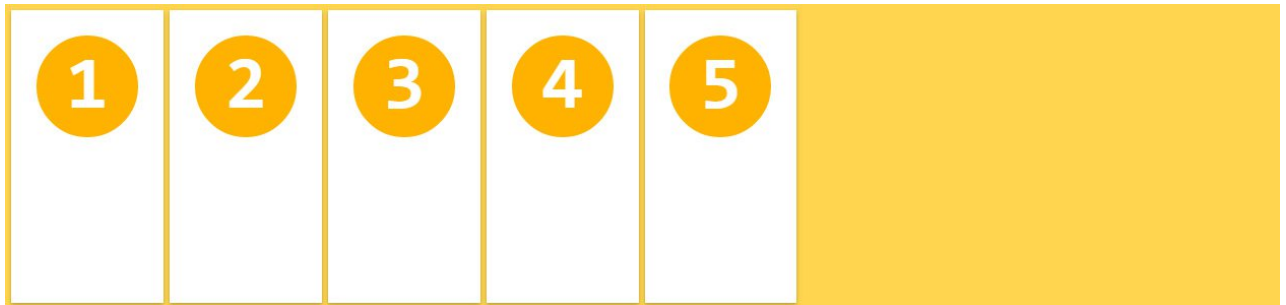
**justify-content: space-around;**



Cofinanciado por:

6. **align-items**: Define o alinhamento dos flex items no **eixo perpendicular** ao eixo principal (cross-axis).

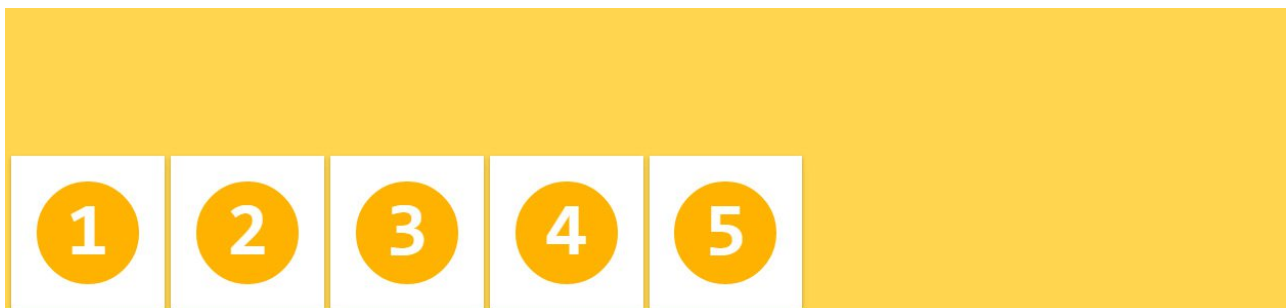
```
.flex-container {  
  display: flex;  
  align-items: stretch;  
}
```



**align-items: flex-start;**



**align-items: flex-end;**



**align-items: center;**



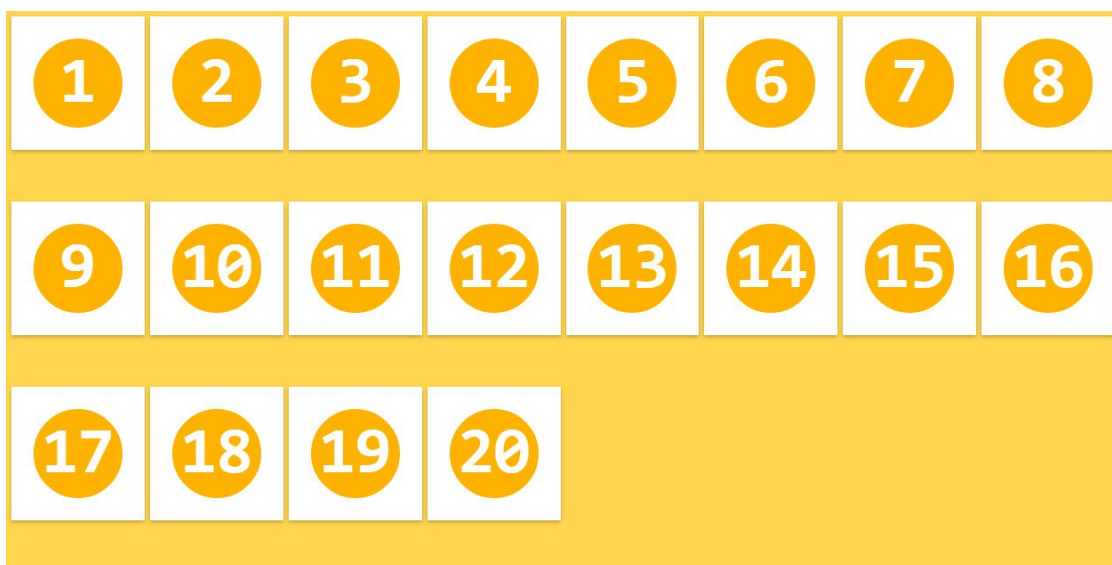
Cofinanciado por:

**align-items: baseline;**

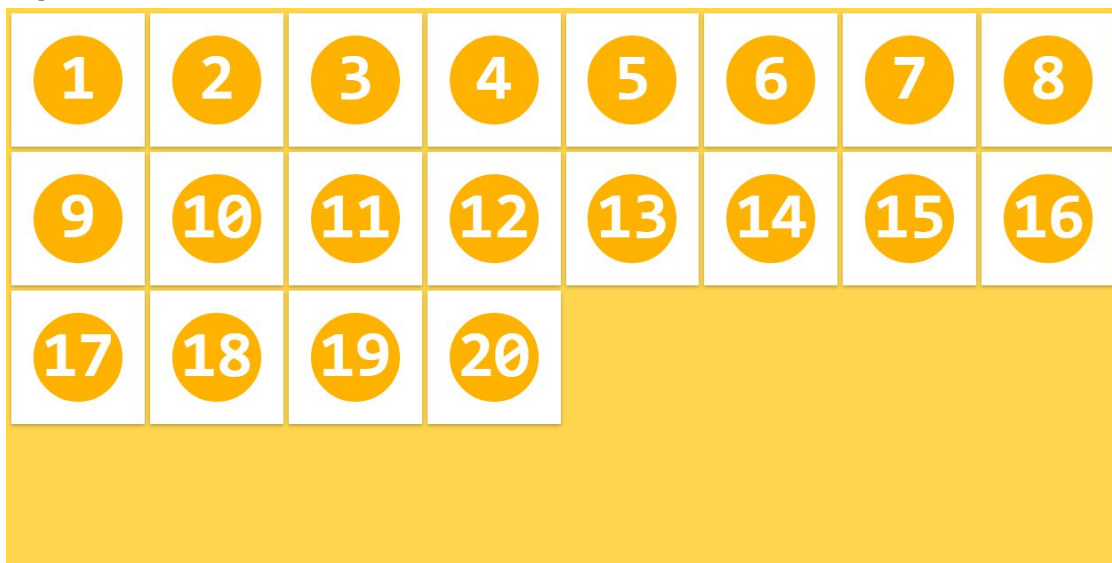


7. **align-content:** Define o alinhamento no flex container (quando ainda resta espaço no cross-axis):

```
.flex-container {  
  display: flex;  
  align-content: stretch;  
}
```



**align-content: flex-start;**



Cofinanciado por:



**align-content:** flex-end;

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20				

**align-content:** center;

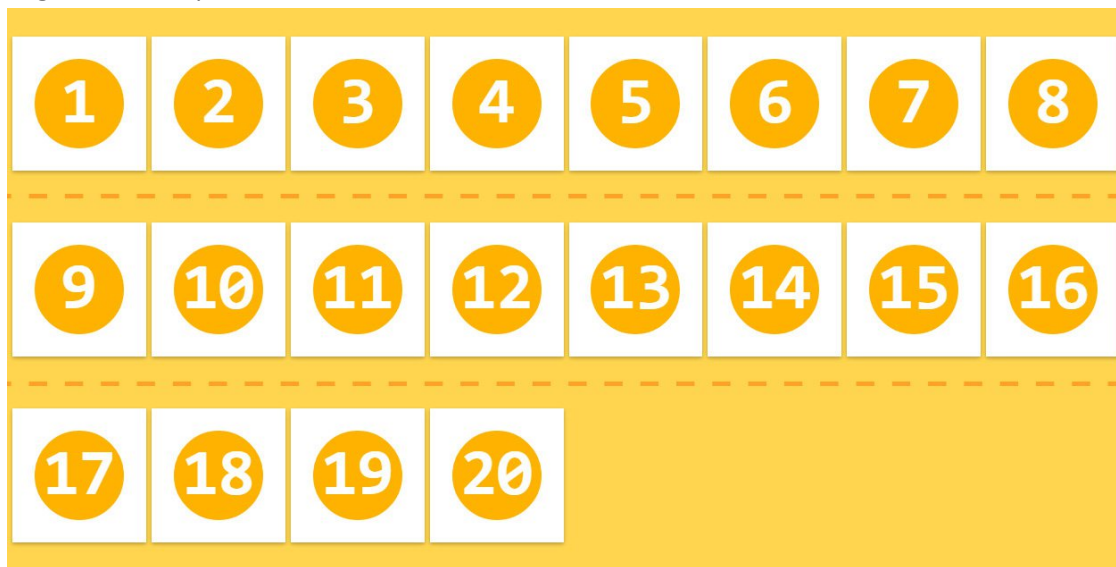
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20				

**align-content:** space-between;

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20				

Cofinanciado por:

align-content: space-around;



## 8. Propriedades específicas aos flex items

```
.flex-item {  
  order: #;  
  
  flex-shrink: #;  
  flex-grow: #;  
  flex-basis: #;  
  flex: # # #;  
  
  align-self: auto;  
}
```

### 8.1. **order**: altera a ordem do item no flex container



### 8.2. **flex-shrink**: especifica o quanto um flex item pode encolher em relação aos restantes items, sendo que quando for 0 o flex item tem de manter o tamanho original



Cofinanciado por:



8.3. **flex-grow**: especifica o quanto um flex item pode expandir em relação aos restantes items, sendo que quando for 0 o item tem de manter o tamanho original



8.4. **flex-basis**: Especifica o tamanho inicial de um flex item



8.5. **flex**: Esta propriedade aglomera o flex-grow, flex-shrink e o flex-basis. Entre os valores atribuídos mais frequentemente, temos o **auto** (1 1 auto) e o **none** (0 0 auto).

8.6. **Align-self**: Esta propriedade serve para sobrepor os valores (override) de um flex item em particular



Cofinanciado por: