



Design Web

Prof. Romerito Campos

Plano de Aula

- Objetivo: compreender as propriedades flex aplicadas ao container flex.
- Conteúdo programático:
 - `align-items` e `align-content` (container)
- introdução ao flex-item.
 - Propriedade `flex`: **item flex**
 - flex, min-width, flex-grow, flex-shrink

Align-items (container)

align-items (container)

- Define como os itens serão acomodados no eixo `cross`.
- O fluxo dos itens é em relação ao eixo `cross-start` ou `cross-end`.
- Há 5 valores para esta propriedade:
 - `stretch` (padrão), `flex-start`, `flex-end`, `center`, `baseline`.

align-items (container)

- `flex-start`: [Exemplo01](#)

```
.flex {  
  display: flex;  
  height: 120px;  
  padding: 5px;  
  align-items: flex-start;  
  border: 1px solid black;  
}  
.item {  
  border: 1px solid black;  
  width: 100px;  
}
```

```
<div class="flex">  
  <div class="item">  
    box 1  
  </div>  
  <div class="item">  
    box 2  
  </div>  
  <div class="item">  
    box 3  
  </div>  
</div>
```

align-items

- `flex-end`: [Exemplo02](#)

```
.flex {  
  display: flex;  
  height: 120px;  
  padding: 5px;  
  align-items: flex-end;  
  border: 1px solid black;  
}  
.item {  
  border: 1px solid black;  
  width: 100px;  
}
```

```
<div class="flex">  
  <div class="item">  
    box 1  
  </div>  
  <div class="item">  
    box 2  
  </div>  
  <div class="item">  
    box 3  
  </div>  
</div>
```

align-items

- center: [Exemplo03](#)

```
.flex {  
  display: flex;  
  height: 120px;  
  padding: 5px;  
  align-items: center;  
  border: 1px solid black;  
}  
.item {  
  border: 1px solid black;  
  width: 100px;  
}
```

```
<div class="flex">  
  <div class="item">  
    box 1  
  </div>  
  <div class="item">  
    box 2  
  </div>  
  <div class="item">  
    box 3  
  </div>  
</div>
```

align-items

- `baseline`: [Exemplo04](#)

```
/* vai parecer com flex-start */
.flex {
  display: flex;
  height: 120px;
  padding: 5px;
  align-items: baseline;
  border: 1px solid black;
}
.item {
  border: 1px solid black;
  width: 100px;
}
```

```
<div class="flex">
  <div class="item">
    box 1
  </div>
  <div class="item">
    box 2
  </div>
  <div class="item">
    box 3
  </div>
</div>
```


align-items

- `baseline`: [Exemplo05](#)
- Neste exemplo, há de fato impacto no uso de baseline

```
<div class="flex">
  <div class="item" style="font-size: 40px; margin-top: 10px">
    box 1
  </div>
  <div class="item">
    box 2
  </div>
  <div class="item">
    box 3
  </div>
</div>
```

align-content (container)

align-content (container)

- Controla como o espaço extra no eixo `cross` pode ser usado.
- Semelhante a `justify-content`, que atua no eixo principal.
- **Válido para containers multi-linhas.**
- Há 6 valores possíveis: `flex-start`, `flex-end`, `center`, `space-around`, `space-between` e `stretch`.

align-content

- `flex-start`: [Exemplo06](#)

```
.flex {  
  display: flex;  
  flex-wrap: wrap;  
  width: 400px;  
  height: 120px;  
  padding: 5px;  
  align-content: flex-start;  
  border: 1px solid black;  
}  
  
.item {  
  border: 1px solid black;  
  width: 100px;  
}
```

align-content

- [Exemplo Geral](#)
- Vejamos como ficam alinhados todos os elementos de acordo com cada valor de propriedade

align-content vs align-items

- Podem parecer equivalentes, mas produzem efeitos diferentes.
- align-content atua no espaço extra entre as linhas (coluna).
 - considera o espaço no eixo `cross`
- align-items atua alinhando os elementos dentro de cada linha(coluna).
 - considera o eixo `cross`

Comparação