

# Flexbox

Flexbox é o nome abreviado para o módulo CSS [Flexible Box Layout] , projetado para facilitar o layout dos elementos em uma dimensão - como uma linha ou uma coluna. Para usar o flexbox, você aplica `display: flex` ao elemento **pai** dos elementos que deseja dispor; todos os seus **filhos** diretos se tornam *itens flexíveis*.

```
<!-- Código HTML de exemplo-->
<div class = "pai">
  <div class = "filho"></div>
  <div class = "filho"></div>
  <div class = "filho"></div>
</div>
```

```
/* Código CSS de exemplo */
.filho{
  width: 100px;
  height: 150px;
  margin: 1%;
  background-color: tomato;
}
```

## Elemento pai

Um contêiner se torna flexível definindo a propriedade `display` como `flex`:

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

Por padrão todos os "filhos" automaticamente passarão a ter a mesma altura (*height*) que a do elemento "filho" mais alto, mas isso só funciona se os "filhos não tiverem o *height* definido ou estiverem com o valor *auto*.

## Propriedades do elemento pai

### flex-direction

Define em qual direção o container deve empilhar os itens.

- `column`: empilha os itens flexíveis verticalmente (de cima para baixo).
- `column-reverse`: empilha os itens flexíveis verticalmente (mas de baixo para cima).
- `row`: (padrão) empilha os itens flexíveis horizontalmente (da esquerda para a direita).
- `row-reverse`: empilha os itens flexíveis horizontalmente (mas da direita para a esquerda).

```
.pai {  
  display: flex;  
  flex-direction: column;  
  flex-direction: column-reverse;  
  flex-direction: row;  
  flex-direction: row-reverse;  
}
```

## flex-wrap

Define se os itens flexíveis que ultrapassem a largura do container passarão para a próxima linha.

- `wrap`: especifica que os itens flexíveis passarão para a próxima linha, se necessário.
- `nowrap`: (padrão) especifica que os itens flexíveis não passarão para a próxima linha.
- `wrap-reverse`: especifica que os itens flexíveis passarão para a próxima , se necessário, na ordem inversa.

```
.pai {  
  display: flex;  
  flex-wrap: wrap;  
  flex-wrap: nowrap;  
  flex-wrap: wrap-reverse;  
}
```

## justify-content

Usada para definir o alinhamento horizontal entre os itens flexíveis, distribuindo o espaço livre.

- `center`: alinha os itens flexíveis no centro do contêiner.
- `flex-start`: (padrão) alinha os itens flexíveis no início do contêiner.
- `flex-end`: alinha os itens flexíveis no final do contêiner.
- `space-around`: exibe os itens flexíveis com espaço antes, entre e depois dos itens.
- `space-between`: exibe os itens flexíveis com espaço entre as colunas,.

```
.pai {  
  display: flex;  
  justify-content: center;  
  justify-content: flex-start;  
  justify-content: flex-end;  
  justify-content: space-around;  
  justify-content: space-between;  
}
```

## align-items

Usada para definir o alinhamento vertical de itens flexíveis com alturas diferentes.

- `center`: alinha os itens flexíveis no meio do contêiner.
- `flex-start`: (padrão) alinha os itens flexíveis na parte superior do contêiner.
- `flex-end`: alinha os itens flexíveis na parte inferior do contêiner.
- `baseline`: alinha os itens flexíveis por suas linhas de base.
- `stretch`: estica os itens flexíveis para preencher o contêiner.

```
.pai {  
  display: flex;  
  align-items: center;  
  align-items: flex-start;  
  align-items: flex-end;  
  align-items: baseline;  
  align-items: stretch;  
}
```

## align-content

Semelhante ao **justify-content**, é usada para definir o alinhamento vertical entre os itens flexíveis, distribuindo o espaço livre.

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

- `center`: valor alinha os itens flexíveis no meio do contêiner.
- `flex-start`: (padrão) exibe as linhas flexíveis no início do contêiner.
- `flex-end`: exibe as linhas flexíveis no final do contêiner.
- `space-between`: exibe as linhas flexíveis com espaço igual entre elas.
- `space-around`: exibe as linhas flexíveis com espaço antes, entre e depois delas.
- `stretch`: estica as linhas flexíveis para ocupar o espaço restante. Para que ele funcione, é importante que os filhos tenham a altura (height) como *auto*.

```
.pai {  
  display: flex;  
  flex-wrap: wrap;  
  height: 600px;  
  align-content: center;  
  align-content: flex-start;  
  align-content: flex-end;  
  align-content: space-between;  
  align-content: space-around;  
  align-content: stretch;  
}
```

## Propriedade dos elementos filhos

Os elementos filho diretos de um contêiner flex automaticamente se tornam itens flexíveis.

### order

Por padrão, os elementos filhos são mostrados em sua ordem de origem e tem por padrão **order: 0**. Para mudar um elemento filho de lugar, um número maior que 0 (o elemento vai para a direita) ou um número menor que 0 (o elemento vai para a esquerda).

```
.filho-2{  
  order: -1;  
}
```

### flex-grow

Essa propriedade faz com que o elemento cresça, somando ao tamanho do elemento o valor do espaço vazio. Ele aceita um valor número maior que 0 que define a proporção que ele irá ter do espaço vazio, se todos os elementos tiverem **flex-grow: 1**, o espaço vazio irá ser distribuído de forma igual para todos, se um dos elementos tiver **flex-grow: 2**, este irá receber o dobro do espaço vazio a mais que os outros.

```
.filho-2{  
  flex-grow: 2;  
}
```

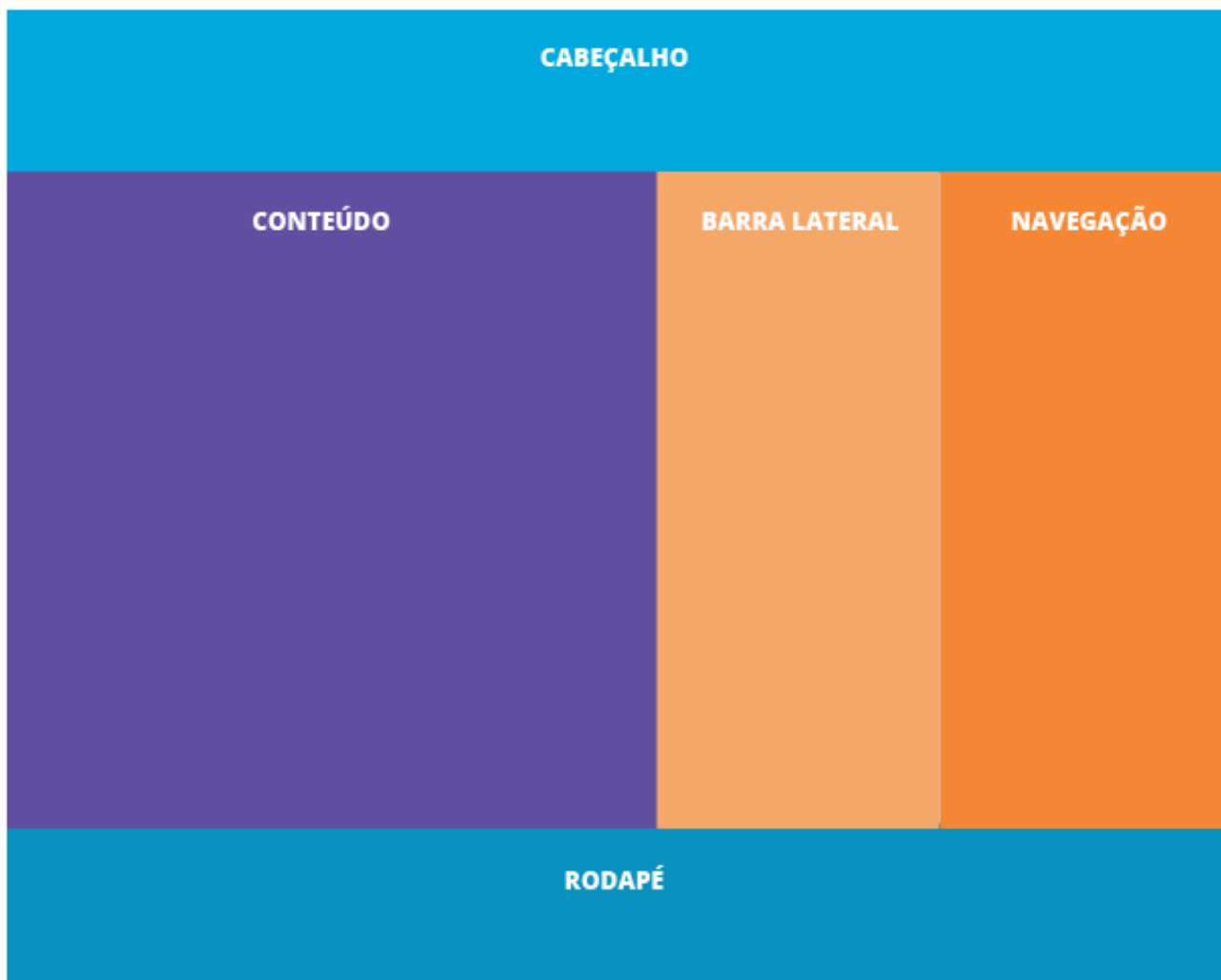
### align-self

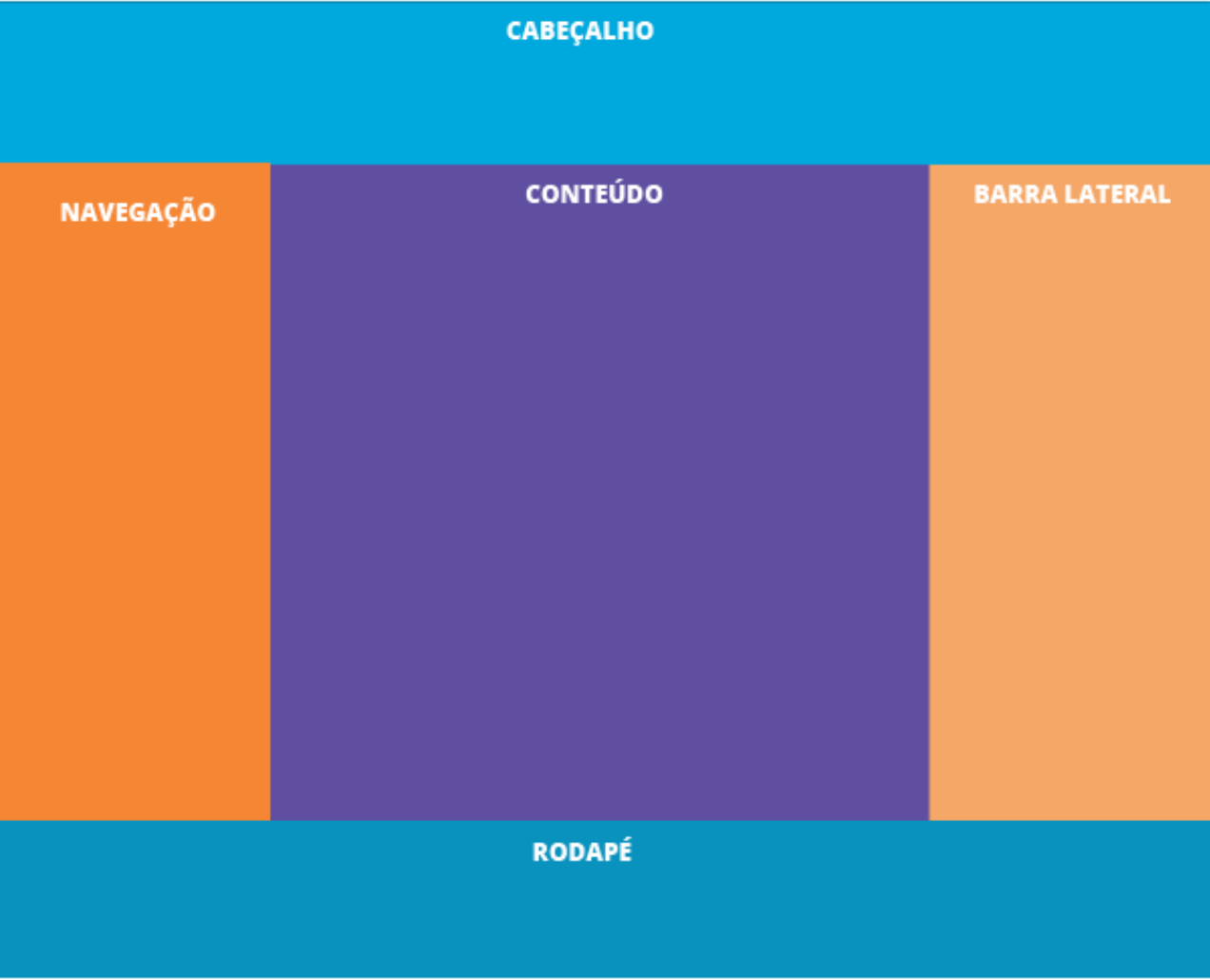
Muda o alinhamento padrão do elemento ou definido pelo `align-item` para que ele fique flexível. Usa-se os mesmos valores que os utilizados no `align-item`.

```
.filho-2{  
  align-self: center;  
  align-self: flex-start;  
  align-self: flex-end;  
  align-self: baseline;  
  align-self: stretch;  
}
```

## Atividades de fixação

Reproduza os seguintes layouts de tela.





**NAVEGAÇÃO**

**CABEÇALHO**

**CONTEÚDO**

**RODAPÉ**