
Facens

Flexbox

Prof. Me. Edson Martin Feitosa

Facens

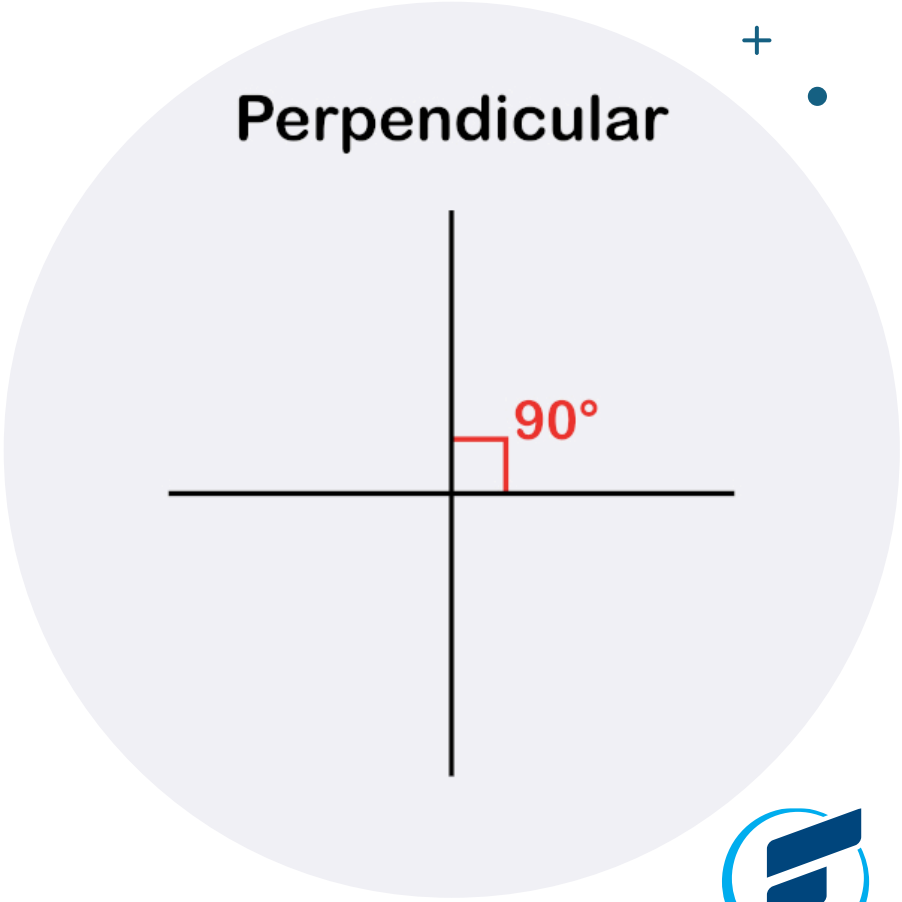
Flexbox

- **O Flexible Box Module, geralmente chamado de flexbox**, foi projetado tanto como um modelo de layout unidimensional quanto como um método capaz **de organizar espacialmente os elementos em uma interface, além de possuir capacidades avançadas de alinhamento**.
- Quando se descreve o flexbox como sendo unidimensional, enfatiza-se o fato de que ele lida com o layout em uma dimensão de cada vez - seja uma linha ou uma coluna.
- Por um longo tempo, as **únicas ferramentas compatíveis entre browsers** disponíveis para criação de layouts CSS eram coisas como **floats e posicionamento**. Estas são boas e funcionam, mas em alguns casos também são limitadas e frustrantes.

Facens

Eixos do flexbox

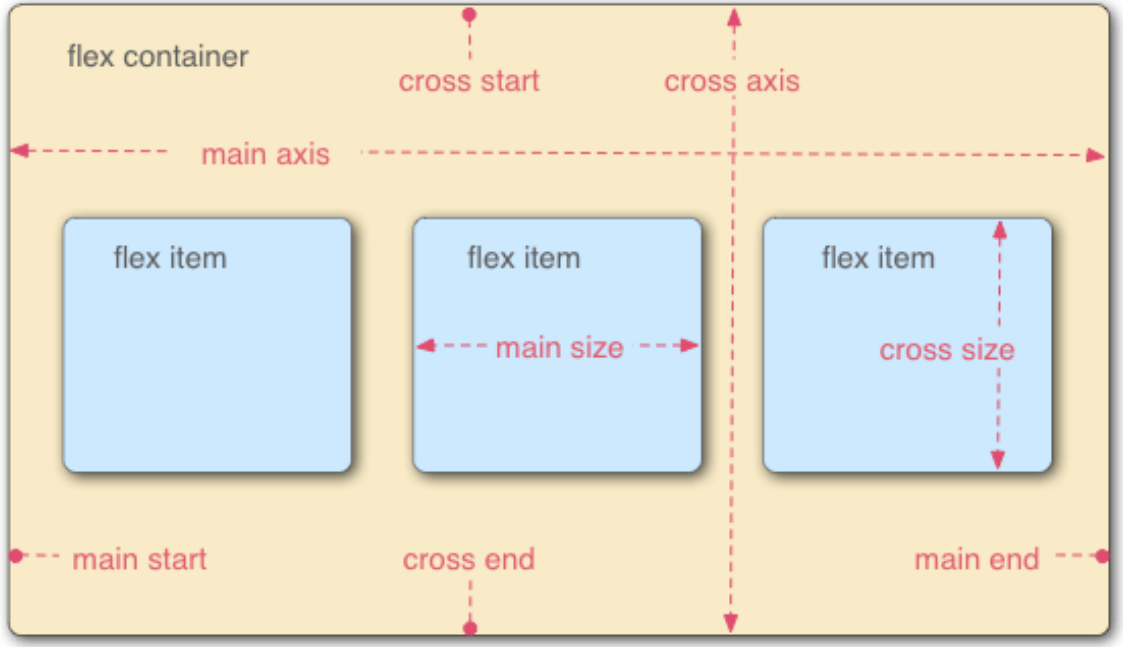
- É preciso ter em mente que **todas as operações realizadas relacionam-se a dois eixos: o eixo principal e o eixo transversal**.
- O **eixo principal é definido através da propriedade flex-direction** e o **eixo transversal** encontra-se na **direção perpendicular a ele**.
- Como esses eixos são as engrenagens fundamentais do flexbox é necessário compreender minuciosamente o seu funcionamento.



Facens

Terminologia

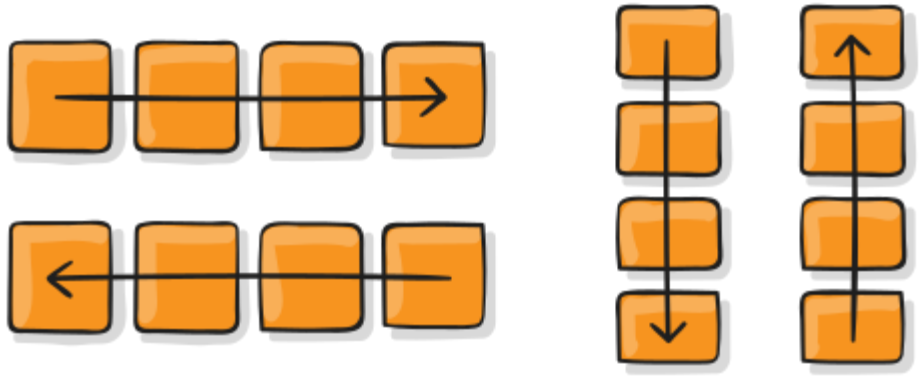
- O **main axis (Eixo principal)** é o eixo que corre na direção em que os flex items estão dispostos. **O início e o fim do eixo é chamado main start e main end**.
- O **cross axis (eixo perpendicular)** que corre na direção em que os flex items são dispostos. **O início e o fim deste eixo são chamados de cross start e cross end.**
- O elemento pai que possui display: flex configurado é chamado de **flex container**.
- Os itens iniciados como flexible boxes dentro do flex container são chamados **flex items.**



Facens

Eixo Principal

- Conforme descrito, a propriedade flex-direction define a direção do eixo principal e pode tem quatro valores possíveis:
  - row
  - row-reverse
  - column
  - column-reverse



```
.container {
  flex-direction: row | row-reverse | column | column-
reverse;
}
```

Facens

Row e row-reverse

- Se o valor escolhido for row (linha) ou row-reverse (linha reversa), seu eixo principal se moverá ao longo da linha — na direção inline.
- Row: da esquerda para a direita (padrão do flexbox)

- Row-reverse: da direita para a esquerda

Facens

Column e Column-reverse

- Se o valor escolhido for column (coluna) ou column-reverse (coluna reversa) e o eixo principal se moverá do topo até o fim da página — na direção block.

Column: De cima pra baixo

Column-reverse: De baixo para cima

Facens

Eixo Transversal

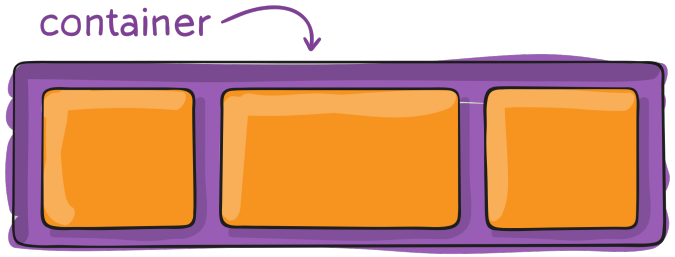
O eixo transversal é perpendicular ao eixo principal, logo, se a propriedade flex-direction estiver definida nas linhas, como row ou row-reverse, o eixo transversal estará na direção das colunas, como column ou column-reverse.

Se o eixo principal for definido nas colunas, como column ou column-reverse, então o eixo transversal estará na direção das linhas, como row ou row-reverse.

Facens

Container Flex

- **A área de um documento que faz uso do flexbox é chamada de container flex**. Para criar essa estrutura, define-se o valor da propriedade display do elemento representado pelo contêiner como flex ou inline-flex. Desse modo, os elementos-filhos desse contêiner tornar-se-ão do tipo flex.



Facens

Exemplo - Container

Se houver mais itens do que é possível caber no container, não haverá uma quebra de linha; ao invés disso, irão ultrapassar o limite horizontal da página. Se alguns elementos forem mais altos que outros, todos os itens se estenderão ao longo do eixo transversal para preencher seu tamanho total.

CSS

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

HTML

```
<div class="box">
  <div>One</div>
  <div>Two</div>
  <div>Three
    <br>has
    <br>extra
    <br>text
  </div>
</div>
```

Resultado

Facens

Exemplo – flex-direction

CSS

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

HTML

```
<div class="box">
  <div>One</div>
  <div>Two</div>
  <div>Three</div>
</div>
```

Resultado

Facens

Quebra de linhas

- Por padrão, todos os itens flexboxs tentarão caber em uma linha. Você pode alterar isso e permitir que os itens sejam agrupados conforme necessário.
- nowrap(padrão): todos os itens flexíveis estarão em uma linha
- wrap: os itens flexíveis serão agrupados em várias linhas, de cima para baixo.
- wrap-reverse: os itens flexíveis serão agrupados em várias linhas de baixo para cima.

```
.container { flex-wrap: nowrap | wrap | wrap-reverse; }
```

Facens

Quebra de linhas - Exemplo

CSS

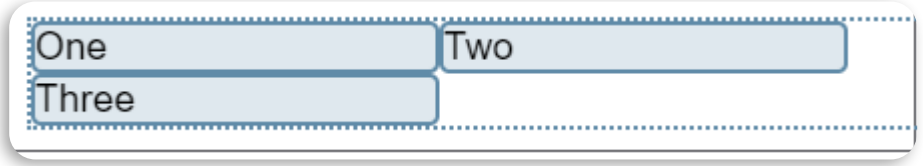
```
.box {
  display: flex;
  flex-wrap: wrap;
}
```

Para **gerar a quebra automática das linhas adicione a propriedade flex-wrap com o valor wrap**. Assim, se elementos forem muito grandes para serem exibidos em uma única linha, eles serão agrupados em outras linhas.

HTML

```
<div class="box">
  <div>One</div>
  <div>Two</div>
  <div>Three</div>
</div>
```

Resultado



Facens

flex-flow

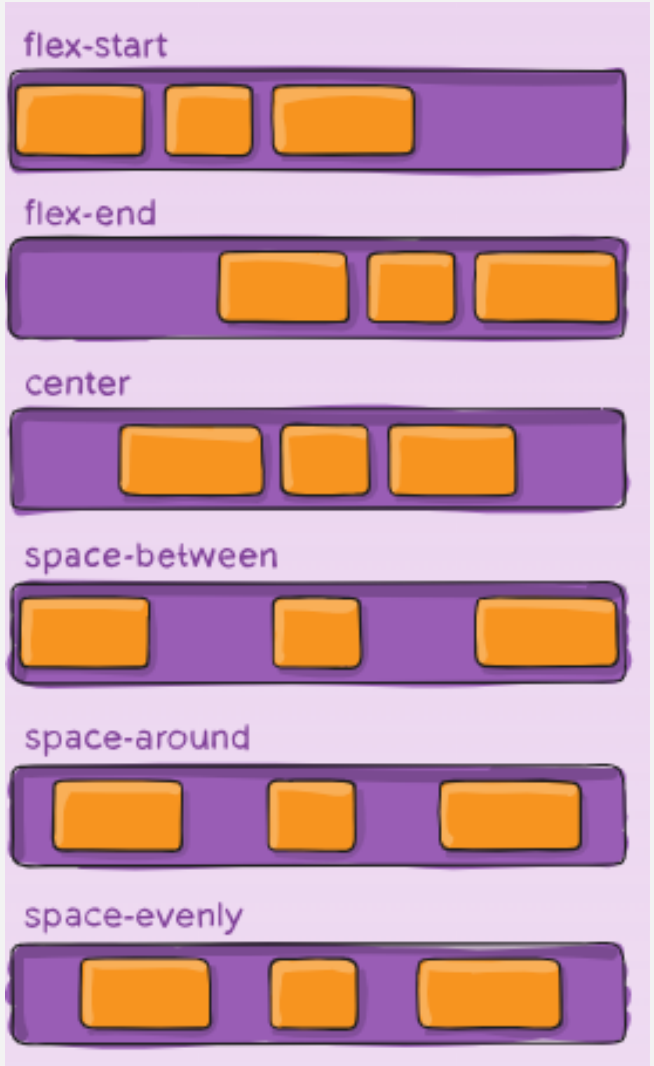
- Este é um atalho para as propriedades flex-directione flex-wrap, que juntas definem os eixos principal e cruzado do flex container.
- O valor padrão é row nowrap.

```
.container {
  flex-flow: row nowrap;
}
```

=

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

Facens

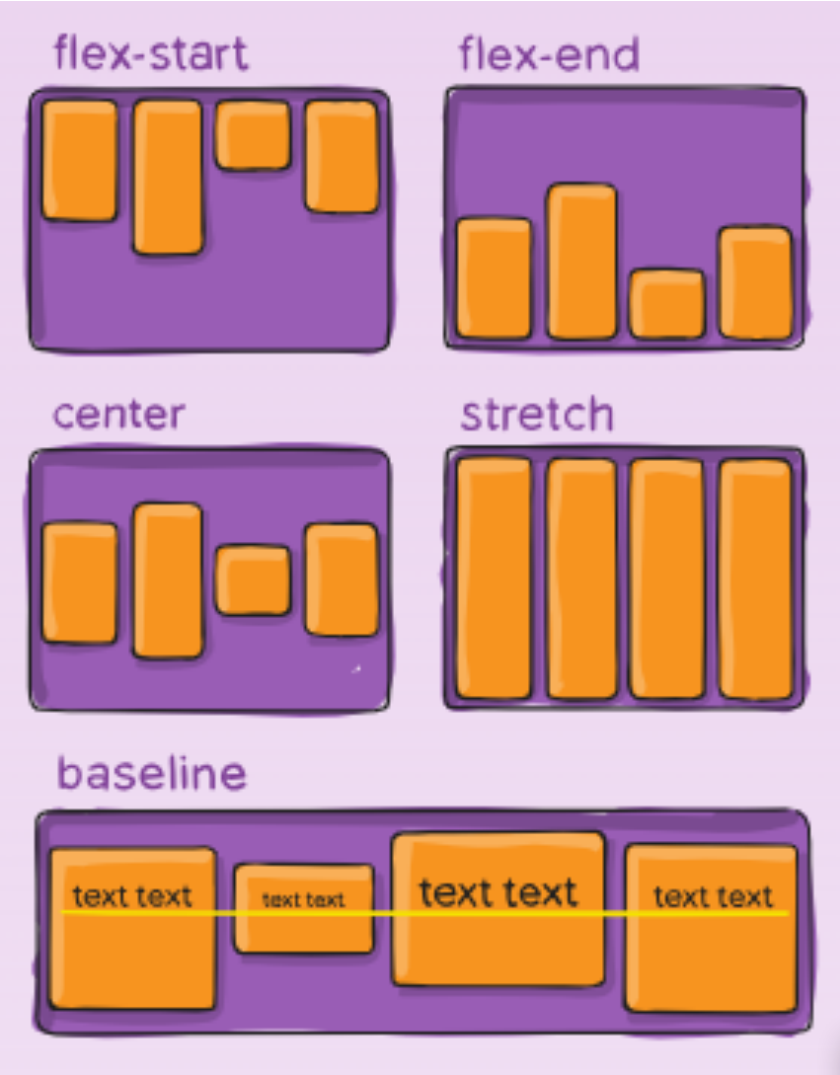


Justificar conteúdo

- A propriedade justify-content é empregada para alinhar os elementos ao longo do eixo principal, cuja direção (row ou column) é definida a partir da propriedade flex-direction. O valor inicial é flex-start.

```
.container { justify-content: flex-start | flex-end | center |
space-between | space-around | space-evenly | start | end |
left | right ... + safe | unsafe; }
```

Facens



Alinhar itens

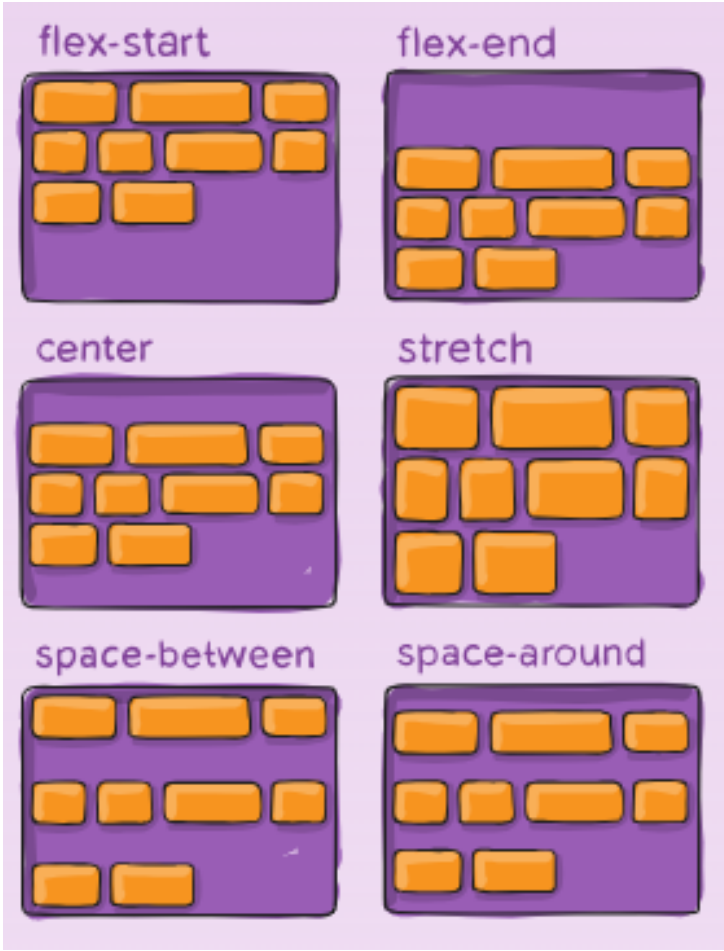
- A propriedade align-items irá alinhar os elementos no eixo transversal.
- O valor inicial desta propriedade é stretch e é por essa razão que, por padrão, os elementos flex se estendem até a maior altura. De fato, eles se esticam para preencher o contêiner flex - o item mais alto define a altura deste.

```
.container { align-items: stretch | flex-start | flex-end |
center | baseline | first baseline | last baseline | start | end |
self-start | self-end + ... safe | unsafe; }
```

Facens

Alinhar Container

- Alinha as linhas de um flex container quando há espaço extra no eixo cruzado, semelhante a como justify-contente alinha itens individuais dentro do eixo principal.
- Esta propriedade só tem efeito em contêineres flexíveis multilinhas, onde flex-wrap está definida como wrap ou wrap-reverse). Um contêiner flexível de linha única (ou seja flex-wrap, onde está definido com seu valor padrão no-wrap) não refletirá align-content.

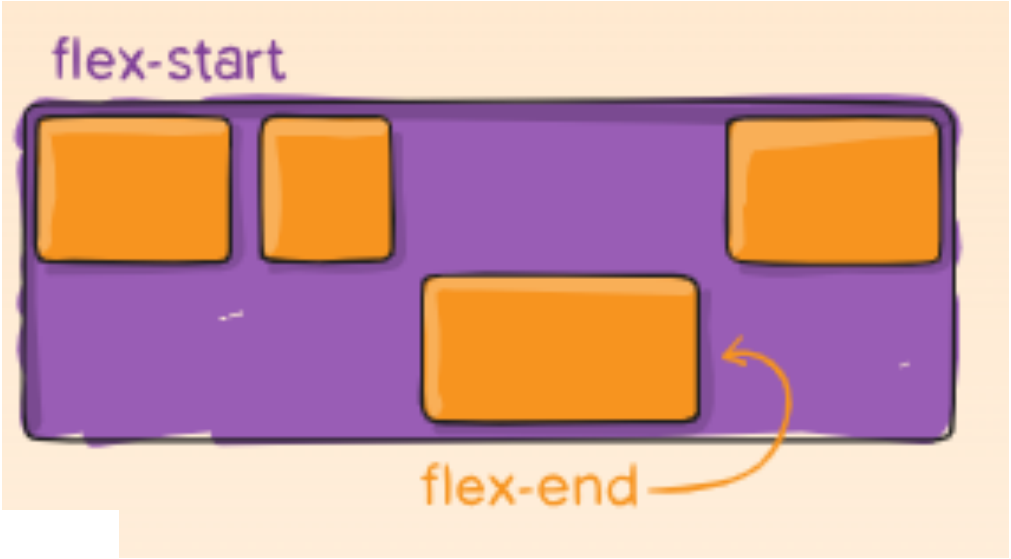



```
.container {
  align-content: flex-start | flex-end | center | space-between | space-around | space-evenly | stretch | start | end | baseline | first baseline | last baseline + ... safe | unsafe;
}
```

Facens

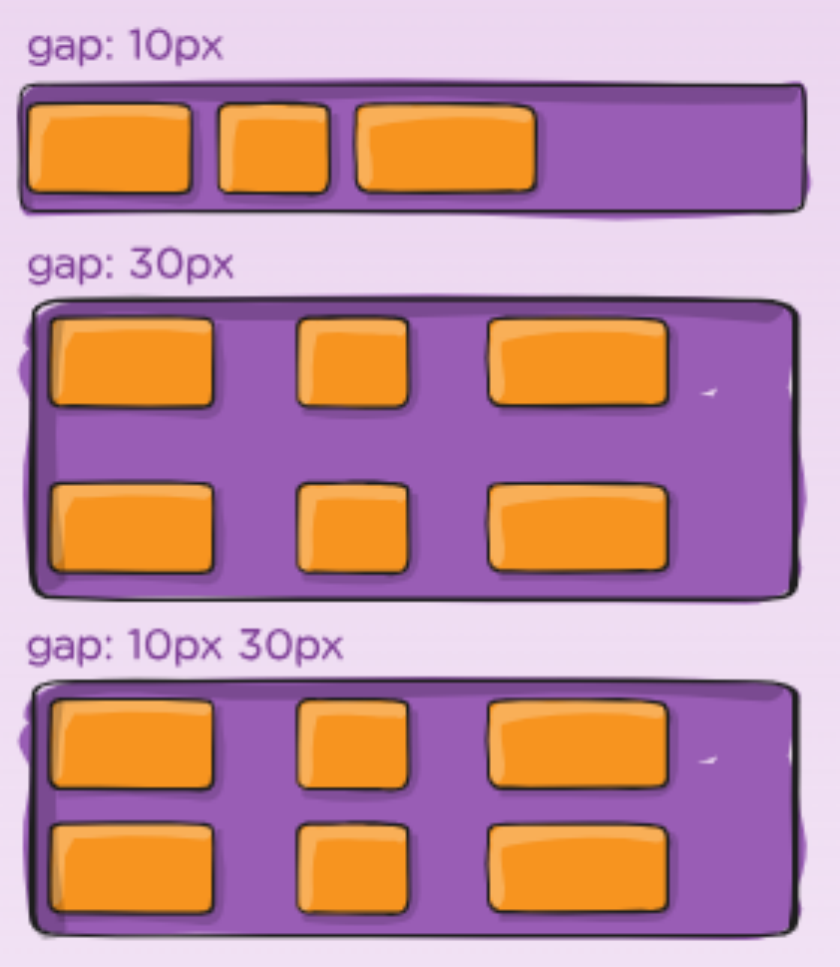
Auto Alinhar

- Isso permite que o alinhamento padrão (ou aquele especificado por align-items) seja substituído por itens flexíveis individuais.




```
.item {
  align-self: auto | flex-start | flex-end | center | baseline |
stretch;
}
```

Facens



Lacunas

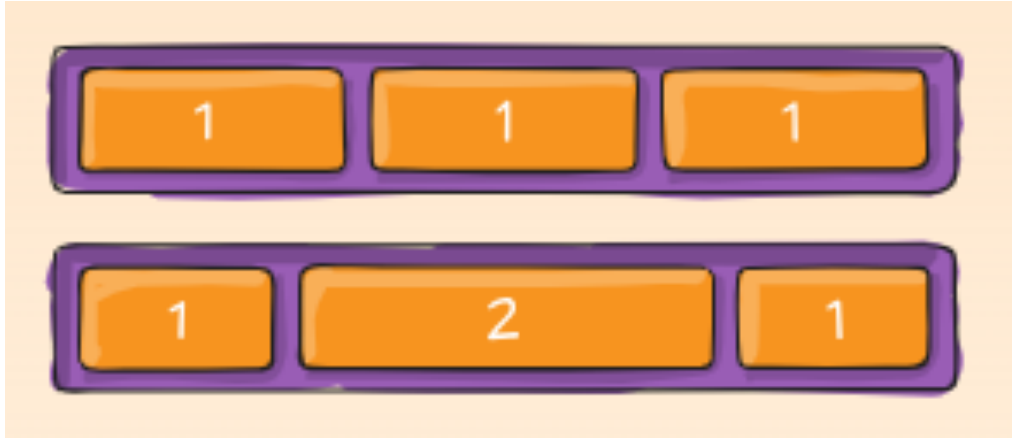
- A propriedade gap controla explicitamente o espaço entre os itens flexíveis. Aplica-se esse espaçamento apenas entre itens que não estão nas bordas externas.

```
.container { display: flex; ... gap: 10px; gap: 10px 20px; /* row-gap column gap */ row-gap: 10px; column-gap: 20px; }
```

Facens

Crescimento Flexível (flex-grow)

- Define a capacidade de um item flexível crescer, se necessário.
- Aceita um valor sem unidade que serve como proporção. Ele determina a quantidade de espaço disponível dentro do flex container que o item deve ocupar.
- Se todos os itens tiverem sido flex-growdefinidos como 1, o espaço restante no contêiner será distribuído igualmente para todos os filhos. Se um dos filhos tiver o valor 2, esse filho ocuparia o dobro do espaço de qualquer um dos outros (ou tentará, pelo menos).




```
.item { flex-grow: 2; /* default 0 */}
```

Facens

Veja um exemplo de layout com flexbox

https://css-tricks.com/snippets/css/a-guide-to-flexbox/#aa-examples

Facens

GitHub com exemplos da aula

https://github.com/prof-amauri-facens/linguagens_programacao_flexbox

Facens