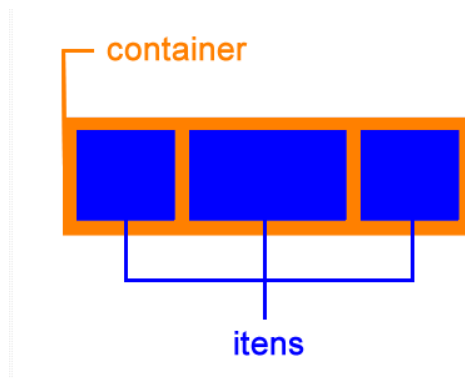


Flexbox: Resumo

❖ O que é o flexbox?

O **Flexbox** é um conceito do CSS3 que visa organizar os elementos de uma página HTML dentro de seus containers de forma dinâmica. Ou seja, independente das suas dimensões eles sempre manterão um layout flexível dentro do seu elemento pai (Geralmente chamado de contêiner), reorganizando-se de acordo com a necessidade.



Portanto, para utilizar esse recurso é necessário ter no HTML ao menos um elemento (container) contendo outros (itens), como no código abaixo:

```
1 <div class="container">
2   <div class="item item1"></div>
3   <div class="item item2"></div>
4   <div class="item item3"></div>
5 </div>
```

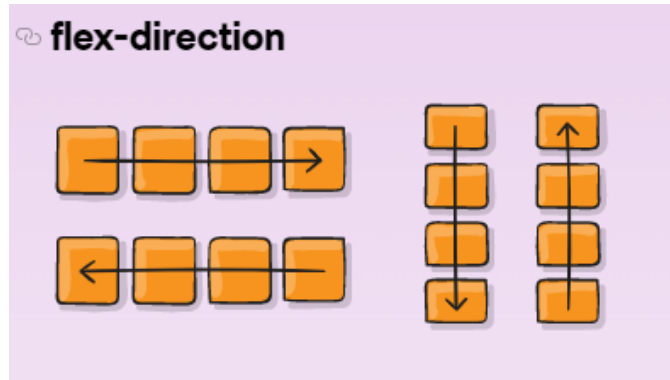
❖ Display:

O primeiro passo para **utilizar o Flexbox** é definir a propriedade **display** do container com o valor **flex**. Isso é necessário para que as demais propriedades apresentem o resultado esperado.

```
1 .container {
2   display: flex;
3 }
```

❖ Flex-Direction:

A propriedade ***flex-direction*** deve ser aplicada ao container definindo assim a direção em que os itens flexíveis são colocados no contêiner flexível.

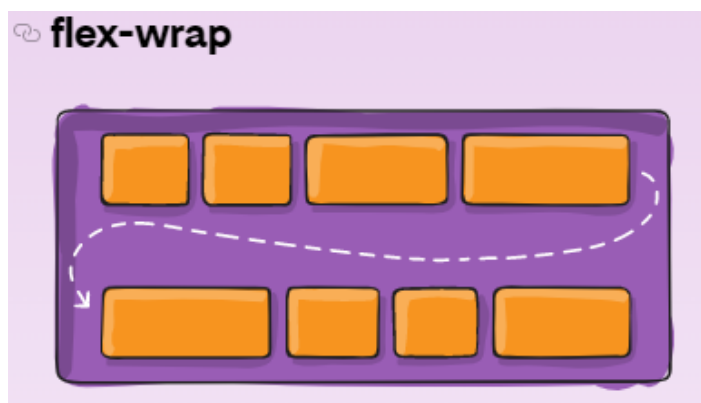


```
.container {  
  flex-direction: row | row-reverse | column | column-reverse;  
}
```

- row(padão): da esquerda para a direita em ltr; direita para a esquerda emrtl
- row-reverse: da direita para a esquerda em ltr; da esquerda para a direita emrtl
- column: igual, rowmas de cima para baixo
- column-reverse: igual, row-reversemas de baixo para cima

❖ Flex-Wrap:

Por padrão, todos os itens flexíveis tentarão caber em uma linha. Você pode alterar isso e permitir que os itens sejam agrupados conforme necessário com essa propriedade.



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

- 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.

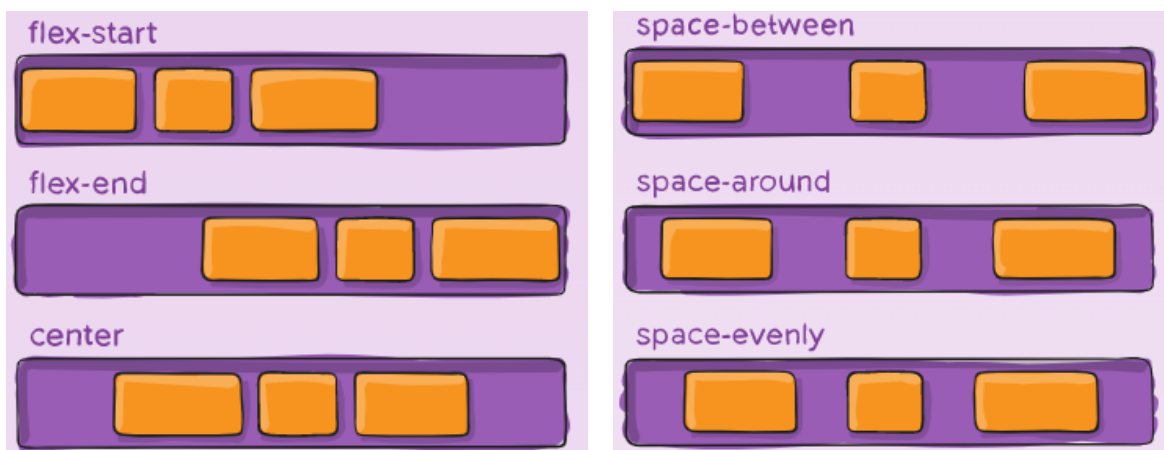
❖ Flex-flow:

Esta é uma abreviação para as propriedades `flex-direction` e `flex-wrap`, que juntas definem os eixos principal e cruzado do contêiner flexível.

```
.container {
  flex-flow: column wrap;
}
```

❖ Justify content:

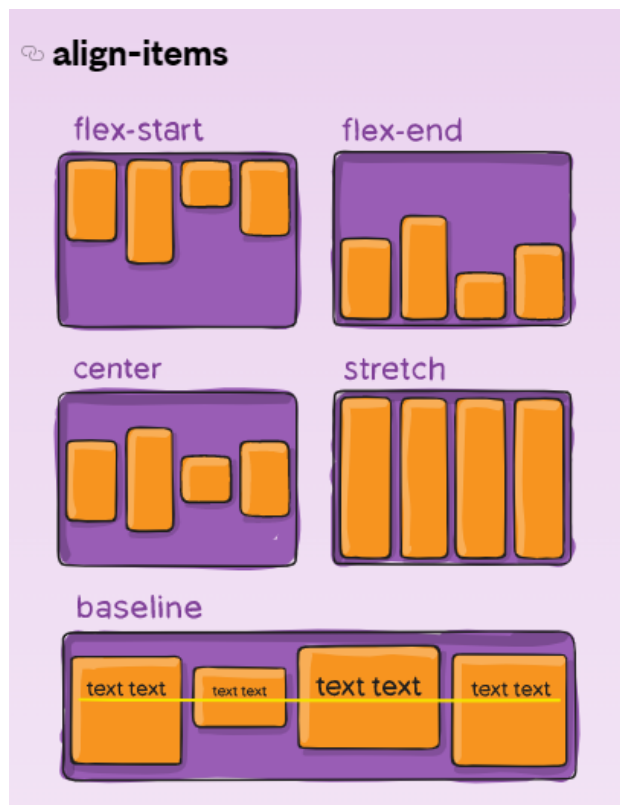
Isso define o alinhamento ao longo do eixo principal. Ele ajuda a distribuir espaço livre extra quando todos os itens flexíveis em uma linha são inflexíveis ou são flexíveis, mas atingiram seu tamanho máximo. Ele também exerce algum controle sobre o alinhamento dos itens quando eles ultrapassam a linha.



- **flex-start(padrão)**: os itens são empacotados no início da direção flexível.
- **flex-end**: os itens são empacotados no final da direção flexível.
- **left**: os itens são empacotados na borda esquerda do contêiner
- **right**: os itens são empacotados na borda direita do contêiner
- **center**: os itens são centralizados ao longo da linha
- **space-between**: os itens são distribuídos uniformemente na linha; primeiro item está na linha inicial, último item na linha final
- **space-around**: os itens são distribuídos uniformemente na linha com espaço igual ao redor deles
- **space-evenly**: os itens são distribuídos de forma que o espaçamento entre quaisquer dois itens (e o espaço até as bordas) seja igual.

❖ Align-items:

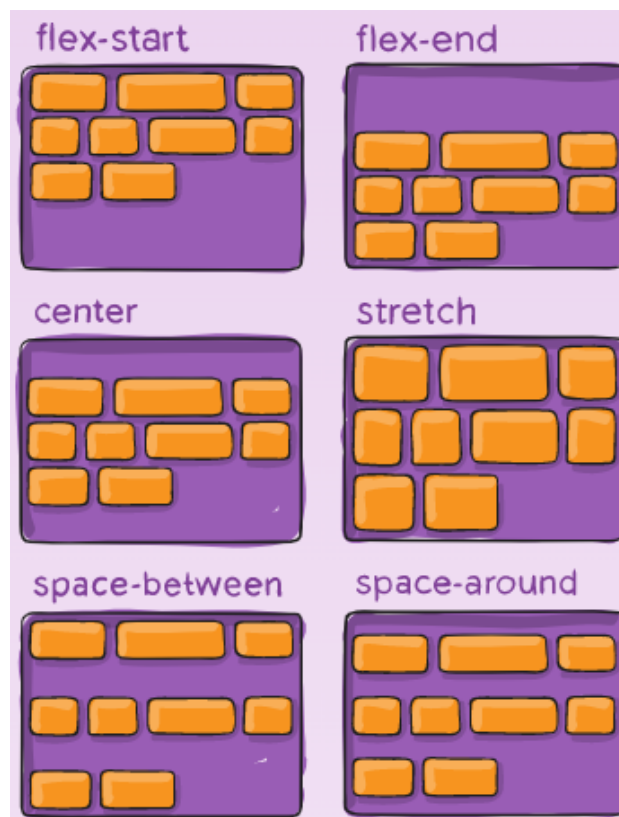
Isso define o comportamento padrão de como os itens flexíveis são dispostos ao longo do **eixo cruzado** na linha atual.



- `stretch`(padrão): esticar para encher o contêiner (ainda respeita `min-width`/`max-width`)
- `flex-start`/ `start`/ `self-start`: os itens são colocados no início do eixo cruzado. A diferença entre eles é sutil, e se trata de respeitar as `flex-direction`regras ou as `writing-mode`regras.
- `flex-end`/ `end`/ `self-end`: os itens são colocados no final do eixo cruzado. A diferença novamente é sutil e é sobre respeitar `flex-direction`regras versus `writing-mode`regras.
- `center`: os itens são centralizados no eixo cruzado
- `baseline`: os itens são alinhados como suas linhas de base se alinham

❖ **Align-content:**

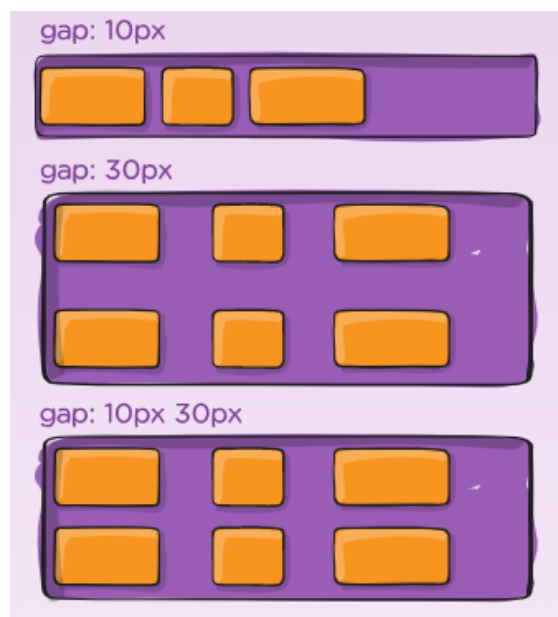
Isso alinha as linhas de um contêiner flexível quando há espaço extra no eixo cruzado



- `normal` (padrão): os itens são embalados em sua posição padrão como se nenhum valor fosse definido.
- `flex-start` / `start`: itens embalados até o início do contêiner. O (mais suportado) `flex-start` honra o `flex-direction` enquanto `start` honra a `writing-mode` direção.
- `flex-end` / `end`: itens embalados até o final do contêiner. O (mais suporte) `flex-end` honra o `flex-direction` enquanto o final honra a `writing-mode` direção.
- `center`: itens centralizados no contêiner
- `space-between`: itens distribuídos uniformemente; a primeira linha está no início do contêiner enquanto a última está no final
- `space-around`: itens distribuídos uniformemente com espaço igual ao redor de cada linha
- `space-evenly`: os itens são distribuídos uniformemente com espaço igual ao redor deles
- `stretch`: as linhas se estendem para ocupar o espaço restante

❖ `gap`, `row-gap`, `column-gap`:

controla explicitamente o espaço entre os itens flexíveis. Aplica-se este espaçamento *apenas entre os itens* e não nas bordas externas.



❖ Align-self:

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

```
.item {  
  align-self: auto | flex-start | flex-end | center | ba  
}
```

