# FLEXBOX

Criação de layouts responsivos sem a necessidade de definição de valores fixos.

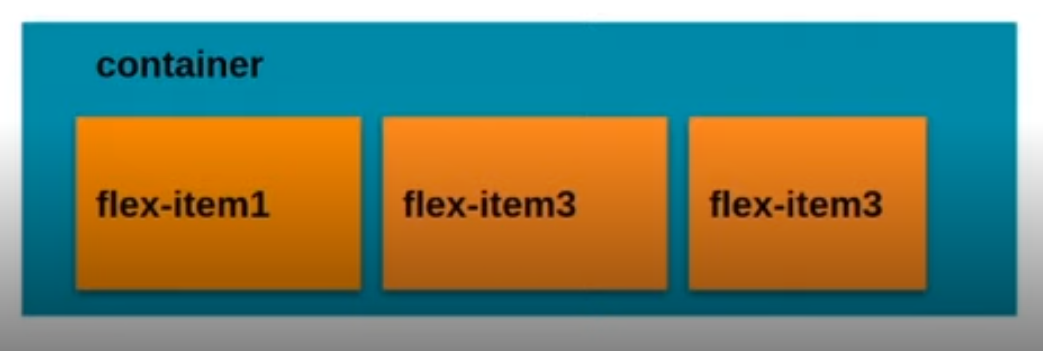
## O que é o Flexbox?

## Flex Container x Flex item

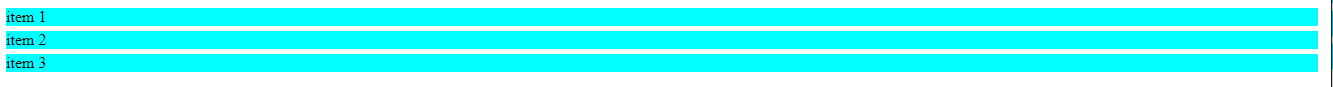
Flex container é a tag pai que envolve os filhos, caso queira alinhar ou trabalhar com os filhos da tag, use **“display:flex”** no **pai.**



Flex items são elementos filhos diretos do Flex container. Eles também podem se tornar flex container.



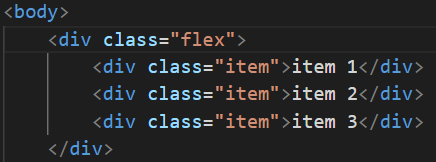
## Propriedades iniciais



Observe que por padrão, uma **div** inicia em um estágio BLOCK (ocupa toda a linha e 100% do container)

### Display:flex

O display flex permite que os items possam ficar um ao lado do outro, lembre-se de colocar esse comando no pai. Para afetar os filhos. Esse comando mostra aos conteúdos que eles só podem ocupar o tamanho correspondente ao seu conteúdo.

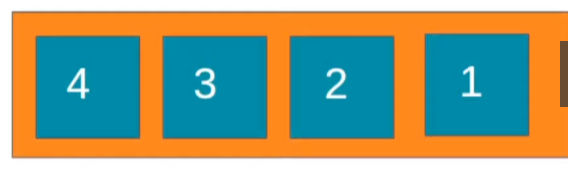
### Flex-direction

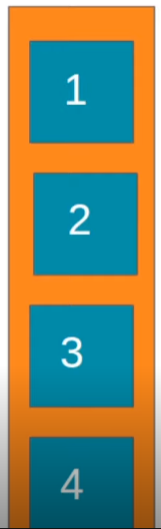
Estabelece a direção que o container irá ocupar. Forma o eixo do container.

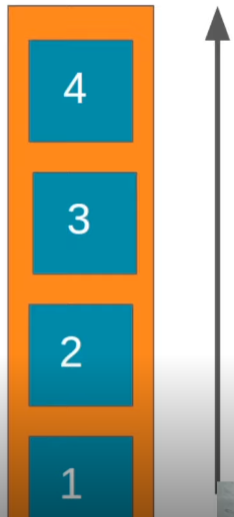
O eixo por padrão é o **ROW** = Da esquerda para a direita.



O **row-reverse** = faz o sentido oposto a direção do texto por padrão.



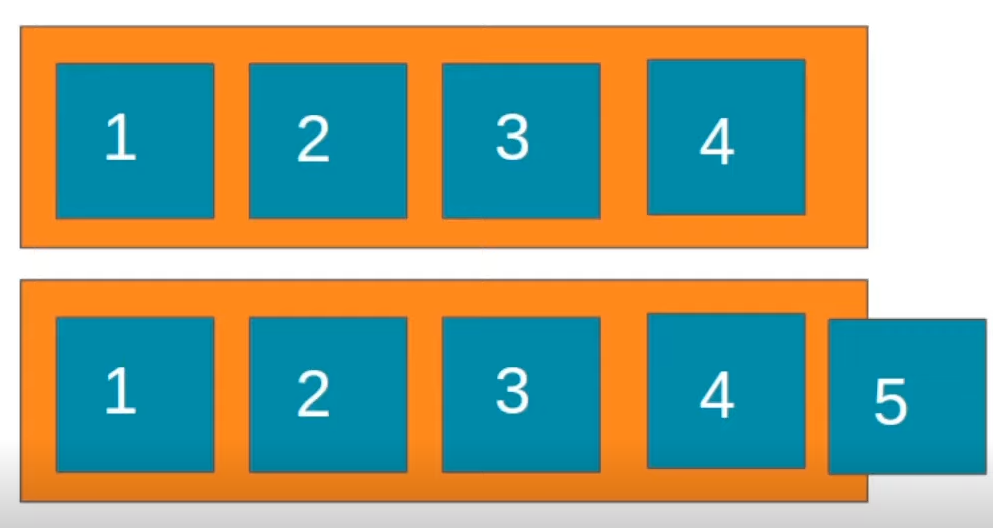
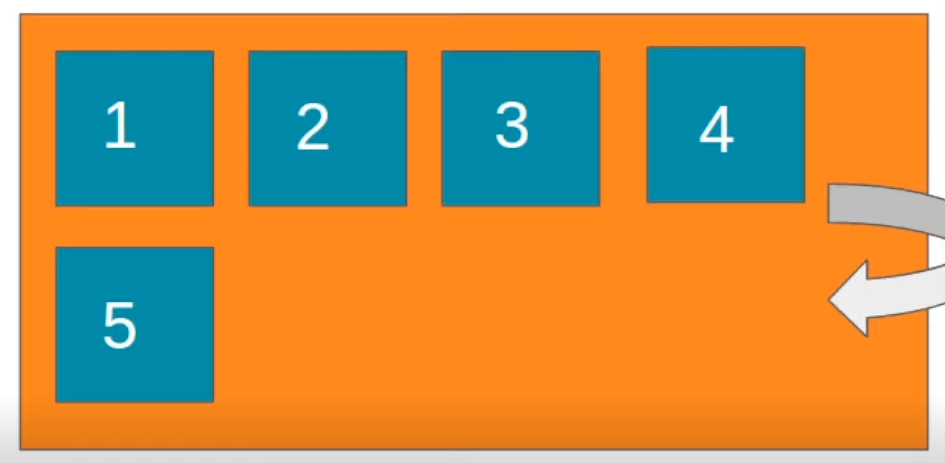
O **column** = faz a direção adotar eixo em coluna 

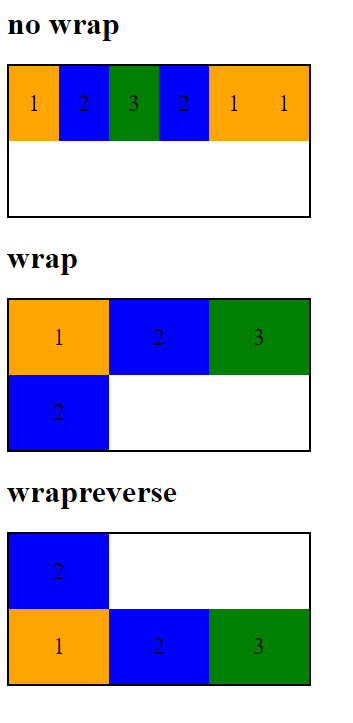
**O column-reverse** = Ordenação inversa do anterior. 

### Flex-wrap

Propriedade que define se os items devem ou não quebrar a linha , por padrão eles não quebram linhas. Dessa forma, os itens passam a ser compactados além do limite do conteúdo. Veja a diferença de com e sem.

SEM COM

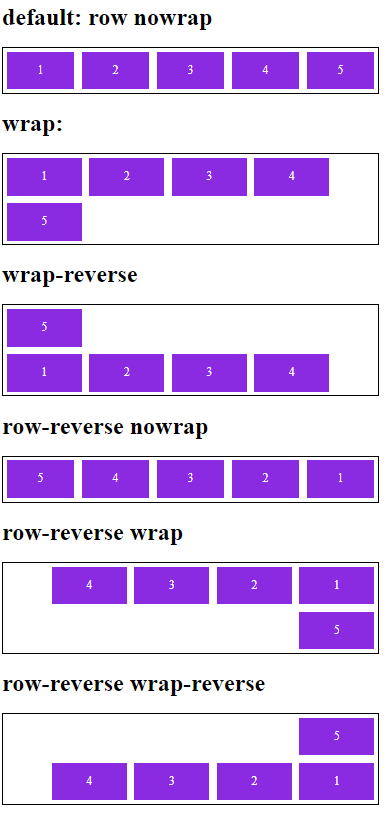
 < ==== **Display row**

### Flex-flow

Atalho para as propriedades flex-direction e flex-wrap

Porem seu uso não é tão comum, visto que, quando mudamos o flex-direction para column, mantemos o padrão do flex-wrap que é nowrap..

Exemplo de código : 



### Justify-content

Trabalha não só com o alinhamento dos itens dentro do container, mas a distribuição de espaço entre eles.

Caso os itens do container estejam ocupando 100% dele, não faz diferença alguma.

Flex-start : Alinhamento no inicio do container, por padrão já é assim.

‘ 

Flex-end: Alinhamento no final do container, na forma column só muda a direção.



Center: ao centro do container



Space-between: espaçamento igual entre os elementos , primeiro e final perto do container



Space-around: espaçamento do meio é 2 vezes maior que o do final.



## Align items

Se adiciona no pai.

Trata do alinhamento dos flex itens de acordo com o eixo do container.

Alinhamento diferente para quando os itens estão em colunas ou em linhas.

Permite o alinhamento central no eixo vertical

### Align-items em linha:

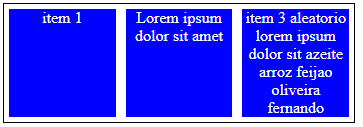
Por padrão com flex 1



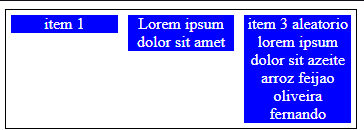
Center: alinhamento dos itens ao centro



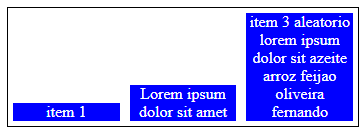
Stretch: padrão, os flex itens crescem igualmente.



Flex-start: alinhamento dos itens no início



Flex-end: alinhamento dos itens no final

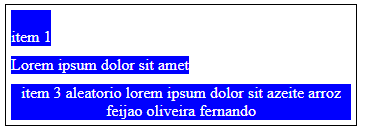


Baseline: alinhamento de acordo com a linha base da tipografia dos itens

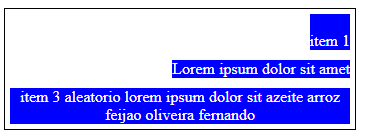
Muito parecido com flex start

### Align-items em coluna:

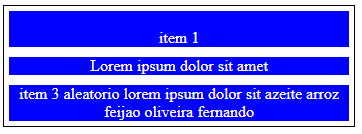
Flex-start



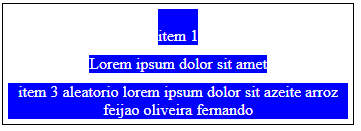
Flex-end



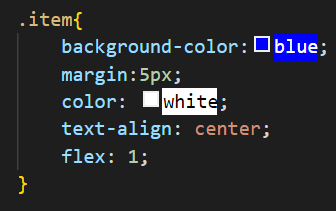
Stretch



Center

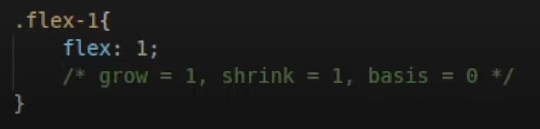
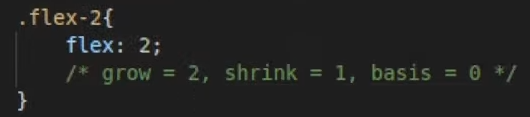


## FLEX Uma abreviação para grow shrink basis

Exemplo de código :  flex : 1

Sem flex e com flex

## Align-content

É colocado no pai

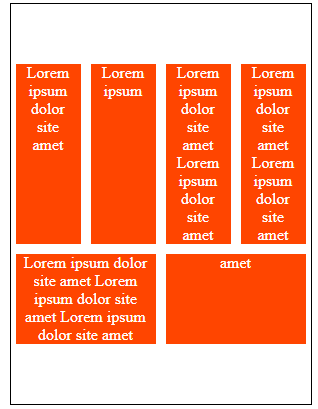
É a propriedade responsável por tratar o alinhnamento das linhas do container em relação ao eixo vertical do container

Para funcionar:

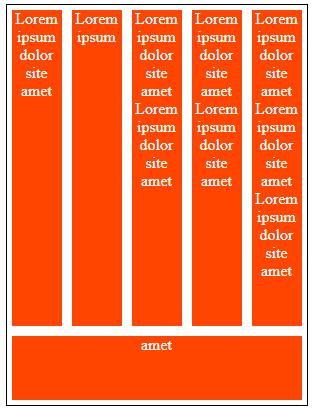
Precisamos que o container quebra as linhas

A altura do container seja maior que a soma das linhas do itens

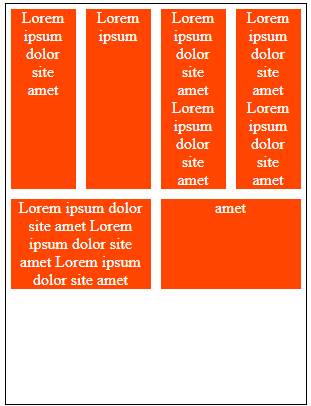
Center: alinhamento dos itens ao centro



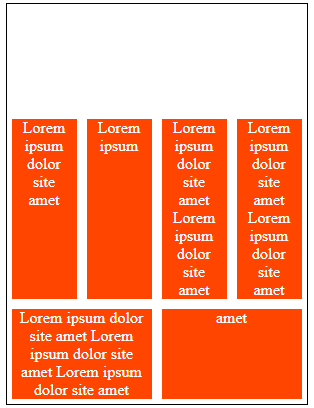
Stretch: PADRÃO, itens crescem igualmente



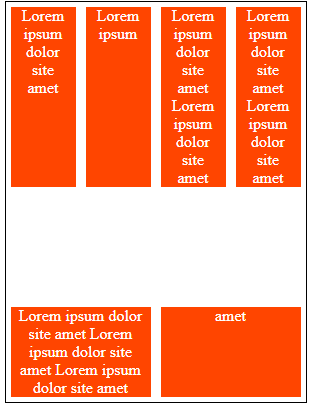
Flex-start: alinhamento dos itens no início

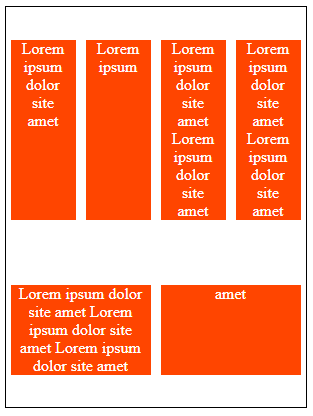


Flex-end: alinhamento dos itens no final



Space- between: cria um espaçamento igual entre os elementos



 Space-around: Os espaçamentos do meio são duas vezes maiores que o inicio e final

## Flex-grow

Organiza os itens no container e atribui peso a eles com fator de crescimento.

Lembre-se que o texto também afeta o tamanho.

Sem flex-grow 

Com flex-grow:1 no item A : 

Com flex-grow:1 no item A, e flex-grow: 2 no item BBB. Observe que ele passa a comparar o peso de cada item. 

Veja que o espaço que sobra pro cc é igual a 0 , o que sobra pro a é 1x maior e o que sobra pra BBB é 2x maior que o A.

## Flex-basis

estabelece o tamanho inicial do item antes da distribuição de espaço restante.

## Flex-shrink

É a propriedade que estabelece a capacidade de REDUÇÃO do container filho. Ele é o oposto do basis

## Order

Usado para mudar a ordem de um filho em relação a outro.

## Align-self

Propriedade usada para alinhamento de modo individual sobre um determinado item.

Auto: (PADRÃO) Respeita a definição padrão de align-items do container

Flex-start : ao início do container

Flex-end: ao final do container

Center: relativo ao centro de acordo com o eixo

Stretch ocupa todo espaço relativo

Baseline: usa a linha base da tipografia

