

DigitalHouse >
Coding School

Eixos



**Certified Tech
Developer**
The Ultimate Degree

Índice

1. [Eixos no Flexbox](#)
2. [flex-direction](#)
3. [justify-content](#)
4. [align-items](#)
5. [align-content](#)

1 | Eixos no Flexbox

“

O Flexbox trabalha com dois eixos para desenvolver todo o seu fluxo interno:

o eixo X e o eixo Y.

Dependendo de como decidimos ordenar os elementos, chamaremos um de **main axis** e o outro de **cross axis**.

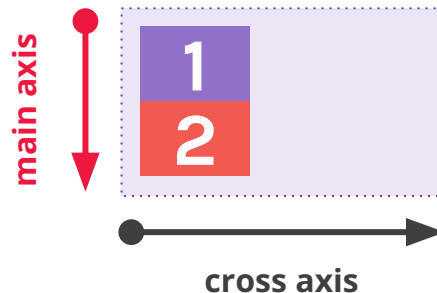
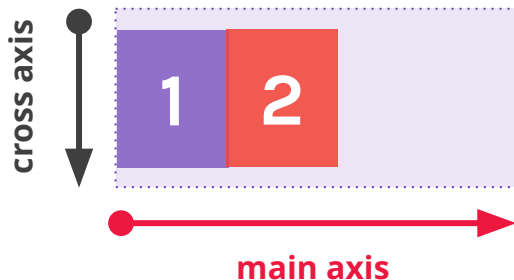
”



Eixos no Flexbox

Quando trabalhamos com um fluxo flexível, nos referimos ao **main axis** e ao **cross axis**.

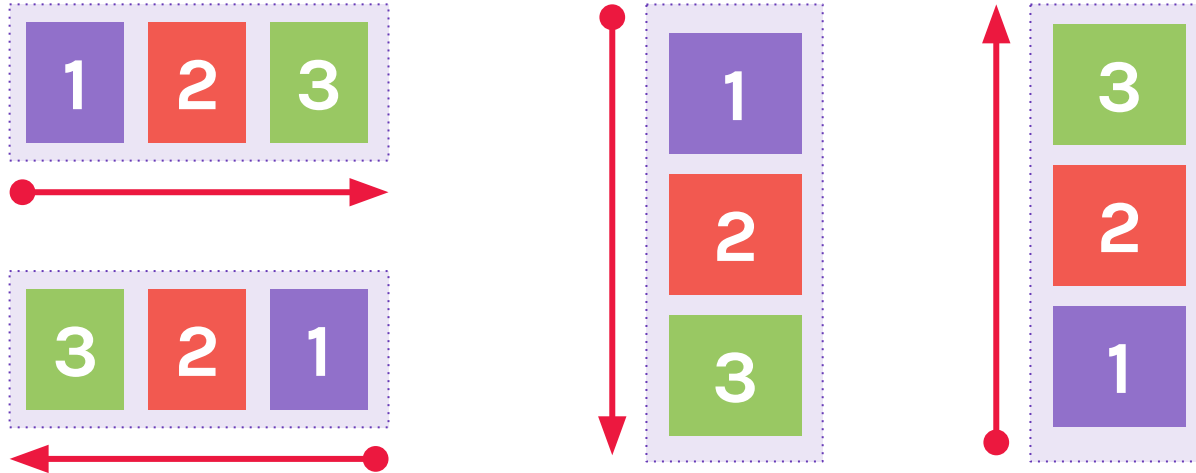
Ao definir o eixo principal do nosso contêiner flexível estamos determinando o fluxo que os elementos dentro do contêiner terão. Dependendo de qual é o eixo principal, os elementos são distribuídos em linhas horizontais ou colunas verticais.



2 | flex-direction

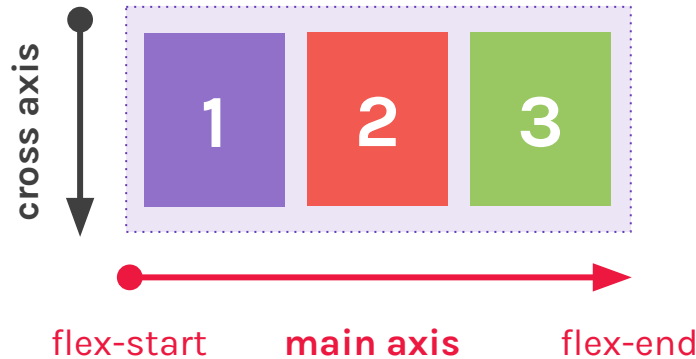
flex-direction

Com esta propriedade, definimos o eixo principal (**main axis**) do contêiner, que pode ser tanto horizontal quanto vertical. O eixo transversal (**cross axis**) será a direção perpendicular ao eixo principal.



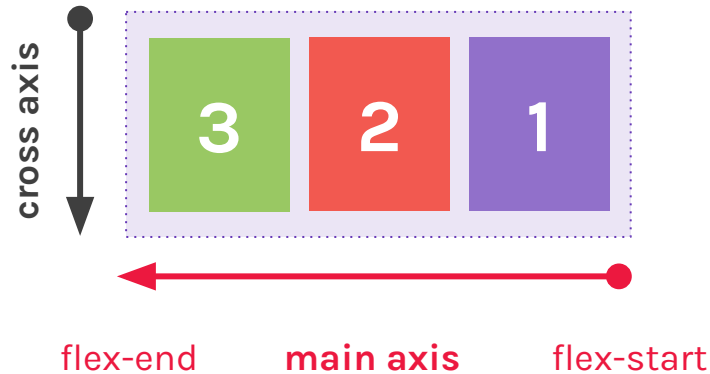
flex-direction: row

Os itens são organizados no **eixo x**, da **esquerda** para a **direita**. Se não deixarmos a propriedade flex-direction clara para o contêiner, **row** será o valor padrão.



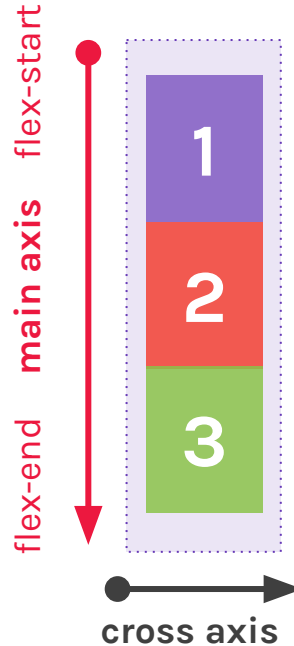
flex-direction: row-reverse

Os itens são organizados no **eixo x**, da **direita** para a **esquerda**. Neste caso, estamos invertendo o início e o fim do eixo principal.



flex-direction: column

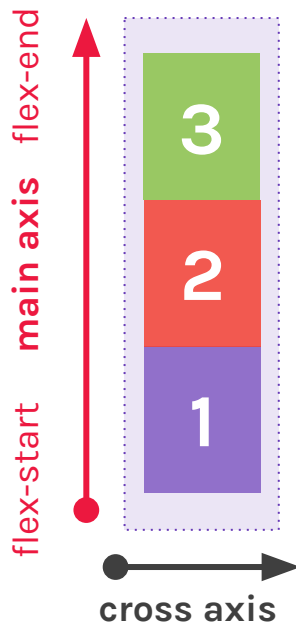
Os itens são organizados no **eixo y**, de **cima** para **baixo**.



flex-direction: column-reverse

Os itens são organizados no **eixo y**, de **baixo** para **cima**.

Neste caso, estamos
invertendo o início
e fim do **main axis**.





Flexbox nos dá **duas propriedades** para **alinhar** facilmente os elementos.

Através do **main axis** com **justify-content**.

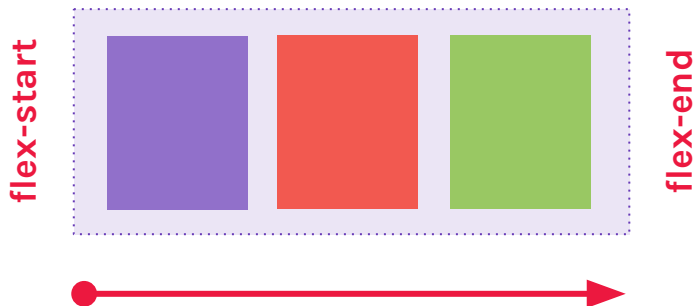
Através do **cross axis** com **align-items**.



3 | justify-content

justify-content

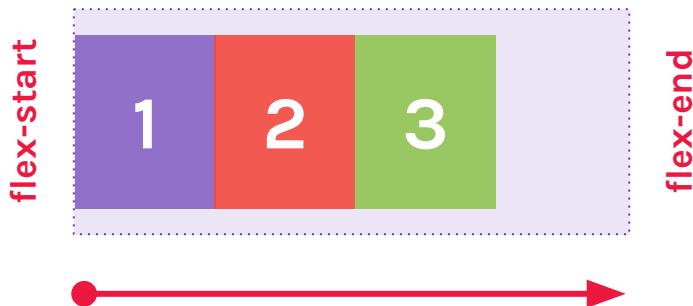
Com esta propriedade, **alinhamos** os **itens** ao longo do **main axis**. Se for horizontal, eles serão alinhados com base na linha. Se for vertical, eles serão alinhados com base na coluna.



justify-content: flex-start

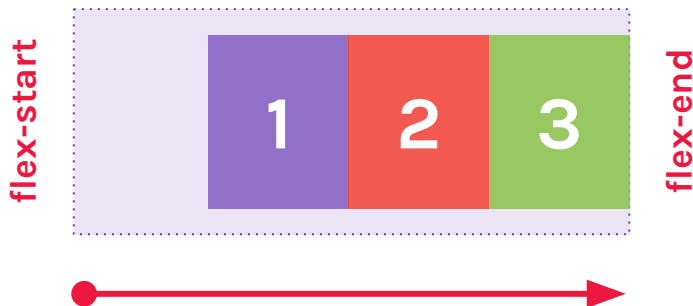
Os itens são alinhados em relação ao **início** do **main axis** que definimos.

Se não especificarmos o justify-content para o contêiner, flex-start será o padrão.



justify-content: flex-end

Os itens são alinhados em relação ao **final** do **main axis** que definimos.



justify-content: center

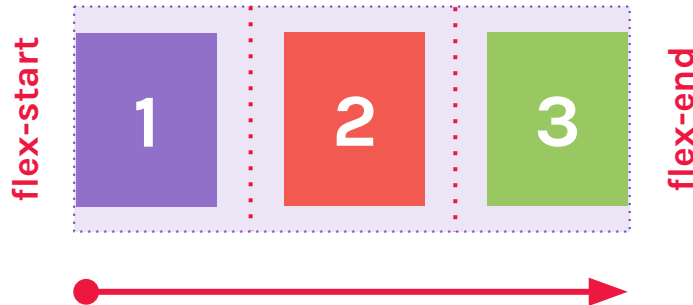
Os itens são alinhados ao **centro** do **main axis**.



justify-content: space-between

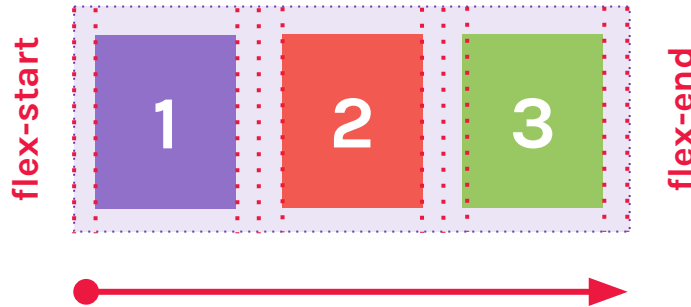
Os itens são **distribuídos uniformemente**. O primeiro item será enviado para o **início** do **main axis** e o último item para o **final**.

O espaço livre será distribuído para separar os itens.



justify-content: space-around

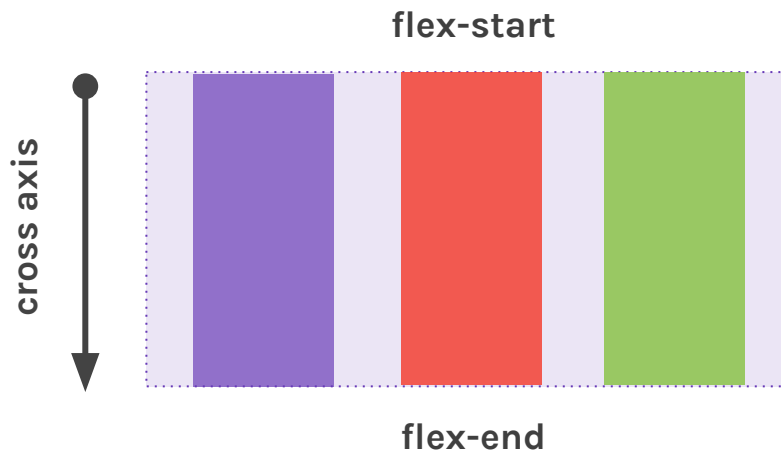
Os itens são **distribuídos uniformemente**. O espaço livre disponível será compartilhado entre todos os elementos. Metade do espaço alocado para cada elemento irá para a direita e a outra para a esquerda (ou para cima e para baixo no caso de colunas).



4 | align-items

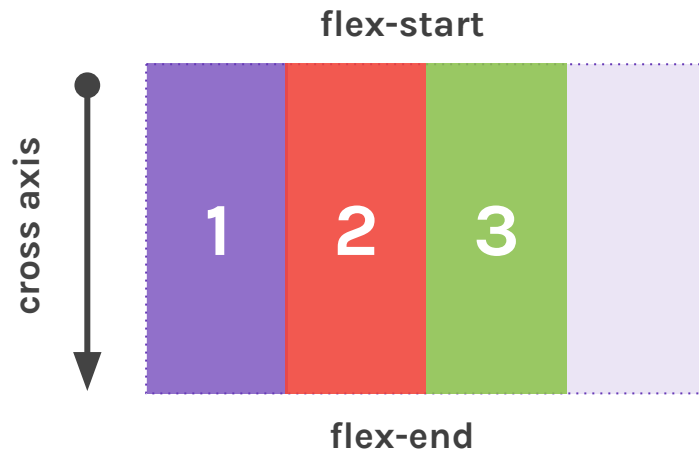
align-items

Com esta propriedade, **alinhamos** os itens ao longo do **cross axis**. Se não declararmos esta propriedade, o valor padrão é **stretch**, ou seja, os itens ocuparão todo o espaço disponível no **cross axis**.



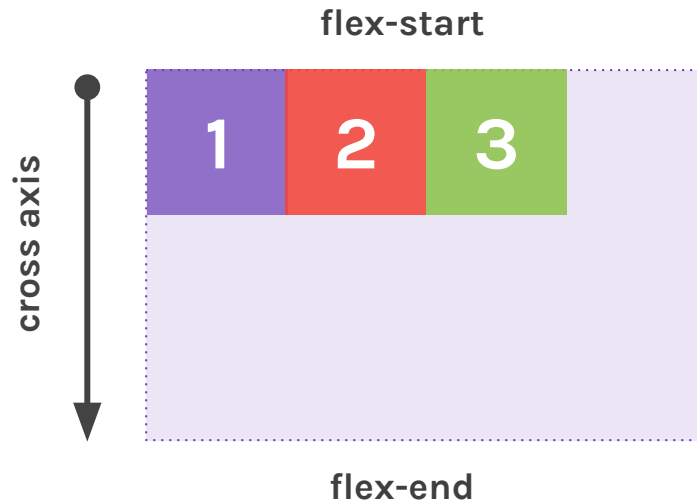
align-items: stretch

Os **itens** se **ajustam** para abranger todo o contêiner. Se o **cross axis** for **vertical**, eles serão ajustados com base na coluna. Se o **cross axis** for **horizontal**, eles serão ajustados com base na linha.



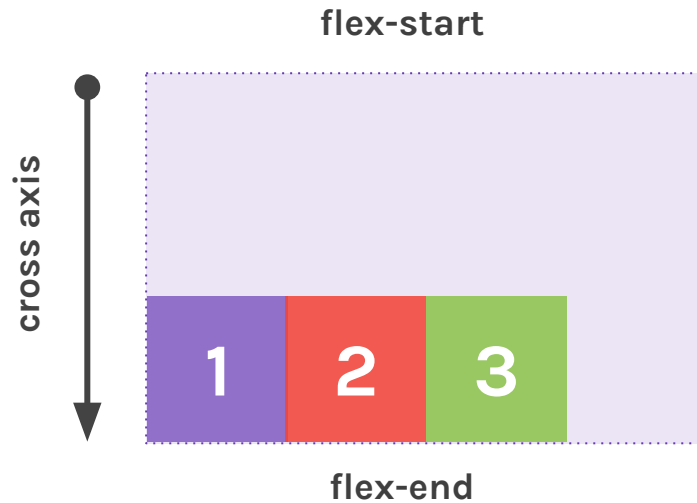
align-items: flex-start

Os itens se alinham no **início** do **cross-axis**.



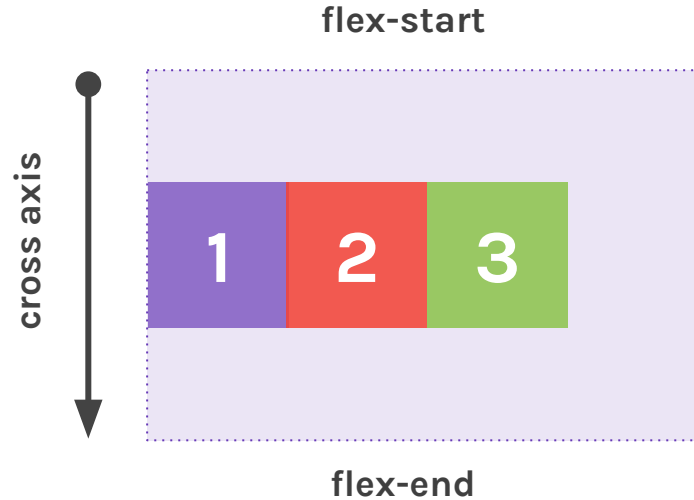
align-items: flex-end

Os itens se alinham no **final** do **cross-axis**.



align-items: center

Os itens se alinham ao **centro** do **cross-axis**.



5 | align-content

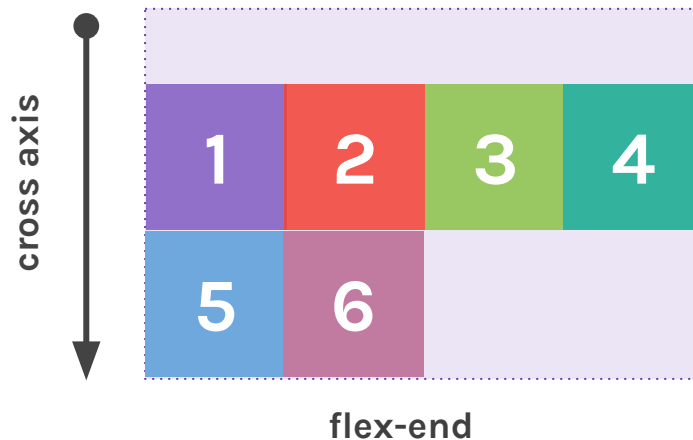
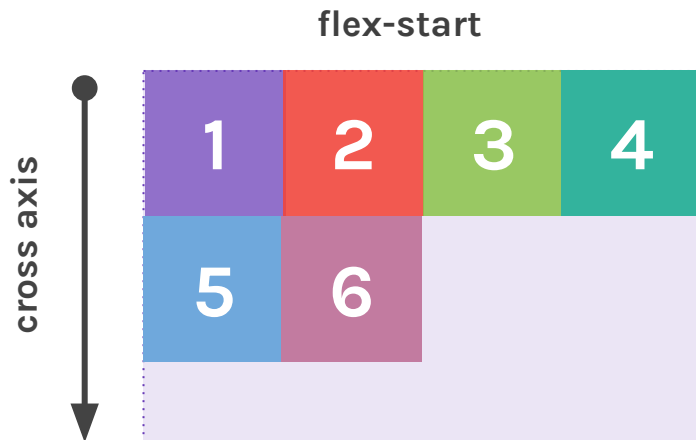
align-content

Se tivermos um contêiner de linha única (onde flex-flow está configurado para no-wrap) usaremos align-items, mas caso estejamos trabalhando com contêineres **multilinhas**, devemos usar **align-content**.

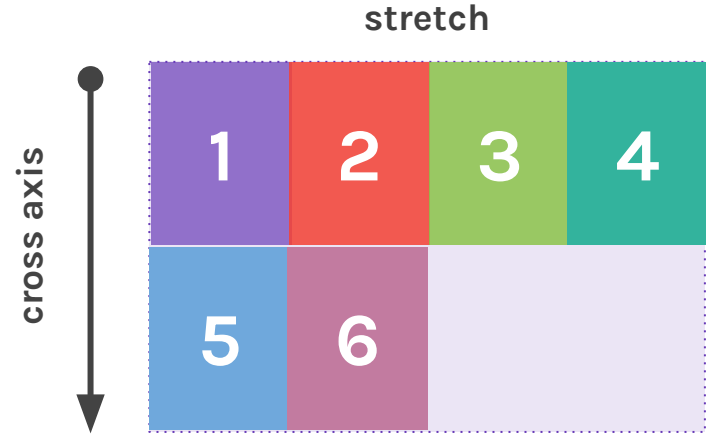
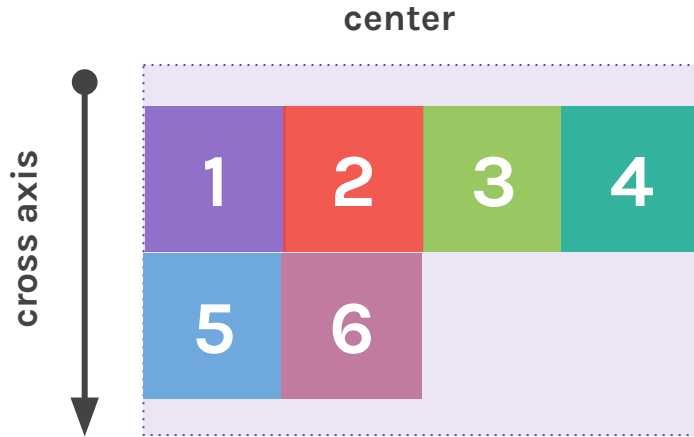
Com esta propriedade, alinhamos os itens ao longo do **cross axis** quando os contêineres flexíveis incluem várias linhas (onde o **flex-flow** é definido como **wrap** ou **wrap-reverse**).

Os valores que a propriedade align-content suporta são semelhantes aos que podemos usar para align-items. Vale a pena testar como ambas as propriedades funcionam para entendê-las bem.

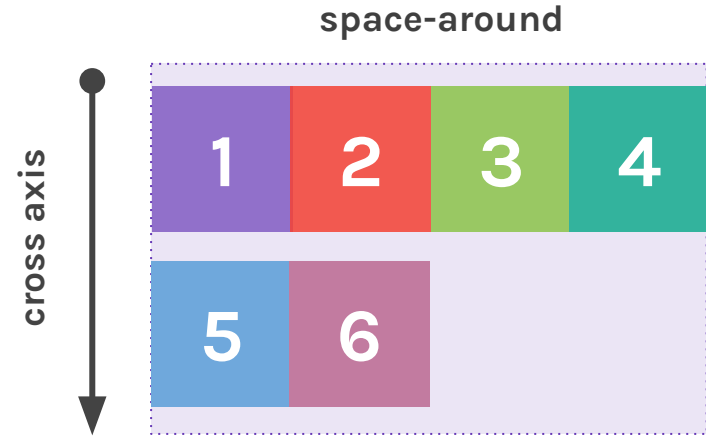
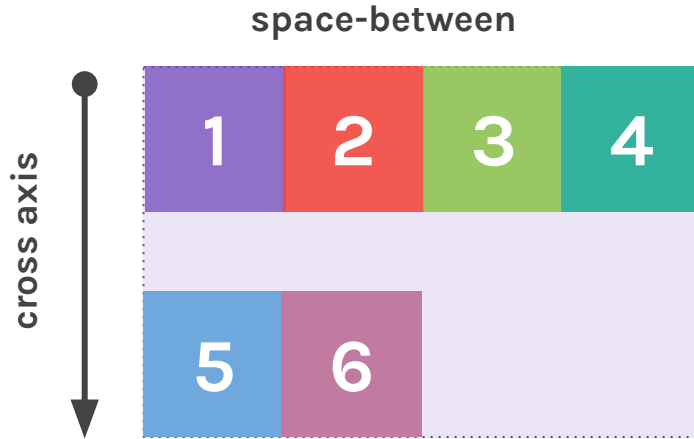
align-content: flex-start | flex-end



align-content: center | stretch



align-content: space-between | space-around





DigitalHouse>