

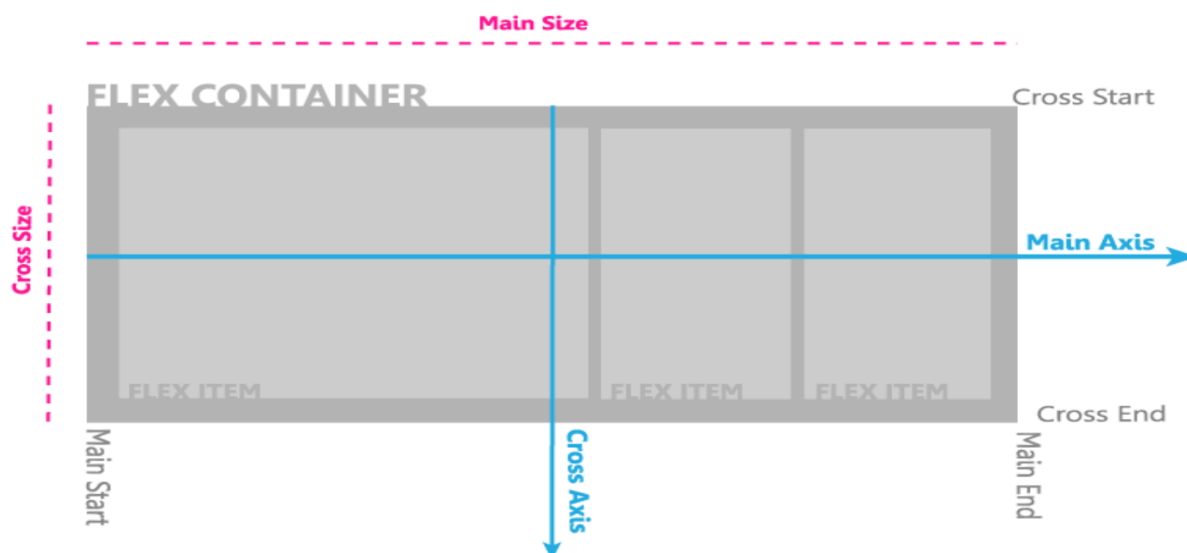
FLEX CONTAINER: elemento do HTML que vai servir como pai de outros itens. Ou seja, uma <div> ou uma <section> por exemplo que possui itens dentro dele.

FLEX ITENS: são os itens presentes dentro dos flex containers.

Algumas propriedades FLEXBOX são atribuídas aos containers e outras aos itens. **Um Flex container é criado ao usar a propriedade `display` de um elemento com o valor `flex`:**

FLEX AXES:

O Flexbox é baseado em `flex-directions`. Essas direções são determinadas por eixos (axes). A imagem abaixo mostra os eixos definidos em um Flex container:



Os flex items são dispostos dentro de um Flex container seguindo o **Main Axis**. O **Main Axis** tem a direção definida pela propriedade **flex-direction**, que pode ser *horizontal* quando flex-direction é **row** ou **row-reverse** e *vertical* quando é **column** ou **column-reverse**.

FLEX LINES:

São linhas “imaginárias” contidas dentro do container. Um Flex container pode ser **single-line** ou **multi-line**, dependendo da propriedade **flex-wrap**:

Single-line: dispõe todos os flex itens em uma única linha.

Multi-line: quebra os flex itens em múltiplas linhas, como acontece com um texto que está muito grande e vai pra linha de baixo (overflow).

CSS:

flex-container {

display: flex; > deixa os flex itens em linha.

flex-direction: column/reversecolumn; > Itens em **cascata**

flex-direction: row/reverserow; > itens **alinhados** (padrão)

flex-wrap: nowrap > os itens **não** quebram a linha

flex-wrap: wrap; > **quebra** de linha ao fim do limite do flex-container

flex-flow: row wrap; > exemplo, é a abreviação do **flex-direction** + **flex-wrap** em uma só linha

justify-content: flex-start; > alinha tudo **horizontalmente** no **inicio** do container. (padrão)

justify-content: flex-end; > alinha tudo **horizontalmente** ao **final** do container.

justify-content: center; > centraliza **horizontalmente** no **meio** do container.

justify-content: space-around; > cria um espaço **horizontalmente** entre os itens distribuindo igualmente os elementos dentro do container.

justify-content: space-between; > mesma coisa do space-around, porém, sem margens no início e final do container.

align-items: center/flex-start/flex-end; > igual ao justify, porém ajusta verticalmente.

align-items: stretch; > ajusta a altura do elemento ao container. (efeito **derretido, tipo**)

align-items: baseline; > alinha de acordo com o conteúdo. (EM casos de caixas irregulares, facilita a organização e a leitura).

align-content: flex-start/flex-end/center; > funciona com duas linhas ou mais de elementos, padronizando como se fosse o espaçamento de uma linha só.

flex-items {

align-self: flex-start/flex-end/center/stretch/baseline; > Funciona da mesma forma que o align items, porém, para um “filho” específico. Esse comando ao invés de pegar o bloco como um todo, pega um item individualmente.

order: 4; > o número chamado, é o número do local pra onde o item selecionado vai. Vamo supor que temos 4 boxes e queremos colocar a box 2 em último lugar. Esse comando serviria pra isso.

flex-grow: 0 (padrão); 1, 2, 3 (exs); > faz o elemento selecionado crescer de forma que ocupa o espaço que sobra dentro do container. Ou ocupa o seu espaço dentro do container +1, caso seja o número usado.

flex-shrink: 1 (padrão); ao contrário do grow, ele diminui o espaço do elemento selecionado em relação aos outros.

flex-basis: auto (padrão); ex: 45px/10%; > é o tamanho dos elementos.

