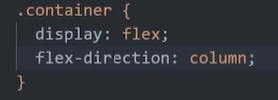
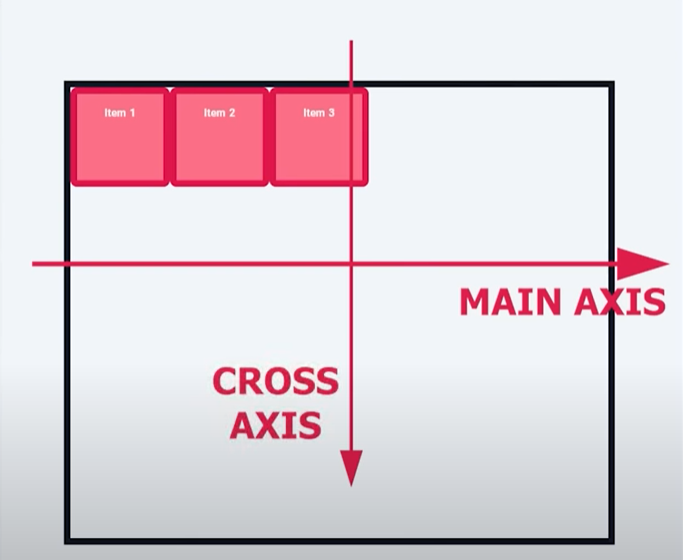
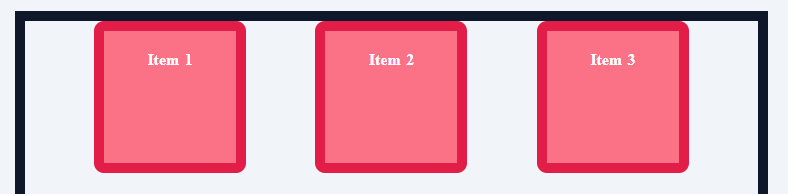
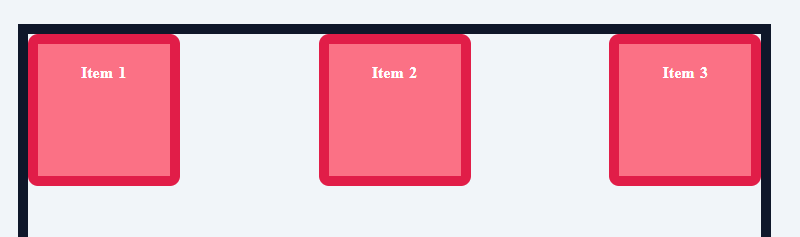
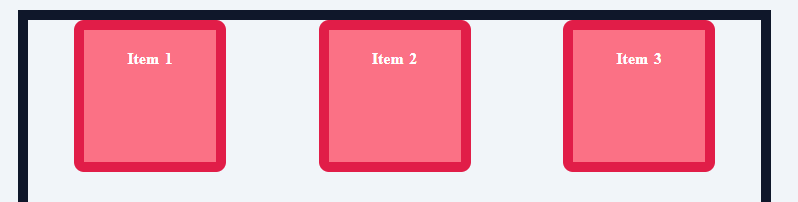
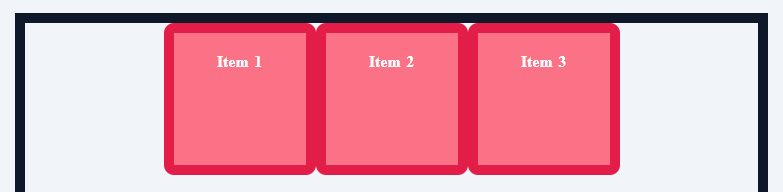
Flexbox :

Pour utiliser les flexbox, il faut d'abord un ***container*** avec des éléments à l'intérieur:



justify-content: flex-start; ( attribute de base )

aligne tout les éléments au début de l'axe principal.

justify-content: flex-end;

aligne tout les éléments à la fin de l'axe principal.

justify-content: center;

Centre tout les éléments sur l'axe principale

justify-content: space-evenly;

Dispose les élément de manièer à avoir un espace égale entre eux et les bords du container

***Positionner les éléments entre eux : (*** [***justify content***](https://www.youtube.com/watch?v=phWxA89Dy94&t=65s) ***)***

justify-content: space-around;

Dispose les éléments de manière à avoir un espace égale entre chaques élément en partant du centre.

justify-content: space-between;

Dispose tout les élément de manière à complété tout l'espace utilisable

On peut ensuite ajouter dans le css au ***container*** l'attribut suivant :

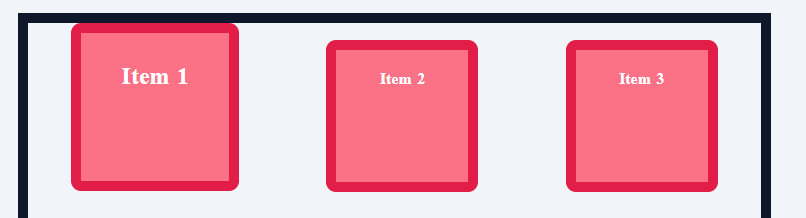
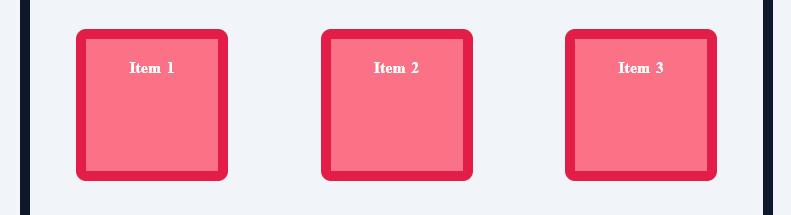
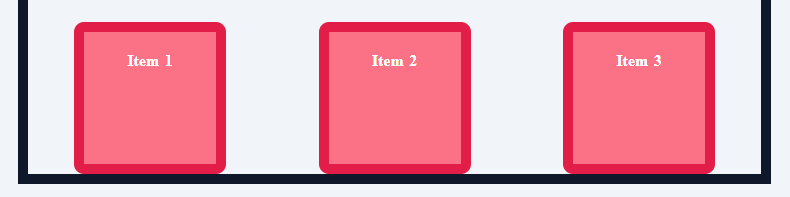
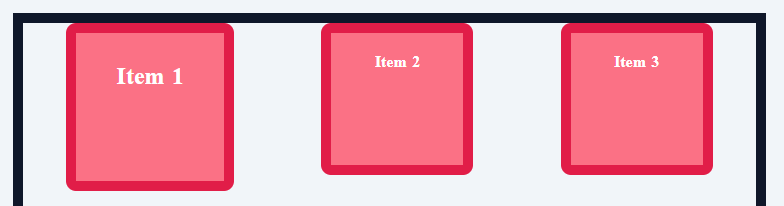
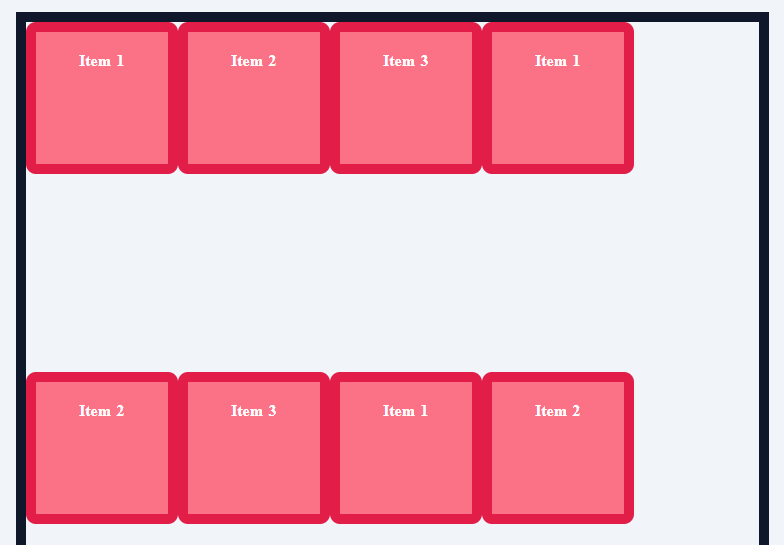
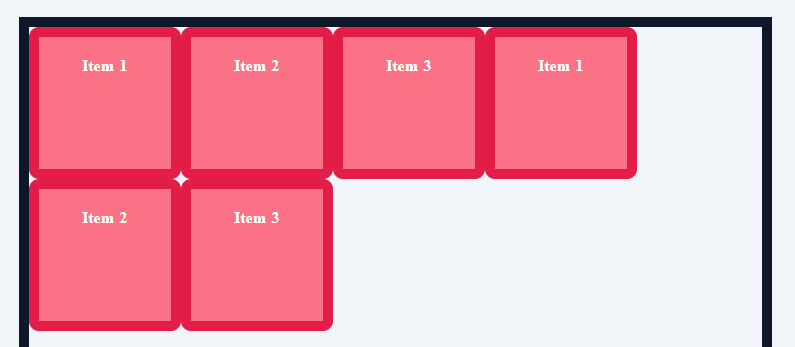
* " [display: flex](https://www.youtube.com/watch?v=phWxA89Dy94&t=11s); "

Cela créer alors deux axes invisibles dans le container, les éléments à l'intérieurs sont alors positionnés sur l'axe principale.

( l'axe principale est horizontal par défaut, cela est modifiable avec l'attribut suivant:

* " [flex-direction: column](https://www.youtube.com/watch?v=phWxA89Dy94&t=39s); "

L'axe principale devient alors l'axe vertical. )



L'attribut " [gap](https://www.youtube.com/watch?v=phWxA89Dy94&t=248s)  " permet d'ajouter un espace entre les éléments du container

align-content ne fonctionne que si l'attribut " flex-wrap: wrap; " est appliqué au container et que nos item wrappent ( que l'attribut wrap soit appliqué ). Cette attribut permet de disposer tout les éléments sur l'axe secondaire.

* Les options de " align-content " sont les mêmes que pour " justify content ".

[***align-content***](https://www.youtube.com/watch?v=phWxA89Dy94&t=216s) ***:***

" flex-wrap: wrap; "

Les éléments ne se limitent plus à une ligne.

" align-item: flex-start; "

Dispose les éléments à la base de l'axe secondaire en ignorant la position de leur contenu.

" flex-wrap: nowrap; " ( attribut de base)

Lorsque l'on a trop d'éléments pour le container, les éléments se rapetissent afin de tenir dans le container pour tenir sur une seule ligne.

[***flex-wrap***](https://www.youtube.com/watch?v=phWxA89Dy94&t=192s) ***:***

" align-item: end; "

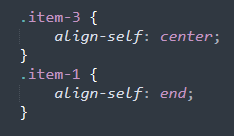
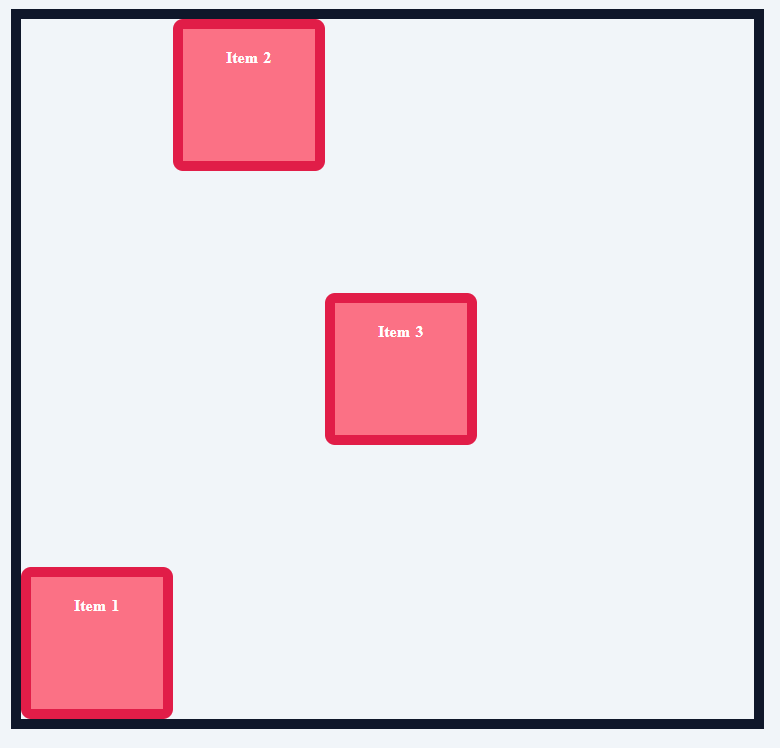
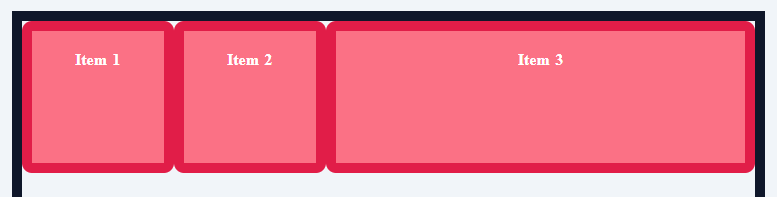
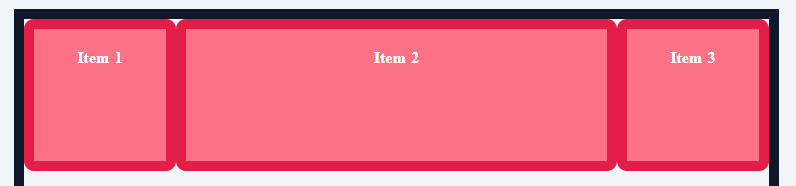
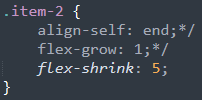
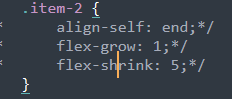
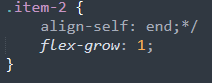
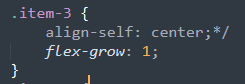
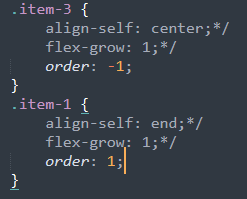
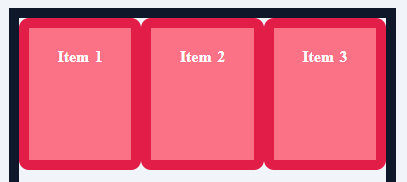
Aligne les éléments de manière à ce que leurs contenu soit alignées et les disposent à la fin de l'axe secondaire.

" [align-item: baseline](https://www.youtube.com/watch?v=phWxA89Dy94&t=101s); " ( attribut de base )

Aligne les éléments de manière à ce que leurs contenu soit alignées et les disposent à la base de l'axe secondaire.

" align-item: center; "

Aligne les éléments de manière à ce que leurs contenu soit alignées et les disposent au centre de l'axe secondaire.



Avec l'attribut " order " en sélectionnant un élément on peut définir sa position par rapport aux autre éléments ( les élément ont tous la position 0 de base ).

Avec l'attribut " flex-basis " en sélectionnant un élément on peut réécrire sa taille.

Avec l'attribut " flex-shrink " en sélectionnant un élément on peut dire à un élément à quel vitesse il doit se réduire quand l'espace disponible l'y oblige.

Avec l'attribut " flex-grow " en sélectionnant un élément on peut dire à un élément de compléter l'espace restant sur l'axe.

Avec l'attribut " [align-self](https://www.youtube.com/watch?v=phWxA89Dy94&t=403s) " en sélectionnant un/ou plusieurs éléments, il est possible de changer uniquement la position de ces éléments.