

FLEXBOX

Les containers flexibles, la direction, les propriétés wrap et flow

Qu'est-ce que FlexBox ?

Flexbox permet de gérer plus facilement la disposition des éléments HTML en permettant facilement de les aligner et de définir l'espace les entourant, et ce même si leur taille est inconnue ou dynamique.

La première idée derrière les **Flexbox** est que les conteneurs (containers) modifient la hauteur et la largeur des éléments qu'ils contiennent pour optimiser l'espace libre. Un conteneur flexible peut ainsi étendre ou diminuer la taille des éléments HTML qu'il contient.

Pour rappel, un conteneur est simplement un élément HTML qui contient d'autres éléments HTML.

Flexbox permet de définir des propriétés sur les conteneurs flexibles et sur les éléments qu'ils contiennent, les éléments **flex**.

Un certain nombre de propriétés CSS peuvent être ajoutés sur les conteneurs pour définir le comportement des éléments à l'intérieur de ceux-ci.

Définir un conteneur flexible

Pour obtenir un conteneur flex, il suffit d'ajouter la propriété `display` avec la valeur `flex` à un conteneur :

```
MesTests > style.css > .container
1  .container{
2      display: flex;
3  }
```

Les éléments enfants du conteneur deviennent alors des éléments flexibles (ou flex items).

Définir l'axe principal du conteneur

Comme nous l'avons vu, Flex permet de disposer les éléments sur un axe horizontal ou vertical.

Pour cela, il faut définir l'axe principal (main axis) du container.

Pour définir l'axe principal du conteneur, il suffit d'ajouter la propriété **flex-direction** :

```
index.html style.css x
MesTests > style.css > .container
1  .container{
2      display: flex;
3      flex-direction: row;
4  }
```

Les valeurs possibles de cette propriété sont **row**, **row-reverse**, **column** et **column-reverse**.

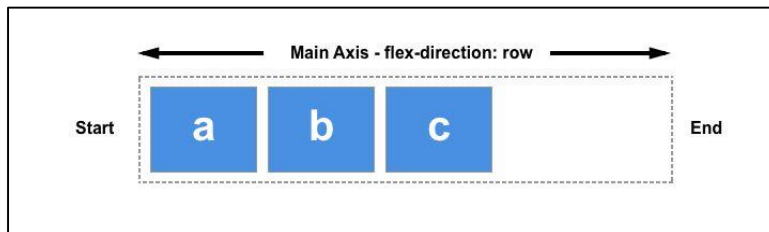
Par défaut, la valeur de la propriété est à **row**.

Les éléments peuvent donc s'aligner :

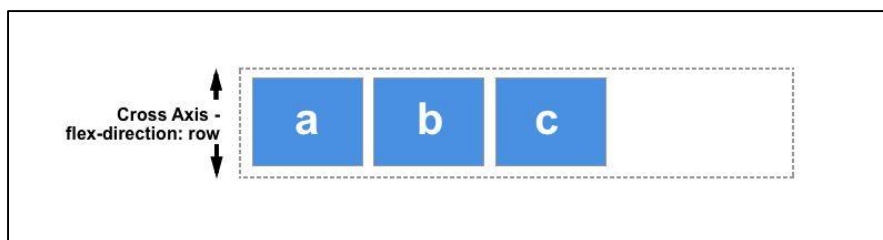
- Horizontalement de gauche vers la droite avec **row**
- Horizontalement de droite vers la gauche **row-reverse**
- Verticalement de haut en bas avec **column**
- Verticalement de bas en haut avec **column-reverse**

L'axe secondaire (cross axis) est l'axe perpendiculaire à l'axe principal.

Par exemple si l'axe principal est row, l'axe secondaire sera column :



Alors nous aurons :



Définir si les éléments peuvent aller ou non à la ligne

Une propriété permet de définir si les éléments peuvent ou non aller à la ligne : **flex-wrap**

Les valeurs possibles de cette propriété sont **nowrap**, **wrap** et **wrap-reverse**.

Par défaut, cette propriété vaut **nowrap**.

```
index.html style.css X
MesTests > style.css > .container
1  .container{
2  flex-wrap: wrap;
3  }
```

Ainsi les éléments contenus peuvent :

- Être sur une ligne maximum obligatoirement avec **nowrap**
- Être sur plusieurs lignes avec plus d'éléments sur la ligne **la plus haute wrap**
- Être sur plusieurs lignes avec plus d'éléments sur la ligne la plus basse **wrap-reverse**

A noter qu'avec nowrap, la taille des éléments sera modifiée pour tenir sur la ligne.

Si par exemple l'axe principal est horizontal, alors leur **width** sera diminuée si l'ensemble des éléments ne tient pas sur la ligne.

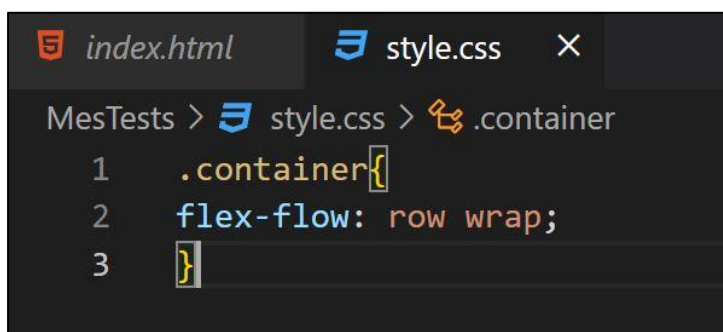
Leur **width** diminuera jusqu'à une largeur minimale qui correspond à la largeur de leur contenu. S'il n'y a toujours pas assez de place sur la ligne, les éléments dépasseront de la ligne et seront hors zone d'affichage.

Le raccourci flex-flow

Une propriété permet de définir **flex-direction** et **flex-wrap** en une seule fois, il s'agit de la propriété **flex-flow**.

La première valeur passée est utilisée pour **flex-direction** et la seconde pour **flex-wrap**.

On peut ainsi écrire par exemple directement :



```
index.html style.css X
MesTests > style.css > .container
1  .container{
2  flex-flow: row wrap;
3  }
```

Des exemples

Retrouvez des exemples pour ces propriétés : [FlexBox](#)