

FLEXBOX

pour faire de la mise en page avec Flexbox, il faut :

1. Définir un conteneur.
2. Et placer à l'intérieur plusieurs éléments.

sur une même page web, vous pouvez avoir plusieurs conteneurs (plusieurs cartons, si vous préférez). Vous pouvez en créer autant que nécessaire pour obtenir la mise en page que vous voulez.

```
<div class="container">
  <div class="element element1">Élément 1</div>
  <div class="element element2">Élément 2</div>
  <div class="element element3">Élément 3</div>
</div>
```

Alignez les éléments d'un conteneur avec `display: flex;`

Il suffit d'une seule propriété CSS, et tout change ! Cette propriété, c'est **flex** et on l'applique au conteneur :

```
.container {
  display: flex;
}
```

Donnez leur une direction avec la propriété `flex-direction`

Flexbox permet d'agencer ces éléments dans le sens que l'on veut. Avec **flex-direction**, on peut les positionner verticalement ou encore les inverser. Cette propriété CSS peut prendre les valeurs suivantes :

- `row` : organisés sur une ligne (par défaut) ;
- `column` : organisés sur une colonne ;
- `row-reverse` : organisés sur une ligne, mais en ordre inversé ;
- `column-reverse` : organisés sur une colonne, mais en ordre inversé.

Ici, on fait simplement :

```
.container {
  display: flex;
  flex-direction: column;
}
```

Retournez à la ligne avec la propriété `flex-wrap`

Par défaut, les blocs essaient de rester sur la même ligne s'ils n'ont pas la place, quitte à "s'écraser", et provoquer parfois des anomalies dans la mise en page (certains éléments pouvant dépasser de leur conteneur). Si vous voulez, vous pouvez demander à ce que les blocs aillent à la ligne lorsqu'ils n'ont plus la place, avec **flex-wrap**.

Voilà les différentes valeurs de `flex-wrap` :

1. `nowrap` : pas de retour à la ligne (par défaut) ;
2. `wrap` : les éléments vont à la ligne lorsqu'il n'y a plus la place ;
3. `wrap-reverse` : les éléments vont à la ligne, lorsqu'il n'y a plus la place, en sens inverse.

```
.container {
  display: flex;
  flex-wrap: nowrap;
  /* OU wrap;
  OU wrap-reverse; */
}
```

Alignez sur l'axe principal avec la propriété `justify-content`

Pour faire simple, partons sur des éléments organisés horizontalement (c'est le cas par défaut).

Pour changer leur alignement, on va utiliser **justify-content**, qui peut prendre ces valeurs :

- **flex-start**: alignés au début (par défaut) ;
- **flex-end**: alignés à la fin ;
- **center**: alignés au centre ;
- **space-between**: les éléments sont étirés sur tout l'axe (il y a de l'espace entre eux) ;
- **space-around**: idem, les éléments sont étirés sur tout l'axe, mais ils laissent aussi de l'espace sur les extrémités.

On écrit alors :

```
.container {
  display: flex;
  justify-content: flex-start
}
```

Alignez sur l'axe secondaire avec align-items

Si nos éléments sont placés dans une direction horizontale (ligne), l'axe secondaire est... vertical. Et inversement : si nos éléments sont dans une direction verticale (colonne), l'axe secondaire est horizontal.

La propriété **align-items** permet de changer leur alignement sur l'axe secondaire, grâce aux valeurs :

- **stretch**: les éléments sont étirés sur tout l'axe (valeur par défaut) ;
- **flex-start**: alignés au début ;
- **flex-end**: alignés à la fin ;
- **center**: alignés au centre ;
- **baseline**: alignés sur la ligne de base (semblable à flex-start).

Pour ces exemples, nous allons partir du principe que nos éléments sont dans une direction horizontale

.container {

```
display: flex;
justify-content: center;
align-items: center;
}
```

<pre><h2>Start</h2> <div class="container start"> <div class="element element1"></div> <div class="element element2"></div> <div class="element element3"></div> </div> <h2>End</h2> <div class="container end"> <div class="element element1"></div> <div class="element element2"></div> <div class="element element3"></div> </div> <h2>Center</h2> <div class="container center"> <div class="element element1"></div> <div class="element element2"></div> <div class="element element3"></div> </div> <h2>Space Between</h2> <div class="container spaceBetween"> <div class="element element1"></div> <div class="element element2"></div> <div class="element element3"></div> </div> <h2>Space Around</h2> <div class="container spaceAround"> <div class="element element1"></div> <div class="element element2"></div> <div class="element element3"></div> </div> <h2>Stretch</h2> <div class="container stretch"> <div class="element element1"></div> <div class="element element2"></div> <div class="element element3"></div> </div></pre>	<pre>.container { display: flex; width: 650px; flex-wrap: wrap; height: 400px; border: solid 2px #000 } .start { justify-content: flex-start } .end { justify-content: flex-end; } .center { justify-content: center; } .spaceBetween { justify-content: space-between; } .spaceAround { justify-content: space-around; } .stretch { justify-content: stretch; } .element { width: 150px; min-height: 100px; } .element1 { background-color : orange; } .element2 { background-color : blue; } .element3 { background-color : lightgreen; }</pre>
---	---