

HTML CSS ressource numéro 13

Sources :

<https://openclassrooms.com/>

L'ancien site 2minuteslearning.fr qui à malheureusement disparu

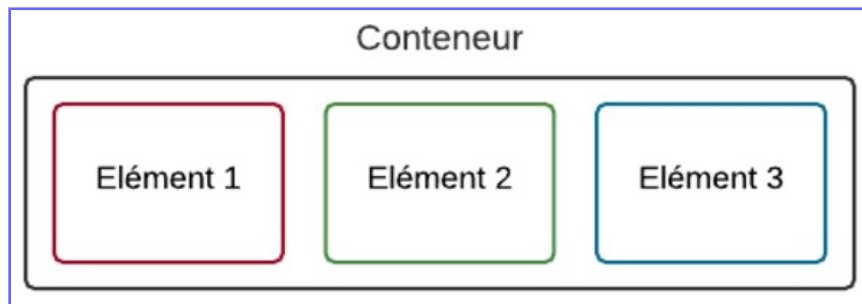
<https://www.vectorskin.com/balises-html5/>

La mise en page avec Flexbox

Flexbox est une nouveauté de CSS très puissante...

Le principe de la mise en page avec **Flexbox** est simple : vous définissez un conteneur, et à l'intérieur vous placez plusieurs éléments.

Commençons par étudier le fonctionnement d'un conteneur.



Le conteneur est une balise HTML, et les éléments sont d'autres balises HTML à l'intérieur

```
<div id="conteneur">
  <div class="element">Elément 1</div>
  <div class="element">Elément 2</div>
  <div class="element">Elément 3</div>
</div>
```

A la une

Element 1
Element 2
Element 3

Sed ut tum ad senem senex de
senectute, sic hoc libro ad amicum
amicissimus scripsi de amicitia. Tum
est Cato locutus, quo erat nemo fere
senior temporibus illis, nemo

On constate que les éléments sont placés les uns en dessous des autres...

Si on mets une propriété CSS, tout change. Cette propriété, c'est **flex**, et elle s'applique au conteneur

```
#conteneur
{
  display: flex;
}
```

A la une

Element 1Element 2Element 3

Sed ut tum ad senem senex de
senectute, sic hoc libro ad amicum
amicissimus scripsi de amicitia. Tum
est Cato locutus, quo erat nemo fere
senior temporibus illis, nemo

On constate maintenant que les éléments sont alignés...

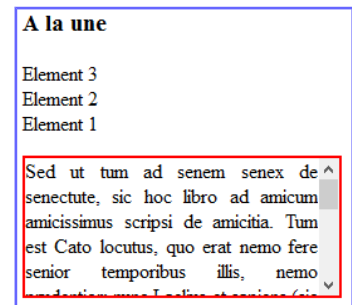
La direction

Flexbox nous permet d'agencer ces éléments dans le sens que l'on veut. Avec **flex-direction**, on peut les positionner verticalement ou encore les inverser. Il 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

Par exemple,

```
#conteneur
{
  display: flex;
  flex-direction: column-reverse;
}
```



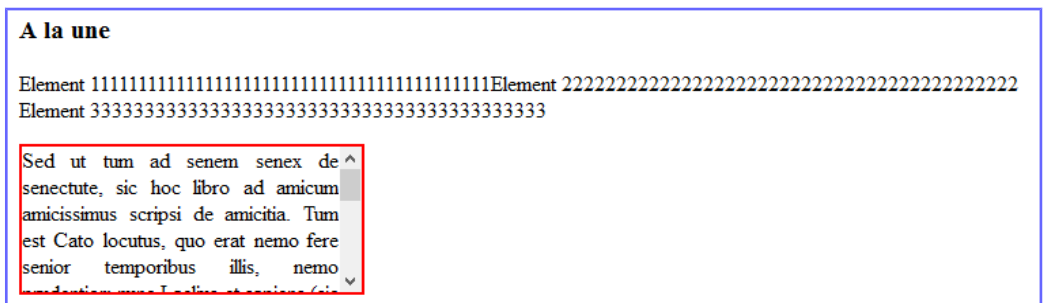
Le retour à la ligne

Par défaut, les blocs essaient de rester sur la même ligne s'ils n'ont pas la place (ce qui peut provoquer des bugs de design parfois). 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** qui peut prendre ces valeurs :

- nowrap** : pas de retour à la ligne (par défaut)
- wrap** : les éléments vont à la ligne lorsqu'il n'y a plus la place
- wrap-reverse** : les éléments vont à la ligne lorsqu'il n'y a plus la place en sens inverse

Pour notre exemple, j'ai rajouté des 1, des 2 et des 3 pour rallonger l'élément;)

```
#conteneur
{
  display: flex;
  flex-wrap: wrap;
}
```



Aligner sur l'axe principal

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 :

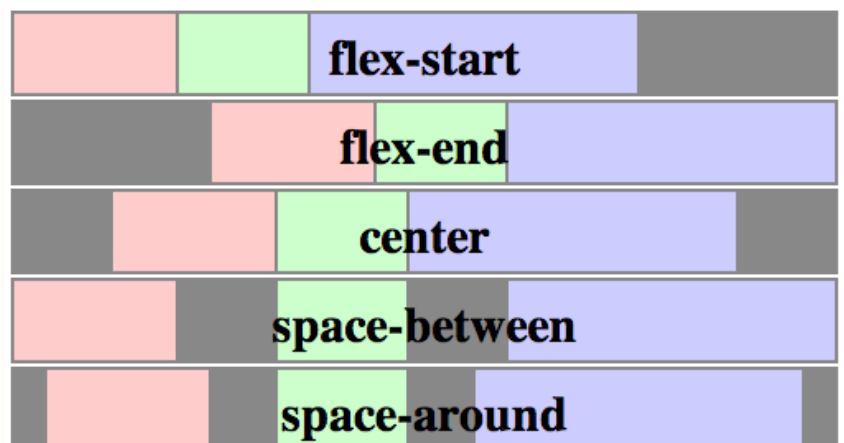
flex-start : alignés au début (par défaut)

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)

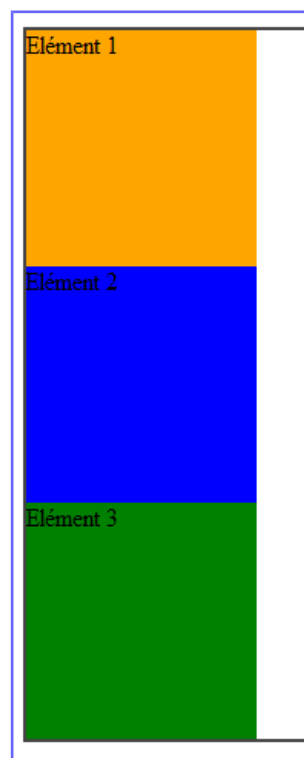
space-around : idem, les éléments sont étirés sur tout l'axe, mais ils laissent aussi de l'espace sur les extrémités



On peut aussi travailler avec des blocs de couleur...

CSS

```
#conteneur
{
  border: 2px solid #444;
}
.element
{
  width: 150px;
  height: 150px;
}
.element:nth-child(1)
{
  background-color: orange;
}
.element:nth-child(2)
{
  background-color: blue;
}
.element:nth-child(3)
{
  background-color: green;
}
```



En résumé

Il existe plusieurs techniques pour positionner les blocs sur la page. Flexbox est la technique la plus récente et de loin la plus puissante.

Le principe de Flexbox est d'avoir un conteneur, avec plusieurs éléments à l'intérieur. Avec `display: flex;` sur le conteneur, les éléments à l'intérieur sont agencés en mode Flexbox (horizontalement par défaut)

Flexbox peut gérer toutes les directions. Avec `flex-direction`, on peut indiquer si les éléments sont agencés horizontalement (par défaut) ou verticalement. Cela définit ce qu'on appelle l'axe principal.

L'alignement des éléments se fait sur l'axe principal avec `justify-content`, et sur l'axe secondaire avec `align-items`.

Avec `flex-wrap`, on peut autoriser les éléments à revenir à la ligne s'ils n'ont plus d'espace.

S'il y a plusieurs lignes, on peut indiquer comment les lignes doivent se répartir entre elles avec `align-content`.

Chaque élément peut être réagencé en CSS avec `order` (pas besoin de toucher au code HTML !).

Avec la super-proprété `flex`, on peut autoriser nos éléments à occuper plus ou moins d'espace restant.

Pas simple ?

Ressources complémentaire

<https://la-cascade.io/flexbox-guide-complet/>