**Cours Web HTML/CSS(en cours)**

LIENS UTILES COURS

Les grids 🡸 <https://gridbyexample.com/examples/>

https://css-tricks.com/snippets/css/a-guide-to-flexbox/

HTML 🡸 https://developer.mozilla.org/fr/docs/Web/HTML

Le CSS 🡸https://css-tricks.com/centering-css-complete-guide/

Le CSS 🡸Feuilles de style en cascade

XML 🡸https://developer.mozilla.org/fr/docs/Web/XML/XML\_introductionXML🡸https://www.w3schools.com/xml/xml\_syntax.asp

**Les flexbox**

**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.

Exemple :

#conteneur

{

*display*: *flex*;

*flex-direction*: column;

}

**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.

Exemple :

#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.

Par exemple :

#conteneur

{

*display*: *flex*;

*justify-content*: space-around;