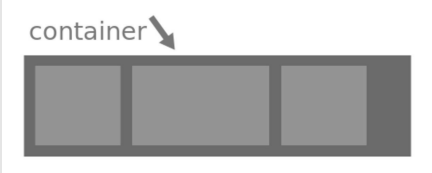
Sources

Cette anti-sèche reprend principalement le tutoriel
A Complete Guide to Flexbox (en)
et accessoirement le tutoriel :
alsacreations : CSS3 Flexbox Layout module : flexibilité (fr)
Notez le code couleurs :
Violet : pour les attributs du container parent (considérer comme synonimes parent et container)
Orange : pour les atributs des articles enfants (considérer comme synonimes items, articles, enfants)
Lorsqu'un attribut peut prendre plusieurs valeurs, la valeur par défaut est indiquée en ***gras.***

Propriétés du parent (couleur violet #88499C)



display: flex; / or inline-flex /

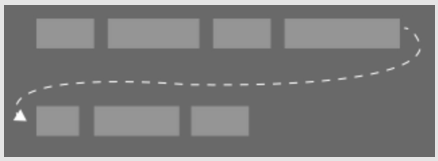
```
.container {
  display: flex; /* or inline-flex */
}
```

déclaration obligatoire qui crée le parent flexbox

direction principale

```
.container {
flex-direction: row | row-reverse | column | column-reverse;
}
```

flex-wrap



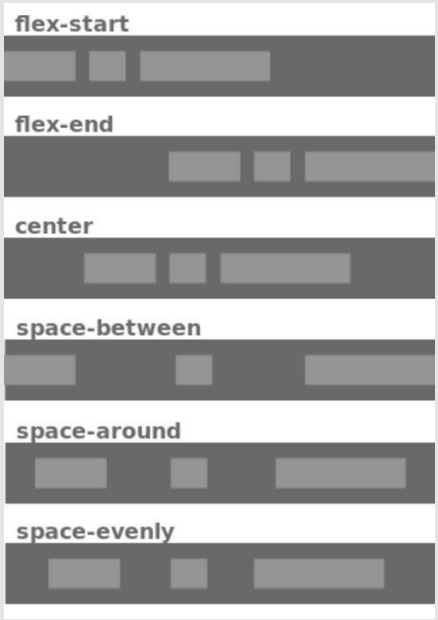
```
.container{
flex-wrap: nowrap | wrap | wrap-reverse;
}
```

wrap ou wrap-reverse va définir le sens de l'**axe secondaire** (cross axes)

flex-flow = flex-direction + flex-wrap

```
.container{
flex-flow: <'flex-direction'> || <'flex-wrap'>`
 }
```

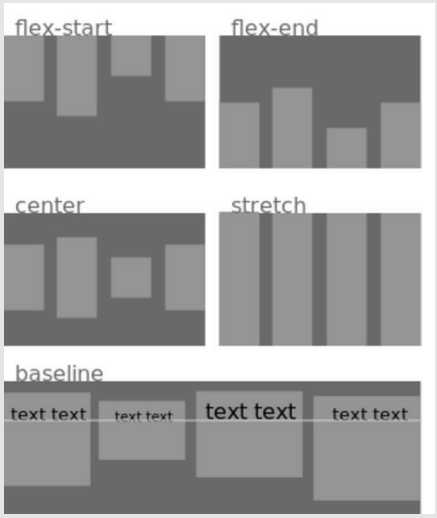
justify-content (axe principal)




```
.container {
justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly;
}`
```

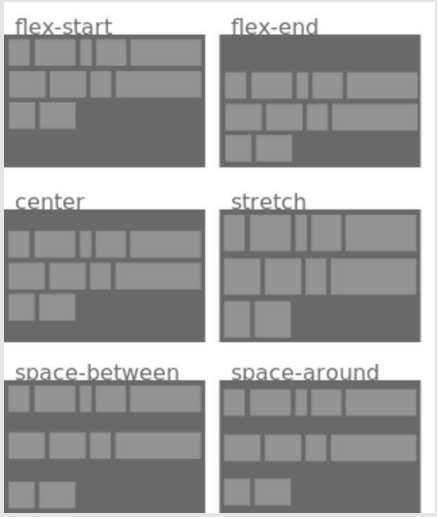
choisir une possibilité parmi les six...

align-items : alignement relatif à une ligne



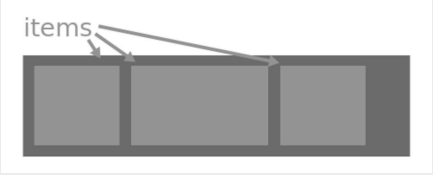

```
.container {
align-items: flex-start | flex-end | center | baseline
| stretch;
}
```

align-content: alignement entre plusieurs lignes




```
.container {
align-content: flex-start | flex-end | center |
space-between | space-around | stretch ;
}
```

Propriétés des enfants (couleur orange #E77F24)

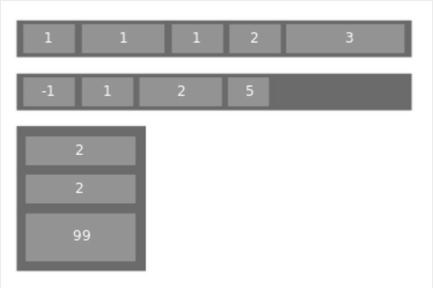


item : align-self (individuels / items)


```
.item {
align-self: auto | flex-start | flex-end | center |
baseline | stretch;
}
```

Permet de surcharger pour des articles les règles générales définies au niveau container-parent.

order: modifier l'ordre.




```
.item {
order: <integer>; / default is 0 /
}
```

Tous les items ayant le même niveau order sont affichés dans l'ordre d'aparition.

flex: raccourci grow + shrink + basis

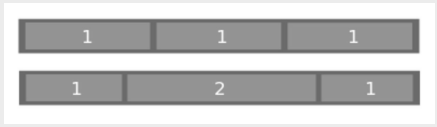
La propriété flex est un raccourci de trois propriétés, qui s'appliquent aux "flex-items" et dont les fonctionnalités sont:
flex-grow : capacité pour un élément à s'étirer dans l'espace restant (0 ou n)
flex-shrink : capacité pour un élément à se contracter si nécessaire (0 ou 1)
flex-basis : taille initiale de l'élément avant que l'espace restant ne soit distribué (<length> ou "auto")

```
.item {
flex: none
flex:<'flex-grow'> <'flex-shrink'> <'flex-basis'>
flex:<'flex-grow'>
}
```

Par défaut, les valeurs de ces propriétés sont :
flex-grow: 0, flex-shrink: 1 et flex-basis: auto
En clair, les flex-items n'occupent initialement que la taille minimale de leur contenu.
Pour rendre **un** élément flexible, il suffit de lui attribuer une valeur de flex-grow (ou flex en raccourci) supérieure à zéro.
Cet élément absorbera alors l'espace restant au sein de son conteneur.
Plusieurs éléments peuvent être rendus flexibles et se répartir l'espace restant. L'espace disponible est alors tout simplement distribué entre les éléments flexibles (proportionnellement à leur coefficient flex-grow)
www.alsacreations.com/tuto/lire/1493-CSS3-Flexbox-Layout-module.html

il est recommandé d'utiliser ce raccourci plutôt que les valeurs séparément.

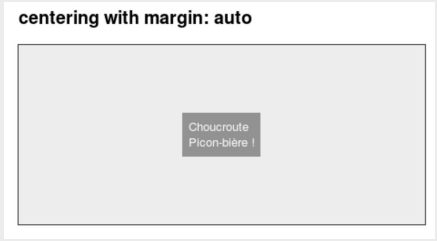
flex-grow: répartition de l'espace




```
item {
flex-grow: <number>; / default 0 /
}
```

La place est répartie proportionnellement aux coefficients attribués.
Pour un effet parfait, il faut avoir assez de place vide avant redistribution.

Utiliser la propriété margin:auto pour centrer



Il est possible de positionner un élément en bas de son conteneur à l'aide d'un :
margin-top: auto,
ou mieux : centrer à la fois horizontalement et verticalement via un simple :
margin: auto.

```
/* paragraphe centré horizontalement et verticalement
*/
.container {
display: flex;
}
.container > p {
margin: auto;
}
```