

# Formation Développement Web

LOUIS ALLARD

[CONTACT@LOUISALLARD.FR](mailto:CONTACT@LOUISALLARD.FR)

[WWW.LOUISALLARD.FR](http://WWW.LOUISALLARD.FR)

# Flexbox

## Qu'est-ce que c'est ?

Les Flexbox sont un ensemble de propriétés CSS qui facilitent le positionnement et l'agencement des éléments sur une page web en créant un conteneur flexible pour ses éléments enfants.

## Comment ?

En déclarant un conteneur flexible avec `display: flex;`, les éléments internes ("flex items") peuvent être ajustés automatiquement en fonction de la taille de l'écran et du contenu, avec des options pour les aligner le long de l'axe principal et secondaire, les envelopper sur plusieurs lignes, et contrôler leur expansion ou réduction selon l'espace disponible.

# Flexbox

## Quel principe ?

A l'instar des grid, les flexbox se basent sur un conteneur principal qui contiendra différents éléments, éléments qui peuvent eux même être d'autres conteneur.

# Flexbox

## Exemple :

```
<div class="container">  
  <div class="a">Un</div>  
  <div class="b">Deux</div>  
  <div class="c">Trois</div>  
</div>
```

## CSS :

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

# Flexbox

## Gérer la direction des blocs avec flex-direction :

Cette propriété css permet de placer les différents blocs du conteneur, en ligne, en colonne, dans un sens ou dans l'autre. Elle se place dans la class du conteneur.

- row : en ligne
- column : en colonne
- row-reverse : en ligne mais en inversant l'ordre
- column-reverse : en colonne mais en inversant l'ordre

# Flexbox

## Flex-wrap :

La propriété flex-wrap permet aux éléments de passer à ligne suivante si leur conteneur ne permet pas de les contenir. Sans elle, les blocs “forceront” par tout les moyens. Cette propriété se place dans la class du conteneur.

- nowrap : laisse la valeur par défaut, donc sans retour à la ligne
- wrap : active le passage à la ligne
- wrap-reverse : idem que wrap mais en sens inverse

# Flexbox

## Changer l'organisation :

Par défaut, les blocs seront organisés horizontalement, cependant ils peuvent l'être de manière verticale.

On parle d'axe principale et secondaire. Par défaut, l'axe principale est donc l'horizontal et le secondaire le vertical.

# Flexbox

## Changer l'organisation : justify-content :

Cette propriété, à plac  dans la class du conteneur, va permettre d'aligner sur l'axe principale.

- flex-start : valeur par d faut, align  au d but
- flex-end : align s   la fin
- center : align s au centre
- space-between : on  tire les  l ments sur tout l'axe
- space-around : on  tire les  l ments sur tout l'axe mais en laissant des espaces aux extr mit s.



# Flexbox

## Changer l'organisation : align-item :

Le principe est le même que pour justify-content, mais pour l'axe secondaire.

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

# Flexbox

## Centrer :

En css, le centrage peut-être délicat, avec flexbox tout deviens plus simple. Il suffit de mettre des marges automatiques sur les class des éléments.

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

```
.bloc {  
  margin: auto;  
}
```