

Le module CSS3 *Flexible box Layout*

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

Le module CSS3 *Flexbox Layout* fournit une façon plus efficace de disposer, aligner et distribuer l'espace entre les items d'un *container*, même lorsque leurs dimensions sont inconnues et/ou dynamiques - d'où le terme "flex".

L'idée principale est de donner à un élément contenant (container) la possibilité de changer les largeur et hauteur des éléments contenus (items), afin de remplir au mieux l'espace disponible, et s'adapter à tous les *devices* et toutes les tailles d'écrans.

Un container flexible permet aux items de s'étendre pour occuper la place disponible ou au contraire les réduit pour leur éviter de déborder.

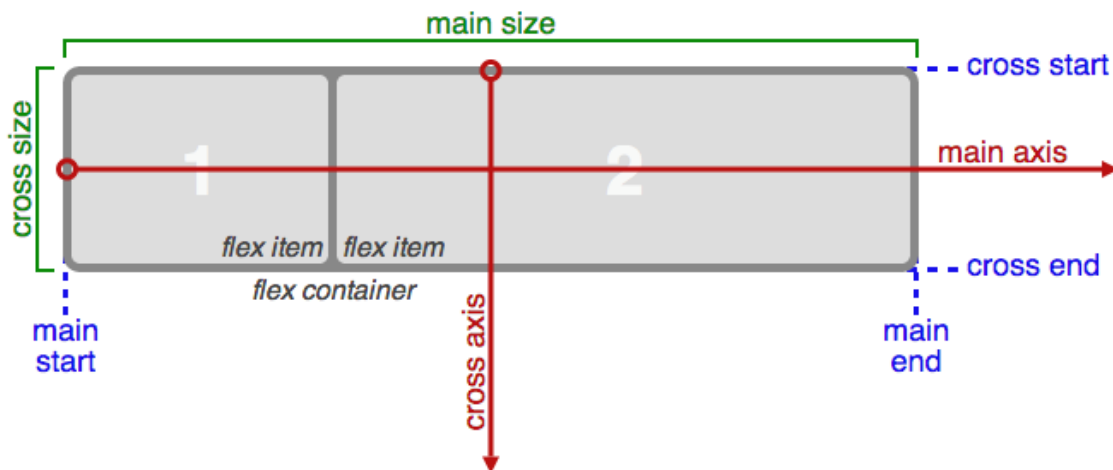
Le plus important à retenir c'est qu'avec Flexbox la disposition n'est pas rigidement directionnelle, contrairement à ce que nous connaissons : *Block* est basé sur un schéma vertical, *Inline* sur un schéma horizontal.

Les bases

Flexbox étant un module, il a des propriétés bien à lui. Certaines concernent le container (l'élément parent, qu'on appelle "flex container"), d'autres concernent le ou les enfants (les "flex items").



Le positionnement habituel est basé sur les directions de flux [block et inline](#), le positionnement flex, lui, est basé sur les directions "flex-flow". L'illustration ci-dessous, tirée des spécifications, explique l'idée qui est à la base du positionnement flex :



Propriétés qui s'appliquent à l'élément parent, le container.

display: flex | inline-flex;

C'est ainsi qu'on définit un container flex, il est block par défaut ou inline selon la valeur donnée. Cela crée un contexte flex pour tous les descendants directs.

flex-direction: row | row-reverse | column | column-reverse

La propriété flex-direction établit l'axe principal.

flex-wrap: nowrap | wrap | wrap-reverse

Cette propriété définit si le container comprend une seule ligne ou plusieurs et la direction sur l'axe perpendiculaire (cross-axis), qui détermine la direction dans laquelle les nouvelles lignes seront emplies.

flex-flow: <'flex-direction'> || <'flex-wrap'>

Cette propriété est un raccourci des propriétés "flex-direction" et "flex-wrap" qui ensemble définissent les axes "main" et "cross" du container flex. La valeur par défaut est row nowrap.

justify-content: flex-start | flex-end | center | space-between | space-around

La propriété justify-content définit l'alignement le long de l'axe principal. Elle permet de distribuer l'espace excédentaire lorsque tous les items flex sur une ligne sont inflexibles ou lorsqu'ils ont atteint leur taille maximale. Elle contrôle aussi l'alignement des items lorsqu'ils débordent.

align-items: flex-start | flex-end | center | baseline | stretch

La propriété align-items définit la façon dont les items d'une ligne sont disposés le long de l'axe "cross". On peut le voir comme la version de justify-content pour "cross axis".

align-content: flex-start | flex-end | center | space-between | space-around | stretch

La propriété align-content aligne les lignes d'un container flex à l'intérieur de l'espace où il reste de l'espace sur l'axe cross, un peu comme justify-content aligne les items sur l'axe principal.

Note : cette propriété n'a pas d'effet quand la flexbox n'a qu'une seule ligne.

Propriétés qui s'appliquent aux éléments enfants, les items flex.

order: <nombre entier>

Par défaut, les items flex sont disposés par ordre d'arrivée. Cependant, la propriété **order** permet de contrôler l'ordre dans lequel ils apparaissent dans le container.

flex-grow: <nombre entier> (par défaut = 0)

La propriété flex-grow définit la possibilité pour un item de grandir, si nécessaire. Elle accepte une valeur sans unité qui sert de proportion. Elle dicte l'espace que peut prendre l'item à l'intérieur de l'espace disponible dans le flex container. Si tous les items ont flex-grow défini à 1, chaque enfant aura le même espace dans le container. Si vous donnez à l'un des enfants une valeur de 2, cet enfant prendra deux fois plus de place que les autres.

flex-shrink: <nombre entier> (par défaut = 1)

La propriété flex-shrink définit la possibilité pour un item flex de rétrécir si nécessaire.

flex-basis: <longueur> | auto (par défaut = auto)

La propriété flex-basis définit la taille par défaut d'un élément avant que l'espace restant soit réparti.

flex: none | [<'flex-grow'> <'flex-shrink'>? <'flex-basis'>?]

Cette propriété est le raccourci de flex-grow , flex-shrink et flex-basis . Les deuxième et troisième paramètres sont optionnels. Les valeurs par défaut sont 0 1 auto.

align-self: auto | flex-start | flex-end | center | baseline | stretch

La propriété align-self permet à des items flex de passer outre aux alignement par défaut ou à ceux spécifiés par align-items. Les valeurs sont les mêmes que pour ce dernier.

Voir les descriptions complètes des propriétés et des valeurs sur :

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

Tester en ligne l'effet des propriétés et des valeurs :

<http://codepen.io/evefevrier/pen/LNBoRV>

<http://bennettfeely.com/flexplorer/>

<http://codepen.io/justd/pen/yydezN>

Exercice préliminaire : <http://flexboxfroggy.com/#fr>

Références en anglais

A Complete Guide to Flexbox

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

The Ultimate Flexbox Cheat Sheet

<http://www.sketchingwithcss.com/samplechapter/cheatsheet.html>