

Le module CSS3

Flexible box Layout

Le module CSS3 *Flexbox Layout*

fournit une façon plus efficace de disposer, aligner et distribuer l'espace entre les items d'un conteneur, même lorsque leurs dimensions sont inconnues et/ou dynamiques d'où le terme « flex ».

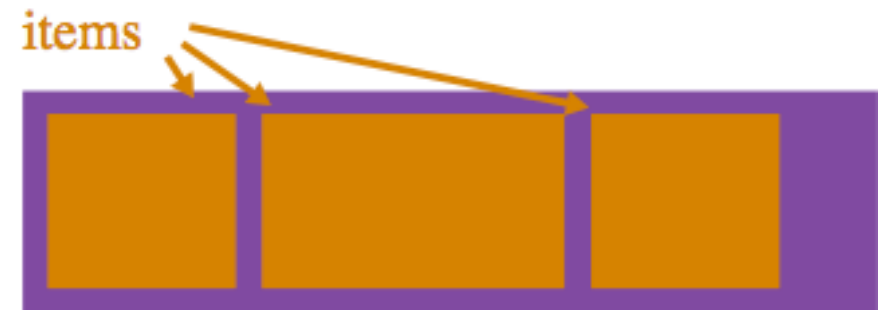
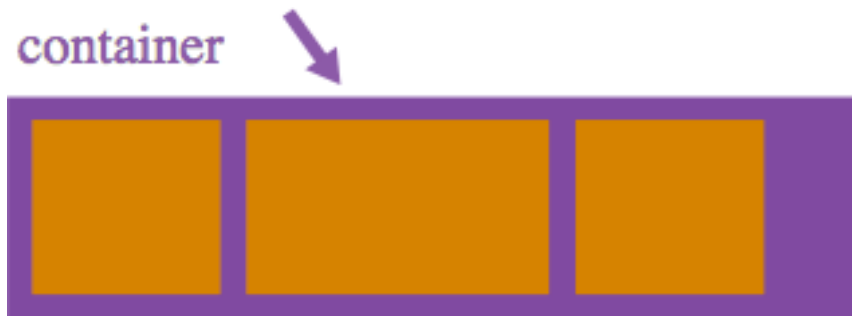
L'idée principale est de donner à un **élément contenant** (conteneur flex), la possibilité de changer les largeurs et hauteurs des **éléments contenus** (items flex), afin de remplir au mieux l'espace disponible, et s'adapter à tous les types d'appareils et toutes les tailles d'écrans.

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

Les bases

Flexbox étant un module des CSS3, il a des propriétés bien à lui.

Certaines concernent **l'élément parent**, le **conteneur flex** ;
d'autres concernent **le ou les enfants**, les **items flex**



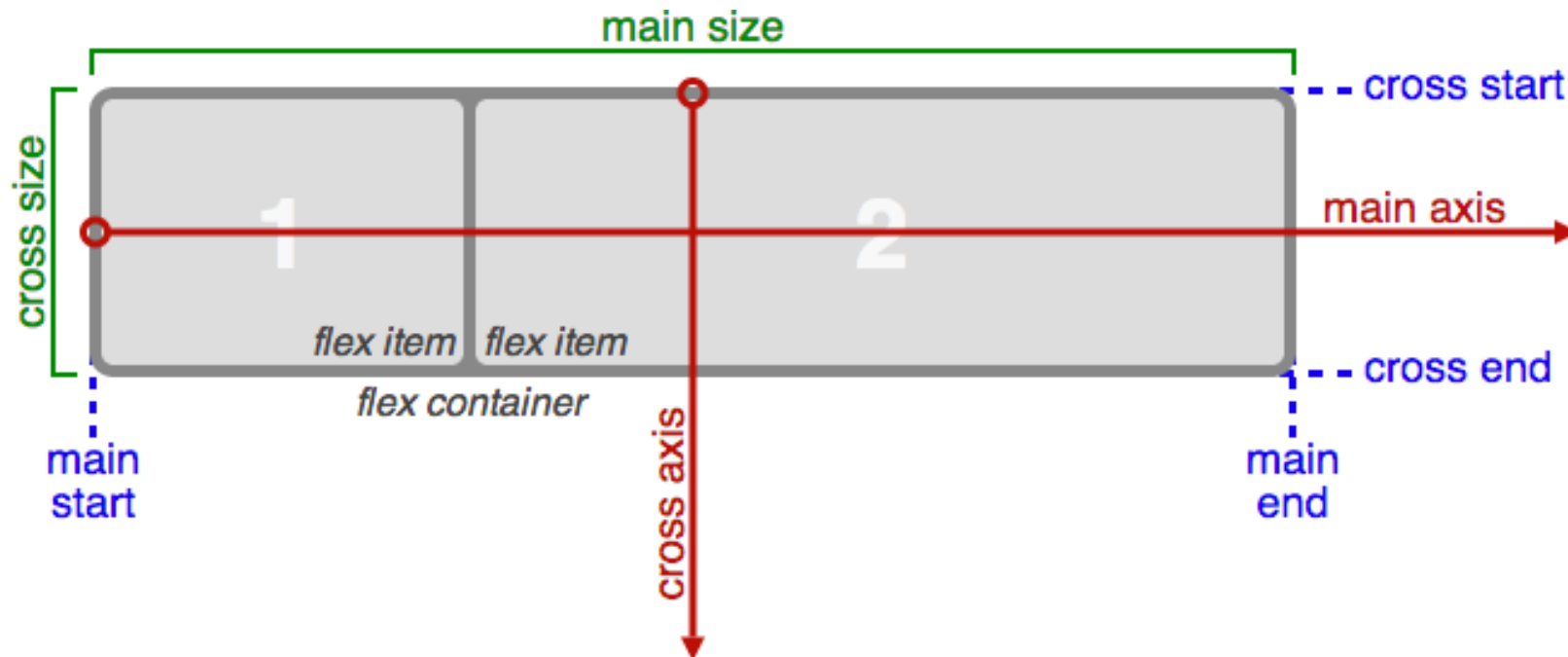
Le positionnement habituel est basé sur les directions de flux *block* et *inline*

Les éléments <i>block</i> crée un schéma vertical	Les éléments <i>inline</i> crée un schéma horizontal
<p><p></p> <p><div></p> <p></p> <p><section></p> <p><h2></p>	<p><small><i><q></p>

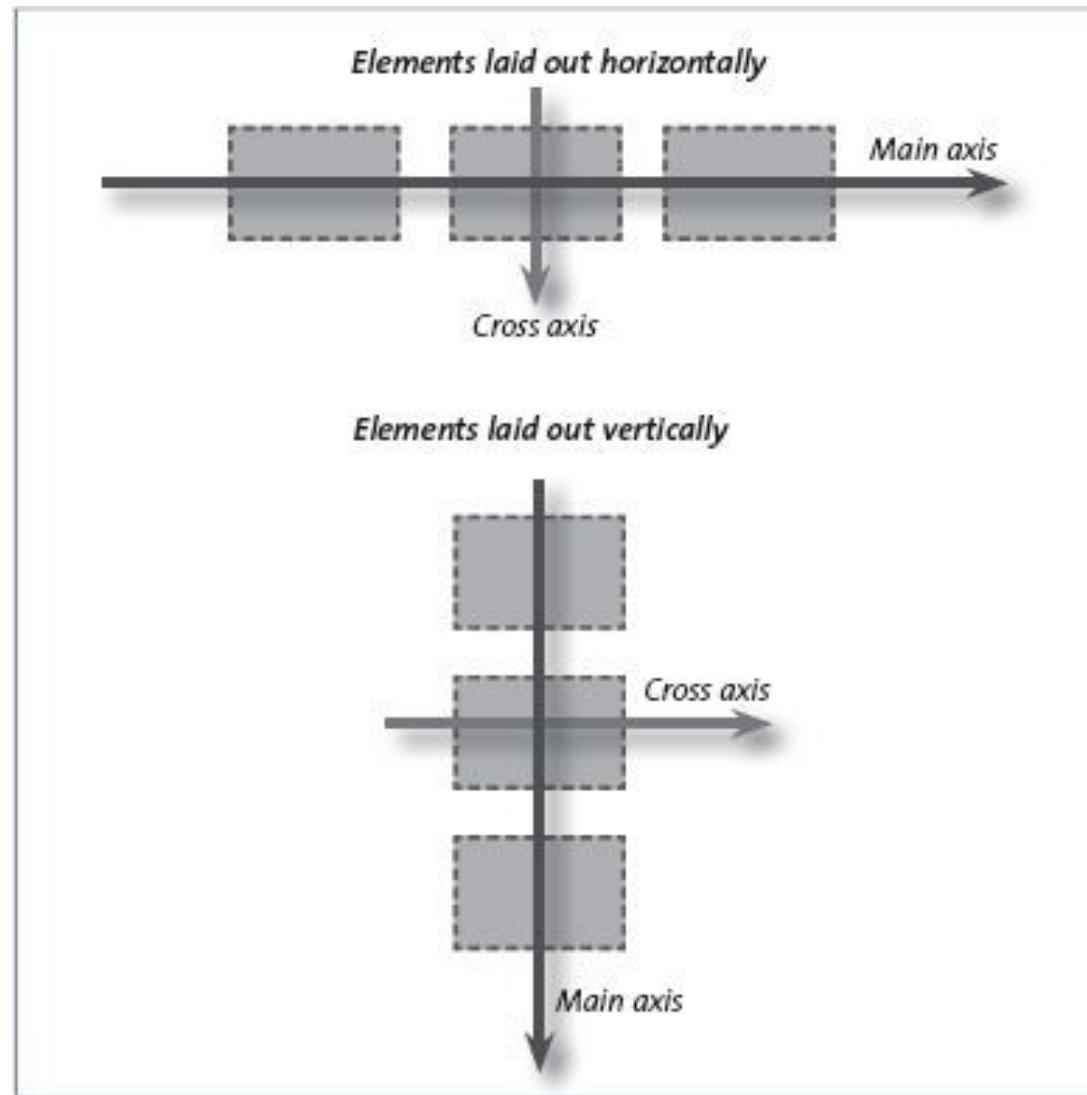
Le positionnement flex

lui, est basé sur la combinaison des directions données par deux axes :

l'axe principal et l'axe secondaire aussi appelé axe **perpendiculaire**.



L'axe principal peut être horizontal ou vertical.



Propriétés du conteneur flex

display

flex | **inline-flex**

C'est ainsi qu'on définit un conteneur flex.

Cela crée un contexte flex pour tous les descendants directs.

La valeur **flex** impose le mode d'affichage externe **block**.

La valeur **inline-flex** impose le mode d'affichage externe **inline**.

Un codepen pour bien distinguer les 2 valeurs possibles de display flex :

<https://codepen.io/evefevrier/pen/MZMEpb>

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 conteneur comprend une seule ligne ou plusieurs et la direction sur l'axe perpendiculaire, qui détermine la direction dans laquelle les nouvelles lignes seront empilées.

Un codepen pour explorer les propriétés du conteneur flex et de ces items :

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

flex-flow

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

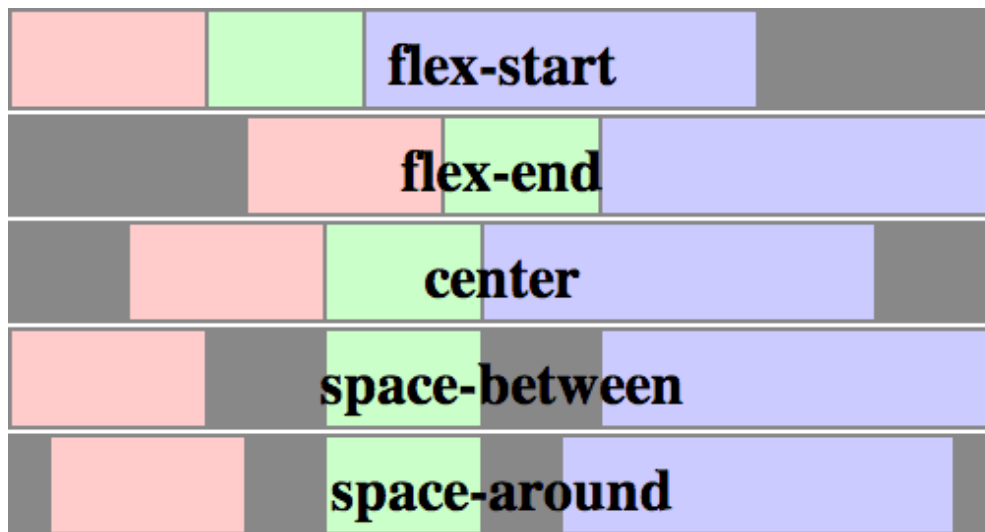
Cette propriété est un raccourci
des propriétés **flex-direction** et **flex-wrap**
qui ensemble définissent les axes principaux et secondaire du conteneur flex.

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 des items flex le long de l'axe principal. Cette propriété est sans effet s'il n'y a pas d'espace disponible à distribuer.



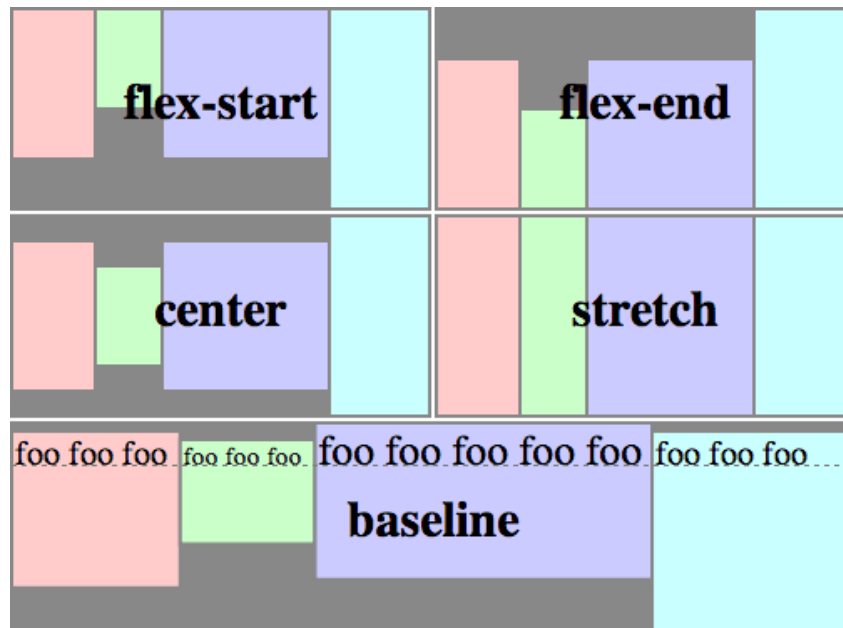
[row] <https://codepen.io/rachelandrew/pen/Owraaj>

[column] <https://codepen.io/rachelandrew/pen/zLyMyV>

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 secondaire.

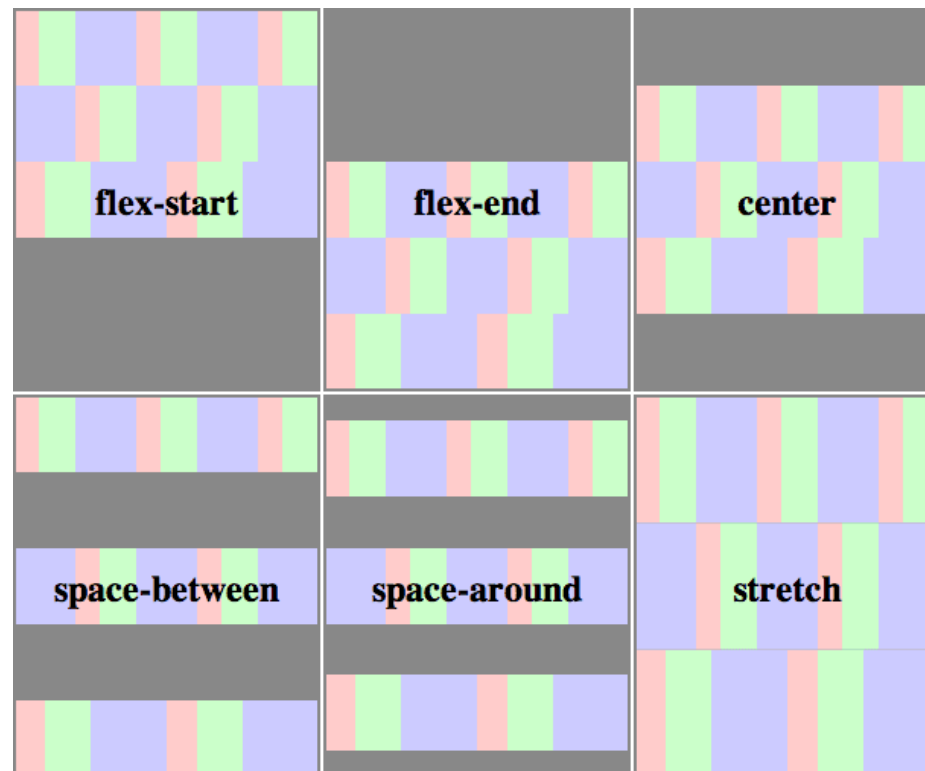


<https://codepen.io/rachelandrew/pen/WKLBpv>

align-content

flex-start | flex-end | center | space-between | space-around | stretch

La propriété **align-content** aligne les lignes d'un conteneur flex à l'intérieur de l'axe secondaire. Cette propriété n'a pas d'effet quand la flexbox n'a qu'une seule ligne ou lorsque le **flex-wrap** est réglé à **no-wrap**.



Propriétés des **flex items**

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 conteneur flex.

Si tous les items ont **flex-grow** défini à 1, chaque enfant aura le même espace dans le conteneur. Si vous donnez à l'un des enfants une valeur de 2, cet enfant prendra deux fois plus de place que les autres, **ou en tout cas il essaiera**.

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**.

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 conteneur.

Ça fonctionne un peu comme un z-index.

Par défaut tout le monde est à 0.

Pour placer un item après les autres on incrémente (+1).

Pour placer un item avant les autres on diminue (-1).

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.



```
.parent {  
  display: flex;  
  flex-direction: column;  
  align-items: stretch;  
}  
  
.oignon {  
  align-self: flex-end;  
}
```

Exercice:

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



Références et ressources

Guide complet en français

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

Version originale en anglais

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

Flexbox-cheatsheet

<https://www.alsacreations.com/outils/lire/1719-flexbox-cheat-sheet-pense-bete.html>

Explorateurs

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

<https://the-echoplex.net/flexyboxes/>