

Flexbox

Pour le moment on a vu que l'on a des éléments de type block et inline, on utilise float pour de la mise en page. On a également vu les tableaux mais ça ne sert pas à mettre un site en page, encore moins un site responsive. Vient alors à la rescousse: Flexbox!

Flexbox permet d'avoir du contenu qui s'adapte, qui est flexible, en fonction de la taille de la page et des containers.

Table des matières

- [Comment ça fonctionne](#)
 - [Les propriétés du container](#)
 - [Flex-direction](#)
 - [Flex-wrap](#)
 - [Flex-flow](#)
 - [Justify-content](#)
 - [Align-items](#)
 - [Align-content](#)
 - [Les propriétés des enfants du container](#)
 - [Order](#)
 - [Flex-grow](#)
 - [Flex-shrink](#)
 - [Flex-basis](#)
 - [Flex](#)
 - [Align-self](#)
- [Quelques liens utiles](#)

Comment ça fonctionne

Pour commencer il nous faut un flex container, ça sera le parent. Dedans on va placer nos éléments. On lui attribue ensuite la propriété **display:flex**. Tous les enfants deviendront du coup des éléments flexibles.

```
<!-- index.html -->
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
```

```
/* style.css */
.flex-container{
display:flex;
}
```

[Voir en pratique sur W3School](#)

Les propriétés du container

[Suivre sur W3School](#)

Flex-direction

Définis dans quel sens vont se positionner les éléments flexible à l'intérieur du container.

```
.flex-container{
  display: flex;
  flex-direction: column; /* column-reverse, row, row-reverse */
}
```

Flex-wrap

Définit si les éléments doivent aller à la ligne ou pas

```
.flex-container{
  display: flex;
  flex-wrap: wrap; /* no-wrap, wrap-reverse */
}
```

Flex-flow

Permet juste de combiner le wrap et la direction

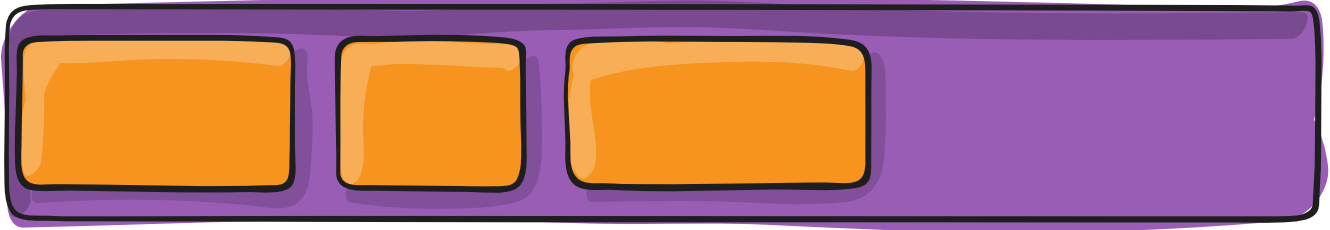
```
.flex-container{
  display: flex;
  flex-flow: row wrap;
}
```

Justify-content

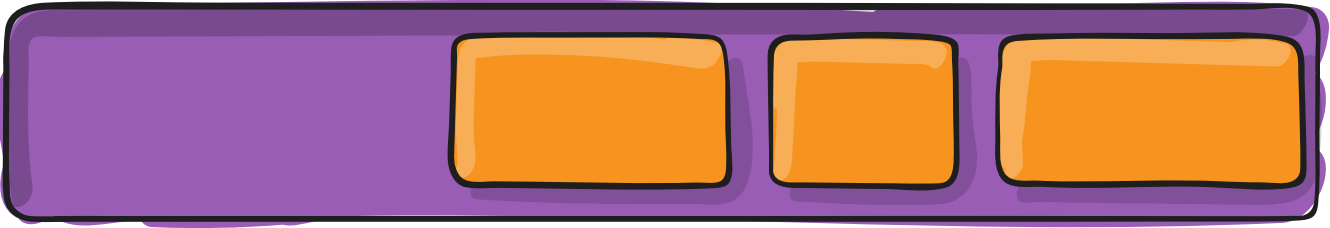
Permet d'aligner horizontalement les éléments à l'intérieur du container

```
.flex-container{
  display: flex;
  justify-content: center; /* flex-start, flex-end, space-around, space-between */
}
```

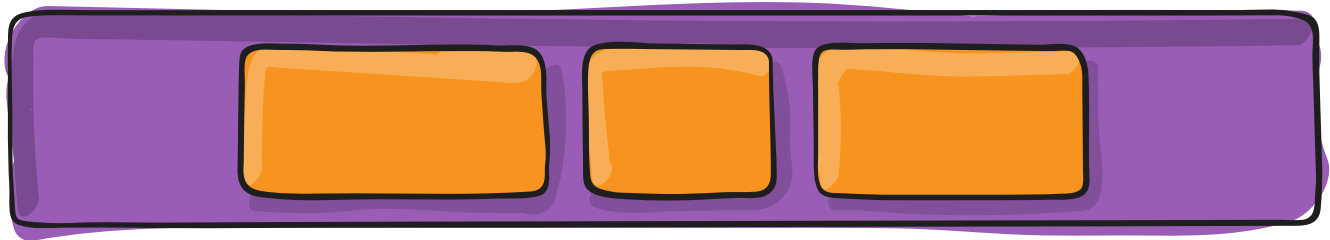
flex-start



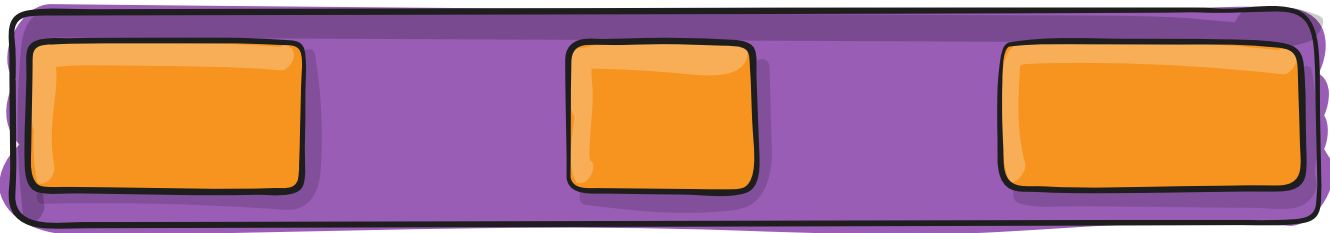
flex-end



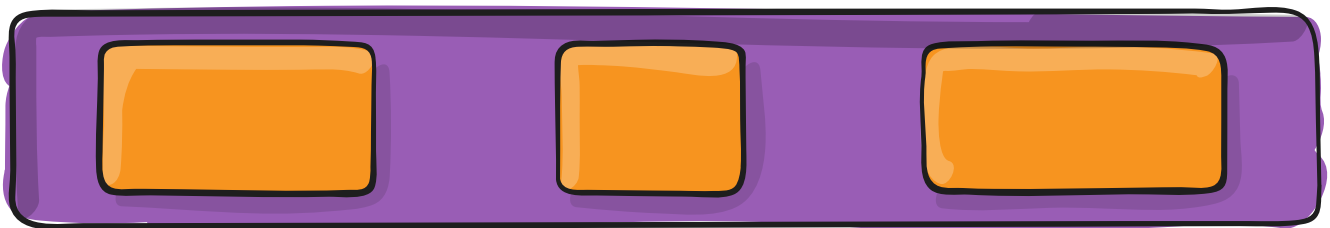
center



space-between



space-around



space-evenly



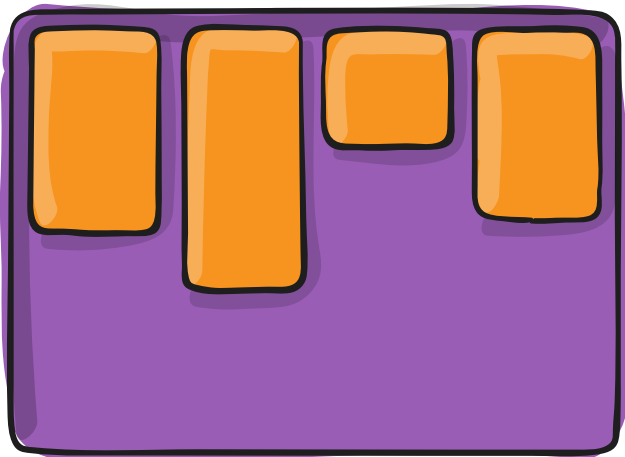


Align-items

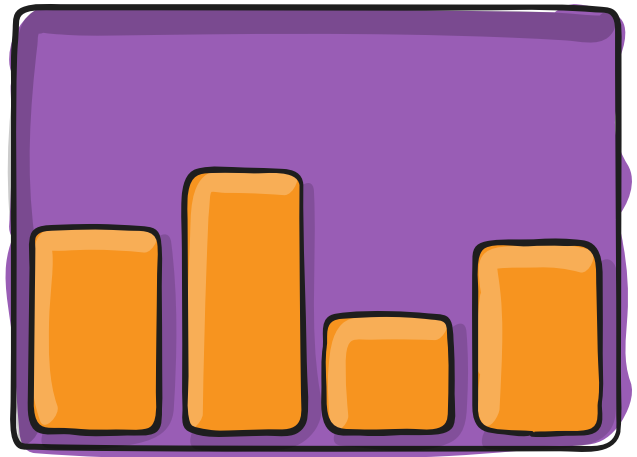
Permet d'aligner verticalement les éléments à l'intérieur du container

```
.flex-container{  
  display: flex;  
  align-items: center; /* flex-start, flex-end, stretch, baseline */  
}
```

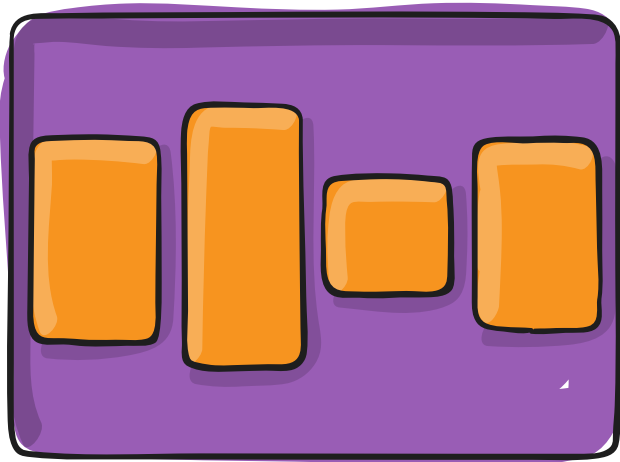
flex-start



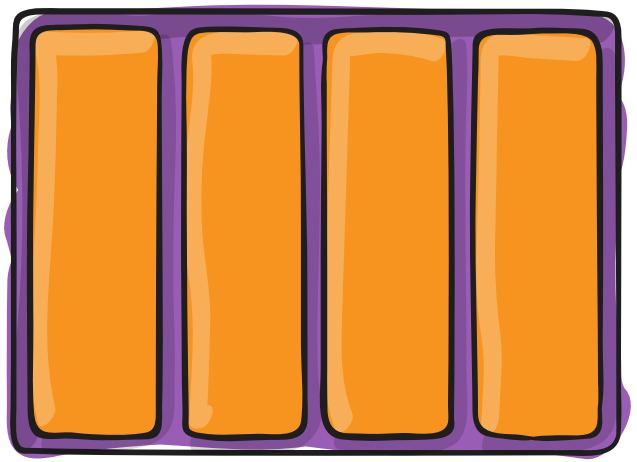
flex-end



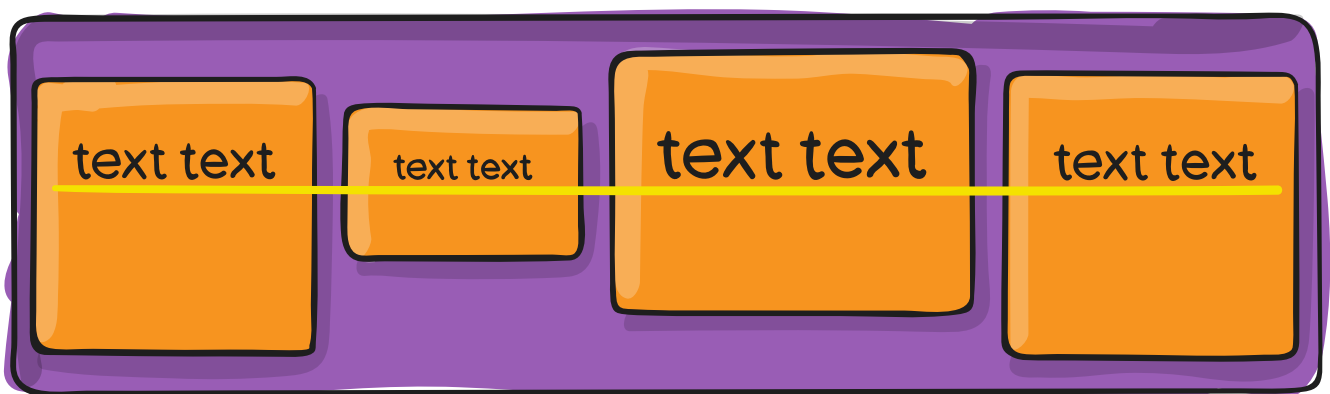
center



stretch



baseline

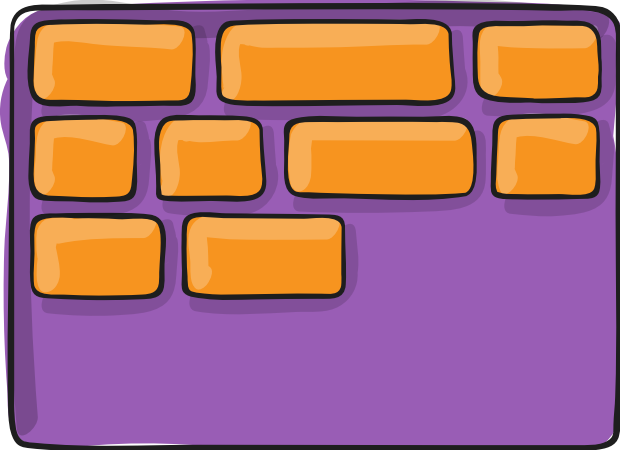


Align-content

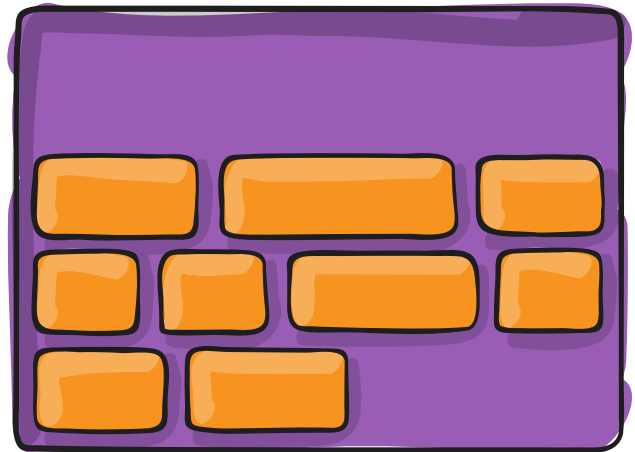
Permet d'aligner les éléments à l'intérieur du container quand il y a plus d'une ligne d'éléments

```
.flex-container{  
  display: flex;  
  align-content: center; /* flex-start, flex-end, stretch, space-between,  
space-around */  
}
```

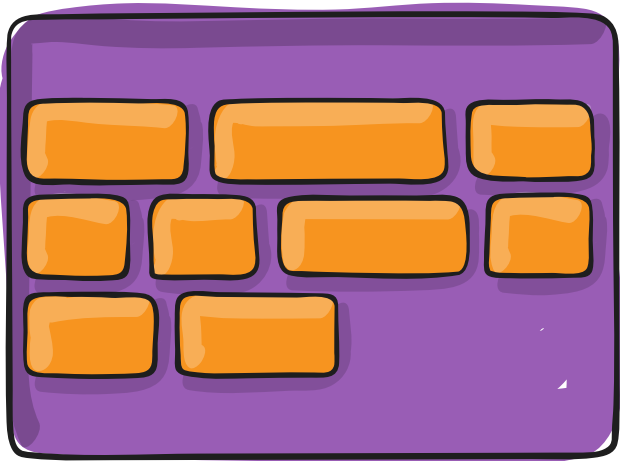
flex-start



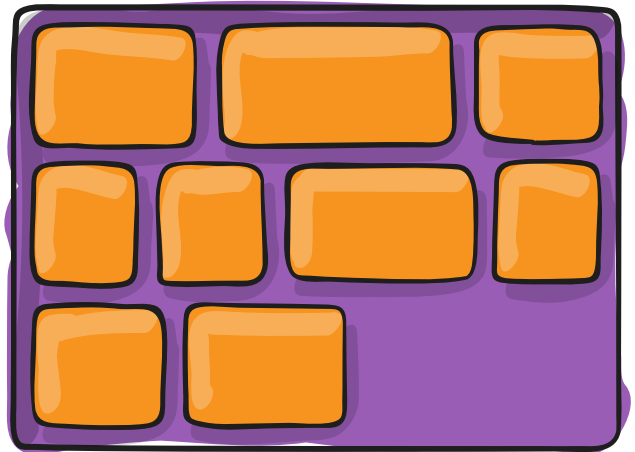
flex-end



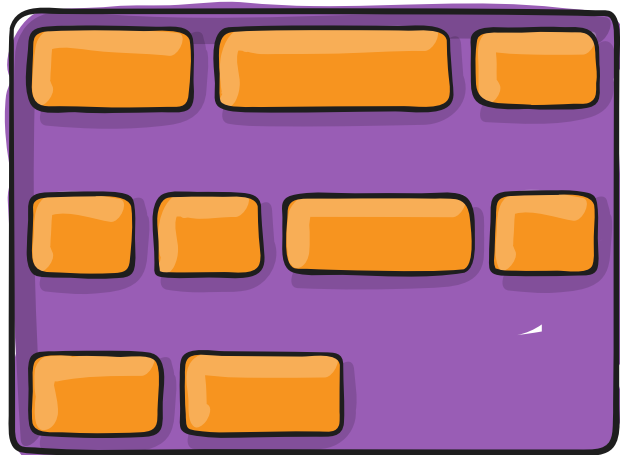
center



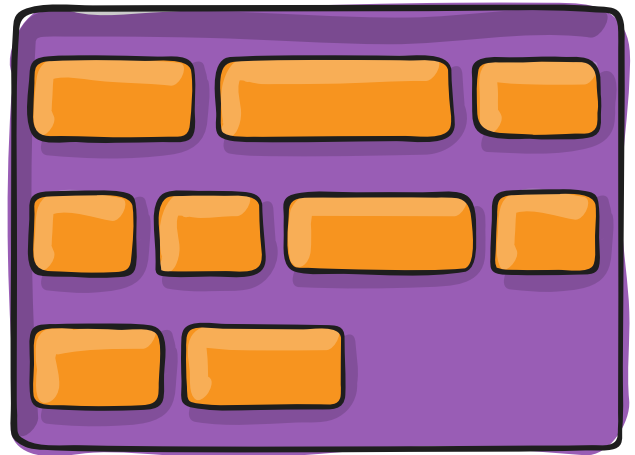
stretch



space-between



space-around



[Suivre sur W3School](#)

Order

Permet de changer l'ordre des éléments. Par défaut les éléments apparaîtront dans l'ordre dans lequel ils sont écrits. On peut donner comme valeur n'importe quel chiffre et du coup l'ordre sera croissant.

```
<!-- index.html -->
<div class="flex-container">
  <div id="element1">1</div>
  <div style="order: 1">2</div>
  <div style="order: 2">3</div>
</div>
```

```
/*styles.css */
#element1 {
  order: 3
}
```

Flex-grow

Permet d'augmenter la taille d'un élément par rapport aux autres. Il suffit de mettre une valeur relative aux autres éléments. La valeur par défaut est 0.

```
#element1{
  flex-grow: 2 /* Double la largeur */
}
```

Flex-shrink

Permet de déterminer si un élément rétrécit par rapport aux autres. Il suffit de mettre une valeur relative aux autres éléments. La valeur par défaut est 1.

```
#element1{
  flex-shrink: 0;
}
```

Flex-basis

Détermine la valeur initial de la longueur de l'élément.


```
#element1{  
  flex-basis: 200px;  
}
```

Flex

Permet de combiner **flex-grow**, **flex-shrink** et **flex-basis**

```
#element1{  
  flex: 0 0 200px;  
}
```

Align-self

Permet de déterminer l'alignement d'un élément dans le container. Cette propriété surpasse l'alignement par défaut défini par **align-items**

```
#element1{  
  align-self: center; /* flex-start, flex-end  
}
```

[Voir en pratique](#)



[Revenir au top](#)

Quelques liens utiles

[Build with Flexbox](#)

[Complete guide to Flexbox](#)



[Revenir au top](#)



[Retour au sommaire du cours](#)