

CSS Flexbox



- > **INFRASTRUCTURES** SYSTÈMES & RÉSEAUX
- > **CYBERSÉCURITÉ** INFRASTRUCTURES & APPLICATIONS
- > **DEVOPS / SCRIPTING** & AUTOMATISATION
- > **DEVELOPPEMENT** WEB & MOBILE
- > **TRANSFORMATION NUMERIQUE DES ENTREPRISES**

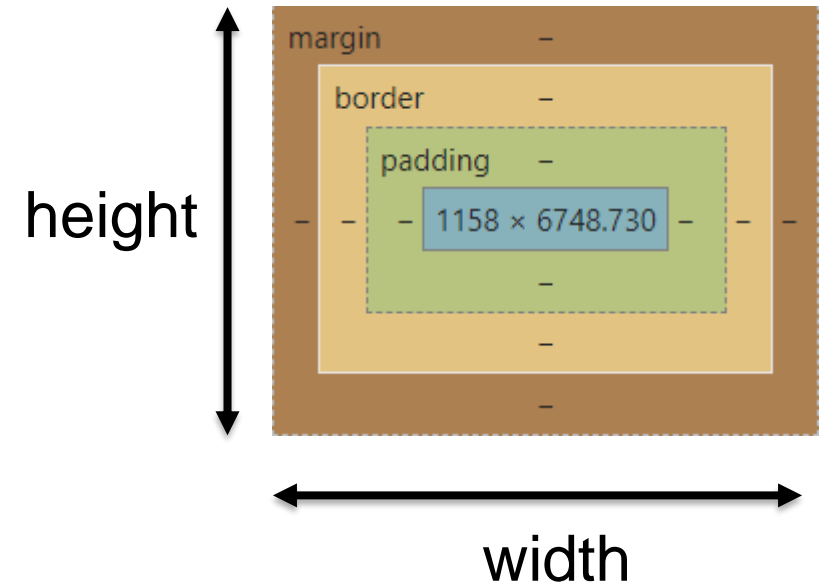
www.adrar-numerique.com

CSS Placements simples: Les boîtes

Le **margin** : l'espace entre la balise
et ce qu'il y a autour

Le **border** : la zone entre le **margin**
et le **padding**

Le **padding** : l'espace entre le **contenu**
et le **border**



Quels sont les moyens de placer du contenu dans une page ?

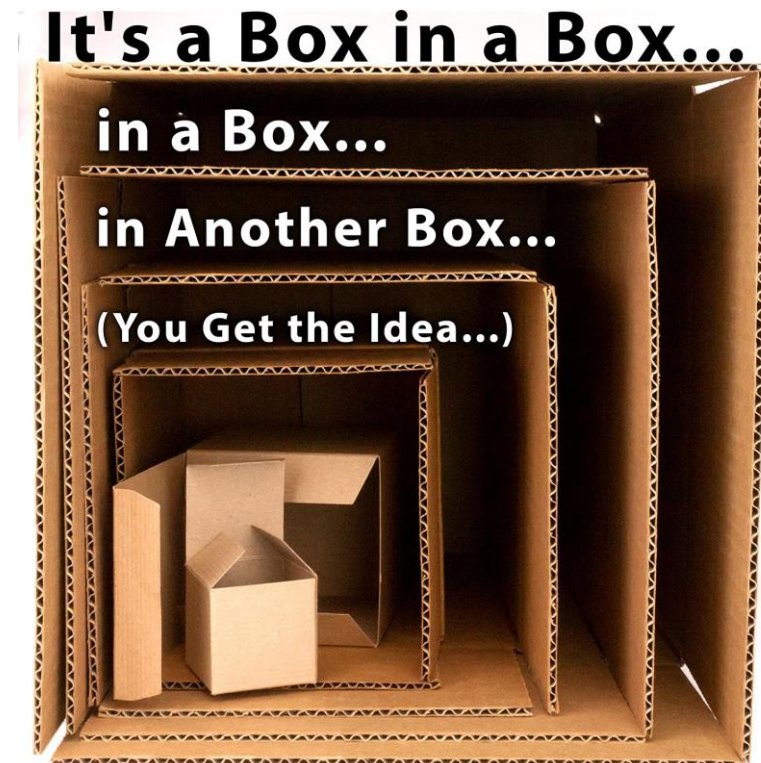
- **Les tableaux** -> Méthode la plus ancienne, dépassée et difficile à mettre en place
- **Les boîtes flottantes** -> Méthode à l'aide de « float », très moyenne et dépassée
- **Les flexbox** -> Méthode récente, boîtes modulaires pratiques et plus simple
- **Les grid** -> Disposition en grille inspirée des tableaux, mais modernisée et simplifiée

Précision : Ces méthodes bien que « concurrentes » ne sont pas incompatibles. Une page peut à la fois utiliser des flexbox et des grid.

CSS Flexbox: Fonctionnement

Le principe est simple : mettre en place un « conteneur » et y placer à l'intérieur plusieurs éléments.

Imaginez une boîte dans laquelle on place d'autres boîtes...



CSS Flexbox: Soyons Flex !

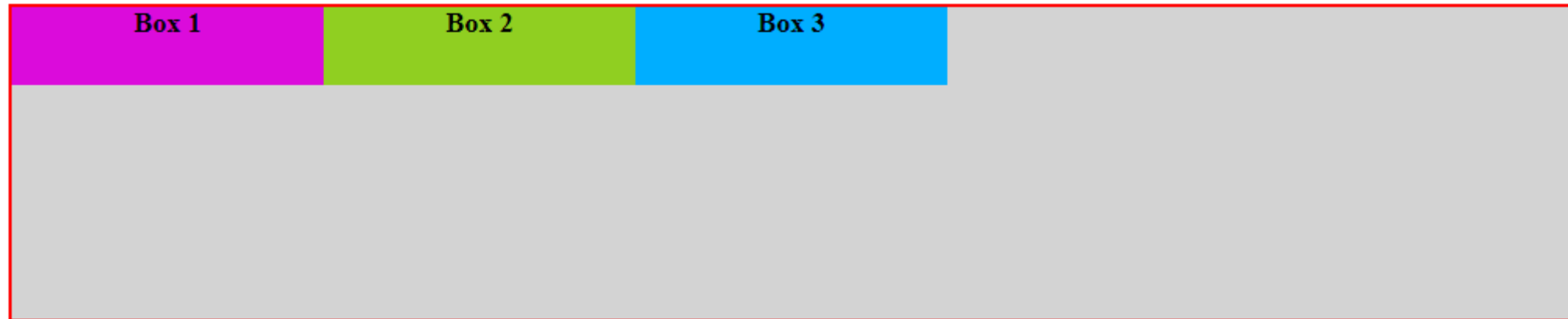
Prenons un conteneur (div) et mettons-y trois conteneurs (div) à l'intérieur pour l'exemple :



```
<body>
  <div id="container">
    <div id="box1" class="box">Box 1</div>
    <div id="box2" class="box">Box 2</div>
    <div id="box3" class="box">Box 3</div>
  </div>
</body>
```


CSS Flexbox: Soyons Flex !

Il suffit de prendre notre conteneur parent (celui en fond gris et bordure rouge) et d'y appliquer la propriété « display: flex »

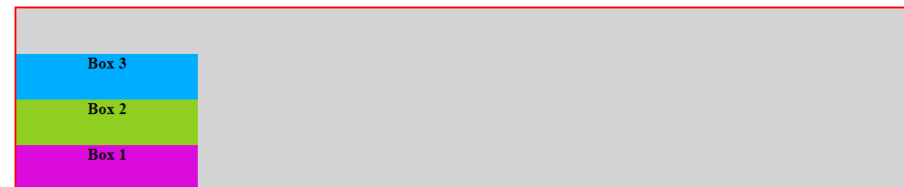
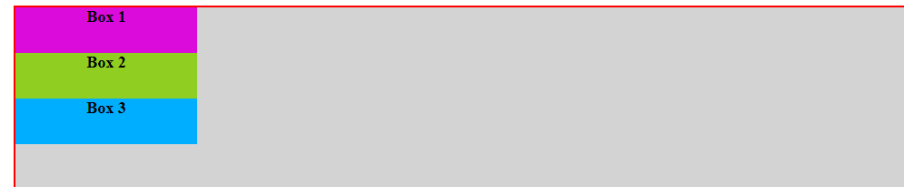
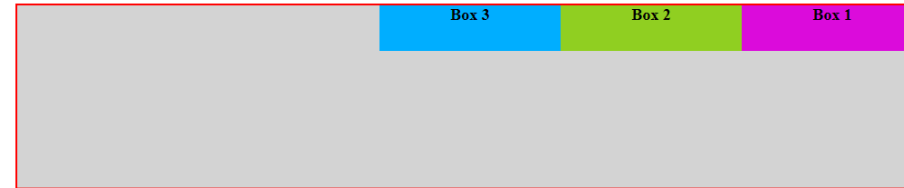
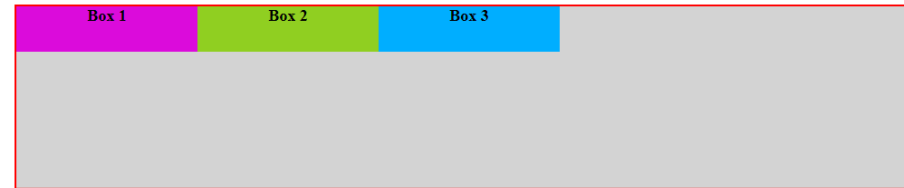


Par défaut, les éléments se placeront sur une ligne et seront alignés en haut à gauche de leur conteneur parent.

```
#container{  
  border: 2px solid red;  
  background-color: lightgray;  
  width: 1000px;  
  height: 200px;  
  display: flex;  
}
```

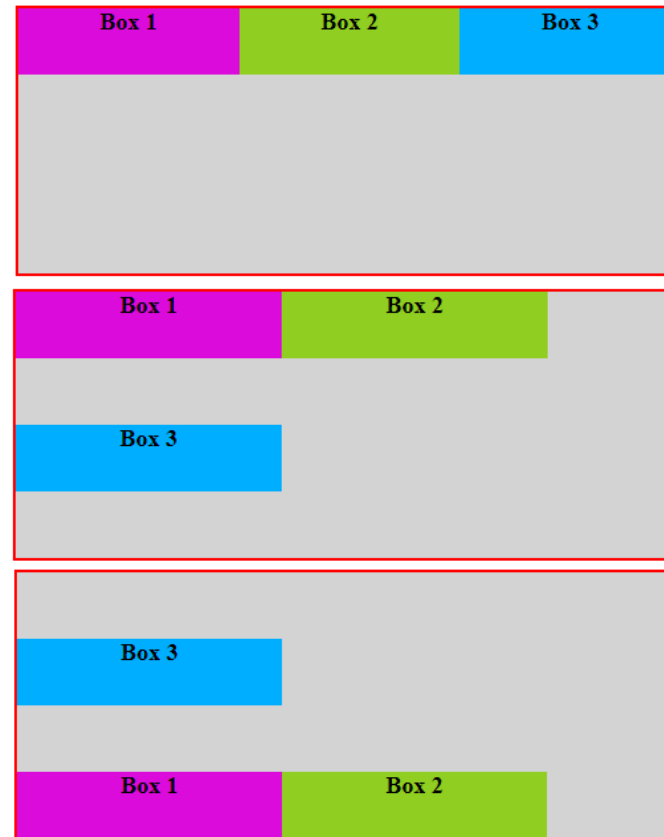
CSS Flexbox: Flex-direction (parent)

- row -> Par défaut
- row-reverse
- column
- column-reverse



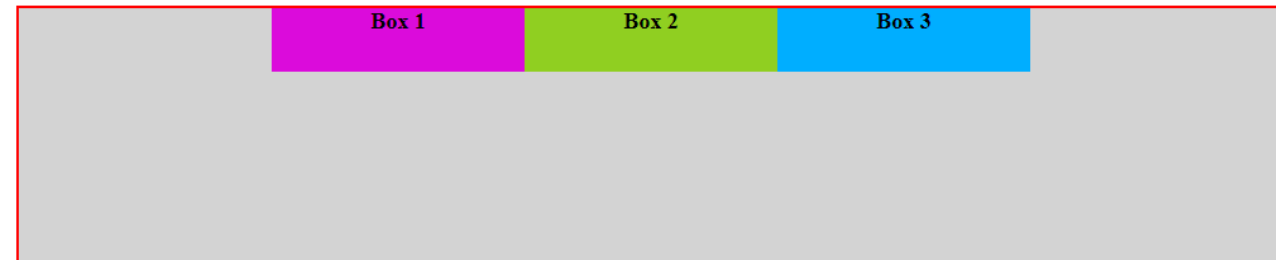
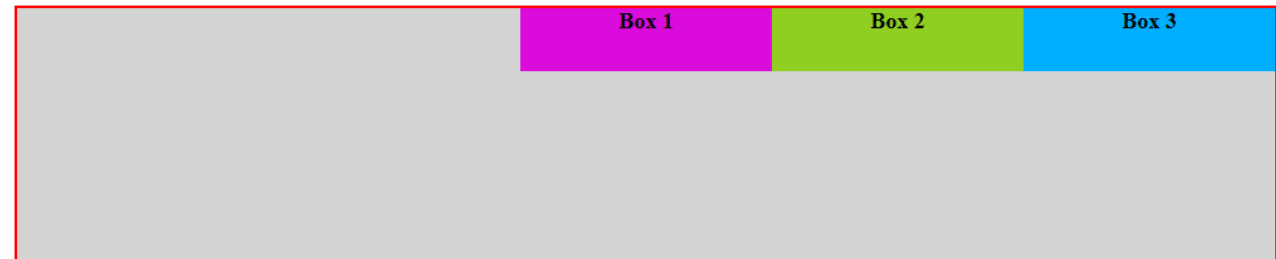
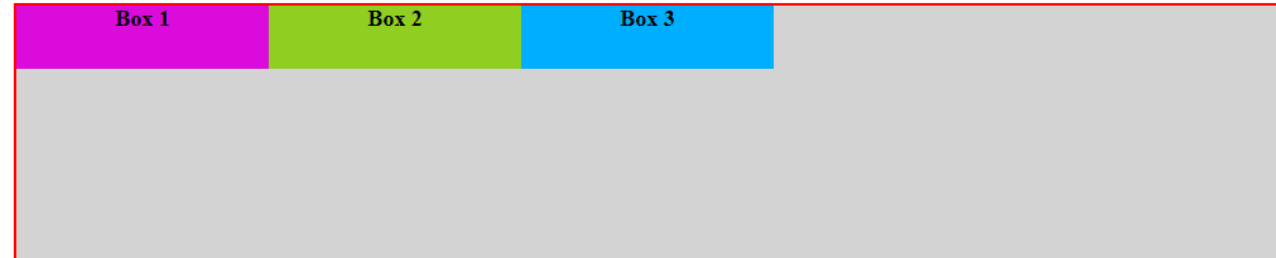
CSS Flexbox: Flex-wrap (parent)

- nowrap -> Par défaut
- wrap
- wrap-reverse



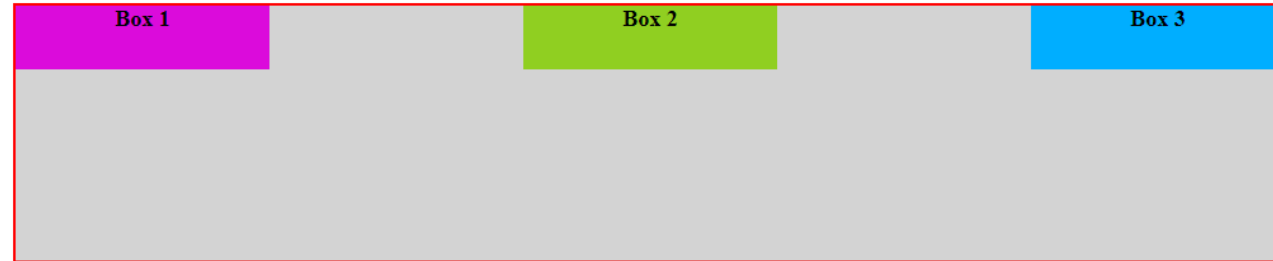
CSS Flexbox: justify-content (parent)

- flex-start -> Par défaut
- flex-end
- center

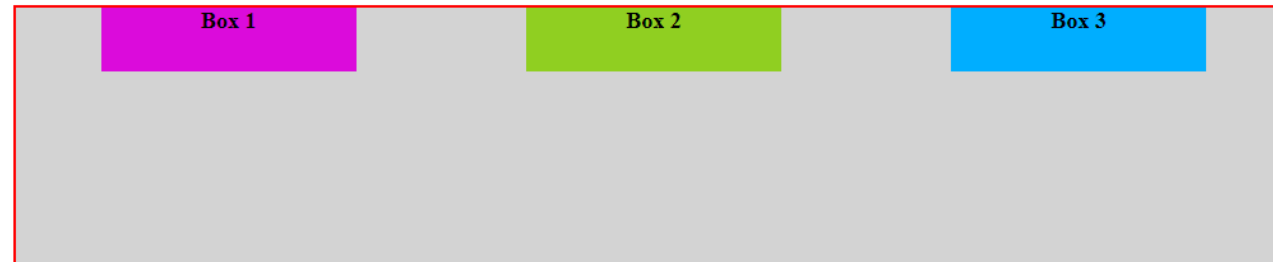


CSS Flexbox: justify-content (parent)

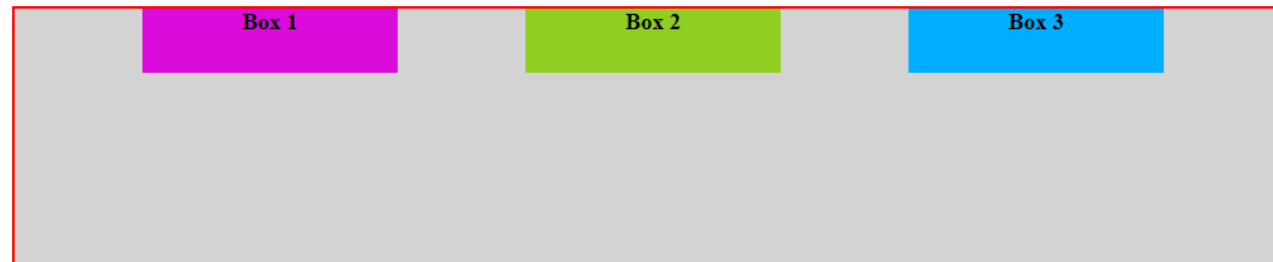
- space-between



- space-around

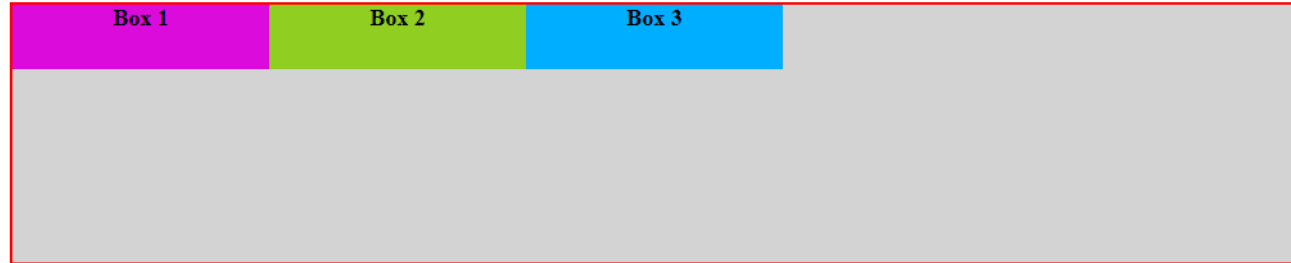


- space-evenly

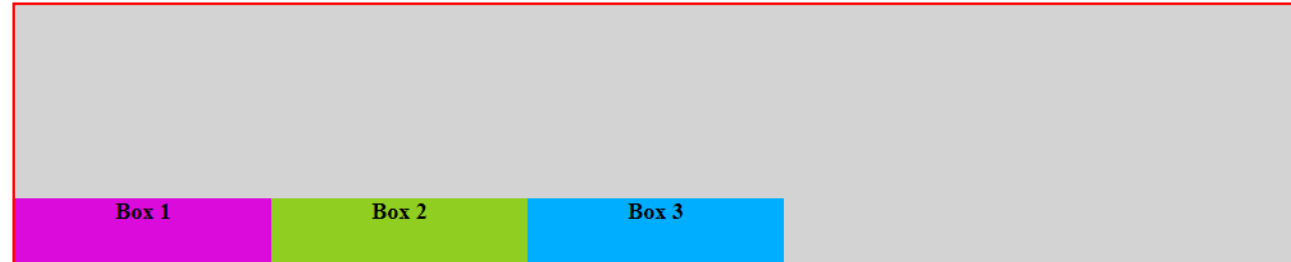


CSS Flexbox: Align-items (parent)

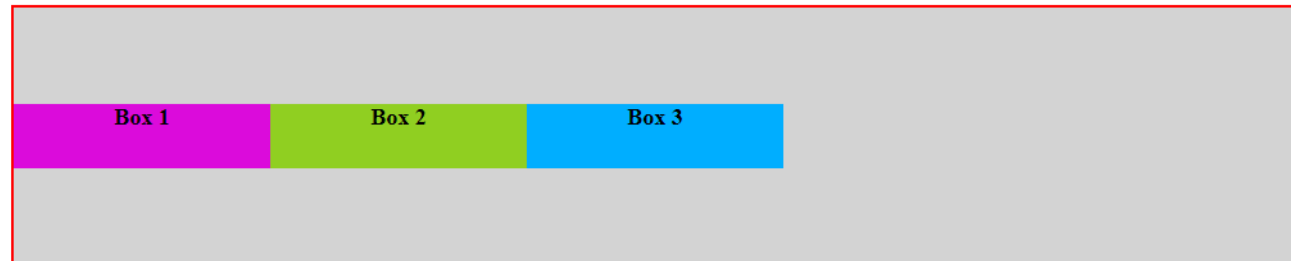
- flex-start -> Par défaut



- flex-end



- center



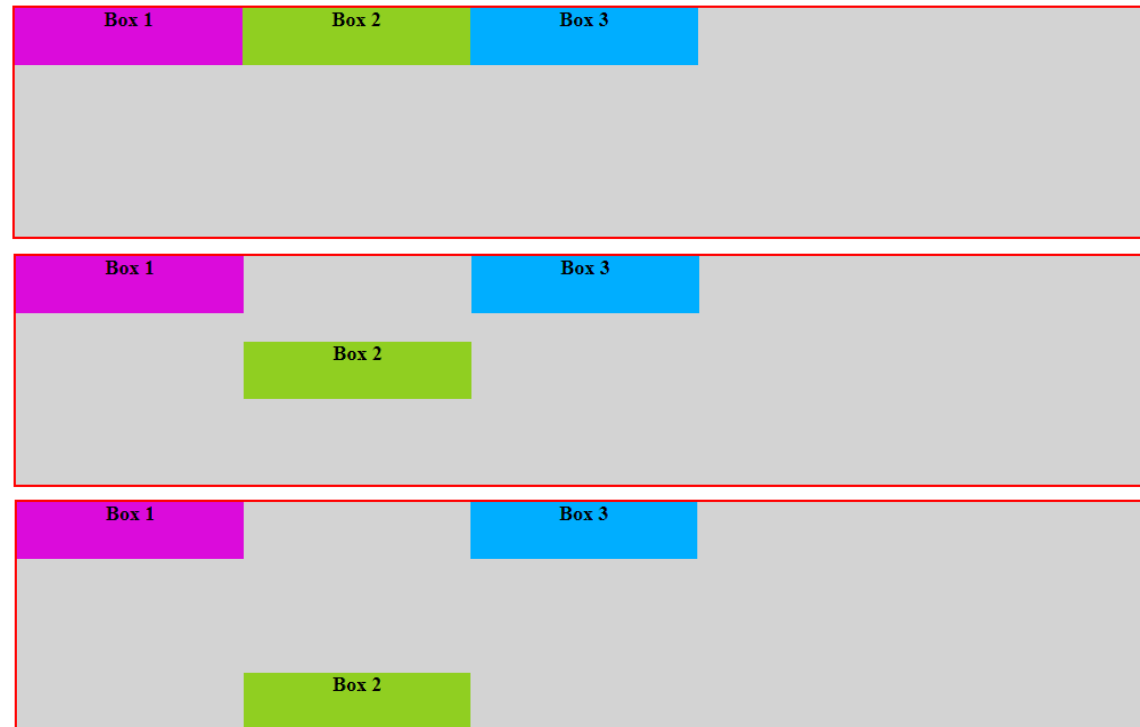
Note : si plusieurs lignes de contenu, utiliser « align-content »

Similaire à « justify-content » mais sur l'autre axe (le même que « align-items »)

CSS Flexbox: align-self (enfant)




Identique à « align-items » mais ne s'applique qu'à un seul élément (Box 2) :

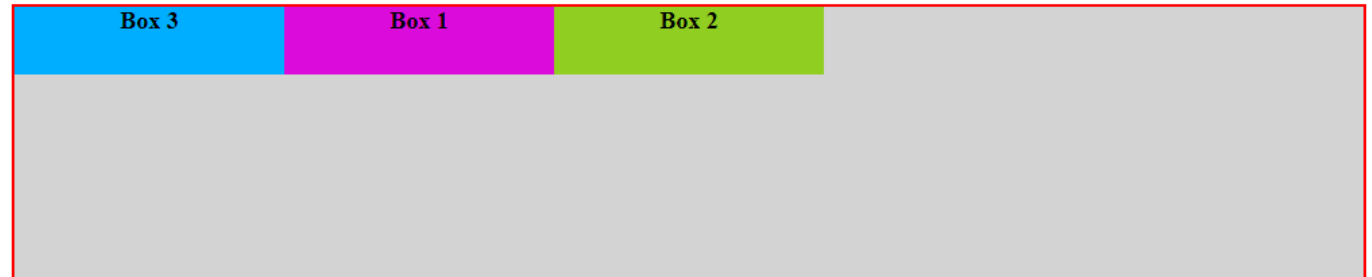
- flex-start
- center
- flex-end



CSS Flexbox: Order (enfant)

Permet de changer l'ordre des éléments avec du CSS :

```
#box1{  
  background-color:  rgb(219, 11, 219);  
  order: 2;  
}  
  
#box2{  
  background-color:  rgb(144, 207, 33);  
  order: 3;  
}  
  
#box3{  
  background-color:  rgb(0, 174, 255);  
  order: 1;  
}
```



CSS Flexbox: Flex (enfant)

Permet d'influencer sur les tailles des éléments :

```
#box1{  
  background-color: #d81b60;   
}  
  
#box2{  
  background-color: #8bc34a;   
  flex: 1;  
}  
  
#box3{  
  background-color: #00bcd4;   
  flex: auto;  
}
```



CSS Flexbox: Aller plus loin !

La doc technique :

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Box_Alignment/Box_Alignment_in_Flexbox

Un site explicatif sur les propriétés des flexbox :

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

Flexbox froggy :

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

CSS Flexbox: Mise en pratique (25 min)

Jouons aux dés avec les Flexbox !

- 1) Utilisez les deux fichiers « main.html » et « main.css » fournis comme base.
 - 2) Modifiez les deux fichiers à votre convenance pour reproduire le visuel demandé !
 - 3) Interdiction d'utiliser les propriétés « margin » ou « padding », tout est faisable avec les flexbox.
 - 4) Pour les plus téméraires, ne rajoutez aucun « id » ni aucune « class », c'est entièrement possible avec les sélecteurs et pseudo-classes CSS 😊
- Indice : vous pouvez, et devez, ajouter des « div » dans le fichier « main.html ».

