

The background features a dark blue gradient. A diagonal line runs from the top-left corner towards the bottom-right. To the left of this line, there are several geometric elements: a dark purple triangle in the top-left; a white circle in the top-left; a light blue square containing concentric circles; a light gray semi-circle; a pink triangle with diagonal lines; a pink square with concentric lines; a light blue square; a light gray triangle; a dark purple triangle; a pink triangle; and a pink square. The text 'CSS FLEXBOX' is written in a bold, white, sans-serif font in the upper right area.

CSS FLEXBOX

CSS : FLEXBOX

INTRODUCTION

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 utilisant la propriété « float », remplacée par des méthodes plus pratiques
- **Les flexbox** -> Méthode récente, boîtes modulaires pratiques et plus simple
- **Les grid** -> Disposition en grille inspirée des tableaux, mais modernisée et simplifiée

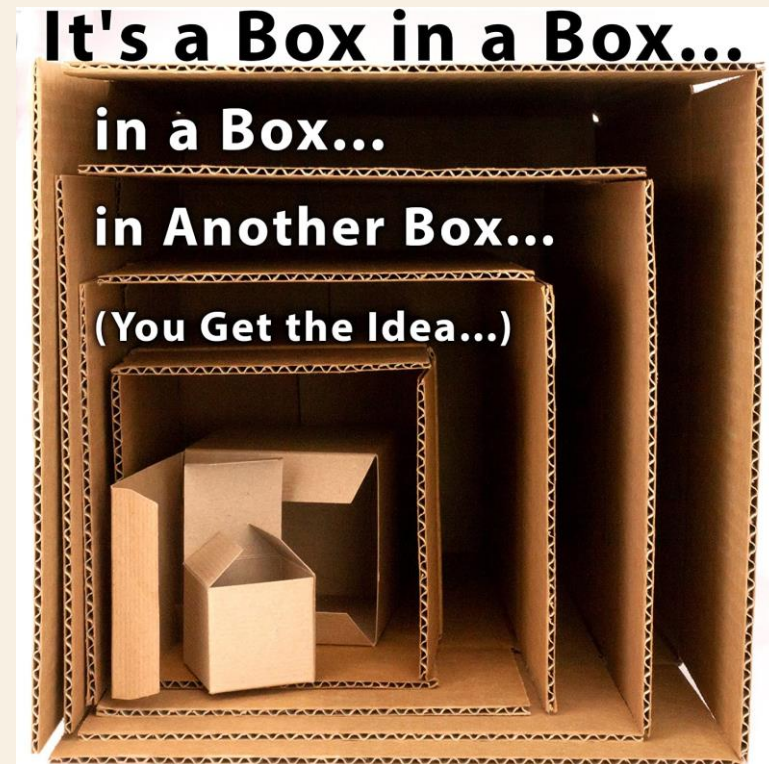
Ces différentes méthodes peuvent être combinées. Ainsi, il n'est pas rare d'utiliser des éléments de positionnement « grid » dans « flexbox » et vice versa.

CSS : FLEXBOX

PRINCIPE GÉNÉRAL

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

FONCTIONNEMENT

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

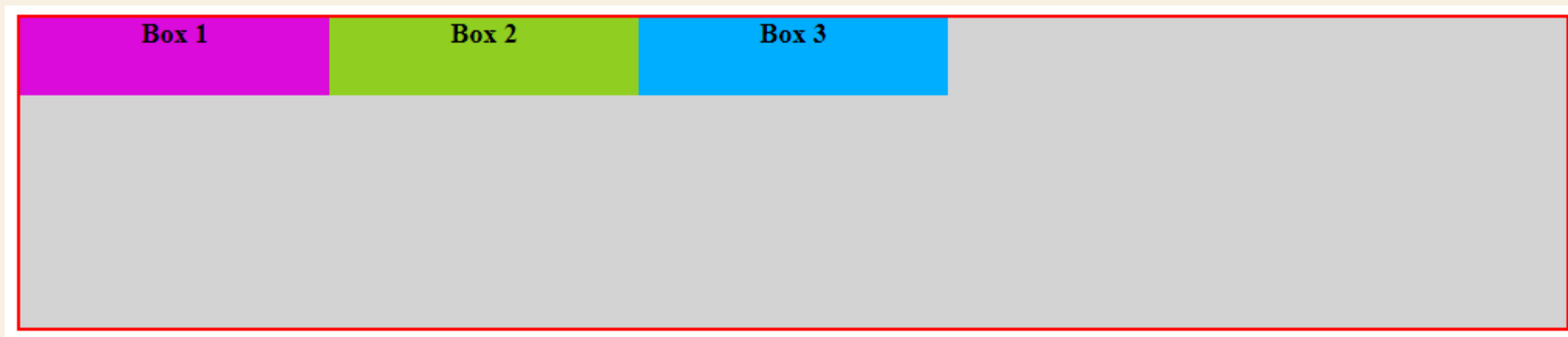


```
<body>
  <div id="container">
    <div id="box1" class="box">Box 1</div>
    <div id="box2" class="box">Box 2</div>
    <div id="box3" class="box">Box 3</div>
  </div>
</body>
```

CSS : FLEXBOX

FONCTIONNEMENT

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



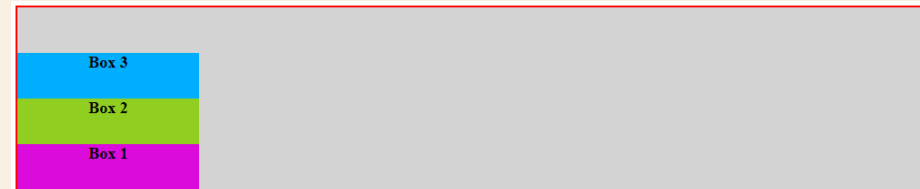
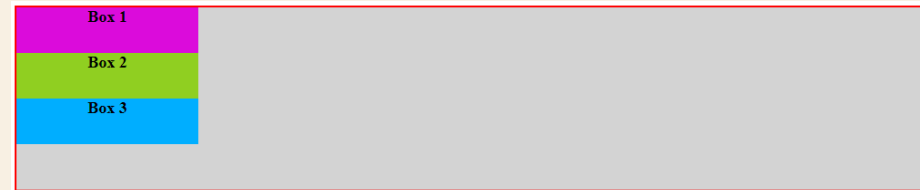
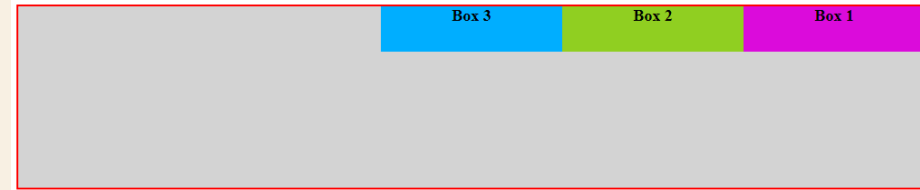
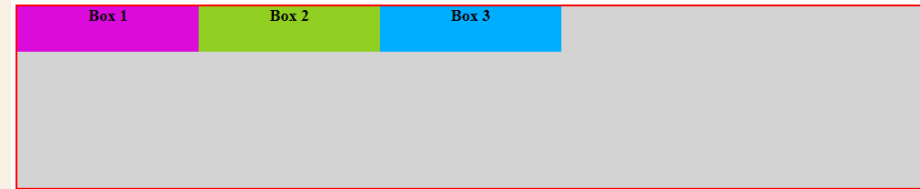
Par défaut, les éléments se placeront sur une ligne et seront alignés en haut à gauche de leur conteneur parent.

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

CSS : FLEXBOX

FLEX-DIRECTION (PARENT)

- La propriété flex-direction définit la façon dont les éléments flexibles sont placés dans un conteneur flexible : elle définit l'axe principal et la direction des éléments (normale ou inversée).
- row -> Par défaut
La direction suit une ligne
- row-reverse
Semblable à <row> mais dans la direction opposée
- column
Les lignes de texte sont empilées
- column-reverse
Semblable à <column> mais dans la direction opposée

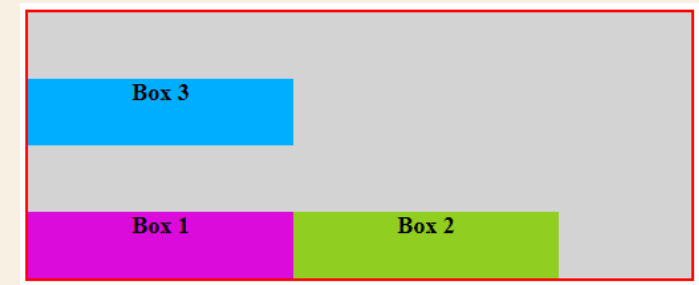
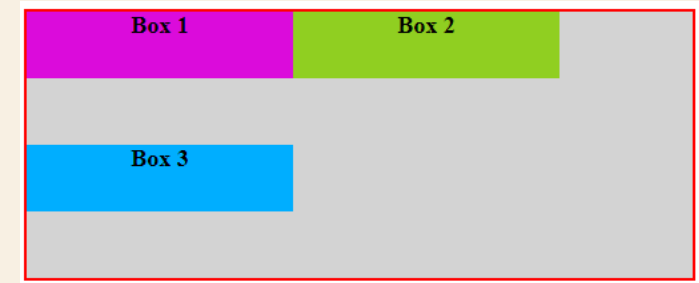
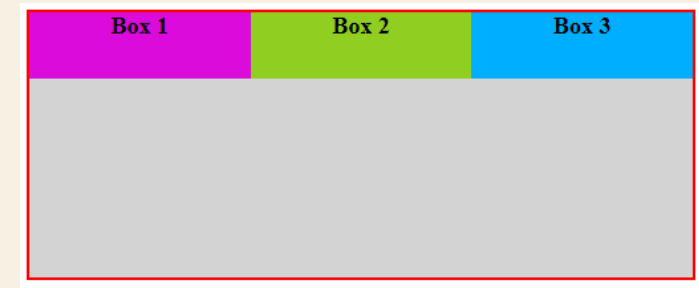


CSS : FLEXBOX

FLEX-WRAP (PARENT)

La propriété flex-wrap indique si les éléments flexibles sont contraints à être disposés sur une seule ligne ou s'ils peuvent être affichés sur plusieurs lignes avec un retour automatique. Si le retour à la ligne est autorisé, la propriété permet également de contrôler la direction dans laquelle les lignes sont empilées.

- nowrap -> Par défaut
Les éléments flexibles sont disposés sur une seule ligne.
Cela peut entraîner un dépassement du conteneur.
- wrap
Les éléments flexibles sont disposés sur plusieurs lignes.
- wrap-reverse

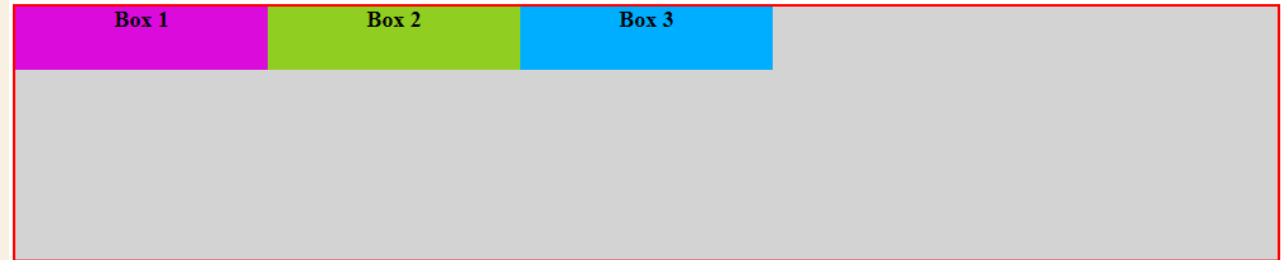


CSS : FLEXBOX

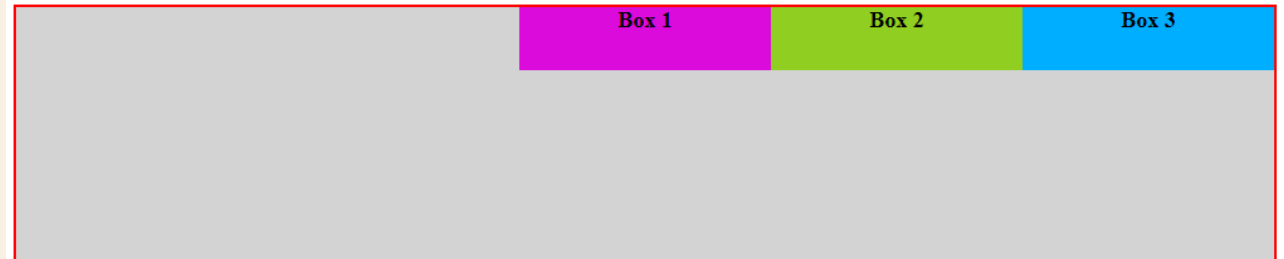
JUSTIFY-CONTENT (PARENT)

La propriété CSS justify-content indique la façon dont l'espace doit être réparti entre et autour des éléments selon l'axe principal d'un conteneur flexible ou selon l'axe en ligne lorsque le conteneur est une grille.

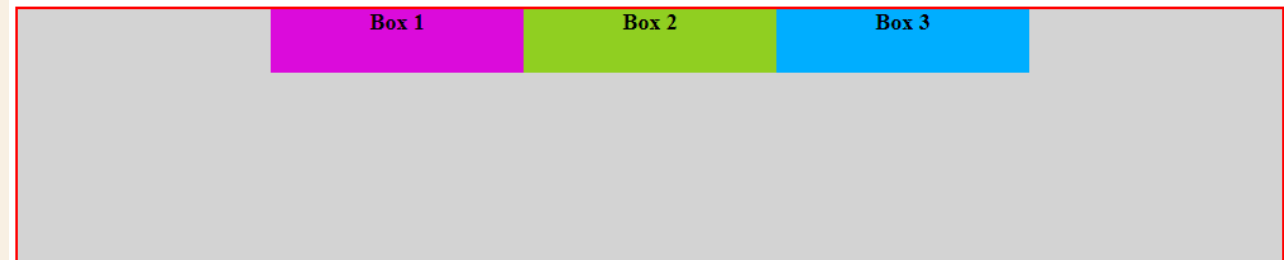
- flex-start -> Par défaut



- flex-end



- center

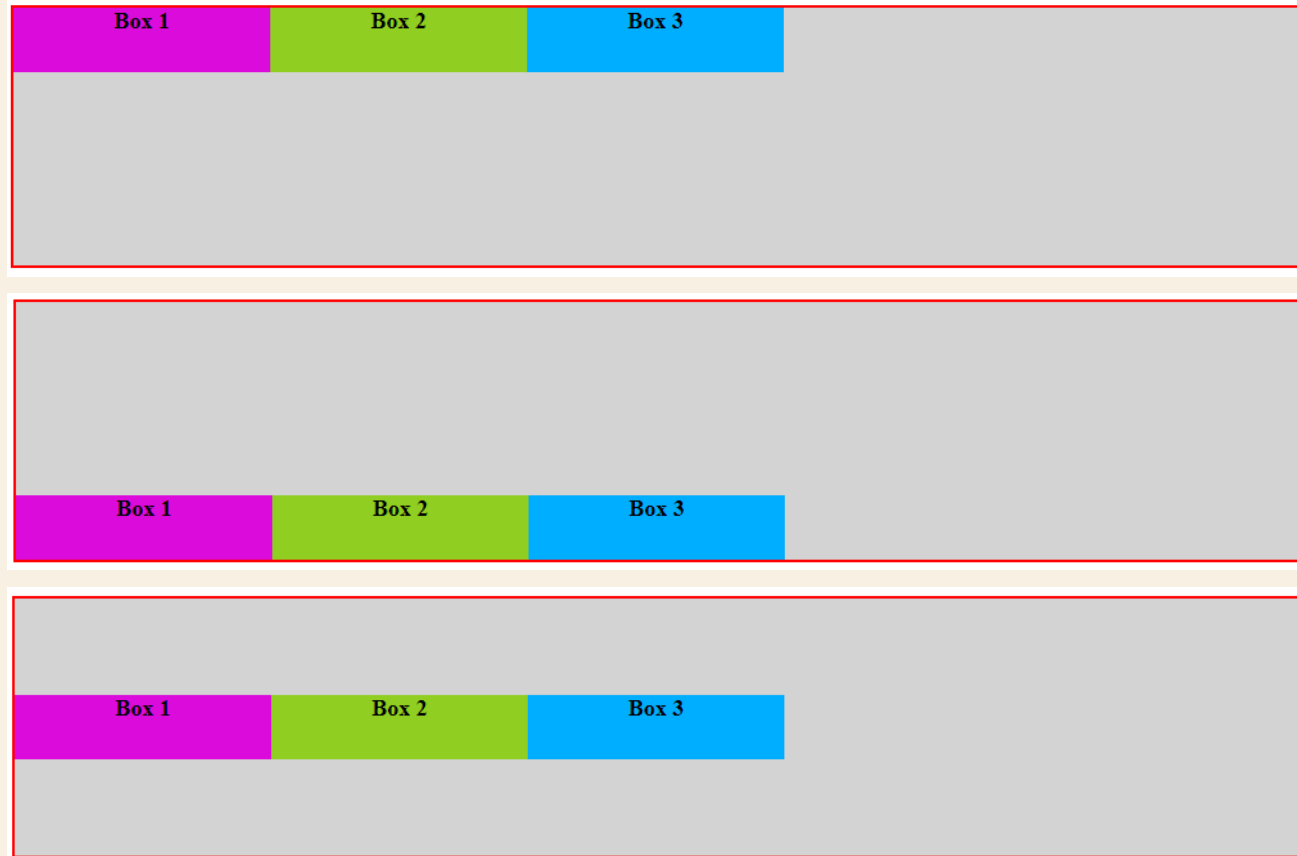


CSS : FLEXBOX

ALIGN-ITEMS (PARENT)

La propriété CSS align-items définit la valeur de align-self sur l'ensemble des éléments-fils directs.

- flex-start -> Par défaut
- flex-end
- Center



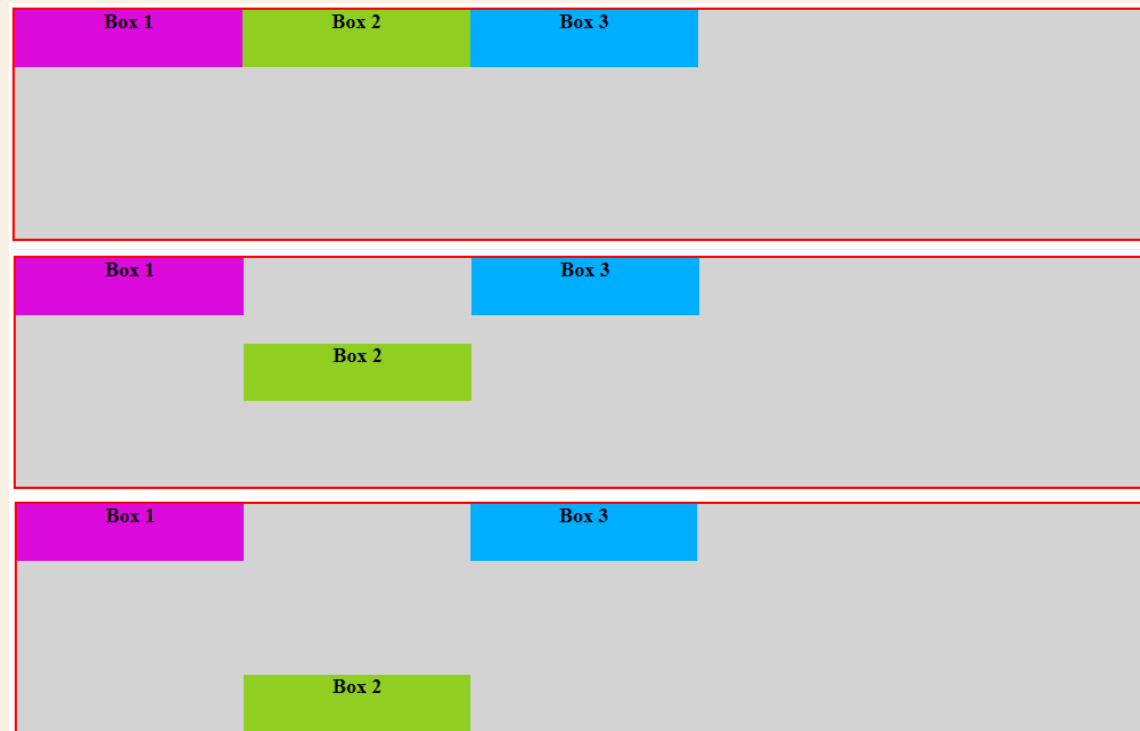
Note : si plusieurs lignes de contenu, utiliser « align-content »
Similaire à « justify-content » mais sur l'autre axe (le même que « align-items »)

CSS : FLEXBOX

ALIGN-SELF (ENFANT)

Identique à « align-items » mais ne s'applique qu'à l'élément sur lequel la propriété est appliqué (self : ici, Box 2) :

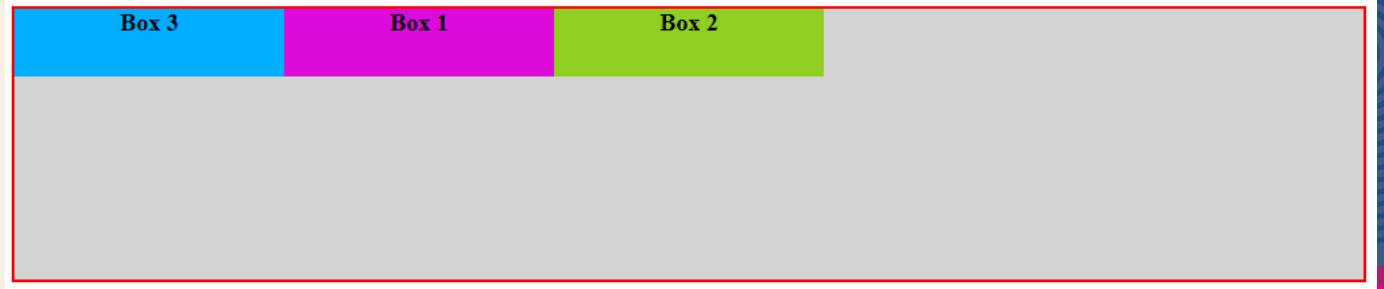
- flex-start
- center
- flex-end



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'influer sur les tailles des éléments :

```
#box1{
  background-color: magenta;
}

#box2{
  background-color: limegreen;
  flex: 1;
}

#box3{
  background-color: cyan;
  flex: auto;
}
```



CSS : FLEXBOX

ALLER PLUS LOIN

Nous sommes loin d'avoir vu tout ce qu'il est possible de faire avec flexbox.

Pour en savoir plus :

- Documentation : 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>
- Pour pratiquer en s'amusant : Flexbox froggy : <https://flexboxfroggy.com/#fr>

CSS : FLEXBOX

MISE EN PRATIQUE

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 ».

Jouons aux dés avec les Flexbox !

