

LES GRILLES ET LES FLEXBOX

Théorie

Introduction aux modèles de mise en page : Flexbox et Grid

CSS propose deux principaux systèmes modernes pour organiser les éléments sur une page web **Flexbox** et **Grid**.

Ces deux méthodes permettent d'aligner et de distribuer des éléments en ligne, en colonnes et en grille. Suffit d'ajouter **display: flex;** ou **display: grid;** aux style d'une balise de type bloc (div, p, section, ...). Puis, ajouter et ajuster **justify-content** selon le type répartitions voulu.

1. FLEXBOX, POUR LES ÉLÉMENTS SUR UNE SEULE LIGNE OU UNE SEULE COLONNE

Flexbox (display: flex;) est conçu pour organiser les éléments sur **un axe principal** :

- Horizontalement (par défaut) ou Verticalement.
- Gère l'espace entre les éléments et leur alignement facilement

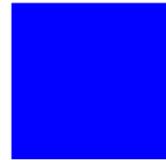
Les propriétés principales de Flexbox :

Propriété	Effet
display: flex;	Active Flexbox sur un conteneur.
flex-direction: row;	Définit l'axe principal en ligne (par défaut).
flex-direction: column;	Définit l'axe principal en colonne .
justify-content	Gère l'alignement sur l'axe principal.
align-items	Gère l'alignement sur l'axe secondaire.

Valeurs de **justify-content** en Flexbox

Valeur	Effet
flex-start	Aligne les éléments à gauche (par défaut).
flex-end	Aligne les éléments à droite.
center	Centre les éléments.
space-between	Répartit l'espace entre les éléments, le premier à gauche, le dernier à droite.
space-around	Ajoute un espace égal autour de chaque élément.
space-evenly	Répartit l'espace de manière égale partout.

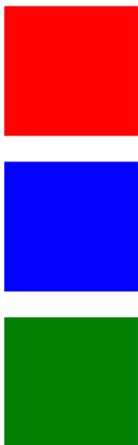
```
<div style="display: flex; justify-content: space-evenly;">
  <div style="width: 100px; height: 100px; background: red;"></div>
  <div style="width: 100px; height: 100px; background: blue;"></div>
  <div style="width: 100px; height: 100px; background: green;"></div>
</div>
</div>
```



```
<div style="display: flex; justify-content: center;">
  <div style="width: 100px; height: 100px; background: red;"></div>
  <div style="width: 100px; height: 100px; background: blue;"></div>
  <div style="width: 100px; height: 100px; background: green;"></div>
</div>
```



```
<div style="display: flex; flex-direction: column; gap: 20px;">
  <div style="width: 100px; height: 100px; background: red;"></div>
  <div style="width: 100px; height: 100px; background: blue;"></div>
  <div style="width: 100px; height: 100px; background: green;"></div>
</div>
```



Dans le dernier exemple, les trois carrés sont maintenant un par-dessus les autres. **justify-content** n'est plus effectif. On utilise **gap** pour spécifier la distance entre chaque éléments. Notons que gap peut aussi être utilisé dans n'importe quel autre cas de figure.

2. CSS GRID : GESTION DES MISES EN PAGE EN DEUX DIMENSIONS

Contrairement à **Flexbox**, **CSS Grid** permet d'organiser les éléments en **lignes et colonnes simultanément**. Ça fait des grilles ! C'est utile pour organiser du contenu mais aussi pour structurer toute une page au complet. Par exemple, le **header** pourrait occuper l'espace du haut, le **nav** les colonnes du centre gauche, le **main** plusieurs colonnes et lignes au centre et le **footer** des colonnes dans le bas de la page. Mais gardons ça pour un peu plus tard. Pour le moment, voici le basic de **grid**.

Les propriétés principales de Grid :

Propriété	Effet
display: grid;	Active Grid sur un conteneur.
grid-template-columns	Définit le nombre et la taille des colonnes . Simplement, le nombre de paramètre et leur valeur indique le nombre de colonne et leur largeur. ex grid-template-columns : auto auto auto; donne trois colonnes d'égale largeur. On peut aussi utiliser des valeurs en pixels et en pourcentages. Les valeurs peuvent être identique ou différentes pour chaque colonnes.
grid-template-rows	Définit le nombre et la taille des lignes . La pluspart du temps on ne l'utilise pas car les rangées sont automatiquement créées lorsque les colonnes sont pleines.
justify-content	Gère l'alignement global de la grille.
justify-items	Centre les éléments à l'intérieur des cellules .

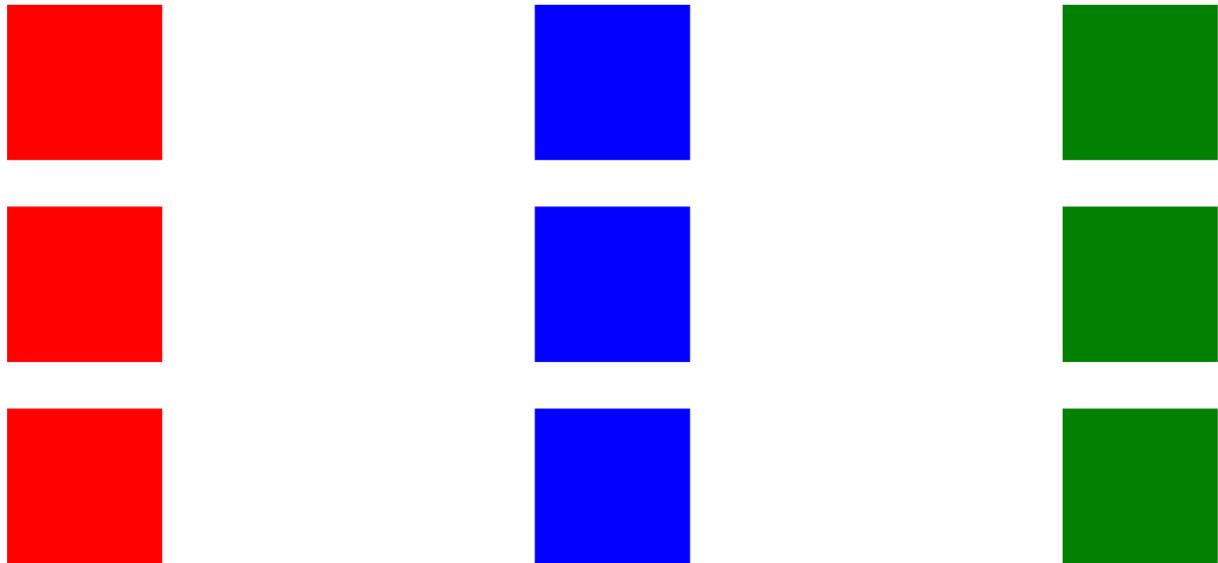
Différences entre justify-content en flex et en grid

Propriété	En flexbox (display: flex;)	En grid (display: grid;)
justify-content: flex-start;	Aligne les éléments au début de l'axe principal	Aligne toute la grille à gauche
justify-content: flex-end;	Aligne les éléments à la fin de l'axe principal	Aligne toute la grille à droite
justify-content: center;	Centre les éléments sur l'axe principal	Centre toute la grille dans son conteneur
justify-content: space-between;	Espacement égal entre les éléments, le premier à gauche et le dernier à droite	Répartit les colonnes sur toute la largeur disponible
justify-content: space-around;	Ajoute un espace égal autour de chaque élément	Ajoute un espace autour de chaque colonne de la grille
justify-content: space-evenly;	Espace égal entre et autour des éléments	Répartit uniformément les colonnes

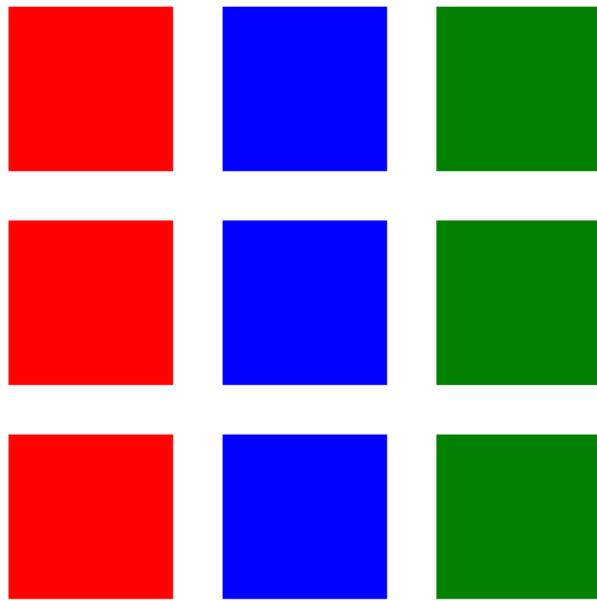
Si l'on veut centrer **chaque élément individuellement dans la grille**, il faut utiliser :

- justify-items: center; (pour toutes les cellules)
- justify-self: center; (pour un seul élément)

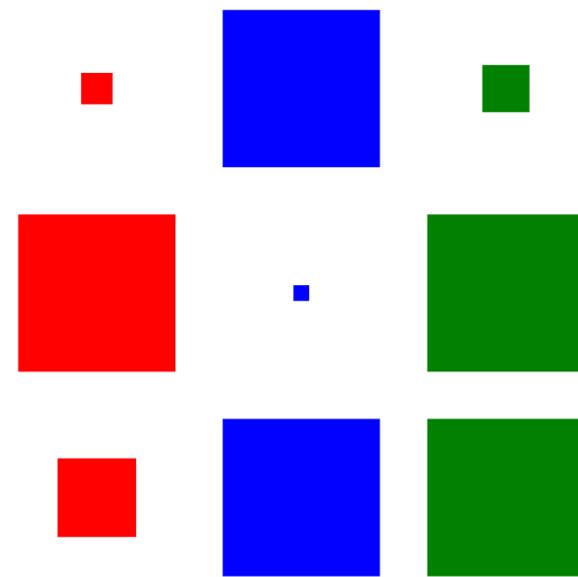
```
<div style="display: grid; grid-template-columns: auto auto auto; justify-content: space-evenly; gap: 30px;">
  <div style="width: 100px; height: 100px; background: red;"></div>
  <div style="width: 100px; height: 100px; background: blue;"></div>
  <div style="width: 100px; height: 100px; background: green;"></div>
  <div style="width: 100px; height: 100px; background: red;"></div>
  <div style="width: 100px; height: 100px; background: blue;"></div>
  <div style="width: 100px; height: 100px; background: green;"></div>
  <div style="width: 100px; height: 100px; background: red;"></div>
  <div style="width: 100px; height: 100px; background: blue;"></div>
  <div style="width: 100px; height: 100px; background: green;"></div>
</div>
```



Le gap est ici utilisé pour l'espace entre les colonnes puisque **space-evenly** force l'espace horizontal. Avec `justify-content: center;` on aurait ceci



```
<div style="display: grid; grid-template-columns: auto auto auto; justify-content: center; justify-items: center; align-items: center; gap: 30px;">
    <div style="width: 20px; height: 20px; background: red;"></div>
    <div style="width: 100px; height: 100px; background: blue;"></div>
    <div style="width: 30px; height: 30px; background: green;"></div>
    <div style="width: 100px; height: 100px; background: red;"></div>
    <div style="width: 10px; height: 10px; background: blue;"></div>
    <div style="width: 100px; height: 100px; background: green;"></div>
    <div style="width: 50px; height: 50px; background: red;"></div>
    <div style="width: 100px; height: 100px; background: blue;"></div>
    <div style="width: 100px; height: 100px; background: green;"></div>
</div>
```



Ici, chaque élément est centré individuellement au centre de sa cellule de grille et la grille est centré au milieu de la page. **justify-items: center;** aligne chaque élément au centre horizontal de sa cellule et **align-items: center;** aligne chaque élément au centre vertical de sa cellule dans une grille qui est au centre de la page **justify-content: center;**