

Généralités

- **Technique de mise en page** sous forme de **grille** qui permet la création de dispositions complexes en CSS pour les documents HTML.

- Permet l'alignement, le centrage et la distribution d'éléments en fonction de deux axes :

- L'axe **vertical** (ou axe des colonnes) →
- L'axe **horizontal** (ou axe des rangées) →



Différences avec le flexbox

- **Flexbox = unidimensionnel** : ne gère **qu'une direction à la fois** (soit les rangées, soit les colonnes).
- **Grid = bidimensionnel** : gère **deux directions à la fois** (rangées + colonnes). Permet de placer les éléments dans les lignes et colonnes choisies, peu importe où se trouvent les lignes et colonnes.

- **Définition d'un conteneur grille**

```
.container{
  display: grid; /* ou inline-grid */
}
```

→ les enfants directs d'un conteneur grille sont des éléments de grille.

Quelques propriétés CSS grid

Propriétés s'appliquant au parent (conteneur grille)

- **grid-template-rows** : définit le nombre de rangées et leur hauteur.
- **grid-template-columns** : définit le nombre de colonnes et leur largeur en valeur fixe (en px) ou en valeur flexible (% ou em).
- *Exemple :*

```
.container{
  grid-template-columns: 25% 75%;
  grid-template-rows: auto;
}
```

"auto" attribue un espace qui s'adapte automatiquement au contenu



- **L'unité fr (fraction)** : utilisable uniquement avec le grid, unité de valeur flexible qui repose sur une fraction de l'espace disponible.

```
.container{
  grid-template-columns: 50px 1fr 2fr;
}
```



- Si **répétition de mêmes valeurs**, possibilité d'utiliser **repeat** :

```
.container{
  grid-template-columns: repeat(4, 25%);
}
```

- **grid-template-areas** : définit un modèle de grille en référençant les noms des zones spécifiées avec la propriété **grid-area**.

- Un **point** ou la mention **none** signifie une zone vide.

```
.container{
  grid-template-areas:
    "one two three"
    "four . five";
}
```

- **grid-template** : raccourci pour définir **grid-template-rows**, **grid-template-columns** et **grid-template-areas** en une seule déclaration.

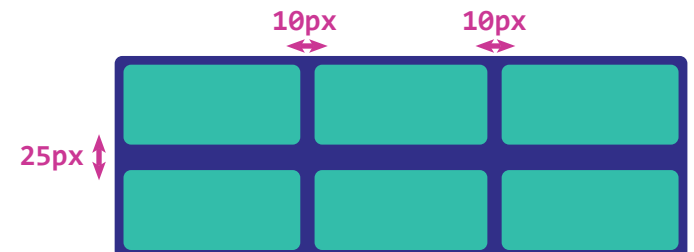
- **row-gap** : définit la taille des gouttières entre les rangées.

- **column-gap** : définit la taille des gouttières entre les colonnes

- **gap** : raccourci pour définir la taille des gouttières entre rangées et colonnes.

```
.container{
  gap: 25px 10px;
}
```

column-gap (points to 10px)
row-gap (points to 25px)



- **justify-content** : alignement du conteneur de grille le long de l'axe horizontal.
- **start** : grille au début du conteneur.



- **end** : grille à la fin du conteneur.



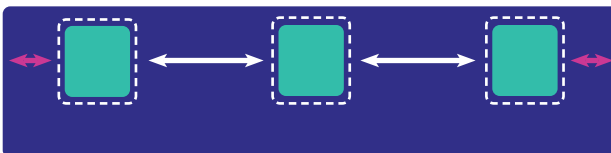
- **center** : grille au centre du conteneur.



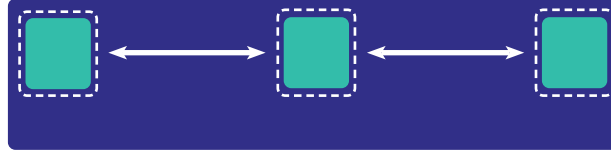
- **stretch** : redimensionne les éléments de la grille pour permettre à la grille de remplir toute la largeur du conteneur.



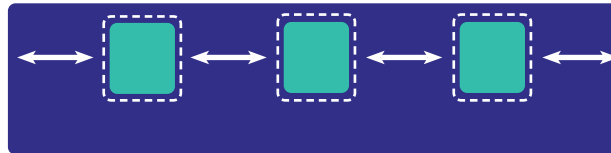
- **space-around** : espace égal entre chaque élément de la grille, avec de faibles espaces aux extrémités du conteneur.



- **space-between** : place un espace égal entre chaque élément de la grille, sans espace aux extrémités du conteneur.



- **space-evenly** : place un espace égal entre chaque élément de la grille, y compris aux extrémités du conteneur.



- **align-content** : alignement du conteneur de grille le long de l'axe vertical.
- **start** : grille au début du conteneur.



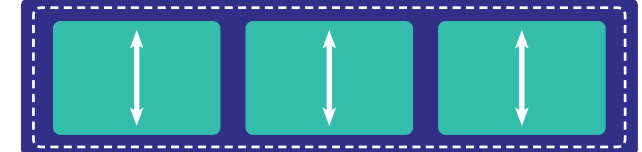
- **end** : grille à la fin du conteneur.



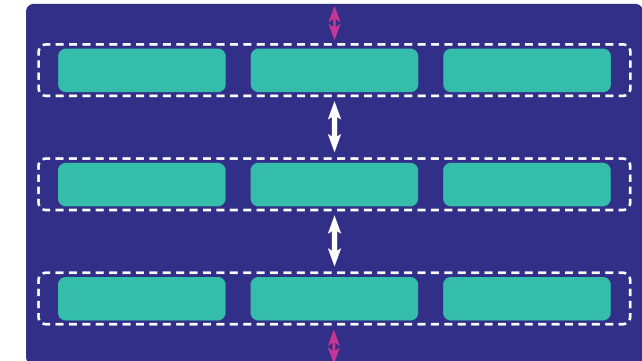
- **center** : grille au centre du conteneur.



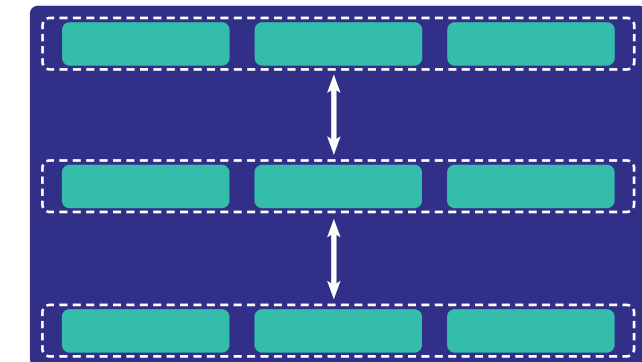
- **stretch** : redimensionne les éléments de la grille pour permettre à la grille de remplir toute la hauteur du conteneur.



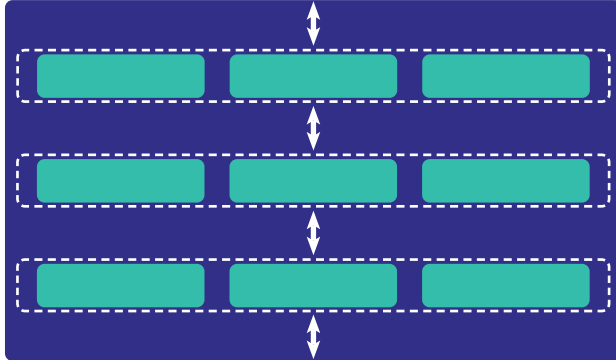
- **space-around** : place un espace égal entre chaque élément de la grille, avec de faibles espaces aux extrémités du conteneur.



- **space-between** : place un espace égal entre chaque élément de la grille, sans espace aux extrémités du conteneur.



- **space-evenly** : place un espace égal entre chaque élément de la grille, y compris aux extrémités du conteneur.



- **place-content** : raccourci pour définir à la fois **align-content** et **justify-content** en une seule déclaration.

```
.container{
  place-content: center;
}
```
- **justify-items** : alignement des éléments le long de l'axe horizontal.
 - **start** : regroupe les éléments au début de leurs cellules.



- **end** : regroupe les éléments à la fin de leurs cellules.



- **center** : aligne les éléments au centre de leurs cellules.



- **stretch** : les éléments remplissent toute la largeur de leurs cellules.



- **align-items** : alignement des éléments le long de l'axe vertical.
 - **start** : regroupe les éléments au début de leurs cellules.



- **end** : regroupe les éléments à la fin de leurs cellules.



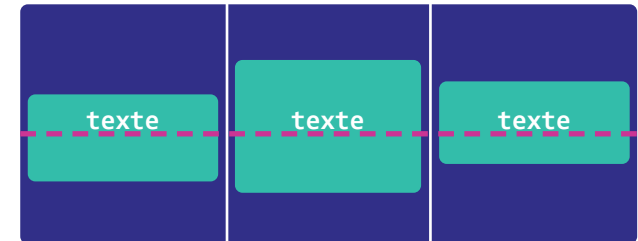
- **center** : regroupe les éléments au centre de leurs cellules.



- **stretch** : les éléments remplissent toute la hauteur de leurs cellules.



- **baseline** : aligne les éléments le long de la ligne de base du texte.



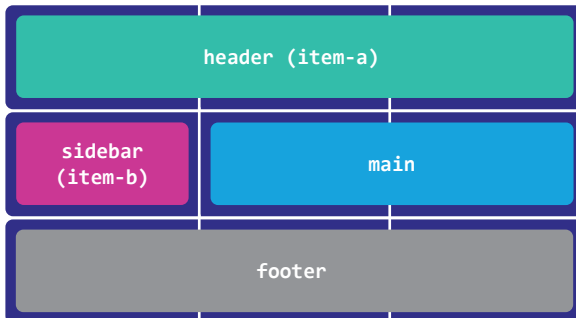
- **place-items** : raccourci pour définir à la fois **align-items** et **justify-items** en une seule déclaration.

Propriétés s'appliquant aux enfants (éléments de la grille)

- **grid-area** : pour attribuer un élément à un nom de zone.

Exemple :

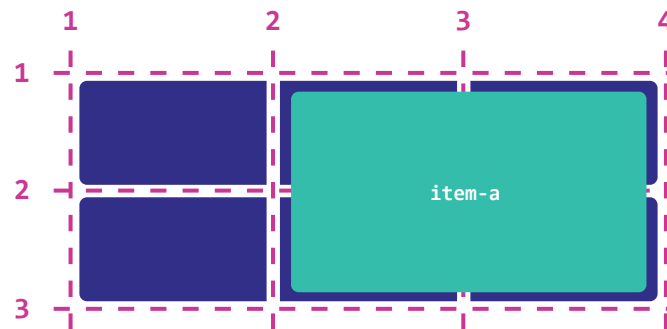
```
.container{
  grid-template-columns: 25% 75%;
  grid-template-rows: auto;
  grid-template-areas:
    "header header header"
    "sidebar main main"
    "footer footer footer";
}
.item-a {
  grid-area: header;
}
.item-b {
  grid-area: sidebar;
}
```



- **grid-column-start / grid-row-start** : définit la ligne où commence l'élément.
- **grid-column-end / grid-row-end** : définit la ligne où l'élément se termine l'élément.

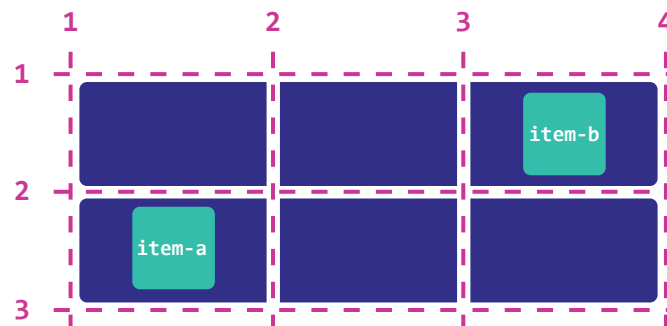
Exemple :

```
.item-a {
  grid-column-start: 2;
  grid-column-end: 4;
  grid-row-start: 1;
  grid-row-end: 3;
}
```



- **grid-column / grid-row** : à utiliser pour positionner les éléments de grille comme ceci :

```
.item-a {
  grid-column: 1 / 2;
  grid-row: 2 / 3;
}
.item-b {
  grid-column: 3 / 4;
  grid-row: 1 / 2;
}
```



- **justify-self** : alignement d'un élément à l'intérieur d'une seule et unique cellule, le long de l'axe horizontal.
 - **start** : élément au début de la cellule.
 - **end** : élément à la fin de la cellule.
 - **center** : élément au centre de la cellule .
 - **stretch** : l'élément remplit toute la largeur de la cellule.
- **align-self** : alignement d'un élément à l'intérieur d'une seule et unique cellule, le long de l'axe vertical.
 - **start** : élément au début de la cellule.
 - **end** : élément à la fin de la cellule.
 - **center** : élément au centre de la cellule.
 - **stretch** : l'élément remplit toute la hauteur de la cellule .
- **place-self** : raccourci pour définir à la fois **align-self** et **justify-self** en une seule déclaration.