

Exercice 1 : Créer une grille simple

Crée une grille contenant **6 éléments** (boîtes) répartis en **2 lignes et 3 colonnes**.

1. Utilise la propriété `grid-template-rows` pour définir deux lignes de **100px** chacune.
2. Utilise `grid-template-columns` pour définir trois colonnes de **200px** chacune.
3. Ajoute une bordure à chaque boîte pour bien voir les séparations.

Exercice 2 : Grille avec colonnes de tailles différentes

Crée une grille avec **4 colonnes** où chaque colonne a une largeur différente :

- Première colonne : 100px
- Deuxième colonne : 200px
- Troisième colonne : 1fr
- Quatrième colonne : 2fr

Ajoute au moins 8 éléments dans la grille pour remplir plusieurs lignes

Exercice 3 : Utiliser `span` pour fusionner des colonnes

Crée une grille de **4 colonnes et 3 lignes**.

1. Place 9 boîtes dans cette grille.
2. Configure les boîtes pour qu'elles utilisent `grid-column` avec des valeurs de `span` :
 - Une boîte occupe 2 colonnes.
 - Une autre boîte occupe 3 colonnes.
 - Une troisième boîte occupe 1 colonne, mais s'étend sur 2 lignes avec `grid-row: span 2`.

Exercice 4 : Grille avec des tailles dynamiques (auto, fr, et minmax)

Crée une grille de **3 lignes et 4 colonnes**, où :

- Les colonnes ont des tailles définies avec `minmax` : entre 150px (minimum) et 1fr (maximum).
- Les lignes ont des tailles dynamiques :
 - La première ligne est de 100px.
 - Les deux autres lignes utilisent `auto`.

Ajoute au moins **12 boîtes** pour remplir la grille.

Exercice 5 : Carte de visite en grille

Crée une **grille asymétrique** pour une carte de visite (layout). La carte doit inclure :

- Une image en haut qui occupe 2 colonnes.
- Un titre occupant 1 colonne.
- Un sous-titre occupant une autre colonne.
- Une description occupant 2 colonnes.

Exercice 6 : Grille responsive avec repeat et auto-fit

Crée une grille responsive où les éléments s'ajustent selon la largeur de l'écran.

1. Utilise la fonction `repeat(auto-fit, minmax(150px, 1fr))` pour définir les colonnes.
2. Ajoute 20 boîtes dans la grille pour observer le comportement.

Exercice 7 : Placement explicite avec grid-area

Crée une grille avec 6 zones et place chaque élément à une position spécifique à l'aide de `grid-area`.

- La grille doit avoir 4 colonnes et 3 lignes.
- Exemple de placement :
 - Une zone "Header" en haut qui occupe toute la première ligne.
 - Une zone "Sidebar" à gauche sur 2 lignes.
 - Une zone "Content" qui occupe 2 colonnes au centre.
 - Une zone "Footer" qui occupe toute la dernière ligne.

Exercice 8 : Modèle de site web avec Grid

Crée un modèle de site web en utilisant CSS Grid.

- Le site doit contenir :
 - Une barre de navigation en haut qui occupe **1 ligne entière**.
 - Une colonne pour le menu à gauche qui occupe **2 lignes**.
 - Une zone principale au centre pour le contenu.
 - Une colonne à droite pour une barre latérale.
 - Un pied de page en bas qui occupe **1 ligne entière**.

Utilise `grid-template-areas` pour définir la disposition.

Exercice 9 : Grille avec gap

Crée une grille avec **4 lignes** et **4 colonnes**, et ajoute des espaces (gap) :

- row-gap : 20px
- column-gap : 40px

Ajoute des couleurs de fond différentes pour chaque boîte.

Exercice 10 : Galerie photo avec Grid

Crée une galerie photo avec des images de tailles différentes.

1. La grille doit être définie avec grid-template-rows et grid-template-columns.
2. Certaines images doivent s'étendre sur **2 colonnes** et/ou **2 lignes** avec span.
3. Ajoute un gap entre les images pour améliorer la présentation.

Exercice 11 : Utiliser span pour des dispositions complexes

Crée une grille avec 5 colonnes et 3 lignes, où :

1. Une boîte occupe 2 colonnes (grid-column: span 2).
2. Une autre boîte occupe 3 colonnes (grid-column: span 3).
3. Une troisième boîte occupe 2 lignes (grid-row: span 2).
4. Une boîte occupe à la fois 2 colonnes et 2 lignes.

Exercice 12 : Auto-placement avec grid-auto-flow

Crée une grille où les éléments sont automatiquement placés en ligne ou en colonne.

1. Définit une grille de 4 colonnes.
2. Utilise la propriété grid-auto-flow pour que les éléments :
 - Se placent automatiquement **en colonne**.
 - Ajoute suffisamment d'éléments (par exemple 12) pour observer l'auto-placement.

Expérimente avec ces valeurs :

- grid-auto-flow: row;
- grid-auto-flow: column;
- grid-auto-flow: dense;

Exercice 13 : Grille auto-remplie avec des tailles minimales et maximales

Crée une grille où les colonnes s'ajustent dynamiquement avec `auto-fit` et `auto-fill`.

1. Utilise la fonction `repeat(auto-fit, minmax(100px, 1fr))` pour les colonnes.
2. Ajoute une dizaine d'éléments et observe comment ils se répartissent selon la largeur de la fenêtre.
3. Change `auto-fit` en `auto-fill` et note la différence.

Exercice 14 : Zone de contenu asymétrique avec `grid-area` et `span`

Crée une disposition complexe utilisant `grid-area`, où :

1. La grille comporte **4 colonnes** et **4 lignes**.
2. Les zones doivent inclure :
 - Un **header** occupant toute la première ligne.
 - Une **sidebar** occupant les 2 premières lignes de la première colonne.
 - Une zone **main-content** occupant les 2 premières lignes et 3 colonnes restantes.
 - Un **footer** occupant toute la dernière ligne.
3. Utilise des noms de zones (`grid-template-areas`) pour organiser la disposition.

Exercice 15: Tableau en grille

Reproduisez un tableau classique avec `display: grid`. Utilisez la fusion pour les en-têtes.

- Créez un tableau de 3 colonnes et 4 lignes.
- La première colonne doit être un en-tête fusionnant **toutes les lignes**.
- Les deux autres colonnes affichent les données.