

An abstract geometric design on the left side of the image. A diagonal line runs from the top-left corner towards the bottom-right. The area to the left of this line is divided into several rectangular sections. The top-left section is dark purple and contains a small white circle. Below it is a blue section with concentric circles. To the right of the blue section is a light gray semi-circle. Further right is a pink section with diagonal lines. Below the blue section is a pink section with a series of nested L-shaped lines. To the right of that is a blue section. Below the pink section with nested L-shapes is a bright pink section. To the right of that is a gray section with a diagonal line. Finally, to the right of the gray section is a large pink triangle. The rest of the image to the right of the diagonal line is a solid blue background.

CSS GRID

CSS : GRID

FONCTIONNEMENT

Le concept général de Grid est de diviser l'espace en zones.

Concrètement, il s'agit de découper l'espace en lignes et en colonnes comme le ferait un tableau.

Mais à la différence des tableaux, la grille est une construction schématique abstraite, elle ne nécessite aucun élément HTML.

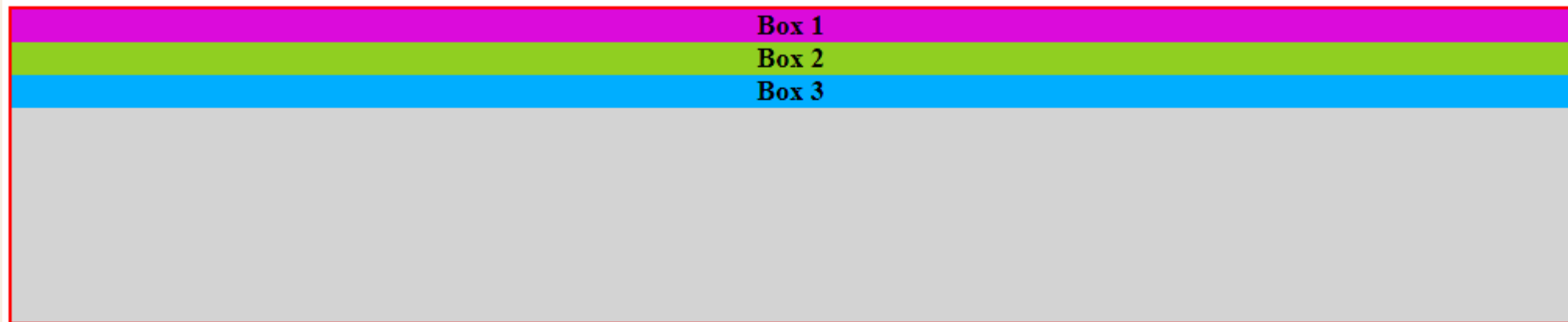
Pour des informations plus exhaustives, vous pouvez vous référer à la documentation officielle :

https://developer.mozilla.org/fr/docs/Web/CSS/CSS_grid_layout/Basic_concepts_of_grid_layout

CSS : GRID

LA DIVISION DE L'ESPACE

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

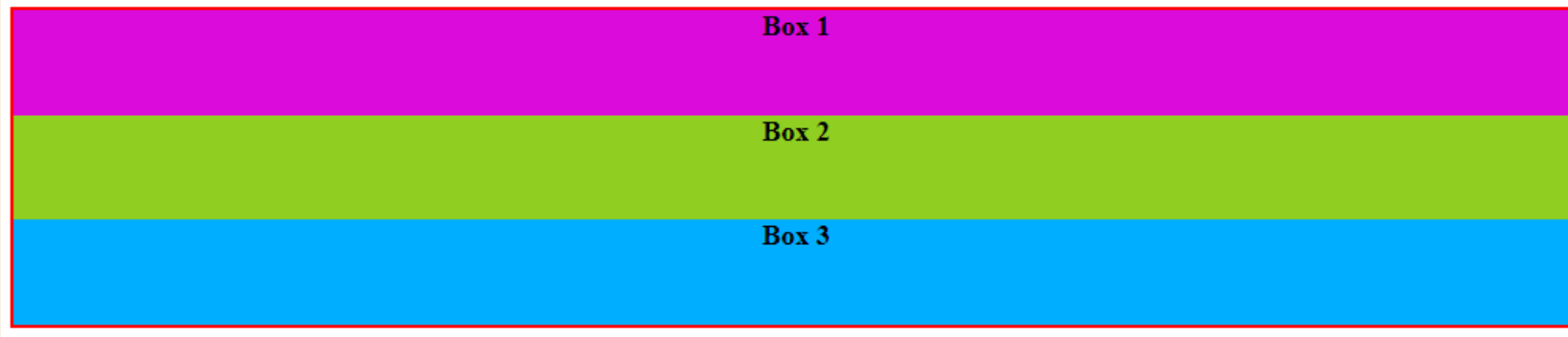


```
<body>
  <div id="container">
    <div id="box1" class="box">Box 1</div>
    <div id="box2" class="box">Box 2</div>
    <div id="box3" class="box">Box 3</div>
  </div>
</body>
```

CSS : GRID

LA DIVISION DE L'ESPACE

Nous devons initialiser notre grille dans le conteneur parent (fond gris et bordure rouge) en y appliquant la propriété « display: grid »



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

Note :

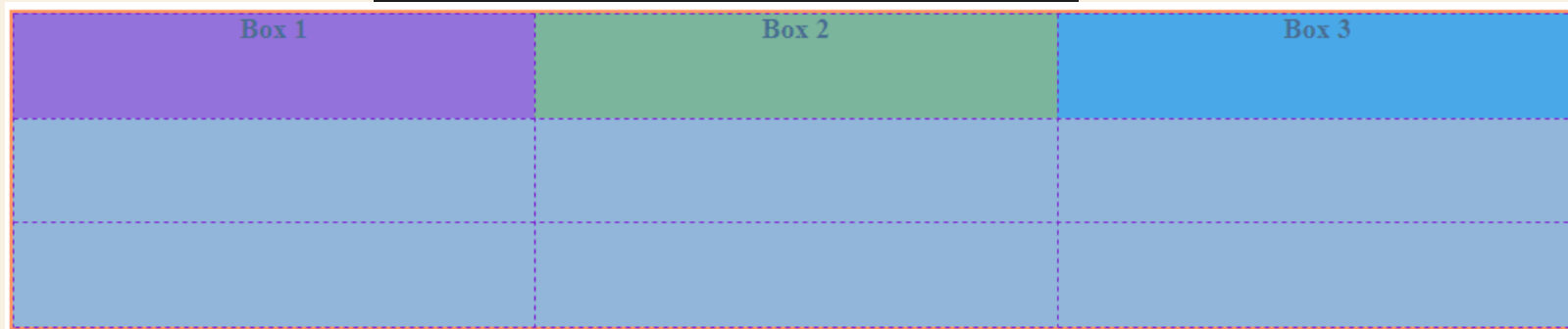
Ce conteneur parent devient maintenant le **conteneur de grille**.
Les « box » à l'intérieur elles, deviennent les **éléments de grille**.

CSS : GRID

CONTENEUR DE GRID (GRID-CONTAINER)

Grâce aux propriétés « grid-template-columns » et « grid-template-rows » nous allons pouvoir définir la segmentation de l'espace.

```
#container{  
  border: 2px solid red;  
  background-color: lightgray;  
  width: 1000px;  
  height: 200px;  
  display: grid;  
  grid-template-columns: 1fr 1fr 1fr;  
  grid-template-rows: repeat(3,1fr);  
}
```



CSS : GRID

CONTENEUR DE GRID (GRID-CONTAINER)

- **grid-template-columns** : va nous permettre de définir le nombre et la taille des colonnes de notre grid. Ainsi, on va définir la taille de chaque colonne.
Toutes les unités CSS peuvent être utilisées. Ainsi, on peut définir une taille (en px), un pourcentage de l'espace du parent.

On peut également utiliser l'unité « **fr** » : Cette unité représente une fraction de l'espace disponible dans le conteneur de la grille. Le code suivant crée trois colonnes égales qui se redimensionnent en fonction de l'espace disponible.

<https://developer.mozilla.org/fr/docs/Web/CSS/grid-template-columns>

```
<div class="wrapper">  
  <div>Un</div>  
  <div>Deux</div>  
  <div>Trois</div>  
  <div>Quatre</div>  
  <div>Cinq</div>  
</div>
```

CSS

```
.wrapper {  
  display: grid;  
  grid-template-columns: 1fr 1fr 1fr;  
}
```

CSS : GRID

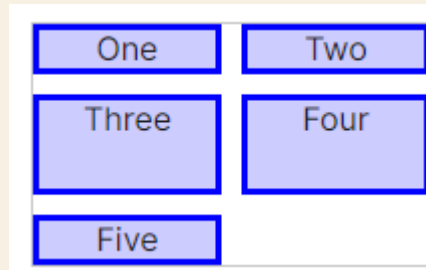
CONTENEUR DE GRID (GRID-CONTAINER)

- **grid-template-rows** : De la même façon, cette propriété va nous permettre de définir le nombre et la taille des lignes de notre grid.

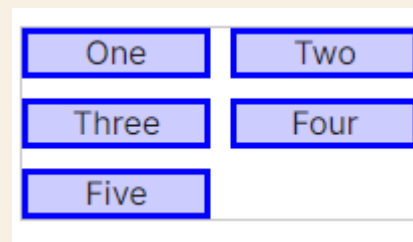
Pour des exemples interactifs et des détails :

<https://developer.mozilla.org/fr/docs/Web/CSS/grid-template-rows>

```
grid-template-rows: 1fr 2fr 1fr;
```



```
grid-template-rows: auto;
```



CSS : GRID

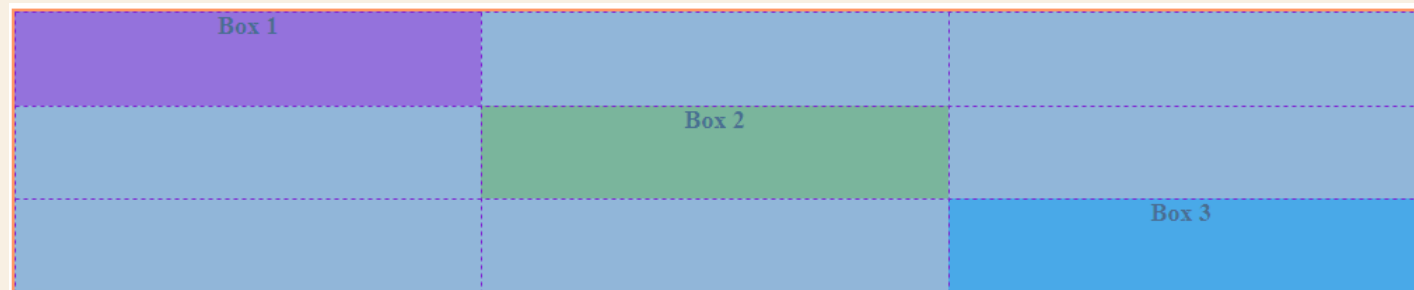
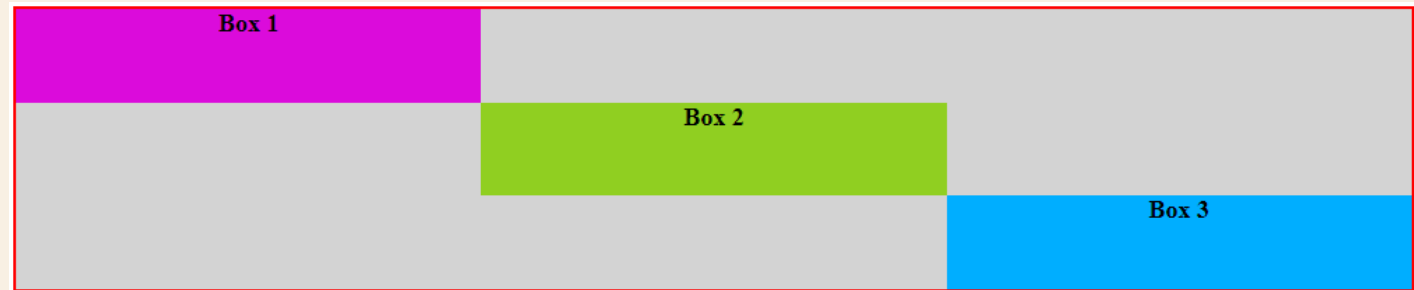
ÉLÉMENTS DE GRID (GRID-ITEMS)

Grâce aux propriétés « grid-column » et « grid-row » nous allons pouvoir placer chaque élément où l'on souhaite.

```
#box1{
  background-color: ■rgb(219, 11, 219);
  grid-column: 1;
  grid-row: 1;
}

#box2{
  background-color: ■rgb(144, 207, 33);
  grid-column: 2;
  grid-row: 2;
}

#box3{
  background-color: ■rgb(0, 174, 255);
  grid-column: 3;
  grid-row: 3;
}
```



CSS : GRID

UNITÉS DE MESURE

Unités	Description
px, em, rem, %,vh,	Les unités classiques absolues et relatives
fr	Fraction de l'espace restant
max-content	Taille de l'élément le plus grand
min-content	Taille de l'élément le plus petit
minmax(min, max)	Taille avec un minimum et un maximum
auto	S'adapte automatiquement au contenu

Note : il existe aussi la fonctionnalité « **repeat**(nombre de fois, unité) » qui permet de créer un motif de répétition de colonnes ou de lignes. (appelés « patterns »)
Ex: « repeat(3, 1fr) » répète 1 fraction d'espace (1fr) 3 fois

CSS : GRID

CENTRAGE

Propriété	Description
justify-items	Aligne les éléments de l'axe principal dans leurs cellules (container)
align-items	Aligne les éléments de l'axe secondaire dans leurs cellules (container)
justify-self	Aligne un grid-item dans sa cellule (axe principal)
align-self	Aligne un grid-item dans sa cellule (axe secondaire)
Valeur	Description
start	Alignement au début de la cellule
end	Alignement à la fin de la cellule
center	Alignement au centre de la cellule
stretch	Etire l'élément pour remplir l'espace de la cellule. Par défaut

CSS : GRID

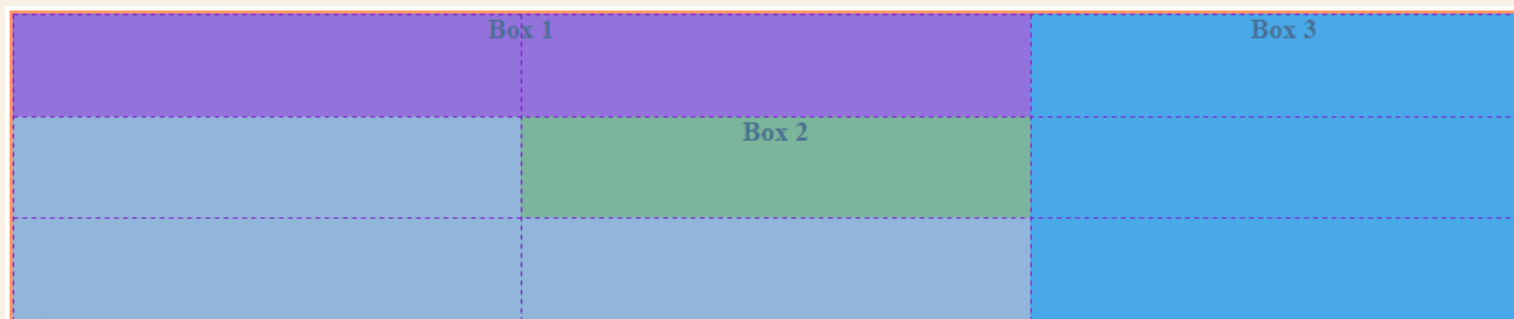
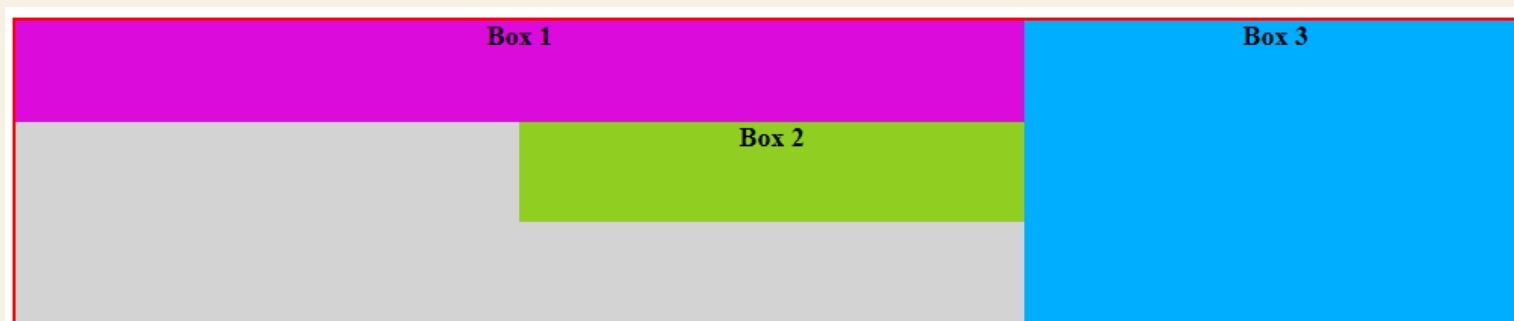
EMPLACEMENTS MULTIPLES / FUSION

Nous pouvons aussi utiliser la valeur « span » pour occuper plusieurs emplacements (lignes ou colonnes) avec nos éléments :

```
#box1{
  background-color: magenta;
  grid-column: 1 /span 2;
  grid-row: 1;
}

#box2{
  background-color: limegreen;
  grid-column: 2;
  grid-row: 2;
}

#box3{
  background-color: cyan;
  grid-column: 3;
  grid-row: 1 /span 3;
}
```



CSS : GRID

POUR ALLER PLUS LOIN

- La documentation: <https://developer.mozilla.org/en-US/docs/Web/CSS/grid>
- Un site explicatif sur les propriétés des grid:
<https://css-tricks.com/snippets/css/complete-guide-grid/>
- Un petit jeu : CSS Grid Garden: <https://cssgridgarden.com/#fr>

CSS : GRID

MISE EN PRATIQUE (30MN)

1. Dans un nouveau dossier, créez un fichier HTML « gridSite.html » et un fichier CSS « gridSite.css »
2. Remplissez le fichier « gridSite.html » avec le HTML minimum et liez la feuille de style « gridSite.css » avec
3. Créez dans le <body> :
 - a) Un titre <h1>
 - b) Un menu <nav> qui contiendra :
 - i. Une liste à puce non numérotée
 - ii. Cinq lignes
 - c) Trois articles, chacun dans une <div>, qui contiendront :
 - i. Un titre <h2>
 - ii. Un paragraphe <p> avec du faux texte dedans (lorem ipsum) sur plusieurs lignes
 - d) Une balise <footer> qui sera notre pied de page et qui contiendra un titre <h3>

Utilisez la grille pour mettre les éléments en place selon le visuel de la page suivante.

CSS : GRID

MISE EN PRATIQUE (30MN)

Je suis le titre principal

• Menu1 • Menu2 • Menu3 • Menu4 • Menu5

Titre article

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent sit amet nunc sed lacus ornare feugiat. Phasellus vulputate, quam eget egestas euismod, lacus nisl suscipit mauris, eget malesuada quam dui id erat. Quisque lectus massa, a consectetur ipsum fringilla quis. Integer tempus eleifend est sed consequat. Nam suscipit ullamcorper urna, efficitur malesuada tortor sollicitudin sed. Donec vel hendrerit ipsum, faucibus luctus dui. Quisque ultricies sagittis odio, eget feugiat diam auctor auctor. Suspendisse varius nisl vitae libero lacinia, placerat bibendum elit pharetra. Duis ut volutpat urna. Maecenas ut eros elit. Donec varius blandit erat. Vestibulum vitae facilisis dolor, sed consequat diam.

Titre article

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent sit amet nunc sed lacus ornare feugiat. Phasellus vulputate, quam eget egestas euismod, lacus nisl suscipit mauris, eget malesuada quam dui id erat. Quisque lectus massa, semper quis leo ut, fermentum interdum erat. Sed facilisis non sem pharetra posuere. Sed interdum eget tellus sit amet accumsan. Mauris aliquam neque massa, a consectetur ipsum fringilla quis. Integer tempus eleifend est sed consequat. Nam suscipit ullamcorper urna, efficitur malesuada tortor sollicitudin sed. Donec vel hendrerit ipsum, faucibus luctus dui. Quisque ultricies sagittis odio, eget feugiat diam auctor auctor. Suspendisse varius nisl vitae libero lacinia, placerat bibendum elit pharetra. Duis ut volutpat urna. Maecenas ut eros elit. Donec varius blandit erat. Vestibulum vitae facilisis dolor, sed consequat diam.

Titre article

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent sit amet nunc sed lacus ornare feugiat. Phasellus vulputate, quam eget egestas euismod, lacus nisl suscipit mauris, eget malesuada quam dui id erat. Quisque lectus massa, semper quis leo ut, fermentum interdum erat. Sed facilisis non sem pharetra posuere. Sed interdum eget tellus sit amet accumsan. Mauris aliquam neque massa, a consectetur ipsum fringilla quis. Integer tempus eleifend est sed consequat. Nam suscipit ullamcorper urna, efficitur malesuada tortor sollicitudin sed. Donec vel hendrerit ipsum, faucibus luctus dui. Quisque ultricies sagittis odio, eget feugiat diam auctor auctor. Suspendisse varius nisl vitae libero lacinia, placerat bibendum elit pharetra. Duis ut volutpat urna. Maecenas ut eros elit. Donec varius blandit erat. Vestibulum vitae facilisis dolor, sed consequat diam.