

Tema 4: Grid Layout en CSS (I)

El diseño de página con Grid Layout es una técnica avanzada en CSS que permite crear diseños de página complejos y flexibles. Grid Layout organiza elementos en filas y columnas, lo que facilita la creación de diseños estructurados y responsivos.

En primer lugar, se necesita definir un contenedor que será el elemento padre que contendrá los elementos de la cuadrícula.

```
.contenedor {  
    display: grid;  
    grid-template-columns: 1fr 1fr 1fr; /* Definir 3 columnas de tamaño igual */  
    grid-template-rows: 100px 200px; /* Definir 2 filas con alturas específicas */  
}
```

- **display: grid;** Esta propiedad de CSS establece que el elemento con la clase `.contenedor` debe utilizar un sistema de cuadrícula (grid) para organizar sus elementos secundarios. Esto significa que los elementos hijos pueden ser colocados en filas y columnas.
- **grid-template-columns: 1fr 1fr 1fr;** Esto establece la distribución de columnas en la cuadrícula. En este caso, se están definiendo tres columnas (1fr significa "1 fracción" o "un espacio igual"). Esto significa que las tres columnas tendrán el mismo ancho.
- **grid-template-rows: 100px 200px;** Aquí se definen las alturas de las filas. La primera fila tendrá una altura de 100px y la segunda fila tendrá una altura de 200px.

La cuadrícula tendrá tres columnas de igual ancho y dos filas con alturas específicas (100px y 200px respectivamente). Esto permitirá colocar elementos secundarios en esta cuadrícula de manera organizada y controlada.

A continuación, se muestran las características principales:

- **display:** especifica que el elemento es un contenedor de cuadrícula y establecer el contexto de formato de su contenido.
 - Valores: `grid`, `inline-grid`.

```
.contenedor {  
    display: grid;  
}
```
- **grid-template-columns:** define el número y tamaño de las columnas en una cuadrícula.
 - Valores: (px, %, fr)

```
.contenedor {  
    grid-template-columns: 100px 1fr 2fr;  
}
```

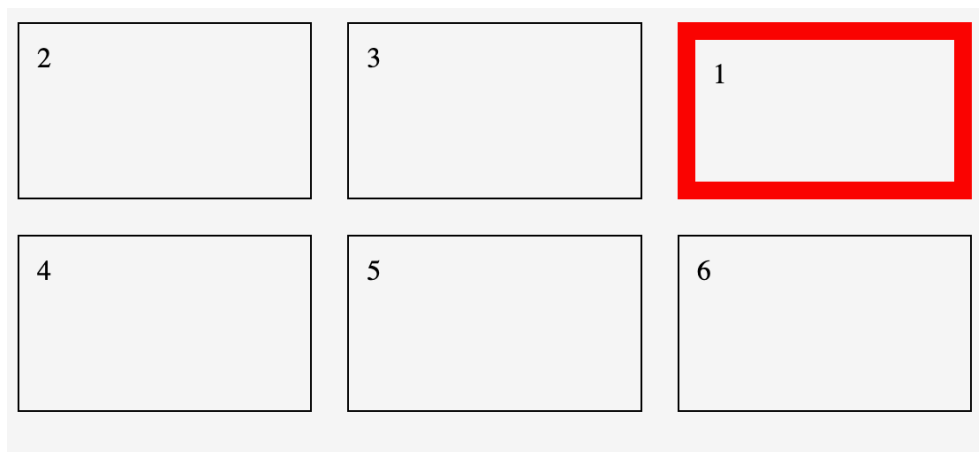
- **grid-template-rows:** definir el número y tamaño de las filas en una cuadrícula.
 - Valores:(px, %, fr)


```
.contenedor {
    grid-template-rows: 100px 200px;
}
```
- **grid-gap:** establecer el espacio entre las filas y columnas de una cuadrícula.
 - **Valores:** Puedes usar valores en píxeles, porcentajes o em.


```
.contenedor {
    grid-gap: 10px;
}
```
- **grid-column y grid-row:** especificar en qué columnas y filas debe estar ubicado un elemento.
 - **Valores:** Puedes usar números para indicar la línea, span para extenderse varias líneas, y más.


```
.elemento {
    grid-column: 2 / span 2;
    grid-row: 1 / 3;
}
```

Ejercicio: Construye la siguiente imagen utilizando los elementos del grid descritos hasta ahora.



- **grid-template-areas:** definir áreas nombradas en la cuadrícula, facilitando la colocación de elementos.
 - **Valores:** Strings que representan el nombre de las áreas.


```
contenedor {
    grid-template-areas: "cabecera cabecera" "sidebar contenido" "pie pie";
}
```

Ejercicio: Construye la siguiente imagen utilizando los elementos del grid descritos hasta ahora.



- **justify-items y align-items:** alinear elementos dentro de las celdas de la cuadrícula.
 - **Valores:** start, end, center, stretch.

```
.contenedor {
  justify-items: center;
  align-items: center;
}
```
- **justify-content y align-content:** alinear el contenido de la cuadrícula dentro de su contenedor.
 - **justify-content: alinear el contenido de la cuadrícula en el eje horizontal.**
 - start: El contenido se alinea al principio del contenedor.
 - end: El contenido se alinea al final del contenedor.
 - center: El contenido se alinea en el centro del contenedor.
 - space-between: El espacio entre los elementos se distribuye de manera uniforme.
 - space-around: El espacio alrededor de los elementos se distribuye de manera uniforme.
 - **align-content: alinear el contenido de la cuadrícula en el eje vertical.**
 - start: El contenido se alinea en la parte superior del contenedor.
 - end: El contenido se alinea en la parte inferior del contenedor.
 - center: El contenido se alinea en el centro del contenedor.
 - stretch: El contenido se estira para llenar el contenedor verticalmente.
 - space-between: El espacio entre los elementos se distribuye de manera uniforme.
 - space-around: El espacio alrededor de los elementos se distribuye de manera uniforme.

```
contenedor {  
    display: grid;  
    grid-template-columns: 100px 100px 100px;  
    grid-template-rows: 50px 50px 50px;  
    align-content: center;  
}
```

Ejercicio:

