

GRID

Es una herramienta que nos permite trabajar con layouts(div, elementos de tipo contenedor, que pueden ser etiquetas semánticas), para organizar el contenido como si fuera una rejilla.

Como rejilla, tendremos elementos de tipo fila y columna.

Necesitamos elementos contenedores o layouts, para distribuir los elementos de la web.

Conceptos básicos:

1.Grid container, elemento padre, que contiene toda la rejilla. Este elemento puede ser un elemento de tipo caja, un div, etiqueta semántica.. Este elemento tiene la propiedad `display:grid;`

2.Grid ítems. Los elementos hijos de este contenedor, que a su vez tendrán filas y columnas.

3.Filas y columnas, dentro de nuestro grid. Para definir las columnas utilizamos la propiedad:

-Dependiendo del número de columnas, los valores se separan por comas

`grid-template-columns: anchura de la columna,anchura de la columna,anchura de la columna;`

`grid-template-columns:200px,200px,200px,200px;`

-Dependiendo de número de filas, los valores se separan por comas.

`grid-template-rows:altura de las filas,altura de las filas,altura de las filas;`

`grid-template-rows:150px,150px,150px,150px;`

El ajuste del display grid, las columnas y las filas, como es la configuración inicial, se definen en el elemento contenedor:

```
.contenedor
{
  display:grid;
  grid-template-columns:250px,250px,250px;
  grid-template-rows:100px,100px,100px;
}
```

USAR FR PARA TAMAÑOS FLEXIBLES

Unidad de medida exclusiva de Grid. FR representa una fracción del espacio disponible.

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

`grid-template-columns:1fr 2fr`, significa que la primera columna ocupará una fracción del ancho del contenedor, y la segunda columna ocupará el doble.

```
grid-template-columns:100px 100px 300px;
```

```
grid-template-columns:1fr 1fr 3fr;
```

GAP

Con esta propiedad podemos agregar espacio entre filas y columnas. Gap viene a ser, como un margen, entre cada elemento.

```
.contenedor
{
  display:grid;
  grid-template-columns:1fr 2fr;
  grid-template-rows:100px,100px;
  gap:20px;
}
```

POSICIONAR ELEMENTOS

Utilizando la propiedad grid-column y grid-row, este posicionamiento es relación al número de filas o columnas que ocupa.(PERMITEN COMBINAR CELDAS Y COLUMNAS)

```
.contenedor
{
  display:grid;
  grid-template-columns:1fr 1fr 1fr 2fr;
  grid-template-rows:100px,100px,100px,100px;
  gap:20px;
}
```

```
.elemento
{
  grid-column:1/4;
  grid-row:2/2; }
```

AGREGAR VARIOS ELEMENTOS A LA VEZ

Añadimos tres columnas con el mismo tamaño a nuestra rejilla:

```
.contenedor
{
  display:grid;
  grid-template-columns:repeat(3,200px);
  grid-template-rows:repeat(4,150px);
  gap:20px;
}
```

En la función repeat, podemos especificar el número de columnas que tenemos que añadir, y el tamaño que tendrá cada columna.

OTRAS PROPIEDADES IMPORTANTES

justify-items: Nos permite alinear los elementos horizontalmente dentro de su celda en la rejilla.

align-items: Alineación de los elementos verticalmente.

grid-auto-rows y **grid-auto-columns.** Definen filas/columnas automáticas para elementos adicionales.



Centro Oficial FP
Digital & Tech

PODEMOS COMBINAR, GRID CON MEDIAQUERIE:

```
@media screen and(min-width:250px)
{
    .contenedor
    {
        display:grid;
        grid-template-columns:repeat(3,200px);
        grid-template-rows:repeat(4,150px);

    }
}
```



Centro Oficial FP
Digital & Tech