

GRID CSS

Es un sistema para creación de Layouts del estilo de tablas. Se compone de filas y columnas, y se pueden agrupar unos dentro de otros.

- Su ventaja respecto a Flex es que podemos crear layouts complejos con pocas líneas de código.
- No es opuesto a Flex ni a Float, se pueden combinar entre ellos.
- Se utilizan propiedades para indicar cuantas filas y columnas lo componen indicando el tamaño de cada una de ellas.

```
/* Sistema de 3x3 */  
.container{  
  display:grid;  
  /* Cantidad de columnas*/  
  grid-template-columns:100px 100px 100px;  
  /* Cantidad de filas*/  
  grid-template-rows: 100px 100px 50px;  
}
```

Podemos indicar a cada item donde ponerlo en la tabla con las reglas: `grid-row-start`, `grid-row-end`, `grid-column-start` y `grid-column-end` (nos va a permitir decidir donde poner cada elemento).

- Tener en cuenta que se cuenta desde 1 no de 0.

Si queremos que un item ocupe 3 columnas y 1 fila, deberíamos de indicar que comience en la columna 1 y termine en la columna 4. Que comience en la fila 1 y termine en la fila 2.

```
/* tenemos un container de 3 filas y 4 columnas */  
/* el item-1 va a ir desde la columna 1 a la 4 (3 columnas) */  
/* el item-1 va a ir desde la fila 1 a la fila 1 (1 fila) */  
.container{  
  display:grid;  
  grid-template-columns:100px 100px 100px 100px;  
  grid-template-rows: 100px 100px 100px;  
}  
.item-1 {  
  grid-column-start: 1;  
  grid-column-end: 4;  
  grid-row-start: 1;  
  grid-row-end: 2;  
}
```

Se pueden invertir el start y el end:

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

Existen los números negativos de filas y columnas negativas para seleccionar elementos desde el final del grid.

```
/* la caja está ocupando todo el grid */
.item-1 {
  grid-column-start: 1;
  grid-column-end: -1;
  grid-row-start: 1;
  grid-row-end: -1;
}
```

En vez de seleccionar inicio y fin de fila/columna, podemos usar **span** para indicar cuanto se expande nuestro item.

```
.item-1{
  grid-column-start:4;
  grid-column-end:span 2; /* se amplia uno mas (total dos bloques) */
  grid-row-start:1;
  grid-row-end:span 3; /*En total va tener 3*/
}
```

Se puede usar la regla reducida **span** para expandir el objeto.

```
.item-1{
  /* Bloque de 2x2 */
  grid-column: span2;
  grid-row: span 2;
}
```

Se puede simplificar la regla grid-column-start-end en una sola usando el símbolo "/".

```
.item-1{
  grid-column:1 /4;
  grid-row: 1/4;
```

```
}
```

Se puede simplificar más usando `grid-area`:

```
.item-1{  
  /* grid-row-start grid-col-start grid-row-end grid-col-end */  
  grid-area: 1/2/3/6;  
}
```

Podemos usar `grid-gap`(alias `gap`) para definir un espacio entre filas y columnas.

```
.container {  
  display:grid;  
  ...  
  grid-gap: 10px; /* mismo valor para filas y columnas */  
  grid-gap: 1em 3em;  
}
```

Si se agrega mas bloques dentro de nuestro grid que cantidad de filas y columnas. grid creara automaticamente ua nueva fila. Esto se llama "grid-implicita".

- Podemos usar `grid-auto-rows:100px` para definir el tamaño dde nuevas celdas.
- Podemos usar `grid-auto-flow` que nos permite definir si las nuevas celdas serán filas o columnas.

```
/* .container {  
  display: grid;  
  grid-template-rows: 100px ;  
  grid-template-columns: 100px 100px 100px;  
  
  grid-auto-rows: 100px;  
} */  
  
.container{  
  display:grid;  
  grid-template-columns:100px 100px 100px 100px;  
  grid-template-rows: 100px 100px 100px;  
  
  grid-auto-rows:100px;  
  
  grid-auto-flow:column;  
  grid-auto-columns:100px;  
}
```

En grid poseemos una nueva medida llamada fracción "fr".

```
/* sistema de 3 x 3 */
.container {
  display: grid;
  grid-template-columns: 1fr 2fr 1fr; /* 25% 50% 25%*/

  grid-template-columns: 200px 1fr; /* 200px 1fr (200px a la 1a columna y 1fr a
la segunda columna para que ocupe el espacio que quiera) */

  grid-template-columns: 200px 1fr 1fr; /* 3 columnas: 200px 50% 50% (se
divide la */
  /* dividimos en fracciones en base al total de elementos que tengamos */
}
```

Grid Templates Areas

Podemos nombrar celdas utilizando `grid-template-areas`

```
/* Container de 3x4*/
.container {
  display: grid;
  grid-template-columns: repeat(3, 1fr); /* 3 columnas x 1 fraccion*/
  grid-template-rows: repeat(4, 100px);
  grid-gap: 10px;

  /*cuando se desea dejar un espacio en blanco se coloca . */
  /*Los formatos en diagonal o en L no se pueden hacer */
  grid-template-areas:
    "sector1 sector1 sector2"
    "sector3 sector3 sector2"
    "sector4 sector5 sector5"
    "sector4 sector5 sector5";
}

.caja-1 {
  grid-area: sector1;
}

.caja-2 {
  grid-area: sector2;
}

.caja-3 {
  grid-area: sector3;
}

.caja-4 {
```

```
    grid-area: sector4;  
}  
  
.caja-5 {  
    grid-area: sector5;  
}
```