

## UNIDAD 5 CSS grid avanzado

### **grid-template-areas:** ... ;

De manera complementaria a las propiedades de **grid-template**, podemos declarar **grid-template-areas** para dotar a áreas concretas de un *grid* con nombres clave. Podemos indicar cuántas columnas o filas ocupa una área repitiendo su nombre varias veces (una por cada columna o fila), y podemos separar cada fila agrupándolas entre comillas. El resultado es bastante visual, lo que ayuda a entender mejor la disposición de las áreas. Además, si dentro de nuestro *grid* no queremos nombrar un espacio concreto como área, podemos usar un punto (".") para indicar esto y se mostrará como una celda más.

Ejemplo:

```
grid-template-areas:  
    "header header header header"  
    "main main . sidebar"  
    "footer footer footer footer";
```

### **grid-area:** ... ;

Propiedad que permite identificar un elemento con una área, y haciendo que el mismo se distribuya conforme a lo designado en la propiedad **grid-template-areas**.

### **order:** ...;

Propiedad equivalente al **order** de *flexbox*, y que permite cambiar el orden en el que aparece un elemento en un espacio *grid*. Por defecto todos los elementos tienen un **order: 0**. Los valores negativos harán que aparezca antes que los elementos cuyo **order** no se haya alterado, y los valores positivos, después.

### **Reglas justify-...**

Las reglas **justify** se aplican al eje x, y alinean el contenido dentro de un espacio *grid*.

### **justify-content: ...;**

Esta propiedad alinea las columnas de un *grid* cuando sobra espacio en dicho *grid*. Los valores que se pueden dar son: **center** (centrado), **start** (al inicio del espacio en el eje x), **end** (al final del espacio), **stretch** (las columnas crecen hasta ocupar todo el espacio), **space-around** (el espacio sobrante se reparte al principio y al final de cada columna), **space-between** (el espacio se reparte proporcionalmente entre las columnas) y **space-evenly** (el espacio se reparte como **space-around**, pero manteniendo la misma proporción entre columnas que en los exteriores).

### **justify-items: ...;**

Propiedad que alinea el contenido en el eje x en el interior de cada celda de la retícula. Los valores que se pueden dar son: **center** (centrado), **stretch** (ocupa el 100% de la celda), **start** (inicio de la celda), **end** (final de la celda).

### **justify-self: ...;**

Propiedad que alinea un elemento concreto dentro de su celda en el eje x, sin afectar al resto de elementos de la fila. Los valores que se pueden dar son: **center** (centrado), **stretch** (ocupa el 100% de la celda), **start** (inicio de la celda), **end** (final de la celda).

### **Reglas align-...**

Las reglas align se aplican al eje y, y alinean el contenido dentro de un espacio *grid*.

### **align-content: ...;**

Reglas de alineación equivalentes a las de **justify-content**, pero en el eje y. Los valores que se pueden dar son los mismos.

### **align-items: ...;**

Reglas de alineación equivalentes a las de **justify-items**, pero en el eje y. Los valores que se pueden dar son los mismos.

### **align-self: ...;**

Reglas de alineación equivalentes a las de **justify-self**, pero en el eje y. Los valores que se pueden dar son los mismos.

### **minmax(0, 1fr)**

En retículas de anchura flexible, un valor mínimo de **0** asegura que el contenido no modifique las dimensiones del *track* (y si es el caso, lo sobrepase).

## Sistema de retícula anidada

La propiedad **subgrid**, que permitiría pasar unas reglas de alineación de *grid* a elementos anidados, no está implementada aún en los navegadores. Eso significa que para mantener una retícula general consistente necesitamos replicarla en elementos anidados. Una forma de hacerlo con cierta consistencia es guardar los valores de **grid-template** en una variable para poder replicarlos fácilmente. Otra forma podría ser usar el valor **inherit**, es decir, hereda lo que tenga el elemento padre. Por ejemplo, en el caso de que el elemento anidado tenga la misma anchura que el padre, podríamos asignarle al elemento hijo estas propiedades:

```
display: grid;  
grid-template-columns: inherit;
```