

## CONCEPTOS BÁSICOS

### Grid

Es un estilo de Layout y se pone como valor de la propiedad display.

Se trabaja a través de rejillas.



### Grid container

Elemento que contiene la propiedad **display:grid**

### Grid item

Hijos **directos** del elemento grid container



```
<div class="grid-container">
  <div class="grid-item">
    <h1>Yo no soy grid item</h1>
  </div>
  <div class="grid-item">
    <h2>Yo tampoco</h2>
  </div>
  <h3 class="grid-item">Yo sí</h3>
</div>
```

### Grid cell

Intersección entre filas y columnas

1	2
3	4
5	6

### Grid tracks

Filas y columnas



### Grid lines

Lineas en horizontal y vertical

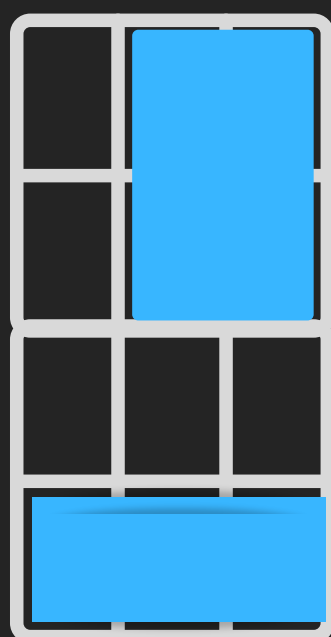


1 2 3

1 2 3 4

### Grid area

Elementos consecutivos, nosotros las definimos



## PROPIEDADES

### Se aplican al grid container:

Definimos cuantas filas y cuantas columnas y de que ancho

**Grid-template-column**

**Grid-template-row**

**Grid-template-column: 150px 150px 1fr;**

1fr = espacio restante

Espacio de margin entre grid cells

**Grid-gap**

**Grid-column-gap**

**Grid-row-gap**



**Grid-gap: 30px;**

### Se aplican al grid item:

¿Cuántas grid cells va a ocupar cada grid item?

Lo definimos usando grid lines

**Grid-row-start**

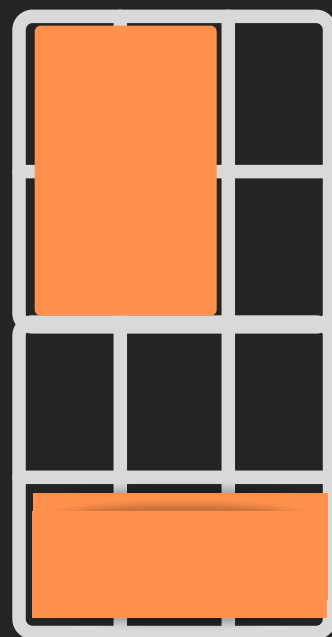
**Grid-row-end**

**Grid-column-start**

**Grid-column-end**

**Grid-row**

**Grid-column**



**Grid-row: 1 / 3    Grid-row: 1 / span 3**

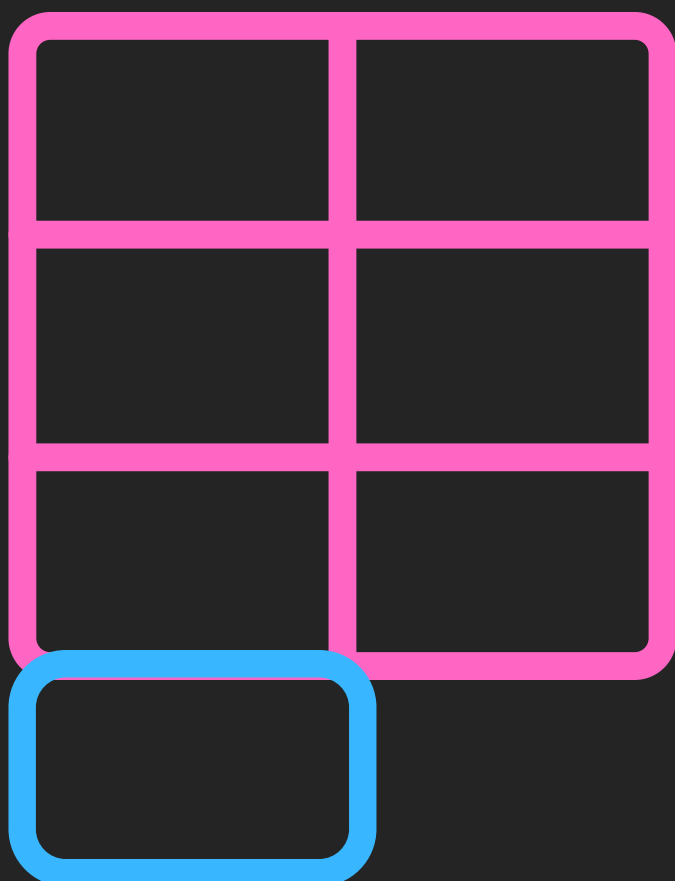
### Explícito

Es el que definimos a través de:  
`grid-template-row` y  
`grid-template-column`

### Implícito

Son los grid-items que quedan fuera del grid que definimos

```
grid-template-columns: repeat(3,150px);  
grid-template-rows: repeat(2,150px);
```



Para el grid implícito tenemos otras 2 propiedades:

1. Define el ancho de las filas que "sobran"

**Grid-auto-rows:** 20px;

**Grid-auto-columns:** 3rem;

2. ¿A dónde va lo que sobra?

**Grid-auto-flow:**

"Dense" rellena los espacios vacíos

row  
column  
dense

## GRID DINÁMICO

Podemos tener filas y columnas auto-ajustables

¿De qué ancho?

**min-content**

**max-content**

**minmax(mínimo,máximo)**

```
grid-template-rows: repeat(3, min-content);  
grid-template-rows: repeat(3, max-content);  
grid-template-rows: repeat(3,  
                             minmax(min-content, 1fr));
```

¿Cuántas veces?



**Auto-fill**

Sigue metiendo  
mientras quepan

**Auto-fit**

Va ajustando el  
tamaño

```
grid-template-rows: repeat(auto-fill,  
                             min-content);
```

# DISEÑO WEB INTERMEDIO

# GRID



## ALINEACIÓN Y CONTROL DE FLUJO

### Valores:

Start | End | Center | Stretch

**Justify:** Eje horizontal

**Align:** Eje horizontal

**Place-items:** Atajo

Alinea los items a lo largo de los ejes

**Align-items:**

**Justify-items:**

**Place-items: align justify**

Alinea el contenido cuando hay espacio de sobra en los ejes

**Align-content:**

**Justify-content:**

**Place-content: align justify**

Se aplican a los grid items y se modifican a ellos mismos

**Align-self:**

**Justify-self:**

**Place-content: align justify**

## GRID-AREAS

Define la estructura haciendo referencia a los nombres de las areas especificadas

La sintaxis tal cual, nos permite visualizar la estructura del grid

### 1. Definir la estructura

```
.container{  
  grid-template-areas:  
    " header header"  
    "aside main "  
    " . main "  
    "none main"  
    " footer footer ";  
}
```

\* El punto significa una columna vacía y "none" es un área indefinida

### 1. Definir nombres de area a los elementos

```
.item-a {  
  grid-area: header;  
}  
  
.item-b {  
  grid-area: aside;  
}
```