



# Grid CSS

<https://cssgridgarden.com/#es>

## Programación Web I

### Comisión Miércoles Noche

Dys. Alicia Rosenthal

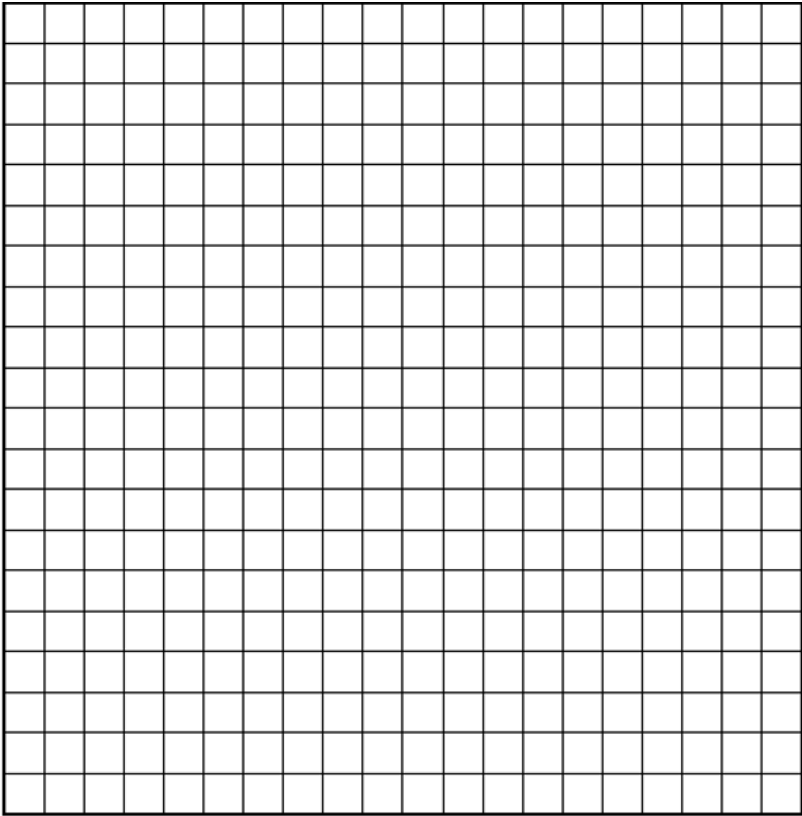
Ing. Damian Spizzirri

### Comisión Viernes Mañana

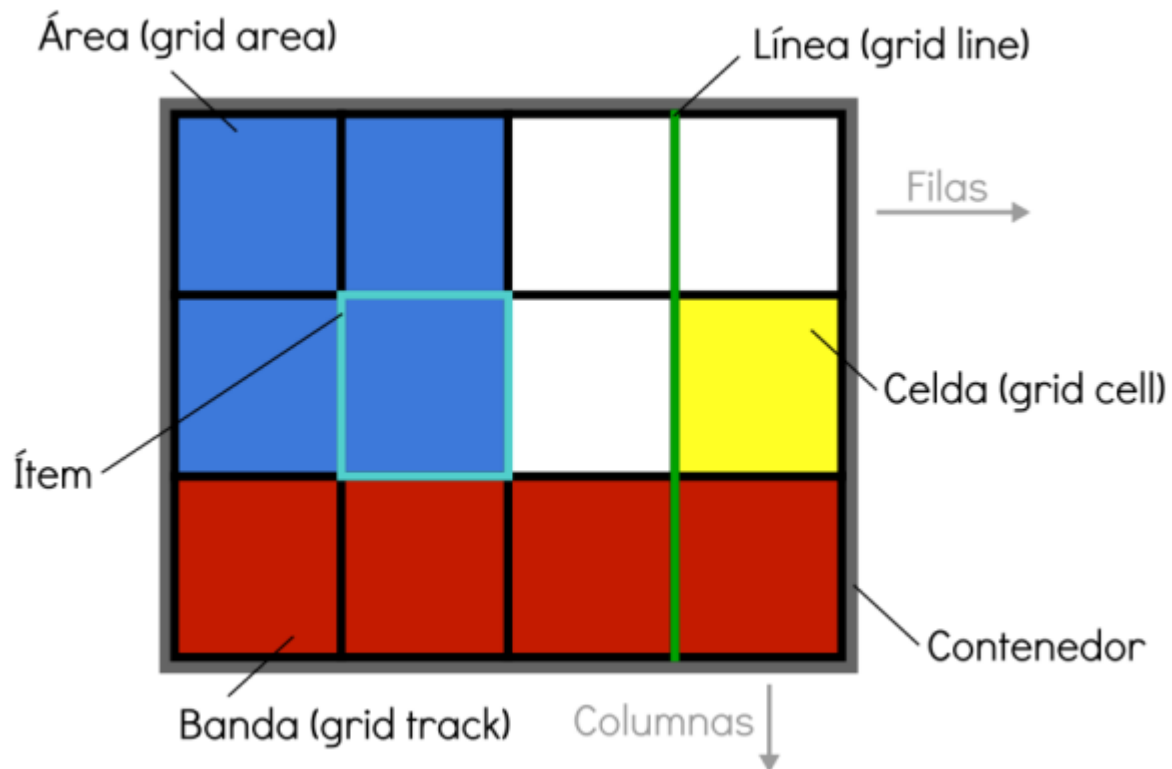
Dys. Alicia Rosenthal

Ing. Gabriel Panik

# Grid - ¿Qué es una grilla o cuadrícula?



# Conceptos básicos



## Línea (grid line):

Separador horizontal o vertical de las celdas de la cuadrícula.

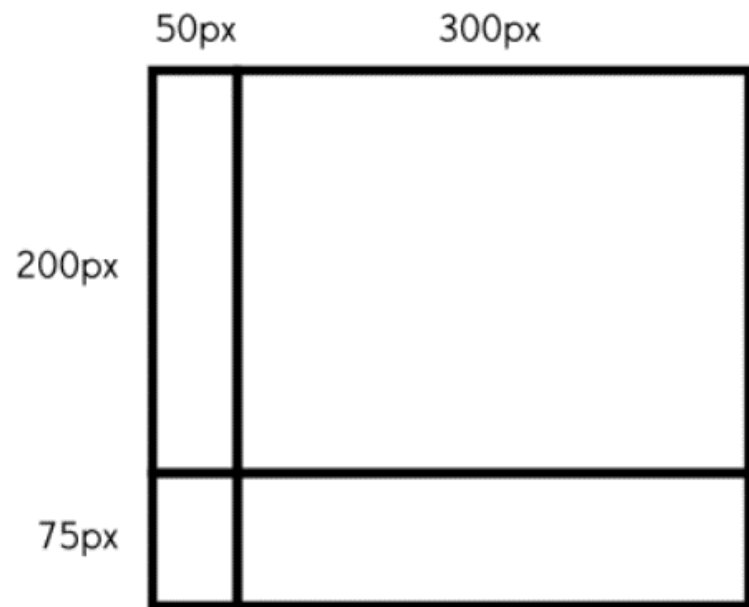
**Ítem:** Cada uno de los hijos que contiene la cuadrícula (*elemento contenedor*).

**Contenedor:** Existe un elemento padre que es el contenedor que definirá la cuadrícula o rejilla.

**Banda (grid track):** Banda horizontal o vertical de celdas de la cuadrícula.

**Celda (grid cell):** Cada uno de los cuadritos (*unidad mínima*) de la cuadrícula.

**Area (grid area):** Región o conjunto de celdas de la cuadrícula.



```
.grid {  
  display: grid;  
  grid-template-columns: 50px 300px;  
  grid-template-rows: 200px 75px;  
}
```

# Ejemplo

# Ahora sí, al código

```
<div class="grid"> <!-- contenedor -->
  <div class="a">Item 1</div> <!-- cada uno de los ítems del grid -->
  <div class="b">Item 2</div>
  <div class="c">Item 3</div>
  <div class="d">Item 4</div>
</div>
```

Propiedad a utilizar en el contenedor:

**display**

Valores posibles:

**grid:** Establece una cuadrícula con ítems en bloque, de forma equivalente a block.

**inline-grid:** Establece una cuadrícula con ítems en línea, de forma equivalente a inline-block.

# Ahora configuramos la cuadrícula

Hay varias alternativas que tenemos para diagramar nuestra cuadrícula,



**Grid con filas y columnas**



**fr: Unidad fracción restante**



**Filas y columnas repetitivas**

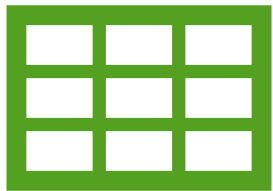


**Grid por áreas**



**Grid con huecos**

# Grid con filas y columnas



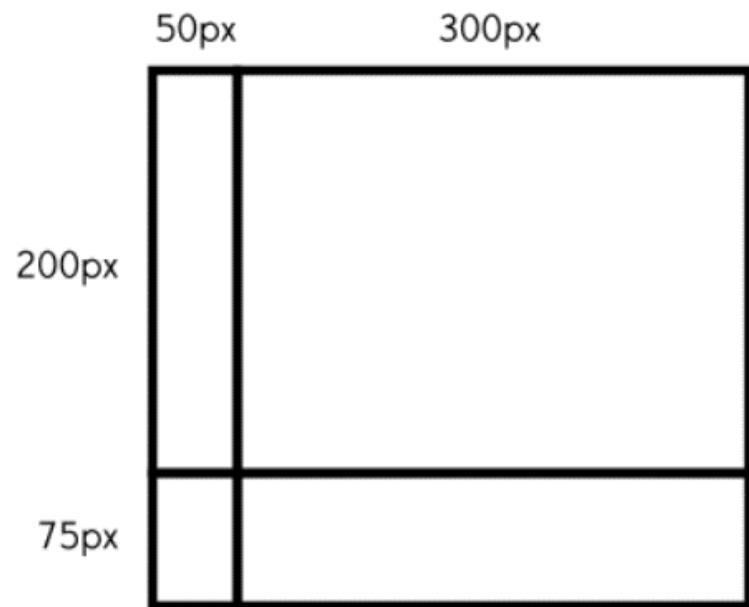
## grid-template-columns

Define una lista de los anchos de cada columna



## grid-template-rows

Define una lista con los altos de cada fila



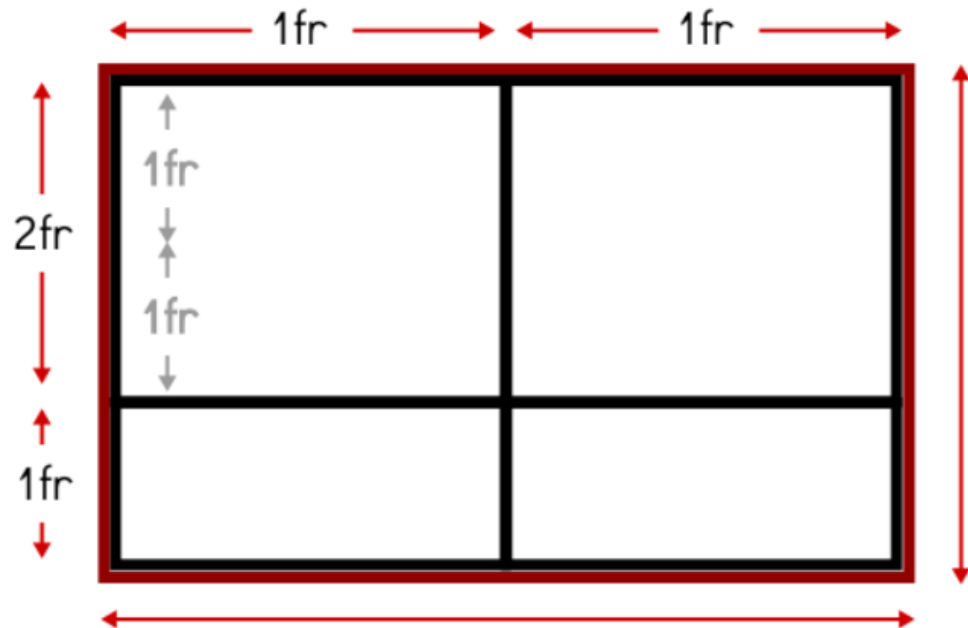
```
.grid {  
  display: grid;  
  grid-template-columns: 50px 300px;  
  grid-template-rows: 200px 75px;  
}
```

# Ejemplo



# fracción de espacio restante en el grid

Se utilizan las mismas propiedades nombradas anteriormente pero con valores con unidad de medida fr



```
.grid {  
  display: grid;  
  grid-template-columns: 1fr 1fr;  
  grid-template-rows: 2fr 1fr;  
}
```

# Ejemplo

# Tip

- Se pueden combinar varias unidades diferentes, pudiendo utilizar píxeles (*px*) y fracciones restantes (*fr*), porcentajes (%) y fracciones restantes (*fr*) o combinaciones similares.



# Filas y columnas repetitivas

- REPEAT ES UNA EXPRESIÓN ÚTIL CUANDO TENEMOS VALORES REPETITIVOS
- INDICAMOS EL NÚMERO DE VECES QUE SE REPITE Y EL VALOR DE REPETICIÓN

```
.grid {  
  display: grid;  
  grid-template-columns: 100px repeat(2, 50px) 200px;  
  grid-template-rows: repeat(2, 50px 100px);  
}
```

```
.grid {  
  display: grid;  
  grid-template-columns: 100px 50px 50px 200px;  
  grid-template-rows: 50px 100px 50px 100px;;  
}
```

Ejemplo  
Son equivalentes

# Grid por áreas

Es posible indicar el nombre y posición concreta de cada área de una cuadrícula.

# Grid por áreas

- **grid-template-áreas**

- Indica la disposición de las áreas en el grid. Cada texto entre comillas simboliza una fila.

- **grid-area**

- Indica el nombre del área. Se usa sobre ítems hijos del grid.



```
.grid {  
  display: grid;  
  grid-template-areas: "head head"  
                      "menu main"  
                      "foot foot";  
}  
  
.a { grid-area: head; background: blue }  
.b { grid-area: menu; background: red }  
.c { grid-area: main; background: green }  
.d { grid-area: foot; background: orange }
```

# Ejemplo

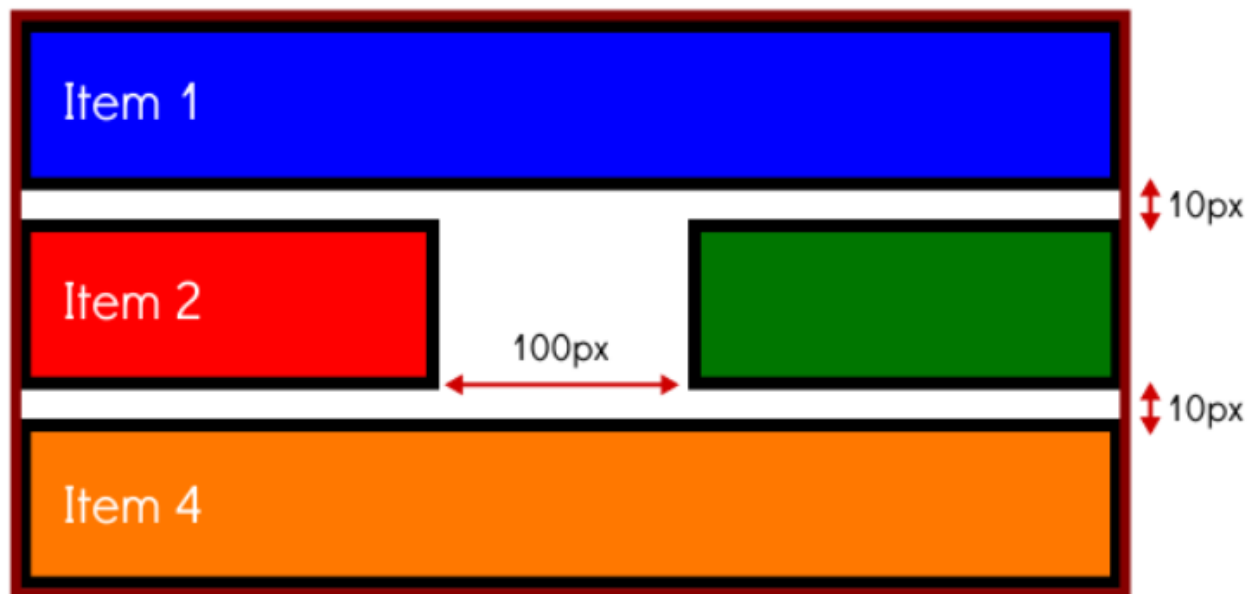


# Grid con huecos

Se especifican “huecos” que son los espacios entre celdas.

# Grid con huecos

- grid-column-gap
  - Establece el TAMAÑO de los huecos entre columnas (líneas verticales).
- grid-row-gap
  - Establece el TAMAÑO de los huecos entre filas (líneas horizontales).

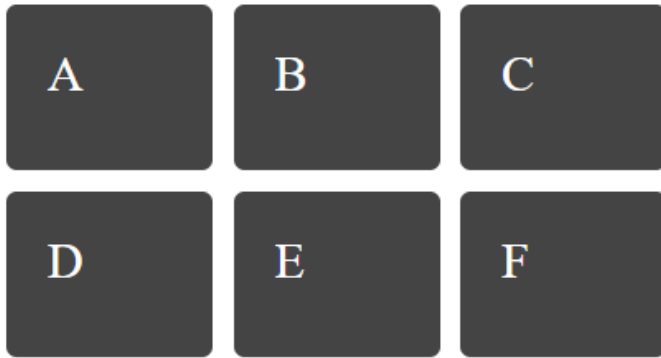


```
.grid {  
  grid-column-gap: 100px;  
  grid-row-gap: 10px;  
}
```

# Ejemplo

# Maquetando

con GRID



```
<div class="wrapper">
  <div class="box a">A</div>
  <div class="box b">B</div>
  <div class="box c">C</div>
  <div class="box d">D</div>
  <div class="box e">E</div>
  <div class="box f">F</div>
</div>
```

```
.wrapper {
  display: grid;
  grid-template-columns: 100px 100px 100px;
  grid-template-rows: 80px 80px;
  grid-gap: 10px;
  background-color: #fff;
  color: #444;
}
.box {
  background-color: #444;
  color: #fff;
  border-radius: 5px;
  padding: 20px;
  font-size: 150%;
}
```

# Maquetación con GRID



```
<div class="wrapper">
  <div class="box a">A</div>
  <div class="box b">B</div>
  <div class="box c">C</div>
  <div class="box d">D</div>
</div>
```

```
.wrapper {
  display: grid;
  grid-gap: 10px;
  grid-template-columns: 100px 100px 100px;
  background-color: #fff;
  color: white;
}
.box {
  background-color: #444;
  padding: 20px;
  border-radius: 5px;
  font-size: 150%;
}
```

```
.a {
  grid-column: 1 / 3;
  grid-row: 1;
}
.b {
  grid-column: 3;
  grid-row: 1 / 3;
}
.c {
  grid-column: 1;
  grid-row: 2;
}
.d {
  grid-column: 2;
  grid-row: 2;
}
```

# Maquetación con GRID