DISEÑO MEDIANTE GRID

En esta unidad veremos:

- Aplicar y conocer los conceptos de Grid.

Información sacada de

- **https://lenguajecss.com/p/css/propiedades/grid-css**
- **https://www.w3schools.com/css/css_grid.asp**

Adaptados, ampliados y modificados por José Jesús Torregrosa García

Módulo Diseño de Interfaces Web – *Ciclo Desarrollo de Aplicaciones Web*

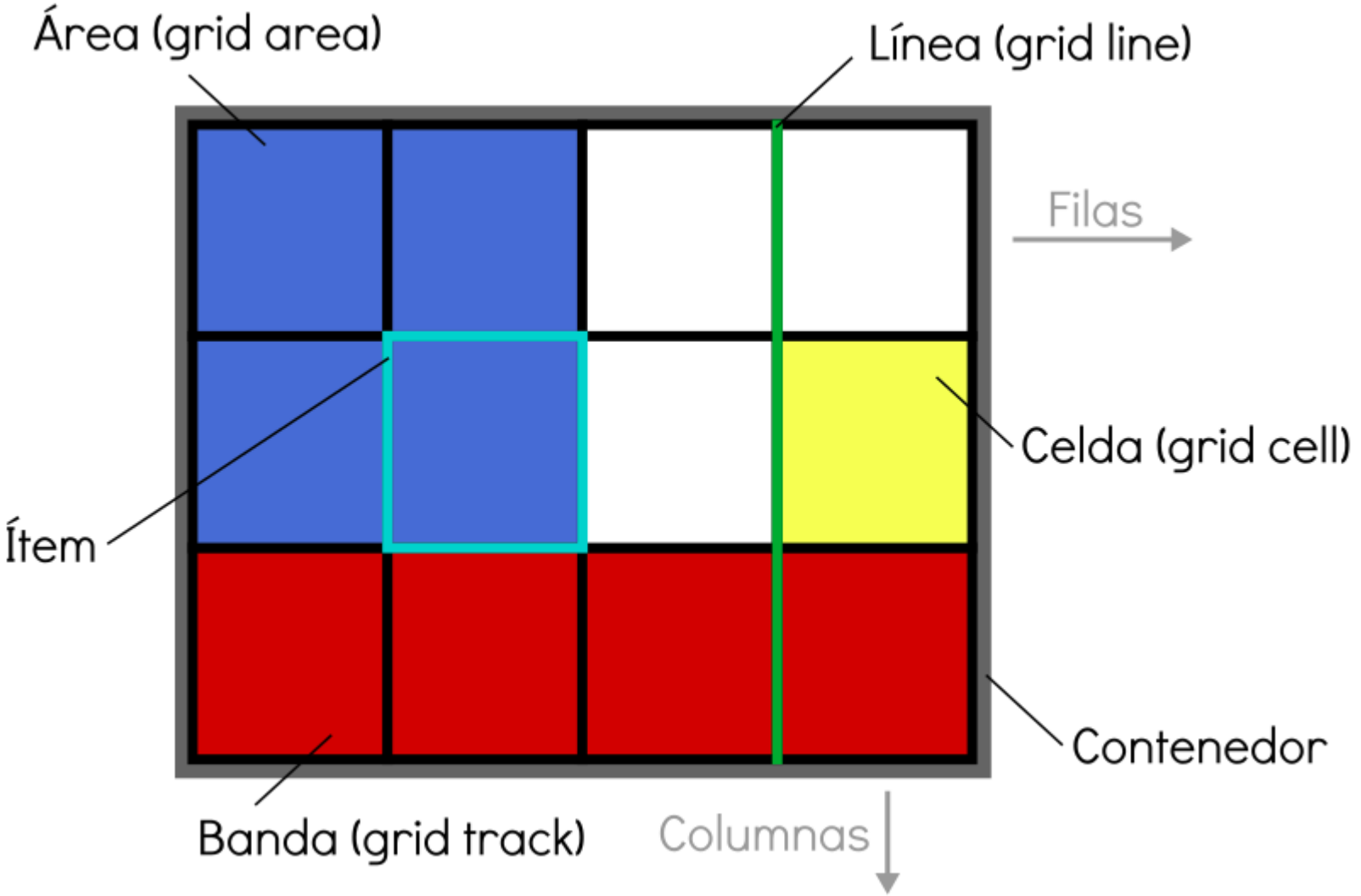
IES San Vicente 2018-2019

Generalitat Valenciana Curso 2018 - 2019

INTRODUCCIÓN

- Hemos visto el modelo de caja, el diseño mediante tablas, floats y position, y el módulo **Flexible Box Layout**,, el W3C ofrece una nueva posibilidad: las cajas flexibles
- El sistema flexbox es una gran mejora, sin embargo, está orientado a estructuras de una sola dimensión, por lo que aún necesitamos algo más potente, algo que sea multidimensional: **Grid**
- Concepto clave en Grid:
  - **Contenedor:** Existe un elemento padre que es el contenedor que definirá la cuadrícula o rejilla.
  - **Ítem:** Cada uno de los hijos que contiene la cuadrícula (elemento contenedor)
  - **Celda (grid cell):** Cada uno de los cuadritos (unidad mínima) de la cuadrícula
  - **Area (grid area):** Región o conjunto de celdas de la cuadrícula
  - **Banda (grid track):** Banda horizontal o vertical de celdas de la cuadrícula.
  - **Línea (grid line):** Separador horizontal o vertical de las celdas de la cuadrícula.

INTRODUCCIÓN



INTRODUCCIÓN

- Para utilizar cuadriculas **Grid CSS**, trabajaremos bajo el siguiente escenario:

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

display:grid - display:inline-grid

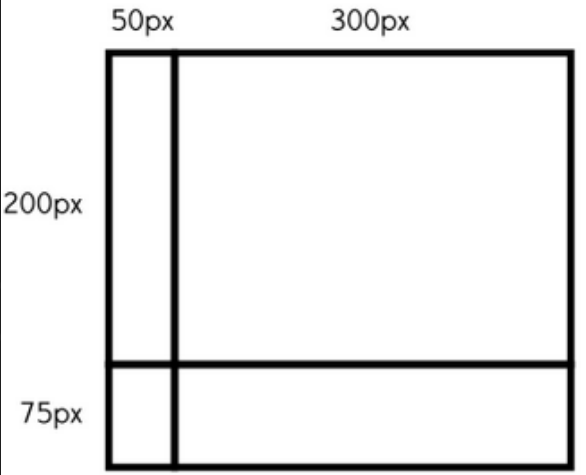
- Display: activar la cuadrícula grid. Valores:
  - **grid:** cuadrícula aparezca encima/debajo del contenido exterior (en bloque)
  - **inline-grid:** permite que la cuadrícula aparezca a la izquierda/derecha (en línea) del contenido exterior.

Tipo de elemento	Descripción
inline-grid	Establece una cuadrícula con ítems en línea, de forma equivalente a inline-block.
grid	Establece una cuadrícula con ítems en bloque, de forma equivalente a block.

GRID CON FILAS Y COLUMNAS EXPLÍCITAS (Ejemplo1.html)

Propiedad	Descripción
grid-template-columns	Establece el tamaño de las columnas (*eje horizontal*).
grid-template-rows	Establece el tamaño de las filas (*eje vertical*).

```
<style>
  .grid {
      display: grid;
      grid-template-columns: 50px 300px;
      /* 2 columnas (la primera con 50px de ancho y
      la segunda con 300px de ancho) */
      grid-template-rows: 200px 75px;
      /* y con 2 filas (la primera con 200px de alto
      y la segunda con 75px de alto). */
  }
</style>
```



GRID CON FILAS Y COLUMNAS EXPLÍCITAS: UNIDADES (Ejemplo2.html)

- También podemos utilizar otras unidades (e incluso combinarlas) como **porcentajes, etc.**
- Está la palabra clave **auto** (que obtiene el tamaño restante) o la unidad especial **fr** (fraction), que simboliza una fracción de espacio restante en el **grid**.

```
<style>
  .grid {
      display: grid;
      grid-template-columns: 1fr 1fr;
      /* 2 columnas con igualdad de ancho  */
      grid-template-rows: 1fr 3fr;
      /* y con 2 filas (la segunda el triple
      de la primera). */
      border: solid 1px #000000;
  }
</style>
```

FILAS Y COLUMNAS REPETITIVAS (Ejemplo3.html)

- En las propiedades grid-template-columns y grid-template-rows podemos indicar expresiones de **repetición**, indicando celdas que repiten un mismo patrón de celdas varias veces. La expresión a utilizar sería la siguiente: **repeat([núm de veces], [valor o valores])**. Veamos un ejemplo:

```
  <style>
    .grid {
        display: grid;
        grid-template-columns: 100px repeat(2, 50px) 200px;
        grid-template-rows: repeat(2, 50px 100px);
        border: solid 1px #000000;
    }
  </style>
</head>
<body>
    <div class="grid"> <!-- contenedor -->
        <!-- cada uno de los ítems del grid -->
        <div class="a">Item 1</div>
        <div class="b">Item 2</div>
        <div class="c">Item 3</div>
        <div class="d">Item 4</div>
        <div class="e">Item 5</div>
        <div class="f">Item 6</div>
        <div class="g">Item 7</div>
        <div class="h">Item 8</div>
    </div>
</body>
```

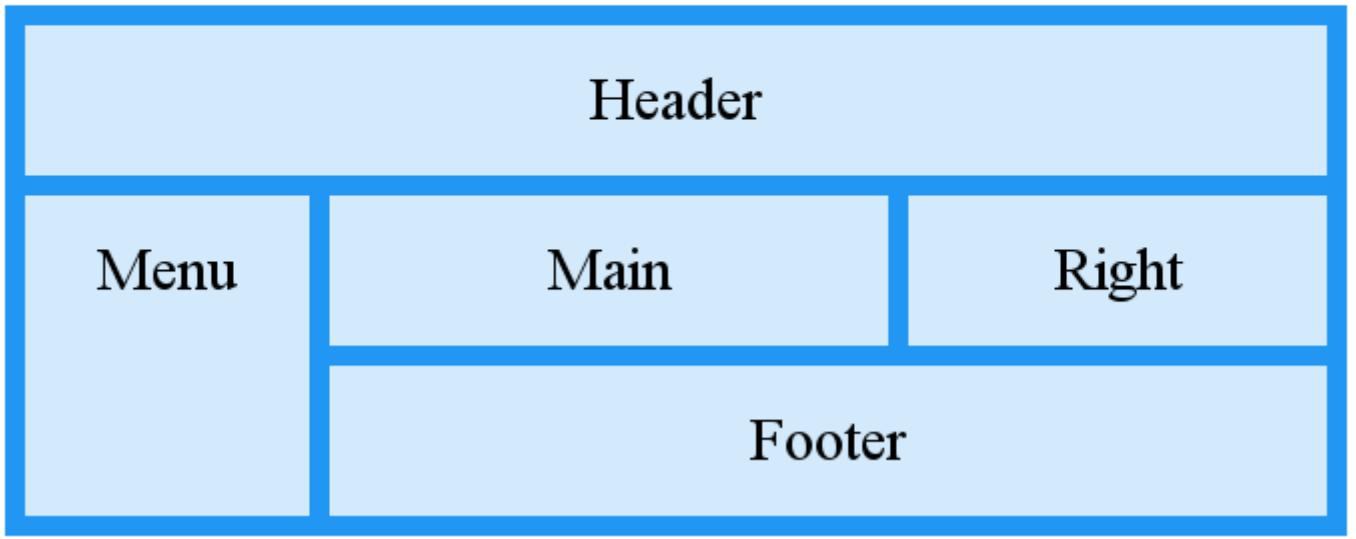
GRID POR ÁREAS Y PROPIEDAD grip-gap (Ejemplo4.html)

- Es sencillo, el código no aparece aquí está en el ejemplo.

Propiedad	Descripción
grid-template-areas	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.

Propiedad Grid area

Contiene 6 columnas y tres filas



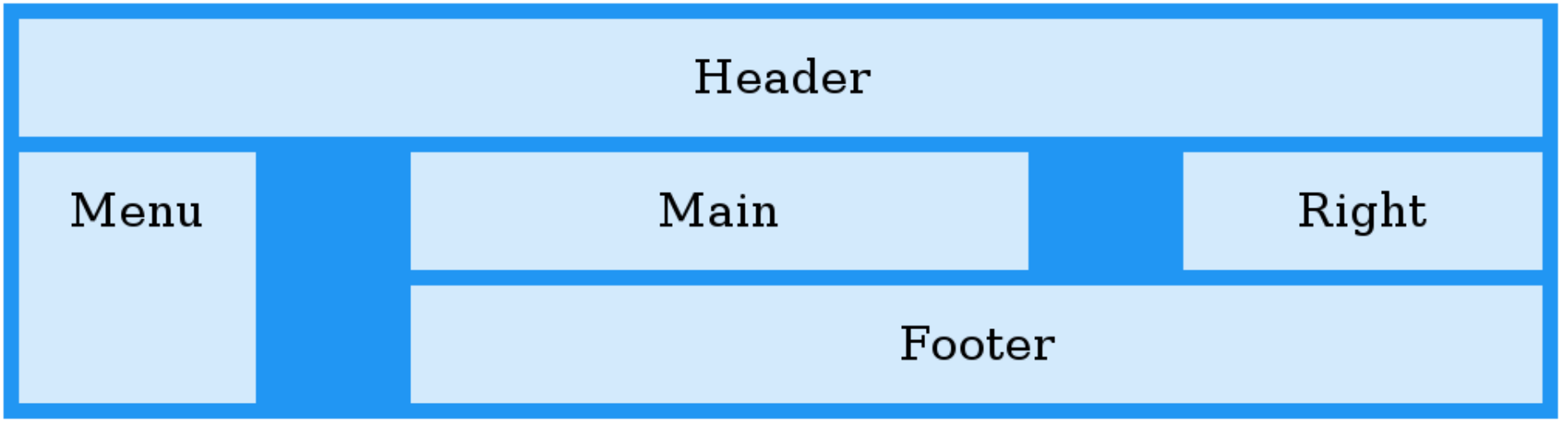
GRID CON HUECOS EN grid-area (Ejemplo5.html)

- El código no aparece aquí está en el ejemplo.

Propiedad	Descripción
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*).

Propiedad Grid area

Contiene 6 columnas y tres filas



POSICIÓN DE LOS ELEMENTOS EN EL GRID

Propiedad	Valores	Descripción
justify-items	start | end | center | stretch	Distribuye los elementos en el eje horizontal.
align-items	start | end | center | stretch	Distribuye los elementos en el eje vertical.

En el caso de que queramos que uno de los ítems hijos tengan una distribución diferente al resto, aplicamos la propiedad **justify-self** o **align-self.** Las propiedades sobre ítems hijos las veremos más adelante.

POSICIÓN DE LOS ELEMENTOS EN EL GRID: justify-content O align-content

Propiedad	Valores
justify-content	start | end | center | stretch | space-around | space-between | space-evenly
align-content	start | end | center | stretch | space-around | space-between | space-evenly

Probar las propiedades en **Ejercicio6.html**

PROPIEDAD grid-autoflow

Probar las propiedades en **Ejercicio6.htm**

```
grid-auto-flow: column;  /* o 'row', 'row dense', 'column dense' */
```

- **row:** Es una palabra clave que especifica que el algoritmo de ubicación automática coloca los elementos, completando cada fila, agregando nuevas filas según sea necesario. Si no se proporciona ninguna row ni column , se asume la row column

- **column :** Es una palabra clave que especifica que el algoritmo de colocación automática coloca elementos, completando cada columna, agregando nuevas columnas según sea necesario.

PROPIEDAD grid-autoflow

Probar las propiedades en **Ejercicio6.htm**

```
grid-auto-flow: column;  /* o 'row', 'row dense', 'column dense' */
```

- **Dense:** Es una palabra clave que especifica que el algoritmo de colocación automática utiliza un algoritmo de empaquetado "denso", que intenta rellenar los huecos iniciales en la cuadrícula, si los elementos más pequeños aparecen más adelante. Esto puede hacer que los artículos aparezcan desordenados, al hacerlo llenarían los huecos que dejan los artículos más grandes.

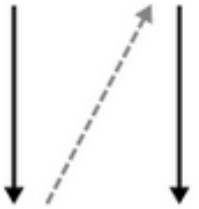
  Si se omite, se usa un algoritmo "disperso", donde el algoritmo de ubicación solo se mueve "hacia adelante" en la cuadrícula al colocar elementos, nunca retrocede para llenar los agujeros. Esto asegura que todos los elementos colocados automáticamente aparecen "en orden", incluso si esto deja agujeros que podrían haber sido llenados por items posteriores.

AJUSTE AUTOMÁTICO DE CELDAS

Propiedad	Valores	Descripción
grid-auto-columns	[*tamaño*]	Indica el tamaño automático de ancho que tendrán las columnas.
grid-auto-rows	[*tamaño*]	Indica el tamaño automático de alto que tendrán las filas.
grid-auto-flow	row | column | dense	Utiliza un algoritmo de autocolocación (intenta rellenar huecos).

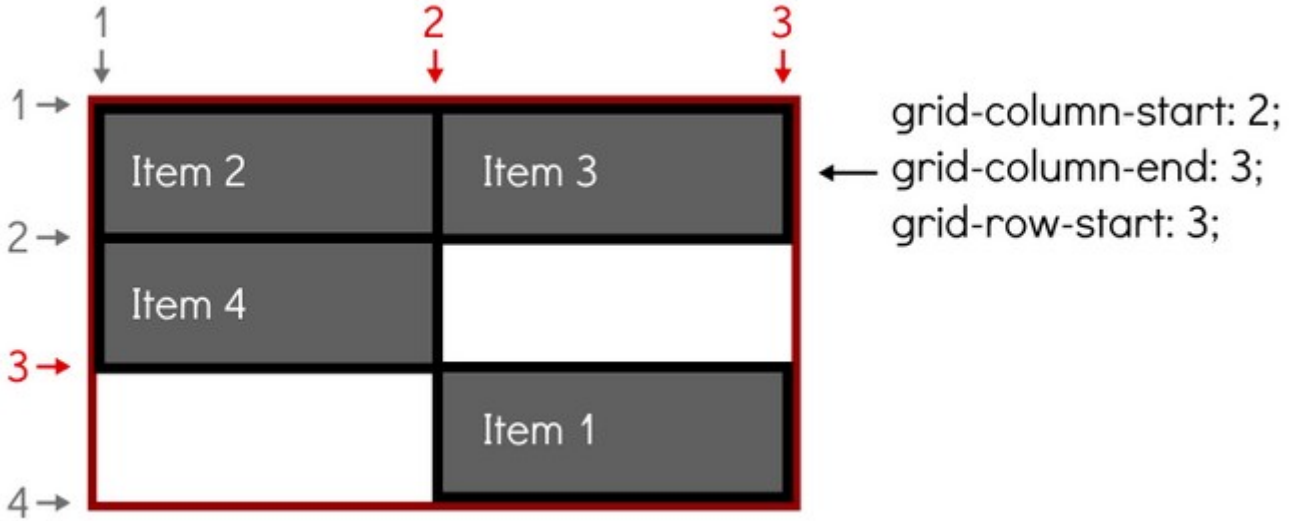
grid-auto-flow: row

grid-auto-flow: column



UN POCO MAS DE grid-area Y AJUSTE AUTOMÁTICO DE CELDAS

- Ver **ejemplo8.html**



ATAJOS GRID-COLUMN Y GRID-ROW

- Ver **Ejercicio9.html**

```
/*.item1 { grid-area: 1 / 1 / 2 / 2; }*/
.item1{
  /* grid-column: <grid-column-start> <grid-column-end> */
  /* grid-row: <grid-row-start> <grid-row-end> */
  grid-row: 1 / 2;
  grid-column: 1 / 2;
}
```

SOPORTE Y CONCLUSIONES

- Dirección Web del soporte es el siguiente:

  http://caniuse.com/#feat=css-grid

- Se aconseja profundizar ligeramente con estas páginas por ver otros ejemplos diferentes:
  - **Grid item:**
  - **https://www.w3schools.com/css/css_grid_item.asp**
  - **Grid container**: https://www.w3schools.com/css/css_grid_container.asp