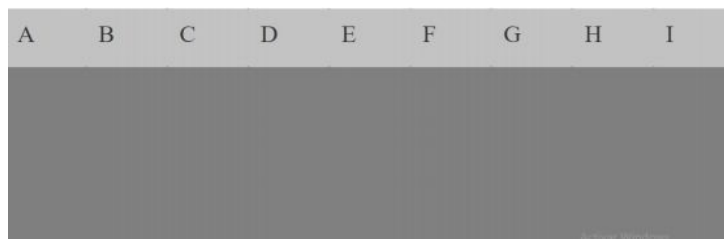


MAQUETACIÓN CSS CON GRID

grid-template-columns: 100px 100px 100px Crea las columnas del documento. En este caso como son 3 columnas de igual tamaño recomendable poner **repeat(3,100px)**. **Recomendable, poner** sus valores de este modo **grid-template-rows: 1fr 1fr 1fr** si sus tamaños son iguales, si uno es mayor que el otro, le damos un valor más: **1fr 2fr 1fr (DAR TAMAÑO AL CONTENEDOR)**

Existen otros valores para colocar al crear las filas y columnas, si queremos que sus valores sean iguales. **repeat(auto-fit, minmax(100px, 1fr))**; se encajara los divs al ancho del grid, ya que esta puesto el 1fr como maximo, **auto-fit** -> ocupa el espacio del grid dando el mismo ancho a sus columnas

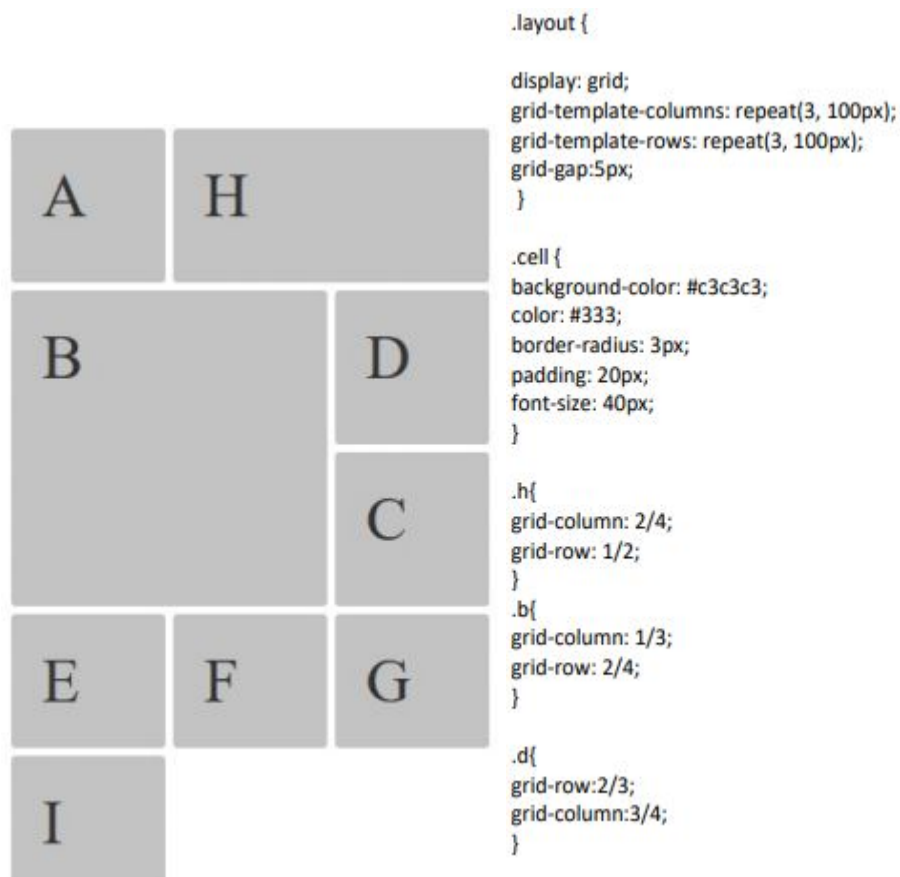


grid-template-rows: Sus valores se colocan igual que el anterior, en este caso para las filas

grid-gap: espacio entre las filas y columnas -> **grid-gap: 5px**

grid-column/grid-row: para colocar los elementos en las celdas especificadas, si hay 4 columnas, sus celdas sería **1/2 2/3 3/4 4/5** igual que para las filas

hay que tener cuidado al intentar colocar en una celda, un elemento div que contiene varios elementos, ya que se colocaran todos en la misma celda, aunque se le diga que ocupe de la columna 2 a la 4 -> **2/4 ... INTENTAD NO ENCAPSULAR EN DIV TODO**



Los elementos se irán colocando en los espacios donde quepan, no hace falta especificar la celda a todos los elementos del contenedor

grid-template-areas: es otra forma de colocar elementos en celdas, hay que

```

.header {
  grid-area: header;
}

.content {
  grid-area: content;
}

```

darle, primero se le da un nombre con **grid-area** a los elementos. Luego en el contenedor, se deberá colocar un template que diga donde se deberán colocar los elementos según su grid-area nombrado anteriormente, si queremos poner una celda en blanco, se colocaran tres puntos ...

```

grid-template-areas:
  "... headerheaderheader"
  "sidebarcontentcontentcontent"
  "footerfooterfooterfooter"
}

```

PARA ALINEAR VARIOS ELEMENTOS DENTRO DE UN DIV, ES NECESARIO TRATAR COMO FLEX ESE ELEMENTO GLOBAL Y ALINEARLO CON LAS RESPECTIVAS ETIQUETAS.

```
<footer>
  <address>
    <p>C/ Principal</p>
    <p>Sevilla</p>
  </address>
  <div class="copyright">
    Todas las fotos realizadas por @pekechis bajo la Licencia CC BY-SA
  </div>
  <div class="social">
     
    
  </div>
</footer>
```

C/ Principal

Sevilla

Todas las fotos realizadas por @pekechis bajo la Licencia CC BY-SA



FOOTER DEBE SER TRATADO COMO FLEX APARTE DE PERTENECER A UN CONTENEDOR TRATADO COMO GRID

```
footer {
  align-items: center;
  display: flex;
  grid-column: 1 / 5;
  grid-row: 4 / 5;
  justify-content: space-between;
```

C/

Principal

Sevilla

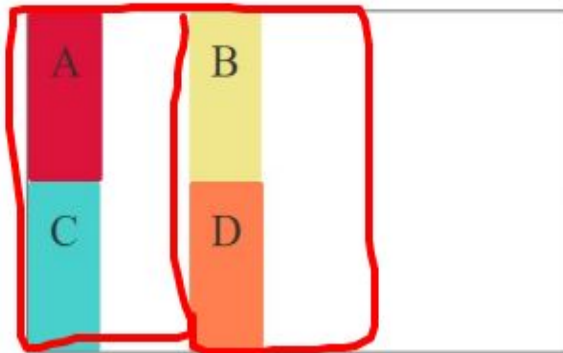
Todas las fotos realizadas por @pekechis bajo la Licencia CC BY-SA



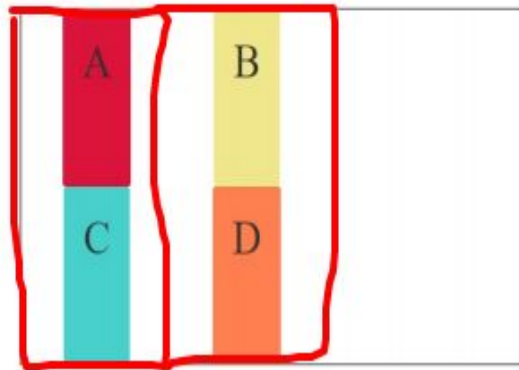
ALINEAR ELEMENTOS RESPECTO A SU CELDA

justify-items: alinear los elementos respecto a su celda horizontalmente

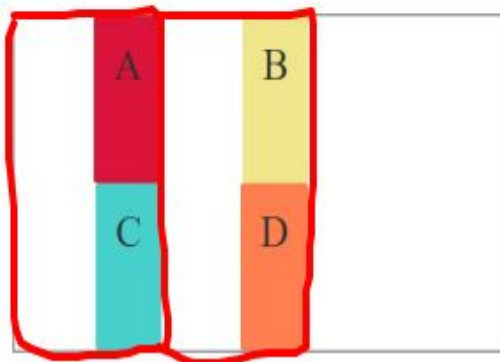
`justify-items:start`



`justify-items:center`

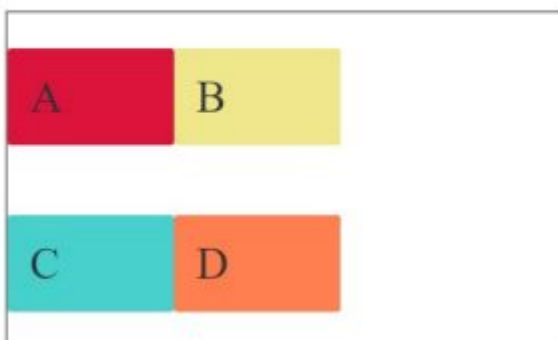


`justify-items:end;`



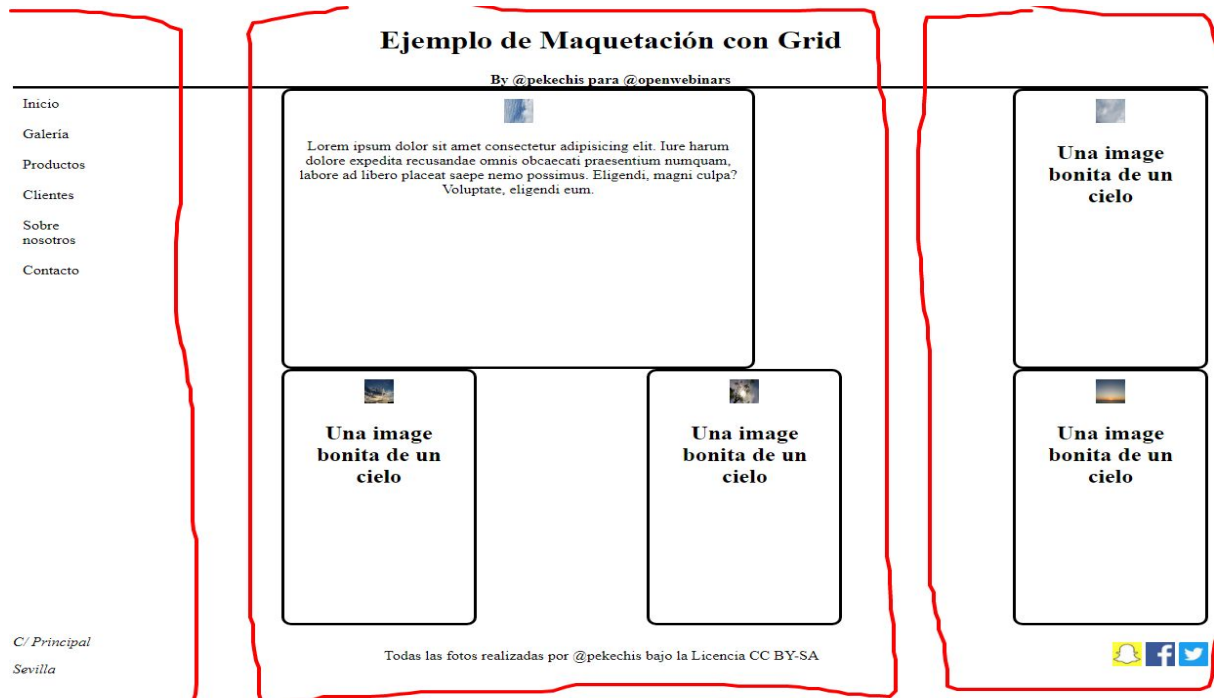
align-items: igual que el anterior, pero alineando los elementos verticalmente, por ejemplo.

`align-items:center;`



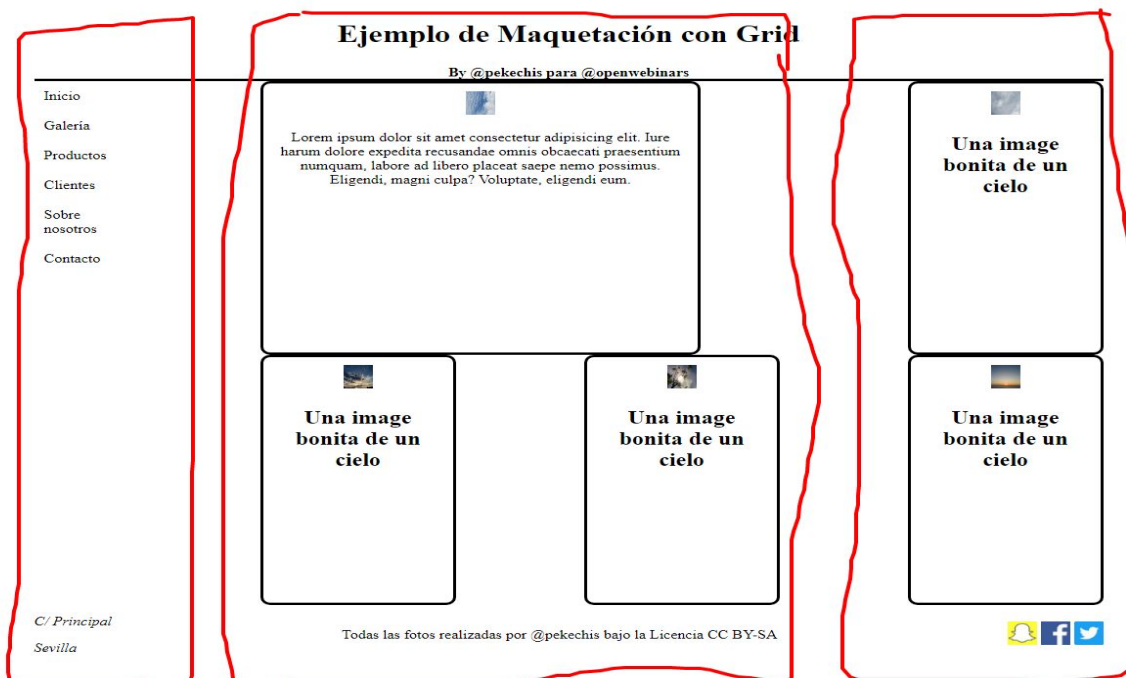
justify-content: si se coloca el **valor center** o el **valor end** alinea el grid completo y lo pone en medio o la derecha del todo

Si colocas los valores **space-between** o **space-around** se colocan dejando diferente espacio entre las columnas, por ejemplo, **space-between** deja el mismo espacio entre los elementos —>



space-around:

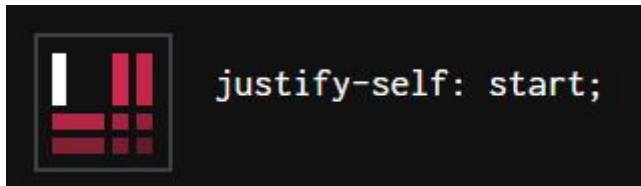
El espacio vacío anterior al primer item y posterior al último item equivale a la mitad del espacio entre dos items



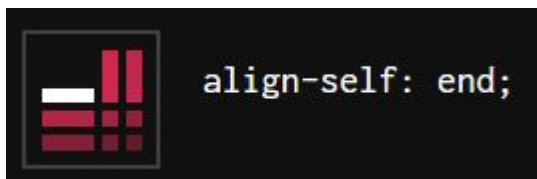
align-content: alinea el grid, verticalmente, tiene los mismos valores que el anterior -> **start,center,end,space-between,space-around**

justify-self/align-self: lo que hace es alinear la celda en la que se especifique su valor, sus valores son los mismos de siempre -> **start,end,center**

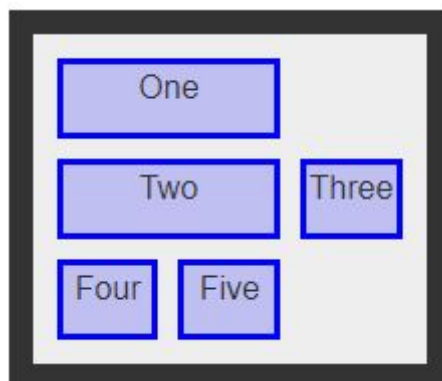
justify-self: start -> se alinea horizontalmente a la izquierda de la celda



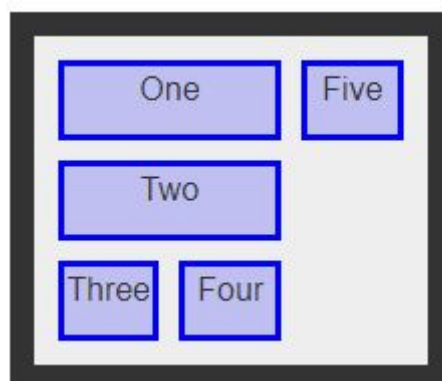
align-self: end -> se alinea verticalmente abajo de la celda(en su suelo),



grid-auto-flow: se coloca en el grid, y especifica cómo se empezaran a añadir las celdas, si de arriba a abajo, o de izquierda a derecha



grid-auto-flow: row:



grid-auto-flow: column