

Webs interesantes:

https://developer.mozilla.org/es/docs/Web/CSS/CSS_Flexible_Box_Layout

https://developer.mozilla.org/es/docs/Web/CSS/CSS_Grid_Layout

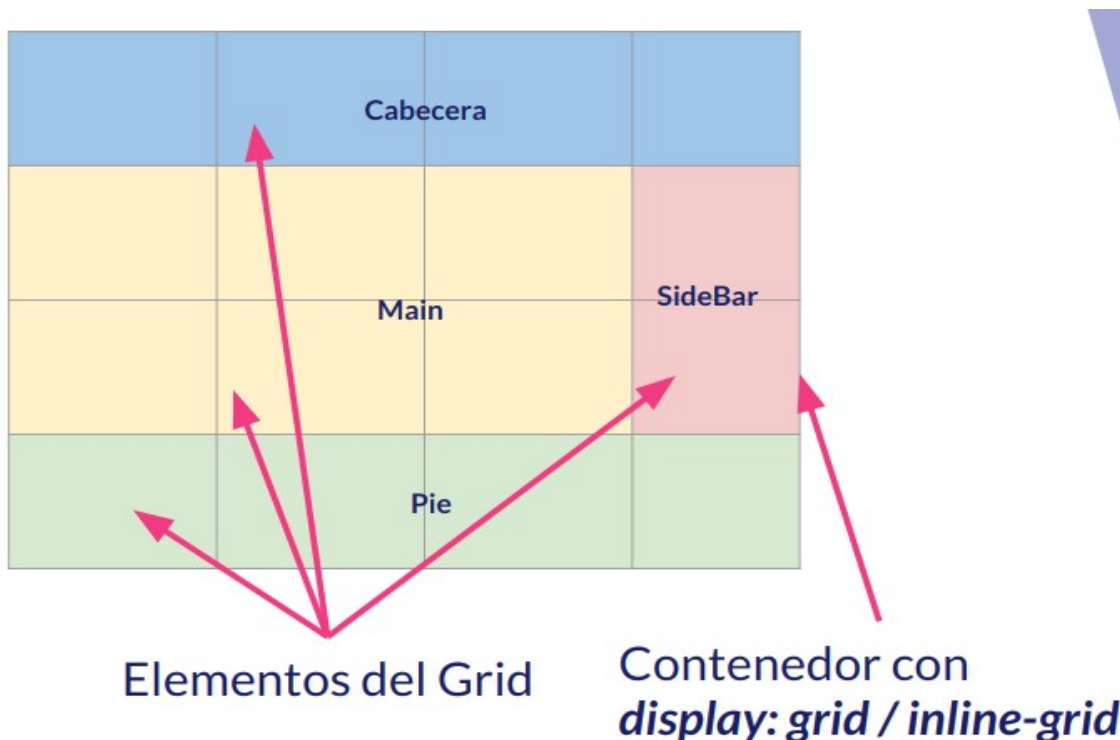
GRID

La idea principal que hay detrás de la maquetación **GRID** es que vamos a tener un elemento, es decir, un etiqueta que nos va a permitir *controlar* propiedades de los elementos que contiene y establecer **estructuras complejas** para distribuirlos.

Por lo tanto, en esta situación, vamos a poder distinguir dos tipos de elementos:

- El **contenedor GRID** que tendrá asignada la propiedad CSS *display:grid* o *display:inline-grid* la estructura y las propiedades de los elementos que contiene.
- Los **elementos GRID** que son los elementos que están dentro del contenedor, elementos que distribuiremos y cuyas propiedades modificaremos.

De manera visual podemos ver una distribución de los elementos en la siguiente imagen:



Pero, ¿qué podremos modificar desde el contenedor?:

- La estructura en filas y columnas y la separación entre ellas.
- Definir áreas del GRID con nombre.
- La alineación horizontal y vertical de los elementos del GRID y del propio GRID dentro del elemento que lo contiene.

Es decir, vamos a poder controlar propiedades que usamos para maquetar y además, vamos a poder maquetar de manera mucho más ágil a lo que lo hacemos con las técnicas tradicionales de maquetado.

PARA DEFINIR LA ESTRUCTURA USAREMOS:

- ▶ `grid-template-columns` : Número de columnas del grid y tamaño de las mismas.
- ▶ `grid-template-rows`: Número de filas del grid y tamaño de las mismas.
- ▶ `grid-row-gap`: Separación entre las filas.
- ▶ `grid-column-gap`: Separación entre las columnas.

COMO SE PUEDEN DEFINIR LAS COLUMNAS:

`grid-template-columns`: Debemos de poner tantos valores de anchura como columnas quiero que tenga el GRID.

`grid-template-columns`: 20% 50% 30%

`grid-template-columns`: 100px auto 100px 100px

`grid-template-columns`: auto auto auto auto auto

`grid-template-columns` : [id] 100px [nombre] 300px [apellidos] auto

De idéntica manera para filas pero con la opción correspondiente de `grid-template-rows`.

REPETICIÓN DE VALORES TANTO EN COLUMNAS COMO EN FILAS:

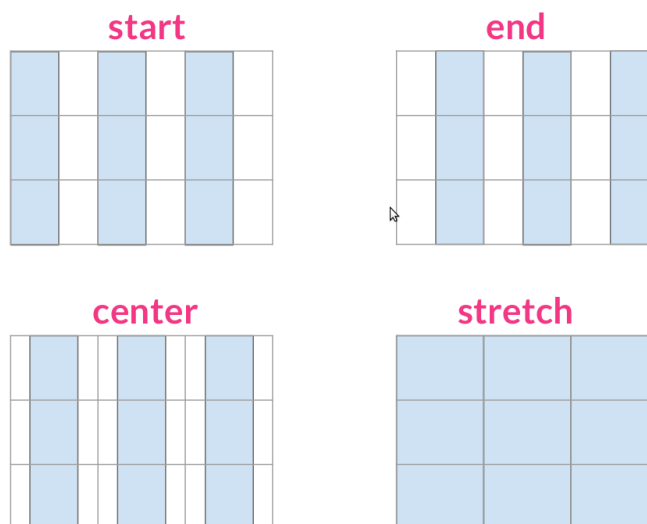
`grid-template-column`: repeat(3, 20% [col-start]) auto

ALINEACIÓN HORIZONTAL DEL CONTENIDO EN LAS REJILLAS:

Para alinear de forma horizontal el contenido en las rejillas usaré la propiedad `justify-items`, que puede contener los siguientes valores:

- `start`
- `end`
- `center`
- `stretch` (por defecto)

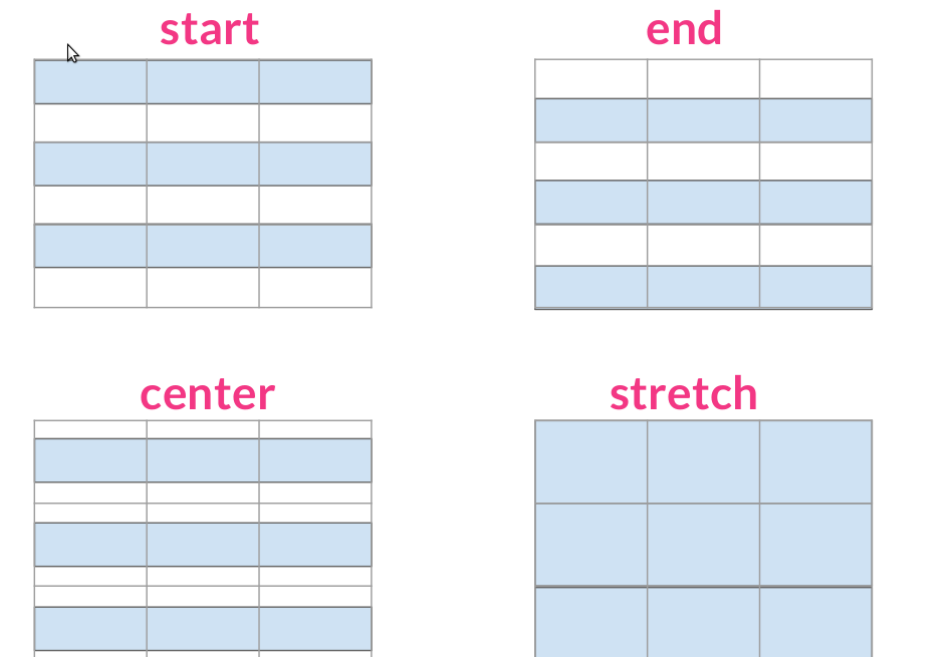
Veamos cómo quedaría:



ALINEACIÓN VERTICAL DEL CONTENIDO EN LAS REJILLAS:

Para alinear de forma vertical el contenido en las rejillas usaré la propiedad `align-items`, que puede contener los siguientes valores:

- `start`
- `end`
- `center`
- `stretch` (por defecto)



“

Puedo juntar las dos alineaciones anteriores en la propiedad `place-items`

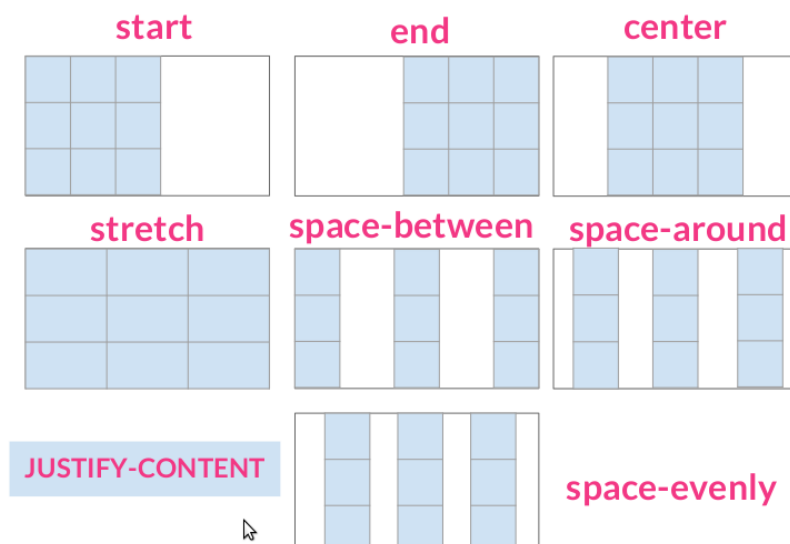
`place-item: start end`

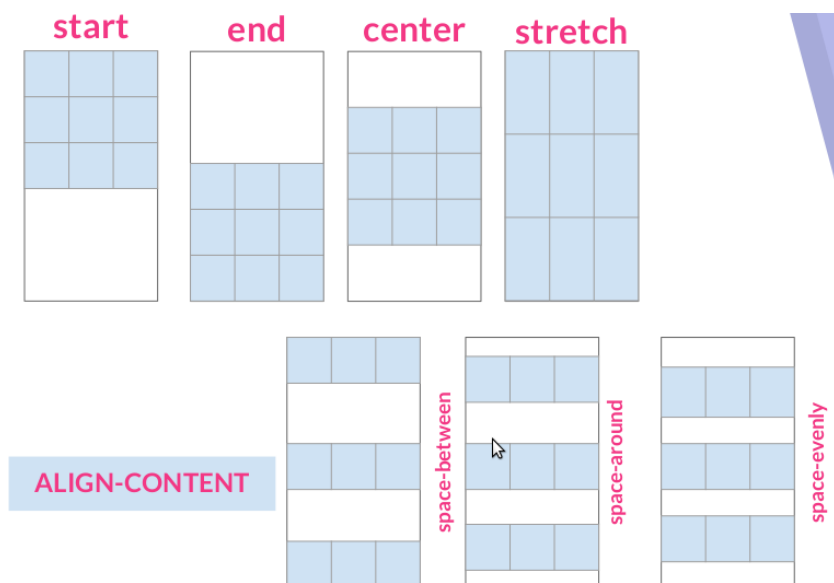
El primer valor hace referencia a la alineación vertical y el segundo valor a la alineación horizontal.

“

En el caso de que el GRID no ocupe todo el contenedor GRID podemos distribuirlo con las propiedades CSS:

`justify-content` (horizontal) y `align-content` (vertical)





Puedo juntar las dos distribuciones anteriores en la propiedad `place-content`:

`place-content: start end`

El primer valor hace referencia a la alineación vertical y el segundo valor a la alineación horizontal.