



GRID

Universidad de Sevilla
IPO

CSS: DISEÑO GRID

Diseños Grid (rejilla)

- Diseños **bidimensionales**
- Centrado vertical y horizontal
- Principios: simetría, equilibrio

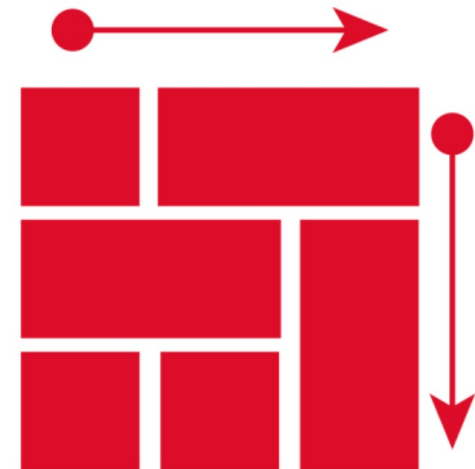


Netlify IPO: grid



Flexbox

ONE DIMENSION



CSS Grids

TWO DIMENSIONS

CSS: GRID | CONCEPTOS

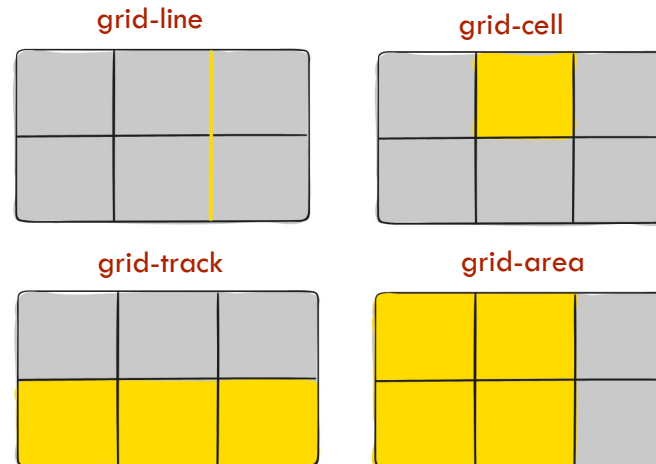
[CSS-tricks: Guía de Grid/](#)

Contenedor grid (Rejilla)

- **display: grid** | inline-grid

Conceptos:

- Contenedor vs continente
- Línea
- Celda (cell)
- Pista (track)
- Área



CSS: GRID | TEMPLATES

Template (Establece el tamaño de las pistas de la rejilla)

Propiedades:

- **grid-template-columns**
- **grid-template-rows**
- **grid-template:** <rows> / <columns>

Unidades:

- Valor **none** (por defecto)
- Valor **fr**
- **minmax**(v-min , v-max)
- Valor **auto**

Constructor repeat

- **repeat**(n, ...)
- **repeat**(auto-fill, ...)
- **repeat**(auto-fit, ...)

Ejemplos:

```
grid-template-columns: 20px 20px
grid-template-columns: 20% auto
grid-template-rows: repeat(4, 1fr)
grid-template-rows: repeat(3, 1fr 20px 1fr)
grid-template-rows: 20px minmax(100px, 300px)
grid-template: repeat(auto-fill, 40px) / 1fr 1fr
grid-template: repeat(auto-fit, minmax(100px, 1fr)) / 40vw
```

NOTA: Los patrones admiten la notación [name] para dar nombre a las líneas de la rejilla

```
grid-template-columns: [c-uno] 50px [c-dos] 100px [c-tres]
```

CSS: GRID | PLACE-ITEMS

Respecto a la celda

- Eje Block (vertical o Y): **align-items:** stretch | start | end | center
- Eje Inline (horizontal o X): **justify-items:** stretch | start | end | center
- Abreviatura: **place-items:** <align-items> <justify-items>

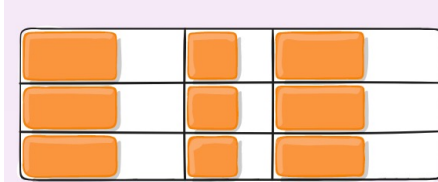
stretch stretch



end stretch



stretch start

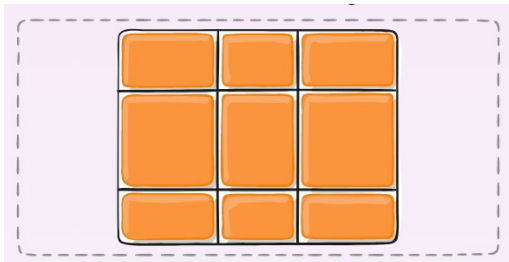


CSS: GRID | CONTENEDOR

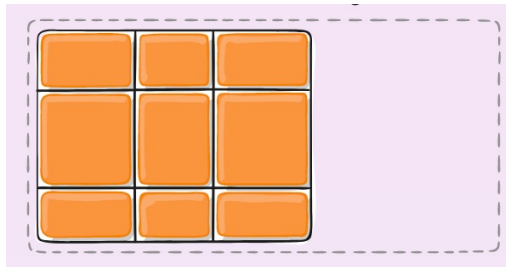
Respecto al contenedor

- OJO: Sólo se aprecia si hay espacio disponible (el tamaño de las celdas debe ser inflexible)
- Eje Block : **align-content:** `stretch` | `start` | `end` | `center` | `space-around` | `space-between` | `space-evenly`
- Eje Inline: **justify-content:** `stretch` | `start` | `end` | `center` | `space-around` | `space-between` | `space-evenly`
- Abreviatura: **place-content:** `<align-content>` `<justify-content>`

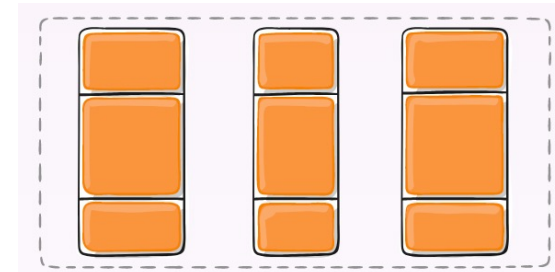
stretch center



stretch start



stretch space-around

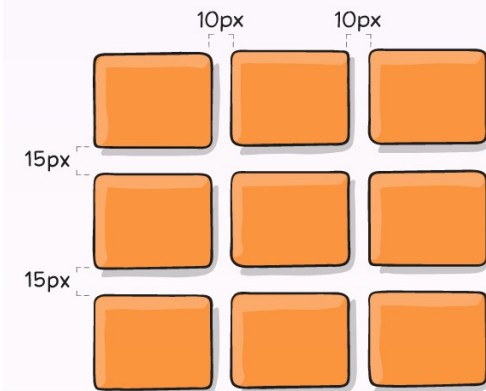


CSS: GRID | GAP

Gap (separación **interior** entre celdas)

- Propiedad **row-gap**
- Propiedad **column-gap**
- Abreviatura **gap**: <row-gap> <column-gap>

gap: 15px 10px

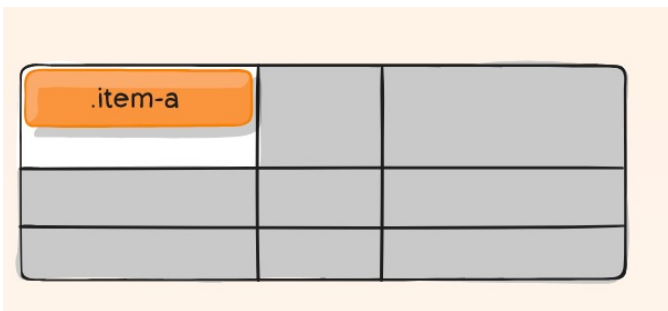


CSS: GRID | SELF

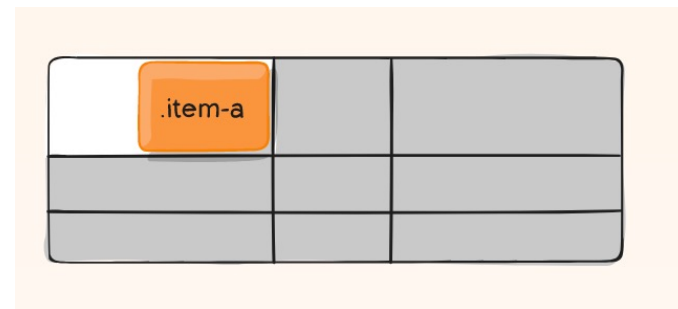
Propiedades para ubicar un elemento particular en el interior de una celda

- Eje Block (Eje Y): **align-self:** **stretch** | start | end | center
- Eje Inline (Eje X): **justify-self:** **stretch** | start | end | center
- Abreviatura: **place-self:** <align-self> <justify-self>

```
.item-a {  
  place-self: stretch start;  
}
```



```
.item-a {  
  place-self: end stretch;  
}
```

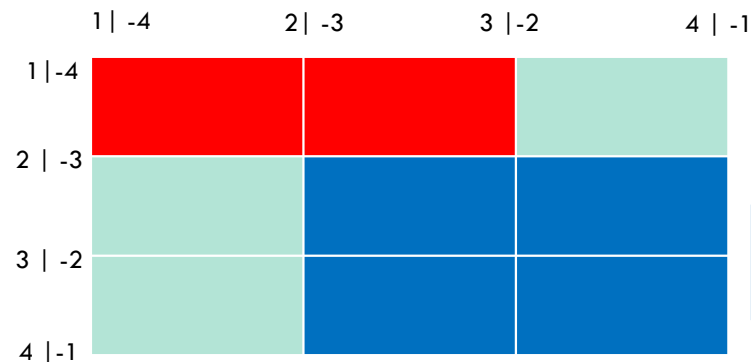


CSS: GRID | REUBICACIÓN

Propiedades para reubicar elementos en celdas (áreas) determinadas

- Propiedad **grid-column**: <grid-column-start> / <grid-column-end>
- Propiedad **grid-row**: <grid-row-start> / <grid-row-end>
- Valores
 - Valores negativos (numeran las líneas desde el final (-1, -2, ...))
 - Valor **span** <n>

grid-columns: 1 / 3
grid-rows: 1 / 2



grid-columns: 2 / span 2
grid-rows: 2 / -1

CSS: GRID | ÁREAS

Una notación alternativa para la definición de rejilla es mediante áreas

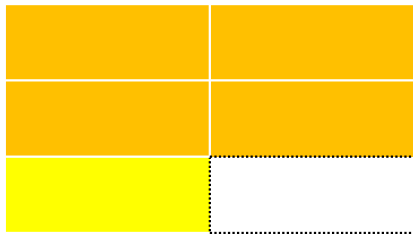
- **grid-template-area:** “rojo rojo naranja” “amarillo azul azul” “. verde verde”
- **grid-area:** cabecera



CSS: GRID EXPLÍCITO VS IMPLÍCITO

Grid explícito/implícito:

- La propiedad grid-template establece la estructura deseada de la rejilla (Grid explícito), pero otros factores (número de elementos, reubicación de elementos, creación de áreas) pueden requerir la ampliación de la rejilla (Grid implícito)

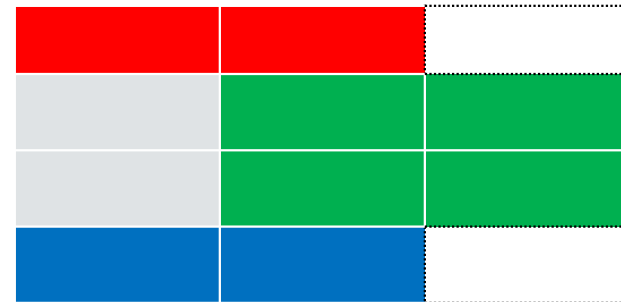


Se parte de una rejilla de dos filas y dos columnas
El contenedor tiene 5 elementos.

Para poder ubicar el quinto elemento (en amarillo)
se añade una nueva fila no prevista originalmente.

Grid explícito: 2x2

Grid implícito: 3x2



Se parte de un contenedor grid de 4 filas y 2 columnas

El contenedor incluye 8 elementos.

Se exige que los elementos de color verde se sitúen a partir de la segunda columna.

Para conseguirlo se añade una nueva columna no prevista originalmente.

Grid explícito: 4x2

Grid implícito: 4x3

CSS: GRID IMPLÍCITO

Propiedades para ajustar los elementos en el grid explícito:

- Propiedad **grid-auto-row** (tamaño de las filas creadas implícitamente)
- Propiedad **grid-auto-column** (tamaño de las columnas creadas implícitamente)
- Propiedad **grid-auto-flow**: **row** | **column** | **dense**
 - Valores row y column: los elementos se van recolocando en la siguiente celda vacía de la fila o columna.
 - El valor dense recoloca los elementos en la primera celda vacía del contenedor. Con esta estrategia se intenta reducir el número de celdas vacías.

