

Manual The CSS Grid Enchiridion

Developer Handbook to CSS Grid



Stephen Koo [Follow](#)

Oct 10, 2017 · 5 min read

A handy reference guide on how to use the CSS Grid.

Resources

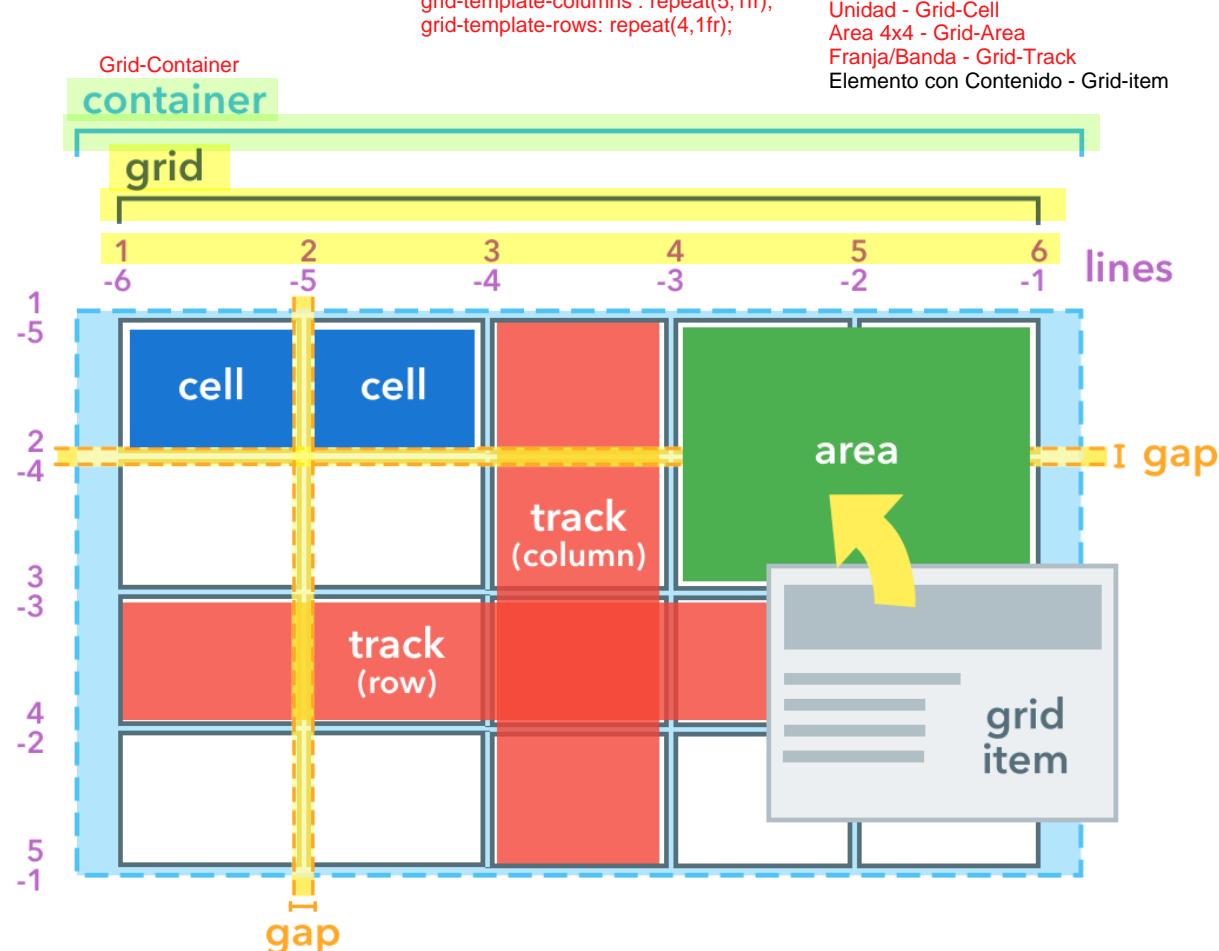
- [MDN | grid](#)
- [CSS-Tricks | A Complete Guide to CSS Grid](#)
- [Grid by Example | Grab & Go Patterns](#)
- [Grid by Example | Examples](#)

<https://codepen.io/stephenykoo>

0. Grid Overview

Grid-Element es cada división de un elemento fila o columna
Grid-Item es un elemento que ocupa cierto número de filas y columnas

Grid elements



Grid properties



Asegurarse navegador soporta GRID

Ensure browser grid support

Evitar el cual usa la más antigua especificacion GRID desde usando Prevent IE (which uses an older Grid spec) from using CSS Grid by checking if a browser soportada algunas de estas nuevas propiedades especficas supports any of these new Grid spec properties:

```
@supports (grid-area: auto) { ... }
@supports (grid-column-gap: 1px) { ... }
@supports (grid-auto-flow: dense) { ... }
@supports (grid-auto-rows: max-content) { ... }
@supports (grid-auto-columns: min-content) { ... }
```

Crear el grid

1. Make the grid

Crear un elemento grid

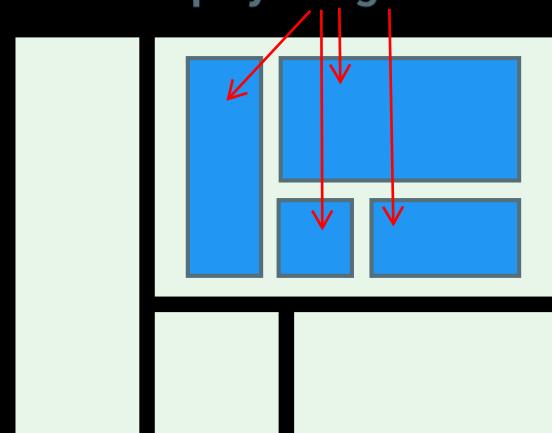
1.1. Make an item a grid



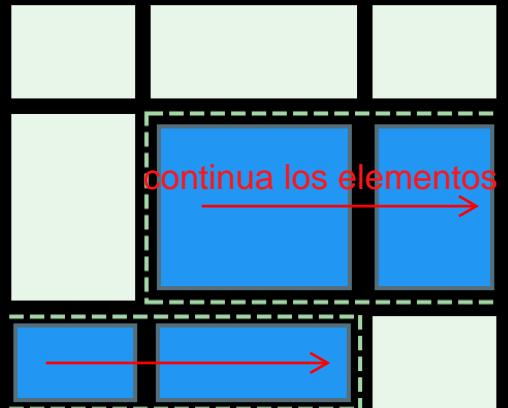
Anidado grid mostrar propiedades

Nested grid display properties

display: subgrid



display: contents



1.2. Choose whether grid is explicit, implicit, or both

Elegir SI

If you want:

Una cantidad establecida de columnas y filas

- A fixed amount of columns and rows — explicit grid

Le indicamos directamente el tamaño al grid

Automaticamente creadas columnas o filas basadas en la cantidad de elementos

- Automatically created columns or rows based on the amount of items — implicit grid

Una cantidad fija de columnas y filas + columnas o filas automáticamente creadas

- A fixed amount of columns and rows + automatically created columns or rows if there are more items than your explicit grid can contain (or when you place items outside your explicit grid) — explicit + implicit grid

de columnas y filas

automaticamente creadas

columnas o filas

si hay más elementos de los que su grid explícito pueda contener

o cuando tu colocas elementos

there are more items than your explicit grid can contain (or when you place items

fueras de tu grid explícito

o cuando tu colocas elementos

outside your explicit grid) — explicit + implicit grid

Establezco el tamaño a los elementos

El contenido establece el tamaño de los elementos que he creado como grid-items

Significado : explicito -
Forma clara y detallada de una cosa

Significado : implícito -
Que está incluido en una cosa,
sin que esta lo diga o lo especifique.

1A. Create an explicit grid

container

Yo le indico el tamaño del contenedor y sus elementos grid-items

```
grid
|- grid-template: grid-template-areas grid-template-rows / grid-template-columns;
|- grid-gap:      grid-row-gap / grid-column-gap;

/* === GRID TRACKS === */
grid-template-rows: 100px 30vmin 20vmax 5em 6rem;
grid-template-columns: 1fr 20% 25vw 30vh;
Line 1   2   3   4   5   6
grid-template: 100px 30vmin 20vmax 5em 6rem / 1fr 20% 25vw 30vh; // Shorthand
Filas           Columns
/* === GRID GAPS === */
grid-gap: 20px 20px;
grid-gap: 20px; // Shorthand
```

- grid-template sets the number and size of the tracks

fija el numero y tamaño de las franjas

- grid-gap sets the size of the gaps

"TAMAÑOS ESTABLECIDOS DIRECTAMENTE"

Tamaños bandas unidades simples

1A.1 Track sizes: Simple units

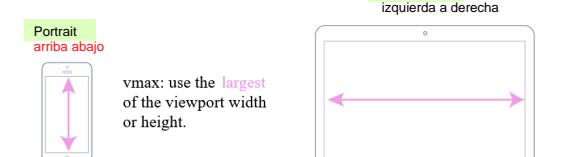
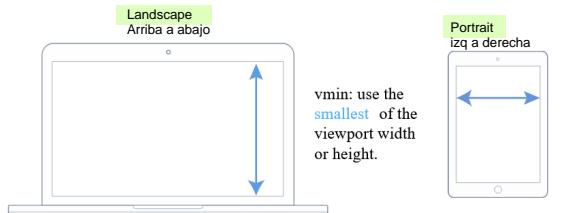
Produce wrapped - contenido esta envuelto cuando no supera el ancho del grid-item

fr, %, px, rem, em, vh, vw, vmin, vmax, none

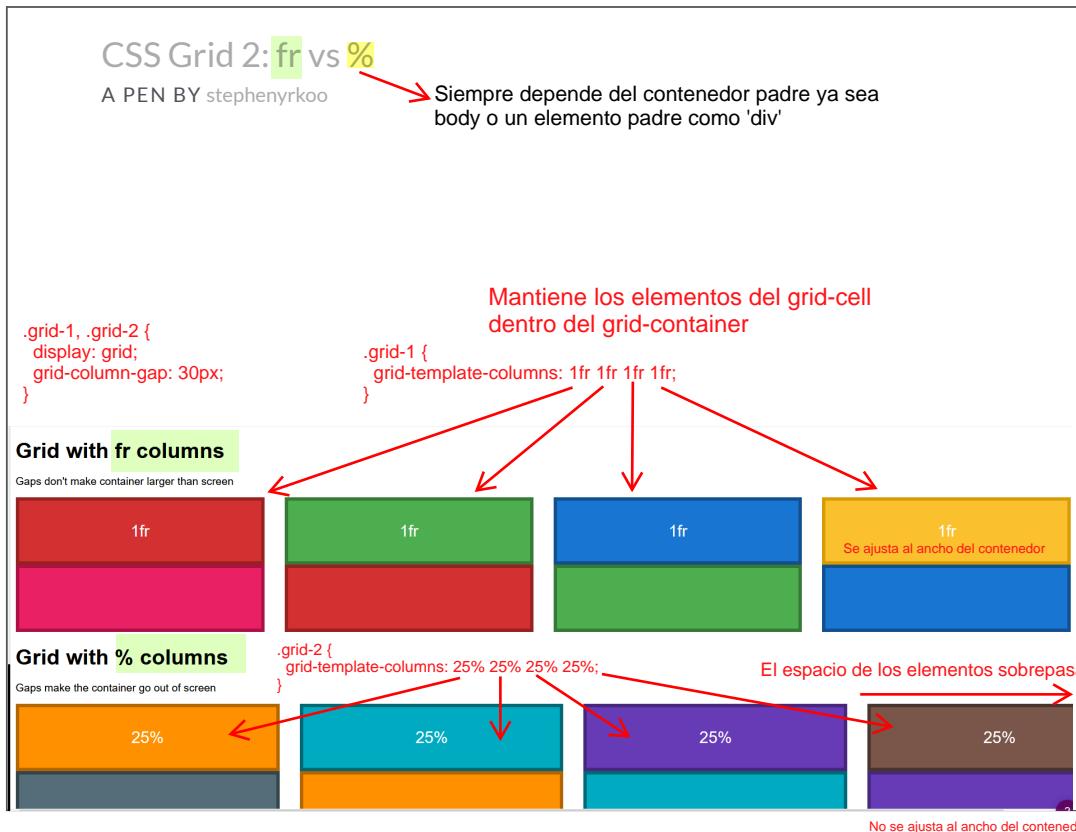
↓ Producir unwrapped - contenido no esta envuelto cuando sobrepasa el ancho del grid-item

DISEÑO
POR
UNIDADES
ABSOLUTAS

Establecer
Tamaño
Explicito
Grid-Container



- Toma una fracción del contenido disponible del espacio libre
- `fr` takes a fraction of the available container's free space Fraccionar los elementos grid-track
 - Ajusta contenidos en espacio potencialmente no causa desbordamiento
 - `fr` factors in gaps, `%` doesn't — potentially causing container overflow



1A.2 Track sizes: Content-based units

No envuelve el texto Envuelve el texto

`max-content, min-content`

Contenido se envuelve dentro del contenedor

- `min-content`: Size of item when wrapped

Contenido no se envuelve dentro del contenedor

- `max-content`: Size of item when unwrapped

wrapped - envuelto

soft-wrap-per
Lineas texto se envuelven

unwrapped - desenvuelto

El contenido no esta envuelto

Un elemento grid-items con `min-content` de espacio hará que el texto crea un salto de linea hacia abajo

Un elemento grid-items todo el tamaño del espacio disponible del elemento si es texto continuara la linea superando el borde maximo del elemento que lo contiene

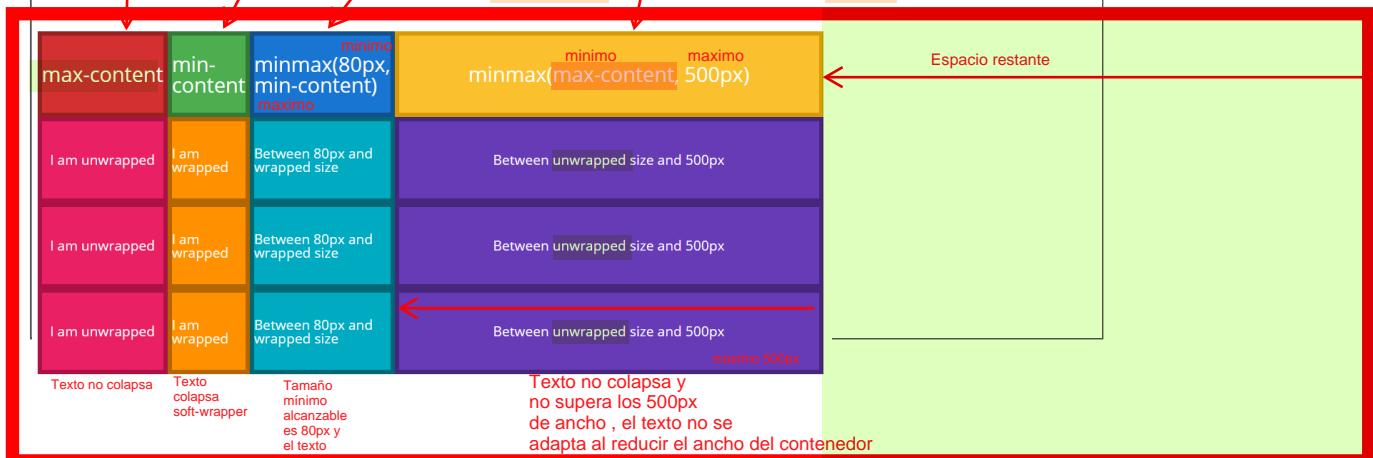
CSS Grid 3: `max-content, min-content`

coge el maximo tamaño disponible pero produce desbordamiento

no envuelve el texto envuelve el texto

IE	Edge	Firefox	Chrome	Safari	Opera
12-18	3-65	2	4-21	11-10.1	10-43
6-10	79	66-72	46-79	11-12.1	44-65
11	80	73	80	13	66
	74-75	81-83	TP		

```
.grid {
  display: grid;
  grid-template-columns: max-content min-content minmax(80px, min-content) minmax(max-content, 500px);
}
```



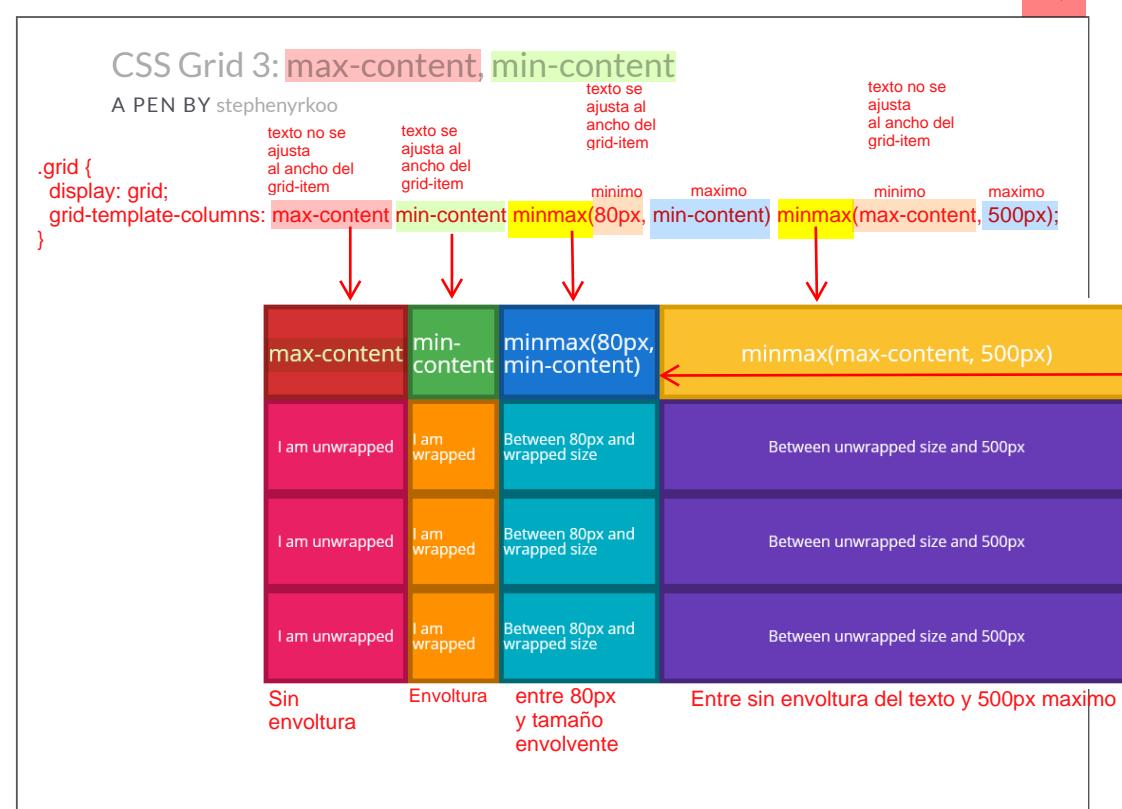
Tamaños Franjas Valores flexibles 1A.3 Track sizes: Flexible values

`min-content, max-content`
`minmax(), auto, fit-content()` ARGUMENTOS

- `minmax()` sets a minimum and maximum size for a track `minmax(minimo,maximo)`

grid-track: columns

grid-track: row



`minmax(min-content, max-content) 1fr;`

auto

si es usado junto a 'fr'

- `auto` acts like:
 - (a) `minmax(min-content, max-content)` if used alongside `fr`.

Maximo tamaño $\text{Max size} == \text{unwrapped content size}$

(b) `minmax(min-content, 1fr)` otherwise. $\text{Max size} == \text{free space}$ $\text{auto} \leftrightarrow \text{minmax(min-content, 1fr)}$

- En resumen `auto` toma el espacio a menos que exista `fr`
- In short, `auto` takes up space unless `fr` exists

todo eso es un grid-item

Produce desbordamiento del texto y contenido cuando supera el limite del grid-items

CSS Grid 4: auto

A PEN BY stephenyorkoo

fit-content(350px) - No crece más de 350px y
ajustaría el texto 'wrapper'
en cada reducción 'soft-wrapper'

```
.grid-1,  
.grid-2 {  
  display: grid;  
  grid-template-columns: fit-content(350px) fit-content(350px) 1fr;  
  grid-template-columns: minmax(min-content, max-content) minmax(min-content, max-content) 1fr;  
  grid-template-columns: auto auto 1fr;  
}
```

auto

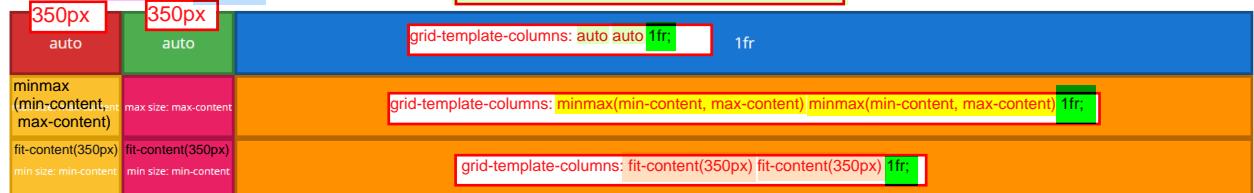
```
.grid-2 {  
  display: grid;  
  grid-template-columns: fit-content(350px) fit-content(350px) 1fr;  
  grid-template-columns: minmax(min-content, max-content) minmax(min-content, max-content) 1fr;  
  grid-template-columns: auto auto;  
}
```

auto

Grid with auto & fr tracks

auto usando fr

OCUPA TODO EL ANCHO DEL GRID-CONTAINER

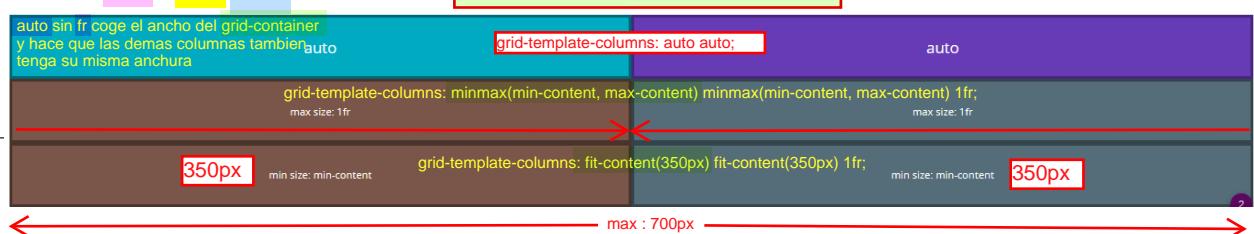


Se produce soft-wrapper

Grid with auto & no fr tracks

auto sin usar fr

OCUPA TODO EL ANCHO DEL GRID-CONTAINER



Se produce soft-wrapper

en la presencia y ausencia de bandas de fr
auto in the presence & absence of fr tracks

- **Actua como 'auto'**
- **fit-content(argument)** **actua como** **minmax(auto, the smaller of max-content or argument)**
argument → **fit-content(argu) == minmax(auto, max-content o argumento)**
- **Remember:** As **auto** **actua differently in the presence of fr**, so too will **fit-content()**

CSS Grid 5: fit-content()

A PEN BY stephenyorkoo

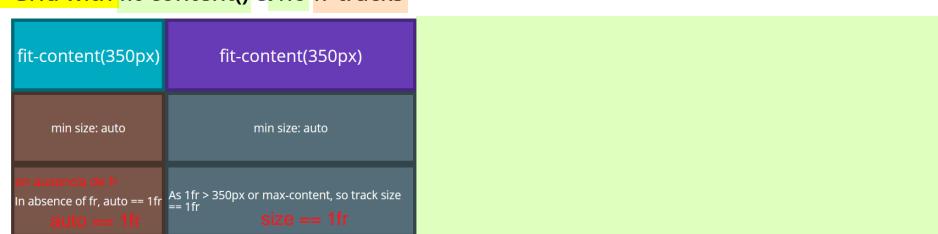
fit-content(350px) fit-content(350px) y 1fr --> actua de diferente manera

Como texto < 350px
max size:max-content



auto 1fr --> actua de diferente manera

Grid with fit-content() & no fr tracks

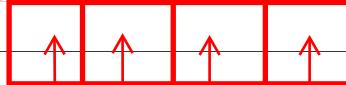


Run Pen

1A.4 Repeated tracks

Los elementos items se expanden ocupando todo el ancho incluso de las columnas items que no tienen el ancho definido
Los items se ajustan al tamaño del grid y si hay espacio sobrante ocupado por los items de las columnas sobrantes no lo ocupa

`repeat(int|auto-fit|auto-fill, track-size)`



en vez de

- Instead of `grid-template-columns: 200px 400px 400px 400px`, we can use `grid-template-columns: 200px repeat(3, 400px)`

repetir banda hasta el contenedor este lleno usar

- To repeat tracks until the container is full, use `auto-fit` or `auto-fill`
 - The only difference between `auto-fit` and `auto-fill` occurs when:
 - (A) Using `repeat(auto-fit|auto-fill, minmax(_, _fr))` && la cuadricula es bastante amplia para ajustar todo los elementos en una simple fila mas
 - (B) The grid is wide enough to fit all items into a single row plus more
 - ... `auto-fit` will spread out existing tracks; tratará comprimir dentro de bandas vacías
 - ... `auto-fill` will try to squeeze in empty tracks

CSS Grid 6: `repeat(auto-fit|auto-fill, _)`

A PEN BY stephenyrkoo

```
container: auto-fit {
  grid-template-columns: repeat(auto-fit, minmax(120px, 1fr));
}
container: auto-fill {
  grid-template-columns: repeat(auto-fit, minmax(120px, 1fr));
}
```

Run Pen

Etiquetando las líneas grid

1A.5 Labelling grid lines

Puedes nombrar líneas cuadriculas para hacer más facil la posicion de los elementos

- You can name grid lines to make it easy to position items by referring to a line's name
- Nombre líneas son actualmente no amigables SASS no las usen con los sass
- Note: Line names are currently not Sass-friendly, so don't use them in sass/scss files just yet

1º Nombre de la columna/fila
2º Unidad de distancia

```
container      nombre tamaño nombre tamaño nombre tamaño
grid-template: [row-line-a] 1fr [row-line-b] 500px [row-line-c] 4fr [row-line-d]
               / [col-line-a] 2fr [col-line-b] repeat(auto-fit,
               [col-line-c] 3fr) [col-line-d];
```

	col-line-a	col-line-b	col-line-c	col-line-c	col-line-d
row-line-a	1fr / 2fr	3fr	3fr	3fr	
row-line-b	500px				
row-line-c					
row-line-d	4fr				

1B. Create an implicit grid

Establecemos el ancho de los CONTENIDOS de los Grid-items

- Extra tracks are created for item overflow (excess items that don't fit in your explicit grid, or if you didn't make an explicit grid)
o si no hiciste un explicito grid

container

```
grid
|- grid-auto-rows: |minmax(max-content, 80vmax)|auto|fit-content(400px)|1fr;
|- grid-auto-columns: |minmax(max-content, 80vmax)|auto|fit-content(400px)|1fr;
```

 **grid-auto-rows** automatically creates rows when:
elemento desborda
(A) item overflow + **grid-auto-flow: row** (default) OR
manualmente colocas un elemento sobre una fila debajo del grid explicito
(B) you manually place an item on a row beyond the explicit grid

 **grid-auto-columns** automatically creates columns when:
elemento desborda
(A) item overflow + **grid-auto-flow: column** (see: 2A. Auto item placement below) OR
manualmente colocas un elemento sobre una columna mas alla del grid explicito
(B) you manually place an item on a column beyond the explicit grid

- Puedes usar todas las unidades disponibles dentro del grid explicito excepto
You can use all units available in the explicit grid, except **repeat()**:

fr, %, px, rem, em, vh, vw, vmin, vmax, minmax(), auto, fit-content()

la secuencia del tamaño de las bandas establecen dentro

- The sequence of track sizes set in **grid-auto-rows** and **grid-auto-columns** will be looped through until all necessary tracks are created.

CSS Grid 7: grid-auto-rows, grid-auto-columns

A PEN BY stephenyrkoo

Grid with grid-auto-rows

grid-auto-rows requires grid-auto-flow: row (default) to create rows for auto-placed items.

grid-auto-rows: 50px 100px 200px automatically creates rows in the sizes 50px, 100px, 200px repeatedly until all items are in the implicit grid.

1	2	3
4	5	6
7	8	9
10	11	12
13	14	15
16	17	18
19	20	

Grid with grid-auto-columns

grid-auto-columns requires you to set grid-auto-flow: column to create columns for auto-placed items.

grid-auto-columns: 50px 100px 200px automatically creates columns in the sizes 50px, 100px, 200px repeatedly until all items are in the implicit grid.

1	4	7	10	13	16	19
2	5	8	11	14	17	20
3	6	9	12	15	18	

1C. Create an explicit + implicit grid

You can use an explicit and implicit grid together:

container

```
.grid {           Filas           Columnas
  explicito grid-template: repeat(2, 200px) / repeat(3, 200px);
  implicito grid-auto-rows: 100px;
}
Despues el primer elemento 200px filas      100px filas sera creado por elementos adicionales
// After the first two 200px rows, 100px rows will be created for additional items
```

Auto Colocar elementos con Grid

2A. Auto-place items on the grid

item

order: -2; (0 default)

- grid-auto-flow: column auto places items in grid by column (from top to bottom, a la derecha de la columna then to the right column).
- grid-auto-flow: row (the default) auto places items in grid by row (from left to right, then to the row below).
- order takes items out of their original order and places them in front or behind the rest of the items. Items have order: 0 by default, so assigning an order <0 puts them in front, >0 puts them behind.

2B. Manually place items on the grid

Posicion elemento por nombre celda

2.1. Position item by cell name

tu puedes colocar los elementos dentro de las areas

- You can place items in grid areas by name.

container

```
grid
|- grid-template: grid-template-areas grid-template-rows / grid-template-columns;
```

item

```
grid-area: <template-area-name>;
```

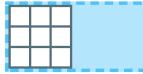
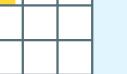
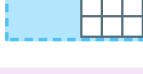
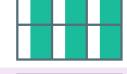
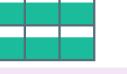
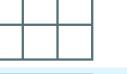
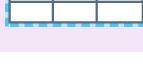
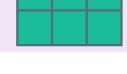
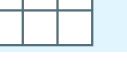
2.2 Position item by line

- You can also position items by line number

item

```
grid-area: grid-row-start / grid-column-start / grid-row-end / grid-column-end;
|- grid-row: grid-row-start / grid-row-end;
|- grid-column: grid-column-start / grid-column-end;
```

3. Align grid elements

	Container Properties				Item Properties	
	justify-content	<th>justify-items</th> <th align-items<="" th=""><th>justify-self</th><th align-self<="" th=""></th></th>	justify-items	<th>justify-self</th> <th align-self<="" th=""></th>	justify-self	
start						
center						
end						
stretch						
space-between						
space-around						
space-evenly						

CSS

Front End Development

Css Grid

Web Development

Coding

Medium

About Help Legal