

La pareja perfecta:

Flexbox
y Grid Layout

30.11.2017

Anxo Fole

Software developer en Plain Concepts



David Gómez

Software developer en Plain Concepts



Qué veremos

1. ¿Qué es Flexbox?
2. ¿Qué es Grid Layout?
3. Conclusiones

¿Qué es Flexbox?



Flexbox Layout

- Nuevo modo de diseño de CSS3.
- Permite alinear y distribuir el espacio entre elementos de un contenedor.
 - Más eficiente.
 - Tamaños desconocidos y/o dinámicos.
- El contenedor puede modificar ancho o alto de los elementos para adaptarse mejor al espacio disponible.

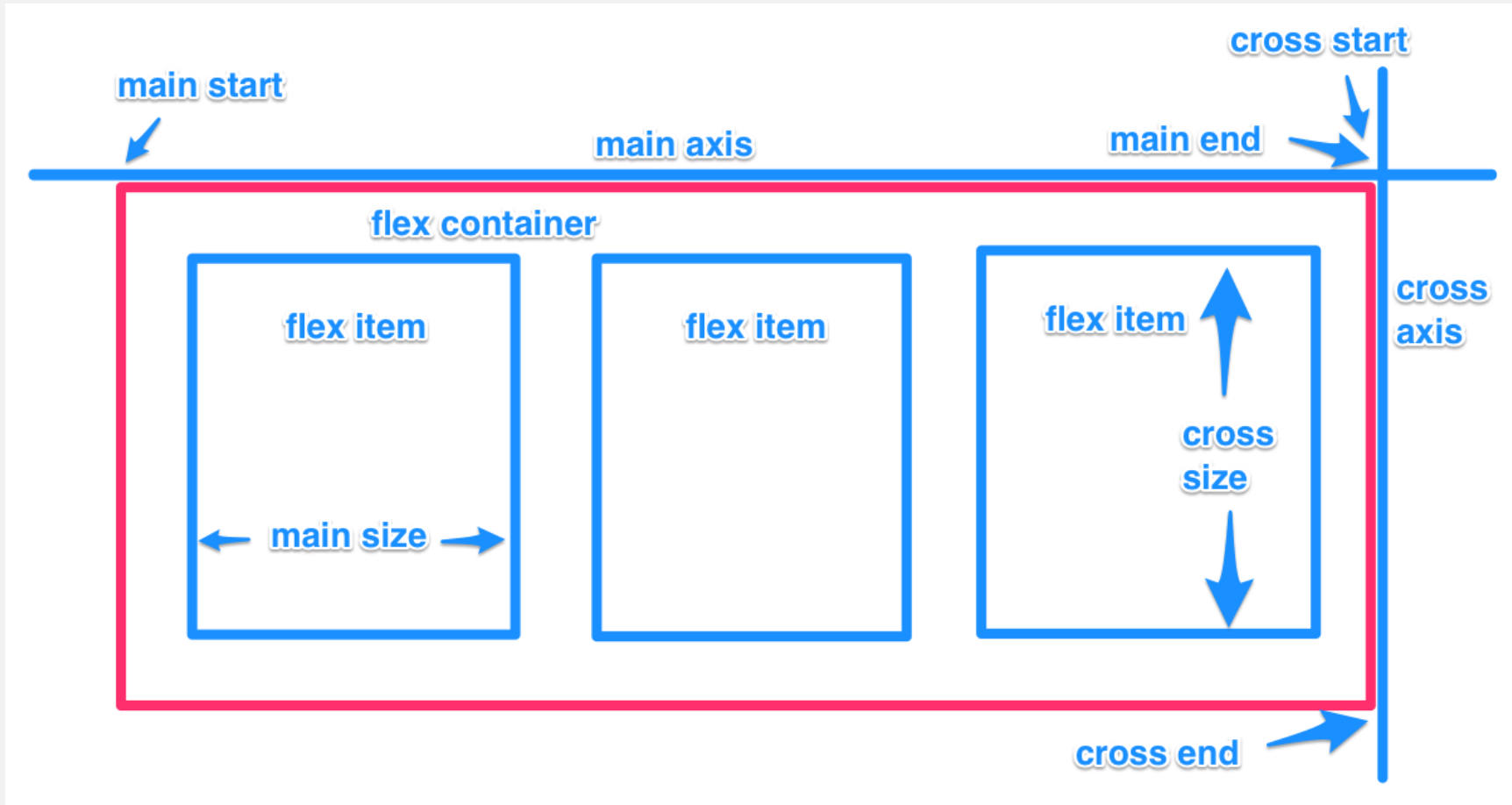
Soporte Navegadores

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49			10.2			
		55	60	10.1	47	10.3		4.4	
⁴ 11	15	56	61	11	48	11	all	56	61
	16	57	62	TP	49				
		58	63		50				
		59	64						

<https://caniuse.com/#search=flexbox>

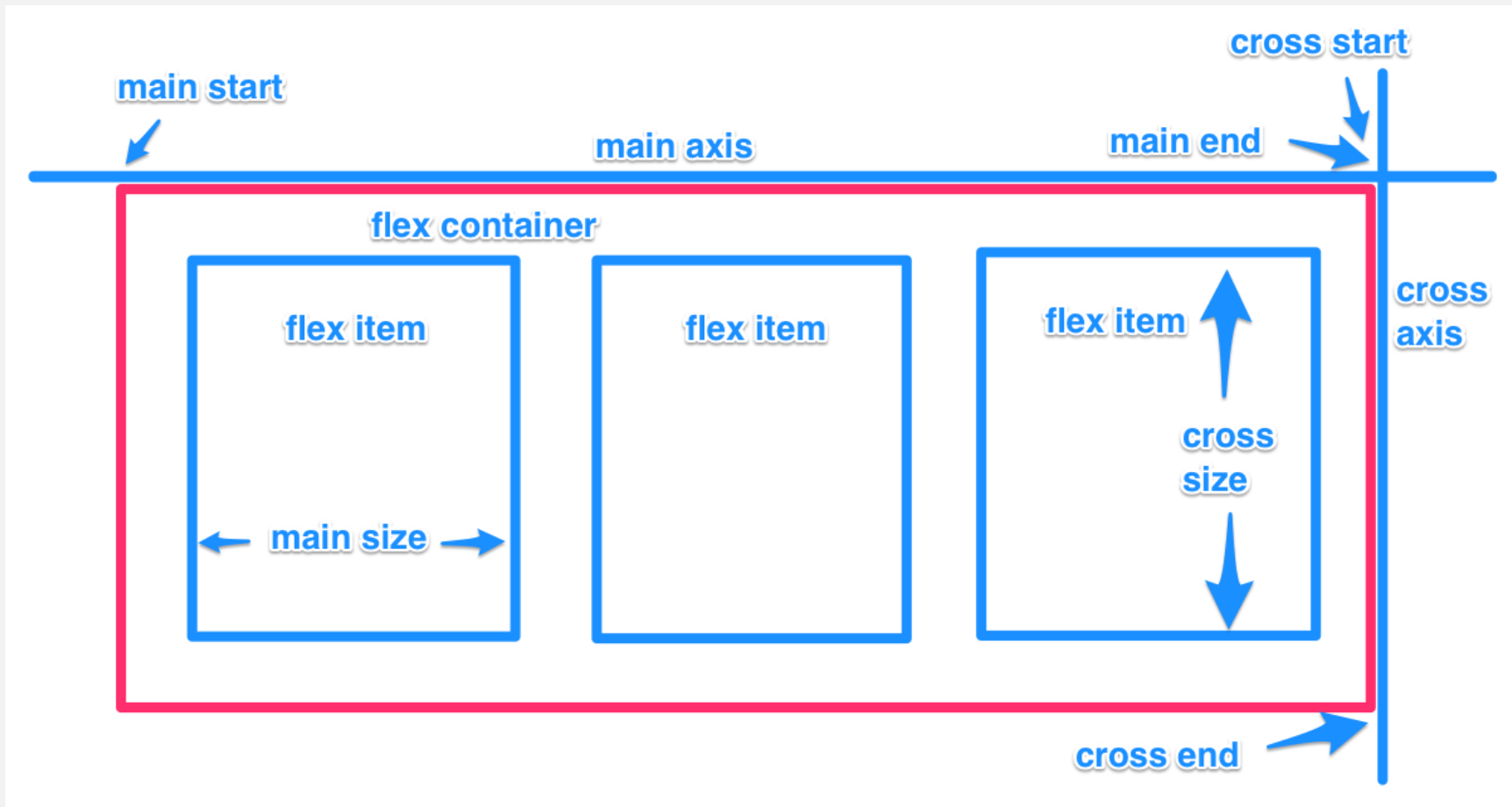
Conceptos básicos

Contenedor y elementos



Propiedades contenedor

Contenedor y elementos

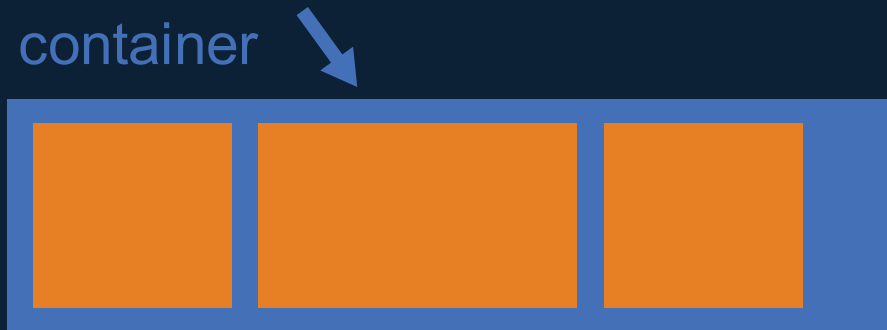


display: flex;

flex | inline-flex

Definimos el contenedor como flexible y todos sus hijos como elementos flexibles.

```
.container {  
  display: flex;  
}
```

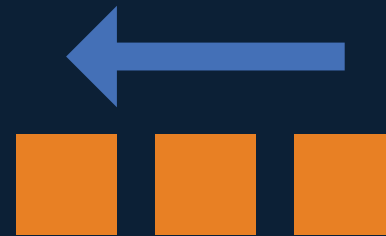
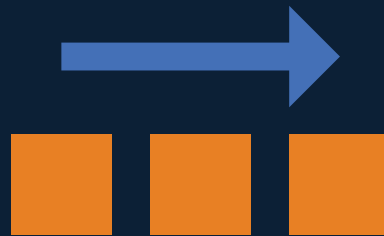


flex-direction: row;

row | column | row-reverse | column-reverse

Establece cual es el eje principal, o main axis.

```
.container {  
  flex-direction: row;  
}
```

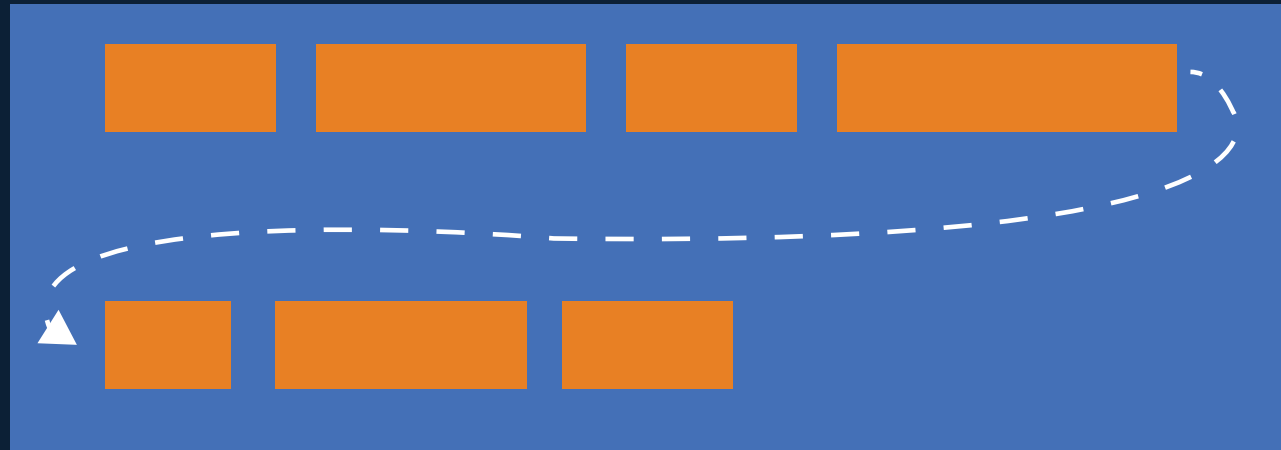


flex-wrap: nowrap;

nowrap | wrap | wrap-reverse

Establece si los elementos se disponen todos en una línea (opción por defecto) o pueden disponerse según lo necesiten.

```
.container {  
  flex-wrap: nowrap;  
}
```



flex-flow: row nowrap;

flex-direction | flex-wrap

Abreviatura de dos propiedades: flex-direction y flex-wrap.

```
.container {  
    flex-flow: row nowrap;  
}
```

justify-content: flex-start;

flex-start | flex-end | center | space-between | space-around | space-evenly

Como alineamos los elementos con respecto al main-axis.

```
.container {  
  justify-content: flex-start;  
}
```

flex-start



flex-end



center



space-between



space-around



space-evenly

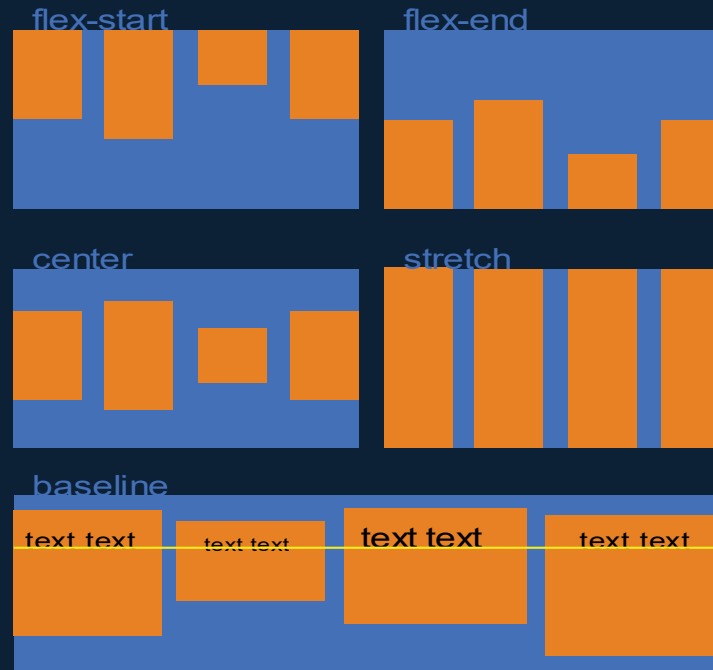


align-items: stretch;

flex-start | flex-end | center | baseline | stretch

Como alineamos los elementos con respecto al cross-axis.

```
.container {  
  align-items: stretch;  
}
```

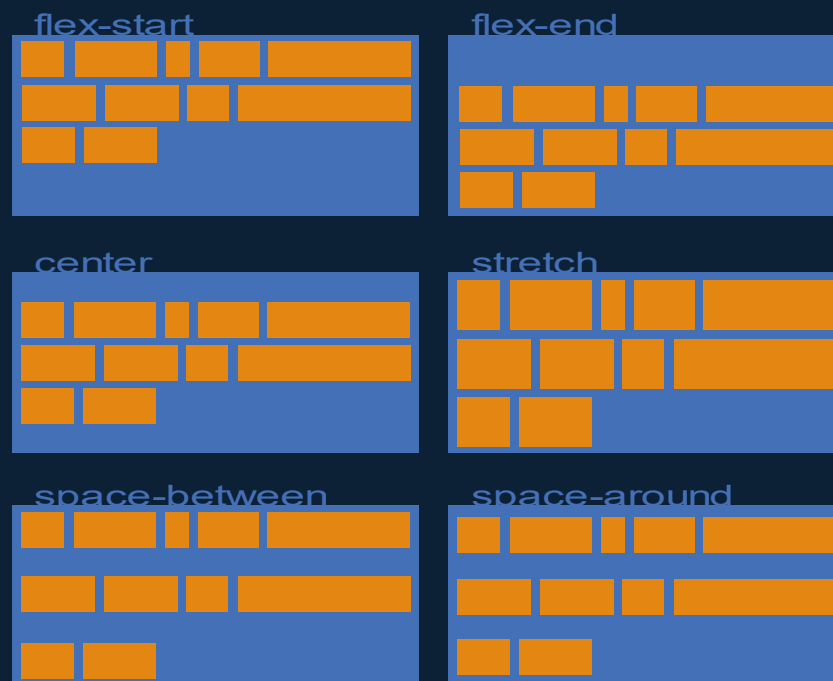


align-content: stretch;

flex-start | flex-end | center | space-between | space-around | stretch

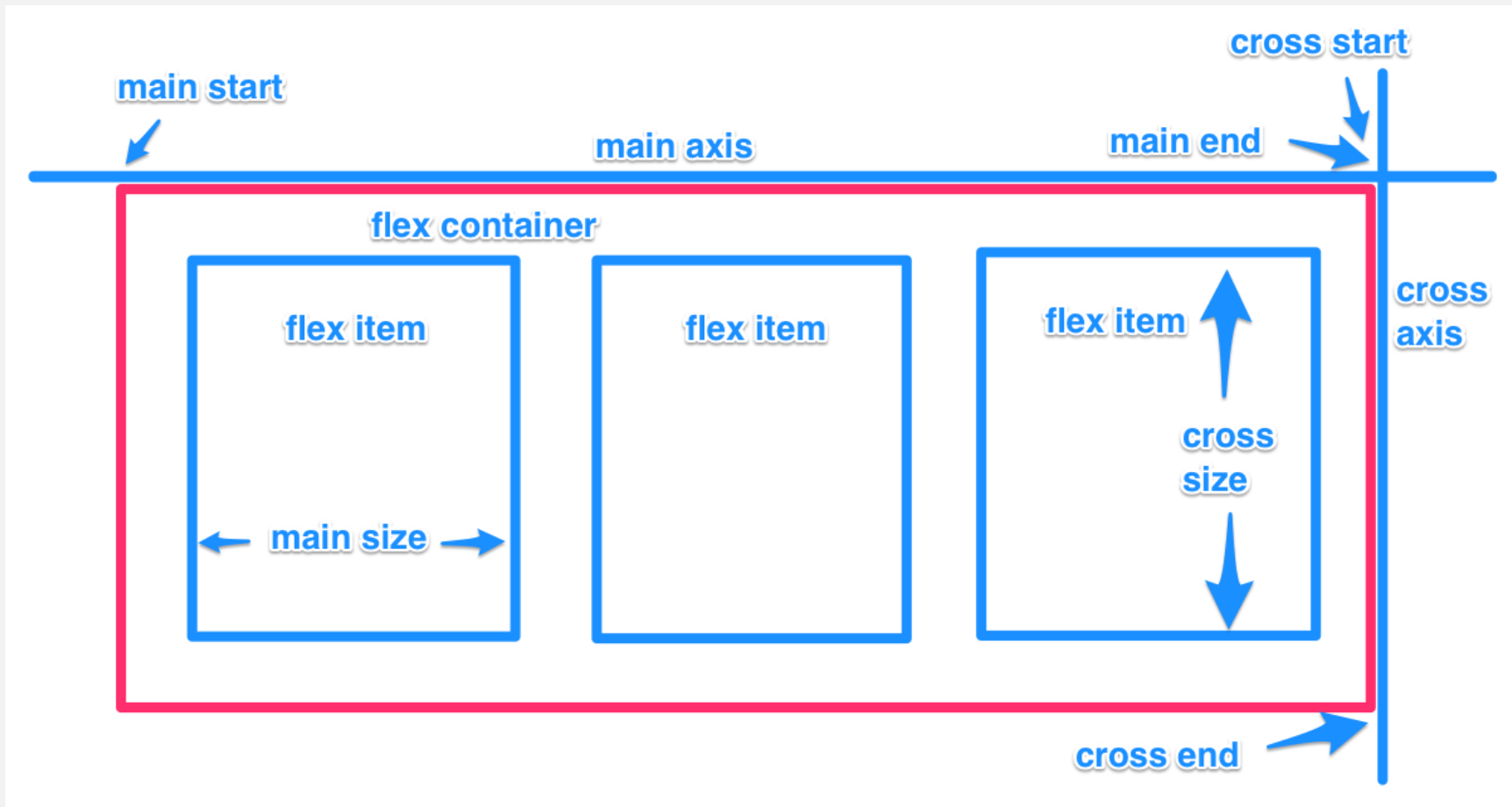
Como alineamos diferentes filas entre ellas cuando hay espacio extra en el cross-axis.
No tiene efecto si solo hay una linea de elementos flex.

```
.container {  
  align-content: stretch;  
}
```



Propiedades elementos

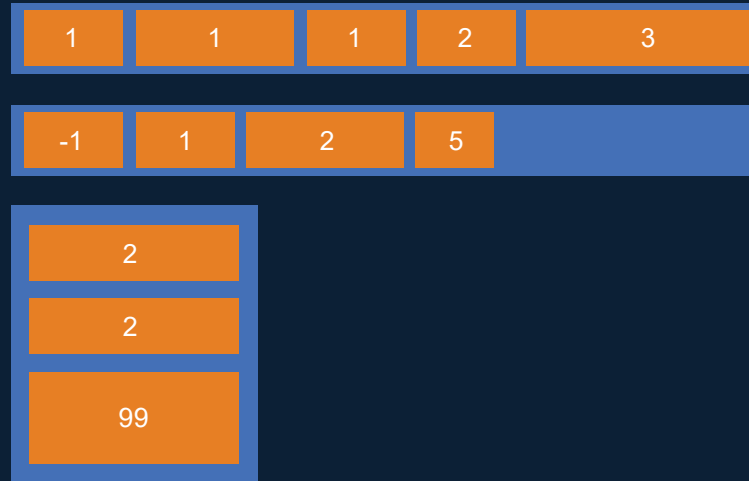
Contenedor y elementos



order: 0;

Establece el orden en el que aparecen los elementos dentro de un contenedor flex. Por defecto los elementos se colocan de acuerdo con el código.

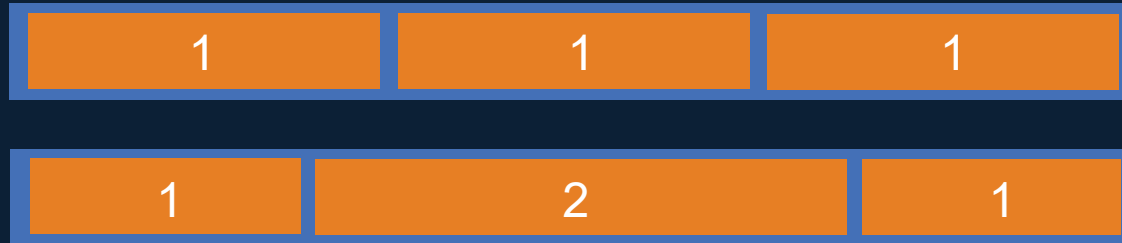
```
.item {  
  order: 1;  
}
```



flex-grow: 0;

Define la habilidad de un elemento flex de crecer si es necesario. Acepta valores sin unidades, solo positivos.

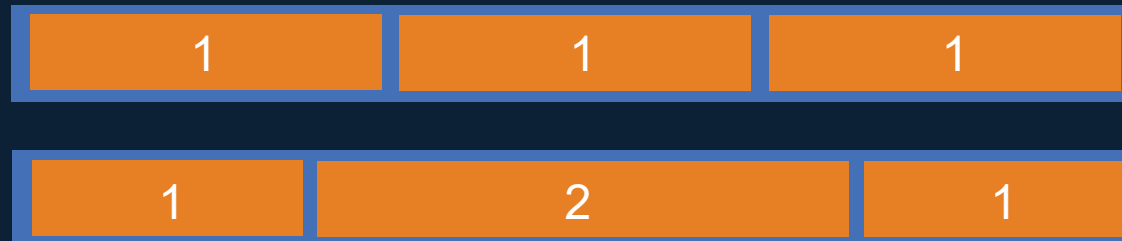
```
.item {  
  flex-grow: 1;  
}
```



flex-shrink: 0;

Define la habilidad de un elemento flex de encogerse si es necesario. Acepta valores sin unidades, solo positivos.

```
.item {  
  flex-shrink: 1;  
}
```



flex-basis: auto;

10px | auto

Define el tamaño por defecto de un elemento antes de que se distribuya el espacio disponible.

```
.item {  
    flex-basis: auto;  
}
```

flex: 0 1 auto;

Abreviatura para flex-grow, flex-shrink y flex-basis combinados.
flex-shrink y flex-basis son opcionales.

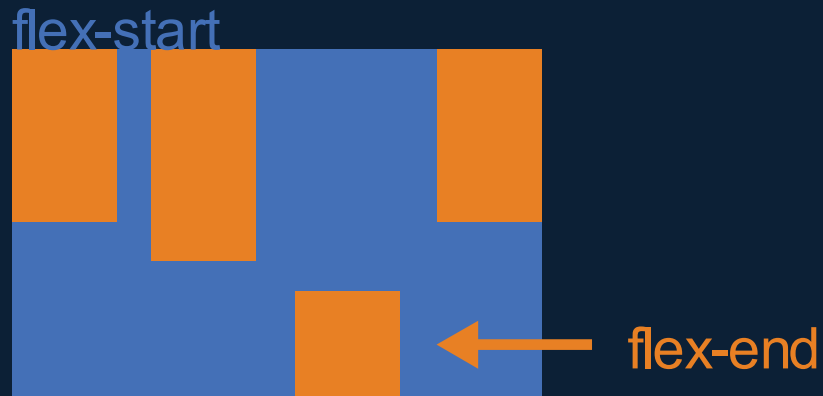
```
.item {  
  flex: 0 1 auto;  
}
```

align-self: auto;

auto | flex-start | flex-end | center | baseline | stretch

Sobreescribe la propiedad de align-items para un elemento concreto

```
.item {  
  align-self: auto;  
}
```





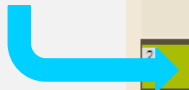
¿Qué es Grid Layout?



Grid layout

Módulo de CSS4 - Nueva especificación – Casi todos los navegadores

Soporta
versiones
antiguas



CSS Grid Layout 📄 - CR

Global69.53% + 5.62% = 75.15%

unprefixed:69.53%

Method of using a grid concept to lay out content, providing a mechanism for authors to divide available space for layout into columns and rows using a set of predictable sizing behaviors

Current alignedUsage relativeDate relativeShow all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			<div>149</div>			10.2			
	<div>215</div>	55	60	10.1	47	10.3		4.4	
<div>311</div>	16	56	61	11	48	11	all	56	61
		57	62	TP	49				
		58	63		50				
		59	64						

NotesKnown issues (2)Resources (12)Feedback

1

Enabled in Chrome through the "experimental Web Platform features" flag in chrome://flags

2

Partial support in IE refers to supporting an older version of the specification.

Grid Container

Nuestro contenedor del `grid`

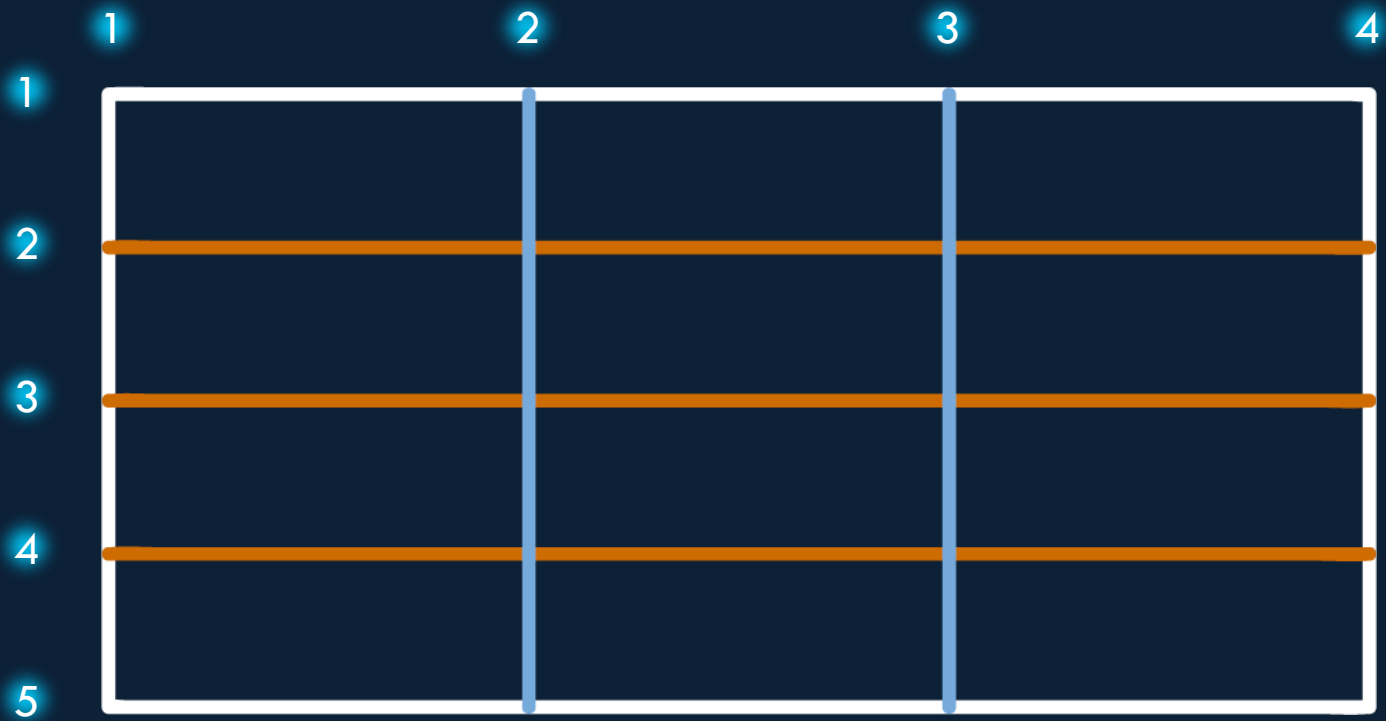
```
<main class="grid">  
  <div class="header"></div>  
  <div class="menu"></div>  
  <div class="main"></div>  
  <div class="footer"></div>  
</main>
```

Grid Item

Los hijos del grid pero no sus predecesores.

Grid Line

Las líneas divisorias entre filas y columnas (Horizontales y verticales)



Grid Track

El espacio entre dos grid lines.



Grid Cell

Unidad mínima del grid



Grid Area

El espacio entre 4 Grid Lines



display: grid;

grid

Define el comportamiento del elemento como grid.

```
<main class="grid">
  <div class="header">HEADER</div>
  <div class="menu">MENU</div>
  <div class="main">MAIN</div>
  <div class="footer">FOOTER</div>
</main>
```

```
.grid {
  display: grid;
}
```


grid-auto-flow

row | column | row dense | column dense

Cambia el sentido en el que fluyen las celdas en el grid.

Si aplicamos 'dense' y no definimos las posiciones de las celdas, se nos posicionarán llenando primero row o column.

```
.grid {  
    grid-auto-flow: column;  
}
```

grid-template

grid-template-columns | grid-template-rows | grid-template-areas

Define el tamaño de las filas y columnas y/o nombre de sus líneas.

fr

```
.grid {  
  display: grid;  
  grid-template-rows: [nombre de línea] tamaño de fila [nombre de línea] ...;  
  grid-template-columns: [nombre de línea] tamaño de columna [nombre de línea] ...;  
}  
  
repeat ([número de filas o columnas] [tamaño de las mismas])
```

HEADER

MENU

MAIN

FOOTER

grid-template-areas

“area” | . | none

De esta forma damos nombre a las celdas y le decimos en que celda se colocarán.

```
.grid {  
  grid-template-areas: “area1    .    area2”  
                      “area1 none  area2”  
                      “area1 none  area2” ;  
}
```

grid-area

Indica a que área pertenece del grid

```
.item {  
  grid-area: nombre;  
}
```

grid-column — grid-row

Posicionando ítems en el grid.

```
.item {  
  grid-column: 1 / 3;  
  grid-column-start: 1;  
  grid-column-end: 3;  
  grid-row: 2 / 3;  
  grid-row-start: 2;  
  grid-row-end: 3;  
}
```

Gap

El espacio/hueco entre celdas

```
.grid{  
  grid-gap: 20px;  
  grid-column-gap: 10px;  
  grid-row-gap: 50px;  
}
```

grid-auto-columns y grid-auto-rows

tamaño de columnas y filas dinámicas

```
.grid {  
  grid-auto-rows: 30px;  
  grid-auto-columns: 30px;  
}
```


justify-items

left | right | center | stretch

```
.grid {  
    justify-items: left;  
}
```

align-items

start | end | center | stretch

```
.grid {  
    align-items: left;  
}
```

justify-self

start | end | center | stretch

```
.item {  
  justify-self: center;  
}
```

align-self

start | end | center | stretch

```
.item {  
  align-self: center;  
}
```

justify-content

left | right | center | stretch | space-around | space-between | space-evenly

```
.grid {  
    justify-content: left;  
}
```

align-content

left | right | center | stretch | space-around | space-between | space-evenly

```
.grid {  
    align-content: left;  
}
```

Conclusiones

Flexbox vs Grid

- Ambos son válidos.
- Idealmente:
 - **Flexbox**: 1 dimensión (fila o columna).
 - **Grid Layout**: 2 dimensiones (filas y columnas).
- Debemos usarlos juntos, no como alternativa.

plain concepts 

iGRACIAS!

afole@plainconcepts.com

dgomez@plainconcepts.com



BILBAO

Ledesma 10bis , 3º Dept
48001 Bilbao. España
T. (+34) 94 6008 168



MADRID

Paseo de la Castellana 163, 10º
28046 Madrid. España
T. (+34) 91 5346 836



SEVILLA

Avenida de la innovación s/n
Edificio Renta Sevilla, 3º A
41020 Sevilla. España



BARCELONA

Av. Josep Tarradellas 10, 6º 1ª
08029 Barcelona. España
T. (+34) 93 3607 114



DUBAI

Dubai Internet City. Building 1
73030 Dubai. EAU
T. (+971) 4 551 6653



LONDON

Impact Hub Kings Cross
24B York Way, N1 9AB
London. UK



SEATTLE

1511, Third Ave
Seattle WA 98101. USA
T. (+1) 206 708 1285