

Clase 5

Diagramación

2022

Objetivos de la clase

- Analizar y construir HTML a conveniencia
- Entender tipos de display
- Aprender a utilizar grillas CSS
- Aprender de media queries

OUR SERVICES

We provide the best services

Consulting

Stars life that waters
firmament our created you.

[Read more](#)

Human Resources

Above behold our sea earth
waters in deep yielding.

[Read more](#)

Accounting

Winged isn't gathered forth
first fruitful without.

[Read more](#)

Marketing

In creepeth own for bring,
under lights, together.

[Read more](#)

Our Services

Want to get a free consultation? [Contact us.](#)

[Ejemplo en codepen](#)
[Referencia](#)

Grids CSS

Características de CSS Grid

- ✓ Diagramación en 2 dimensiones
- ✓ Permite definir una estructura de diagramación independiente del orden en que se encuentra el contenido
- ✓ En combinación con **@media-queries**, permite estructuras flexibles dependientes de las características del cliente (escritorio/móvil)
- ✓ Permite solapar áreas

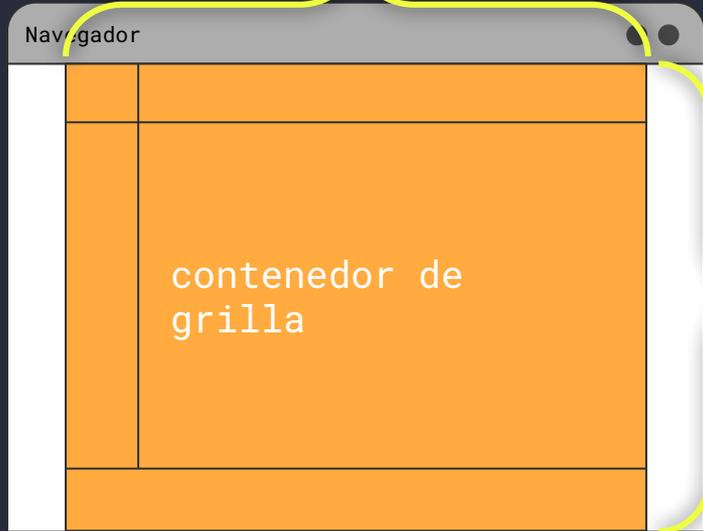
Vocabulario fundamental

Contenedor de grilla: elemento exterior, padre directo de los elementos de grilla.

Elementos de grilla: descendientes directos del contenedor de grilla.

```
<div class="container">
  <div class="item item-1">
    <div class="sub-item"></div>
  </div>
  <div class="item item-2"></div>
  <div class="item item-3"></div>
</div>
<style>
  .container {
    display: grid;
  }
</style>
```

este no!



```
<div class="container">
  <div class="logo"></div>
  <div class="navigation"></div>
  <div class="sidebar"></div>
  <main class="main"></main>
  <footer class="footer"></footer>
</div>
```

```
<style>
```

```
.container {
```

```
  display: grid;
```

```
  grid-template-columns: minmax(auto, 240px) auto;
```

```
  grid-template-rows: 100px auto 100px;
```

```
  max-width: 960px;
```

```
  margin: 0 auto;
```

```
</style>
```

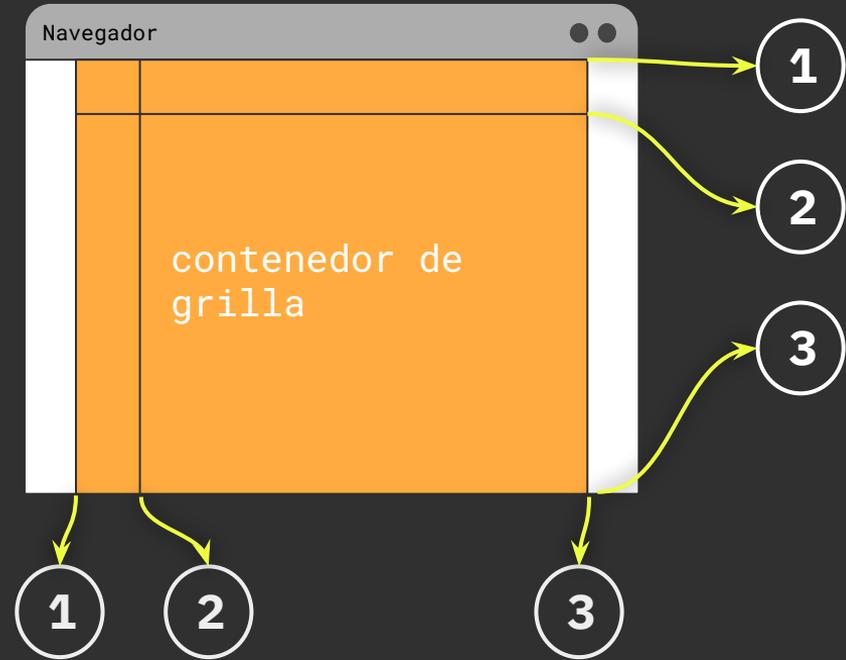
<https://codepen.io/basilio/pen/bGMEeKq> }

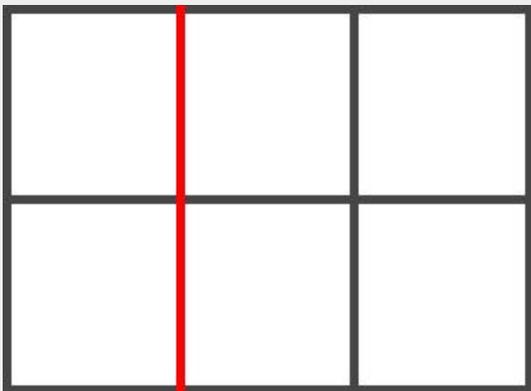
```
.container {  
  display: grid;  
  grid-template-columns: 1fr minmax( 20%, 200px ) minmax( 30%, 300px ) 2fr;  
  grid-template-rows: 100px auto 100px;  
  column-gap: 2rem;  
  row-gap: 1rem;  
  /* es lo mismo que: `gap: 1rem 2rem;` */  
}
```

Espacio entre filas

Espacio entre columnas

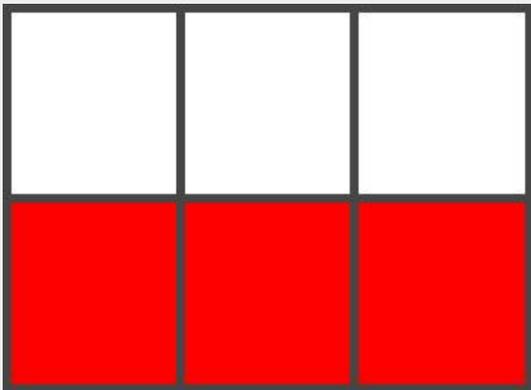
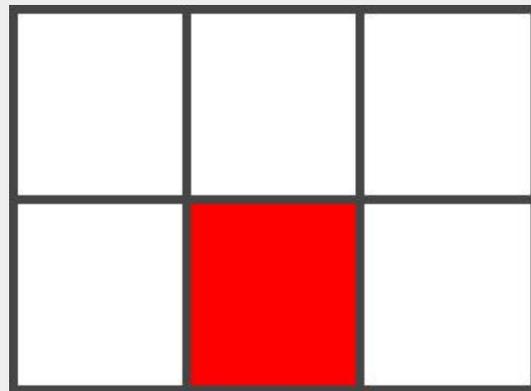
**Definir una
grilla:
identificando
líneas**





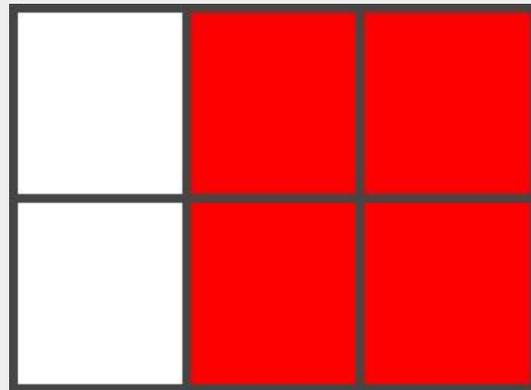
← **Línea:** línea divisoria de la grilla

Celda: área delimitada por dos filas y dos columnas adyacentes →



← **Track (pista):** área delimitada por dos líneas

Área: área delimitada por 4 líneas →



Unidad "fr"

Representa una fracción del espacio restante
en el contenedor de la grilla

Se puede
multiplicar: 3fr

Espacio que no
está asignado a un
track o un área

Referencias

- <https://css-tricks.com/snippets/css/complete-guide-grid/>
- <https://learncssgrid.com/>
- <https://gridbyexample.com/>
- https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout
- <https://www.w3.org/TR/css-grid-1/>

Ejercicio



Louis Lazaris wrote

Interactive Learning Tools For Front-End Developers

Interactive coding tools to help you learn CSS, JavaScript, SQL, React, Vim, regular expressions and Jamstack.

🔥 SEPTEMBER 2, 2021 in [CSS](#), [JavaScript](#), [Round-](#)

[Ups](#)



Louis Lazaris wrote

Interactive Learning Tools For Front-End Developers

Interactive coding tools to help you learn CSS, JavaScript, SQL, React, Vim, regular expressions and Jamstack.

🔥 SEPTEMBER 2, 2021 in [CSS](#), [JavaScript](#), [Round-Ups](#)

<https://codepen.io/basilio/pen/xxjVZWx>

Juegos para aprender grid



Posiciones

Posiciones



```
position: static;
```



```
position: relative;  
top: 2rem;  
left: 2rem;
```



```
position: absolute;  
top: 2rem;  
left: 2rem;
```



```
position: sticky;  
top: 2rem;
```

<https://developer.mozilla.org/es/docs/Web/CSS/position>

Ejercicio

Ubicar elemento con posición absoluta

<https://codepen.io/basilio/pen/YzQWaxX>

Flexbox

Flexbox

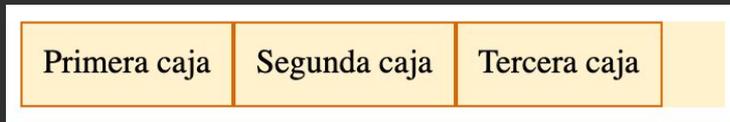
Entrega a un contenedor la habilidad para alterar el ancho o alto, y espaciados de sus elementos para ajustarse lo mejor posible al espacio disponible.

Un contenedor flexible expande sus elementos para rellenar el espacio libre, o los comprime para evitar que rebasen el área prevista.

Los contenedores flexibles funcionan en un eje, horizontal o vertical

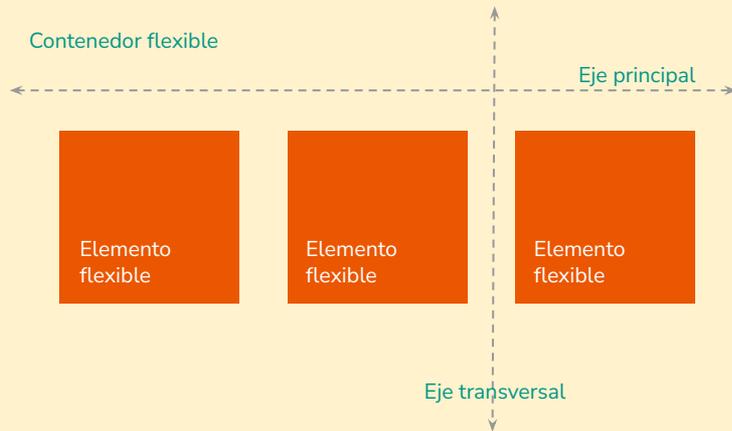
```
<div class="container">
  <div class="box">Primera caja</div>
  <div class="box">Segunda caja</div>
  <div class="box">Tercera caja</div>
</div>
```

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



<https://codepen.io/basilio/pen/ExKQdWp>

Vocabulario básico



Contenedor flexible

Elemento con propiedad `display: flex;`

Elementos flexibles

Descendientes directos de un contenedor flexible

Eje principal

Sobre el cual se suceden los elementos flexibles

Eje transversal

Perpendicular al eje principal

Líneas

Los elementos flexibles pueden quedar contenidos en una línea o utilizar varias según su ancho

Dirección, ejes y orden



`flex-direction: row;`



`flex-direction: row-reverse;`

`flex-direction: column;`



`flex-direction: column-reverse;`

Múltiples líneas

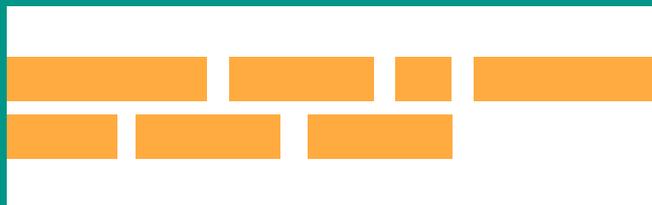
Por defecto, los elementos flexibles de un contenedor flex **se ajustan al ancho disponible**, uno al lado del otro en la misma línea.

Esto puede ser modificado a través de la propiedad `flex-wrap`, permitiendo que estos rompan la línea y se ubiquen en una nueva línea según el ancho de los elementos.

```
flex-wrap: nowrap;
```



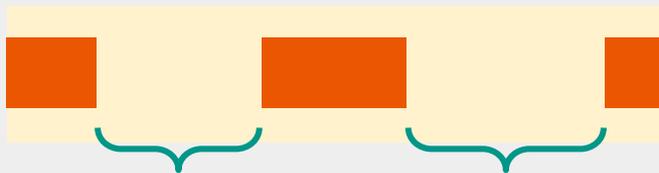
```
flex-wrap: wrap;
```



Distribución

justify-content

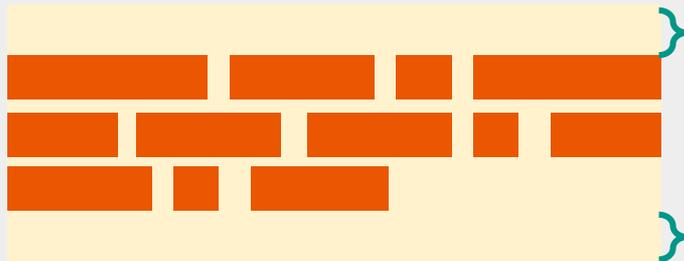
Alineación de elementos en el eje principal



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

align-content

Distribución de espacios restantes en el eje secundario

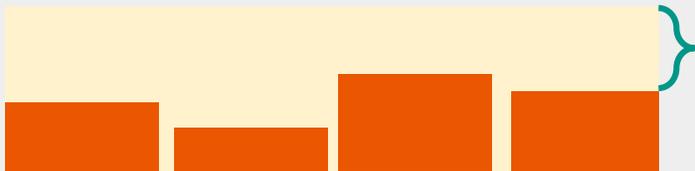


flex-start | flex-end |
center | stretch | baseline

Distribución

align-items

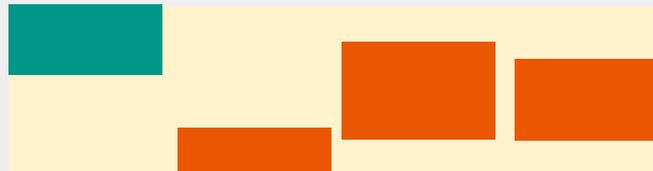
Alineación de elementos en el eje secundario



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

align-self

Alineación del propio elemento en el eje secundario - se aplica en un elemento flexible (no en el contenedor)



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

Dimensiones

Flexbox permite definir reglas de dimensiones para los elementos orientadas a adaptar los tamaños para distintas resoluciones

<https://codepen.io/basilio/pen/xyGoLZ>

`flex-basis: 25%;`



`flex-grow: 1;`



`flex-grow: 2;`



`flex-shrink: 2;`



Referencias

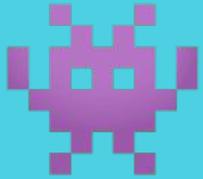
- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- https://developer.mozilla.org/es/docs/Web/CSS/CSS_Flexible_Box_Layout/Usando_las_cajas_flexibles_CSS
- <https://www.w3.org/TR/css-flexbox-1/>

Ejercicio

Menú de navegación

<https://codepen.io/basilio/pen/rNwmeZB>

Juegos para aprender flexbox



01	12 / Agosto	<i>Introducción al diseño front-end</i>
02	19 / Agosto	<i>Introducción a HTML</i>
03	26 / Agosto	<i>Introducción a CSS</i>
04	02 / Septiembre	<i>Variables, Tipografía, Colores, Íconos</i>
05	09 / Septiembre	<i>Diagramación</i>
06	23 / Septiembre	<i>Diagramación II + revisión de bitácoras</i>
07	30 / Septiembre	<i>Media Queries</i>
08	07 / Octubre	<i>Uso de dependencias Javascript</i>
09	14 / Octubre	<i>Accesibilidad + SEO</i>
10	21 / Octubre	<i>Clase práctica + apoyo en trabajos</i>
10	28 / Octubre	<i>Entrega de proyectos + cierre de asignatura</i>
	11 / Noviembre	<i>Entrega de notas</i>