



CSS Grid





**Activen las cámaras los que puedan y
pasemos asistencia**

***Aplicar la propiedad grid
para crear layouts
responsivos.***

- Unidad 1: Flexbox.
- Unidad 2: Grid.
- Unidad 3: Media Queries.
- Unidad 4: Animaciones con CSS.



Te encuentras aquí





Inicio

{desafío}
latam_



/*Crear una galería de imágenes y un layout con grid. */

/* Modificar los gaps de la grilla de un proyecto. */

/* Utilizar fracciones como unidad de medida para definir el tamaño de los ítems de una grilla. */

/* Diferenciar cuándo ocupar flexbox o grid.*/

Objetivos



Desarrollo

{desafío}
latam_



CSS Grid

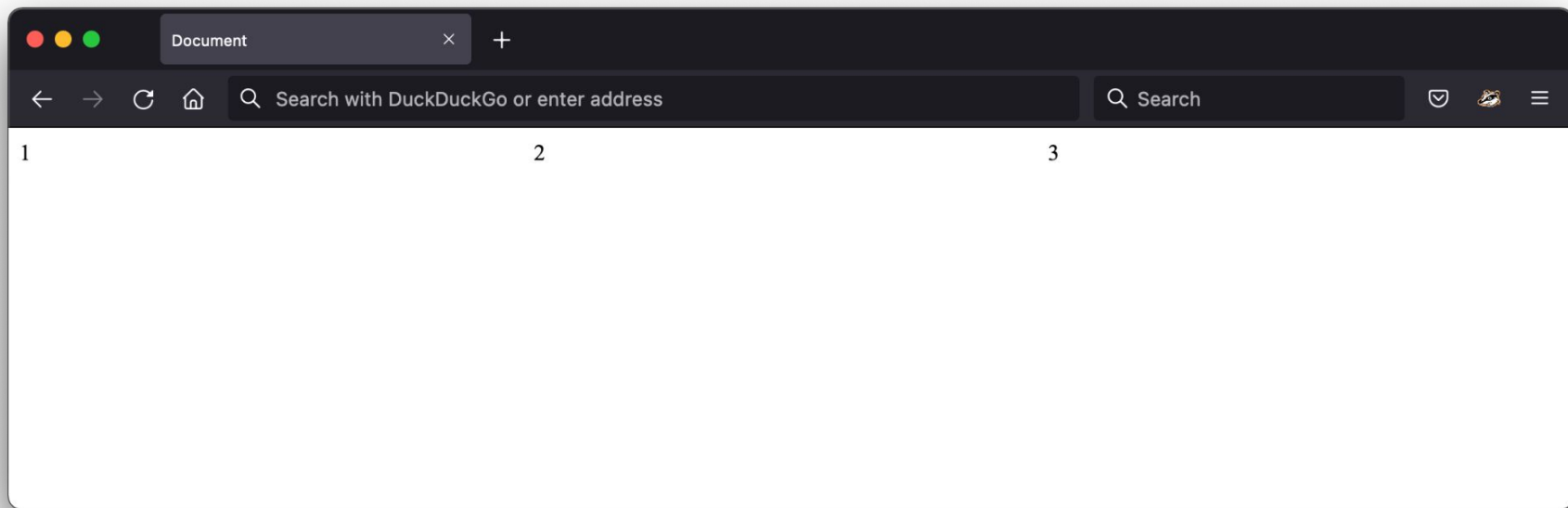
¿Qué es Grid?

CSS grid es una herramienta que nos permite crear layouts complejos de forma sencilla.



CSS Grid

Construyendo un layout de 3 columnas



CSS Grid

Grid-container y Grid-item

```
1 <div class="grid-container">
2   <div class="grid-item">
3     1
4   </div>
5   <div class="grid-item">
6     2
7   </div>
8   <div class="grid-item">
9     3
10  </div>
11 </div>
```

```
1 .grid-container{
2   display: grid;
3   grid-template-columns: auto auto auto;
4 }
5 .grid-item{
6 }
```



3 columnas de tamaño automático

Ejercicio

- Modifica el CSS anterior para tener solo 2 columnas.
- ¿Qué sucede con el elemento sobrante?

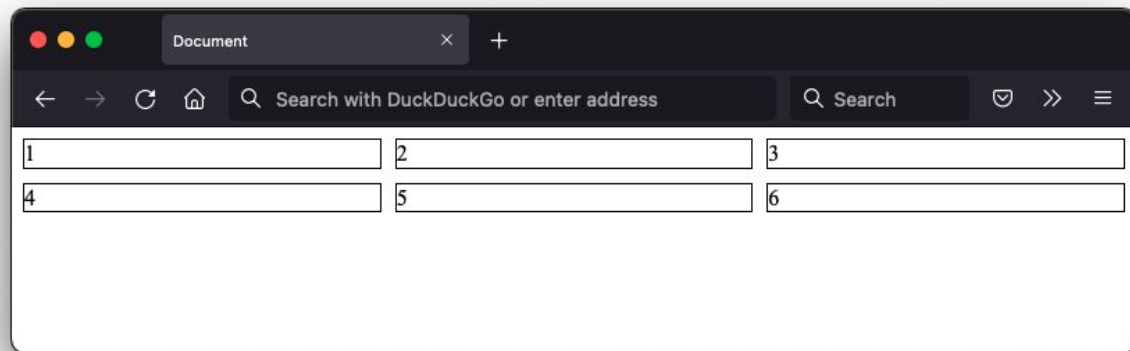
Ejercicio

¡Manos al teclado!



CSS Grid

Gaps



- En esta ocasión dejaremos 3 columnas.
- Debemos tener al menos 6 elementos dentro del grid-container.
- Agregaremos border a los elementos.
- Agregaremos gap: 10px al grid-container.

CSS Grid

Gaps

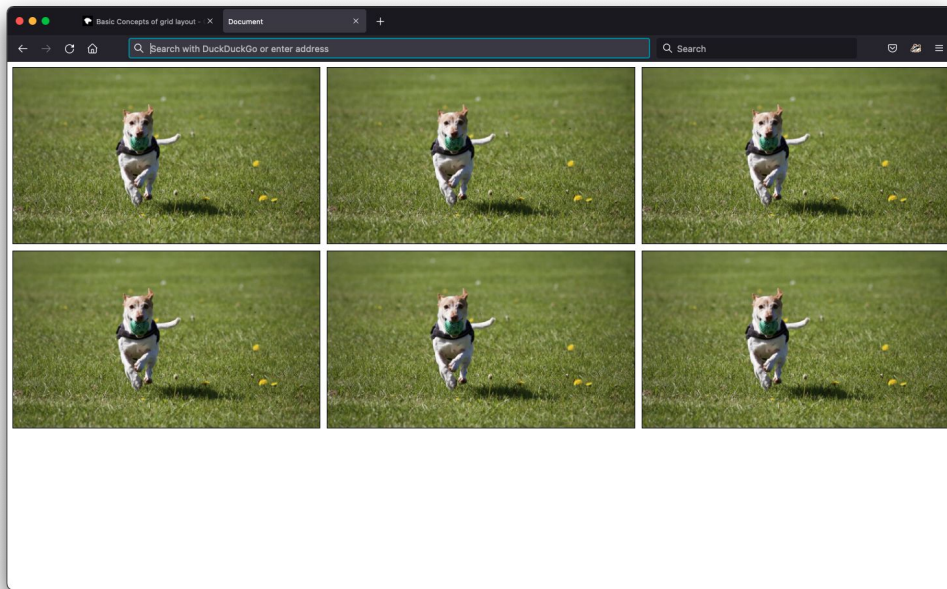
La propiedad gap está compuesta de otras dos propiedades:

- column-gap y row-gap, las que podemos modificar por separado si lo necesitamos.



CSS Grid

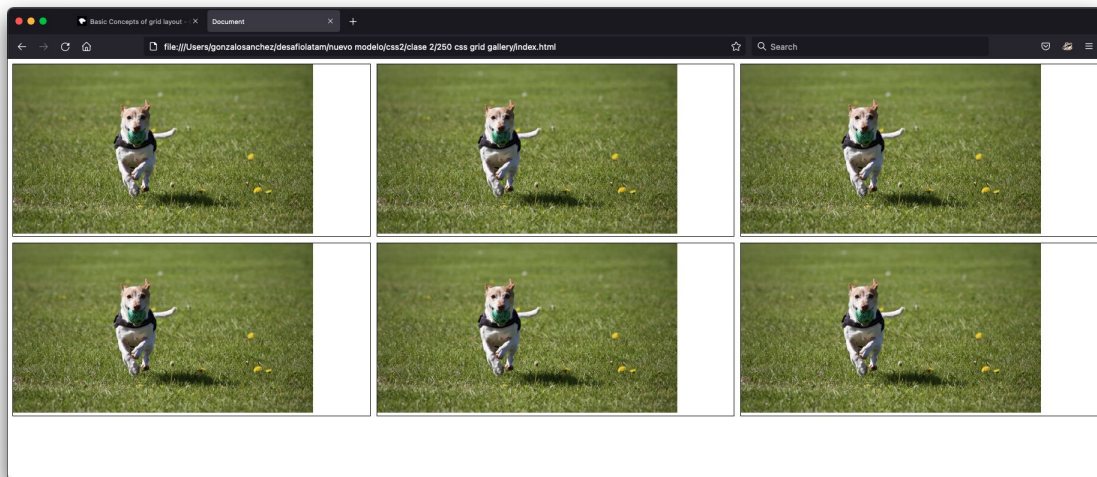
Utilizando las columnas para crear una galería de imágenes



Utilizando el grid-container con lo aprendido anteriormente, podemos crear una galería de imágenes.

CSS Grid

Ajustando el tamaño de la imagen



Dentro de la galería, puede quedar una separación entre las imágenes dependiendo del tamaño del viewport, esto, en caso de que no ajustemos el tamaño de la imagen.

CSS Grid

Utilizando las columnas para crear una galería de imágenes



```
1 <div class="grid-container">
2   <div class="grid-item">
3     
4   </div>
5   <div class="grid-item">
6     
7   </div>
8   <div class="grid-item">
9     
10  </div>
11  <div class="grid-item">
12    
13  </div>
14  <div class="grid-item">
15    
16  </div>
17  <div class="grid-item">
18    
19  </div>
20 </div>
```



```
1 .grid-container {
2   display: grid;
3   grid-template-columns: auto auto auto;
4   gap: 10px;
5 }
6 .grid-item {
7   border: solid 1px black;
8 }
9
10 .grid-item img{
11   height: 100%;
12   width: 100%;
13 }
```



Ejercicio

- Implementa una galería de imágenes utilizando Grid layout.
- Modifica el gap del grid para obtener el resultado visual que desees.
- Modifica el ancho y alto de la imagen para obtener el resultado visual que desees.

Ejercicio

¡Manos al teclado!



*/** Crear una galería de imágenes y un layout con grid. **/* 

*/** Modificar los gaps de la grilla de un proyecto. **/* 

*/** Utilizar fracciones como unidad de medida para definir el tamaño de los ítems de una grilla. **/*

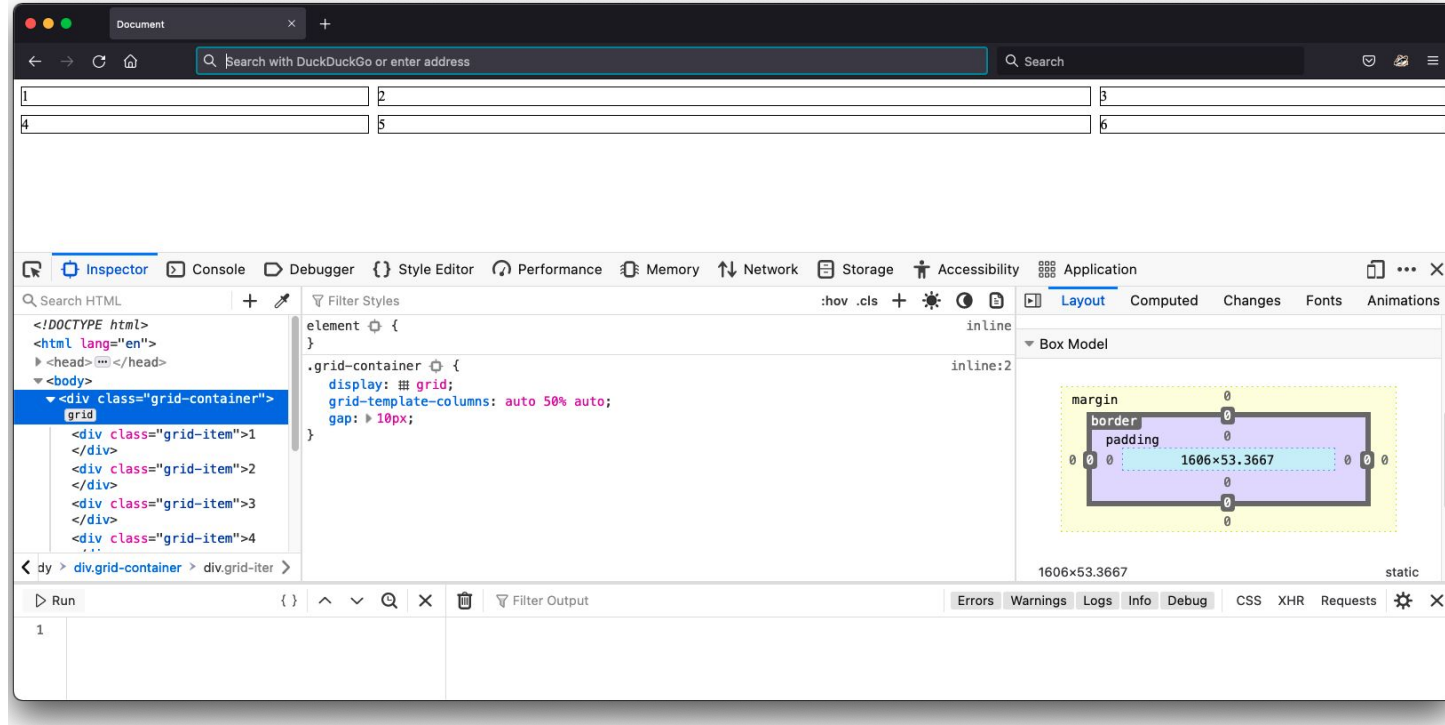
*/** Diferenciar cuándo ocupar flexbox o grid. **/*

Objetivos

CSS Grid

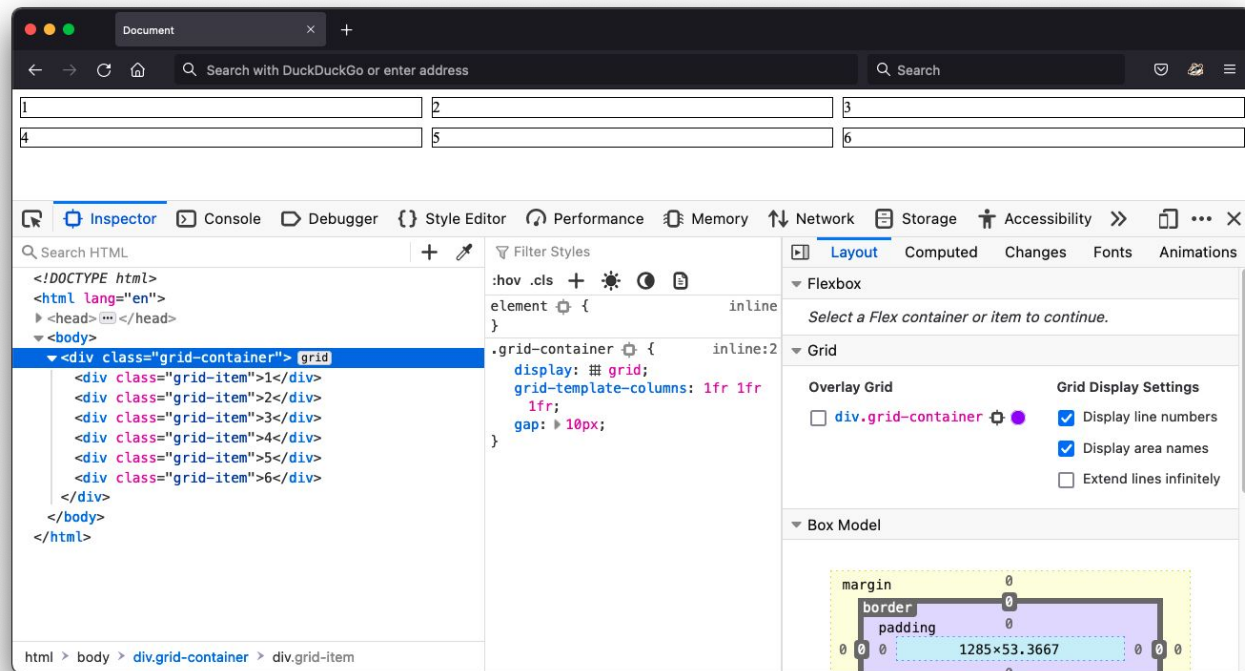
Proporciones

Hasta el momento, hemos dejado todas las columnas en tamaño automático, pero podemos especificar el tamaño de una o todas ocupando un ancho (width), dando a la columna 2 un 50% de ancho.



CSS Grid

Proporciones (fr)

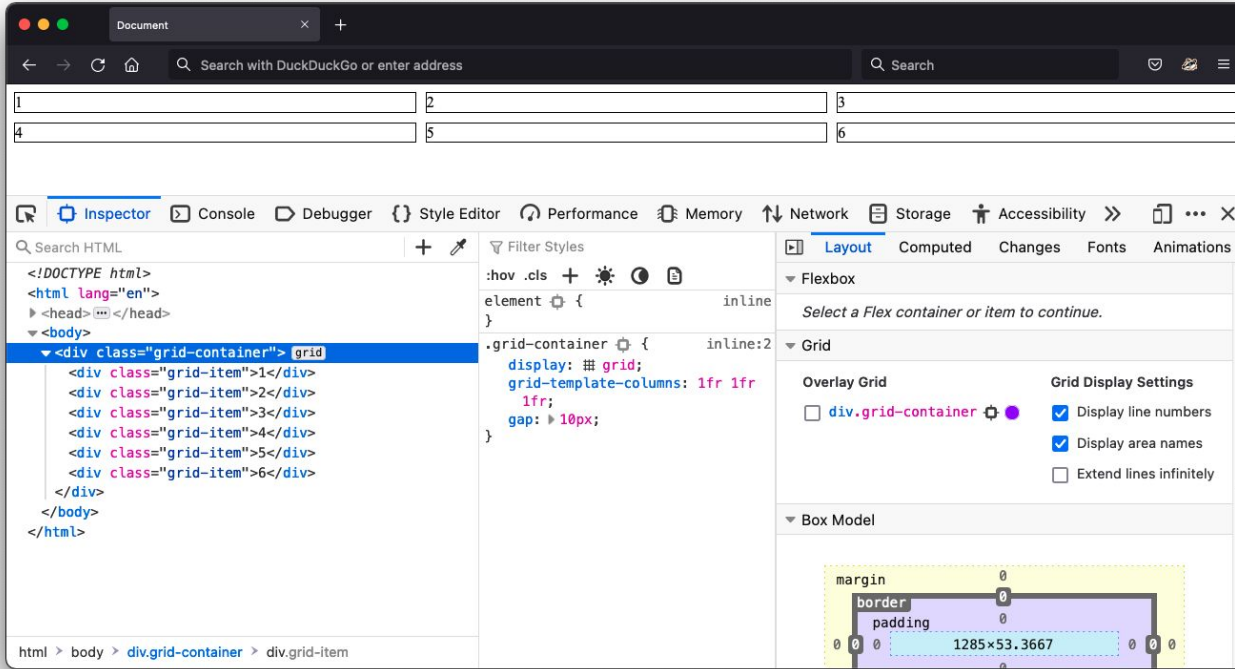


- Otra forma de asignar tamaños es con la unidad fr (fraction).
- Esta unidad divide el espacio libre en fracciones.
- Partamos cambiando el valor de grid-template-column por 1fr 1fr 1fr.

CSS Grid

Proporciones (fr)

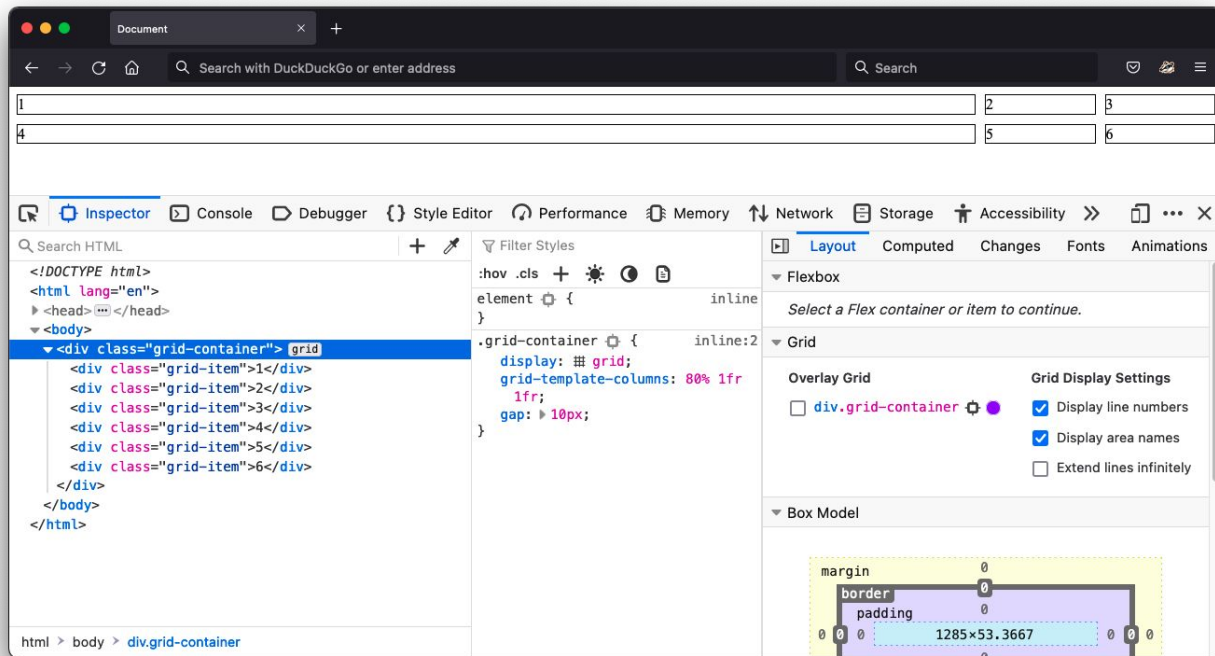
- Probemos cambiar cualquiera de las columnas por 2fr o por 3fr (o cualquier otro valor).



CSS Grid

Combinando fracciones con otras unidades

Al combinar fracciones con otras unidades, tenemos que tener en cuenta que las fracciones utilizan el espacio que va quedando libre.



Ejercicio

- Modifica la galería anterior, especificando al menos una de las columnas en porcentajes y el resto en fracciones.

Ejercicio

¡Manos al teclado!



/* Crear una galería de imágenes y un layout con grid. */ ✓

/* Modificar los gaps de la grilla de un proyecto. */ ✓

/* Utilizar fracciones como unidad de medida para definir el tamaño de los ítems de una grilla. */ ✓

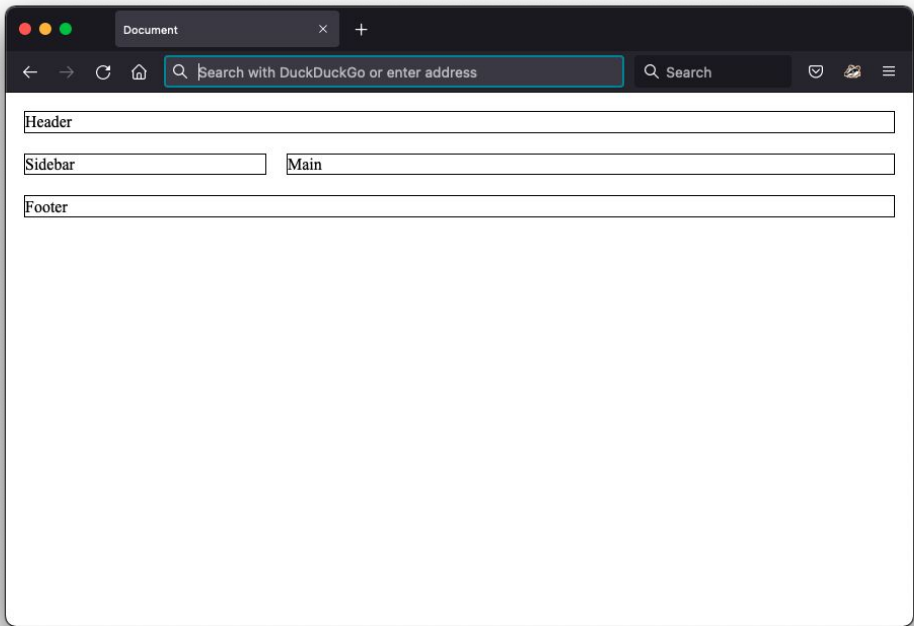
/* Diferenciar cuándo ocupar flexbox o grid. */

Objetivos

Layouts con Grid

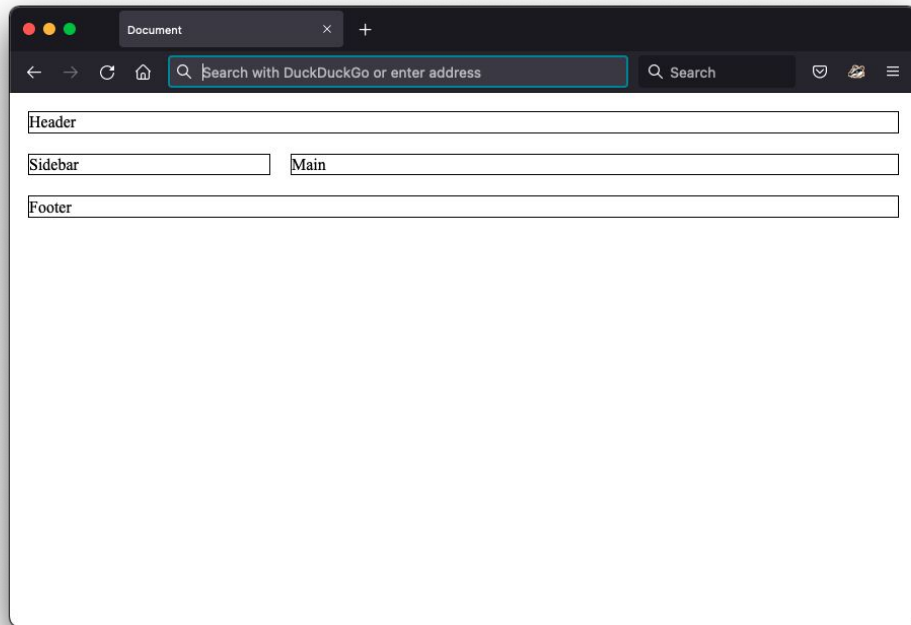
Creando layouts con CSS Grid

- Una de las mejores virtudes de CSS grid, es la capacidad de crear layouts.



Layouts con Grid

Grid area



- Para crear un layout, necesitamos tener diversos div (u otros elementos semánticos) con nombre.

Layouts con Grid

Grid area

Le asignamos un nombre a cada sección del layout con `grid-area`, luego componemos un layout utilizando estos nombres dentro del contenedor `grid`.

```
1  .header {
2    grid-area: header
3  }
4
5  .main{
6    grid-area: main
7  }
8
9  .footer{
10   grid-area: footer
11 }
12 .grid-container{
13   display: grid;
14   grid-template-areas:
15     'header'
16     'main'
17     'footer'
18 }
19 .box {
20   border: solid 1px black;
21   margin: 10px;
22 }
```

{desafío}
latam_

```
1  <div class="grid-container">
2    <div class="header box"> Header </div>
3    <div class="main box"> Main </div>
4    <div class="footer box"> Footer </div>
5  </div>
```

```
1  .header {
2    grid-area: header
3  }
4
5  .main{
6    grid-area: main
7  }
8
9  .sidebar{
10   grid-area: sidebar
11 }
12
13 .footer{
14   grid-area: footer
15 }
16 .grid-container{
17   display: grid;
18   grid-template-areas:
19     'header header'
20     'sidebar main'
21     'footer footer'
22 }
```

Layouts con Grid

Un layout con múltiples columnas

La clave está en grid-template-area, que nos permite “dibujar” con los nombres de las áreas.

```
1  <div class="grid-container">
2    <div class="header box"> Header </div>
3    <div class="sidebar box"> Sidebar </div>
4    <div class="main box"> Main </div>
5    <div class="footer box"> Footer </div>
6  </div>
```

```
1  .header {
2    grid-area: header
3  }
4
5  .main{
6    grid-area: main
7  }
8
9  .sidebar{
10   grid-area: sidebar
11 }
12
13 .footer{
14   grid-area: footer
15 }
16 .grid-container{
17   display: grid;
18   grid-template-areas:
19     'header header'
20     'sidebar main'
21     'footer footer'
22 }
```

Layouts con Grid

Un layout con múltiples columnas

Probemos agregando una columna más en grid-template área, utilizando header, main y footer (uno por línea).

```
1  <div class="grid-container">
2    <div class="header box"> Header </div>
3    <div class="sidebar box"> Sidebar </div>
4    <div class="main box"> Main </div>
5    <div class="footer box"> Footer </div>
6  </div>
```

Ejercicio

- Construir un layout utilizando grid-template-areas.
- El layout debe tener un menú arriba, una sección principal, un sidebar a la derecha y un footer abajo.

Ejercicio

¡Manos al teclado!



/* Crear una galería de imágenes y un layout con grid. */ ✓

/* Modificar los gaps de la grilla de un proyecto. */ ✓

/* Utilizar fracciones como unidad de medida para definir el tamaño de los ítems de una grilla. */ ✓

/* Diferenciar cuándo ocupar flexbox o grid. */ ✓

Objetivos



Cierre



¿Existe algún concepto que no
hayas comprendido?

Reflexionemos

- Revisar la guía que trabajarán de forma autónoma.
- Revisar en conjunto el desafío.

¿Qué sigue?



*Academia de
talentos digitales*

www.desafiolatam.com



/DesafioLatam



/DesafioLatam



/DesafioLatam



/DesafioLatam