

A graphic on the left side of the slide features four overlapping horizontal bars in purple, orange, yellow, and blue. The text 'Agencia de Aprendizaje a lo largo de la vida' is written across these bars in white. The orange bar has a white arrow pointing to the right.

Agencia de
Aprendizaje
a lo largo
de la vida

FULL STACK FRONTEND

Clase 10

CSS 6

CSS Grid



Les damos la bienvenida

Vamos a comenzar a grabar la clase

Clase 09

CSS 5 - Flexbox

- Media Queries.
- ¿Qué es Flexbox?
- Propiedades para el contenedor Flex, y los Flex items.

Clase 10

CSS 6 - Grid

- Implementación de Grid.
- Maquetado con Flex y Grid.

Clase 11

Bootstrap

- ¿Qué es un framework?
- Frameworks CSS.
- Conceptos básicos de Bootstrap.
- Componentes y funcionalidades.
- Grid Layout.
- Bootstrap Themes.

CSS Grid

¿Qué es CSS Grid?

CSS Grid es una muy buena opción para **estructurar, organizar y diseñar** los elementos de nuestro sitio web, ofreciéndonos un sistema de disposición apropiado de forma nativa en el navegador.

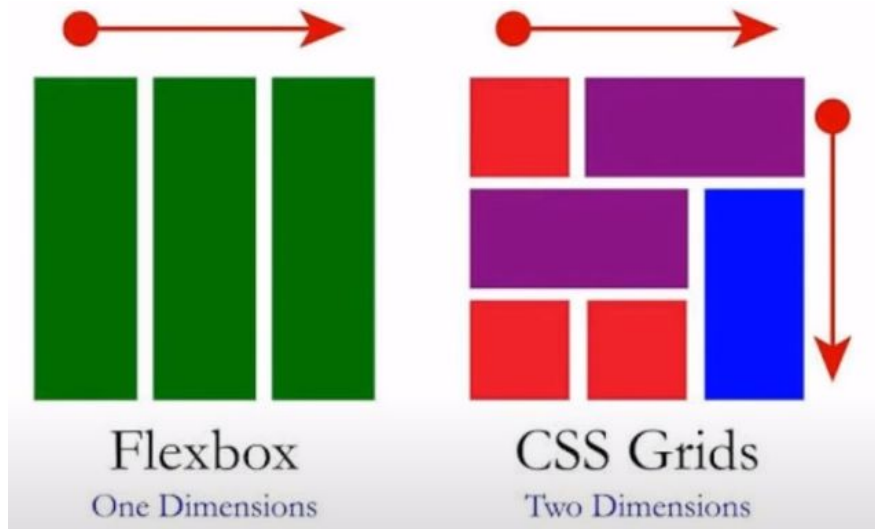
En CSS solemos utilizar múltiples propiedades para estructurar nuestros sitio web, utilizando **inline-block, float, relative** o hasta **flexbox** que sólo utiliza una dimensión.

El módulo CSS Grid entra en acción ofreciéndonos un sistema de filas y columnas **en dos dimensiones**, y total libertad de los ítems para organizar nuestros contenido a gusto.

¿Y donde queda el **Responsive Design**? Con CSS Grid podemos definir filas y columnas como también redefinirlas a nuestro gusto dentro de un media query.

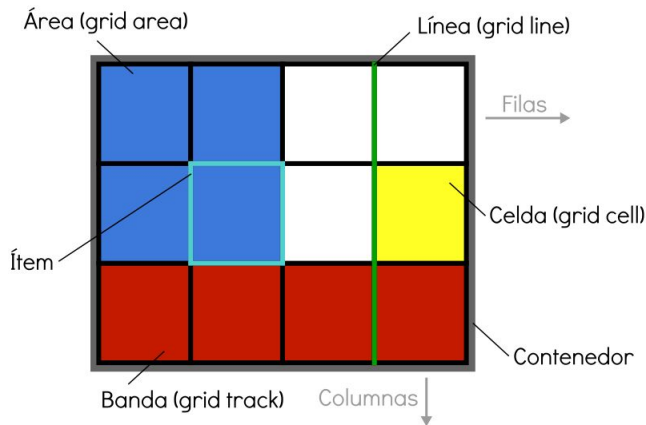
CSS Grid | ¿Flexbox vs CSS Grid?

CSS Grid puede convivir con Flexbox. No se trata de tecnologías excluyentes, sino complementarias.



CSS Grid

El módulo de diseño de CSS Grid ofrece un sistema de diseño basado en cuadrículas, con filas y columnas, lo que facilita el diseño de páginas web **sin tener que usar flotadores y posicionamiento.** [+info](#)



CSS Grid | Conceptos básicos

- **Grid Container:** es nuestro elemento “padre”, donde se asigna un `{display:grid;}` y nos permitirá colocar otras propiedades para manipular nuestro diseño.
- **Grid Item:** son los hijos directos de nuestro container. Estos los manejaremos a nuestra voluntad, nuestras filas y columnas que moveremos a nuestro gusto.
- **Grid Line:** son las líneas divisorias horizontales y verticales.
- **Grid Track:** es el espacio entre dos líneas adyacentes. Filas y columnas.
- **Grid Cell:** nuestras celdas serán el espacio entre dos filas adyacentes y 2 columnas adyacentes.
- **Grid Area:** espacio rodeado por 4 *grid lines*.

CSS Grid | Contenedor

```
<div> <!-- Elemento padre -->
  <div class="grid-item">1</div>
  <div class="grid-item">2</div>
  <div class="grid-item">3</div>
  <div class="grid-item">4</div>
  <div class="grid-item">5</div>
  <div class="grid-item">6</div>
  <div class="grid-item">7</div>
  <div class="grid-item">8</div>
  <div class="grid-item">9</div>
</div>
```

1	2	3
4	5	6
7	8	9

CSS Grid | Display

Un elemento HTML se transforma en un contenedor de grilla cuando tiene su propiedad **display** seteada en **grid** o **inline-grid**:

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

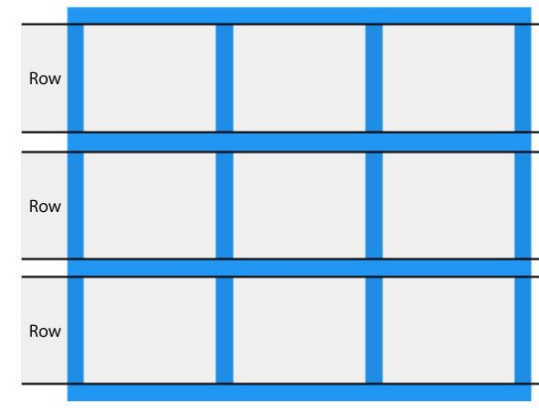
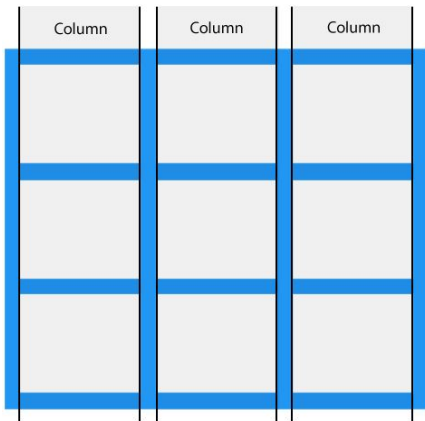
[Ejemplo](#)

```
.grid-container{  
  display: inline-grid;  
}
```

[Ejemplo](#)

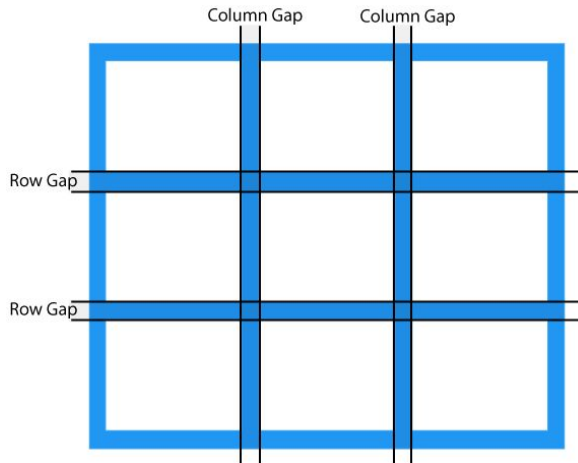
CSS Grid | Grid Items

Se pueden referenciar por fila o por columna, aunque no es la única forma.



CSS Grid | Grid Gaps

Es el espacio entre los ítems. Se pueden ajustar los tamaños de gap con las siguientes propiedades: **grid-column-gap**; **grid-row-gap** y **grid-gap**.



[Ejemplo column-gap](#)

[Ejemplo row-gap](#)

[Ejemplo grid-gap](#)

CSS Grid | Grid Lines

Para colocar un **grid-item** en un contenedor se referencian los números de línea:

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

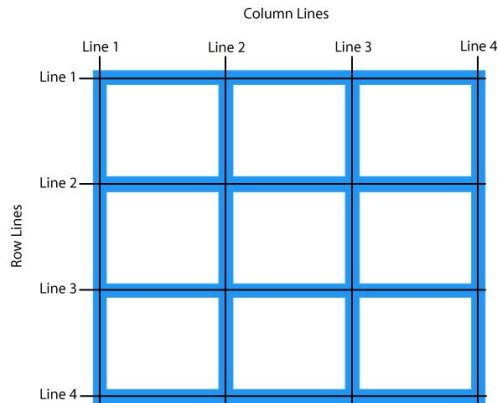
Pone un grid item en la línea de la columna 1, que finaliza en la línea de la columna 3.

[Ejemplo](#)

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

Pone un grid item en la línea de la fila 1, que finaliza en la línea de la fila 3.

[Ejemplo](#)



CSS Grid | Propiedades

Propiedad	Descripción
column-gap	Especifica el espacio entre las columnas. +info
gap	Propiedad abreviada. Espacio entre filas y entre columnas. +info
grid	Propiedad abreviada. Filas y columnas, de la cuadrícula, áreas de la cuadrícula, filas y columnas automáticas y propiedades de flujo automático de cuadrícula. +info
grid-area	Especifica un nombre para el elemento de la cuadrícula. Es una propiedad abreviada para grid-row-start, grid-column-start, grid-row-end y grid-column-end. +info
grid-auto-columns	Especifica un tamaño de columna predeterminado. +info
grid-auto-flow	Determina cómo se insertan los elementos en la cuadrícula. +info
grid-auto-rows	Especifica un tamaño de fila predeterminado. +info

CSS Grid | Propiedades (continuación)

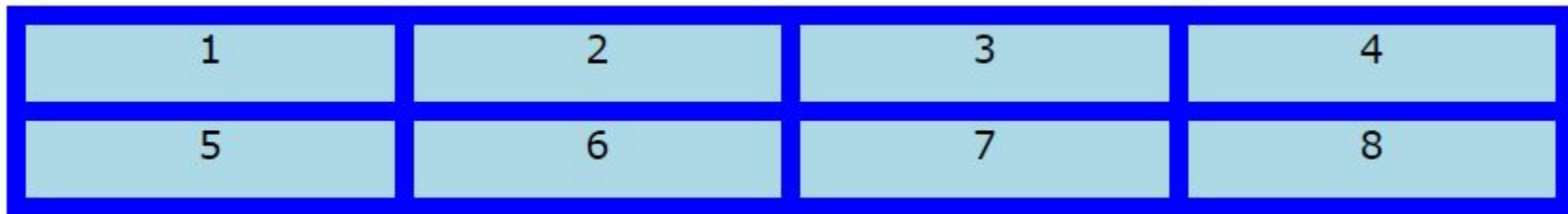
Propiedad	Descripción
grid-column	Una propiedad abreviada para las propiedades grid-column-start y grid-column-end. +info
grid-column-end	Especifica dónde termina el elemento de la cuadrícula. +info
grid-column-gap	Especifica el tamaño del espacio entre columnas. +info
grid-column-start	Especifica dónde comienza el elemento de la cuadrícula. +info
grid-gap	Una propiedad abreviada para las propiedades grid-row-gap y grid-column-gap. +info
grid-row	Una propiedad abreviada para las propiedades grid-row-start y grid-row-end. +info
grid-row-end	Especifica dónde termina el elemento de la cuadrícula. +info
grid-row-gap	Especifica el tamaño del espacio entre filas. +info

CSS Grid | Propiedades (continuación)

Propiedad	Descripción
grid-row-start	Especifica dónde comienza el elemento de la cuadrícula. +info
grid-template	Una propiedad abreviada para las propiedades de las filas de plantilla de cuadrícula, columnas de plantilla de cuadrícula y áreas de cuadrícula. +info
grid-template-areas	Especifica cómo mostrar columnas y filas, utilizando elementos de cuadrícula con nombre. +info
grid-template-columns	Especifica el tamaño de las columnas y cuántas columnas en un diseño de cuadrícula. +info
grid-template-rows	Especifica el tamaño de las filas en un diseño de cuadrícula. +info
row-gap	Especifica el espacio entre las filas de la cuadrícula. +info

CSS Grid | Grid Container

Para que un elemento HTML se comporte como un contenedor de cuadrícula, debemos establecer la propiedad **display** en **grid** (cuadrícula) o **inline-grid** (cuadrícula en línea). Los contenedores de cuadrícula consisten en elementos de cuadrícula, colocados dentro de columnas y filas.



CSS Grid | Grid Container

Este es el código que genera el cuadro de la diapositiva anterior:

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

```
.grid-container {
  width: 800px;
  display: grid;
  grid-template-columns: repeat(4, auto);
  grid-gap: 10px;
  background-color: blue;
  padding: 10px;
}
```

```
.grid-container > div {
  background-color: lightblue;
  text-align: center;
  font-size: 20px;
  height: 40px;
}
```

CSS Grid | grid-template-columns

La propiedad **grid-template-columns** define el número de columnas (y el ancho) de la cuadrícula. Se colocan los valores separados por espacios, y cada uno define el ancho de la columna respectiva. Se pueden establecer anchos en **px**, unidades relativas o **%**, aunque es recomendable utilizar la medida **fr**.

```
/*Medidas en px:*/  
grid-template-columns: 300px 200px 400px;
```

1	2	3	
4	5	6	

CSS Grid | grid-template-columns

```
/*Medidas en %:*/  
grid-template-columns: 50% 35% 15%;
```

1	2	3
4	5	6

```
/*Medida automática:*/  
grid-template-columns: auto auto auto;
```

1	2	3
4	5	6

Si una cuadrícula de 4 columnas tiene más de 4 elementos, se agrega automáticamente una nueva fila para colocar los elementos extra.

CSS Grid | grid-template-columns

Unidad fracción restante (fr):

La unidad especial de Grid **fr** (*fraction*) representa una fracción de espacio restante en el grid.

```
grid-template-columns: 0.5fr 2fr 1fr;
```

1	2	3
4	5	6

CSS Grid | grid-template-columns

La expresión repeat():

Se puede utilizar la expresión **repeat()** para indicar repetición de valores, indicando el número de veces que se repiten y el tamaño en cuestión.

La expresión a utilizar es la siguiente:

repeat([número de veces], [valor o valores]):

```
grid-template-columns: repeat(3, 1fr);
```

1	2	3
4	5	6

CSS Grid | grid-template-rows

La propiedad **grid-template-rows** define la altura de cada fila.

El valor es una lista separada por espacios, donde cada valor define el alto de la fila respectiva.

```
grid-template-rows: 80px 200px;
```

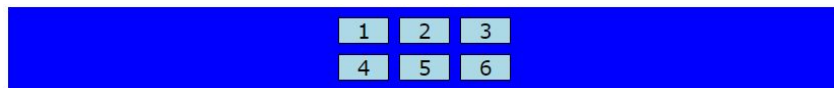
1	2	3
4	5	6

CSS Grid | Propiedad justify-content

La propiedad **justify-content** se utiliza para alinear toda la cuadrícula dentro del contenedor. El ancho total de la cuadrícula debe ser menor que el ancho del contenedor para que la propiedad tenga efecto.



space-evenly



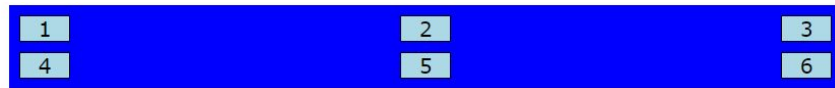
center



space-around



start



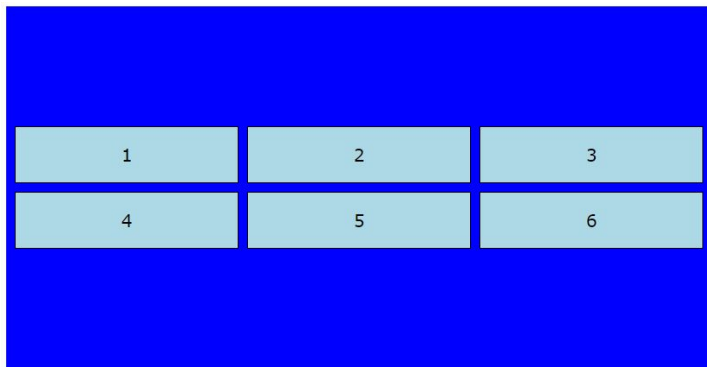
space-between



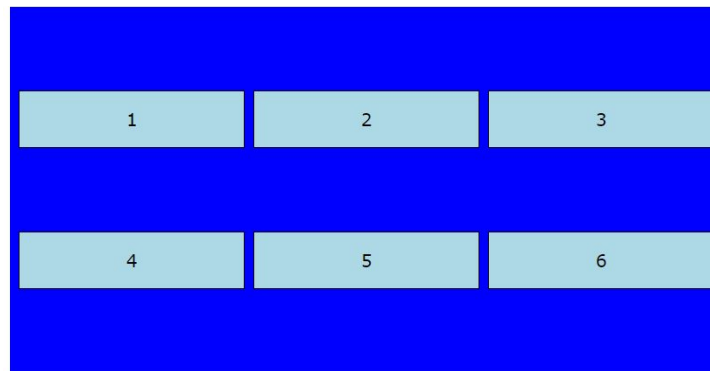
end

CSS Grid | Propiedad align-content

La propiedad **align-content** se usa para alinear verticalmente toda la cuadrícula dentro del contenedor. La altura total de la cuadrícula debe ser menor que la altura del contenedor para que tenga efecto.

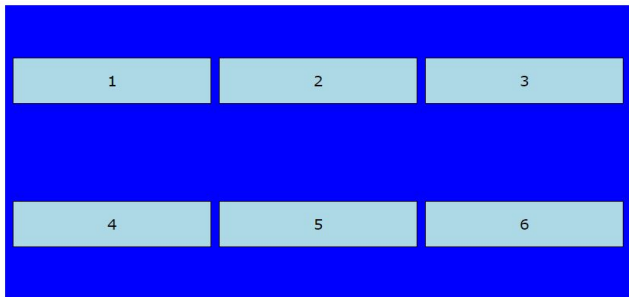


center

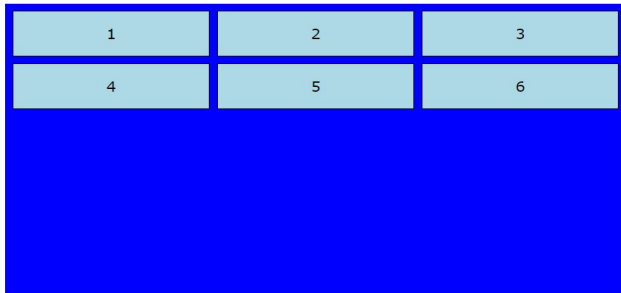


space-evenly

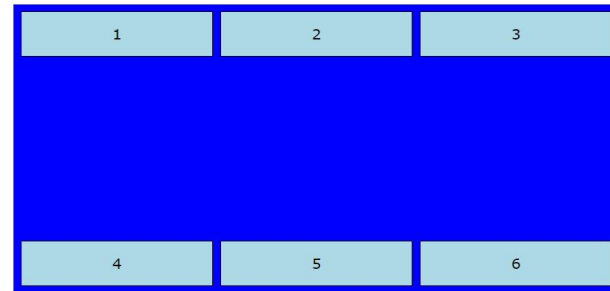
CSS Grid | Propiedad align-content



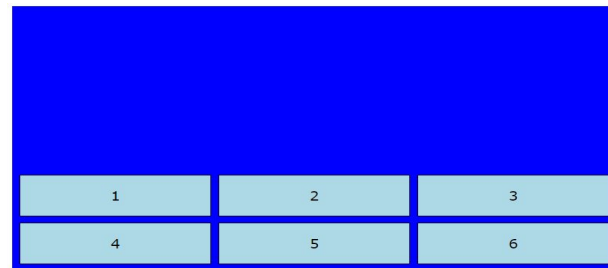
space-around



start



space-between



end

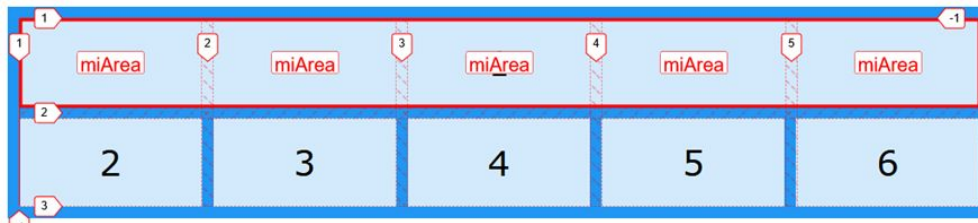
CSS Grid | Propiedad grid-area

La propiedad **grid-area** especifica el tamaño y la ubicación de un elemento de cuadrícula en el diseño, y es una propiedad abreviada para las siguientes propiedades: **grid-row-start**, **grid-column-start**, **grid-row-end** y **grid-column-end**.

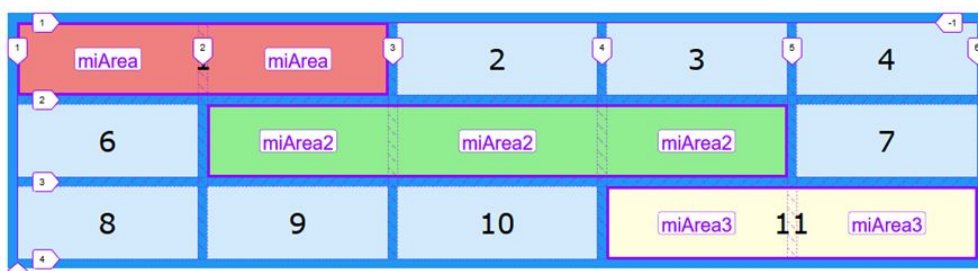
La **grid-area** también se puede utilizar para asignar un nombre a un elemento de la cuadrícula. Y se puede hacer referencia a los elementos de cuadrícula con nombre mediante la propiedad **grid-template-areas** del contenedor de cuadrícula. [+info](#)

CSS Grid | Ejemplos de grid-area

Caso 1: el área ocupa una columna.

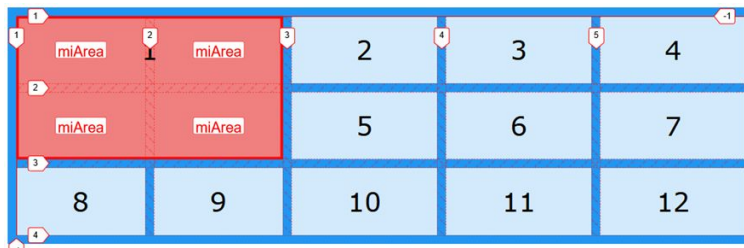


Caso 2: el área ocupa un grupo de celdas en horizontal.

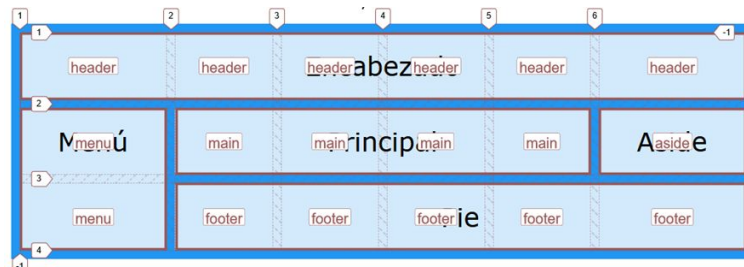


CSS Grid | Ejemplos de grid-area

Caso 3: el área ocupa un grupo de celdas en vertical.

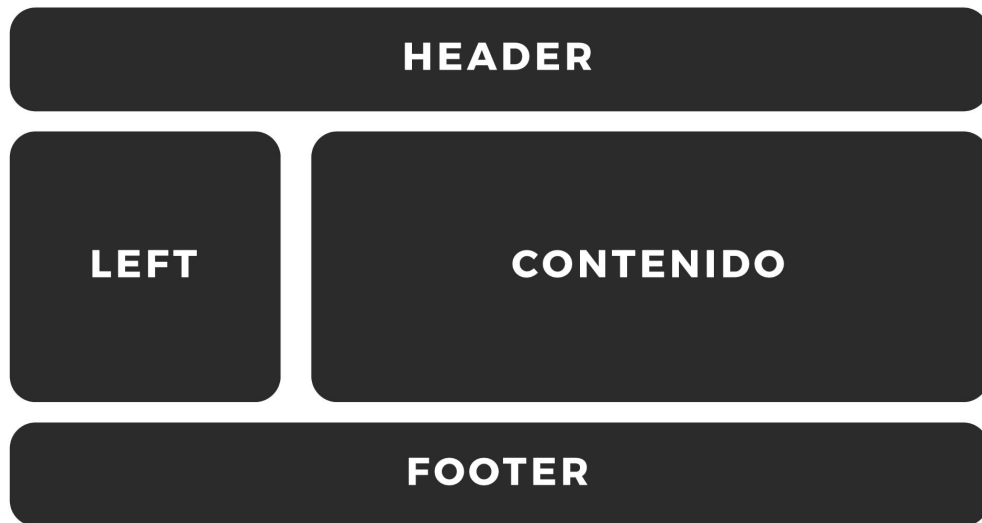


Caso 4: esquema básico de página Web, con 6 columnas y 3 filas.



CSS Grid | Ejemplo

En [este enlace](#) se puede ver cómo crear un pequeño proyecto con CSS Grid.



Artículos de interés

Documentación sobre CSS Grid:

<https://lenguajecss.com/css/maquetacion-y-colocacion/grid-css/>

https://www.w3schools.com/css/css_grid.asp

https://www.w3schools.com/css/css_grid_container.asp

https://www.w3schools.com/css/css_grid_item.asp

https://www.w3schools.com/cssref/pr_grid-area.asp

[Principios básicos del diseño con Grid](#)

Material multimedia:

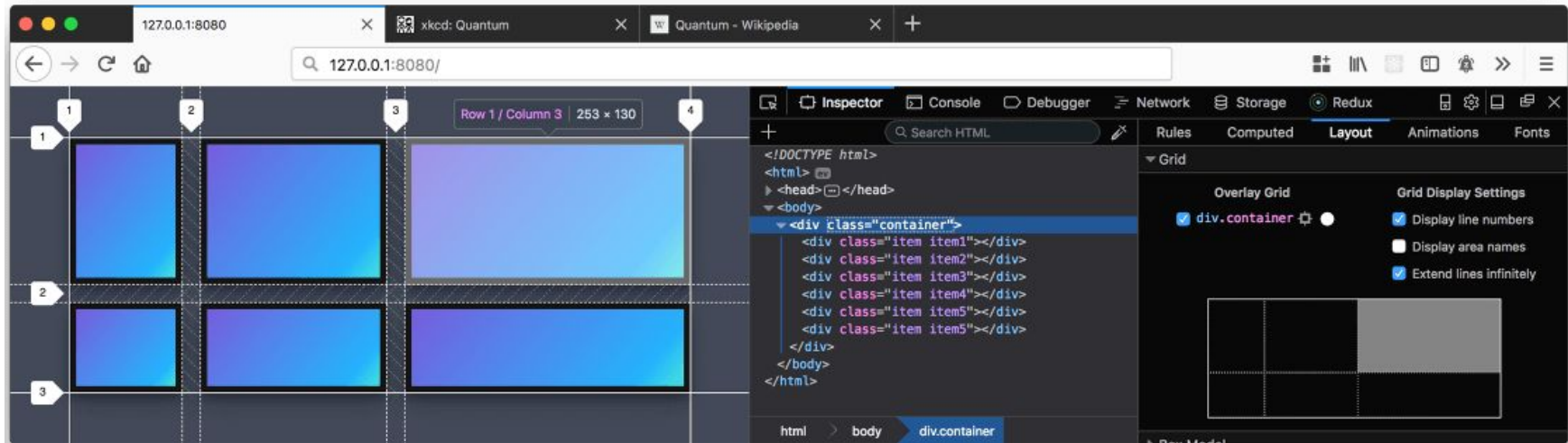
[CSS Grid desde cero](#)

[CSS GRID Página Web Responsive](#)

Material extra

Herramientas

Firefox Browser Developer Edition



Tarea para el Proyecto:

- En función de la distribución (*layout*) elegida para el proyecto utilizar Grid para maquetar el contenido principal (*main*) de las páginas del sitio.

No te olvides de dar el presente

Recordá:

- **Revisar la Cartelera de Novedades.**
- **Hacer tus consultas en el Foro.**
- **Realizar los Ejercicios obligatorios.**

Todo en el Aula Virtual.

Muchas gracias por tu atención.

Nos vemos pronto