



Docs_

Grid vs Flexbox



Diferencias entre CSS Grid y Flexbox, cuál utilizar

Hasta hace no mucho, la mayoría de diseños web se realizaban mediante tablas y algunas otras propiedades CSS. En general fáciles de utilizar y prácticas, aunque no podían utilizarse para conseguir diseños web más complejos. Para esos diseños más complejos CSS evolucionó hasta lo que es hoy, con modelos de diseño más avanzados. Entre los más populares se encuentran Flexbox y el más reciente Grid. Veamos en qué consisten cada uno de ellos

Para qué se usan CSS Grid y CSS Flexbox

Tanto CSS Grid como CSS Flexbox utilizan para ayudar a diseñar la estructura de una página web, aunque con diferencias entre ellas. Cada una tiene unas ventajas de uso según las características del proyecto que vayas a realizar y, según cada caso concreto, te interesará utilizar una o la otra. Veamos un poco más:

Qué es CSS Grid y CSS Flexbox

Flexbox se introdujo por primera vez en el 2009 como un nuevo sistema de formateo, de manera que puedas construir páginas responsive y organizar tus elementos fácilmente sin necesidad de utilizar técnicas demasiado complejas.

Para empezar a usar Flexbox solo tienes que crear un contenedor utilizando la propiedad **display: flex;**. Después de eso, cualquier elemento dentro del contenedor Flexbox se convierte en un elemento flex.

Cada elemento flex se visualiza en una única fila, organizada de izquierda a derecha. Piensa aquí en una tradicional barra de menú con diferentes elementos de menú.

Puedes cambiar la dirección de los elementos para que se visualicen verticalmente, por ejemplo en un menú vertical, utilizando **flex-direction: column**

Por otro lado, **Grid** hizo su aparición con la idea de mejorar flex, y extender su uso más allá de barras horizontales y verticales de elementos, llegando incluso al maquetado completo de páginas web, con diseño responsive y con más fácil mantenimiento.

Diferencias entre CSS Grid y CSS Flexbox

La diferencia básica entre ambas está en que **CSS Grid está basada en un formato de dos dimensiones**, mientras que **CSS Flexbox está basada en un formato de una sola dimensión**.

Un diseño basado en CSS Grid basado en filas y columnas, en dos dimensiones:

Cabecera		
Menú	Principal	Derecha
	Pie de página	

Un diseño basado en CSS Flexbox, diseñado en una sola dimensión:

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

Grid está pensado para trabajar en un formato de dos dimensiones, mientras que Flexbox solo para una dimensión. Flexbox puede trabajar en filas o en columnas, mientras que Grid puede trabajar en ambas.

Para trabajar con el formato global de una página web, CSS Grid va a servirte de más ayuda.

Ejemplos de CSS Grid y CSS Flexbox

Un ejemplo práctico de uso de Flexbox y Grid

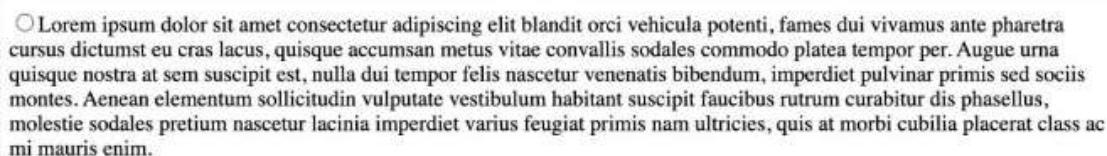
Veamos el resultado del siguiente código HTML compuesto por un formulario con una sola casilla de selección, pero junto a un texto muy largo.

```
<form action="">
```

```
  <input type="radio" name="gender" value="male">Lorem ipsum dolor sit amet consectetur  
adipiscing elit blandit orci vehicula potenti, fames dui vivamus ante pharetra cursus  
dictumst eu cras lacus, quisque accumsan metus vitae convallis sodales commodo platea  
tempor per. Augue urna quisque nostra at sem suscipit est, nulla dui tempor felis  
nascetur venenatis bibendum, imperdiet pulvinar primis sed sociis montes. Aenean  
elementum sollicitudin vulputate vestibulum habitant suscipit faucibus rutrum curabitur  
dis phasellus, molestie sodales pretium nascetur lacinia imperdiet varius feugiat primis  
nam ultricies, quis at morbi cubilia placerat class ac mi mauris enim.
```

```
</form>
```

No es habitual encontrar un elemento de selección tan largo, pero este caso nos daría como resultado algo parecido a esto:

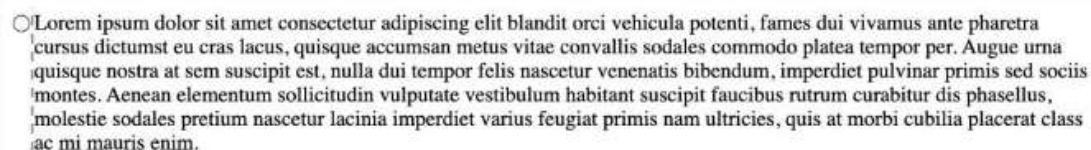


☐ Lorem ipsum dolor sit amet consectetur adipiscing elit blandit orci vehicula potenti, fames dui vivamus ante pharetra cursus dictumst eu cras lacus, quisque accumsan metus vitae convallis sodales commodo platea tempor per. Augue urna quisque nostra at sem suscipit est, nulla dui tempor felis nascetur venenatis bibendum, imperdiet pulvinar primis sed sociis montes. Aenean elementum sollicitudin vulputate vestibulum habitant suscipit faucibus rutrum curabitur dis phasellus, molestie sodales pretium nascetur lacinia imperdiet varius feugiat primis nam ultricies, quis at morbi cubilia placerat class ac mi mauris enim.

Si no queremos que el texto sobrepase la línea imaginaria junto al botón de selección, solo tenemos que añadir:

```
form { display: flex }
```

Entonces los elementos quedan ajustados automáticamente de manera horizontal:



☐ Lorem ipsum dolor sit amet consectetur adipiscing elit blandit orci vehicula potenti, fames dui vivamus ante pharetra cursus dictumst eu cras lacus, quisque accumsan metus vitae convallis sodales commodo platea tempor per. Augue urna quisque nostra at sem suscipit est, nulla dui tempor felis nascetur venenatis bibendum, imperdiet pulvinar primis sed sociis montes. Aenean elementum sollicitudin vulputate vestibulum habitant suscipit faucibus rutrum curabitur dis phasellus, molestie sodales pretium nascetur lacinia imperdiet varius feugiat primis nam ultricies, quis at morbi cubilia placerat class ac mi mauris enim.

Un ejemplo de uso con Grid:

Por ejemplo, para conseguir un menú de este tipo:



```
<header>

  <div>Inicio</div>

  <div>Página</div>

  <div>Buscar</div>

</header>
```

Nos bastaría con especificar `display: flex;` en el header:

```
header {

  display: flex;

}
```

En este caso, Flexbox se encarga de instalar cada elemento en la fila horizontalmente sin que tengas que hacer nada más. Ideal para un menú.

Sin embargo, si modificamos el código con **`display: grid;`** en lugar de '**`display: flex;`**', veríamos que los tres elementos del header no aparecen alineados horizontalmente, en lugar de eso por defecto se apilarían uno sobre otro, en forma vertical.

Para solucionarlo, podríamos añadir un código extra como este con `grid-template-columns:`

```
header {

  display: grid;

  grid-template-columns: repeat(6, 1fr);

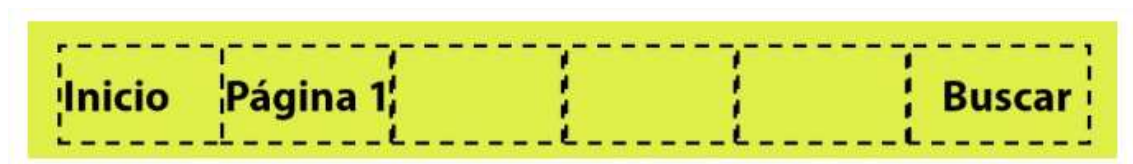
}
```

De esta manera estamos indicando que queremos que la fila disponga de columnas donde añadir diferentes elementos. Fíjate que, en este caso, estamos de alguna manera pre-diseñando el formato del menú con seis espacios, mientras que en el caso de 'display: flex' los elementos se van añadiendo uno junto al otro mientras haya espacio suficiente en la fila.

En nuestro ejemplo, con Flexbox podrías añadir más de seis elementos si hubiera espacio suficiente, mientras que con Grid, el formato queda predefinido a un número de columnas concretas.

Por eso se dice que Flexbox está orientado primero al contenido y Grid está orientado primero al diseño.

En principio, el efecto visual sería el mismo en ambos casos. Sin embargo, si echáramos un vistazo (con herramientas de desarrollador del navegador), veríamos **el grid o rejilla de ocho celdas que se ha generado**

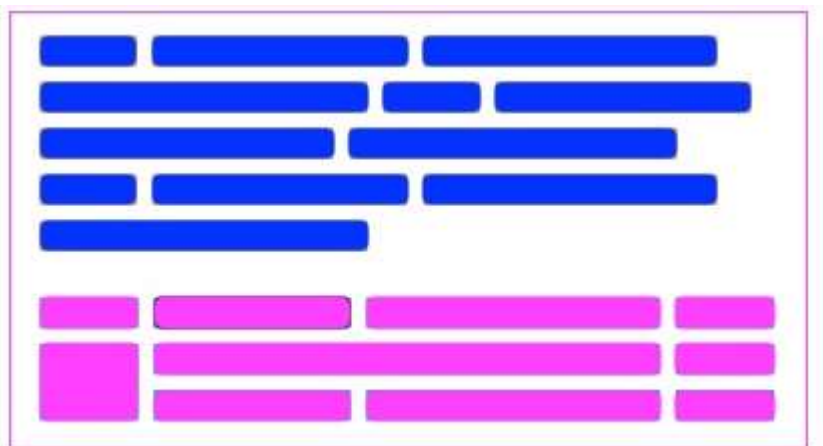


Grid funciona en dos dimensiones, y te permite, si quieres, definir la anchura de columnas y filas.

Pero ¿qué ocurre si el número de elementos sobrepasa el espacio disponible en la columna Flexbox?

Flexbox también puede envolver. **Un contenedor Flexbox comienza a generar filas** automáticamente cuando los elementos que se muestran sobrepasan la primera fila.

En este caso, las nuevas filas creadas son independientes de las filas anteriores, es decir, no se organizan por columnas, tal como lo haría Grid o una tabla:



En color azul, unos elementos envueltos con Flexbox que van apareciendo en filas inferiores según se vayan rellenando las filas superiores. En rosa, elementos alineados con Grid en filas y columnas con tamaños y posiciones concretas.

Si quieres superponer elementos, entonces es mejor utilizar Grid. Si quieres superponer elementos en Flexbox necesitarías utilizar trucos tradicionales como márgenes negativos o posicionamientos absolutos. Mientras que Grid nos va a dejar posicionar elementos.

En general:

- CSS Grid es un sistema de diseño en dos dimensiones, maneja filas y columnas, mientras que Flexbox funciona como un sistema de una sola dimensión.
- CSS Flexbox está orientado primero al contenido, CSS Grid está orientado primero al diseño.
- Tanto Flexbox como Grid pueden añadir elementos en dirección horizontal o vertical
- Grid puede expandirse en múltiples filas y columnas. Flex no.
- Los elementos y contenido interno se alinean exactamente igual en Grid y en Flexbox
- Grid tiene espacios, líneas y plantillas. Flex solo funciona linealmente.
- Utiliza Grid para generar diseños responsive en áreas grandes.

Cuándo usar CSS Grid y CSS Flexbox

Cuando usar Flexbox:

- Si tienes un proyecto de diseño pequeño entre manos, con unas pocas filas o columnas
- Cuando necesitas alinear elementos. Flexbox es perfecto para esta tarea, ya que solo tienes que definir un contenedor usando `display: flex` y luego elegir la dirección que quieras (horizontal o vertical)
- Si no sabes de ante mano cómo va a ser el diseño global de la página. Flexbox es ideal para ir añadiendo elementos sobre la marcha y lo que quieres simplemente es que todo entre en su sitio al final.

Cuando usar Grid:

- Si tienes entre manos un proyecto con un diseño complejo. El formato de dos dimensiones de Grid es ideal generar diseños complejos
- Si necesitas tener huecos y espacios entre diferentes filas o columnas. Con Grid puedes definir fácilmente espacios entre filas y columnas sin necesidad de utilizar la propiedad margin.
- Necesitas superponer elementos. Superponer elementos es muy fácil, solo necesitas utilizar las propiedades grid-column y grid-row para superponer elementos.
- Si necesitas disponer de un formato de página (layout) previo. Si necesitas disponer de la estructura de la página y luego añadir los elementos, Grid es la mejor opción.

¿Cuál es mejor, CSS Grid o CSS Flexbox?

Ambas siguen diferentes modelos de diseño y formato, de manera que no puede decirse que una sea mejor que la otra en forma general, aunque sí que hay diseñadores que son más partidarios del uso de Grid o de Flexbox.

Como hemos visto, tanto Grid como Flexbox tienen sus ventajas e inconvenientes. Según la tarea que vayas a realizar, será más recomendable el uso de una o la otra.

Recordar que Flexbox está orientado a diseños de una sola dimensión (horizontal o vertical), mientras que Grid está orientado a diseños en dos dimensiones.

Quizás donde radique la mayor diferencia entre ambos sistemas de diseño, y lo que puede decantarte a usar uno u otro, es que Flexbox está orientado primero al contenido y Grid lo está al diseño. Si quieres diseñar y maquetar una web antes incluso de añadir contenidos, entonces Grid es para ti. Si quieres ir añadiendo contenido que luego vas a maquetar dentro de un diseño, ve a por Flexbox.

Un punto extra a favor de Flexbox, es que suele conseguir que las páginas se rendericen de forma más rápida que con Grid, pero en muchas ocasiones esa diferencia de velocidad no es significativa.

