



Frontend Fundamentals

Flexbox y CSS Grid

Cuando se lanzó CSS Grid, se pensó que este vendría a reemplazar a Flexbox, sin embargo, esto no es completamente una aseveración correcta, ya que con CSS Grid se pueden realizar cosas que con Flexbox no, y viceversa. Lo ideal sería utilizarlos en conjunto.

Por ejemplo, se puede tener un contenedor con propiedad de Grid y que sus descendientes tengan propiedad Flex, o al revés. Por tanto, es muy importante que conozcas sus características.

Si únicamente necesitas que un contenedor esté al lado de otro, por simplicidad, posiblemente la mejor opción sería utilizar Flexbox en lugar de Grid. No obstante, si tu diseño (layout) está compuesto de filas y columnas que van cambiando, entonces CSS Grid posiblemente sea la mejor opción.

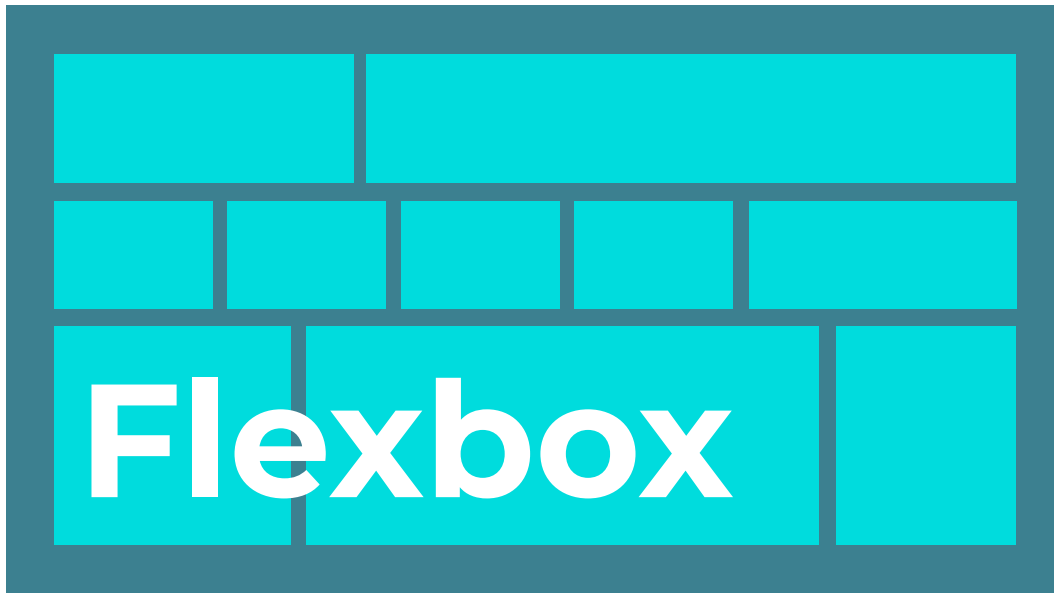
Pero, entonces, ¿CSS Grid sustituye a Flexbox? Cuando conozcas CSS Grid, habrá algunas características que te gustarán más, debido a la facilidad con la que se puede crear un layout. Hacer un layout completo es más fácil con CSS Grid que con Flexbox.

Durante esta experiencia educativa comprenderás las diferencias entre CSS Grid y Flexbox.



Flexbox

Como lo indica Mdn (2022), Flexbox es un método disponible en CSS para alinear y estructurar visualmente el contenido de un elemento HTML. Flexbox nos permite usar la propiedad de display para indicar cuando un elemento y su contenido, en caso de tener, puede alinearse de manera flexible.



Al elemento que recibe la propiedad display: flex se le conoce como flex container, mientras que a los elementos contenidos en el elemento en mención se les conoce como flex items.

Es posible controlar el posicionamiento del contenido desde el flex container aplicando propiedades de alineamiento vertical (column) y horizontal (row).

Cuando se define un flex container, sus elementos hijos (flex items) se modifican a ponerse uno al lado de otro, debido a que, por defecto, el contenido toma una dirección de row. La dirección se puede modificar a vertical a través de la propiedad flex-direction que se aplica al flex container.

A través de las propiedades justify-content y align-items se puede modificar el alineamiento de los hijos, dependiendo de la dirección que estén siguiendo.

Ejemplo:

Si quisiéramos que los flex items estén alineados al centro horizontalmente, esto se podría lograr con justify-content: center, siempre y cuando el flex-direction sea row.

Puesto que justify-content alinea a los flex items en el eje principal, align-items lo hace en el eje secundario. Por lo tanto, observa en el ejemplo anterior que si el flex-direction fuera column para alinear horizontalmente, necesitarías usar la propiedad align-items.

En la siguiente imagen puedes visualizar las propiedades de Flexbox.

Explicación

Flex container

Es el elemento padre que contienen los flex items.



Para componentes pequeños.

Encabezados.



Menús de navegación.

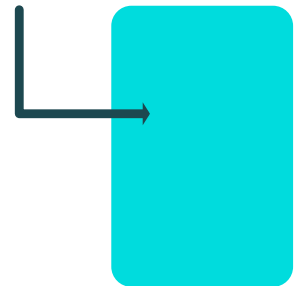
Flex items

Son todos los elementos de flex container, llamados elementos hijos.

Los flex items son flexibles.

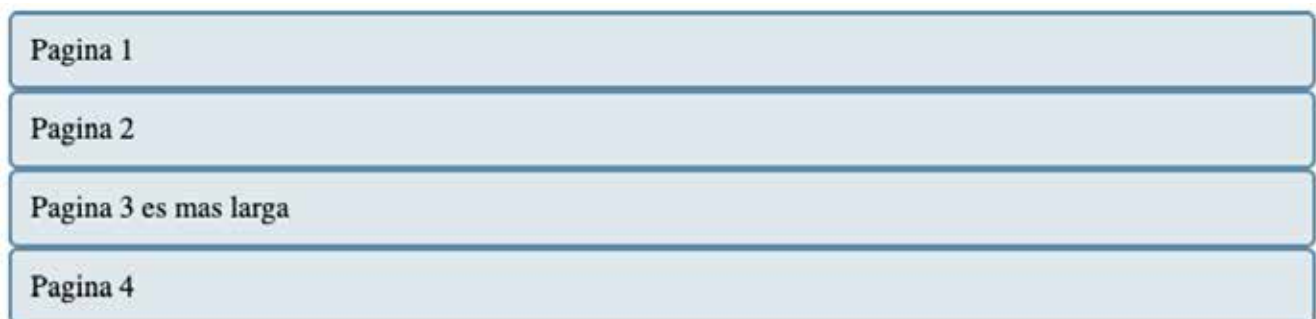
Usar márgenes para crear separaciones entre elementos (se hace desde los hijos).

Para diseños unidimensionales, controlan solo filas o columnas.



Uno de los ejemplos más recurrentes para el uso de Flexbox es tener una lista de elementos mostrada de manera horizontal. Cuando tenemos el caso en el que se requiere mostrar horizontalmente y que no se necesita tener un espacio adicional, habrá que decidir lo que se hará con ese espacio; podrías, por un lado, mostrar el espacio fuera de los elementos, separarlos con espacio entre ellos o alrededor de ellos, o colocar ese espacio dentro de cada uno de los elementos.

Considera el siguiente ejemplo sin el uso de flex:



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

A partir del siguiente código HTML:

```
<nav>
  <ul>
    <li><a href="#">Pagina 1</a></li>
    <li><a href="#">Pagina 2</a></li>
    <li><a href="#">Pagina 3 es mas larga</a></li>
    <li><a href="#">Pagina 4</a></li>
  </ul>
</nav>
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Para distribuir el espacio entre o alrededor de los elementos se utiliza la propiedad justify-content. Y con la opción space-between se pueden crear espacios iguales entre los elementos, como se puede ver en el siguiente ejemplo:



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Para crear este efecto, el CSS tendría que contener estas líneas:

```
nav ul {
  display: flex;
  justify-content: space-between;
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Otra manera de utilizar el espacio considerado es colocándolo dentro de los elementos. Para esto, se utilizará flex: auto y esto hace que todos los elementos tengan el mismo tamaño y que el espacio sea distribuido uniformemente entre todos los elementos:



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Y el código CSS ahora tendría las siguientes líneas:

```
nav ul {  
  display: flex;  
}  
  
nav li {  
  flex: auto;  
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

También pueden alinearse los elementos en el eje principal utilizando márgenes automáticos, esto permitirá alinear algunos elementos a la derecha y otros a la izquierda para que los elementos se vean así:



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Para lo cual, hay que hacer un pequeño cambio en el código HTML:

```
<nav>  
  <ul>  
    <li><a href="#">Pagina 1</a></li>  
    <li><a href="#">Pagina 2</a></li>  
    <li><a href="#">Pagina 3 es mas larga</a></li>  
    <li class="push-right"><a href="#">Pagina 4</a></li>  
  </ul>  
</nav>
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Y en el CSS:

```
nav ul {  
  display: flex;  
  margin: 0 -10px;  
}  
  
nav li {  
  margin: 0 10px;  
}  
  
.push-right {  
  margin-left: auto;  
}
```

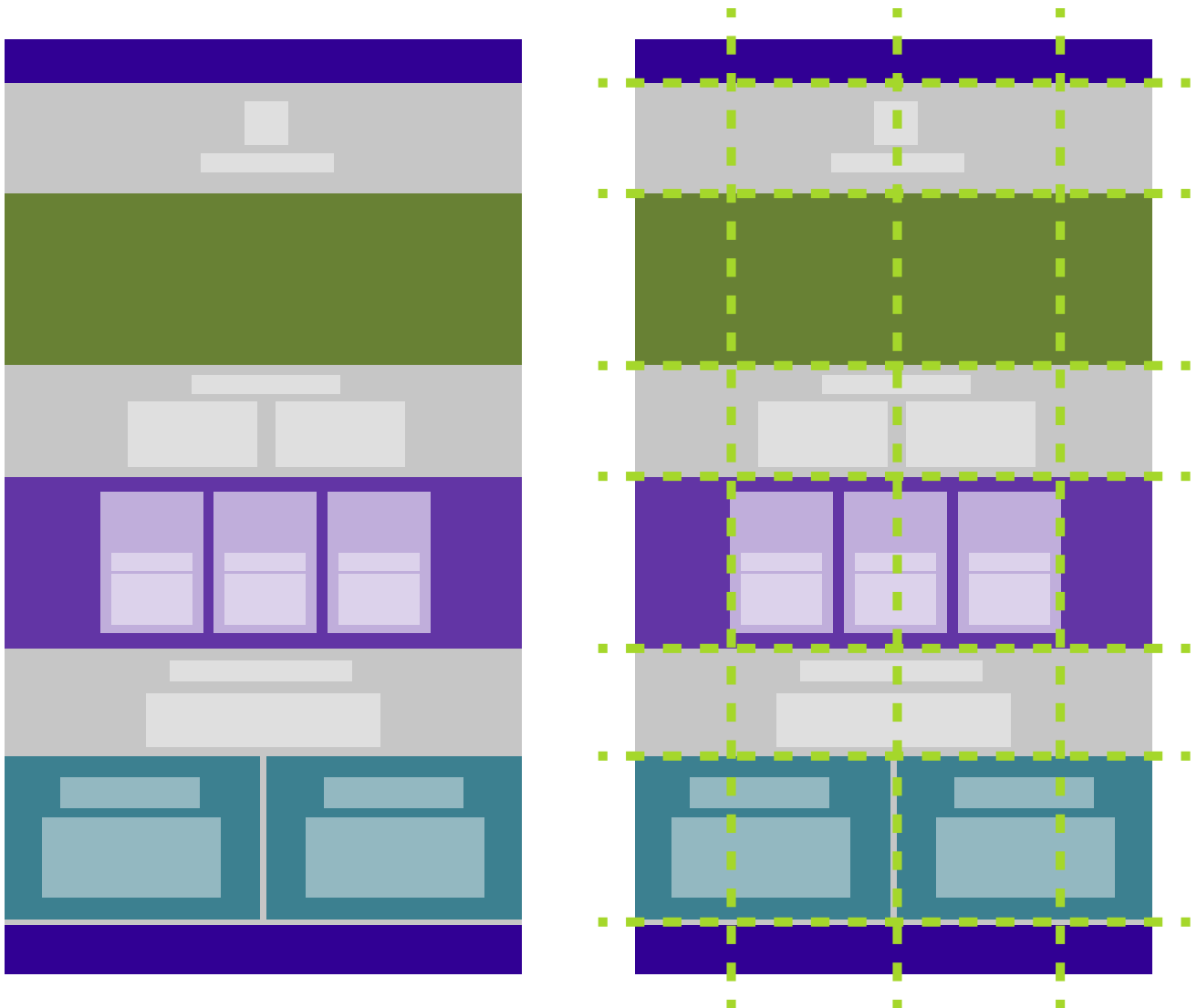
Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Como podrás ver, el uso de Flexbox nos permite organizar los elementos de una manera más sencilla y eficaz. Sin embargo, hay más métodos de diseño que podrían, en conjunto con Flexbox, ayudarte en el diseño de tu página web.

CSS Grid

Uno de estos métodos es CSS Grid. El término grid, cuadrícula o rejilla en CSS, según lo define W3Schools (2022), hace referencia a la apariencia de filas y columnas que generan los elementos de CSS.

Anteriormente, para lograr esta apariencia solo se consideraban los displays de inline-block y propiedades como float. Después, con la aparición de Flexbox se pudo simular este comportamiento de manera flexible. Sin embargo, no existía ninguna forma nativa dedicada a este formato que es muy común encontrar en las páginas web modernas.



En CSS3 tenemos un nuevo conjunto de propiedades que permiten lograr este objetivo de una manera más práctica y sencilla, que toma el nombre de Grid CSS.

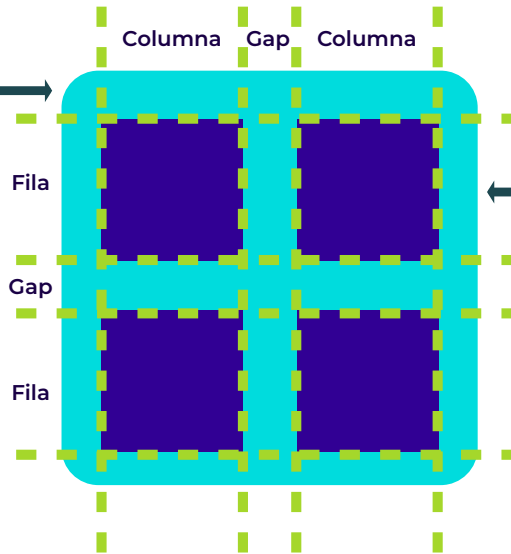
La siguiente imagen muestra las propiedades de Grid CSS.

Grid container

Es el contenedor o elemento **padre** que contiene a los **grid items**.

Usa **gap** para crear separaciones entre elementos **(se hace desde el padre)**.

Grid no reemplaza a Flexbox y **Flexbox** no reemplaza a Grid.



Grid items

Son los **hijos** directos de un grid container **y los que componen la cuadrícula**.

Para diseños bidimensionales, controlan **filas y columnas a la vez**.

- Para diseños de **página completa**.
- **Sistema de columnas**.
- **Galería de imágenes**.

El siguiente ejemplo consiste en mostrar un grid de tres columnas con filas creadas utilizando un mínimo de 100 píxeles y con el máximo automático. Cada uno de los elementos se colocan utilizando un posicionamiento en línea:



Utilizando el siguiente código HTML:

```
<div class="wrapper">
  <div class="one">Uno</div>
  <div class="two">Dos</div>
  <div class="three">Tres</div>
  <div class="four">Cuatro</div>
  <div class="five">Cinco</div>
  <div class="six">Seis</div>
</div>
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Las características del div “wrapper” se definieron en el CSS de la siguiente manera:

```
.wrapper {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-gap: 10px;
  grid-auto-rows: minmax(100px, auto);
}

* {
  box-sizing: border-box;
}

.wrapper {
  max-width: 940px;
  margin: 0 auto;
}

.wrapper > div {
  border: 2px solid #d9480f;
  border-radius: 5px;
  background-color: #d9480f;
  padding: 1em;
  color: #d9480f;
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Las características de cada sección son las siguientes:

```
.one {  
  grid-column: 1 / 3;  
  grid-row: 1;  
}  
.two {  
  grid-column: 2 / 4;  
  grid-row: 1 / 3;  
}  
.three {  
  grid-column: 1;  
  grid-row: 2 / 5;  
}  
.four {  
  grid-column: 3;  
  grid-row: 3;  
}  
.five {  
  grid-column: 2;  
  grid-row: 4;  
}  
.six {  
  grid-column: 3;  
  grid-row: 4;  
}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Tecmilenio no guarda relación alguna con las marcas mencionadas como ejemplo. Las marcas son propiedad de sus titulares conforme a la legislación aplicable, se utilizan con fines académicos y didácticos, por lo que no existen fines de lucro, relación publicitaria o de patrocinio.

Después de revisar las dos formas de organizar elementos en CSS, se puede concluir que ambas son complementarias.

Si bien, los espacios en las etiquetas son invisibles, posicionan el contenido en un lugar determinado. El espacio que ocupan solo se puede visualizar cuando se abren las herramientas de desarrollo del navegador, pero ¿qué estarían haciendo Flex y Grid respectivamente?

Flexbox posiciona el contenido de las etiquetas contenedoras, mientras que Grid genera una cuadrícula que ayuda a dividir las cajas que son etiquetas contenedoras.

Referencias bibliográficas

- Mdn. (2022). CSS. Recuperado de <https://developer.mozilla.org/es/docs/Learn/CSS>
- W3Schools. (2022). *CSS Tutorial*. Recuperado de <https://www.w3schools.com/css/>

Para saber más

Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.

Lecturas

Para conocer más acerca de **Flexbox y CSS Grid**, revisa lo siguiente:

- Lenguaje CSS. (s.f.). *Modelo de cajas flexibles (Flex)*. Recuperado de <https://lenguajecss.com/css/maquetacion-y-colocacion/flexbox/>

Videos

Para conocer más acerca de **Flexbox y CSS Grid**, te sugerimos revisar lo siguiente:

- Fazt Code. (2019, 20 de septiembre). CSS Grid | *Introducción Práctica desde Cero* [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=QBOUSrMqISQ>

Asegúrate de:

- Comprender el uso correcto de CSS Grid y Flexbox.
- Comprender que ambas tecnologías no son enemigas u opuestas, sino complementarias.
- Identificar cuándo puedes utilizar una tecnología y cuándo la otra.

Requerimientos Técnicos

El uso y descarga del software deberá apegarse a los términos y condiciones del sitio oficial del fabricante y su uso será responsabilidad de quien lo descargue. Tecmilenio no tiene licencia ni posee los derechos sobre dicho software.

- Computadora con acceso a internet.
- Contar con el editor de código Visual Studio Code instalado.
- Contar con el plugin para despliegue del sitio web Live Server instalado.

Prework

Bienvenido a tu *prework* 4. A continuación, enlistaremos algunos materiales y recursos que te serán de ayuda. Ten en cuenta que completar tu prework te va a preparar para tener conceptos claros.

- Revisar el tema **Flexbox** y **CSS Grid**.

Antes de revisar el tema, considera lo siguiente: algunos recursos textuales y de video pueden estar en inglés. Si no tienes dominio sobre el idioma, te recomendamos el uso de un traductor y de activar los subtítulos en el reproductor de YouTube:

- Busca el logo de configuración, dentro de la ventana del video.
- Haz clic ahí y selecciona Subtítulos.
- Haz clic en Activar.

Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.

Revisa este video para comprender los conceptos clave de la **sintaxis en Flexbox**:

- jonmircha. (2019, 20 de noviembre). *Curso Flexbox CSS: (1/13) Introducción* - #jonmircha [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=AAtvnv6LNMk&list=PLvq-jlkSeTUbQc3dGsssp8lxAi5npMrys>

Revisa este video para aprender a manipular el Grid CSS:

- jonmircha. (2020, 30 de noviembre). *Curso Grid CSS: (1/17) Introducción y Conceptos Básicos* - #jonmircha [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=gCox2CqYvto&t=21s>

La obra presentada es propiedad de ENSEÑANZA E INVESTIGACIÓN SUPERIOR A.C. (UNIVERSIDAD TECMILENIO), protegida por la Ley Federal de Derecho de Autor; la alteración o deformación de una obra, así como su reproducción, exhibición o ejecución pública sin el consentimiento de su autor y titular de los derechos correspondientes es constitutivo de un delito tipificado en la Ley Federal de Derechos de Autor, así como en las Leyes Internacionales de Derecho de Autor.

El uso de imágenes, fragmentos de videos, fragmentos de eventos culturales, programas y demás material que sea objeto de protección de los derechos de autor, es exclusivamente para fines educativos e informativos, y cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por UNIVERSIDAD TECMILENIO.

Queda prohibido copiar, reproducir, distribuir, publicar, transmitir, difundir, o en cualquier modo explotar cualquier parte de esta obra sin la autorización previa por escrito de UNIVERSIDAD TECMILENIO. Sin embargo, usted podrá bajar material a su computadora personal para uso exclusivamente personal o educativo y no comercial limitado a una copia por página. No se podrá remover o alterar de la copia ninguna leyenda de Derechos de Autor o la que manifieste la autoría del material.