



CENTRO UNIVERSITARIO FELIPE CARRILLO PUERTO

ADA 7: Flexbox y bootstrap

8vo Cuatrimestre

Por

Canul Ceh Adriana Guadalupe

Trabajo presentado como cumplimiento de los requisitos de la materia de: Programación de
Sitios web

Maestro: Erik Romero Lara

Mérida, Yucatán, 6 de marzo 2025

Flexbox

Es un método de diseño de página unidimensional para comparar elementos en filas o columnas. Los elementos de contenido se ensanchan para rellenar el espacio adicional y se encogen para caber en espacios más pequeños.

Se puede lograr obtener cualquier forma conveniente y flexible un diseño de página sencillo con requisitos como:

- Centrar verticalmente un bloque de contenido dentro de su elemento padre.
- Hacer que todos los elementos secundarios de un contenedor ocupen una cantidad igual del ancho/alto disponible, independientemente del ancho/alto que haya disponible.
- Hacer que todas las columnas en una compaginación en columnas múltiples adopten la misma altura incluso si contienen cantidades diferentes de contenido.

Los elementos que se pueden colocar en “cajas flexibles” deben de tener un valor especial en el elemento “display”. Algunos ejemplos en los que se pueden utilizar son:

```
Section {
```

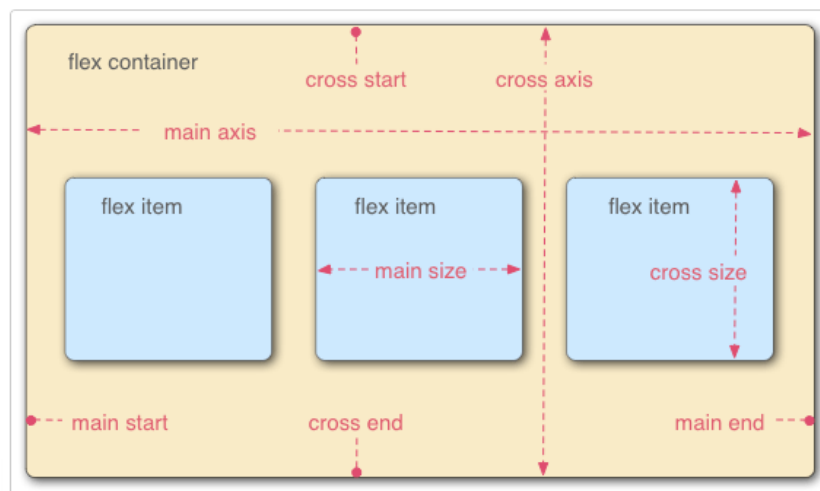
```
    Display: flex;
```

```
}
```

Nota: Hace que el elemento <section> se convierta en un contenedor flex y sus hijos en elementos flexibles. Es decir, el elemento al que se le da el valor de “display de flex” actúa como un elemento a nivel de bloque en términos de cómo interactúa con el resto de la página, pero sus elementos secundarios se presentan como elementos flexibles.

Modelo flexible

Como los elementos se presentan como cajas flexibles, se distribuyen con respecto a dosejes:



- El **eje principal (main axis)** es el eje que corre en la dirección en que se coloca los elementos flexibles, es decir que según se dispongan las filas en una página o hacia abajo según se dispongan las columnas en una página. **El inicio y el final de este eje se denominan inicio.**
- El **eje transversal (cross axis)** es el eje que corre perpendicular a la dirección en la que se colocan los elementos flexibles. El inicio y el final de **este eje se denominan inicio transversal (cross start) y extremo cruzado (cross end).**
- El elemento padre que tiene establecido “display: flex” (el elemento <section> es el ejemplo) se llama **contenedor flexible.**
- Los elementos que se presenta como cajas flexibles dentro del contenedor flexible se denominan **elementos flexibles** (son los elementos <article> es el ejemplo).

¿Columnas o filas?

Los elementos flexbox proporcionan una propiedad llamada “flex-direction” que especifica en qué dirección corre el eje principal; por defecto, está establecido en el valor “row”, por lo que se presenta en una fila en la dirección en que se escribe el idioma predeterminado de tu navegador (de izquierda a derecha, en caso de un navegador en español).

```
Section {
    Flex-direction: column;
}
```

Este ejemplo vuelve a colocar los elementos en una disposición en columna, al igual que antes de agregar cualquier CSS.

Delimitar

Un problema que surge cuando tienes una cantidad fija de ancho o alto en un diseño es que los hijos de un elemento flexbox eventualmente desborda el contenedor y romper el diseño. Algunos ejemplos son:

```
Section {
    Flex-wrap: wrap;
    Flex: 200px;
}
```

Ahora hay varias filas y en cada fila caben tantos elementos hijo de un elemento flexbox como sean necesarios, y cualquier desbordamiento hace saltar el elemento hacia la línea siguiente. La declaración “flex: 200px” que hemos establecido en los artículos significa que cada uno tendrá al menos 200px de ancho.

Dimensionamiento flexible de elementos flexibles.

```
article {  
    flex: 1;  
}
```

Este es un valor de proporción sin unidades que especifica la cantidad de espacio disponible sobre el eje principal que ocupa cada elemento flexible. En este caso, damos a cada elemento `<article>` un valor de 1, lo que significa que todos ocuparán una cantidad igual del espacio libre restante después de que se hayan establecido elementos como el área de relleno y el margen. Es una proporción, lo que significa que dar a cada elemento flexible un valor de 400000 tendría exactamente el mismo efecto.

```
article:nth-of-type(3) {  
    flex: 2;  
}
```

El tercer artículo ocupa el doble del ancho disponible que los otros dos; ahora hay cuatro unidades de proporción disponibles en total. Los primeros dos elementos flexibles tienen una cada uno, por lo que ocupan $1/4$ del espacio disponible cada uno. El tercero tiene dos unidades, por lo que ocupa $2/4$ del espacio disponible (o $1/2$).

También se puede especificar un valor de tamaño mínimo dentro del valor flexible.

```
article {  
    flex: 1 200px;  
}
```

```
article:nth-of-type(3) {  
    flex: 2 200px;  
}
```

Este establece básicamente que a cada flexible se le da primero 200px del espacio disponible. Después de eso, el resto del espacio disponible se reparte de acuerdo con las unidades de proporción.

Flex: forma completa y abreviada

Flex es una propiedad abreviada que puede especificar hasta tres valores diferentes:

- El valor de proporción sin unidades, se puede especificar individualmente usando la propiedad “flex-grow”.
- Un segundo valor de proporción sin unidades, “flex-shrink” que entra en juego cuando los elementos flexibles desbordan el contenedor. Hay que especificar cuánto de la cantidad desbordada se quita al tamaño de cada elemento flexible para evitar que el contenedor se desborde.
- El valor de tamaño mínimo se puede especificar individualmente usando el valor “flex-basis”.

Alineación horizontal y vertical

Se usar funciones de los elementos flexbox para elementos alineables flexibles sobre el eje principal y transversal. Ejemplo:

```
div {
  display: flex;
  align-items: center;
  justify-content: space-around;
}
```

“Align-items” controla donde se ubican los elementos flexibles en el eje transversal.

- Por defecto, el valor es “stretch”, que ensancha todos los elementos flexibles para rellenar el elemento primario en la dirección del eje transversal. Si el elemento padre no tiene un ancho fijo en la dirección del eje transversal, todos los elementos flexibles son tan largos como los elementos flexibles más largos.
- El valor center que utilizamos en nuestro código anterior mantiene las dimensiones intrínsecas de los elementos, pero los centra sobre el eje transversal. Es por eso que los botones de nuestro ejemplo ahora están centrados verticalmente.
- También puedes tener valores como “flex-start y flex-end”, que alinean todos los elementos al inicio y al final del eje transversal, respectivamente.

Puedes anular el comportamiento “align-items” para elementos flexibles individuales aplicándoles la propiedad “align-self”.

```
button:first-child {
  align-self: flex-end;
}
```

“justify-content” controla dónde se ubican los elementos flexibles sobre el eje principal.

- El valor por defecto es “flex-start”, que asienta todos los elementos al comienzo del eje principal.
- Puedes usar “flex-end” para que se asienten al final.
- “Center” también es un valor de “justify-content” (para contenido alineal), que asienta los elementos flexibles sobre el centro del eje principal.
- El valor “space-around” que hemos usado antes es útil porque distribuye todos los elementos de manera uniforme sobre el eje principal y deja un poco de espacio en cada extremo.
- Hay otro valor, “space-between” que es muy similar a “space-around”, pero no deja espacio en los extremos.

Cajas flexibles anidadas.

Es perfectamente aceptable configurar un elemento flexible para que también sea un contenedor flexible, de modo que los elementos secundarios también se dispongan como cajas flexibles.

Hay un elemento `<section>` que contiene tres elementos `<article>`. El tercer elemento `<article>` contiene tres elementos `<div>`. Ejemplo:

section - article

article

article - div - button

div button

div button

button

button

Bootstrap

Bootstrap es un framework CSS utilizado en aplicaciones frond-end; es decir, en la pantalla de interfaz con el usuario para desarrollar aplicaciones que se adaptan a cualquier dispositivo. El framework combina CSS y JavaScript para estilizar los elementos de una página HTML, ya que permite mucho más cambiar el color de los botones y los enlaces.

Es una herramienta que proporciona interactividad en la página, por lo que ofrece una serie de componentes que facilitan la comunicación con el usuario, como menús de navegación, controles de páginas, barras de progreso, etc.

Asimismo, es usada principalmente por su adaptable uso del CSS a teléfonos, tabletas y computadores de escritorio, es compatible con todos los navegadores y es fácil de usar por cualquiera que tenga conocimientos básicos de HTML y CSS.

Funcionamiento

Bootstrap está constituido por una serie de archivos CSS y JavaScript responsables de asignar características específicas a los elementos de las páginas. Además, contiene un archivo principal llamado “Bootstrap.css” la estructura del framework se compone de dos directorios:

CSS: Contiene los archivos necesarios para estilización de los elementos y una alternativa al tema original.

JS: Contiene la parte posterior del archivo Bootstrap .js (original y minificado), responsable de la ejecución de aplicaciones de estilo que requieren manipulación interactiva.

Para asignarle una característica a un elemento, simplemente se debe informar la clase correspondiente en la propiedad “class” del elemento que será estilizado.

Para garantizar la responsividad, el framework funciona con:

- la estilización del elemento <div>;
- el uso del class container.

El elemento <div>, funciona para crear una serie de notas, similar a una tabla, capaz de estructurar la página de forma adaptable. Este elemento es más flexible, ya que permite definir y cambiar el tamaño de la longitud fácilmente.

Bootstrap le ha asignado al elemento <div> una característica de class container, que funciona para determinar las dimensiones apropiadas para los elementos insertados en ese espacio.

Básicamente, el framework funciona con tres tipos de containers:

Container: como un conjunto con una propiedad de ancho máximo, que determina qué tamaño de tela es ideal para crear el diseño de página.

Container-fluid: considera la longitud total de la tela del dispositivo para definir el diseño. Para esto, se considera la propiedad width —100% en todos los límites de tamaño de tela—.

Container-{ breakpoint}: considera width —100% hasta alcanzar un cierto tamaño—.

Otra interesante función de este framework es la cantidad de componentes que pueden ser usados para proporcionar una mejor interacción y perfeccionar la comunicación con el usuario.

Alertas: permite una configuración simple y rápida de diferentes tipos de alertas, con colores específicos, según la situación. Para mostrarle al usuario una alerta que indique atención, por ejemplo, simplemente debemos usar alert-danger y aparecerá un cuadro de texto con un fondo rojo:

¡Atención! ¡Cuidado mensaje de alerta!

Carrusel: Un componente ampliamente utilizado en Bootstrap es el Carrusel, una presentación de diapositivas, es decir, una herramienta que permite la visualización de imágenes de manera receptiva. También permite la inclusión de efectos especiales para la transición de imágenes y controles de visualización, como por ejemplo los indicadores de “siguiente” y “anterior”.

Barra de navegación: Permite la construcción de un sistema de navegación sensible, ya que es posible configurar diferentes formas de presentar el menú, elegir entre posicionamiento lateral o superior y, también, definir una forma de visualización que se pueda extender o contraer. Asimismo determina cómo mostrar los enlaces del menú, que pueden ser en forma de botón, enlace, menú suspenso, entre otras configuraciones, para facilitar la implementación de la navegación del sitio.

Versiones

Bootstrap 3 se lanzó en 2013; sin embargo, también cubre versiones más actualizadas como Bootstrap 4 que fue lanzada en 2018 y Bootstrap 5 que fue lanzada en 2021 y es la más actualizada hasta el momento.

Referencias

https://developer.mozilla.org/es/docs/Learn_web_development/Core/CSS_layout/Flexbox

https://www.w3schools.com/bootstrap/bootstrap_get_started.asp#:~:text=What%20is%20Bootstrap%3F,well%20as%20optional%20JavaScript%20plugins

<https://rockcontent.com/es/blog/bootstrap/>