****

**Instituto Superior Tecnológico “Compu Sur”, con condición de Superior Universitario**

CARRERA DE

TECNOLOGÍA SUPERIOR EN DESARROLLO DE SOFTWARE – IV SEMESTRE

PATRICIO SUÁREZ-5 DE JUNIO DEL 2025

ACTUACIÓN EN CLASE

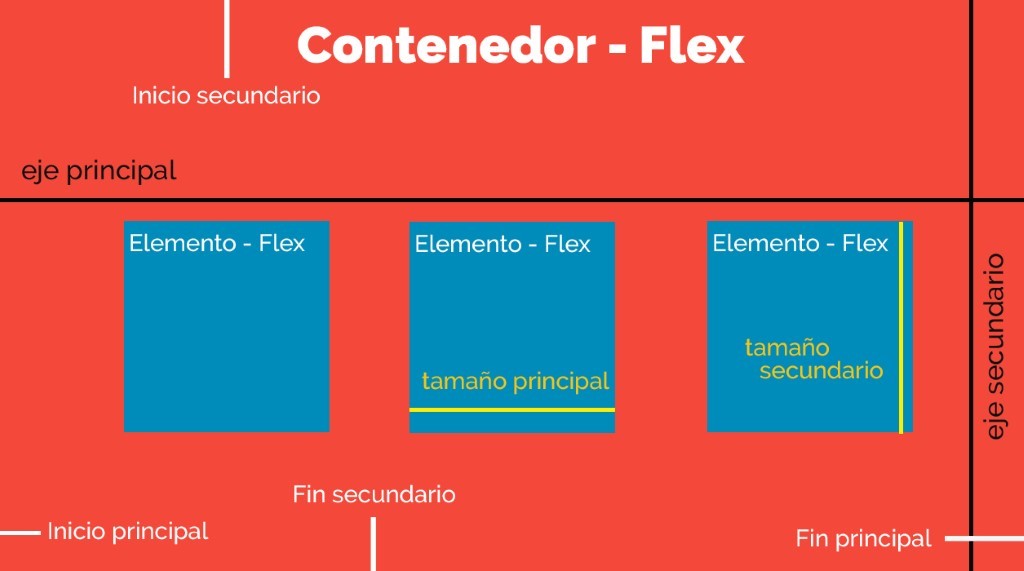
**LENGUAJE WEB**

1. **Consultar que es FlexBox y cuáles son sus funciones.**

En lo referente al desarrollo web bajo HTML/CSS es siempre recomendable mantener un buen diseño de página web, por lo que siempre se han buscado formas para que las páginas web tengan una buena estructura tanto interna como externa, y esto ha cambiado constantemente con los avances de los años y el perfeccionamiento de HTML/CSS, la principal herramienta para el desarrollo web desde sus inicios, por lo que en el marco de este desarrollo, siempre se ha buscado que el estilo de una página web encaje perfectamente en todos los entornos y más tarde, en todos los dispositivos.

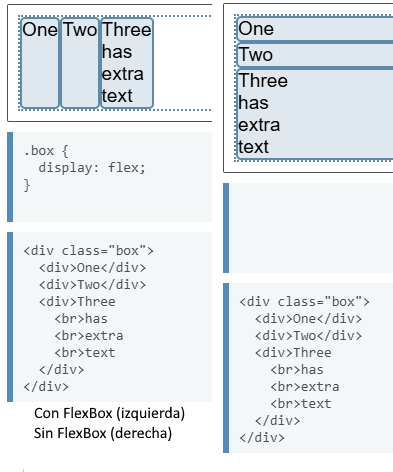
Es por eso que uno de los aspectos más importantes el desarrollo web ha sido el flujo de diseño, el cual en un inicio también aplicaba a cuando cambiaba el tamaño de la ventana de navegador o se cambiaba o actualizaba de navegador, haciendo esto algo díficil para los primeros desarrolladores web, que desarrollaron la flotación y posicionamiento como soluciones, y que funcionaban perfectamente, pero al costo de sobrecargar el código y limitar el desarrollo de una página web, por lo que ante las limitacines de la época, estos problemas se mantuvieron durante muchos años, hasta que se desarrolló una alternativa para el flujo de desarrollo: Flexbox.

Flexbox, desarrollado en los últimos años, es básciamente un modulo CSS3 creado específicamente para mejorar los diseños web en general, reduciendo las líneas de código necesarias para obtener una buena resolución de página en todos los dispositivos ya que se encarga de todo lo relaciondo a las posiciones de elementos de una página web y la resolución de esta manteniendo un ancho y alto adecuado, manteniendo también una buena relación entre tamaño de la interfaz web y los elementos del sitio web. Así, Flexbox se convierte en una herramienta vital para el desarrollo responsivo con dispositivos móviles que desde hace una década es parte del desarrollo web.

Para entender Flexbox y empezar a utilizarlo, es necesario entender su funcionalidad principal que es la ya mencionada que es la relacionada al dimensionamientos de los objetos y para esto se debe entender su estructura y sus componentes básicos. En este caso, Flexbox contará con dos tipos de cajas, que se conocerán como *Caja-Flex* (o *Contenedor-Flex*) y *Elemento-Flex*. Estos serán respectivamente, el primer y segundo elemento principales de la estructura de sitios web que de Flexbox establecerá, por lo que la Caja-Flex (que será una) será el complemento donde se contendrán a todos los Elementos Flex (que pueden ser en cualquier número) y estos Elementos Flex serán los que se ubicarán de acuerdo a las propiedades de tamaño, alto, ancho, tec, que se le implementen. Además, cada diseño Flexbox tendrá ejes principales y secundario para el posicionamiento horizontal y vertical, respectivamente de los elementos flex.

**Características de Flexbox**

Una vez que se sabe qué es Flexbox, junto con sus elementos principales y se quiere empezar a utilizarlo en un proyecto de cualquier tipo, será hora de revisar sus propiedades para poder utilizar este complemento de la mejor forma y aprovechar sus beneficios.

Primero, los contenedores de Flexbox contenerán una dirección (que son cuatro: inicio y fin cada uno de tipo principal o secundario) que irán de inicio a final y definen el flujo de cada elemento en cada uno de estos contenedores, además: el contenedor tendrá algunas propiedades más para sus elementos como son; el despliegue sobre una una fila, el inicio sobre el margen inicio del eje principal, que no se ajustan a la dimensión principal, etc. Gracias a estas características los elementos se allinean por defecto en un sola fila y se ajustan en caso de haber demasiados dentro de un contenedor.

En el gráfico pueden verse tres cajas de contenido con (a la izquierda) o sin (a la derecha) Flexbox, observando como se aplican estas características al implementarse Flexbox en un documento, evidénciandose que tan útil puede ser esta herramiento a la hora de crear menús personalizados y responsivos.

**Propiedades de Flexbox**

Ahora que se han visto las características principales de Flexbox, es hora de ver sus propiedades principales, las que permiten aprovechar al máximo todas las características de Flexbox, debido a que estas propiedades definen la ubicación y tamaño de los elementos en un contenedor.

Las propiedades principales de Flexbox serán las siguientes:

* Flex-direction
* flex-wrap
* Justify-conent
* Align-items.

A continuación se verá lo más importante sobre cada una de estas, pero antes, hay que saber que para la aplicación de estas propiedades, se debe implementar Flexbox en el CSS de nuestra página web, lo que solamente necesita la inserción de un contenedor o caja flex con estas líneas de código:

*.contenedor {*

*Display: flex*

*}*

Ahora sí, se irá con una descripción rápida de cada propiedad principal de Flexbox y sus funcionalidades.

**Flex-direction:** Flex-direction define el eje principal y secundario de los elementos dentro de un contenedor, estos ejes serán verticlaes (formando columnas) u horizontales (formando filas). Tendrá las subpropiedades row, row-reverse, column y column resverse, funcionando cada una de acuerdo a lo que menciona su nombre.

**Flex-wrap:** Cambiará la alineación de los elementos, que por defecto estarán un una misma línea, permitiendo ordenarlos de forma personalizada en más de una fila o columna. Para este ordenamiento se utilizan las subpropiedades nowrap, wrap y wrap reverse, en este caso nowrap como supone su nombre no implica ningún wrap, wrap activa el efecto de wrap y wrap reverse lo reliza hacia el otro lado.

**Justify-content:** Permite dar un direccionamiento en el eje horizontal a los elementos dentro de un contenedor, lo que se logra gracia a sus subpropiedades que llevarán de forma horizontal a direcciones distintas, estas son: stretch, que centra sin ajustar los elementos, flex-start, que los direccionará a la izquierda, center, que centrará pero ajustando los elementos, flex-end, que al contrario de flex-start direccionará a la derecha y junto a estas propiedades principales habrán tres (space-between, space around y space-evenly) que configurarán un espaciado entre los elementos, cada una de diferente manera.

**Align-items:** Su funcionamiento es similar al de Justify-content pero funcionará para el eje vertical en lugar del horizontal. En este caso sus sobropedidades efectuarán casi la misma función aunque de arriba abajo, estas son: flex-start que direccionará los elementos arriba, flex-end que por el contrario los direccionará abajo, center, que los centrará con un ajuste vertical, strecth, que los centrará sin dicho ajuste y baseline que direccionará y ajustará en base a una línea dentro del contenedor.

Como se pudo ver, cada propiedad tiene varias subpropiedades, que básicamente permiten direccionar de forma personalizada los elementos de un contenedor, estas subpropiedades se aplicarán en un CSS junto a la propiedad especificada:

*.contenedor {*

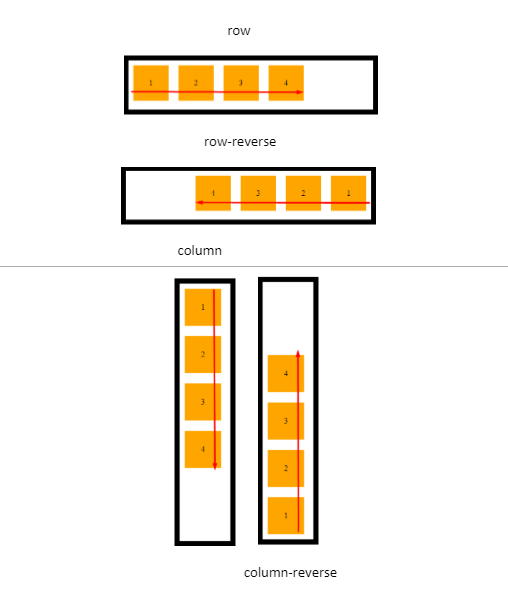
*Display: flex*

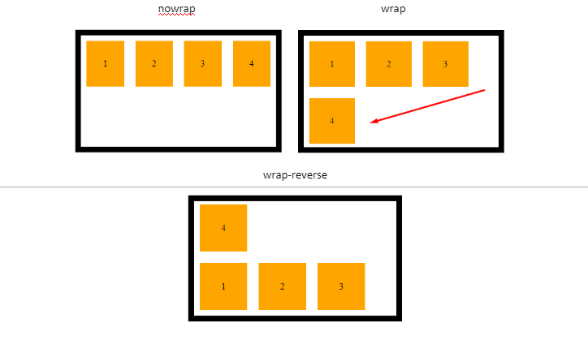
*[nombre de la propiedad]: [nombre de la subpropiedad]*

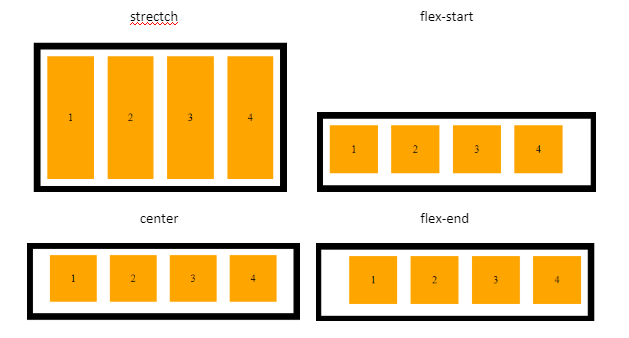
*}*

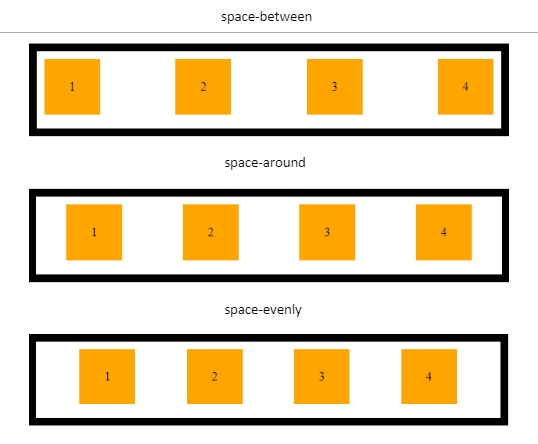
Una vez conocidas todas las propiedades, estas se pueden mezclar para dar lugar a distintas combinaciones que pueden tener los elementos que se coloquen dentro de un contenedor o caja con Flexbox, dotando así a nuestro sitio web de una interfaz amigable, simple y responsiva para todos los usuarios que lo visiten desde cualquier navegador o dispositivo.

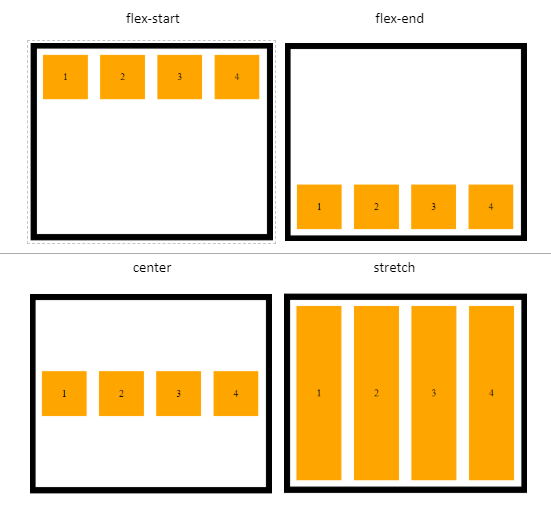
En la siguiente páginas las gráficas decada propiedad de Flexbox con sus subpropiedades.

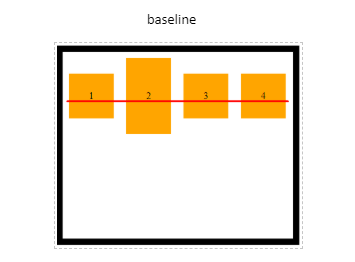


Arriba: subpropiedades de flex-direction, abajo: subpropiedades de flex-wrap.



**Propiedades de Justify-content.**



 **Propiedades de Align-item.**

1. **Consultar qué es CSS Grid y sus funciones**

Desde el inicio del desarrollo web, se ha intentado buscar una forma de plasmar y reproducir elementos en la web de forma sencilla y rápida, por lo cual se han buscado formas de mejorar los diseños que los primeros HTML y CSS ofrecían, debido a las altas limitaciones de HTML en solitario (hasta hoy; por eso las primeras páginas web destacan por su simplicidad) y de CSS en sus inicios, aunque desde la llegada de CSS el diseño web cambió radicalmente y se dieron desde finales de los 2000 páginas más conocidas a las de hoy en día (que continuaron evolucionando), lo cierto es que su forma de ajustar elementos en páginas ha sido mediante métodos de tablas, flotación, posicionamiento e inline block, los cuales tenían demasiadas limitaciones a la ahora de ajustar los elementos de una página web (lo que también se conoce como maquetado).

Como se habrá mencionado en el artículo sobre Flexbox, esta opción apareció como una solución para ordenar elementos en las páginas y es bastante útil habiendo revisado sus funciones, sin embargo Flexbox tiene un enfoque unidireccional, por lo que no será la mejor opción en el caso de requerir elemenetos en un diseño web más avanzado.

Estas limitaciones de CSS tradicional o Flexbox se tuvieron varios años hasta que finalmenet apareció CSS

**Imágenes:**

Flexbox: Patricio Suárez y SomosPNT.com

CSSGrid: PD

**Fuentes:**

https://somospnt.com/blog/148-flexbox-que-es-y-para-que-sirve

<https://developer.mozilla.org/es/docs/Web/CSS/CSS_flexible_box_layout/Basic_concepts_of_flexbox>

https://developer.mozilla.org/es/docs/Learn\_web\_development/Core/CSS\_layout/Flexbox