DISEÑO DE INTERFACES WEB UD-2

FLEXBOX

ÍNDICE

[FLEXBOX 3](#_Toc148075701)

[A. EJES DEL CONTENEDOR 4](#_Toc148075702)

[B. PROPIEDADES QUE AFECTAN AL CONTENEDOR 5](#_Toc148075703)

[1. display: flex 5](#_Toc148075704)

[2. flex-direction 5](#_Toc148075705)

[3. flex-wrap 7](#_Toc148075706)

[4. flex-flow 9](#_Toc148075707)

[5. justify-content 10](#_Toc148075708)

[6. align-items 11](#_Toc148075709)

[7. align-content 14](#_Toc148075710)

[C. PROPIEDADES QUE AFECTAN A LOS ITEMS 17](#_Toc148075711)

[1. order 17](#_Toc148075712)

[2. flex-grow 18](#_Toc148075713)

[3. flex-shrink 19](#_Toc148075714)

[4. flex-basis 20](#_Toc148075715)

[5. flex 20](#_Toc148075716)

[6. align-self 20](#_Toc148075717)

# FLEXBOX

A partir de CSS2 disponemos de contenedores que permiten disponer de una manera más automática a los elementos que están en su interior. Con el objetivo de simplificar el código ya que sólo habrá que configurar los contenedores para “decirle” cómo deben comportarse los elementos que se encuentran en su interior.

Se parte de un HTML tipo:

Texto

Descripción generada automáticamente

En este HTML vemos que el <section> que va a actuar de contenedor tiene en su interior 5 elementos <div>. Es conveniente que al elemento que va a hacer de contenedor se le asigne un atributo id para que, al hacer las reglas CSS, sea más fácil actuar sobre él. Aun así, en este ejemplo la regla CSS en lugar de ser #contenedor{} podría ser directamente section{} ya que sólo tenemos un elemento section en nuestra página.

El CSS base sobre el que partiremos será el siguiente:

Texto

Descripción generada automáticamente

Que produce el siguiente resultado en pantalla:

Imagen que contiene Interfaz de usuario gráfica

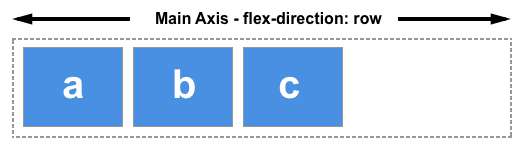
Descripción generada automáticamente

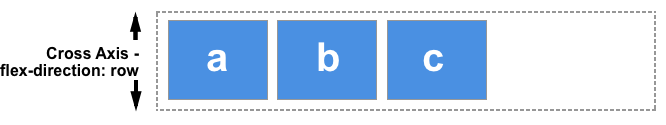
## EJES DEL CONTENEDOR

Los contenedores flexbox se ven afectados por dos ejes: el eje principal o main axis y el eje cruzado o cross axis que es siempre perpendicular al primero.

El eje principal viene definido por la propiedad flex-direction de manera que:

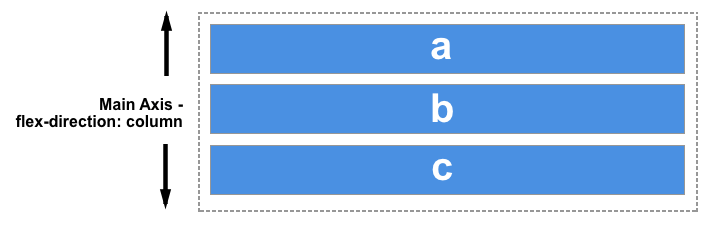
* En los valores row y row-reverse el eje principal será el “ancho” de la caja contenedora, mientras que su eje transversal será el “alto” de la caja:

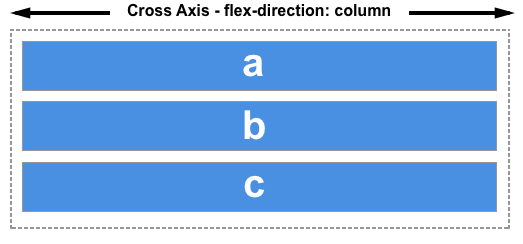




En el valor row el eje principal tiene dirección izquierda-derecha y en el valor row-reverse tiene la dirección contraria.

* En los valores column y column-reverse el eje principal será el “alto” de la caja contenedora:





En el valor column el eje principal tiene dirección arriba-abajo y en el valor column -reverse tiene la dirección contraria.

## PROPIEDADES QUE AFECTAN AL CONTENEDOR

1. display: flex

La propiedad base para que este tipo de posicionamiento es la propiedad display con el valor “flex”. Esta propiedad se debe de aplicar sobre el elemento que actúa como contenedor.

Texto

Descripción generada automáticamente

Si ahora vemos el resultado en pantalla veremos que la disposición de los elementos <div> ha cambiado:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

1. flex-direction

La propiedad flex-direction afecta a cómo se posicionan los elementos dentro del contenedor, esta propiedad acepta 4 valores, 2 para un alineamiento en fila y 2 para un alineamiento en columna:

* *flex-direction: row*. Es el valor por defecto, en este caso los elementos se posicionan en fila colocándose a la izquierda el primer elemento y a la derecha el último elemento.

Texto

Descripción generada automáticamenteInterfaz de usuario gráfica

Descripción generada automáticamente con confianza media

* *flex-direction: row-reverse*. En este caso los elementos se vuelven a colocar en fila pero el posicionamiento de estos es de derecha a izquierda, colocándose a la derecha el primer elemento del contenedor y a la izquierda el último.

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

Además, como se puede ver en la imagen, los elementos se colocarán lo más a la derecha posible dentro del contenedor.

* *flex-direction: column*. Este valor coloca los elementos en columna, situando en primer lugar al primer elemento del contenedor y en último lugar al último elemento.

Texto

Descripción generada automáticamente

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

* *flex-direction: column-reverse*. En este caso el valor vuelve a colocar los elementos en columna pero el orden es el contrario, arriba del todo aparecerá el último elemento del contenedor y en último lugar aparecerá el primero.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Al igual que ocurre con row-reverse si el contenedor tuviera una altura fija, los elementos se posicionarían lo más abajo posible dentro de la caja:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Imagen que contiene Texto

Descripción generada automáticamente

1. flex-wrap

Para entender esta propiedad vamos a modificar el HTML y vamos a incluir otros 5 elementos <div> dentro de nuestro contenedor.

Texto

Descripción generada automáticamente

En pantalla se pasará de esto:

Interfaz de usuario gráfica

Descripción generada automáticamente

A esto:

Interfaz de usuario gráfica

Descripción generada automáticamente

Lo que ha ocurrido es que el contenedor al no poder incluir a todos los elementos con sus reglas CSS predefinidas (en este caso con un width de 100px) ha repartido la anchura de la caja de forma equitativa entre los elementos.

Este es el comportamiento por defecto de flexbox pero podemos alterarlo, por ejemplo, obligándole a que los elementos mantengan sus propiedades base pero se alineen en diferentes filas (o columnas si estamos en un flex-direction: columna).

Esta propiedad admite 3 valores posibles:

* *flex-wrap: nowrap*. Es el valor por defecto y en este caso se realizará una modificación del ancho o alto de los elementos para que sigan entrando en la caja. Es lo que se ve más arriba.
* *flex-wrap: wrap*. En este caso los elementos se ordenarán en diferentes filas/columnas de arriba abajo (filas) y de izquierda a derecha (columnas):

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

En el caso de que fueran columnas:

Texto

Descripción generada automáticamente

Imagen que contiene Forma

Descripción generada automáticamente

* *flex-wrap: reverse*. Lo que va a hacer este valor es ordenar las filas/columnas de abajo arriba (filas) y de derecha a izquierda (columnas):

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

En el caso de que el alineamiento fuera en columnas:

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

1. flex-flow

Al igual que ocurre con otras propiedades CSS como border, margin, etc. que agrupan a otras propiedades en su interior, las propiedades flex-direction y flex-wrap se pueden agrupar en una única propiedad, la propiedad flex-flow.

En esta propiedad incluiremos ambos valores indicando primero la dirección del flex y, a continuación, si queremos que exista wrap o no.

Por ejemplo:

Texto

Descripción generada automáticamente con confianza mediaInterfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. justify-content

Esta propiedad nos va a permitir justificar los elementos contenidos dentro de un contenedor flex pudiéndolos colocar fácilmente al inicio, al final, en el centro, etc. del contenedor.

Afecta a cómo se distribuyen los elementos en el eje principal del contenedor.

Esta propiedad admite varios valores posibles:

* *justify-content: flex-start*. Es la propiedad por defecto. Coloca los elementos desde el inicio del eje principal. Por ejemplo:

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

* *justify-content: flex-end*. Coloca los elementos desde el final del eje principal. Por ejemplo:

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

* *justify-content: center*. Coloca los elementos en el centro del eje principal. Por ejemplo:

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

* *justify-content: space-between*. Distribuye los elementos de forma uniforme a lo largo del eje principal, colocando el primer elemento justo al inicio del eje y el último justo al final. Por ejemplo:

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

* *justify-content: space-around.* También distribuye los elementos de forma uniforme a lo largo del eje principal, pero en este caso todos los elementos tienen el mismo espacio alrededor. Por ejemplo:

Texto

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. align-items

Esta propiedad nos va a permitir alinear los elementos del contenedor a lo largo del eje cruzado de este.

Admite los siguientes valores:

* *align-items: stretch*. Es el valor por defecto. En el caso de que los ítems row no tengan una altura fijada, los elementos ocuparan el eje cruzado por completo:

Texto

Descripción generada automáticamenteGráfico, Gráfico de rectángulos

Descripción generada automáticamente

Si estuviéramos utilizando ítems columna lo que no deberían tener es el ancho fijado para que de esa manera los elementos ocuparan todo el ancho de la caja.

Texto

Descripción generada automáticamenteImagen que contiene firmar, calle, grande, verde

Descripción generada automáticamente

* *align-items: center*. Este valor nos va a colocar los elementos en el centro del eje cruzado. Por ejemplo, en el caso de un row:

Texto

Descripción generada automáticamente

Escala de tiempo

Descripción generada automáticamente

* *align-items: flex-start*. Los elementos se colocarán al inicio del eje cruzado (arriba en row, abajo en row reverse, izquierda en column y derecha en column-reverse). Por ejemplo:

Texto

Descripción generada automáticamente

Escala de tiempo

Descripción generada automáticamente con confianza media

* *align-items: flex-end*. Los elementos se colocarán al final del eje cruzado. Por ejemplo:

Texto

Descripción generada automáticamenteImagen que contiene Gráfico

Descripción generada automáticamente

* *align-items: baseline*. Los elementos en este caso se van a alinear por la base del texto, en el caso de que todos los elementos tengan el mismo tamaño de letra este valor funcionará como un flex-start.

Texto

Descripción generada automáticamente

Gráfico, Gráfico en cascada

Descripción generada automáticamente

1. align-content

Ajusta las líneas dentro de un contenedor flex cuando hay espacio extra en el eje transversal.

Esta propiedad no tiene efecto en cajas flexibles de una sola línea.

Admite los siguientes valores:

* *align-content: stretch*. Es el valor por defecto. En este caso las líneas de elementos son estiradas para utilizar el espacio sobrante de forma que el espacio libre en el contendor es dividido por igual entre todas las líneas.

Texto

Descripción generada automáticamente

Gráfico

Descripción generada automáticamente con confianza media

* *align-content: center*. En este caso las líneas de elementos son ajustadas hacia el dentro del contenedor haciendo que todas las líneas queden unidas entre sí y centradas dentro del contenedor (en el eje transversal). De esta forma el espacio entre el borde inicial del contenedor y la primera línea y el borde final del contenedor y la última línea es el mismo.

Texto

Descripción generada automáticamente

Escala de tiempo

Descripción generada automáticamente

* *align-content: flex-start*. Las líneas de elementos se sitúan en el borde inicial del contenedor de forma que se sitúan al inicio del eje transversal.

Texto

Descripción generada automáticamente

Escala de tiempo

Descripción generada automáticamente con confianza media

* *align-content: flex-end*. Las líneas de elementos se sitúan en el borde final del contenedor de forma que se sitúan al final del eje transversal.

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente con confianza media

* *align-content: space-between*. Las líneas son distribuidas de manera uniforme en el contenedor flexible. El espaciado se hace de modo que la separación entre cualquier par de elementos adyacentes sea el mismo. Los bordes transversales de inicio y de fin del contenedor son unidos a los bordes de la primera y última línea, respectivamente.

Texto

Descripción generada automáticamente

Escala de tiempo

Descripción generada automáticamente

* *align-content: space-around*. Las líneas son distribuidas uniformemente de modo que el espacio entre cualquier par de elementos adyacentes sea el mismo. El espacio vacío antes de la primera línea y el espacio después de la última es igual a la mitad del espacio entre cualquier par de líneas adyacentes.

Texto

Descripción generada automáticamente

Escala de tiempo

Descripción generada automáticamente

## PROPIEDADES QUE AFECTAN A LOS ITEMS

1. order

Con esta propiedad vamos a poder decidir el orden de inserción de los ítems en el contenedor. Por ejemplo, vamos a asignarle el order: -1 al item con el id “i2” y el order: 1 al item con el id “i3”.

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza baja Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Imagen que contiene verde, cuarto, grande, refrigerador

Descripción generada automáticamente

El elemento 2 se ha colocado en primer lugar y el elemento 3 se ha colocado en último lugar. Esto se produce porque por defecto todos los ítems tienen el valor 0 en su “order”.

En el caso de que 2 ítems tengan el mismo orden (sea el 0 u otro asignado) se respetará el orden establecido por el código HTML.

1. flex-grow

Especifica el factor de crecimiento de un elemento flexible, en su dirección principal. El factor de crecimiento especifica qué cantidad del espacio restante dentro del contenedor flexible, debería ocupar el ítem en cuestión.

Por defecto el valor es 0.

Si el número asignado es el mismo el reparto será equitativo:

Texto

Descripción generada automáticamente Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza baja Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

En este caso 1/3 (1+1+1) del espacio libre disponible va a cada uno de los elementos.

Si el número es diferente el reparto no es equitativo, por ejemplo:

Texto

Descripción generada automáticamente Texto

Descripción generada automáticamente Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza baja

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

En este caso el reparto es el siguiente 1/6 (1+3+2) del espacio restante se asigna al elemento 8, 3/6 se le asignan al elemento 9 y 2/6 se le asignan al elemento 10.

1. flex-shrink

Especifica el factor de contracción de un flex item. Los flex items se encogerán para llenar el contenedor de acuerdo con su número.

El valor por defecto es 1. Si un elemento obtiene el valor 0, ese elemento no se puede reducir.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

En este caso el elemento 1 tiene un ancho de 200px y con el shrink a 0 su tamaño no se ve reducido.

Por su parte el item 2 tiene un shrink a 2 lo que hace que su tamaño se reduzca al doble que los otros elementos.

1. flex-basis

Específica la base flexible, la cual es el tamaño inicial de un elemento flexible. Esta propiedad determina el tamaño de una caja de contenidos a no ser que se haya especificado de otra forma usando box-sizing.

Funciona como un width (en formatos fila) o un height (en formatos columna).

Texto

Descripción generada automáticamente

Gráfico

Descripción generada automáticamente con confianza baja

1. flex

La propiedad flex reúne las tres anteriores en una única, de forma que el primer valor sería el flex-grow, el segundo el flex-shrink y el último el flex-basis.

Por ejemplo:

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza baja

Interfaz de usuario gráfica, Texto, Chat o mensaje de texto

Descripción generada automáticamente

1. align-self

Permite la alineación de un elemento de forma independiente. Por ejemplo, partiendo de este alineamiento del contendor:

Texto

Descripción generada automáticamenteEscala de tiempo

Descripción generada automáticamente

Si añadimos la siguiente regla CSS:

Texto

Descripción generada automáticamente

El resultado obtenido fija al elemento 3 en el centro del eje transversal.

Imagen que contiene Gráfico

Descripción generada automáticamente