FLEXBOX CSS

Conceptos

Para empezar a utilizar FlexBox lo primero que debemos hacer es conocer algunos de los elementos básicos de este nuevo esquema, que son los siguientes:

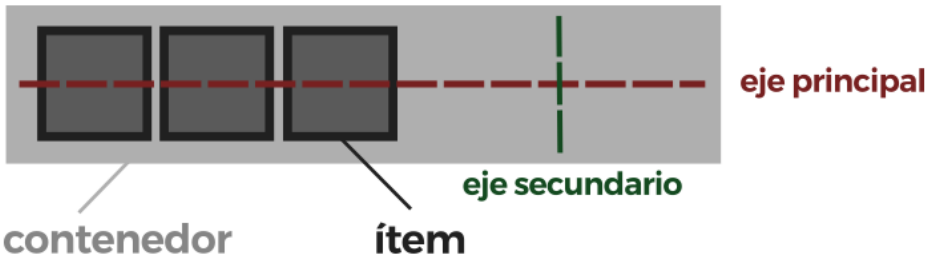

Imagen 1. Manz (2016). Elementos básicos del esquema [Imagen].

Contenedor: Es el elemento padre que tendrá en su interior cada uno de los ítems flexibles. Observa que al contrario que muchas otras estructuras CSS, por norma general, en Flex establecemos las propiedades al elemento padre.

- Eje principal: Los contenedores flexibles tendrán una orientación principal específica. Por defecto, es en horizontal (en fila).
- Eje secundario: De la misma forma, los contenedores flexibles tendrán una orientación secundaria, perpendicular a la principal. Si la principal es en horizontal, la secundaria será en vertical, y viceversa.

Ítem: Cada uno de los hijos flexibles que tendrá el contenedor en su interior.

Una vez tenemos claro esto, imaginemos el siguiente escenario

```
<div class="container"> <!-- Flex container -->
  <div class="item item-1">1</div> <!-- Flex items -->
  <div class="item item-2">2</div>
  <div class="item item-3">3</div>
</div>
```

Imagen 2. Manz (2016). Ejemplo práctico FlexBox [Imagen].

Para activar el modo FlexBox, hemos utilizado sobre el elemento contenedor la propiedad display que vimos en Tipos de elementos, y especificar el valor Flex o inline-flex (dependiendo de cómo queramos que se comporte el contenedor):

Tipo de elemento	Descripción
`inline-flex`	Establece un contenedor en línea, similar a `inline-block` (ocupa solo el contenido).
`flex`	Establece un contenedor en bloque, similar a `block` (ocupa todo el ancho del padre).

Imagen 3. Manz (2016). Tipo de elementos Flexbox [Imagen].

Por defecto, y sólo con esto, observaremos que los elementos se disponen todos sobre una misma línea. Esto ocurre porque estamos utilizando el modo FlexBox y estaremos trabajando con ítems flexibles básicos, garantizando que no se desbordarán ni mostrarán los problemas que, por ejemplo, tienen los porcentajes sobre elementos que no utilizan FlexBox.

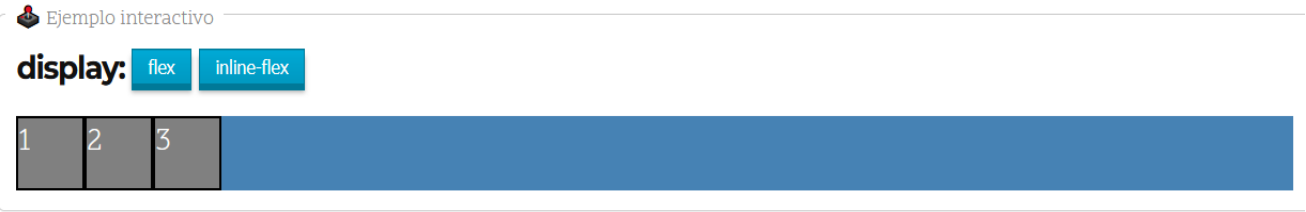

Imagen 4. Manz (2016). Display Flexbox [Imagen].

Dirección de los ejes

Existen dos propiedades principales para manipular la dirección y comportamiento de los ítems a lo largo del eje principal del contenedor. Son las siguientes:

Propiedad	Valor	Significado
`flex-direction`	**row** | row-reverse | column | column-reverse	Cambia la orientación del eje principal.
`flex-wrap`	**nowrap** | wrap | wrap-reverse	Evita o permite el desbordamiento (multilinea).

Imagen 5. Manz (2016). Propiedades Dirección de ejes-Flexbox [Imagen].

Mediante la propiedad flex-direction podemos modificar la dirección del eje principal del contenedor para que se oriente en horizontal (por defecto) o en vertical. Además, también podemos incluir el sufijo -reverse para indicar que coloque los ítems en orden inverso.

Valor	Descripción
`row`	Establece la dirección del eje principal en horizontal.
`row-reverse`	Establece la dirección del eje principal en horizontal (invertido).
`column`	Establece la dirección del eje principal en vertical.
`column-reverse`	Establece la dirección del eje principal en vertical (invertido).

Imagen 6. Manz (2016). Valor Flexbox [Imagen].

Esto nos permite tener un control muy alto sobre el orden de los elementos en una página. Veamos la aplicación de estas propiedades sobre el ejemplo anterior, para modificar el flujo del eje principal del contenedor:

```css
.container {
  background: steelblue;
  display: flex;
  flex-direction: column;
}

.item {
  background: grey;
}
```

Imagen 7. Manz (2016). Ejemplo práctico 2 Flexbox [Imagen].

A continuación podemos ver un ejemplo interactivo:

Imagen 8. Manz (2016). Flexbox, explicación en español del "nuevo" sistema de elementos flexibles de #CSS [Imagen].

Por otro lado, existe otra propiedad llamada flex-wrap con la que podemos especificar el comportamiento del contenedor respecto a evitar que se desborde (nowrap, valor por defecto) o permitir que lo haga, en cuyo caso, estaríamos hablando de un contenedor FlexBox multilínea.

Valor	Descripción
nowrap	Establece los ítems en una sola línea (no permite que se desborde el contenedor).
wrap	Establece los ítems en modo multilínea (permite que se desborde el contenedor).
wrap-reverse	Establece los ítems en modo multilínea, pero en dirección inversa.

Imagen 9. Manz (2016). Propiedad Flexbox CSS [Imagen].

Teniendo en cuenta estos valores de la propiedad flex-wrap, podemos conseguir cosas como la siguiente:

```css
.container {
  background: steelblue;
  display: flex;
  width: 200px;
  flex-wrap: wrap;  /* Comportamiento por defecto: nowrap */
}

.item {
  background: grey;
  width: 50%;
}
```

Imagen 10. Manz (2016). Ejemplo práctico 3 Flexbox [Imagen].

En el caso de especificar nowrap (u omitir la propiedad flex-wrap) en el contenedor, los 3 ítems se mostrarían en una misma línea del contenedor. En ese caso, cada ítem debería tener un 50% de

ancho (o sea, 100px de los 200px del contenedor). Un tamaño de 100px por ítem, sumaría un total de 300px, que no cabrían en el contenedor de 200px, por lo que FlexBox reajusta los ítems flexibles para que quepan todos en la misma línea, manteniendo las mismas proporciones.

Sin embargo, si especificamos wrap en la propiedad flex-wrap, lo que permitimos es que el contenedor se pueda desbordar, pasando a ser un contenedor multilínea, que mostraría el ítem 1 y 2 en la primera línea (con un tamaño de 100px cada uno) y el ítem 3 en la línea siguiente, dejando un espacio libre para un posible ítem 4.

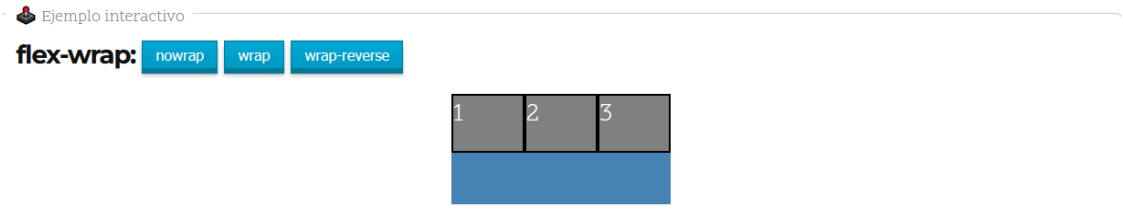

Imagen 11. Manz (2016). Flex-wrap [Imagen].

Atajo: Dirección de los ejes

Recuerda que existe una propiedad de atajo (short-hand) llamada flex-flow, con la que podemos resumir los valores de las propiedades flex-direction y flex-wrap, especificándolas en una sola propiedad y ahorrándonos utilizar las propiedades concretas:

```
.container {
  /* flex-flow: <flex-direction> <flex-wrap>; */
  flex-flow: row wrap;
}
```

Imagen 12. Manz (2016). Ejemplo práctico 4 Flexbox [Imagen].

Propiedades de alineación

Ahora que tenemos un control básico del contenedor de estos ítems flexibles, necesitamos conocer las propiedades existentes dentro de flexbox para disponer los ítems dependiendo de nuestro objetivo. Vamos a echar un vistazo a 4 propiedades interesantes para ello, la primera de ellas actúa en el eje principal, mientras que el resto en el eje secundario:

Propiedad	Valor	Eje
justify-content	**flex-start** | flex-end | center | space-between | space-around | space-evenly	1
align-content	flex-start | flex-end | center | space-between | space-around | space-evenly | **stretch**	2
align-items	flex-start | flex-end | center | **stretch** | baseline	2
align-self	**auto** | flex-start | flex-end | center | stretch | baseline	2

Imagen 13. Manz (2016). Propiedades de alineación Flexbox [Imagen].

De esta pequeña lista, hay que centrarse en primer lugar en la primera y la tercera propiedad, que son las más importantes (las otras dos son casos particulares que explicaremos más adelante):

- **justify-content**: Se utiliza para alinear los ítems del eje principal (por defecto, el horizontal).
- **align-items:** Usada para alinear los ítems del eje secundario (por defecto, el vertical).

Sobre el eje principal

La primera propiedad, justify-content, sirve para colocar los ítems de un contenedor mediante una disposición concreta a lo largo del eje principal:

Valor	Descripción
flex-start	Agrupa los ítems al principio del eje principal.
flex-end	Agrupa los ítems al final del eje principal.
center	Agrupa los ítems al centro del eje principal.
space-between	Distribuye los ítems dejando el máximo espacio para separarlos.
space-around	Distribuye los ítems dejando el mismo espacio alrededor de ellos (izq/dcha).
space-evenly	Distribuye los ítems dejando el mismo espacio (solapado) a izquierda y derecha.

Imagen 14. Manz (2016). Eje principal Flexbox [Imagen].

Con cada uno de estos valores, modificaremos la disposición de los ítems del contenedor donde se aplica, pasando a colocarse como se ve en el ejemplo interactivo siguiente (nótense los números para observar el orden de cada ítem):

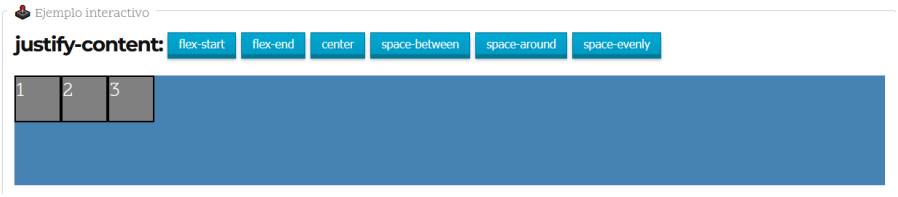

Imagen 15. Manz (2016). Justify Content Flexbox [Imagen].

Una vez entendido este caso, debemos atender a la propiedad align-content, que es un caso particular del anterior. Nos servirá cuando estemos tratando con un contenedor Flex multilínea, que es un contenedor en el que los ítems no caben en el ancho disponible, y por lo tanto, el eje principal se divide en múltiples líneas (por ejemplo, usando flex-wrap: wrap).

De esta forma, align-content servirá para alinear cada una de las líneas del contenedor multilínea. Los valores que puede tomar son los siguientes:

Valor	Descripción
flex-start	Agrupa los ítems al principio del eje principal.
flex-end	Agrupa los ítems al final del eje principal.
center	Agrupa los ítems al centro del eje principal.
space-between	Distribuye los ítems desde el inicio hasta el final.
space-around	Distribuye los ítems dejando el mismo espacio a los lados de cada uno.
stretch	Estira los ítems para ocupar de forma equitativa todo el espacio.

Imagen 16. Manz (2016). Align Content Flexbox CSS [Imagen].

Con estos valores, vemos como cambiamos la disposición en vertical (porque partimos de un ejemplo en el que estamos utilizando flex-direction: row, y el eje principal es horizontal) de los ítems que están dentro de un contenedor multilínea.

En el ejemplo siguiente, veremos que al indicar un contenedor de 200 píxeles de alto con ítems de 50px de alto y un flex-wrap establecido para tener contenedores multilínea, podemos utilizar la propiedad align-content para alinear los ítems de forma vertical de modo que se queden en la zona inferior del contenedor:

```
.container {
  background: #CCC;
  display: flex;
  width: 200px;
  height: 200px;

  flex-wrap: wrap;
  align-content: flex-end;
}

.item {
  background: #777;
  width: 50%;
  height: 50px;
}
```

Imagen 17. Manz (2016). Ejemplo práctico 5 Flexbox [Imagen].

Sobre el eje secundario

La otra propiedad importante de este apartado es align-items, que se encarga de alinear los ítems en el eje secundario del contenedor. Hay que tener cuidado de no confundir align-content con align-items, puesto que el primero actúa sobre cada una de las líneas de un contenedor multilínea (no tiene efecto sobre contenedores de una sola línea), mientras que align-items lo hace sobre la línea actual. Los valores que puede tomar son los siguientes:

Valor	Descripción
flex-start	Alinea los ítems al principio del eje secundario.
flex-end	Alinea los ítems al final del eje secundario.
center	Alinea los ítems al centro del eje secundario.
stretch	Alinea los ítems estirándolos de modo que cubran desde el inicio hasta el final del contenedor.
baseline	Alinea los ítems en el contenedor según la base del contenido de los ítems del contenedor.

Imagen 18. Manz (2016). Eje secundario Flexbox [Imagen].

Veamos un ejemplo interactivo con justify-content y align-items:

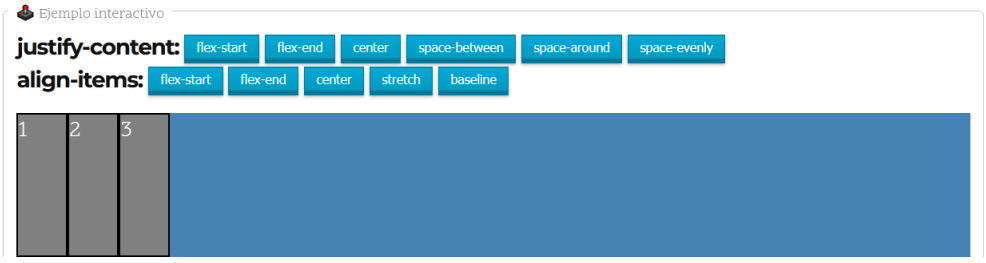

Imagen 19. Manz (2016). Ejemplo Interactivo Flexbox CSS [Imagen]

Por otro lado, la propiedad align-self actúa exactamente igual que align-items, sin embargo es la primera propiedad de flexbox que vemos que se utiliza sobre un ítem hijo específico y no sobre el elemento contenedor. Salvo por este detalle, funciona exactamente igual que align-items.

Gracias a ese detalle, align-self nos permite cambiar el comportamiento de align-items y sobreescribirlo con comportamientos específicos para ítems concretos que no queremos que se comporten igual que el resto. La propiedad puede tomar los siguientes valores:

Valor	Descripción
flex-start	Alinea los ítems al principio del contenedor.
flex-end	Alinea los ítems al final del contenedor.
center	Alinea los ítems al centro del contenedor.
stretch	Alinea los ítems estirándolos al tamaño del contenedor.
baseline	Alinea los ítems en el contenedor según la base de los ítems.
auto	Hereda el valor de **align-items** del padre (si no se ha definido, es **stretch**).

Imagen 20. Manz (2016). Propiedad Align self Flexbox [Imagen].

Si se especifica el valor auto a la propiedad align-self, el navegador le asigna el valor de la propiedad align-items del contenedor padre, y en caso de no existir, el valor por defecto: stretch. Veamos un ejemplo para verlo en funcionamiento:

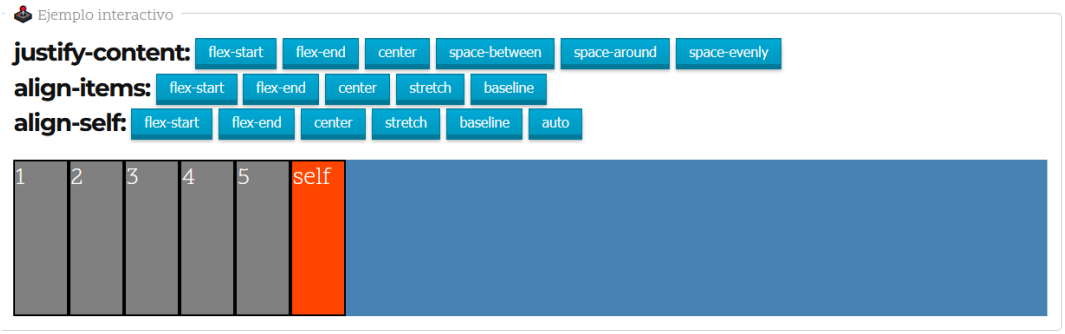

Imagen 21. Manz (2016). Flexbox [Imagen].

Atajo: Alineaciones

Existe una propiedad de atajo con la que se pueden establecer los valores de align-content y de justify-content de una sola vez, denominada place-content:

```
.container {
  display: flex;
  place-content: flex-start flex-end;

  /* Equivalente a... */
  align-content: flex-start;
  justify-content: flex-end;
}
```

Imagen 22. Manz (2016). Ejemplo práctico 6 Flexbox [Imagen].

Consejo: para profundizar mejor sobre propiedades de hijos, Huecos (gaps) y orden de los Items, consulta la bibliografía a continuación.

R**EFERENCIAS BIBLIOGRAFICAS**

Manz. (2021). Flexbox CSS. Recuperado de Lenguaje CSS: https://lenguajecss.com/css/maquetacion-y-colocacion/flexbox/