

FLEXBOX

CONCEPTOS BÁSICOS

Diseño de Interfaces web

Ciclo de Desarrollo de Aplicaciones Web

INDICE

- 1. ¿Qué es flexbox?
- 2. Maquetar usando flexbox
- 3. Otras propiedades de flexbox

1. ¿QUÉ ES FLEXBOX?

- “Flexible Box Layout” (“Diseño de caja flexible”) nos aporta una solución para los desarrollos “responsive”.
- Permite crear un conjunto de elementos flexibles que se adaptan automáticamente a su contenedor y con el que podemos controlar parámetros como la alineación, dirección (horizontal/vertical), ajuste de la fila según tamaños, etc.

... COMPATIBILIDAD DE FLEXBOX

- Como siempre podemos mirar la compatibilidad en los distintos navegadores en
- <https://caniuse.com/#search=flexbox>

... FLEXBOX EN W₃C

- Podemos ver el manual de flexbox en w3c en
- <https://www.w3.org/TR/css-flexbox-1/>

... COMO DISPONER LOS ELEMENTOS

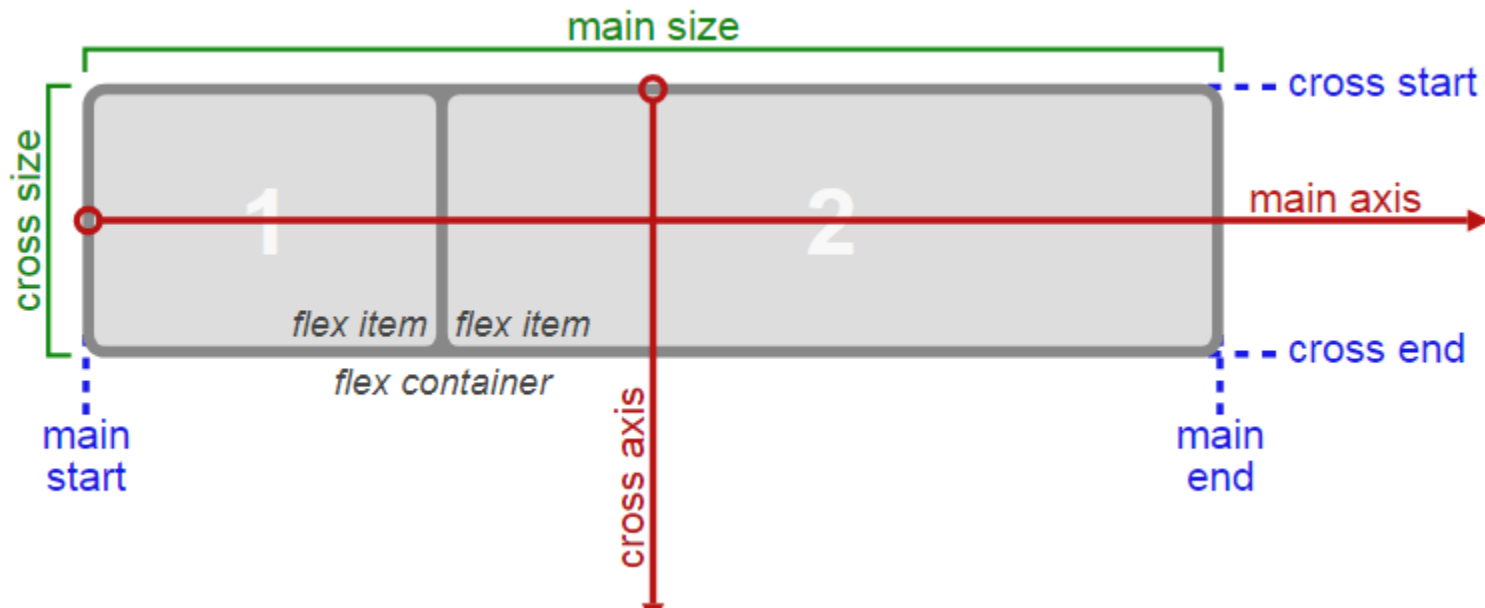
- Existen cuatro modos de disponer los elementos, tres de presentación y otro de posición:
 - – Elementos en línea (`display:inline`)
 - – Elementos en bloque (`display:block`)
 - – Elementos en tabla (`display:table`)
 - – Tipos de posición
(`position:absolute/relative/fixed`)

... COMO DISPONER LOS ELEMENTOS EN FLEXBOX

- Flexbox es una mezcla de todos ellos en cuanto a como afecta a la disposición de una estructura de elementos contenidos en una capa padre o contenedor, y se define como “flex” en la propiedad “display” (`display: flex`).
- Podemos crear una estructura de elementos en línea similar a una tabla, o hacer que funcionen como un bloque y en orden inverso en la siguiente resolución, todo ello a la vez que se adapta automáticamente al tamaño que necesitamos, tanto en anchura como en altura.

FLUJO FLEXIBLE

- El W3C define a esta estructura como un “flujo flexible” de elementos en dirección (arriba/abajo/izquierda/derecha) y tamaño (anchura/altura) según los ejes principal (horizontal) y transversal (vertical).



2. MAQUETAR USANDO FLEXBOX

- Para poder probar todas las posibilidades que ofrece Flexbox CSS, vamos a crear una capa `class="contenedor-flex"` que hará de padre y un número impar de capas `class="elemento"` numeradas.

- Para los ejemplos nos viene bien que sea un número impar para ver tanto la ordenación como la adaptación y su resultado, y conviene también que sea un número alto (>5) para ver un listado de elementos no muy corto.

Propiedades del Contenedor de

FLEXBOX

Propiedad	Descripción	Ejemplo
<u>align-content</u>	Modifica el comportamiento de la propiedad flex-wrap. Es similar a alinear elementos, pero en lugar de alinear elementos flexibles, alinea líneas flexibles	<code>align-content:stretch; align-content:center; align-content:flex-start; align-content:flex-end; align-content:space-between; align-content:space-around;</code>
<u>align-items</u>	Alinea verticalmente los elementos flexibles cuando los elementos no utilizan todo el espacio disponible en el eje transversal	<code>align-items:stretch; align-items:center; align-items:flex-start; align-items:flex-end; align-items:baseline;</code>
<u>display</u>	Especifica el tipo de cuadro utilizado para un elemento HTML	<code>display: flex;</code>
<u>flex-direction</u>	Especifica la dirección de los elementos flexibles dentro de un contenedor flexible	<code>flex-direction:row; flex-direction:row-reverse; flex-direction:column; flex-direction:column-reverse;</code>
<u>flex-flow</u>	Una propiedad abreviada para flex-direction y flex-wrap	<code>Flex-flow: row wrap;</code>
<u>flex-wrap</u>	Especifica si los elementos flexibles se deben envolver o no, si no hay suficiente espacio para ellos en una línea flexible	<code>flex-wrap: wrap; flex-wrap: nowrap; flex-wrap: wrap-reverse;</code>
<u>justify-content</u>	Alinea horizontalmente los elementos flexibles cuando los elementos no utilizan todo el espacio disponible en el eje principal	<code>justify-content:flex-start; justify-content:flex-start; justify-content:center; justify-content:space-between; justify-content:space-around;</code>

EJ 1

- Vamos a ver un ejemplo `flex-wrap` con `wrap` y otro con `nowrap`
- Aprovechamos para ver el uso de variables en CSS.
- Y también vemos cómo hacer cálculos usando `calc()` para poner el porcentaje de la anchura de los elementos.

PROPIEDAD FLEX-DIRECTION

- Vamos a ver la propiedad “flex-direction“, que puede tomar 4 valores y se aplica al padre (contenedor):
- `flex-direction: row;` -> Los elementos se visualizan de izquierda a derecha (valor por defecto, similar al ejemplo 1)
- `flex-direction: row-reverse;` -> Los elementos se visualizan de derecha a izquierda.
- `flex-direction: column;` -> Los elementos se visualizan de arriba hacia abajo.
- `flex-direction: column-reverse;` -> Los elementos se visualizan de abajo hacia arriba.

EJ 2

- Probamos
- `flex-direction: row;`
- `flex-direction: row-reverse;`
- `flex-direction: column;`
- `flex-direction: column-reverse;`

PROPIEDAD FLEX-WRAP

- Su valor afecta a cómo se distribuyen los elementos en fila y, por consiguiente, a su tamaño. Los posibles valores son:
- `flex-wrap: nowrap;` -> Los elementos se muestran en línea, en una sola fila, y su tamaño se ajusta al contenedor siempre y cuando la suma de todos ellos sea mayor o igual que el 100% de la anchura del contenedor. Si es inferior, se siguen mostrando en línea pero conservan su tamaño. Este es el valor por defecto, y como veíamos en el ejemplo 1, aunque la anchura de los elementos es el 25% del contenedor, todos se muestran en línea modificando su tamaño para que la suma total no sea superior al 100% de su contenedor.
- `flex-wrap: wrap;` -> Los elementos se muestran en línea, pero si su anchura supera la del contenedor, se distribuyen en varias filas.
- `flex-wrap: wrap-reverse;` -> Los elementos se muestran en línea, pero si su anchura supera la del contenedor, se distribuyen en varias filas, y además lo hacen en orden inverso al de maquetación.

EJ 3

- Ya vino en el ej 1 ejemplos de wrap y nowrap.
- Ahora vamos a ver el wrap-reverse
- `flex-wrap:wrap-reverse;`
- Este ejemplo de ordenación de Flexbox es muy curioso, ya que, además de realizar la adaptación de tamaños por filas, realiza su ordenación a la inversa, de abajo hacia arriba, y en este caso de izquierda a derecha (que es el sentido por defecto).
- Podemos modificar el sentido de la orientación horizontal si además utilizamos “flex-direction:row-reverse” como podemos ver en el siguiente ejemplo:
- `flex-wrap:wrap-reverse;`
- `flex-direction:row-reverse;`

Prueba a utilizar flex-flow que
utiliza es una propiedad abreviada
para flex-direction y flex-wrap

ALINEACIÓN HORIZONTAL

- En cuanto a la alineación horizontal de los elementos en Flexbox, encontramos la propiedad “**justify-content**”, que alinea los elementos a lo largo del eje principal (main axis) de su contenedor.
- Esto lo veremos más claramente con los siguientes ejemplos tras la definición de sus valores más utilizados:
- – **justify-content: flex-start;** -> Alinea los elementos en horizontal desde el inicio de la dirección del eje principal de su contenedor (partiendo desde el inicio de la línea). Este es el **valor por defecto**. Es importante destacar que, como veremos más adelante, la dirección establecida en “flex-direction” afecta a la alineación.
- – **justify-content: flex-end;** -> Alinea los elementos en horizontal desde el final de la dirección del eje principal de su contenedor (partiendo desde el final de la línea)
- – **justify-content: center;** -> Alinea los elementos al centro del eje principal de su contenedor. Similar a un texto alineado al centro.
- – **justify-content: space-between;** -> Alinea los elementos justificándolos a lo largo del eje principal de su contenedor. Similar a un texto justificado. Los elementos laterales se pegan a los extremos y el resto se distribuyen a lo largo del eje principal dejando el mismo espacio entre ellos.
- – **justify-content: space-around;** -> Alinea los elementos distribuyendo sus centros de forma horizontal a lo largo del eje principal de su contenedor, dejando el mismo espacio lateral de separación al comienzo, al final y entre ellos.

EJ 4

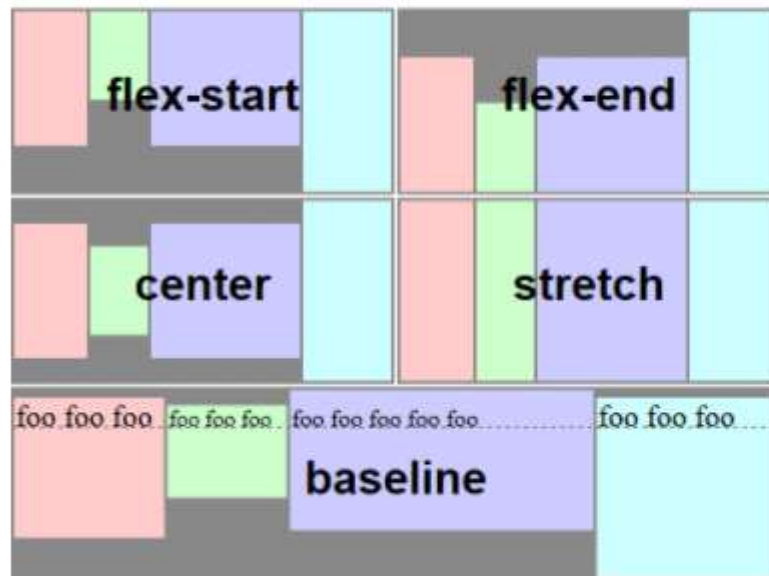
- Esta propiedad como mejor se comprende es mediante un ejemplo,
- `justify-content`

ALINEACIÓN VERTICAL

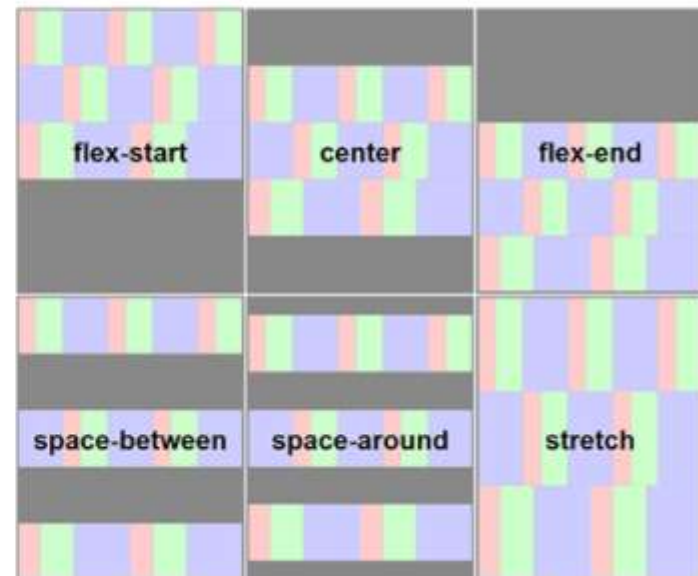
- La alineación vertical se realiza a través del llamado “eje transversal” (cross axis), y para ello contamos con tres propiedades diferentes, “align-items”, “align-self” y “align-content”.
- – “align-items” establece la alineación predeterminada para todos los elementos del contenedor, incluidos los elementos independientes.
- – “align-self” permite alinear elementos independientes del contenedor.
- – “align-content” alinea las líneas/filas de elementos de un contenedor.

...ALINEACIÓN VERTICAL

align-items / align-self



align-content



...ALINEACIÓN VERTICAL

- Por lo tanto, “align-items” nos sirve para alinear los elementos y “align-content” para alinear las filas de estos, mientras que “align-self” nos permite alinear elementos de forma independiente. Sabiendo esto, podemos deducir que “align-content” solo funciona cuando tenemos más de una fila de elementos.
- En las siguientes diapositivas vamos a ver los valores que pueden tomar cada una de ellas (los valores de “align-items” son validos para “align-self”):

...ALINEACIÓN VERTICAL align-items

Valor de align-items	Descripción
stretch	Valor por defecto. La altura de los elementos se ajusta al tamaño del contenedor (o fila), dividiendo el espacio sobrante entre todos los elementos por igual.
center	Alinea al centro vertical los elementos a lo largo del eje transversal de su contenedor.
flex-start	Alinea en vertical los elementos desde el inicio de la dirección del eje transversal de su contenedor
flex-end	Alinea en vertical los elementos desde el final de la dirección del eje transversal de su contenedor
baseline	Alinea en vertical las “líneas base” de los elementos a lo largo del eje transversal de su contenedor.
initial	Pone la propiedad en su valor inicial
inherit	Hereda de su padre

EJ5 Y EJ6

- `align-items`

EJ5

Ejemplos para

`stretch`

`center`

`flex-start`


`flex-end`

EJ6

(el `baseline` va en este otro ejemplo)

Ponemos estilos diferentes a los elementos para poder apreciarlo. Como podemos observar, lo que se alinea en vertical es la línea base del contenido de cada elemento, en este caso, la línea base de cada número, aunque entre ellos tengan alturas de línea o tamaños diferentes.

...ALINEACIÓN VERTICAL align-content

Valor de align-items	Descripción
stretch	Las líneas se ajustan al tamaño del contenedor. Es el valor por defecto.
center	Las líneas se ponen en el centro del contenedor
flex-start	Las líneas se ponen al principio del contenedor
flex-end	Las líneas se ponen al final del contenedor
space-between	Las líneas se distribuyen uniformemente en el contenedor flexible 
space-around	Las líneas se distribuyen uniformemente en el contenedor flexible, con espacios de la mitad del tamaño en cada extremo
space-evenly	las líneas están distribuidas uniformemente en el contenedor flexible, con el mismo espacio alrededor de ellas

EJ 7

- Esta propiedad como mejor se comprende es mediante un ejemplo,
- `align-content`

...Para un centrado perfecto

Establezca las propiedades `justify-content` y `align-items` en el centro, y el elemento flexible estará perfectamente centrado.

Pruébalo.

3. OTRAS PROPIEDADES DE FLEXBOX

- **order:** -> Posiciona un elemento en el orden asignado por el número entero especificado en la propiedad, teniendo en cuenta el valor asignado al resto de elementos, cuyo valor por defecto es 0 (cero). Esto significa que, a no ser que se especifique un orden a todos los elementos, cualquier número aplicado a un solo elemento lo llevará a la última posición. Esto es debido a que por defecto todos tienen “cero”, por lo que si asignamos por ejemplo “order:3” a cualquier elemento, este se colocará al final del listado.
- Ejemplo:

```
.contenedor{
    display:flex;
}
.elemento:nth-child(even) {
    order:1;
}
```
- **Ejemplo 8**
- Con este ejemplo podemos entender mejor el funcionamiento de esta propiedad, y es que estamos asignando el orden/posición “1” a todos los elementos pares, que se sitúan tras los elementos impares, que tendrían por defecto el valor “0”.

PROPIEDAD FLEX

- “flex” que se aplica a los elementos del contenedor y que a su vez se estructura de tres propiedades diferentes y que se pueden presentar por separado, “flex-grow”, “flex-shrink” y “flex-basis”, y se define de la siguiente forma:
- `flex: none | <flex-grow> <flex-shrink> <flex-basis>`

PROPIEDAD FLEX

- `- flex-grow: <número>;` -> Determina el factor de incremento de tamaño de uno o varios elementos en relación al resto. Cuando este valor se omite, se establece en 0.
- `- flex-shrink: <número>;` -> Determina el factor de disminución de tamaño de uno o varios elementos en relación al resto teniendo en cuenta el espacio “vacío” o “libre” entre ellos. Cuando este valor se omite, se establece en 1.
- `- flex-basis: auto | 0 | <width>` -> Especifica el tamaño principal inicial de los elementos. Cuando este valor se omite, se establece en 0.

flex-grow

- La propiedad `flex-grow` especifica el factor de crecimiento de un elemento flexible (que tiene asignado `display: flex`), en su dirección principal. El factor de crecimiento especifica qué cantidad del espacio restante dentro del contenedor flexible, debería ocupar el ítem en cuestión.
- La dirección principal puede ser la altura o el ancho del elemento, dependiendo del valor de `flex-direction`.
- El espacio restante es el tamaño del contenedor flexible menos el tamaño de todos los elementos flexibles juntos. Si todos los ítems dentro del contenedor tienen el mismo factor de crecimiento, todos los elementos reciben la misma cantidad del espacio restante. De lo contrario, el espacio restante se distribuye en función de los diferentes factores de crecimientos de cada ítem.
- Veamos el **EJEMPLO 9**

CON MEDIA QUERIES

- Con media queries exige cambiar muy poco código:
- <https://codepen.io/estelle/pen/brDpB>

RESUMEN

- Conocer la maquetación usando flexbox
- Flexbox es el complemento perfecto para la maquetación de diferentes módulos de contenido en lista y mediante pequeñas propiedades poder cambiar su aspecto, distribución, tamaño y alineación para que se adecúe sin problemas al dispositivo donde lo estemos visualizando.

BIBLIOGRAFÍA

- <https://www.yunbitsoftware.com/blog/2017/03/30/flexbox-css3-tutorial-descripcion-ejemplos-html/>

FIN DEL CAPÍTULO.



- Y ahora ¡te toca a ti!
- Profundiza en los conceptos vistos en el tema.
- Realiza las actividades propuestas.
- Amplia información con Internet y la bibliografía.