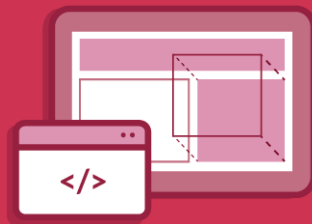


# Diseño WEB

## Clase 6

### Box Modeling II



# Posiciones



## Position

Es una propiedad CSS pensada para ubicar un elemento en una coordenada puntual. Puede ser en base al BODY o al elemento que contiene el objeto posicionado. Se suele usar position (entre otras cosas) para:

- Superponer elementos (como las ventanas modales).
- Crear barras de herramientas fijas (como facebook).
- Crear publicidades que te sigan con el scroll.
- Hacer un menú con submenú adentro.

## Casos en que no se usa...

No se usa position cuando podés lograr lo mismo con float, display o margin.

- ¿Querés el div que contiene toda la web esté centrado horizontalmente con respecto al ancho de la pantalla?  
Dale un ancho fijo y *margin: auto;*



- ¿Querés hacer columnas?  
Dale *display: inline-block* o *float: left*
- ¿Querés hacer una botonera mostrando cada botón uno al lado del otro?  
A los botones, dale *display: inline-block*;

### Cómo ubicar un elemento.

1. En primer lugar hay que definir qué tipo de posición se quiere usar.
2. Luego hay que indicarle las coordenadas en las que se posiciona ese objeto.
3. Se hace indicando al objeto un valor numérico para las propiedades *top*, *bottom*, *left*, *right*.
4. No se puede indicar dos propiedades opuestas a la vez (un objeto no puede estar en *top* y *bottom*).

### ¿Qué position existen?

Hay cuatro valores posibles para el position.

- *fixed*: Los bordes serán calculados en base al browser. Si el usuario hace scroll, el objeto sigue fijo.

```
#caja {
  position: fixed;
  top: 20px;
  right: 10px;
}
```

- *absolute*: Los bordes serán cualquier elemento contenedor que tenga una position relative. Si no encuentra ninguno, usa el body. Si se hace scroll, el objeto posicionado se mueve.

```
#caja {
  position: absolute;
  top: 20px;
  right: 10px;
}
```

- *relative*: Sólo delimita los límites de cualquier elemento hijo con position absolute.

```
#caja {
```



```
position: relative;
top: 5px;
right: 12px;
}
```

- *static*: Se usa para quitar un position

```
#caja {
  position: static;
}
```

## Menú con submenú

El position (tanto *relative* como *absolute*) se usa (entre otras) para **hacer un menú que tenga un submenú emergente**. Los ítems del primer menú son relativos (sirven como borde de cualquier hijo). La lista dentro de un list-item es absoluta. Por defecto la sub-lista tiene *display: none*. Recién cuando un list-item detecta el *:hover*, si adentro tiene una lista, darle *display: block*



## Propiedad z-index

El z-index entra en juego cuando dos elementos que tienen *position* se superponen. Esta propiedad acepta como valor un número (sin ninguna unidad, ni px, ni cm, ni nada). **A valor más alto, se mostrará por encima de los demás elementos.**



Por defecto todos los objetos tienen z-index:1. Si dos objetos tienen el mismo valor de z-index y se superponen, el que fue creado después en el HTML se verá encima del otro.

### **Position sticky (pegajoso)**

Es un nuevo valor del *position* que **mezcla un elemento sin *position* con uno *fixed*.**

Al igual que los demás, usa coordenadas *top*, *bottom*, *left* y *right*, pero se comporta distinto:

- Mientras el objeto se encuentre en una ubicación con valor superior al indicado, se moverá junto con el scroll.
- Cuando el objeto esté por debajo de ese valor (por fuera de los bordes), quedará fijo, como si estuviese *fixed*.

Ideal, por ejemplo, para un menú con anclas, si se tiene demasiado scroll vertical.





# Desafío

- Agregar un menú con las secciones al desafío anterior, en el que se reprodujo este mockup:



A la sección “Personajes” agregar sub-menú con cada uno de los personajes dentro. Textos:

[https://es.wikipedia.org/wiki/Star Wars: Episodio VIII - Los %C3%BAltimos Jedi#Personajes](https://es.wikipedia.org/wiki/Star_Wars:_Episodio_VIII_-_Los_%C3%BAltimos_Jedi#Personajes)

(Duracion : 30 min)





# Desafío

- Agregar un menú a nuestro proyecto con las distintas secciones del sitio de banda. *Ejemplo: Discografía, Integrantes, Merchandising, Tours.* Para cada sección agregar sub menús con los contenidos correspondientes.

(Duracion : 60 min)





# ¿Que es Flexbox?



Flexbox es un modo de diseño que nos permite **crear estructuras para sitios web de una forma más fácil**. Si ya sabes de HTML y CSS probablemente alguna vez habrás visto que los sitios web se realizan utilizando la propiedad float, para desplazar contenedores. **Con Flexbox ya no necesitaras usar float para posicionar tus elementos**, al contrario, con Flexbox podrás posicionar y distribuir los elementos como tú quieras.

**Puedes posicionar un elemento en la posición que desees horizontalmente y por si fuera poco también en forma vertical.** Posicionar elementos de forma vertical no es tarea fácil sin Flexbox, incluso muchas veces teníamos que hacer uso de Javascript para hacer cálculos y saber cuánto margen poner a un elemento para centrarlo verticalmente.

No solo puedes posicionar elementos vertical y horizontalmente, sino que puede establecer como se distribuirán, el orden que tendrán e incluso el tamaño que tendrán





en proporción a otros elementos. **Esto es perfecto para crear diseños adaptables a dispositivos móviles** (Responsive Design).

Flexbox no es una propiedad ni un conjunto de propiedades. **Flexbox es un nuevo modelo de layout que viene a incorporarse a los ya existentes en css (se está trabajando actualmente en nuevo modelo: el grid layout):**

- **Block** (los elementos aparecen uno debajo de otro ocupando todo el ancho disponible)
- **Inline** (los elementos aparecen uno al lado del otro en una línea y saltan a la línea siguiente al ocupar el espacio disponible)
- **Table** (los elementos imitan la distribución de una tabla HTML, con filas, encabezados y columnas).
- **Positioned** (los elementos pueden romper el flujo y posicionarse en cualquier lugar del documento).

**Un modelo de layout es un set de algoritmos que determinan el tamaño y la posición de los elementos con respecto a sus hermanos y ancestros.**

## ¿Qué se puede hacer con flexbox?

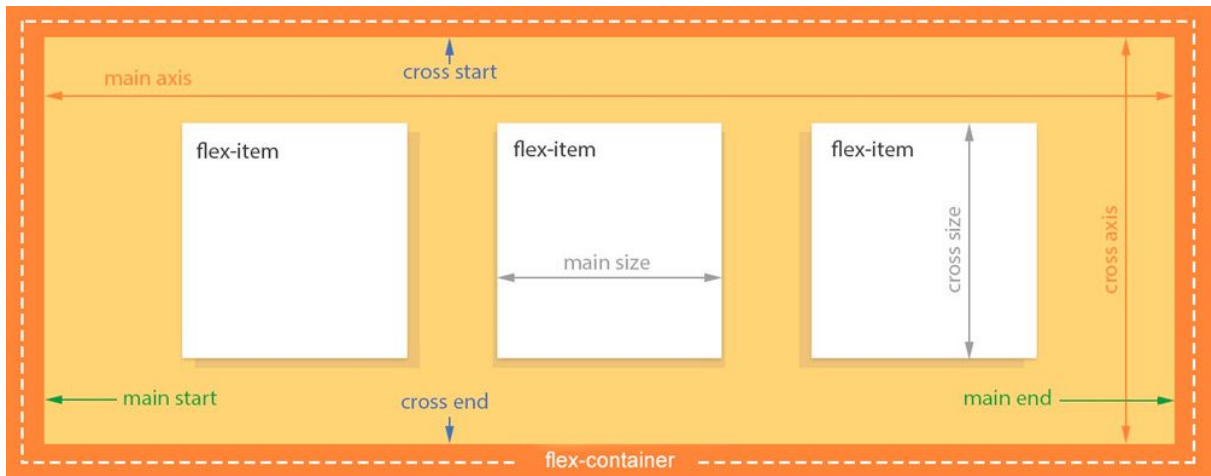
Entendiendo lo que significa un modelo de layout, con flexbox podemos hacer lo siguiente:

- Distribuir los elementos en sentido vertical u horizontal.
- Reordenar la aparición de los elementos sobreescribiendo su aparición en el navegador.
- Ajustar dinámicamente las dimensiones de los elementos para evitar desbordamientos (overflow) respecto a su padre.
- Redefinir el sentido del flujo de los elementos (hacia arriba, hacia abajo, hacia la izquierda o hacia la derecha).
- Alinear los elementos respecto al padre o respecto a sus hermanos.

## Conceptos básicos de Flexbox

Para entender bien este modelo de Layout debemos de entender algunos conceptos básicos. Primero la disposición flex debe de estar constituido por elementos padres e hijos, **el padre será el contenedor Flexible “flex container” y los hijos inmediato serán los elementos Flexibles “flex item”.**





En la imagen anterior vemos como la W3C nos presenta gráficamente el modelo flexbox con sus diferentes propiedades y su pensado funcionamiento