

# JavaScript Basic

### Guía #3: Multimedia Interactive Web Application

Esta guía hace parte del curso mencionado en el título y les permitirá estudiar con más detenimiento los conceptos de clase, además de servir como instrumento de la misma. La guía contará con código base, el desarrollo conceptual de los temas y enlaces externos para el estudio independiente.

## 1. Repasando el concepto de cajas

Todos los elementos dentro de HTML son cajas con comportamientos peculiares, por ejemplo una div es una caja de tipo **bloque**, mientras un span es una caja de tipo **línea**.

### Lectura 1: Box Model

[Artículo de CSS-Tricks](https://css-tricks.com/the-css-box-model/)

* Concepto de caja
* Posicionamiento de cajas

### Lectura 2: Box Sizing

[Artículo de CSS-Tricks](https://css-tricks.com/box-sizing/)

* Tipos de cálculo de tamaño de casa
* Trucos de reseteo

**Dev Tricks:** Si quieres ver las cajas de una aplicación podrías adicionar esta regla a tus estilos:

\*{  
 outline: solid red;  
 }

## 2. Construyendo un reproductor de audio

Trabajaremos un reproductor de audio basado en el diseño propuesto por [@Julie](https://codepen.io/juliepark) y que podrán ver en [el siguiente codepen](https://codepen.io/juliepark/pen/YLPBNv):



Que después de haber leído las lecturas recomendadas claramente visualizamos de la siguiente manera:



Para esta oportunidad trabajaremos con **FlexBoxes**, para apoyarnos en una paleta de color podremos buscar opciones [acá](https://coolors.co/browser/latest/4), en este caso para el ejercicio trabajaremos con [esta paleta](https://coolors.co/630017-ce001e-e0e0e0-2d2d30-161616). La distribución del color se vería algo así:



Cabe resaltar que en ese ejemplo aun falta mucha GUI por hacer, y vamos a desarrollarlo presencialmente, pero al final todo reposará en el repositorio del curso.

### Iconos

Para los iconos del reproductor usaremos la versión gratuita local de [Font Awesome](https://fontawesome.com/), cabe resaltar que si el proyecto no requiere estar offline, la versión provista por el CDN. Podremos descargar los paquetes desde [acá](https://fontawesome.com/download)

## 3. Comprendiendo las APIs de Audio y Video

Antes de iniciar es bueno que tengamos clara la diferencia entre la etiqueta y el API pues mientras una hace referencia al elemento desde HTML, la otra hace referencia a la lógica del mismo.

### [HTML Audio Tag](https://www.w3schools.com/tags/tag_audio.asp)

**Nota:** La etiqueta Audio responde también a los [atributos globales de HTML](https://www.w3schools.com/tags/ref_standardattributes.asp) y [Atributos globales de los elementos](https://www.w3schools.com/tags/ref_eventattributes.asp).

### [HTML Audio & Video DOM Reference](https://www.w3schools.com/tags/ref_av_dom.asp)

## 4. Creando nuestro reproductor

Para nuestro caso crearemos una clase que permita crear instancias de reproductores y reciba como parametro elementos del DOM para hacer las veces de reproductor, controles, carátula, etc.

### Definición de los elementos:

El reproductor en cuestión contará con los siguientes elmentos:

* **album Cover**: Elemento que contiene la imágen del álbum
* **current time**: Tiempo actual de la reproducción
* **total time**: Tiempo total del archivo
* **progress bar**: Barra de progreso de la reproducción
* **song name**: Nombre de la canción
* **artist name**: Nombre del artista
* **buttons**
  + **queue**: Botón para mostrar/ocultar la cola
  + **volume**: Botón para subir/bajar/mutear el volumen
  + **back**: Botón para reproducir la canción anterior
  + **play/pause**: Botón para reproducir/pausar
  + **next**: Botón para pasar a la siguiente canción
  + **add**: Botón para agregar la canción a nuestra biblioteca