

Multimedia

Módulo 01

Implementación JS

Implementación

Javascript

Lo interesante de **audio y video** es que poseen métodos desde Javascript para **poder manipularlos**.

Para comenzar **a trabajar** debemos vincular un archivo js a nuestro archivo html, por ejemplo

```
<script src="js/codigo.js"></script>
```



Implementación

Javascript

Para poder comenzar luego de **vincular JS**, debemos identificar a nuestro elemento ya sea **video o audio en el DOM**.

Esto puede lograrse de diversas formas, **por ejemplo un ID**,

```
<audio muted id="audio" controls src="imagenes/musica-01.mp3"></audio>
```

Tanto para el **elemento audio como para video** las técnicas que **implementaremos son válidas**.



Implementación

Javascript

Luego desde nuestro **codigo.js** trabajaremos accediendo al **elemento**,

```
var musica = document.getElementById('musica');
```

El próximo paso será entender qué necesitamos hacer, por ejemplo si queremos simplemente accionar el audio o el video trabajamos con el **método play()**.

```
musica.play();
```



Implementación

Javascript

El [paso anterior](#) puede generarse a través de una función determinada como un click o directamente cuando carga la página si accionamos en el **onload() del body**.

Por otro lado si queremos accionar de forma inversa, es decir parando la reproducción trabajamos con **pause()**

```
musica.pause();
```

También tal acción puede **llamarse con funciones o directamente** en el **onload del body**.



Implementación

Javascript

Para poder saber la ruta del archivo que se está reproduciendo, podemos trabajar **con .src**

```
alert(document.getElementById('audio').src)
```

Esto también luego nos puede servir para poder combinarlo con elementos más **complejos y determinar qué si se dan ciertas circunstancias o no la ruta del audio o video cambiará.**

Esto claramente nos permitirá realizar listas de reproducción interactivas y dinámicas.



Implementación

Javascript

También como existe el método también podemos utilizar el estado **paused()** , similar al método llamado pause() para poder saber si un elemento está pausado o no.

En caso de **detectarlo podemos realizar un condicional como el siguiente,**

```
if (musica.paused == true) {  
    musica.play(); }  
}
```



Implementación

Javascript

También podemos mutear un video o audio, de esta forma no necesitamos los controles activos a través del atributo controls sino qué podemos personalizarlos a gusto,

```
document.getElementById('musica').muted = true
```

Tanto el valor que obtenemos en este estado como así en paused, es booleano, por lo tanto espera un true o un false.

De esta manera también puede ser [utilizado en un condicional](#).



Implementación

Javascript

Como vimos en la [diapositiva anterior](#), se puede generar una acción determinada por el true de un estado, **por ejemplo muted**

```
if(document.getElementById('musica').muted == true){  
    alert('Está muteado!')  
}
```



Revisión

- Repasar los conceptos vistos de **Javascript con audio y video**
- Trabajar con el **bonus track** debajo de todo para practicar los **elementos individualmente**.
- Ver todos los videos y materiales necesarios antes de continuar



¡Muchas gracias!

¡Sigamos trabajando!