

Programación

Ejercicios

Diciembre 2021

Cruz García, Iago



[Anotaciones previas](#)

[Ejercicios](#)

[Ejercicios - Objetos](#)

[Ejercicio 1](#)

[Ejercicio 2](#)

[Ejercicios - Array](#)

[Ejercicio 1](#)

[Ejercicio 2](#)

[Ejercicio 3](#)

[Ejercicio 4](#)

[Ejercicio 5](#)

Anotaciones previas

Estos ejercicios son para familiarizarse con el lenguaje, la sintaxis y cómo resolverlos. Los primeros son sencillos y se va incrementando la dificultad. A continuación se presentan una serie de instrucciones que son necesarias para la resolución de los ejercicios:

- [alert\(parámetro\)](#): esta instrucción permite mostrar por pantalla un cartel con texto para mostrar la solución de algunos ejercicios.
- [console.log\(parámetro\)](#): esta instrucción permite mostrar en consola (F12 en el navegador) la solución de algunos ejercicios o trazar el código para comprobar que todo se ejecuta correctamente.
- [prompt\(texto, ejemplo\)](#): Muestra en pantalla un recuadro de **texto** y un cuadro para introducir texto con un **ejemplo**.
- Para poder ejecutar código JavaScript en Visual Studio Code debéis crear un fichero JavaScript (miScript.js) y un HTML básico (index.html por ejemplo) y dentro de la etiqueta <head> escribir los siguiente:
 - <script src="miScript.js"></script> comillas incluidas
- Ahora que sabemos encapsular creando funciones o métodos, se pueden hacer los ejercicios en el mismo fichero, simplemente comentando las llamadas a métodos que no necesiteis.

```
ejercicio_1()  
//ejercicio_2()  
//ejercicio_3()
```

Ejercicios

IMPORTANTE: A partir de ahora algunos ejercicios deben hacerse en múltiples ficheros .js, por lo que en vez de entregar todos en un mismo main.js, será necesario dividirlos en directorios. Se aconseja la estructura de “PrácticaX_ejercicio1” y dentro el index.html y los ficheros .js necesarios.

Para estos ficheros, lo mejor es agrupar aquellas funciones o métodos que realicen tareas similares (entradas.js o salidas.js por ejemplo). En caso de duda, no importa que un método quede aislado en un fichero.

El fichero que realice las llamadas a los métodos, que aune toda la funcionalidad, debe nombrarse como main.js y no debe tener más que un método que se llame igual y una llamada a este mismo.

Ejercicios - Objetos

En estos ejercicios vamos a practicar los objetos y la herencia mediante prototipos.

Ejercicio 1

Crea un objeto que se llame Instrumento, 2 variables prototipo y una función prototipo. Lo que sea prototipo del objeto, será heredado:

```
var Instrumento = function() {  
  }  
Instrumento.prototype.timbre = "";  
Instrumento.prototype.nombre = "";  
Instrumento.prototype.emitirSonido = function() {  
  console.log("Que sonido más chulo emite " + this.nombre);  
}
```

A continuación, **crea 3 objetos que heredarán de este: Cuerda, Percusión y Viento**. Acuérdate que para crear la herencia haremos lo siguiente:

```
for (id in Instrumento.prototype){  
  Cuerda.prototype[id] = Instrumento.prototype[id];  
}
```

Comprueba que funciona la herencia instanciando un objeto de cada tipo:

```
var guitarra = new Cuerda();  
var trompeta = new Viento();  
var xilofono = new Percusion();
```

Y utilizando el método del “padre” (Instrumento) emitirSonido();

Prueba a cambiar el nombre de cada instancia accediendo a .nombre. Verás que son capaces de acceder aunque crees el objeto sin el atributo nombre.

Ejercicio 2

Con la estructura de objetos anterior, crea nuevos objetos que hereden de Cuerda (Guitarra, Bajo), de Viento (Flauta, Oboe) y Percusión (Bombo, Marimba). Para realizar la herencia de estos nuevos, tendrás que convertir los atributos y funciones de los anteriores en prototipos y realizar la misma estructura para la herencia. Añade métodos nuevos a Cuerda, Viento y Percusión y otros diferentes a cada objeto nuevo. Comprueba como Guitarra no debería tener acceso a los métodos de Viento, pero sí tendrá el método emitirSonido()

Ejercicios - Array

En esta sección trabajaremos los vectores o array.

Ejercicio 1

Crea un array de 5 valores numéricos de forma manual. Comprueba que cuando accedes a cada índice, este no coincide con el número en cuestión. Recordamos que el índice empieza en 0.

Ejercicio 2

Esta vez el array de 5 valores debe rellenarse mediante la entrada de datos (prompt). Verás que los arrays en JavaScript no tienen restricción de tipo, pudiendo almacenar valores numéricos, booleanos, caracteres, etc. sin problema. Esto supone que nosotros debemos controlar qué tipo de dato es el que se almacena.

Ejercicio 3

Crea un array de 5 valores con objetos creados en los ejercicios de objetos. Como todos los objetos vienen del mismo padre (Instrumento) verás que puedes utilizar la estructura `array['índice'].emitirSonido()` para verlos todos. Por eso es importante la herencia, para poder utilizar métodos o atributos sin tener que comprobar en cada ocasión que objeto es.

Ejercicio 4

Realiza un tablero de 3 en raya mediante arrays anidados. Para ello:

```
var tablero = [['-', '-', '-'], ['- ', '- ', '- '], ['- ', '- ', '- ']];
```

Para comprobar que el tablero existe:

```
var salida = "";
for (var i = 0; i < tablero.length; i++) {
    for (var j = 0; j < tablero[i].length; j++) {
        salida = salida + tablero[i][j] + " "
    }
    salida = salida + "\n";
}
console.log(salida);
```

Ejercicio 5

Con el tablero anterior, crea haz una función que permita introducir O o X en el. Puedes hacerlo de tal manera que el usuario tenga que escribir la fila y la columna donde quiere escribir primero, o asignar unas posiciones (como el pad numérico del teclado por ejemplo). Comprueba que se escriben donde quieres.