

PRÁCTICA 2.03 Objetos

Normas de entrega

- En cuanto al **código**:
 - en la **presentación interna**, importan los **comentarios**, la claridad del código, la significación de los nombres elegidos; todo esto debe permitir considerar al programa como **autodocumentado**. No será necesario explicar que es un **if** un **for** pero sí su funcionalidad. Hay que comentar las cosas destacadas y, sobre todo, las **funciones** y **clases** empleadas. La ausencia de comentarios será penalizada,
 - en la **presentación externa**, importan las leyendas aclaratorias, información en pantalla y avisos de ejecución que favorezcan el uso de la aplicación,
 - si no se especifica lo contrario, la información resultante del programa debe aparecer en la consola del navegador **console.log(información)**,
 - los ejercicios deben realizarse usando **JavaScript ES6** y usar el modo estricto (**use strict**) No se podrá utilizar *jQuery* ni cualquier otra biblioteca (si no se especifica lo contrario en el enunciado),
 - para el nombre de **variables**, **constantes** y **funciones** se utilizará *lowerCamelCase*,
- En cuanto a la **entrega** de los archivos que componen los ejercicios:
 - todos los ejercicios en **una carpeta** (creando las **subcarpetas** necesarias para documentación anexa como imágenes o estilos) cuyo nombre queda a discreción del discente,
 - el nombre de los ficheros necesarios para resolver el ejercicio será el número de ejercicio que contenga,
 - el código contendrá ejemplos de ejecución, si procede, y
 - la carpeta será comprimida en formato **ZIP** y será subida a **Aules** de forma puntual.

Ejercicio 1 - Constructor (de objetos)

Crea una función que simule ser un constructor de objetos (recuerda que es una función que devuelve un objeto) que genere objetos del tipo `curso`.

Las propiedades de este objeto serán: nombre del curso, año, descripción y alumnado que es un array. Estas propiedades tendrán valores vacíos de momento.

Estos parámetros se le pasarán a la función para que cree el objeto, cuando llamemos a la función. Verifica que permite crear un objeto empleando el constructor y que puedes acceder a las diferentes propiedades del objeto.

Ejercicio 2 - Mostrando (objetos)

Sobre el ejercicio anterior, escribe una función para hacer un informe completo (que muestre toda la información que contiene) del objeto **curso** y que lo muestre por consola, ya sabes como: debidamente formateado.

Nota: para recorrer el objeto utilizar la estructura `for (propiedad in objeto) {};`

Ejemplo resultado:

TIPO	NOMBRE	VALOR
string	nombre	Matematicas
number	año	2024
string	descripción	Curso de mates
object	alumnado	Juan,Pepe,Damian

Ejercicio 3 - Recorriendo (objetos)

Crea un objeto denominado **discente** que tenga las siguientes propiedades: id, nombre, apellidos, aficiones (que será un *array* de *string*) y notas (que será un objeto con las propiedades primera, segunda y tercera con las notas de cada evaluación).

Añade las siguientes funciones al objeto (simulando ser métodos de una clase):

- **calcularMedia**, que calculará la nota media de las tres evaluaciones,
- **imprimirAficiones**, que imprimirá por consola las aficiones del alumnado e
- **imprimirInforme**, que imprime por consola un informe completo.

Las impresiones deben hacerse debidamente formateadas.

Ejemplo resultado:

Informe completo del discente:

```
ID: 2
Nombre: Ana López Martín
Aficiones: pintar, bailar, tocar el piano
Notas: Primera Evaluación: 9, Segunda Evaluación: 8.5, Tercera Evaluación: 9.8
Nota Media: 9.10
```

Ejercicio 4 - Modificando (objetos)

Modifica la función `curso` creada en el ejercicio 1, para que sólo permita enviar los parámetros del curso usando la firma siguiente:

```
function curso(nombreCurso, añoCurso, descripcionCurso) {
```

Dentro, añade, al objeto `curso`, un método denominado **matricular** que recibe como parámetro un objeto **discente** (creado en el ejercicio anterior) y que los añade a la clave `alumnado` del objeto `curso`. Utiliza para ello el **spread operator** para esta tarea.

Ejercicio 5 - Mostrando II (objetos)

Diseña una función que permita imprimir por consola cualquier objeto pasado como parámetro. Se deberá comprobar el tipo de dato de cada clave del objeto y actuar en consecuencia dependiendo de:

- si es un número o una cadena, se imprime tal cual
- si es un objeto, deberá diferenciar si es un *array* (que imprimirá sus valores), si es una función (que no imprimirá nada) o si es un objeto (que lo imprimirá debidamente formateado).

La salida debe estar, como es tradicional, debidamente formateada e indicando el tipo de dato que contiene cada clave del objeto. Por ejemplo:

nombreDeLaClave → **nombreDelValor** → **tipoDeDatoDelValor**