

PRÁCTICA 7.01 Módulos con clase(s)

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,
 - todo el código debe estar situado dentro del evento `window.onload = () => {};` o a través del evento `document.addEventListener("DOMContentLoaded", () => {});`,
 - 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*,
 - para la asignación de eventos se utilizará `addEventListener()` indicando sus tres parámetros en su definición,
 - debes dividir tu código en **bibliotecas temáticas** de funciones y/o clases a partir de este ejercicio,
 - se usarán las funcionalidades **import** y **export** para crear bibliotecas de funciones temáticas a partir de esta práctica,
 - todo el código que sea susceptible de retrasar su ejecución debe escribirse de forma asíncrona.
- 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 - Prototipos

Añade el método `repetir()` dentro del prototipo del objeto `String` para que acepte un entero con el número de ocasiones que tiene que repetir la cadena. Por ejemplo:

```
console.log(";Viva JavaScript!".repetir(3));
```

Este código devuelve tres veces la cadena de texto que contiene. En el caso de recibir un tipo inesperado o negativo, deberá lanzar un ~~insulte~~ error informativo. **No es necesario** utilizar la importación de módulos en este ejercicio.

Ejercicio 2 - Curso

Diseña una clase **Curso** que (todo el contenido debe ser mostrado en el **DOM**):

- recibirá en su constructor un nombre, el número del aula asignado, el número de módulos y el número del alumnado,
- el alumnado se representará como un *array* de objetos de la clase **Alumnado**. En esta clase se guardarán los atributos DNI, nombre, apellidos, fecha de nacimiento y nota media (un valor que guarde la nota con la que entró al módulo) y **módulos**, que será representado como un *array* de objetos de tipo módulo,
- los módulos (asignaturas) se almacenarán un objeto de objetos de la clase **Módulos** (que contendrá el nombre del módulo, el número de horas y el profesorado que lo imparte, que se guardará como un *array* de objetos de la clase **Profesorado** (diseñala con los datos básicos)

Recuerda comprobar siempre los prototipos de los objetos antes de insertarlos en la clase y si el objeto a introducir ya está en el *array*. En las clases que lo requieran, no olvides crear los *setters* y *getters*. Con todo esto recordado, implementa los siguientes métodos:

- en **Alumnado** para modificar su nota media, otros para devolverla (*setter* y *getter*) y un último método para indicar si es mayor de edad o no,
- en **Curso**, un método para obtener la nota media (como media de las notas medias de todo el alumnado a través del método de la clase **Alumnado**),
- en **Curso**, un método que muestre (debidamente formateado) el profesorado de los módulos,
- en **Curso**, un método que matricule al alumnado en el curso (debe comprobar si el objeto posee el prototipo de alumnado),
- en **Curso**, listado del alumnado en orden ascendente o descendente (especificado por un parámetro),
- en **Curso**, otro para hacer un informe completo del curso que constará de una tabla con el alumnado matriculado en los módulos, el profesorado asignado a cada módulo y los datos del curso (debidamente formateado).

Crea un fichero para cada clase y recuerda utilizar la **sintaxis de módulos** (`import` y `export`) y **verifica que funcione correctamente** con un par de ejemplos.