

TEMA 4. Programación con funciones, arrays y objetos definidos por el usuario.

OBJETOS

1. Crear objeto Vehículo.
 - a. Crear constructor.
 - b. Crear propiedades: *marca*, *modelo*, *color* y *año fabricación*.
 - c. Instanciar dos objetos.
 - d. Mostrar por pantalla una propiedad con notación por puntos. Objeto 1 - marca.
 - e. Mostrar por pantalla una propiedad con notación por corchetes. Objeto 2 - color.
 - f. Crear el método *mostrarDatos*.
 - g. Crear un método que tenga argumentos *acelerar(velocidad)*. Mostrará por pantalla que el coche ha acelerado a la velocidad pasada.
 - h. Crear un método con función interna *arrancar*. Mostrará por pantalla que el coche de marca *_*, modelo *_*, de color *_* ha arrancado.
 - i. Asignar una nueva propiedad *cilindrada*.
 - j. Asignar un nuevo método *frenar*. Mostrará por pantalla que el coche de marca *_*, modelo *_*, color *_* ha parado).
 - k. Realizar las siguientes modificaciones en el Objeto 1 : la propiedad *marca* no podrá ser eliminadas, la propiedad *modelo* tendrá como valor por defecto 'xx' y no podrá ser modificado y los *métodos* no serán enumerables.
 - l. Borrar la propiedad *marca*, modificar la propiedad *modelo* y volver a mostrar los datos de Objeto 1.
 - m. Muestra todas las propiedades de Objeto1 utilizando: `for in` , `Object.hasOwnPropertyNames()`, `Object.keys()`.

1.b. Utilice el código empleado para la generación de las instancias del objeto Vehículo, y modifíquelo para que los valores de cada una de sus propiedades se impriman en una tabla HTML. Utilice la generación de código HTML desde código JavaScript. Cada instancia del objeto debe ocupar una línea y el valor de cada propiedad debe ocupar una celda de dicha línea. El resultado debe ser similar a:

| MARCA | MODELO | COLOR | AÑO FABRICACIÓN |
|---------|------------|--------|-----------------|
| Ferrari | Scaglietti | Rojo | 2010 |
| BMW | Z4 | Blanco | 2010 |
| Seat | Toledo | Azul | 1999 |
| Fiat | 500 | Verde | 1995 |

2. Cree un script que defina un objeto llamado *Producto_alimenticio*. Este objeto debe presentar las propiedades código, nombre y precio, además del método *imprimeDatos*, el cual escribe por pantalla los valores de sus propiedades. A continuación, cree tres instancias de este objeto y guárdelas en un array. Posteriormente, utilice el método *imprimeDatos* para mostrar por pantalla los valores de los tres objetos instanciados.
3. Añadir al objeto Array un método que permita eliminar de un array un determinado valor.

4. Crear una pseudoclase llamada **Factura** que permita instanciar objetos de ese tipo de la siguiente forma:
- ```
Factura(cliente, elementos, info)
```
- Cliente** es un objeto que guarda los datos del cliente (nombre, dirección, teléfono y nif) y **elementos** es un array que contiene la siguiente información por cada uno de los ítems que puede tener la factura: descripción, cantidad y precio.
- Info**, sobre cada factura se desea guardar los siguientes datos: base imponible, iva, total y forma de pago. Por defecto, tendrán como valores 0, 21,0,"contado" respectivamente.

Añade con posterioridad a la pseudoclase Factura:

- Propiedad empresa que guardará información sobre la empresa que emite la factura (nombre, dirección, teléfono y cif).
- Método que calcule el total de la factura (con el IVA aplicado);
- Método que muestre el total.

Realiza una instanciación de Factura y muestra su total.

Modifica el ejercicio para que todas las propiedades sean privadas y el método calculaTotal. El método muestraTotal será privilegiado.

5. Crea un objeto Furgoneta y un objeto Todoterreno, ambas tienen que heredar del objeto Vehiculo. El objeto Furgoneta tiene que añadir una propiedad para indicar el número de pasajeros y el objeto Todoterreno debe de añadir una propiedad que indique si tiene tracción o no. Crea una instancia de cada uno de ellos y muestra los datos.
6. Definir la siguiente jerarquía de objetos, haciendo uso de los prototipos de JavaScript:
- Objeto Persona con las propiedades nombre, edad y género, y el método obtDetalles(), que muestra por pantalla las propiedades de la persona.
  - Objeto Estudiante, que hereda de Persona, e incluye las propiedades curso y grupo y el método registrar().
  - Objeto Profesor, que hereda de Persona, e incluye las propiedades asignatura y nivel y el método asignar().

Crear los objetos y casos de prueba necesarios para comprobar el correcto funcionamiento de la jerarquía.

7. Queremos implementar la siguiente estructura:
- Un Empleado se define con las propiedades nombre (cuyo valor por defecto es una cadena vacía), y un departamento (cuyo valor por defecto es "General").
  - Un Director está basado en Empleado. Añade la propiedad informes (cuyo valor por defecto es un array vacío).
  - Un Trabajador está basado también en Empleado. Añade la propiedad proyectos (cuyo valor por defecto es un array vacío).
  - Un Ingeniero está basado en Trabajador. Añade la propiedad máquina (cuyo valor por defecto es una cadena vacía) y sobrescribe la propiedad departamento con el valor "Ingeniería".

Crear los objetos y casos de prueba necesarios para comprobar el correcto funcionamiento de la jerarquía.