

Unidad 5.

JavaScript Ejercicios.

- Objetos definidos por el Desarrollador.

Ejercicio 1.

Escribir el código de una web, tal que:

1. Queremos crear un objeto coche con tres atributos: tipo, modelo, color. En este caso el objeto tiene unos valores concretos, ya que no hemos utilizado un constructor de una clase para cargarlo de valores. Así que, el tipo es "Fiat", el modelo es "500" y el color es "white". ¿Cómo mostrarías en un párrafo el valor de sus distintos atributos?
2. Queremos crear un objeto persona con 5 atributos: Nombre, apellido, edad, colorOjos. En este caso al igual que antes vamos a inicializar el objeto con unos valores concretos: Juan, Lopez, 50, verdes. ¿Cómo mostrarías en un párrafo los valores de los distintos atributos?

Nota: Ten en cuenta que para mostrar el valor de los atributos puedes utilizar dos notaciones diferentes: `persona.apellido` ó `persona["apellido"]`

Comprueba que puedes hacerlo de ambas formas.

Ejercicio 2.

Partiendo del ejercicio anterior. Queremos que el objeto persona, tenga un método "muestraNombre" que devuelva: El nombre y el apellido de la persona. Como llamarías al método y lo mostrarías en un tercer párrafo. Con un `document.write`, y en un `alert`.

Nota: Prueba poniendo y quitando paréntesis en las llamadas al método.

Ejercicio 3.

Queremos crear un script que defina un objeto llamado Producto.

Tiene las propiedades código, nombre y precio, además del método `imprimeDatos`, el cual escribe por pantalla los valores de sus propiedades.

Comprueba realmente si el método `imprimeDatos` es una función.

Posteriormente, crea tres instancias de este objeto y guárdalas en un array.

Posteriormente, utiliza el método `imprimeDatos` para mostrar por pantalla los valores de los tres objetos instanciados. Nota: Comprueba para qué sirve

La ejecución, mostrará:

código → 1
nombre → Moscatel
precio → 5

código → 2
nombre → Tarta
precio → 12

código → 3
nombre → Queso
precio → 6

```
for (let i of mi_array) {  
    document.write(i.imprimeDatos() + '<br />');  
}
```

Ejercicio 4.

Queremos crear un constructor de una Clase llamada "Persona" con los atributos: nombre, edad, genero.

Y crearemos un método llamado "obtDetalles" que nos mostrará en pantalla mediante un document.write los valores de los atributos. Dicho método no estará incluido en la definición del constructor, sino que será una expansión del constructor usando prototype.

Crea una variable persona1 y asígnale los valores: Juan, 38, Masculino.

Utiliza el método creado para mostrar el contenido del objeto creado.

Ejercicio 5.

Una Pseudoclase es aquella, cuyo constructor se forma con atributos de otra clase y atributos propios. En la creación del constructor por tanto tendrán que aparecer también los atributos que va a heredar.

Y gracias al operador call podemos decir de qué clase provienen y poderlos usar en esta nueva clase.

Y por último tendremos que extender, esta nueva clase utilizando la palabra reservada constructor para decirle que use también los datos de la primera clase.

Aquí tienes como realizarlo:

```
// PseudoClase Estudiante, será hija de Persona
function Estudiante(nombre, edad, genero, curso, grupo) {
    this.curso = curso;
    this.grupo = grupo;
    Persona.call(this, nombre, edad, genero);
}

// Herencia desde Persona en el prototype
Estudiante.prototype = new Persona();

// Asigno constructor de Estudiante en el prototype
Estudiante.prototype.constructor = Persona;
```

Teniendo en cuenta esto, crea un nuevo Estudiante, con los siguientes datos:

nombre: Carmelito, edad: 20, genero: masculino, curso: Desarrollo Web, grupo: 2º

Para mostrar los datos, crea un método dentro de la clase estudiante. Que mostrará todos los datos de ese estudiante.

Ejercicio 6.

Siguiendo con el ejercicio anterior, vamos a extender el prototipo de la clase hija (la pseudoclase) con un método llamado "registrar" que usará un document.write para mostrar "El estudiante (Carmelito) ha registrado algo".

Ejercicio 7.

Al igual que has hecho en el ejercicio anterior, ahora crea una Pseudoclase "Profesor" de tal forma que en sus argumentos, recibirá: nombre, edad, genero, asignatura, nivel.

Crea un profesor1 con los siguientes datos: juan, 38, masculino,

Y extenderemos la clase Profesor con un método "asignar" que mostrará mediante un document.write "El profesor (Juan) ha asignado algo".

Acceder también al método obtDetalles del profesor creado.

Ejercicio 8.

Ahora sobre el ejercicio anterior, vamos a crear en el constructor de la clase "Profesor" un atributo que siempre va a tener el mismo valor al instanciar el objeto.

El nuevo atributo para profesor es: Cargo y el valor para inicializarlo es "Profesor"

Realiza también una extensión del prototipo de Profesor y añade el método mostrarCargo, que mostrará el cargo de dicho profesor.

Ejercicio 9.

Teniendo en cuenta la creación de objetos y herencia:

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.

Ejercicio 10.

Crear una pseudoclase llamada Factura que permita instanciar objetos de ese tipo de la siguiente forma:

Factura(cliente,elementos) 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.

Además 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.

Crea una instancia de Cliente con el nombre cliente1.

Crea una instancia del Empresa con el nombre empresa1.

Crea una instancia de Factura con el nombre factura1.

Muestra todos los datos de cliente1, empresa1 y factura1.

Añade con posterioridad a la pseudoclase Factura:

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

Ejercicio 11.

Teniendo en cuenta que los objetos nativos de JavaScript también tienen el prototype.

Añade al objeto Array un método que permita eliminar del array un determinado valor.

El Array tendrá unos valores predeterminados: 1,2,3,4,5,6,4,8,2

1,2,3,4,5,6,4,8,2

Quitar elemento de Array mediante Prototype

1,2,4,5,6,4,8,2

La solución pintarla en un div, utilizando innerHTML.

Ejercicio 12.

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.

Instrucciones para Crear la PseudoClase Vehiculo:

-Crear propiedades: marca, modelo, color y año fabricación

-Crear el método mostrarDatos

En este caso vamos a utilizar la notación \$(marca), esto nos permitira mostrar la marca, por ejemplo, o \$(this.marca).

-Crear un método que tenga argumentos acelerar(velocidad). Mostrará por pantalla que el coche ha

acelerado a la velocidad pasada.

-Crear un método con función interna arrancar. Mostrará por pantalla que el coche de marca __, modelo__, de color__ ha arrancado.

-Asignar una nueva propiedad cilindrada

-Asignar un nuevo método frenar que mostrará por pantalla que el coche de marca__, modelo__, color__ ha parado).

Nota: Utiliza la notación \$(marca) \$(this.marca)