

## Relación de actividades para la unidad 4 (PARTE III)

15. Escribe un programa JavaScript que obtenga una lista de nombres tecleados por el usuario, y los almacene en un array hasta que el usuario entre la cadena vacía. Visualiza en la página la siguiente información:

- El primer nombre tecleado por el usuario.
- El último nombre tecleado por el usuario.
- La lista de nombres en orden ascendente, introducidos por el usuario, cada uno de ellos en una línea diferente.

16. Implementa un JavaScript que cree un array de objetos y los muestre en una tabla. Los objetos tendrán un nombre, un apellido y unos estudios. Los estudios pueden ser: Web con código 0, ó Sistemas con código 1. No es obligatorio determinar los estudios. Cuando esté completada la información y se pulse sobre el botón "Carga", se crea un objeto y se añade al array. Se mostrarán los objetos almacenados en el array cuando se pulsa sobre el botón "Muestra". El array se inicializará cuando se pulse el botón "Inicializa".

Nombre:

Apellidos:

Estudios:

Nombre	Apellidos	Estudios
Marcos	Pérez	Sin determinar
Laura	García	0-Web
Pepe	Castaño	1-Sistemas

17. Realiza un programa que utilice una función llamada *calcularIVA* a la que se le pasarán dos parámetros: precio e IVA. La función deberá calcular el valor del IVA sobre el precio y devolver el resultado.

Precio:

IVA %:

21

18. Implementa un JavaScript que use una función llamada *ordenarArrayObjetos* que reciba un array de objetos y una cadena que represente el nombre de una propiedad de los objetos. La función deberá ordenar el array de objetos de acuerdo con los valores de la propiedad especificada en un *prompt*, en orden ascendente. Además, se mostrará por consola el resultado del ordenado.

Por ejemplo, si tenemos el siguiente array de objetos:

```
const personas = [  
  { nombre: 'Marcos', edad: 20 },  
  { nombre: 'Laura', edad: 18 },  
  { nombre: 'Ignacio', edad: 35 },  
  { nombre: 'César', edad: 19 }  
];
```

Después de llamar a la función `ordenarArrayObjetos` pasando como primer argumento el array de personas y como segundo argumento la cadena solicitada al usuario, supongamos edad, debería mostrar el siguiente array ordenado.

1. **0:** {nombre: 'Laura', edad: 18}
2. **1:** {nombre: 'César', edad: 19}
3. **2:** {nombre: 'Marcos', edad: 20}
4. **3:** {nombre: 'Ignacio', edad: 35}
5. **length:** 4
6. **[[Prototype]]:** Array(0)

**19.** Programa un script que cree una clase llamada Coche que se **defina en un fichero independiente**, con los siguientes componentes y requisitos de funcionalidad:

- a. Tendrá dos propiedades: “marca” y “modelo”.
- b. Contará con dos métodos: “arrancar” y “parar”.
- c. El método “arrancar” visualizará por consola el mensaje “El coche de marca [marca] y modelo [modelo] ha arrancado”.
- d. Si ya está arrancado, visualizará por consola el mensaje “El coche de marca [marca] y modelo [modelo] ya está arrancado”.
- e. El método “parar” visualizará por consola el mensaje “El coche de marca [marca] y modelo [modelo] ha parado”.
- f. Si ya está parado, visualizará por consola el mensaje “El coche de marca [marca] y modelo [modelo] ya está parado”.

Crear dos objetos a partir de la clase Coche, con diferentes valores para las propiedades “marca” y “modelo”. Llamar a los métodos `arrancar()` `parar()` para cada uno de los objetos y comprobar que se visualizan los mensajes adecuados en la consola.

**20.** Realiza un programa en JavaScript que cree una clase llamada Animal con las propiedades nombre y edad; y un método llamado `comer()` que visualice por consola “El animal está comiendo”. A continuación, crea una clase hija llamada Gato que herede de la clase Animal y añada una propiedad adicional raza y un método llamado `maullar()` que visualice por consola “miau”.

Crear una instancia de la clase Gato, establecer un nombre, edad y raza; y llamar a los métodos `comer()` y `maullar()` para comprobar que funcionan correctamente y visualiza también la consulta de sus propiedades.

**Indicaciones de entrega para las actividades de programación:**

Nombrar cada código fuente de la siguiente manera:

actividad\_*NUM*\_ut4.html o actividad\_*NUM*\_ut4.js dependiendo de la necesidad.

*NUM*: será sustituido por el número de la actividad correspondiente.

Estas actividades se subirán a [moodle.iescierva.net](http://moodle.iescierva.net), junto al resto de actividades de la unidad 4, en un único fichero comprimido, en la fecha que se indique en clase.