

Actividades - Responde y/o documenta con capturas de pantallas donde se pueda verificar tu autoría:

1. En un script vamos definir la siguiente lista de nombres: (0,5 puntos)

'Juan', 'Pedro', 'Luis', 'María', 'Julia'

- Primero definiremos una función llamada `mostrar()`, que mostrará por consola los elementos que contiene nuestra lista.
- Después se le solicitará al usuario que introduzca un nombre por teclado.
- A continuación se definirá la función `buscarItem()`, a la que se le pasará la lista y el nombre introducido por el usuario. Esta función determinará si el nombre en concreto está en la lista, y si es así en qué posición se encuentra. Si el nombre no está en la lista la función debe devolver un valor negativo.
- Por último, se informará por pantalla.

2. Crea un script que contenga un array con los valores siguientes: (0,5 puntos)

Dragonborn, Skyrim, Dovahkiin, Dragon, Fire, Game

- Muestra por pantalla: El número de elementos que tiene el array y el contenido del array.
- Utiliza el método **`push()`** para añadir los elementos siguientes:

Videojuego, Dragones

- Muestra por pantalla: El número de elementos que tiene el array y el contenido del array.
- Solicita al usuario que introduzca una cadena de caracteres y añádela al array.
- Muestra por pantalla: El número de elementos que tiene el array y el contenido del array.

3. Crea un script que contenga un array con los valores siguientes: (0,5 puntos)

Dragonborn, Skyrim, Dovahkiin, Dragon, Fire, Game

- Muestra por pantalla: El número de elementos que tiene el array y el contenido del array.
- Utiliza el método **unshift()** para añadir los elementos siguientes:
- *Videojuego, Dragones*
- Muestra por pantalla: El número de elementos que tiene el array y el contenido del array.
- Solicita al usuario que introduzca una cadena de caracteres y añádela al array (con **unshift()**).
- Muestra por pantalla: El número de elementos que tiene el array y el contenido del array.

4. Crea un script que contenga un array con los valores siguientes: (0,5 puntos)

Dragonborn, Skyrim, Dovahkiin, Dragon, Fire, Game

- Muestra por pantalla: El número de elementos que tiene el array y el contenido del array.
- Utiliza el método **pop()** para eliminar el último elemento del array.
- Muestra por pantalla: El número de elementos que tiene el array y el contenido del array.
- Muestra por pantalla: El elemento eliminado del array.

5. Crea un script que contenga un array con los valores siguientes: (0,5 puntos)

Dragonborn, Skyrim, Dovahkiin, Dragon, Fire, Game

- Muestra por pantalla: El número de elementos que tiene el array y el contenido del array.
- Utiliza el método **shift()** para eliminar el primer elemento del array.
- Muestra por pantalla: El número de elementos que tiene el array y el contenido del array.
- Muestra por pantalla: El elemento eliminado del array.

6. Crea un script que contenga un array con los valores siguientes: (0,5 puntos)

Dragonborn, Skyrim, Dovahkiin, Dragon, Fire, Game

Muestra por pantalla:

- El número de elementos que tiene el array.
- Cada uno de los elementos del array junto a su índice, debe quedar algo parecido a lo siguiente:

0: Dragonborn

1: Skyrim

2: Dovahkiin

3: Dragon

4: Fire

5: Game

- Vamos a crear un nuevo array a partir del anterior, para ello:
 - Solicita al usuario que elija el índice del elemento del array anterior que será el primer elemento en el nuevo array.
 - Solicita al usuario que elija el índice del elemento del array anterior que será el último elemento en el nuevo array.

7. Crea un script que contenga un array con los valores siguientes: (0,5 puntos)

Dragonborn, Skyrim, Dovahkiin, Dragon, Fire, Game

- Muestra por pantalla: El número de elementos que tiene el array y el contenido del array.
- Solicita al usuario que introduzca un carácter que actúe como separador.
- Muestra por pantalla una cadena de caracteres compuesta por los elementos del array, pero separados por el carácter que ha introducido el usuario.
- Para comprobar los cambios en el array, muestra por pantalla el número de elementos que contiene, y el contenido del array.

8. Realiza un programa que reciba una cadena con el siguiente formato: (0,5 punto)

“nombre:apellidos:telefono:email:codigopostal”

- Tras recibir la cadena, debe desglosar y mostrar la siguiente información:
 - Código postal.
 - Apellidos.
 - Email.
 - Suponiendo un formato de email “direccion@servidor” debe mostrar el nombre del servidor asociado.

9. Crea una función llamada “pideYComprueba” que mediante un bucle y utilizando prompt, solicite al usuario que introduzca tantos números enteros como quiera. (2 puntos)
- El usuario podrá estar introduciendo números hasta que inserte el carácter “x”.
 - Cada vez que el usuario introduzca un número deberás comprobar que es un número entero válido (parsearlo) y guardarlo en un array.

- La función debe devolver un array con los números introducidos por el usuario ya parseados.
- Crea una función llamada “mitad” a la que se le pase un array como parámetro y que divida todos los valores de ese array entre 2. NO tiene que crear un array nuevo. Debe ser el array original el que debe modificar.
- Modifica la función anterior, reutilizando las funciones que ya tengas y crea una función llamada “porlaMitad” a la que se le pase un array como parámetro y que divida todos los valores de ese array entre 2. *La función debe devolver array nuevo. NO debe modificar el array original.*

10. Define un script en el que se crea un array para almacenar los cinco sueldos de operarios y luego mostrar el total de gastos en sueldos (cada actividad en una función). A la hora de implementar el script debes seguir los siguientes requisitos (2 puntos):

- Definiremos un objeto de la clase Array, indicándole que queremos almacenar 5 valores. Lo llamaremos sueldos.
- Definiremos una función “cargar()” que se encargue de cargar el array sueldos que hemos creado con los sueldos de los operarios.
 - Para ello, llamaremos a la función cargar() enviándole el array sueldos.
 - En la función, a través de un bucle for recorreremos las distintas componentes del array y almacenamos valores enteros que ingresamos por teclado.
 - Para conocer el tamaño del array accedemos a la propiedad length de la clase Array.
- Definiremos también la función “calcularGastos(sueldos)”, en la que sumamos todas las componentes del array sueldos, imprimimos en la página los valores y el total de gastos.

11. Realizar la función de comparación que ordene el array de **personas.js** por los siguientes criterios: (2 puntos)

1. En primer lugar se ordenará por el cargo (primero Profesor, después Administrativo y por último Estudiante).
2. Después por edad, descendente.
3. Y en último lugar, por nombre, ascendente.

De forma que el array anterior debería quedar ordenado de la siguiente forma:

```
[["Luisa",60,"Profesor"], ["Agustín",50,"Profesor"], ["Adrián",47,"Profesor"],  
["Jose",47,"Profesor"], ["Toñi",48,"Administrativo"],  
["Juan",36,"Administrativo"], ["Rosa",75,"Estudiante"],  
["Fermín",64,"Estudiante"], ["Blas",36,"Estudiante"], ["Zoraida",36,"Estudiante"],  
["Pepe",31,"Estudiante"], ["Felipe",25,"Estudiante"], ["Ana",20,"Estudiante"],  
["Pedro",19,"Estudiante"], ["Juan",16,"Estudiante"], ["Miriam",15,"Estudiante"] ]
```

Muestra el resultado:

- por consola en formato Array
- formando la siguiente cadena: *Luisa de 60 años es Profesor*