

EJERCICIOS

Ejercicio 1

1. Dado un array con los días de la semana obtén:
 - Todos los días que empiezan por 'M'.
 - El primer día que empieza por 'M'.
 - La posición en el array del primer día que empieza por 'M'.
 - Indica si algún día empieza por 'S'.
 - Indica si todos los días acaban por 's'.
 - Devuelve otro array con los días en mayúsculas.
 - Indica si algún día es el 'martes'.

DESARROLLO - Ej1

```
<script>
/*Variables para almacenar datos del array*/
let dias_m = []; // Guarda los días que empiecen con m
let primer_dia_m = ""; // Guarda el primer día con m del array
let pos_primer_m = 0; // Posición del primer día del array con m
let hay_dia_s = 0; // Indica si hay algún día con s, inicializado en verdadero
let dias_s = 1; // Indica si todos los días acaban con s
let dias_mayus = []; // Guarda los días en mayúsculas
let es_martes = 0; // Indica si algún día es martes

//Array con los días de la semana
let dias_semanas = ["lunes", "martes", "miercoles", "jueves", "viernes", "sabado", "domingo"];

/*Recorre los días*/
for (let i = 0; i < dias_semanas.length; i++) {
    // Compara los días que empiecen con "m"
    if(dias_semanas[i].startsWith("m") ) {
        /*Guarda los días en el array*/
        dias_m.push(dias_semanas[i]);

        //Solo permite guardar una sola vez cuando se encuentra un día con m, siendo el primero y unico
        if(primer_dia_m === "") {
            primer_dia_m = dias_semanas[i]; // Guarda el primer día con m
            pos_primer_m = i; //Guarda la posición
        }
    }
}
```

```
// Comprueba si hay 's' en algun día de la semana, si es así, devuelve true
if(dias_semanas[i].includes("s")) {
    hay_dia_s = true;
}

// Como esta inicializado como verdadero, al encontrar un día que no termine en s, cambia a falso
if(!dias_semanas[i].endsWith("s")) {
    dias_s = false;
}

/*Guardar todos los días en mayuscula*/
dias_mayus.push(dias_semanas[i].toUpperCase());

//Comprueba si algun día es martes
if(dias_semanas[i] == "martes") {
    es_martes = true;
}
}
```

```
// Imprime los días que empiecen con m
console.log("Días que empiezan con 'm': " + dias_m.join(" "));
console.log("Primer día de la semana con 'm': " + primer_dia_m);
console.log("Posición del primer día con m: " + pos_primer_m);
console.log("¿La semana tiene días con 's'? " + ((hay_dia_s) ? "Verdadero":"Falso"));
console.log("¿Todos los días acaba con 's'? " + ((dias_s) ? "Verdadero":"Falso"));
console.log("Los días en mayusculas: " + dias_mayus.join(" "))
console.log("¿Hay martes en semana? " + ((es_martes) ? "Si":"No"));
```

```
Días que empiezan con 'm': martes miercoles
Primer día de la semana con 'm': martes
Posición del primer día con m: 1
¿La semana tiene días con 's'? Verdadero
¿Todos los días acaba con 's'? Falso
Los días en mayusculas: LUNES MARTES MIERCOLES JUEVES VIERNES SABADO DOMINGO
¿Hay martes en semana? Si
```

-
2. Dado el siguiente array de ciudades y población, realiza un programa que permita al usuario visualizar las ciudades con mayor población que el número introducido:

```
const ciudades = [{nombre: 'Alicante', poblacion: 3792621},
                  {nombre: 'Elche', poblacion: 8175133},
                  {nombre: 'Torrevieja', poblacion: 2695598},
                  {nombre: 'Orihuela', poblacion: 2099451},
                  {nombre: 'Callosa', poblacion: 1526006}
];
```

DESARROLLO - EJ2

```
<!-- Lucero anay Cahuana Cornejo -->
<!-- Objetivo: mostrar las ciudades mas pobladas segun el marcador introducido por el usuario -->
<script>
  // Array de objetos brindado para el desarrollo del ejercicio
  const ciudades = [{ nombre: 'Alicante', poblacion: 3792621 },
    { nombre: 'Elche', poblacion: 8175133 },
    { nombre: 'Torrevieja', poblacion: 2695598 },
    { nombre: 'Orihuela', poblacion: 2099451 },
    { nombre: 'Callosa', poblacion: 1526006 }
  ];

  // Se solicita al usuario un numero de habitantes para filtrar las ciudades
  let valor_min = parseInt(prompt("Introduce un número de habitantes para filtrar las ciudades: "))

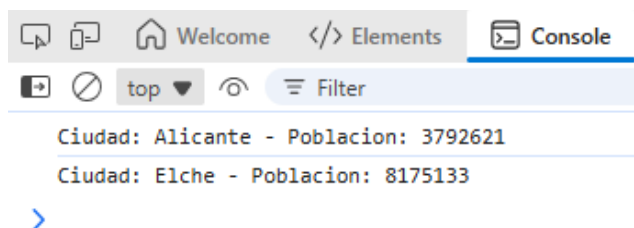
  // Recorre las ciudades
  for (let i in ciudades) {
    // Ahora se filtra las ciudades que son mayores al valor mínimo introducido por el usuario
    if(ciudades[i].poblacion > valor_min) {
      //Imprime los datos de las ciudades y poblacion mayores al valor min
      console.log("Ciudad: "+ciudades[i].nombre + " - Poblacion: "+ciudades[i].poblacion )
    }
  }
</script>
```

Esta página dice

Introduce un número de habitantes para filtrar las ciudades:

Aceptar

Cancelar



-
3. Dado el siguiente array de nombres y edades, realiza un programa que permita al usuario visualizar la primera persona que tenga la edad introducida:

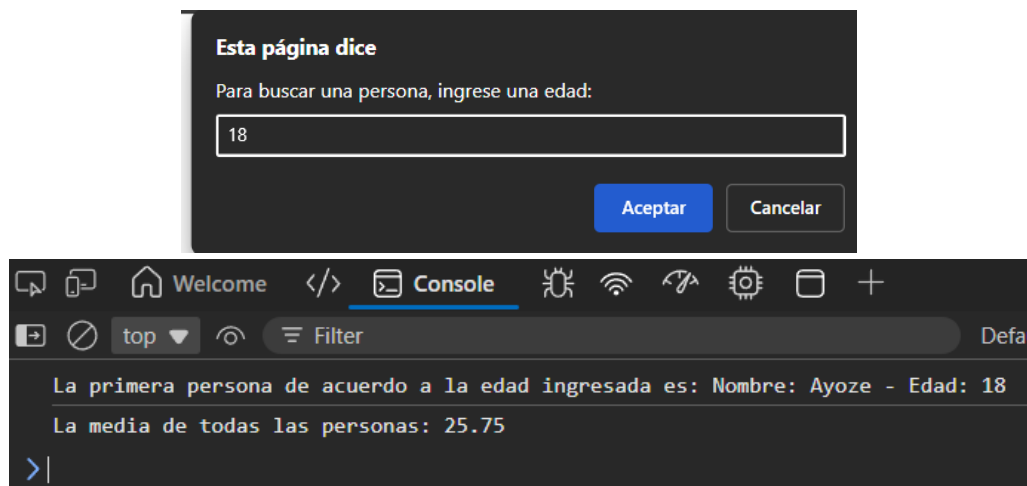
```
const personas = [ {name: "María", age: 20},
  {name: "Bernardo", age: 28},
  {name: "Pancracio", age: 22},
```

```
    {name: "Andrea", age: 19},  
    {name: "Sara", age: 29},  
    {name: "Jorge", age: 32},  
    {name: "Yurena", age: 38},  
    {name: "Ayoze", age: 18}  
  ];  
};
```

Muestra por consola la edad media de las personas.

DESARROLLO - EJ3

```
<body>  
  <!-- Objetivo: mostrar una persona que sea igual su edad la introducida por el usuario,  
  ademas de la media de todos. --> You, hace 1 segundo • Uncommitted changes  
  <script>  
    const personas = [{ name: "María", age: 20 },  
      { name: "Bernardo", age: 28 },  
      { name: "Pancracio", age: 22 },  
      { name: "Andrea", age: 19 },  
      { name: "Sara", age: 29 },  
      { name: "Jorge", age: 32 },  
      { name: "Yurena", age: 38 },  
      { name: "Ayoze", age: 18 }  
    ];  
  
    // Guarda la edad introducida por el usuario  
    let edad_entrada = parseInt(prompt("Para buscar una persona, ingrese una edad: "));  
  
    // Guardara la posicion de la persona encontrada  
    let posicion = -1;  
  
    // Recorre el array de objetos  
    for (let i in personas) {  
      // Compara la edad de la persona con la edad introducida por el usuario  
      if (personas[i].age === edad_entrada) {  
        // como se encuentra se guarda la posicion de la persona encontrada  
        posicion = i;  
        // Termina el recorrido.  
        break;  
      }  
    }  
  
    let suma_edades = 0;  
  
    for (let i in personas) {  
      suma_edades += personas[i].age;  
    }  
  
    let persona_encontrada = (posicion !== -1) ?  
      "La primera persona de acuerdo a la edad ingresada es: " + "Nombre: " +  
      personas[posicion].name + " - Edad: " + personas[posicion].age: You  
      "No hay ninguna persona encontrada segun la edad ingresada."  
  
    console.log(persona_encontrada);  
    console.log("La media de todas las personas: " + (suma_edades/personas.length));  
  </script>
```



4. Dado el siguiente array:

```
const array = [ 95, 95, 14, 83, 58, 33, 65, 52, 7, 72, 13, 46, 19, 31, 27, 36, 30, 86,
88, 88, 68, 16, 5, 14, 41, 56, 89, 11, 6, 29, 72, 11, 69, 36, 16, 11, 82, 84, 32, 84, 95,
98, 76, 99, 100, 12, 89, 1, 92, 27, 66, 48, 38, 49, 30, 40, 87, 19, 31, 37, 5, 32, 9, 33,
98, 94, 5, 15, 4, 88, 47, 34, 83, 8, 31, 4, 2, 72, 31, 39, 15, 10, 46, 78, 11, 21, 92, 22,
83, 3, 6, 71, 39, 54, 50, 77, 13, 85, 7, 36 ];
```

- Haz una función que devuelva el array ordenado sin modificar el array original.
 - Haz una función que devuelva los números impares y ordenados.
 - Haz una función de devuelva los números impares de dos cifras.
 - Haz una función que devuelva un array de 0 a 100 con la frecuencia de cada número en el array original.
-

DESARROLLO – EJ4

```
<body>
  <!-- Objetivo: crear funciones que ordenen el array segun el tipo de orden a declarar -->
  <script>

    //Array de numeros desordenados
    const array = [95, 95, 14, 83, 58,
      33, 65, 52, 7, 72, 13, 46, 19, 31, 27,
      36, 30, 86, 88, 88, 68, 16, 5, 14, 41,
      56, 89, 11, 6, 29, 72, 11, 69, 36, 16,
      11, 82, 84, 32, 84, 95, 98, 76, 99, 100,
      12, 89, 1, 92, 27, 66, 48, 38, 49, 30,
      40, 87, 19, 31, 37, 5, 32, 9, 33, 98, 94,
      5, 15, 4, 88, 47, 34, 83, 8, 31, 4,
      2, 72, 31, 39, 15, 10, 46, 78, 11, 21,
      92, 22, 83, 3, 6, 71, 39, 54, 50, 77,
      13, 85, 7, 36];
```

```
/**
 * Funcion que ordena los elementos del array de mayor a menor
 * */
function ordenarArray(array) {
  //Crea una copia del array
  let copia = array.slice();

  // Recorre el array
  for (let i = 0; i < copia.length - 1; i++) {
    // Recorre el array una posicion adelantado
    for (let j = i + 1; j < copia.length; j++) {
      // Se comprueba si la segunda posicion es mayor que la primera del array
      if (copia[j] > copia[i]) {
        let aux = copia[i]; // Guarda el valor
        copia[i] = copia[j]; // La primera posicion cambia al segundo
        copia[j] = aux; // La segunda posicion cambia por el primer valor
      }
    }
  }

  return copia;
}
```

```
/*
 * Funcion que crea un array de num impares segun el array introducido
 */
function numerosImpares(array) {

    // Guardara los num impares
    let num_impares = [];

    // Recorre los numeros
    for (let i = 0; i < array.length; i++) {
        // Comprueba que el num sea impar
        if (array[i] % 2 !== 0) {
            // Agrega al final
            num_impares.push(array[i]);
        }
    }

    return ordenarArray(num_impares);
}
```

```
function imparesDosCifras(array) {
    let resultado = [];

    for (var i = 0; i < array.length; i++) {
        if (array[i] % 2 !== 0 && array[i] >= 10 && array[i] <= 99) {
            resultado[resultado.length] = array[i];
        }
    }

    return resultado;
}
```

```
function frecuencia(arr) {
    var freq = [];

    for (var i = 0; i <= 100; i++) {
        freq[i] = { numero: i, frecuencia: 0 };
    }

    for (var i = 0; i < arr.length; i++) {
        freq[arr[i]].frecuencia = freq[arr[i]].frecuencia + 1;
    }

    return freq;
}
```

```
// Ordenar array
let array_ordenado = ordenarArray(array);
// Imprime el array ordenado
console.log("Array ordenado: " + array_ordenado.join(" "));

// Imprimir los numeros impares
let numeros_impares = numerosImpares(array); //array
console.log("Numeros impares: " + numeros_impares.join(" "));

// Imprime los numeros impares de dos cifras
let impares2c = imparesDosCifras(array);
console.log("Numeros impares de dos cifras: " + impares2c.join(" "));

// Frecuencia
var freq = frecuencia(array);
for (var i = 0; i < freq.length; i++) {
  console.log("Número " + freq[i].numero + " - Frecuencia: " + freq[i].frecuencia );
}
/script>
```

```
Array ordenado: 100 99 98 98 95 95 95 94 92 92 89 89 88 88 88 87 86 85 84 Ej4.html:111
84 83 83 83 82 78 77 76 72 72 72 71 69 68 66 65 58 56 54 52 50 49 48 47 46 46 41 40 39
39 38 37 36 36 36 34 33 33 32 32 31 31 31 31 30 30 29 27 27 22 21 19 19 16 16 15 15 14
14 13 13 12 11 11 11 11 10 9 8 7 7 6 6 5 5 5 4 4 3 2 1

Numeros impares: 99 95 95 95 89 89 87 85 83 83 83 77 71 69 65 49 47 41 39 Ej4.html:115
39 37 33 33 31 31 31 31 29 27 27 21 19 19 15 15 13 13 11 11 11 11 9 7 7 5 5 5 3 1

Numeros impares de dos cifras: 95 95 83 33 65 13 19 31 27 41 89 11 29 11 Ej4.html:119
69 11 95 99 89 27 49 87 19 31 37 33 15 47 83 31 31 39 15 11 21 83 71 39 77 13 85
```

Número 0 - Frecuencia: 0

Número 1 - Frecuencia: 1

Número 2 - Frecuencia: 1

Número 3 - Frecuencia: 1

Número 4 - Frecuencia: 2

.

.

.

Número 95 - Frecuencia: 3

Número 96 - Frecuencia: 0

Número 97 - Frecuencia: 0

Número 98 - Frecuencia: 2

Número 99 - Frecuencia: 1

Número 100 - Frecuencia: 1

5. Vamos a hacer un programa que va pidiendo al usuario que introduzca las notas de un examen y las va guardando en un array. El usuario cada vez puede introducir:

- Una sola nota
- Varias separadas por coma (ej. 4.5, 6, 8.75)
- Se le continuará pidiendo notas hasta que pulse ‘Cancelar’.
- Las notas introducidas se irán almacenando en un array y una vez que estén todas, se mostrará por la consola:
 - El array con los datos suministrados por el usuario
 - El array “limpiado”: quitaremos del array todos los elementos que no sean números o no estén entre 0 y 10.
 - La nota del 1º suspenso
 - El N.º total de aprobados y sus notas
 - La nota media del examen, redondeada a 2 decimales
 - Las notas finales: serán cada nota aumentada un 10% por buena actitud y redondeada a un entero

Por ejemplo, si el usuario va introduciendo las siguientes notas: 9, 4.5, 5, seis, pi, 23, 7, por consola se mostrará:

- Notas introducidas: 9, 4.5, 5, seis, pi, 23, 7.
- Notas válidas: 9, 4.5, 5, 7.
- El primer suspenso es 4.5
- Hay 3 aprobados: 9, 5, 7
- La nota media es 6.38
- Las notas finales son 10, 5, 6, 8

➤ Organizaremos el código en 2 ficheros:

1. index.js: tendrá el código para pedir los datos al usuario y mostrar los mensajes anteriormente indicados. Sólo contendrá el código de pedir las notas, sentencias console.log y llamadas a funciones del fichero functions.js.
2. functions.js: sólo contendrá las funciones llamadas por index.js (deben llamarse EXACTAMENTE como os digo). Son:
 - addItem(notas,item): se le pasa el array actual de notas y el nuevo valor introducido por el usuario y devuelve un array con todos los elementos que tenía más el nuevo introducido, sin modificar.

- `clearItems(notas)`: devuelve el array “limpio” y con sus elementos convertidos a números.
 - `primerSuspenso(notas)`: devuelve el primer elemento que está suspendido.
 - `aprobados(notas)`: devuelve un array con sólo los aprobados. 3
- DAW - Desarrollo Web Entorno Cliente U4 – Array: Programación Funcional.
- `notaMedia(notas)`: devuelve la media de las notas redondeada a 2 decimales.
 - `cambiaNotas(notas, incremento)`: devuelve un nuevo array con cada nota incrementada en el porcentaje indicado y redondeada a 2 decimales

En el fichero `functions.js` utilizaremos métodos de arrays (programación funcional) en lugar de bucles.

EJERCICIO 5: DESARROLLO

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Ejercicio 5</title>
    <meta name="description" content="">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="">
  </head>
  <body>
    <!-- Usado en XAMPP -->
    <script type="module" src="index.js"></script>
  </body>
</html>
```

```
// index.js
import { addItem, clearItems, primerSuspenso, aprobados, notaMedia, cambiaNotas } from './functions.js';

var notas = [];
var entrada;

// Bucle principal
do {
  // Solicita la entrada del usuario
  entrada = prompt("Introduce una nota o varias separadas por coma (Cancelar para terminar:");

  // Si el usuario no cancela ni esta vacío se agrega a notas
  if (entrada !== null && entrada.trim() !== "") {
    notas = addItem(notas, entrada);
  }
} while (entrada !== null);

let notasLimpias = clearItems(notas);
let suspenso = primerSuspenso(notasLimpias);
let aprob = aprobados(notasLimpias);
let media = notaMedia(notasLimpias);
let notasFinales = cambiaNotas(notasLimpias, 10);

console.log("Notas introducidas: " + notas.join(", "));
console.log("Notas válidas: " + notasLimpias.join(", "));
console.log("El primer suspenso es " + suspenso);
console.log("Hay " + aprob.length + " aprobados: " + aprob.join(", "));
console.log("La nota media es " + media);
console.log("Las notas finales son " + notasFinales.join(", "));
```

```
// 1. Añadir elementos a un array sin modificar el original
function addItem(notas, item) {
  // Separa por coma, elimina espacios y concatena
  return notas.concat(item.split(",").map(function(n) { return n.trim(); }));
}

// 2. Limpiar el array: convertir a número y dejar solo 0-10
function clearItems(notas) {
  return notas
    .map(function(n) { return Number(n); })
    .filter(function(n) { return !isNaN(n) && n >= 0 && n <= 10; });
}

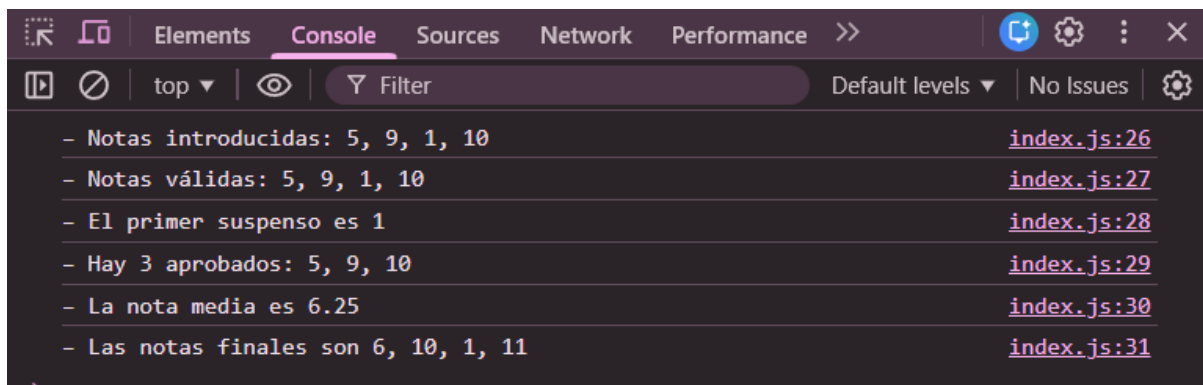
// 3. Primer suspenso (nota < 5)
function primerSuspenso(notas) {
  return notas.find(function(n) { return n < 5; });
}

// 4. Aprobados (nota >= 5)
function aprobados(notas) {
  return notas.filter(function(n) { return n >= 5; });
}

// 5. Nota media, redondeada a 2 decimales
function notaMedia(notas) {
  if (notas.length === 0) return 0;
  return Math.round(notas.reduce(function(acc, n) { return acc + n; }, 0) / notas.length * 100) / 100;
}

// 6. Cambiar notas por un incremento porcentual y redondear a entero
function cambiaNotas(notas, incremento) {
  return notas.map(function(n) { return Math.round(n * (1 + incremento / 100)); });
}

export { addItem, clearItems, primerSuspenso, aprobados, notaMedia, cambiaNotas };
```



Examen: Manipulación objetos, crear objetos o nos da los datos de los objetos. Permite usar apuntes.