**EJERCICIOS Arrays**

1. Sea el siguiente array:

Var números = [1,3,5,7,9,11];

Crea las siguientes funciones:

1. Una función anónima y autoejecutable que emplee un bucle for y que realice la suma de los elementos del array.
2. Una función llamada sumaElementos2, que calcule la suma de los elementos del array, y que será invocada como argumento desde el método .forEach del objeto array.
3. Una función anónima que calcule la suma de los elementos del array. Esta función se pasará como argumento al método .forEach del objeto array:
4. Transforma el código del apartado e, para que, en lugar de la función definida, se emplee una función flecha:
5. Crea un array con los nombres de los siguientes países: España, Francia, Suecia, Italia, Noruega, Portugal, Holanda.

Realiza el menor número de operaciones en el array de forma que al visualizar al final el mismo nos dé como resultado: Bélgica, Alemania, Dinamarca, Francia, Grecia, Holanda, Italia, Noruega, Portugal, Suecia, Suiza.

1. Sea los objetos “alumno” definidos en nuestro fichero .js:

Obj1 = {"nombre":"juan","edad":34};

Obj2 = {"nombre":"eva","edad":24};

Obj3 = {"nombre":"ana","edad":17};

Obj4 = {"nombre":"pepe","edad":31};

Obj5 = {"nombre":"margarita","edad":16};

Obtener:

1. Crea una función (que emplee **parámetros rest**) que permita sumar las edades de todos los alumnos del vector Alumnos. No se permite crear un array para recorrer los objetos.
2. Crea el vector Alumnos formado por los objetos anteriores (busca el método del objeto array para realizar esta tarea con una única orden).
3. Crea un vector llamado AlumnosMayoresDeEdad con los alumnos que tienen >= 18 años.
4. Partiendo del vector Alumnos, obtener un array que contenga solo los nombres de los mismos (emplear una función flecha).
5. Obtener los objetos que forman el array AlumnosMayoresDeEdad y almacenarlos en objeto1, objeto2 y objeto3 respectivamente con una única orden.
6. Contar el número de alumnos que tienen menos de 18 años.
7. Ordenar el array Alumnos por edad (de menor a mayor).
8. Sean los objetos Obj6, y Obj7 siguientes:

Obj6 = {"nombre":"jerónimo","edad":19};

Obj7 = {"nombre":"noa","edad":20};

Se desea añadirlos al vector AlumnosMayoresDeEdad empleando el operador de propagación;

1. Localiza al alumno “jerónimo” en el vector AlumnosMayoresDeEdad con una única orden (empleando una función flecha) y guarda el objeto en la variable jero.
2. Desestructura al objeto jero creado en el apartado anterior:
3. Crea un programa que permita introducir un número entero n. Comprueba que sea mayor que cero. A continuación se creará matriz cuadrada (formada por n filas y n columnas). Como contenido de cada celda se generará un número aleatorio entre 1 y 10 (No se comprobará si un número se repite). Por ejemplo, si se han introducido el valor 4 el resultado deberá ser algo similar a:

**Elementos del array:**

2 3 1 6

9 3 4 2

6 6 8 7

1 5 5 3

Además se deberá mostrar:

1. El numero total de elementos del array
2. La suma de las todos los elementos situados en columnas pares (el valor 0 se considera par).
3. La suma de todos los elementos situados en la diagonal (comenzando en la fila 0, columna 0)
4. La matriz traspuesta equivalente

La matriz traspuesta se calcula cambiando las filas y las columnas, como en el ejemplo siguiente:

2 3 1 6

9 3 4 2

6 6 8 7

1 5 5 3

**Matriz Traspuesta:**

2 9 6 1

3 3 6 5

1 4 8 5

6 2 7 3

1. Se desea implementar la clase **Cola**. Una cola es una estructura de datos en la que las inserciones se hacen al final y los borrados de elementos se hacen siempre por el principio. Para implementar la estructura cola utilizaremos un array. Es decir, cuando añadamos un elemento, se añadirá al final del array, y si borramos, extraeremos el que se encuentra en la primera posición.

Los **atributos** de la clase cola son:

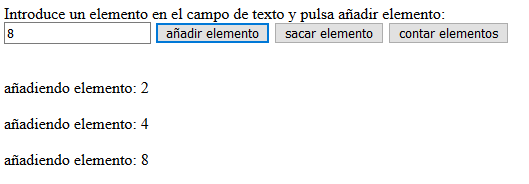
* Elementos, array donde se almacenan los “elementos”

Las **operaciones** que se definen para la clase Cola son las siguientes:

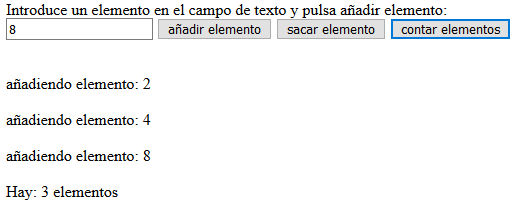
* constructor(), crea el array vacío.
* anadirElemento(e), donde e es el elemento que se añade.
* extraerElemento(), devolverá el elemento situado al principio de la cola.
* consultarNumeroElementos(), devolverá el número de elementos que haya en la cola.
* estaVacia(), esta función devolverá true si la cola no tiene elementos. False en caso contrario.

Emplea el fichero UD4\_cola.html que se adjunta que dispone de la siguiente interfaz:



Al pulsar el botón añadir elemento, se añade el contenido del cuadro de texto a la cola y se visualiza el mensaje siguiente:

Al pulsar el botón contar elementos, se muestra el número de elementos que haya en la cola



Al pulsar en el botón sacar elemento, se muestra y elimina el primer elemento de la cola.

