

Ejercicios Unidad 4 – Arrays, funciones y objetos

Realiza cada ejercicio en una carpeta diferente dentro de la carpeta Estructuras de la unidad 4

Parte 1: Arrays

1. Ejercicio

Dado un array de enteros, obtener el mayor, el menor y la media. Los números del array se leerán desde un input donde se introducirán separados por comas. Al pulsar el botón “Calcular” se obtendrán los resultados. Vista de ejemplo:

Ejercicio 1: Mayor, Menor y Media de Arrays

Introduce los números separados por comas

2. Ejercicio

- Declara un array que “clasificacion” con los siguientes valores:
Ana, Oswaldo, Raúl, Celia, María, Antonio (vamos a suponer que es el orden de clasificación de un concurso, en un momento dado)
- Imprime la clasificación inicial usando for. Indica la posición de cada concursante mediante document.write.
- El concurso continúa, y se van modificando esas posiciones anteriores. Debemos cambiar en el array:
 - Celia adelanta a Raúl
 - Antonio es descalificado y se elimina del concurso
 - Detrás de Ana y antes de Oswaldo se clasifican dos nuevos concursantes: Roberto y Amaya, en ese orden
 - Hay una nueva participante que pasa a encabezar la clasificación: Marta
- Imprime la clasificación actualizada y comprueba que se ha hecho correctamente (hazlo usando for..in y for..of)

Clasificación inicial

1. Ana
2. Oswaldo
3. Raúl
4. Celia
5. María
6. Antonio

3. Ejercicio concatenar arrays de igual longitud

Concatenar dos arrays A y B de la siguiente forma: A0, B0, A1, B1, A2, B2, ..., An, Bn. Los vectores tendrán la misma longitud, si no la tienen, el programa debe avisar y no realizar la concatenación

Ejercicio: Concatenar 2 Arrays de igual longitud

Introduce los primeros números separados por comas

Introduce los segundos números separados por comas o pulsa para generarlo aleatoriamente

4. Ejercicio concatenar arrays de distinta longitud

Repite el ejercicio anterior permitiendo arrays de diferente longitud. Para ello, insertaremos en ambos casos, tantos aleatorios como se indique en una lista desplegable del 1 al 5

Ejercicio: Concatenar 2 Arrays de diferente longitud

Introduce los primeros enteros separados por comas

Introduce los segundos enteros separados por comas

75,41,71,93,77,30,16,63

5. Ejercicio

Leer una cadena de texto en un formulario y generar un array con la función `split()`.

Posteriormente, mostrar la siguiente información:

- a. Número de palabras
- b. primera palabra
- c. última palabra
- d. las palabras colocadas en orden inverso
- e. las palabras ordenadas de la a la z
- f. las palabras ordenadas de la z a la a.

Sacar toda esta información en una ventana nueva.

6. Ejercicio

Escribe una función iterativa que dado un número entero `n`, retorne un array con todos los números enteros en orden decreciente desde `n` a 1.

7. Ejercicio. Quitar repetidos

Quitar los elementos repetidos de un array de números y ordénalos de menor a mayor.

8. Ejercicio. Recorridos

Dado un Array de números devolver un Array en donde cada número sea multiplicado por dos.

- a. Con `for`, `for..in`, `for..of`
- b. Con `map`

9. Ejercicio

Dado un array de cadenas:

- a. muestra la primera cadena del array que tenga más de 5 letras
- b. muestra la posición de la primera cadena del array que tenga más de 5 letras
- c. elimina todos aquellos que tengan más de 5 letras

Parte 2: Conjuntos y Mapas

10. Ejercicio Mapa de Módulos

Utiliza una estructura *map* para almacenar información sobre módulos impartidos en 2º DAW. Cada módulo es un par clave-valor: (*Abreviatura, Nombre*) por ejemplo: ("DWECE", "Desarrollo Web en Entorno Cliente").

Requisitos:

- Añade la información con posterioridad a la creación de la estructura.
- Toda la información se mostrará en la página HTML bajo el título <h1>Mapa de Módulos</h1>. Evitar uso de console.log en resultado. Distinguir bien cada apartado en la visualización.
- Usa comentarios de manera adecuada.
- Muestra cuántos módulos hay almacenados
- Muestra el contenido de la estructura
- Devuelve las abreviaturas de todos los módulos guardados
- Devuelve el nombre completo de todos los módulos
- Consulta si está el módulo "DIW"
- Si está, elimínalo y comprueba que se ha eliminado
- Vacía el mapa.

11. Ejercicio Conjunto de Matrículas

A partir del siguiente mapa de matriculas de 5 alumnos (cada matricula es un array de módulos).

Juan, [DIW, DWECE]

Ana, [DIW, DWES, DWECE, Ingles]

Maria, [DIW, DWES, DAWEB, BBDD, Inglés]

Pepe, [DWES, DWECE]

Eva, [DIW, DAWEB, SI]

De modo parecido al ejercicio anterior:

- Construye el mapa llamado matrículas
- Obtén el número de alumnos matriculados en DIW
- Obtén el conjunto de todos los módulos de los que se han matriculado los alumnos (pista: propaga en un array para construir el conjunto) y muéstralos.
- Indica el número total de módulos
- Obtén los módulos que tienen en común María y Eva (intersección)

Enlace de ayuda:

<https://code.tutsplus.com/es/tutorials/understanding-sets-with-javascript--cms-29789>

12. Repetidos

Repite el ejercicio 7 usando conjuntos.

13. De conjunto a Array

Convierte el conjunto del ejercicio 12 a array

14. Filtrado y mapeado de conjuntos

A partir del siguiente conjunto: `var mySet = new Set().add('foo').add('bar').add('foobar');`

- Convertir todos los elementos a mayúsculas
- Obtener aquellos elementos con menos de 4 letras

15. Dados

- Realizar un programa en JavaScript que simule el lanzamiento de dos dados.
- Repetir dicha acción 100 veces, almacenando en un array el resultado de cada dado y la suma de ambos dados.
- Indicar cuántas veces han salido los ojos de serpiente (1 – 1) y el seis doble (6 – 6).
- Guardar en un conjunto las diferentes combinaciones que han aparecido.

16. DNI

Crear un proyecto donde introduzcamos a través de un formulario en HTML un campo de texto y un botón de enviar. En el campo de texto escribiremos un DNI (8 dígitos y letra). Seguidamente al pulsar el botón enviar, calculará la letra del DNI.

El cálculo de dicha letra se realiza a través del **siguiente algoritmo**:

Teniendo un array con las siguientes letras:

```
var letras = ['T', 'R', 'W', 'A', 'G', 'M', 'Y', 'F', 'P', 'D', 'X', 'B', 'N', 'J', 'Z', 'S', 'Q', 'V', 'H', 'L', 'C', 'K', 'E', 'T'];
```

Calculamos el resto de dividir nuestro número de DNI entre 23. En función del resto obtenido así elegimos la letra. (Si el resto es 0, entonces la letra es 'T').

Condicionantes a aplicar:

- En primer lugar, se debe de comprobar que el usuario ha introducido 8 dígitos y una letra al final en una sola instrucción.
- En una sola instrucción se debe comprobar si el número es menor que 0 o mayor que 99999999 y si ese es el caso, se muestra un mensaje al usuario indicando que el número proporcionado no es válido y el programa no muestra más mensajes.
- Si el número es válido, se calcula la letra que le corresponde según el método explicado anteriormente.
- Una vez calculada la letra, se debe comparar con la letra indicada por el usuario. Si no coinciden, se muestra un mensaje al usuario diciéndole que la letra que ha indicado no es correcta. En otro caso, se muestra un mensaje indicando que el número y la letra de DNI son correctos.

17. Accesos

- a) Crea un mapa de 5 usuarios donde se almacena para cada *username* el número de veces que acceden al sistema y la fecha del último acceso (obtenida de manera aleatoria).
- b) Obtén el *username* del usuario que más veces ha accedido y el que menos.
- c) Obtén el *username* del usuario que accedió el primero y elimínalo.