

Desarrollo web en entorno cliente – Práctica 3

INSTRUCCIONES DE ENTREGA:

Organiza tus soluciones para cada ejercicio en carpetas individuales con el nombre correspondiente de cada ejercicio. Comprime todas las carpetas en un único archivo ZIP y súbelo a la Aules.

Ejercicio 1 (1 punto)

Realiza una función llamada “cadAleatoria” a la que le podamos pasar tantos parámetros como queramos, pero esos parámetros deberán ser siempre cadenas de caracteres.

La función deberá devolver de forma aleatoria una de las cadenas de caracteres que le hemos pasado como argumento.

Ejercicio 2 (2 puntos)

Realiza una función llamada “*invierteParcial*” que acepte dos parámetros: un array y un número entero.

La función debe devolver un nuevo array en el que se invierte una porción específica del array original. Esta porción consistirá en los elementos desde el inicio del array hasta el índice especificado por el número entero.

Ejemplo:

array original: [3, 2, 1, 7, 8, 10], numero: 4 → array resultado: [7, 1, 2, 3, 8, 10]

Ejercicio 3 (3 puntos)

Realiza una función constructora para crear objetos de tipo “Personaje”, que contenga las propiedades siguientes:

Nombre: almacenará un String.

Fuerza: almacenará un número entero. Deberá ser entre el 1 y el 20.

Resistencia: almacenará un número entero. Deberá ser entre el 1 y el 20.

Inteligencia: almacenará un número entero. Deberá ser entre el 1 y el 20.

Sabiduría: almacenará un número entero. Deberá ser entre el 1 y el 20.

Una vez creado el constructor crea un array que almacene 6 objetos Personaje creados por ti con diferentes valores en sus propiedades.

Realiza una función llamada “ordenaLuchador” al que se le pase por argumento el nombre de una de las propiedades: *Fuerza*, *Resistencia*, *Inteligencia* o *Sabiduría*, y muestre por consola el nombre de los 6 luchadores almacenados en el array anterior, ordenados de manera descendiente por el valor de la propiedad pasada por argumento.

Ejercicio 4 (4 puntos)

Realiza una función constructora para crear objetos de tipo “Coche”, que contenga las propiedades siguientes:

- *fMat*: almacenará la fecha de matriculación del coche. Deberá ser de tipo Date.
- *matricula*: guardará la matrícula del coche. Deberá ser un número entero.
- *marca*: guardará la marca del coche. Deberá ser un string.
- *motor*: indicará el estado en el que se encuentra el motor, puede estar “apagado” o “arrancado”.

Y los métodos siguientes:

- *arrancar*: establece el estado del motor en “arrancado”
- *apagar*: establece el estado del motor en “apagado”

Una vez creado el constructor, realiza una función llamada “*flotaDeCoches*” que admita por parámetro un número entero (n) y devuelva un array con n coches creados automáticamente siguiendo la siguiente lógica:

- El primer coche debe tener como fecha de matriculación, el día 1 de enero de 2000, el segundo el día 2 y así consecutivamente.
- La matrícula del primer coche debe ser el número 0, la del segundo el 1 y así consecutivamente.
- La marca del coche se establecerá en el siguiente orden: primero “Toyota”, luego “Mercedes”, luego “Opel”, luego “Seat” y luego “Peugeot”. Si el número fuera mayor a 5 debe volver a empezar el orden de nuevo por “Toyota”, luego “Mercedes”, etc
- El estado del motor será establecido aleatoriamente entre “arrancado” y “apagado”.

Por último, realiza un script que muestre por pantalla cada coche generado por la función anterior como una lista no ordenada () cuyos ítems sean las propiedades del mismo y su valor.