

Práctica 3 – Funciones en JavaScript

Sobre la práctica: los ejercicios aquí presente son opcionales. Aquellos que se deban entregar serán especificados en el aula virtual del campus del Instituto y se enviarán por ese medio.

FUNCIONES

1. Crear una variable llamada “nombreYApellido” con un array que contenga dos strings (el primero sería un nombre y el segundo un apellido). Crear una función que imprima en consola la frase “Su nombre es *nombre en el array* y su apellido es *apellido del array*”.
2. Crear una función “elementoN” que toma un array y un número y nos retorna el elemento que se encuentra en ese número de dirección en el array.
3. Crear una función “dobleDelPrimero” que toma el primer elemento de un array, lo multiplica y lo retorna el mismo array con ese elemento agregado al principio.
4. Creemos una variable global llamada “pendientes” conteniendo un array de strings. Crearemos tres funciones: una “agregarTarea” que agrega un string más al final del array, otra de “tareaResuelta” que elimine el primer elemento del array y una “tareaUrgente” que agregue un string al principio del array.
5. Crear una función llamada es divisible, a la que le pasamos dos números y nos dice si el primer número es divisible por el segundo. Debe retornar un valor booleano.
6. Crear una función llamada “cupoLleno” que compruebe si un array (que pasaremos como argumento) tiene una longitud mayor a 6. En caso positivo, retornar true, en caso negativo false.
7. Creemos dos variables globales: una llamada “canción” que albergue un string y otra variable “incluido” con un valor booleano. Crear una función que reciba como argumento el nombre de una canción y que, en caso de que “incluido” sea true, hará que la consola arroje “Canción ya incluida”. En caso falso, cambiará la variable “canción” por el valor del argumento y hará que la consola diga “¡Nueva canción incluida!”. Luego de eso, la consola deberá imprimir el valor de la variable “canción”.
8. Crear una función llamada “descuento” que toma un número, y si ese número es mayor a 2000, deberá restarle un 25% de ese número.
9. Crear una función llamada “parteDelDía”, a la cual se le pasa un número y, en caso de que el número sea entre 6 y 12, imprima en la consola “Es mañana”, si es entre 13 y 19 imprima “Es de tarde”, y si es entre 20 y 24 o entre 1 a 5, diga “Es de noche”. ¿Qué requisitos tiene esta función? Anotarlos como comentario dentro de la misma.
10. Crear una función llamada “reemplazar” que tome un array, un número de dirección dentro del array, y un elemento. El objetivo de esta función es retornar un nuevo array donde el elemento que estaba en la dirección dada sea eliminado y, en su lugar, se coloque el elemento que pasamos por parámetro (recuerden el uso de la función splice de los arrays).
11. Crear una función “compararCadenas” que toma dos strings y nos dice cuál de los dos es más extenso o si son iguales.
12. Crear una función “imprimirArray” que, tomando un array, imprima todos los elementos que contiene.
13. Crear una función “caracteresTotales” que, tomando un array de strings, nos sume que tan largo son los strings que contiene en total

14. Crear una función “limpiarDelImpares” que, tomando un array de números, lo retorne sin los números impares (pueden usar una función auxiliar que verifique si un numero es impar)
15. Crear una función que imprima una línea de asteriscos con una longitud que pasamos por parámetro. Usando esto, crear una función que imprima un triangulo de asteriscos de 5 pisos.
16. Creemos el objeto “telefono” que tiene por propiedad un número, contactos (un array con strings) y un proveedor (un string con el nombre de la empresa proveedora). Crear dos métodos para teléfono: uno que se llame «sonar» y que haga que la consola imprima «Ring ring ring» y otro que sea “llamar” y que tome un número, busque en sus contactos ese dirección dentro del array e imprima en la consola «Llamando a [nombre de la dirección]». Por ejemplo, si tengo de contactos a [“Martín”, “Lucas”, “Karen”] y hago telefono.llamar(1) me devuelve “Llamando a Lucas”.