



# Ejercicios Básicos de JS

*El objetivo de este documento es introducir al lenguaje JS aplicando conceptos básicos de programación.*

1. Crear una función que cuente el número de caracteres de una cadena de texto.
2. Crear una función que devuelva la subcadena indicada desde el inicio de la cadena hasta una posición definida. Por ejemplo:  
`tuFuncion("TaeKwonDo", 6)` retornará "TaeKwo".
3. Crear una función que dado un String retorne un arreglo de subcadenas separadas por algún caracter específico. Por ejemplo:  
`tuFuncion('TaeKwonDo*es*un*deporte*olimpico', '*')` retornará `['Taekwondo', 'es', 'un', 'deporte', 'olimpico']`.
4. Crear una función que muestre una determinada cantidad de veces algún texto definido y el número de la repetición separados por espacio. Por ejemplo:  
`tuFuncion('TaeKwonDo', 4)` devolverá `1.TaeKownDo 2.TaeKownDo 3.TaeKownDo 4.TaeKownDo`.
5. Crear una función para contar la cantidad de veces que aparece una palabra dada en una frase dada. Por ejemplo:  
`tuFuncion("TaeKwondo es más que un deporte, es deporte olímpico", "deporte")` devolverá `2`.
6. Dada una cadena de caracteres que tiene un patrón no deseable, crear una función que elimine dicho patrón de la cadena. Por ejemplo:  
`tuFuncion("<li>Lunes, <li>Martes, <li>Miércoles, <li>Jueves, <li>Viernes", "<li>")` retornará `"Lunes, Martes, Miércoles, Jueves, Viernes"`.
7. Mejora tu código para que al recibir una cadena así:  
`"<li>Lunes</li> <li>Martes</li> <li>Miercoles</li> <li>Jueves</li> <li>Viernes</li>"`.  
Puedas retornar un arreglo así:  
`["Lunes", "Martes", "Miércoles", "Jueves", "Viernes"]`.
8. Crear una función que te permita obtener un número aleatorio entre 0 y 1000.



9. Crear una función que te permita obtener un número aleatorio entre 0 y 1 con 2 decimales.
10. Crear una función que determine si un número es decimal.
11. Crear una función que permita obtener el resultado de aplicar un descuento en porcentaje a un valor dado. Por ejemplo:  
Si el valor total es 10000 y el descuento es del 15% entonces, deberá devolver 8500.
12. Crear una función que dada una fecha de nacimiento permita obtener la edad de la persona.
13. Crear una función que permita decir si una cadena de caracteres cumple con un formato aceptable de email, esto es, más de 2 caracteres para el identificador, el símbolo @ y dominios de correo aceptables. Podés utilizar expresiones regulares.
14. Crear una función que permita decir que una cadena de caracteres contiene un nombre y un apellido. Como premisas iniciales, considere que no puede contener caracteres especiales, puede contener mayúsculas y minúsculas y cada palabra debe estar separada por un espacio.
15. Dado un arreglo de números enteros, crear una función que retorne el mismo arreglo con los valores elevados al cubo.
16. Dado un arreglo con números decimales obtener el mínimo y el máximo valor del arreglo.
17. Crear una función que, dado un arreglo de números retorne un objeto con dos arreglos. En uno de ellos debe contener los valores menores a un valor dado y en el otro debe contener los valores mayores o iguales al valor dado.
18. Mejorar la función del ejercicio anterior para que los valores de los arreglos del objeto retornado estén ordenados crecientemente (los que son menores al valor dado) y decrecientemente (los que son mayores al valor dado).
19. Crear una función que elimine los valores duplicados de un arreglo.
20. Mejorar el código del ejercicio 19 para que los arreglos del objeto retornado no tengan valores duplicados.
21. Crear una función que elimina valores duplicados de cualquier arreglo, con diferentes valores y tipos de datos.
22. Crear una función para obtener el promedio de un arreglo de números.

## Ejercicio Obligatorio

**Fecha de entrega: Lunes 17/04/2023**

- 1) Crear una clase llamada **Competidor** que refiere a un competidor de una competencia de Taekwondo. Tenga en cuenta que el competidor a ser cargado es de la categoría que vienen trabajando en prácticas anteriores.

La clase recibirá un objeto al momento de instanciarse con los siguientes datos: legajo del competidor (GAL: Global Athletic License), Apellido, Nombre, DU, fecha de nacimiento, país de origen, graduación (refiere al cinturón que ha logrado en sus años de aprendizaje del arte marcial. Valores posibles: 1ro a 10mo GUP y de 1ro a 9no DAN), clasificación general nacimiento (refiere a su valor en el ranking nacional), email de contacto, género (sólo permite masculino o femenino).

Restricciones de los datos:

- Todos los datos del objeto son obligatorios.
- El GAL tiene 9 caracteres, los primeros 3 son letras y los 7 restantes son números. Por ejemplo: ABC1234567
- El apellido y nombre no rebasan los 100 caracteres.
- El email tiene que ser un email válido.
- La edad del competidor debe ser de 6 años o más.
- El país debe estar dentro de los países aceptados.
  - Rusia, Ucrania, Estados Unidos, Canada, Perú, Chile, Argentina, México, Brasil, Bolivia, Ecuador, Venezuela, Colombia, Paraguay, Uruguay.
  - Crear un método que contenga el listado de países aceptados. Utilice un JSON para obtener la información a validar.



- El género debe estar dentro de los valores aceptados.
- Validar que el ranking sea un número entre 0 y 900 pudiendo ser decimal de una posición.

- 2) Crear un formulario para cargar todos los datos del competidor.
- 3) Crear un método que devuelva todo el perfil del competidor.
- 4) A partir de un arreglo con la información de 3 competidores genere 3 instancias de la clase de forma automatizada e imprima el perfil de cada competidor.