

## Ejercicios Propuestos

1. Programa una función que cuente el número de caracteres de una cadena de texto.
2. Programa una función que te devuelva el texto recortado según el número de caracteres indicados.
3. Programa una función que dada una String te devuelva un Array de textos separados por cierto carácter. Por ejemplo. *miFuncion("hola que tal", ' ');* devuelve *["hola","que","tal"]*.
4. Programa una función que repita un texto dado n veces.
5. Programa una función que invierta las palabras de una cadena de texto.
6. Programa una función para contar el numero de veces que se repite una palabra en un texto largo.
7. Programa una función que valide si una palabra es PALINDROMO.
8. Programa una función que elimine cierto patrón de caracteres de un texto dado. Por ejemplo. *miFuncion("xyz1, xyz2, xyz3, xyz4 y xyz5", "xyz");* Devuelve *"1,2,3,4 y 5"*
9. Programa una función que obtenga un número aleatorio entre 501 y 600.
10. Programa una función que reciba un número y evalúe si es capicúa o no (que se lee igual en un sentido que en otro). Por ejemplo. Con 2002 devuelve true.
11. Programa una función que calcule el factorial de un número.
12. Programa una función que determine si un número es primo o no.
13. Programa una función que determine si un número es par o impar.
14. Programa una función para convertir grados Celsius a Farenheit y viceversa.
15. Programa una función para convertir números de base binaria a decimal y viceversa.
16. Programa una función que devuelva el monto final después de aplicar un descuento a una cantidad dada.
17. Programa una función que dada una fecha válida determine cuantos años han pasado hasta el día de hoy. (La función recibe como parámetro una fecha.)
18. Programa una función que dada una cadena de texto cuente el número de vocales y consonantes que hay en ella.
19. Programa una función que valide que un texto sea un nombre válido.
20. Programa una función que valide que un texto sea un email válido.
21. Programa una función que dado un array numérico devuelve otro array con los números elevados al cuadrado.
22. Programa una función que dado un array devuelva el número más alto y el más bajo de dicho array.

23. Programa una función que dado un array de números devuelva un objeto con 2 arreglos, en el primero almacena los números pares y en el segundo los impares.
24. Programa una función que dado un arreglo de números devuelva un objeto con dos arreglos. El primero tendrá los números ordenados en forma ascendente y el segundo de forma descendente.
25. Programa una función que, dado un arreglo de elementos, elimine los duplicados.
26. Programa una función que dado un arreglo de números obtenga el promedio.
27. Teniendo las siguientes entradas en una función "scissors", "paper", "scissors", "rock", "paper", "paper". Obtener las siguientes salidas en consola. "Player 1 won!", "Player 2 won!", "Draw!"
28. Escribe una función que reciba dos cadenas de texto y devuelva la mezcla de ellas a partir de los caracteres que tengan en común. Si no tienen caracteres en común, solo concatenan una después de otra.

```
// "abcde" + "cdefgh" => "abcdefgh"
// "abaab" + "aabab" => "abaabab"
// "abc" + "def" => "abcdef"
// "abc" + "abc" => "abc"
```

29. Crear una función que convierta una cadena en formato Snake case a lowerCamelCase y viceversa.

```
//Snake case --> hola_a_todos
//lowerCamelCase --> holaComoEstan;
```

30. Crear un sistema para jugar 21

Reglas. Participan 5 jugadores. La baraja debe de barajearse antes de empezar.

A cada jugador se le reparten 2 cartas de forma aleatoria.

El joker vale 10

El Jack vale 10

La queen vale 10

El King vale 10

El As vale 1 o 11 pero para el ejercicio vale 11.

El jugador que tenga la suma más cercana a 21 GANA

El jugador que se pase de 21 pierde en automático.

Si hay empate, gana el jugador que tenga la carta más alta.

Se debe de mostrar el resultado de cada jugador y el ganador

Se debe mostrar en que momento pierden los jugadores. Si se pasan de 21 o si al comparar con el resto de los jugadores y el lugar en el que quedan.

### 31. Programa una clase llamada Película.

*La clase recibirá un objeto al momento de instanciarse con los siguientes datos: id de la película en IMDB, título, director, año de estreno, país o países de origen, géneros y calificación en IMBD.*

- *Todos los datos del objeto son obligatorios.*
- *Valida que el id IMDB tenga 9 caracteres, los primeros 2 sean letras y los 7 restantes números.*
- *Valida que el título no rebase los 100 caracteres.*
- *Valida que el director no rebase los 50 caracteres.*
- *Valida que el año de estreno sea un número entero de 4 dígitos.*
- *Valida que el país o países sea introducidos en forma de arreglo.*
- *Valida que los géneros sean introducidos en forma de arreglo.*
- *Valida que los géneros introducidos están dentro de los géneros aceptados\*.*
- *Crea un método estático que devuelva los géneros aceptados\*.*
- *Valida que la calificación sea un número entre 0 y 10 pudiendo ser decimal de una posición.*
- *Crea un método que devuelva toda la ficha técnica de la película.*
- *A partir de un arreglo con la información de 3 películas genera 3 instancias de la clase de forma automatizada e imprime la ficha técnica de cada película.*

*\* Géneros Aceptados: Action, Adult, Adventure, Animation, Biography, Comedy, Crime, Documentary, Drama, Family, Fantasy, Film Noir, Game-Show, History, Horror, Musical, Music, Mystery, News, Reality-TV, Romance, Sci-Fi, Short, Sport, Talk-Show, Thriller, War, Western.*