

Ejercicios Unidad III

1. Contraseña válida

Escribir una función llamada `contrasenaValida` que reciba un string y retorne `true` si el string es igual a "2Fj(jjbFsuj" o "eoZiugBf&g9". De lo contrario debe retornar `false`.

```
// escribe tu respuesta acá

// código de prueba
console.log(contrasenaValida("2Fj(jjbFsuj")) // true
console.log(contrasenaValida("eoZiugBf&g9")) // true
console.log(contrasenaValida("hola")) // false
console.log(contrasenaValida("")) // false
```

2. Calcular impuestos

Escribir una función llamada `calcularImpuestos` que reciba dos argumentos numéricos: `edad` e `ingresos`. Si `edad` es igual o mayor a 18 y los ingresos son iguales o mayores a 1000 debe retornar `ingresos` * 40%. De lo contrario retornar 0.

```
// escribe tu respuesta acá

// código de prueba
console.log(calcularImpuestos(18, 1000)) // 400
console.log(calcularImpuestos(40, 10000)) // 4000
console.log(calcularImpuestos(17, 5000)) // 0
console.log(calcularImpuestos(30, 500)) // 0
```

3. IMC (índice de masa corporal)

El índice de masa corporal (IMC), o BMI por sus siglas en inglés, es un valor que determina la cantidad de grasa de una persona.

El BMI se calcula con la siguiente formula: $\text{peso} / \text{altura}^2$

Escribir una función llamada `bmi` que reciba dos argumentos: peso y altura, y retorne un string con las siguientes posibilidades:

- * "Bajo de peso" si el BMI < 18.5
- * "Normal" si está entre 18.5 y 24.9
- * "Sobrepeso" si está entre 25 y 29.9
- * "Obeso" si es igual o mayor a 30

```
// escribe la función bmi acá

// código de prueba
console.log(bmi(65, 1.8)) // "Normal"
console.log(bmi(72, 1.6)) // "Sobrepeso"
console.log(bmi(52, 1.75)) // "Bajo de peso"
console.log(bmi(135, 1.7)) // "Obeso"
```

4. Imprimir un arreglo

Escribir una función llamada `imprimirArreglo` que reciba un arreglo e imprima cada elemento en una línea a parte:

```
// escribe tu respuesta acá

// código de prueba
console.log(imprimirArreglo(1, "Hola", 2, "Mundo"))
// 1
// Hola
// 2
// Mundo
```

5. Número de Likes

Escribe una función llamada `likes` que reciba un número y retorne un string utilizando el formato de K para miles y M para millones.

Por ejemplo:

- * 1400 se convierte en 1K
- * 34,567 se convierte en 34K
- * 7'456,345 se convierte en 7M.

Si el número es menor a 1000 se debe devolver el mismo número como un string.

```
// escribe tu respuesta acá

// código de prueba
console.log(likes(983)) // "983"
console.log(likes(1900)) // "1K"
console.log(likes(54000)) // "54K"
console.log(likes(120800)) // "120K"
console.log(likes(25222444)) // "25M"
```

6. FizzBuzz

Escribir una función llamada `fizzBuzz` que reciba un número y retorne un string de acuerdo a lo siguiente:

- * "fizz" si el número es múltiplo de 3.
- * "buzz" si el número es múltiplo de 5.

* "fizzbuzz" si el número es múltiplo tanto de 3 como de 5.

* Si no cumple ninguna de las condiciones anteriores debe retornar el mismo número.

```
// escribe tu respuesta acá

// código de prueba
console.log(fizzBuzz(6)); // "fizz"
console.log(fizzBuzz(20)); // "buzz"
console.log(fizzBuzz(30)); // "fizzbuzz"
console.log(fizzBuzz(8)); // 8
```

7. Contar rango de números

Escribir una función llamada `contarRango` que reciba dos números y retorne cuántos números que hay entre ellos (excluyéndolos):

****Nota:**** Utiliza un ciclo en tu solución. Puedes asumir que el primer número va a ser menor que el segundo.

```
// escribe tu respuesta acá

// código de prueba
console.log(contarRango(1, 9)) // 7
console.log(contarRango(1332, 8743)) // 7410
console.log(contarRango(5, 6)) // 0
```

8. Sumar rango de números

Escribir una función llamada `sumarRango` que reciba dos argumentos: número inicial y número final. La función debe retornar la suma de los números en ese rango (incluyéndolos).

****Nota:**** puedes asumir que el número inicial va a ser menor o igual que el número final.

```
// escribe tu respuesta acá

// código de prueba
console.log(sumarRango(0, 10)) // 55
console.log(sumarRango(12, 14)) // 39
console.log(sumarRango(5, 5)) // 0
```

9. Número de aes (letra "a")

Escribir una función llamada `numeroDeAes` que reciba un string y retorne el número de veces que aparece la letra "a":

```
// escribe tu respuesta acá

// código de prueba
console.log(numeroDeAes("abracadabra")) // 5
console.log(numeroDeAes("etinol")) // 0
console.log(numeroDeAes("")) // 0
```

10. Número de caracteres

Escribir una función llamada `numeroDeCaracteres` que reciba un string y un caracter (un string de un caracter). La función debe retornar el número de veces que aparece el caracter en el string.

```
// escribe tu respuesta acá

// código de prueba
console.log(numeroDeCaracteres("Hola Mundo", "o")) // 2
console.log(numeroDeCaracteres("MMMMM", "m")) // 0
console.log(numeroDeCaracteres("eeee", "e")) // 4
```

11. Sumar arreglo

Escribir una función llamada `sumarArreglo` que reciba un arreglo de números y retorne la suma de todos los elementos.

```
// escribe tu respuesta acá

// código de prueba
console.log(sumarArreglo([3, 1, 2])) // 6
console.log(sumarArreglo([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])) // 55
console.log(sumarArreglo([])) // 0
```

12. Multiplicar arreglo

Escribir una función llamada `multiplicarArreglo` que reciba un arreglo de números y retorne la multiplicación de todos los elementos.

```
// escribe tu respuesta acá

// código de prueba
console.log(multiplicarArreglo([4, 1, 2, 3])) // 24
console.log(multiplicarArreglo([1, 2, 3, 4, 5, 6, 7, 8])) // 40320
console.log(multiplicarArreglo([])) // 1
```

13. Remover ceros

Escribir una función llamada `removeCeros` que reciba un arreglo de números y retorne un nuevo arreglo excluyendo los ceros (0).

```
// escribe tu respuesta acá

// código de prueba
console.log(removerCeros([0, 1, 0, 2, 0, 3])) // [1, 2, 3]
console.log(removerCeros([9, 3, 6, 4])) // [9, 3, 6, 4]
console.log(removerCeros([0, 0, 0])) // []
```

14. Sumar arreglo entre el rango

Escribir una función llamada `sumarArreglo` que reciba tres argumentos: un arreglo de números, la posición inicial y la posición final. La función debe retornar la suma de todos los números dentro del rango (la posición inicial y la posición final, incluyéndolas).

****Nota:**** puedes asumir que la posición inicial va a ser menor o igual a la posición final, y que están dentro de los límites del arreglo.

```
// escribe tu respuesta acá

// código de prueba
console.log(sumarArreglo([1, 2, 3], 1, 2)) // 5
console.log(sumarArreglo([1, 2, 3, 4, 5, 6, 7, 8, 9, 10], 3, 6)) // 22
console.log(sumarArreglo([7, 8, 9], 0, 0)) // 0
```

15. Transcribir ADN a ARN

Escribir una función llamada `transcribir` que reciba un string (una cadena de ADN) y retorne otro string (su complemento ARN).

Los complementos son los siguientes:

- * G -> C
- * C -> G
- * T -> A
- * A -> U

```
// escribe tu función acá

// código de prueba
console.log(transcribir("ACGT")) // "UGCA"
console.log(transcribir("ACGTGGTCTTAA")) // "UGCACCAGAAUU"
```

16. Capitalizar palabra

Escribir una función llamada `capitalizar` que reciba un string y capitalice la primera letra:

```
// escribe tu función acá

// código de prueba
console.log(capitalizar("pedro")) // "Pedro"
console.log(capitalizar("hola mundo")) // "Hola mundo"
console.log(capitalizar("")) // ""
```

17. Capitalizar cada palabra

Escribir una función llamada `capitalizar` que reciba un string y capitalice la primera letra ****de cada palabra****:

```
// escribe tu función acá

// código de prueba
console.log(capitalizar("hola mundo")) // "Hola Mundo"
console.log(capitalizar("make it real")) // "Make It Real"
console.log(capitalizar("")) // ""
```

18. Encontrar el número máximo

Escribir una función llamada `max` que reciba un arreglo de números y retorne el número máximo:

****Nota:**** Intentarlo hacer sin el método `Math.max` de JavaScript.

```
// escribe tu función acá

// código de prueba
console.log(max([3, 9, 6])) // 9
console.log(max([67, 35, 54, 26])) // 67
console.log(max([5, 9, 2, 4, 5, 7])) // 9
```

19. Encontrar el número mínimo

Escribir una función llamada `min` que reciba un arreglo de números y retorne el número mínimo:

****Nota:**** Intentarlo hacer sin el método `Math.min` de JavaScript.

```
// escribe tu función acá

// código de prueba
console.log(min([3, 9, 6])) // 3
console.log(min([67, 35, 54, 26])) // 26
console.log(min([5, 9, 2, 4, 5, 7])) // 2
```

20. Generar contraseña

Escribir una función llamada `password` que reciba un string y retorne un nuevo string realizando los siguientes cambios:

* Las mayúsculas se reemplazan por minúsculas.

- * Se eliminan los espacios en blanco.
- * Reemplaza el caracter "a" por "4".
- * Reemplaza el caracter "e" por "3".
- * Reemplaza el caracter "i" por "1".
- * Reemplaza el carater "o" por "0".

```
// escribe tu función acá

// código de prueba
console.log(password("hola")) // "h0l4"
console.log(password("esta es una prueba")) // "3st43sun4pru3b4"
console.log(password("")) // ""
```

21. Encontrar números pares en un arreglo

Escribir una función llamada `pares` que reciba un arreglo de números y retorne un nuevo arreglo con los números pares únicamente.

```
// escribe tu función acá

// código de prueba
console.log(pares([1, 2, 3, 4, 5, 6])) // [2, 4, 6]
console.log(pares([])) // []
```

22. Encontrar posiciones de números pares

Escribir una función llamada `posiciones` que reciba un arreglo de números y retorne un nuevo arreglo ****con las posiciones**** donde se encuentran números pares.

```
// escribe tu función acá

// código de prueba
console.log(posiciones([1, 2, 3, 4, 5, 6])) // [1, 3, 5]
console.log(posiciones([])) // []
```

23. Duplicar elementos de un arreglo

Escribir una función llamada `duplicar` que reciba un arreglo de números y retorne un nuevo arreglo donde cada número esté multiplicado por dos (2).

```
// escribe tu función acá

// código de prueba
console.log(duplicar([1, 2, 3])) // [2, 4, 6]
console.log(duplicar([])) // []
```

24. Encontrar palabras que empiecen por "a"

Escribir una función llamada `empiezanConA` que reciba un arreglo de strings y retorne un nuevo arreglo con todas las palabras que empiecen por "a" (mayúscula o minúscula).

```
// escribe tu función acá

// código de prueba
console.log(empiezanConA(["beta", "alfa", "Arbol", "gama"])) // ["alfa", "Arbol"]
console.log(empiezanConA(["beta", "delta", "gama"])) // []
console.log(empiezanConA([])) // []
```

25. Encontrar palabras que terminan en "s"

Escribir una función llamada `terminanConS` que reciba un arreglo de strings y retorne un nuevo arreglo con todas las palabras que terminan con "s" (mayúscula o minúscula).

```
// escribe tu función acá

// código de prueba
console.log(terminanConS(["pruebas", "arroz", "árbol", "tokens"])) // ["pruebas", "tokens"]
console.log(terminanConS(["beta", "delta", "gama"])) // []
console.log(terminanConS([])) // []
```

26. Imprimir una matriz

Escribir una función llamada `imprimirMatriz` que reciba una matriz (un arreglo de arreglos) e imprima todos los elementos del arreglo.

```
// escribe tu función acá

// código de prueba
console.log(imprimirMatriz([
  [1, 2, 3],
  [4, 5, 6],
  [7, 8, 9]
]))

// 1
// 2
// 3
// 4
// 5
// 6
// 7
// 8
// 9
```

27. Traducir números a palabras

Escribir una función llamada `numerosAPalabras` que reciba un arreglo de números (cada número en el rango de 0 a 9) y retorne un nuevo arreglo convirtiendo cada número a su versión en palabras.


```
// escribe tu función acá

// código de prueba
console.log(numerosAPalabras([0, 1, 2, 3, 4])) // ["cero", "uno", "dos", "tres", "cuatro"]
console.log(numerosAPalabras([5, 6, 7, 8, 9])) // ["cinco", "seis", "siete", "ocho", "nueve"]
```

28. Traducir palabras a números

Escribir una función llamada `palabrasANumeros` que reciba un arreglo de strings y retorne un nuevo arreglo traduciendo cada palabra a su versión numérica (del 0 al 9). Si la palabra no es un número traducir a -1.

```
// escribe tu función acá

// código de prueba
console.log(["cero", "uno", "dos", "tres", "what?", "cuatro"]) // [0, 1, 2, 3, -1, 4]
console.log(["cinco", "seis", "siete", "ocho", "nueve"]) // [5, 6, 7, 8, 9]
```

29. Número de asteriscos en un arreglo

Escribir una función llamada `numAsteriscos` que reciba un arreglo y retorne el número de asteriscos:

```
// escribe tu función acá

// código de prueba
console.log(numAsteriscos(['', '*', '', '*'])) // 2
console.log(numAsteriscos(['*', '*', '*'])) // 3
console.log(numAsteriscos([])) // 0
```

30. Número de asteriscos en una matriz

Escribir una función llamada `numAsteriscos` que reciba una matriz (un arreglo de arreglos) y retorne el número de asteriscos:

```
// escribe tu función acá

// código de prueba
console.log(numAsteriscos([
  ['*', '', '*'],
  ['', '*', ''],
  ['*', '', '*']
]))
// 5
```

31. Distancia entre dos strings

Escribir una función llamada `distancia` que reciba dos strings y retorne el número de caracteres diferentes (comparando posición por posición).

****Nota:**** Puedes asumir que los strings siempre tienen la misma longitud. Sin embargo, si quieres agregarle más dificultad puedes pensar cómo solucionarlo si un string es más largo que el otro (la diferencia entre las longitudes agregaría al resultado).



PROGRAMACIÓN CON JAVASCRIPT

```
// escribe tu función acá

// código de prueba
console.log(distancia("hola", "hola")) // 0
console.log(distancia("sol", "tol")) // 1
console.log(distancia("carro", "correr")) // 3
```