

## Desarrollo Web

### Ejercicios Prácticos

Este grupo de ejercicios tendrá una nota más alta que otros ejercicios, es importante que los haga para practicar la parte de programación de javascript y sus funciones.

Por su propio bien profesional, hacer los siguientes ejercicios a conciencia para practicar la parte de programación con javascript y el uso del DOM.

Haga un menú que lleve a la resolución de cada uno de los ejercicios, puede hacerlo en una sola página utilizando saltos por ejemplo:

`<a href="#" onclick="location.hash='section-name';return false;">Ir a la sección</a>`

Elabore el DOM de cada ejercicio:

1. 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.
  - a. El BMI se calcula con la siguiente formula:  $\text{peso} / \text{altura}^2$
  - b. Escribir una función llamada bmi que reciba dos argumentos: peso y altura, y retorne un string con las siguientes posibilidades:
    - i. "Bajo de peso" si el BMI < 18.5
    - ii. "Normal" si está entre 18.5 y 24.9
    - iii. "Sobrepeso" si está entre 25 y 29.9
    - iv. "Obeso" si es igual o mayor a 30
2. 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.

3. Contraseña Válida: Hacer un formulario que solicite el usuario y contraseña y validar si es correcta o no, tener en cuenta algún método para estandarizar el ingreso y comparación de contraseña (por ejemplo, mayúsculas y minúsculas)
4. Calculo de 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  $\text{ingresos} * 40\%$ . De lo contrario retornar 0.
5. Imprimir un arreglo: hacer una función que reciba un arreglo e imprima un UL (unsorted list) de cada ítem, por ejemplo: `imprimirArreglo(1, "Hola", 2, "Mundo")`: la salida debería de ser:
  - a. Hola
  - b. 2
  - c. mundo

6. Numero 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.
  - a. Ejemplo: `numeroDeCaracteres("Hola Mundo", "o")` // retorna 2
7. Sumar Arreglo: Escribir una función llamada `sumarArreglo` que reciba un arreglo de números y retorne la suma de todos los elementos.
  - a. Ejemplo: `sumarArreglo([3, 1, 2])` // suma = 6
8. Multiplicar Arreglo: lo mismo que el inciso 7 sólo que con multiplicaciones
9. Sumar arreglo entre 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).
  - a. Ejemplo: `sumarArreglo([1, 2, 3, 4, 5, 6, 7, 8, 9, 10], 3, 6)` // suma del rango 3 al 6 =22
10. Encontrar el número máximo de un elemento de un arreglo, hacer este ejercicio sin usar `Math.max`, y luego hágalo usándolo.
  - a. Ejemplo: `max([3, 9, 6])` // Máximo = 9