

Absolutamente! Aquí tienes una serie de 10 ejercicios detallados, enfocados en la manipulación de **Datos Primitivos** en JavaScript (Strings, Números y Booleanos), con una progresión gradual de dificultad.

---

### Ejercicio 1: Longitud de un Nombre

#### Consigna:

Crea una función llamada `obtenerLongitudNombre` que reciba un argumento `nombre` (un string). La función debe retornar la cantidad de caracteres que tiene ese string.

#### Ejemplo:

- `obtenerLongitudNombre("Ana")` debería retornar `3` .
  - `obtenerLongitudNombre("Carlos")` debería retornar `6` .
- 

### Ejercicio 2: Unir Nombre y Apellido

#### Consigna:

Define una función llamada `generarNombreCompleto` que reciba dos argumentos: `nombre` (string) y `apellido` (string). La función debe retornar un nuevo string que sea la concatenación del nombre y el apellido, separados por un espacio.

#### Ejemplo:

- `generarNombreCompleto("Maria", "González")` debería retornar `"Maria González"` .
  - `generarNombreCompleto("Juan", "Perez")` debería retornar `"Juan Perez"` .
- 

### Ejercicio 3: Convertir a Mayúsculas

#### Consigna:

Escribe una función llamada `convertirAMayusculas` que tome un argumento `texto` (un string). La función debe retornar un nuevo string que sea la versión en mayúsculas del `texto` original.

#### Ejemplo:

- `convertirAMayusculas("hola mundo")` debería retornar `"HOLA MUNDO"` .
  - `convertirAMayusculas("JavaScript")` debería retornar `"JAVASCRIPT"` .
- 

### Ejercicio 4: Suma de Números Flotantes

#### Consigna:

Implementa una función llamada `sumarFlotantes` que acepte dos argumentos numéricos: `num1` y `num2` , los cuales pueden ser números decimales (flotantes). La función debe retornar la suma de estos dos números.

#### Ejemplo:

- `sumarFlotantes(3.5, 2.1)` debería retornar `5.6` .
  - `sumarFlotantes(10, 0.75)` debería retornar `10.75` .
-

## Ejercicio 5: Calcular el Resto de una División

### Consigna:

Define una función llamada `obtenerResto` que reciba dos argumentos numéricos enteros: `dividendo` y `divisor`. La función debe retornar el resto de la división del `dividendo` por el `divisor`.

### Ejemplo:

- `obtenerResto(10, 3)` debería retornar `1` (porque 10 dividido 3 es 3 con resto 1).
  - `obtenerResto(7, 2)` debería retornar `1`.
  - `obtenerResto(8, 4)` debería retornar `0`.
- 

## Ejercicio 6: Comparación de Edades

### Consigna:

Crea una función llamada `esMayorQue` que tome dos argumentos numéricos: `edad1` y `edad2`. La función debe retornar `true` si `edad1` es estrictamente mayor que `edad2`, y `false` en caso contrario.

### Ejemplo:

- `esMayorQue(25, 20)` debería retornar `true`.
  - `esMayorQue(18, 18)` debería retornar `false`.
  - `esMayorQue(15, 20)` debería retornar `false`.
- 

## Ejercicio 7: Negación Lógica

### Consigna:

Escribe una función llamada `negarBooleano` que reciba un argumento booleano: `valorBooleano`. La función debe retornar el valor booleano opuesto al que se le pasó (es decir, si recibe `true` debe retornar `false`, y viceversa).

### Ejemplo:

- `negarBooleano(true)` debería retornar `false`.
  - `negarBooleano(false)` debería retornar `true`.
- 

## Ejercicio 8: Comprobación de Credenciales Simples

### Consigna:

Implementa una función llamada `validarAcceso` que reciba dos argumentos string: `usuario` y `contrasena`. La función debe retornar `true` si `usuario` es exactamente "admin" Y `contrasena` es exactamente "secreto123". En cualquier otro caso, debe retornar `false`.

### Consideraciones:

Usa el operador lógico AND ( `&&` ).

---

## Ejercicio 9: Cadena Vacía o Solo Espacios

**Consigna:**

Define una función llamada `esCadenaVacía0Espacios` que tome un argumento `cadena` (string). La función debe retornar `true` si la `cadena` está vacía o si contiene solo caracteres de espacio en blanco. De lo contrario, debe retornar `false`.

**Ejemplo:**

- `esCadenaVacía0Espacios("")` debería retornar `true`.
- `esCadenaVacía0Espacios(" ")` debería retornar `true`.
- `esCadenaVacía0Espacios("hola")` debería retornar `false`.
- `esCadenaVacía0Espacios(" abc ")` debería retornar `false`.

**Consideraciones:**

Considera el método `.trim()` de los strings que elimina los espacios en blanco de ambos extremos de un string.

---

**Ejercicio 10: Formato de Moneda Simple****Consigna:**

Crea una función llamada `formatearPrecio` que reciba un argumento numérico: `precio`. La función debe retornar un string que represente el precio con el símbolo de moneda "\$" al principio y dos decimales.

**Ejemplo:**

- `formatearPrecio(15.5)` debería retornar `"$15.50"`.
- `formatearPrecio(7)` debería retornar `"$7.00"`.
- `formatearPrecio(99.999)` debería retornar `"$100.00"` (redondeado o truncado a dos decimales, se espera redondeo estándar).

**Consideraciones:**

Investiga cómo trabajar con la representación de números con decimales fijos en JavaScript (por ejemplo, el método `toFixed()`).