

AE4_ABP-Ejercicio individual [Actividad Evaluada]

AE4_ABP-Ejercicio individual [Actividad Evaluada]

Contexto

Imagina que estás organizando tus finanzas personales. Necesitas una herramienta sencilla que te ayude a calcular cuánto dinero puedes ahorrar cada mes con base en tus ingresos y tus gastos. Para lograrlo, vas a implementar una serie de funciones en JavaScript que realicen estos cálculos y te ayuden a mantener tu presupuesto bajo control. A lo largo del ejercicio, pondrás en práctica conceptos fundamentales sobre funciones.

Actividad

- 1. ¿Qué es una función y para qué sirve?**
Escribe una breve definición con tus propias palabras sobre qué es una función en programación y cuál es su propósito.
- 2. Definir funciones**
Define una función llamada **calcularGastosTotales** que reciba tres parámetros: **renta**, **comida**, y **transporte**. Esta función debe calcular la suma total de los tres gastos y **retornar el resultado**.
- 3. Paso de parámetros en una función**
Define una función llamada **calcularAhorroMensual** que reciba dos parámetros: **ingresosMensuales** y **gastosTotales**. Dentro de la función, resta los gastos a los ingresos y **devuelve el ahorro mensual**.
- 4. Retorno de una función**
Usa **console.log()** para mostrar el resultado de invocar **calcularAhorroMensual** con datos de ejemplo. El valor retornado debe ser visible en consola.
- 5. Variables locales y variables globales**
Declara una variable global llamada **moneda = "MXN"**. Luego, dentro de la función **mostrarResumenFinanciero**, define variables locales para los ingresos, gastos y ahorro. Imprime el resumen financiero en consola, incluyendo la moneda global.
- 6. Invocación de una función**
Invoca **calcularGastosTotales** con valores de ejemplo (por ejemplo: 5000, 2000, 1500). Usa ese resultado como argumento para **calcularAhorroMensual**.
- 7. Alcance de las variables locales**
Crea una función **verificarSaldo** que defina una variable local llamada **saldoSeguro** dentro de la función. Intenta imprimir **saldoSeguro** fuera de la función y observa qué error aparece. Explica por qué ocurre.
- 8. El problema de las variables globales**
Imagina que defines una variable global llamada **descuento = 100**. Luego, dentro de otra función llamada **calcularDescuento**, también defines **descuento = 50**. Ejecuta ambas

funciones e imprime el valor de **descuento** antes y después de cada una. Analiza qué ocurrió con la variable global y por qué es importante evitar este tipo de situaciones.

9. Crear una función anidada

Crea una función llamada **gestionarFinanzas** que contenga dentro otra función llamada **imprimirResumen**. **gestionarFinanzas** debe calcular los gastos totales y el ahorro, y luego invocar la función **imprimirResumen** para mostrar un mensaje como:
"Tus gastos fueron de X y tu ahorro fue de Y."

Entrega:

- Un archivo zip con tu código y las respuestas a las preguntas.
- Duración: 1 jornada de clases.
- Ejecución: Individual.

Estado de la entrega

Estado de la entrega	Todavía no se han realizado envíos
Estado de la calificación	Sin calificar
Última modificación	-

DESARROLLO AE4_ABP-EJERCICIO INDIVIDUAL

1. ¿Qué es una función y para qué sirve?

Las funciones son bloques de código que se ejecutan solo cuando las llamamos, lo que nos permite reutilizar el código sin tener que escribirlo una y otra vez. Las funciones son herramientas poderosas que nos permiten estructurar de manera más organizada nuestro código.

2. Definir funciones

```
function calcularGastosTotales(renta, comida, transporte) {  
    let gastosTotales = renta + comida + transporte;  
    return gastosTotales;  
}
```

3. Paso de parámetros en una función

```
function calcularAhorroMensual(ingresosMensuales, gastosTotales) {  
    let ahorroMensual = ingresosMensuales - gastosTotales;  
    return ahorroMensual;  
}
```

4. Retorno de una función

```
let gastos = calcularGastosTotales(5000, 2000, 1500);  
let ahorro = calcularAhorroMensual(10000, gastos);  
console.log("Ahorro mensual:", ahorro);
```

5. Variables locales y variables globales

```
let moneda = "MXN";  
function mostrarResumenFinanciero() {  
    let ingresos = 10000;  
    let gastos = calcularGastosTotales(5000, 2000, 1500);  
    let ahorro = calcularAhorroMensual(ingresos, gastos);  
  
    console.log("Resumen financiero:");  
    console.log("Ingresos:", ingresos, moneda);  
    console.log("Gastos:", gastos, moneda);  
    console.log("Ahorro:", ahorro, moneda);  
}  
  
mostrarResumenFinanciero();
```

6. Invocación de una función

```
let resultado = calcularGastosTotales(5000, 2000, 1500);  
calcularAhorroMensual(10000, resultado);
```

7. Alcance de las variables locales

```
function verificarSaldo() {  
    let saldoSeguro = 2000;  
    console.log("Saldo dentro de la función:", saldoSeguro);  
}
```

```
verificarSaldo();
```

```
console.log(saldoSeguro);
```

Observa qué error aparece. Explica por qué ocurre

Error: ReferenceError: saldoSeguro is not defined

Este error ocurre porque saldoSeguro es una variable local, creada dentro de una

8. El problema de las variables globales

```
let descuento = 100; // Variable global  
function calcularDescuento() {  
    let descuento = 50; // Variable local con el mismo nombre  
    console.log("Descuento dentro de la función:", descuento); // 50  
}
```

```
console.log("Descuento global antes:", descuento);
```

```
calcularDescuento();
```

```
console.log("Descuento global después:", descuento);
```

Analiza qué ocurrió con la variable global y por qué es importante evitar este tipo de situaciones:

Si bien la variable global descuento se llama igual, a que está dentro de la función es otra diferente, porque fue creada dentro de la función, siendo una variable local, y solo ahí opera, no cambia ni afecta a la que está fuera (global), pese a que tienen el mismo nombre.

9. Crear una función anidada

```
function gestionarFinanzas() {  
    let renta = 5000;  
    let comida = 2000;  
    let transporte = 1500;  
    let ingresos = 10000;  
  
    let gastos = calcularGastosTotales(renta, comida, transporte);  
    let ahorro = calcularAhorroMensual(ingresos, gastos);  
  
    function imprimirResumen() {  
        console.log(`Tus gastos fueron de ${gastos} ${moneda} y tu ahorro fue de  
        ${ahorro} ${moneda}.`);  
    }  
    imprimirResumen();  
}
```

```
gestionarFinanzas();
```