

Ejercicio 1: Contador con Closure

Objetivo: Crear una función que use un closure para contar cuántas veces ha sido llamada.

Instrucciones:

1. Crea una función llamada `createCounter` que, cuando se invoque, retorne otra función.
2. La función retornada debe incrementar un contador interno y devolver el valor actual del contador cada vez que se llame.

Ejemplo de uso:

javascript

```
const counter = createCounter();  
console.log(counter()); // 1  
console.log(counter()); // 2  
console.log(counter()); // 3
```

Ejercicio 2: Generador de saludos personalizados

Objetivo: Usar un closure para crear saludos personalizados.

Instrucciones:

1. Crea una función llamada `createGreeting` que reciba un parámetro saludo (por ejemplo: "Hola", "Buenos días") y retorne una función que reciba el nombre de una persona.
2. La función retornada debe combinar el saludo y el nombre en un solo mensaje.

Ejemplo de uso:

javascript

```
const greetInSpanish = createGreeting("Hola");  
console.log(greetInSpanish("Juan")); // "Hola, Juan"  
console.log(greetInSpanish("María")); // "Hola, María"
```

```
const greetInEnglish = createGreeting("Hello");  
console.log(greetInEnglish("John")); // "Hello, John"
```

Ejercicio 3: Creación de un banco simple

Objetivo: Simular un sistema bancario básico con depósitos y retiros utilizando closures.

Instrucciones:

1. Crea una función llamada `createBankAccount` que tenga un balance inicial y retorne un objeto con dos métodos: `deposit` y `withdraw`.
2. El método `deposit` debe incrementar el balance interno con la cantidad depositada.

3. El método `withdraw` debe reducir el balance si hay fondos suficientes, y devolver el balance actual en ambos casos.

Ejemplo de uso:

javascript

```
const myAccount = createBankAccount(100);  
myAccount.deposit(50); // Balance: 150  
myAccount.withdraw(30); // Balance: 120  
myAccount.withdraw(150); // No suficiente saldo, Balance: 120
```