

POO(PROGRAMACIÓN ORIENTADA A OBJETOS)

La Programación Orientada a Objetos (POO) es un paradigma de programación que se basa en el concepto de objetos y se utiliza para organizar y estructurar el código de manera más modular y reutilizable. JavaScript, un lenguaje de programación ampliamente utilizado, también permite implementar la POO.

En este documento, exploraremos los conceptos básicos de la POO en JavaScript, incluyendo clase, objeto, instancia, atributo y métodos. A continuación, se presentará un taller de seis ejercicios prácticos para reforzar estos conceptos.

Clase

Una clase es una plantilla o un modelo para crear objetos. Define las propiedades y los comportamientos que los objetos de esa clase tendrán. En JavaScript, las clases se definen utilizando la palabra clave `class`.

```
class Persona {  
  // Cuerpo de la clase  
}
```

En el ejemplo anterior, se define una clase llamada "Persona".

Objeto

Un objeto es una instancia de una clase. Representa una entidad del mundo real con características específicas. Los objetos tienen atributos y pueden realizar acciones mediante sus métodos.

```
const miObjeto = new Persona();
```

En el ejemplo anterior, se crea un objeto llamado "miObjeto" a partir de la clase "Persona" utilizando la palabra clave `new`.

Instancia

Una instancia es un objeto específico creado a partir de una clase. Cada objeto creado a partir de una clase se considera una instancia independiente.

```
const instancia1 = new Persona();  
const instancia2 = new Persona();
```

En el ejemplo anterior, se crean dos instancias (instancia1 e instancia2) de la clase "Persona".

Atributo

Un atributo es una propiedad de un objeto. Define las características o los datos asociados a un objeto. Cada objeto puede tener diferentes valores para sus atributos.

```
class Persona {  
  
  constructor(nombre, edad) {
```

```

    this.nombre = nombre;
    this.edad = edad;
  }

  saludar() {
    console.log(`¡Hola! Mi nombre es ${this.nombre} y tengo ${this.edad} años.`);
  }
}

```

En el ejemplo anterior, la clase "Persona" tiene dos atributos: nombre y edad. Estos atributos se inicializan en el constructor de la clase.

Métodos

Un método es una función asociada a un objeto. Define las acciones o los comportamientos que un objeto puede realizar. Los métodos pueden acceder y manipular los atributos de un objeto.

En el ejemplo anterior, la clase "Persona" tiene un método llamado saludar(), que muestra un saludo en la consola utilizando los atributos nombre y edad del objeto.

Instanciación de una clase con atributos y acceso a métodos

En este caso debemos pasar nombre y edad que son los parámetros del constructor. Luego accederemos a los atributos y métodos.

```

const persona = new Persona("ADSO", 2);
//acceso a atributos
console.log(persona.nombre); //imprimirá ADSO
console.log(persona.edad); //imprimirá 2
console.log(persona.saludar()); //imprimirá ¡Hola! Mi nombre es ADSO y tengo 2 años.

```

Taller de Ejercicios

1. Crea una clase llamada "Perro" con un atributo llamado "nombre". Agrega un método llamado ladrar() que imprima en la consola "¡Woof!" junto al nombre del perro.
2. Crea una instancia de la clase "Perro" llamada "miPerro" con un nombre de tu elección. Llama al método ladrar() de la instancia.
3. Crea una clase llamada "CuentaBancaria" con atributos "saldo" (inicializado en 0) y "titular". Agrega un método llamado depositar(cantidad) que sume la cantidad proporcionada al saldo actual.
4. Crea una instancia de la clase "CuentaBancaria" llamada "miCuenta" con un titular de tu elección. Deposita 100 en la cuenta llamando al método depositar().

5. Agrega un método llamado retirar(cantidad) a la clase "CuentaBancaria" que reste la cantidad proporcionada del saldo actual. Asegúrate de verificar si hay suficiente saldo antes de realizar la operación.
6. Llama al método retirar() en la instancia "miCuenta" para retirar 50. Imprime en la consola el saldo actual después de la operación.

Contacto: gangaritah@gmail.com