



Programación Orientada a Objetos (POO) en Python

La **Programación Orientada a Objetos (POO)** es un paradigma fundamental en Python que ayuda a estructurar programas de manera modular y reutilizable.

La POO se basa en el concepto de **objetos**, que son instancias de **clases**. Una clase es como un plano o plantilla para crear objetos.

Conceptos Clave

1. **Clase (Class)**: Una plantilla para crear objetos. Define un conjunto de **atributos** (datos/variables) y **métodos** (funciones) que los objetos creados a partir de la clase poseerán.
 - *Ejemplo:* `class Coche:`
2. **Objeto (Object)**: Una instancia específica de una clase.
 - *Ejemplo:* `mi_coche = Coche()`
3. **Atributos (Attributes)**: Las variables que pertenecen a una clase o a un objeto. Representan el estado o las propiedades de un objeto.
 - *Ejemplo:* `color, velocidad`
4. **Métodos (Methods)**: Las funciones definidas dentro de una clase que realizan acciones o definen el comportamiento de un objeto.
 - *Ejemplo:* `acelerar(), frenar()`
5. **Herencia (Inheritance)**: Permite que una nueva clase (subclase/clase hija) herede atributos y métodos de una clase existente (superclase/clase padre). Esto fomenta la **reutilización de código**.
6. **Encapsulamiento (Encapsulation)**: Agrupar los datos (atributos) y los métodos que operan sobre esos datos en una sola unidad (la clase), y controlar el acceso a ellos.
7. **Polimorfismo (Polymorphism)**: La capacidad de diferentes objetos para responder a un mismo mensaje (nombre de método) de diferentes maneras.