

# ¿Qué es la herencia en Python?

Dentro de la POO en Python, la herencia es un método por el cual permite que una subclase pueda reutilizar atributos y métodos de una clase padre o superclase.

Esta funcionalidad evita repetir código y mantener la estructura estable.

---

En el código de la siguiente página se aplica el principio de **herencia** en Python.

La clase **padre** *Vehicle* define los atributos comunes a todos los vehículos (brand, model) y un método info() que devuelve esa información.

Luego, la clase **hija** car hereda de *Vehicle*. Gracias a **super()**, reutiliza los atributos de la clase padre y, además, incorpora un atributo propio (doors). También sobrescribe el método **info()** para mostrar la información completa (marca, modelo y número de puertas).

Finalmente, se crean instancias de la clase car (objetos como vehicle1, vehicle2, vehicle3), cada una con sus propios valores de marca, modelo y número de puertas. Al llamar a **info()**, cada objeto devuelve sus datos de forma personalizada.

---

Justo debajo de la imagen se puede ver otra imagen, ésta representa el output tras la ejecución del código.

```
#!/usr/bin/env python3

# Main class
class Vehicle:
    def __init__(self, brand, model):
        self.brand = brand
        self.model = model

    def info(self):
        return f"Vehicle: {self.brand} - {self.model}"

# Subclass
class car(Vehicle):
    def __init__(self, brand, model, doors):
        super().__init__(brand, model)
        self.doors = doors

    def info(self):
        return f"Car: {self.brand} - {self.model}, Doors: {self.doors}"

# Objects
vehicle1 = car("Toyota", "Corolla", 4)
vehicle2 = car("Audi", "A4", 4)
vehicle3 = car("Honda", "Civic", 2)

print(vehicle1.info())
print(vehicle2.info())
print(vehicle3.info())
```

```
miguel@DESKTOP-93SQ670:/mnt/c/Users/miguel/Desktop$ python3 file.py
Car: Toyota - Corolla, Doors: 4
Car: Audi - A4, Doors: 4
Car: Honda - Civic, Doors: 2
```