

¿Qué es el polimorfismo en Python?

Dentro de la POO en Python, el polimorfismo significa que una misma función puede comportarse de forma diferente según el objeto que reciba dicha función.

En el mundo real el polimorfismo se usa para hacer el código más flexible y reutilizable sobre todo a largo plazo ya que no se necesita diferentes funciones para cada tipo de objeto si todos hacen la misma función, además de que, en caso de extender el programa con más clases, solo se necesitan añadir los métodos o funciones de dichas clases, mientras que el resto del código seguirá funcionando sin tener que cambiar nada.

En el código de la siguiente página se puede ver como se aplica el polimorfismo, cada objeto recibe la misma función, pero dependiendo del objeto que recibe la función, éste la usa de forma distinta.

Explicándolo mejor sería así, se definen 3 clases diferentes (**Dog**, **Cat**, **Human**), cada una de ellas tiene un método llamado 'speak()'.

Después hay una lista de objetos llamada **things**, dentro tiene las 3 clases anteriores, esta lista se crea para después enlazarla con un bucle **for** que lo que hace es recorrer cada objeto de la lista para juntarlo con la acción que cada objeto realiza.

De ahí de que cada objeto recibe la misma función, pero la usa de forma distinta.

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

# Main class
class Dog:
    def speak(self):
        return "The dog says: Woof!"

class Cat:
    def speak(self):
        return "The cat says: Meow!"

class Human:
    def speak(self):
        return "The human says: Hello!"

# Lets use the polymorphism just here
things = [Dog(), Cat(), Human()]

for item in things:
    print(item.speak())
```

```
miguel@DESKTOP-93SQ670:/mnt/c/Users/miguel/Desktop$ python3 file.py
The dog says: Woof!
The cat says: Meow!
The human says: Hello!
```