

# Python Programming

## Módulo 3

# Polimorfismo

# Polimorfismo en Python

El **polimorfismo** nos posibilita invocar un determinado método de un objeto, y que podrá obtenerse distintos resultados según sea el objeto. Esto se debe a que distintos objetos (que pueden o no ser creados con la misma clase) pueden tener un método con un mismo nombre, pero que realicen distintas operaciones.

Un ejemplo simple:

```
class gato():
    def hablar(self):
        print("MIAU")

class perro():
    def hablar(self):
        print("GUAU")

def escucharMascota(animal):
    animal.hablar()
```

Un ejemplo con el concepto de herencia:

```
class Animal:
    def hablar(self):
        pass

class Perro(Animal):
    def hablar(self):
        print("Guau!")

class Gato(Animal):
    def hablar(self):
        print("Miau!")
```

En este caso cuando, se cree un objeto *"Perro"*, se sustituirá el método *hablar* (de la clase padre *"Animal"*) por el de la clase hija (*"Perro"*). Lo mismo ocurre con *"Gato"*.

El **polimorfismo** es una cuestión de organización y de buena práctica para el programador que trabaja con muchos objetos y métodos. Permitiendo que el desarrollador no deba escribir, pensar y recordar muchos nombres de métodos diferentes, sino que pueda llamar al método del objeto adecuado con el mismo nombre que llamaría a otros.

# ¡Muchas gracias!

¡Sigamos trabajando!