



Universidad Autónoma de Baja California  
Campus Tijuana  
Facultad de contaduría y administración



Alumno:

García Bretado David Antonio

Grupo:

732

Try-Except

Programación en Python

Maestro:

Ray Brunet Parra Galaviz

Tijuana B.C. a 5 de septiembre del 2018

## Programación orientada a objetos Python

### Elementos y Características de la POO

Los elementos de la POO, pueden entenderse como los *materiales* que necesitamos para diseñar y programar un sistema, mientras que las características, podrían asumirse como las *herramientas* de las cuáles disponemos para construir el sistema con esos materiales.

Entre los elementos principales de la POO, podremos encontrar a:

### Clases

Las clases son los modelos sobre los cuáles se construirán nuestros objetos. Podemos tomar como ejemplo de clases, el gráfico que hicimos en la página 8 de este documento.

En Python, una clase se define con la instrucción `class` seguida de un nombre genérico para el objeto.

```
class Objeto:  
    pass
```

```
class Antena:  
    pass
```

```
class Pelo:  
    pass
```

```
class Ojo:  
    pass
```

### Propiedades

Las propiedades, como hemos visto antes, son las características intrínsecas del objeto. Éstas, se representan a modo de variables, solo que técnicamente, pasan a denominarse *propiedades*:

```
class Antena():  
    color = ""  
    longitud = ""
```

```
class Pelo():
    color = ""
    textura = ""
```

```
class Ojo():
    forma = ""
    color = ""
```

## Métodos

Los métodos son *funciones* (como las que vimos en el capítulo anterior), solo que técnicamente se denominan métodos, y representan acciones propias que puede realizar el objeto (y no otro):

```
class Objeto():
    color = "verde"
    tamaño = "grande"
    aspecto = "feo"
    antenas = Antena()
    ojos = Ojo()
    pelos = Pelo()

    def flotar(self):
        pass
```

## Objeto

Las clases por sí mismas, no son más que modelos que nos servirán para crear objetos en concreto. Podemos decir que una clase, es el razonamiento abstracto de un objeto, mientras que el objeto, es su materialización. A la acción de crear objetos, se la denomina *instanciar una clase* y dicha instancia, consiste en asignar la clase, como valor a una variable:

```
class Objeto():
    color = "verde"
    tamaño = "grande"
    aspecto = "feo"
    antenas = Antena()
    ojos = Ojo()
    pelos = Pelo()
```

```
def flotar(self):  
    print 12
```

```
et = Objeto()  
print et.color  
print et.tamano  
print et.aspecto  
et.color = "rosa"  
print et.color
```

### Accediendo a los métodos y propiedades de un objeto

Una vez creado un objeto, es decir, una vez hecha la instancia de clase, es posible acceder a su métodos y propiedades. Para ello, Python utiliza una sintaxis muy simple: el nombre del objeto, seguido de punto y la propiedad o método al cuál se desea acceder:

```
objeto = MiClase()  
print objeto.propiedad  
objeto.otra_propiedad = "Nuevo valor"  
variable = objeto.metodo()  
print variable  
print objeto.otro_metodo()
```

Fuentes:

[https://librosweb.es/libro/python/capitulo\\_5/programacion\\_orientada\\_a\\_objetos.html](https://librosweb.es/libro/python/capitulo_5/programacion_orientada_a_objetos.html)