

Clases y Objetos

Definiciones necesarias:

¿Qué es un Objeto?

Los objetos son la clave para entender la Programación Orientada a Objetos. Si miramos a nuestro alrededor encontraremos un sin fin de objetos de la vida real: perro, escritorio, televisor, bicicleta, etc...

¿Qué es un Atributo?

Los atributos o propiedades de los objetos son las características que puede tener un objeto: Si el objeto fuera Perro, los atributos podrían ser: tamaño, edad, color, raza, etc...

¿Qué es un Método?

Los métodos son la acción o función que realiza un objeto. Si nuestro objeto es Perro, los métodos pueden ser: caminar, ladrar, saltar, dormir, etc...

¿Qué es una Clase?

Con todos los conceptos anteriores explicados, se puede decir que una clase es una plantilla genérica de un objeto. La clase proporciona variables iniciales de estado (donde se guardan los atributos) e implementaciones de comportamiento (métodos)

Clases y Objetos

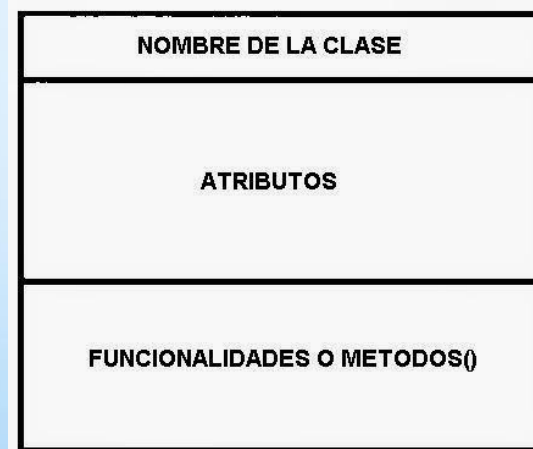
Definiciones necesarias:

¿Qué es una Instancia?

Ya sabemos que una clase es una estructura general del objeto. Por ejemplo, podemos decir que la clase Mascota necesita tener un nombre y una especie, pero no nos va a decir cual es el nombre y cual es la especie, es aquí donde entran las instancias. Una instancia es una copia específica de la clase con todo su contenido.

Ejemplo: Mateo = Mascota ("Mateo", "Gato")

Aquí podemos decir que Mateo es una instancia de la clase Mascota.



Las clases nos dan la posibilidad de crear estructuras de datos más complejas. En nuestro ejemplo crearemos una clase Mascotas que realizará un seguimiento del nombre y la especie (que pasaremos como atributos).

Clases y Objetos

Definiendo una Clase en Python

¿Qué es una Instancia?

Ya sabemos que una clase es una estructura general del objeto. Por ejemplo, podemos decir que la clase Mascota necesita tener un nombre y una especie, pero no nos va a decir cual es el nombre y cual es la especie, es aquí donde entran las instancias. Una instancia es una copia específica de la clase con todo su contenido.

Ejemplo: `Mateo = Mascota("Mateo", "Gato")`

Aquí podemos decir que Mateo es una instancia de la clase Mascota.

```
class Mascota():
    def __init__(self, nombre, especie):
        self.nombre = nombre
        self.especie = especie
    def darNombre(self):
        return self.nombre
    def darEspecie(self):
        return self.especie
    def __str__(self):
        return "%s es un %s" % (self.nombre, self.especie)
```

Guardamos en el modulo Mascotas.py

Clases y Objetos

Probando la clase en Interactivo

```
>>> from Mascotas import Mascota
>>> rocco=Mascota('Rocco','Perro')
>>> tito=Mascota('Tito','Gato')
>>> pepe=Mascota('Pepe','Pato')
>>> pancho=Mascota('Pancho','Loro')
>>> print pepe
Pepe es un Pato
>>> print '%s es mi %s' % (pepe.darNombre(),pepe.darEspecie())
Pepe es mi Pato
>>> print '%s es mi %s' % (rocco.darNombre(),rocco.darEspecie())
Rocco es mi Perro
>>> print pancho
Pancho es un Loro
>>> print tito
Tito es un Gato
```

Clases y Objetos

Definiendo una SubClase en Python

Las sus clases heredan todo lo que pertenece a una clase y permite definir nuevos atributos y métodos específicos para la clase.

Ej: A veces no nos alcanza definiendo solo una clase como Mascota. Por ejemplo, algunas mascotas pueden ser perros y a la mayoría de ellos les gusta perseguir gatos o pueden ser gatos y la mayoría de los gatos odia a los perros. Entonces definimos:

```
#Subclases
class Perro(Mascota):
    def __init__(self,nombre,persigue_gatos):
        Mascota.__init__(self,nombre,"Perro")
        self.odia_perros=persigue_gatos
    def persigue_gatos(self):
        return self.persigue_gatos

class Gato(Mascota):
    def __init__(self,nombre,odia_perros):
        Mascota.__init__(self,nombre,"Gato")
        self.odia_perros=odia_perros
    def odiaPerros(self):
        return self.odia_perros
```

Clases y Objetos

Probando la Subclase en Modulo

```
from Mascotas import *
pluto=Perro('Pluto',True)
snoopy=Perro('Snoopy',False)
tom=Gato('Tom',True)
felix=Gato('Felix',False)
l=[pluto,snoopy,tom,felix]
for i in l:
    print i
    if i.darEspecie()=='Gato':
        if i.odia_perros:
            print ' Odia a los Perros'
        else:
            print ' No odia a los Perros'
    if i.darEspecie()=='Perro':
        if i.persigue_gatos:
            print ' Persigue Gatos'
        else:
            print 'NO persigue Gatos'
```

Clases y Objetos

Probando la Subclase en Modulo

```
C:\Python27\python.exe "C://ARTEMAR/UAA/JIT-CITA 2017/EJERCICIOS/mascotasVer.py"
```

```
Pluto es un Perro
```

```
    Persigue Gatos
```

```
Snoopy es un Perro
```

```
    Persigue Gatos
```

```
Tom es un Gato
```

```
    Odia a los Perros
```

```
Felix es un Gato
```

```
    No odia a los Perros
```

Elaborar un programa que:

Lea el archivo alumnos.csv

Despliegue el contenido

Solicite se ingrese un numero de Cedula (CI)

Busque el archivo fichaCI.csv

Si encuentra

desplegar su contenido

Si no encuentra

Buscar datos en el archivo alumnos.csv

Si encuentra:

Solicitar edad y ciudad

Grabar en fichaCI.csv

Si no encuentra

Informar

Desplegar datos de alumnos.csv

Solicitar cedula (99 para salir)

Indicaciones especiales:

Crear una clase alumnos

Dentro de la clase definir:

La inicialización con el nombre del archivo a leer

Definir la función **listar**(donde se despliega el contenido del archivo alumnos)

Definir función **buscar** (donde se realiza la búsqueda del alumno por cedula)