

Python Classes

Programación Web 2

Basado en parcialmente en:

<https://docs.python.org/3/tutorial/classes.html>

Comparación con Java

- Las ideas de clases y objetos de Java son similares en Python
- En Python todo es un objeto, incluso las clases
- Las clases tienen métodos y atributos, pero sólo pueden ser públicos
- Las clases son objetos y se pueden modificar en tiempo de ejecución
- La herencia funciona casi igual, pero no hay interfaces
- Pueden haber atributos de clase y de objeto
- Los atributos de una clase se crean dinámicamente
- Los métodos **deben** tener un primer parámetro que funcionará como this
- También se cuenta con garbage collector
- Seguiremos la misma convención de nombres de Java

Declaración e instanciación de Clases

Los atributos que se declaran fuera de un método, serán atributos de clase

```
class ClassName:  
    <statement-1>  
    .  
    .  
    .  
    <statement-N>
```

self funciona como this de Java, pero si se desea puede usarse otro nombre

```
>>> class Complex:  
...     def __init__(self, realpart, imagpart):  
...         self.r = realpart  
...         self.i = imagpart  
...  
>>> x = Complex(3.0, -4.5)  
>>> x.r, x.i  
(3.0, -4.5)
```

Variables de clase y de instancia

¿Qué ocurre si cambiamos el valor de una variable de clase?

```
class Dog:

    barking = 'wof wof'          # class variable shared by all instances
    def __init__(self, name):
        self.name = name        # instance variable unique to each instance

>>> d = Dog('Fido')
>>> e = Dog('Buddy')
>>> d.barking                # shared by all dogs
'wof wof'
>>> e.barking                # shared by all dogs
'wof wof'
>>> d.name                   # unique to d
'Fido'
>>> e.name                   # unique to e
'Buddy'
```

Herencia

Toda clases debe derivar de alguna otra

Si no se especifica, la clase derivará de la clases Object

Python soporta la herencia múltiple pero su uso no es recomendable desde el punto de vista de diseño

```
class
DerivedClassName(BaseClassName)
:
    <statement-1>
    .
    .
    .
    <statement-N>
```

WHY USE OOP AND CLASSES OF OBJECTS?

- mimic real life
- group different objects part of the same type

