

Python Katas

Alexander Acosta Jiménez
Ana María Bedoya Hernández

February 20, 2016

Tabla de contenidos

1. Introducción a Python
2. Tipos básicos
3. Colecciones
4. Control de flujo
5. Funciones
6. Clases
7. Herencia
8. Módulos

Introducción a Python

- ▶ Qué es Python?
- ▶ Características
 - ▶ Lenguaje interpretado
 - ▶ Tipado dinámico
 - ▶ Fuertemente tipado
 - ▶ Multiplataforma
 - ▶ Orientado a objetos



Tipos básicos

Al igual que en cualquier lenguaje de programación Python tiene tipos básicos como:

- ▶ Numérico
 - ▶ Entero
 - ▶ Flotante
- ▶ Cadenas
- ▶ Booleanos

Por inferencia de tipos en Python se pueden definir variables de la siguiente manera:

```
entero = 1
flotante = 1.2
cadena = "cadena"
booleano = True
```

Colecciones

En Python existen tres tipos de colecciones:

1. Listas
2. Tuplas
3. Diccionarios

Colecciones

1. Listas

La lista es un tipo de colección ordenada, es equivalente a lo que en otros lenguajes se conoce como arrays o vectores. Pueden contener cualquier tipo de dato: números, cadenas, booleanos, ... y también listas.

► Ejemplo:

```
l = [22, True, "una lista", [1, 2]]
```

Figure: Creación de una lista en Python.

Colecciones

2. Tuplas

Una tupla es una lista inmutable, es decir, no puede modificarse de ningún modo después de su creación. Se define igual que una lista, salvo que el conjunto se encierra entre paréntesis en lugar de entre corchetes. También tienen un orden definido.

► Ejemplo:

```
t = (1, 2, True, "python")
```

Figure: Creación de una tupla en Python.

Colecciones

3. Diccionarios

Los diccionarios deben su nombre a que son colecciones que relacionan una clave y un valor. El primer valor se trata de la clave y el segundo del valor asociado a la clave. La diferencia principal entre los diccionarios y las listas o las tuplas es que se accede a su valor por medio de su clave usando corchetes.

► Ejemplo:

```
diccionario = {"Manzana": "Roja",  
               "Limon": "Verde",  
               "Mango": "Amarillo"}  
print diccionario["Manzana"]
```

Figure: Creación de un diccionario en Python.

Control de flujo

Sentencias condicionales

► Sentencia if

```
fav = "mundogeek.net"  
# si (if) fav es igual a "mundogeek.net"  
if fav == "mundogeek.net":  
    print "Tienes buen gusto!"  
    print "Gracias"
```

Figure: Sintaxis de if en Python.

Control de flujo

Sentencias condicionales

► Sentencia if...else

```
if fav == "mundogeek.net":  
    print "Tienes buen gusto!"  
    print "Gracias"  
else:  
    print "Vaya, que lástima"
```

Figure: Sintaxis de if...else en Python.

Control de flujo

Sentencias condicionales

► Sentencia if...elif

```
if numero < 0:  
    print "Negativo"  
elif numero > 0:  
    print "Positivo"  
else:  
    print "Cero"
```

Figure: Sintaxis de if...elif en Python.

Control de flujo

Bucles

► Ciclo while

El bucle while(Mientras) ejecuta un fragmento de código mientras se cumpla una condición.

```
edad = 0
while edad < 18:
    edad = edad + 1
    print "Felicidades, tienes " + str(edad)
```

Figure: Sintaxis de while en Python.

Control de flujo

Bucles

► Ciclo for...in

El ciclo for se utiliza como una forma genérica de iterar sobre una secuencia.

```
secuencia = ["uno", "dos", "tres"]  
for elemento in secuencia:  
    print elemento
```

Figure: Sintaxis de for...in en Python.

Funciones

Una función es un fragmento de código con un nombre asociado que realiza una serie de tareas y devuelve un valor.

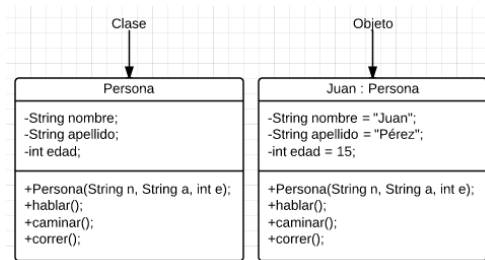
► Ejemplo:

```
def mi_funcion(param1, param2):  
    print param1  
    print param2
```

Figure: Esta función imprime los dos parámetros ingresados.

Clases y objetos

Una clase no es más que una plantilla genérica a partir de la cual se pueden instanciar objetos; en esta plantilla es donde se definen qué atributos y métodos tendrán los objetos de esa clase.



Clases y objetos

► Ejemplo:

```
class Coche:
    """Abstraccion de los objetos coche."""
    def __init__(self, gasolina):
        self.gasolina = gasolina
        print "Tenemos", gasolina, "litros"

    def arrancar(self):
        if self.gasolina > 0:
            print "Arranca"
        else:
            print "No arranca"

    def conducir(self):
        if self.gasolina > 0:
            self.gasolina -= 1
            print "Quedan", self.gasolina, "litros"
        else:
            print "No se mueve"
```


Herencia

La herencia sirve para crear clases nuevas a partir de clases preexistentes. Se heredan atributos y métodos de las clases viejas para modelar una nueva situación.

Herencia

► Ejemplo:

```
class Figura(object):
    def area(self):
        pass

from math import pi

class Circulo(Figura):
    def __init__(self, radio=0):
        self.radio = radio

    def area(self):
        return pi * self.radio * self.radio

class Triangulo(Figura):
    def __init__(self, base=0, altura=0):
        self.base = base
        self.altura = altura

    def area(self):
        return self.base * self.altura / 2.
```

Módulos y paquetes

Módulos

Para facilitar el mantenimiento y la lectura, los programas demasiado largos pueden dividirse en módulos, agrupando elementos relacionados.

```
def mi_funcion():  
    print "una funcion"  
  
class MiClase:  
    def __init__(self):  
        print "una clase"  
  
print "un modulo"
```

Figure: Ejemplo de un módulo en Python.

Módulos y paquetes

Módulos

Si se quisiera utilizar la funcionalidad definida en el módulo anterior en nuestro programa se necesita importarlo.

```
import modulo
```

Figure: Sintaxis de import en Python.

El import no solo hace que tengamos disponible todo lo definido dentro del módulo, sino que también ejecuta el código dentro de él.

Módulos y paquetes

Paquetes

Si los módulos sirven para organizar el código, los paquetes sirven para organizar los módulos. Los paquetes son tipos especiales de módulos (ambos son de tipo `module`) que permiten agrupar módulos relacionados. Se representan mediante directorios.

```
import paq.subpaq.modulo

paq.subpaq.modulo.func()
```

Figure: Ejemplo de paquetes en Python.

Bibliografía

- ▶ Libro: Python para Todos. Autor: Raúl González Duque.