



Programación Orientada a Objetos

Python (revisión y extensión)

Mg. Ing. César Aranda

cesar.aranda@ingenieria.uncuyo.edu.ar
unidados@gmail.com

Ingeniería en Mecatrónica

Objetivos y contenidos

- Rever las características sintácticas y semánticas del lenguaje Python
- Analizar el comportamiento en códigos sencillos
- Reconocer objetos estándares presentes en Python
- Mencionar diferentes módulos y su utilidad para resolver ciertos problemas propuestos en la práctica.

Referencias

- Von Rossum, G. (2017): El tutorial de Python 3. Editorial Python Software Foundation. Disponible en URL <http://tutorial.python.org.ar/>, accedido en 2017.
- <http://pyspanishdoc.sourceforge.net/lib/lib.html>
- Chazalet, S. (2017): Python3, los fundamentos del lenguaje. 2da edición. Ediciones Eni.
- Pérez castaño, A. (2016): Python fácil. Editorial Marcombo
- Beazley, D. y Jones, B. (2013). Python Cookbook. 3ra edición. O'Reilly Media. California.
- Hinojoza Gutierrez, A.P. (2016): Python, paso a paso. Editorial Ra-Ma. Madrid
- González Duque, R. (2008). Python para todos. UPR <http://mundogeek.net/tutorial-python/>, accedido en 2015.
- Hinojoza Gutierrez, A.P. (2016): Taller de Python. En URL https://www.psicobyte.com/descargas/taller_python.pdf , accedido en 2016

Ing. César Aranda

3

Palabras reservadas de Python

and	as	assert	break
class	continue	def	del
elif	else	except	exec
finally	for	from	global
if	import	in	is
lambda	not	or	pass
print	raise	return	try
while	yield	False	None
True			

Ing. César Aranda

4

Operadores

3 Matemáticos:

```
4
5 + (suma)
6 - (resta)
7 * (multiplicación)
8 / (división)
9 ** (exponente)
10 // (división entera)
11 % (módulo)
```

3 de Cadena

```
4
5 + (concatenación)
6 * (multiplicación)
```

3 Lógicos:

```
4
5 and (y lógica)
6 or (o lógica)
7 not (negación)
```

3 Identidad:

```
4
5 is
6 is not
```

9 Pertenencia:

```
10
11 in
12 not in
```

¿Se puede hacer?

```
a < b == c
```

Ing. César Aranda

3 Comparación:

```
4
5 == (igualdad)
6 != (desigualdad)
7 <> (desigualdad)
8 > (mayor que)
9 < (menor que)
10 >= (mayor o igual que)
11 <= (menor o igual que)
```

```
A and not B or C
```

```
(A and (not B)) or C
```

¿Producen el mismo resultado?

5

Tipos de datos, Variables y Comentarios

3 Numéricos:

```
4
5 entero = 4
6 int()
7
8 entero_largo = 145L
9 long()
10
11 flotante = 1.797
12 float()
13
14 complejo = 34 + 5j
15 complex()
16
17 logico = True, False
18 bool()
```

1 Cadenas:

```
2
3
4
5 cadena = "Monty Python Flying Circus"
6
7 cadena = 'Monty Python Flying Circus'
8
9 cadena = """Monty
10 Python
11 Flying
12 Circus"""
13
14 str()
15
16 El caracter de escape es "\"
```

```
2
3 None
4
```

#

Ing. César Aranda

¿Las variables son variables?

6

Listas y Tuplas

```
3 lista = ["John Cleese", "Terry Jones", "Terry Gilliam", \
4 "Eric Idle", "Michael Palin", "Graham Chapman"]
5
6 print lista[0]
7
8 list()
9
10 metalista= [{"arenque", "loro muerto", \
11 "vaca catapultada"}, {"Ni", "Jehová"}]
12
13 print metalista[1][0]
```

```
3 tupla = "La vida de Brian", "El sentido de la vida", \
4 "Los caballeros de la mesa cuadrada"
5
6 la_misma_tupla = ("La vida de Brian", "El sentido de la vida", \
7 "Los caballeros de la mesa cuadrada")
8
9 print tupla[2]
10
11 tupla_de_un_elemento = "elemento",
12
13 tuple()
```

Ing. César Aranda

¿Cuál es mutable?

7

Complemento pitónico

```
cuadrados = []
for x in range(10):
    cuadrados.append(x**2)
```



```
cuadrados = [x ** 2 for x in range(10)]
```

```
combs = []
for x in [1,2,3]:
    for y in [3,1,4]:
        if x != y:
            combs.append((x, y))
```



```
[(x, y) for x in [1,2,3] for y in [3,1,4] if x != y]
```

```
frutafresca = [' banana', ' mora de Logan ', 'maracuya ']
[arma.strip() for arma in frutafresca]
```

Ing. César Aranda

8

Diccionarios y Conjuntos

```
3 diccionario = {"título": "La vida de Brian", \
4 "Año": 1979, "Director": "Terry Jones"}
5
6 print diccionario["título"]
7
8 dict()
```

{ }

```
3 set = {"spam", "SPAM", "Spam", "spaaaaam", "spaMM"}
4
5 set()
```

¿Diccionario
o Conjunto?

```
canasta = {'manzana', 'naranja', 'manzana', 'pera', 'naranja', 'banana'}
```

```
{'pera', 'manzana', 'banana', 'naranja'}
>>> 'naranja' in canasta
True
>>> 'yerba' in canasta
False
```

```
a = set('abracadabra')
b = set('alacazam')
```

```
{'a', 'r', 'b', 'c', 'd'}
{'a', 'c', 'z', 'm', 'l'}
```

a & b

{'a', 'c'}

Ing. César Aranda

9

Control de Flujo 1

```
1#!/usr/bin/python
2# -*- coding: utf-8 -*-
3"""bruja.py"""
4
5peso = 50
6
7peso_de_un_ganso = 5
8
9if peso < peso_de_un_ganso:
10    print "¡Es una bruja!"
11
12elif peso > peso_de_un_ganso:
13    print "Ah, pues no lo es"
14
15else:
16    print "Pues a mí me convirtió en grillo"
17    print "..."
18    print "Y mejoré."
```

```
1#!/usr/bin/python
2# -*- coding: utf-8 -*-
3"""romani.py"""
4
5frase = "Romani ite domum"
6
7i = 0
8
9while i < 100:
10    print frase
11    i += 1
```

Ing. César Aranda

10

Control de Flujo 2

```
1#!/usr/bin/python
2# -*- coding: utf-8 -*-
3"""monsieur.py"""
4
5platos = ("moules marinieres", "paté de foie gras", \
6"caviar de beluga", "huevos Benedictinos", \
7"tarte de puerros, que es una tarta de puerros", \
8"ancas de rana Amandine", "puré de champiñones", "galletita")
9
10for plato in platos:
11    if plato != "galletita":
12        print "Monsieur se come el plato de " + plato
13    else:
14        print "BOOM"
```

```
1#!/usr/bin/p
2# -*- coding
3"""pelis.py"""
4
5peliculas = {"La vida de Brian": 1979, \
6"Los caballeros de la mesa cuadrada": 1974, \
7"El sentido de la vida": 1986}
8
9for nombre in peliculas:
10    print "la película " + nombre + " se rodó en " + str(peliculas[nombre])
11
12for nombre, fecha in peliculas.iteritems():
13    print "la película " + nombre + " se rodó en " + str(fecha)
```

Ing. César Aranda

11

Complemento pitónico

t = 12345, 54321, 'hola!'

x, y, z = t

¿cuál es
la forma
correcta?

dict([('sape', 4139), ('guido', 4127), ('jack', 4098)])

dict(sape=4139, guido=4127, jack=4098)

{'sape': 4139, 'jack': 4098, 'guido': 4127}

```
4 numeros = [1, 2, 4, 5, 8, 6]
5
6 for n in numeros:
7     if n == 3:
8         break
9 else:
10    print('No se encontró el número 3')
```

```
for n in range(2,10):
    for x in range(2,n/2+1):
        if n % x == 0:
            print(n, ' es divisible por ', x)
            break
    else:
        print(n, 'es primo')
```

Ing. César Aranda

12

Control de Flujo: Excepciones

```
1#!/usr/bin/python
2# -*- coding: utf-8 -*-
3"""error.py"""
4
5dividendo = 3
6divisor = 0
7
8try:
9    resultado = dividendo / divisor
10    print "La división resulta: " + str(resultado)
11except:
12    if divisor == 0:
13        print "No puedes dividir por cero"
```

Ing. César Aranda

```
1#!/usr/bin/python
2# -*- coding: utf-8 -*-
3"""error2.py"""
4
5dividendo = "A"
6divisor = 2
7
8try:
9    resultado = dividendo/divisor
10except ZeroDivisionError:
11    if divisor == 0:
12        print "No puedes dividir por cero"
13except TypeError:
14    print "Hay que ser bruto: eso no es un número"
15else:
16    print "La división resulta: " + str(resultado)
```

13

Funciones 1

```
1#!/usr/bin/python
2# -*- coding: utf-8 -*-
3"""conejo.py"""
4
5def conejo_canibal(caballero):
6
7    if caballero == "Arturo":
8        print "No, hombre, al rey Arturo no"
9    else:
10        print "El conejo asesina a Sir " + caballero
11
12conejo_canibal("Robin")
13conejo_canibal("Galahad")
14conejo_canibal("Arturo")
```

Ing. César Aranda

```
1#!/usr/bin/python
2# -*- coding: utf-8 -*-
3"""pilatos.py"""
4
5def liberar(nombre):
6
7    nuevo_nombre = nombre.replace("r", "g")
8    nuevo_nombre = nuevo_nombre.replace("R", "G")
9
10    return nuevo_nombre
11
12malosos = ["Roger", "Roderik el Rufián", "Brian"]
13
14for nombre in malosos:
15    print "Liberag a " + liberar(nombre)
```

Funciones 2

```
def get_nros():  
    monton = [12, 3, 19, 90, 87, 13, 55, 32]  
    for cual in monton:  
        yield cual*2  
  
for nro in get_nros():  
    print nro
```

```
def pedir_confirmacion(prompt, reintentos=4, recordatorio='Por favor, intente nuevamente!'):  
    while True:  
        ok = input(prompt)  
        if ok in ('s', 'S', 'si', 'Si', 'SI'):
```

Ing. César Aranda

15

Técnicas de iteración

```
a = ['Mary', 'tenia', 'un', 'cordero']  
for i in range(len(a)):  
    print(i, a[i])
```

```
caballeros = {'gallahad': 'el puro', 'robin': 'el valiente'}  
for k, v in caballeros.items():  
    print(k, v)
```

```
for i, v in enumerate(['ta', 'te', 'ti']):  
    print(i, v)
```

```
preguntas = ['nombre', 'objetivo', 'color favorito']  
respuestas = ['lancelot', 'el santo grial', 'azul']  
for p, r in zip(preguntas, respuestas):  
    print('Cual es tu {0}? {1}'.format(p, r))
```

```
canasta = ['manzana', 'naranja', 'manzana', 'pera', 'naranja', 'banana']  
for f in sorted(set(canasta)):  
    print(f)
```

Ing. César Aranda

16