



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Quintana Martínez Marco Antonio

Profesor:

Estructura de Datos y Algoritmos I

Asignatura:

17

Grupo:

09-Introducción a Python (I).

No de Práctica(s):

Galdamez Pozos Yoav Farid

Integrante(s):

2020-2

Semestre:

31/03/2020

Fecha de entrega:

CALIFICACIÓN: _____

Objetivo:

Aplicar las bases del lenguaje de programación Python en el ambiente de Jupyter notebook.

Introducción:

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional.

Desarrollo:

```
h = 20
print(h)
```

```
u = "Hola Mundo"
print(u)
```

20

Hola Mundo

```
h = i = j = 20
print(h,i,j)
```

20 20 20

```
SEGUNDOS_POR_DIA = 60*60*24
PI=3.14
```

```
cadena1 = 'Hola |'
cadena2 = "Farido"
print(cadena1)
print(cadena2)

concat_cadenas = cadena1 + cadena2
print(concat_cadenas)
```

Hola

Farido

Hola Farido

```
num_cadena = concat_cadenas + ' ' + str(7)
```



```
print(num_cadena)
```

Hola Farido 7

```
numm_cadena = '{} {} {}'.format(cadena1, cadena2, 27)
```

```
print(numm_cadena)
```

Hola Farido 27

```
nummm_cadena= "Cambiando el orden: {1} {2} {0} #".format(cadena1, cadena2, 3)
```



```
print(nummm_cadena)
```

Cambiando el orden: Farido 3 Hola #

3

2

```
print(2+1)
```

```
print(2*1)
```

1

```
print(2-1)
```

2.0

```
print(2/1)
```

0

```
print(2%1)
```

```
print(((2*1)+30)/50)
```

0.64

```
print(2**3)
```

8

```
print(8 > 29)           False
print(9 < 1)            False
print(_((11*2)-2) == (10+10)) True
print("curso" != "cUrsO") True
```

```
lista_vocales=["a", "e", "i", "o", "u"] ['a', 'e', 'i', 'o', 'u']
print(lista_vocales)                    a
print(lista_vocales[0])                  i
print(lista_vocales[2])                  u
print(lista_vocales[4])
```

```
lista_vocales=[["a", 0], ["e", 1], ["i", 2], ["o", 3], ["u", 4]]
print(lista_vocales)
```

```
|
print(lista_vocales[0])
print(lista_vocales[2])

print(lista_vocales[4][1])
print(lista_vocales[4][0])
..._
[['a', 0], ['e', 1], ['i', 2], ['o', 3], ['u', 4]]
['a', 0]
['i', 2]
4
u
```

```
lista_vocales[4][0] = 3
print(lista_vocales[4]) [3, 4]
```

```
print(tupla_vocales) ('a', 'e', 'i', 'o', 'u')
print(tupla_vocales[0]) a
print(tupla_vocales[2]) i
print(tupla_vocales[4]) u
```

```
tupla_vocales=[["a", 0], ["e", 1], ["i", 2], ["o", 3], ["u", 4]]
print(tupla_vocales)
```

```
print(tupla_vocales[0])
print(tupla_vocales[2])
```

```
print(tupla_vocales[4][1])
print(tupla_vocales[4][0])
```

```
['a', 0], ['e', 1], ['i', 2], ['o', 3], ['u', 4]]
['a', 0]
['i', 2]
4
u
```

```
print("Valor actual {}".format(lista_vocales[0]))
lista_vocales[0]=20
print("Valor cambiado {}".format(lista_vocales[0]))
tupla_vocales[0]=20
```

Valor actual ['a', 0]

Valor cambiado 20

```
from collections import namedtuple

pais = namedtuple('paises', ['nombre', 'numero'])

pais1 = pais('Mexico', 1)

pais2 = pais('Francia', 2)

print(pais1.nombre, pais1.numero)

print(pais2.numero, pais2.nombre)

print("Campos de tupla {}".format(pais1._fields))
```

Mexico 1

2 Francia

Campos de tupla ('nombre', 'numero')

```
elementos = {'hidrogeno': 1, 'helio': 2, 'carbon': 6}
print(elementos)
print(elementos['hidrogeno'])
```

{'hidrogeno': 1, 'helio': 2, 'carbon': 6}

1

```
elementos['litio'] = 3
elementos['nitrogeno'] = 8
```

```
print(elementos)
```

{'hidrogeno': 1, 'helio': 2, 'carbon': 6, 'litio': 3, 'nitrogeno': 8}

```
perros = {}
perros['shiba'] = {'nombre': 'Gohan', 'peso': 20}
perros['chihuahua'] = {'nombre': 'Carlos', 'peso': 5}

print(perros)
```

```
{'shiba': {'nombre': 'Gohan', 'peso': 20}, 'chihuahua': {'nombre': 'Carlos', 'peso': 5}}
```

```
print(perros['shiba'])
print(perros['shiba']['nombre'])
print(perros['shiba']['peso'])
perros['shiba']['peso'] = 18
print(perros['shiba']['peso'])
```

```
{'nombre': 'Gohan', 'peso': 20}
```

```
Gohan
```

```
20
```

```
18
```

```
perros['shiba'].update({'pedigree': True})
print(perros['shiba'])
```

```
{'nombre': 'Gohan', 'peso': 18, 'pedigree': True}
```

```
print(perros.items())
```

```
print(perros.keys())
```

```
dict_items([('shiba', {'nombre': 'Gohan', 'peso': 18, 'pedigree': True}), ('chihuahua', {'nombre': 'Carlos', 'peso': 5})])
dict_keys(['shiba', 'chihuahua'])
```

```
def imprime_nombre(nombre):
    print("hola" + nombre)
```

```
imprime_nombre("Farido")
```

holaFarido

```
def cuadrado(x):
    return x**2

x=5
print("El cuadrado de {} es {}".format(x, cuadrado(x)))
```

El cuadrado de 5 es 25

```
- def varios(x) :  
-     return x**2, x**3, x**4  
val1, val2, val3 = varios(2)  
print("{} {} {}".format(val1, val2, val3))
```

4 8 16

```
- def cuadrado_default(x=3):  
-     return x**2  
cuadrado_default()  
val4, _, val5 = varios(2)  
print("{} {}".format(val4, val5))
```

4 16

```
- def funcion_v1():  
-     print(vg) Global  
funcion_v1()  
print(vg) Global
```

```
- def funcion_v2():  
-     vg = "Local" Local  
-     print(vg)  
funcion_v2() Global  
print(vg)
```

```
- def funcion_v3():  
-     print(vg)  
-     vg = "Local"  
-     print(vg)
```

funcion_v3()

Traceback (most recent call last):

```
File "C:/Users/Invitado2/PycharmProjects/Practcia09/ejercicio01", line 159  
    funcion_v3()  
File "C:/Users/Invitado2/PycharmProjects/Practcia09/ejercicio01", line 156  
    print(vg)
```

UnboundLocalError: local variable 'vg' referenced before assignment

```
def funcion_v4():
    global vg
    print(vg)
    vg = "Local"
    print(vg)
funcion_v4()
print(vg)
```

Global

Local

Local

```
PI = 3.14
BASE_TR = 2
BASE_REC = 4
RADIO = 3
BASE_Menor = 2
BASE_Mayor = 4
ALTURA_TR = 3
ALTURA_REC = 2
LADO_A= 5
LADO_B=6
ALTURA_TRA=3
CATETO_A=3
CATETO_B=4

perimetro_tr=(CATETO_A+CATETO_B+BASE_TR)
perimetro_rec=((BASE_REC*2)+(ALTURA_REC*2))
perimetro_circ=((2*PI)*RADIO)
perimetro_tra=(BASE_Mayor+BASE_Menor+LADO_A+LADO_B)
print("El perimetro de un triangulo rectangulo de lados {}, {} y {} es {}".format(CATETO_A, CATETO_B, BASE_TR, perimetro_tr))
print("El perimetro de un rectangulo de lados {} y {} es {}".format(BASE_REC, ALTURA_REC, perimetro_rec))
print("El perimetro de un circulo de radio {} es {}".format(RADIO, perimetro_circ))
print("El perimetro de un trapecio de lados {}, {}, {} y {} es {}".format(BASE_Menor, BASE_Mayor, LADO_B, LADO_A, perimetro_tra))

area_tr=(BASE_TR*ALTURA_TR)/2
area_rec=(BASE_REC*ALTURA_REC)
def area_circ(x):
    z=(PI*(x**2))
    return z
area_tra=((BASE_Mayor+BASE_Menor)*ALTURA_TRA)/2
print("El area de un triangulo rectangulo de base {} y altura {} es {}".format(BASE_TR, ALTURA_TR, area_tr))
print("El area de un rectangulo de base {} y altura {} es {}".format(BASE_REC, ALTURA_REC, area_rec))
z=RADIO
print("El area de un circulo de radio {} es {}".format(RADIO, area_circ(z)))
print("El area de un trapecio de base mayor {}, base menor {} y altura {} es {}".format(BASE_Mayor, BASE_Menor, ALTURA_TRA, area_tra))

print("FARIDO")
```



```
El perimetro de un triangulo rectangulo de lados 3, 4 y 2 es 9
El perimetro de un rectangulo de lados 4 y 2 es 12
El perimetro de un circulo de radio 3 es 18.84
El perimetro de un trapecio de lados 2, 4, 6 y 5 es 17
El area de un triangulo rectangulo de base 2 y altura 3 es 3.0
El area de un rectangulo de base 4 y altura 2 es 8
El area de un circulo de radio 3 es 28.26
El area de un trapecio de base mayor 4, base menor 2 y altura 3 es 9.0
FARIDO
```

Conclusión:

El aprender un nuevo lenguaje de programación me llena de ilusión ya que me siento cada vez más preparado para afrontar las asignaturas que vienen por delante, me siento cada vez más seguro de lo que sé, Python me está resultando más sencillo que C pero imagino que es porque ya tengo las nociones básicas de la programación y dado eso creo que podría aprender Python de una manera más fácil.

Bibliografía:

Tutorial oficial de Python: <https://docs.python.org/3/tutorial/>

Galería de notebooks: <https://wakari.io/galler>