



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

M.I. MARCO ANTONIO MARTINEZ QUINTANA

Profesor:

EDA I

Asignatura:

17

Grupo:

9

No de Práctica(s):

MAGALLANES GARCÍA ELVIRA VALENTINA

Integrante(s):

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada:

2020-2

Semestre:

31/03/2020

Fecha de entrega:

Observaciones:

CALIFICACIÓN: _____

Objetivo:

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

Introducción:

Variables y tipos:

- Los nombres de las variables son alfanuméricos (a-z, A-Z, 0-9) y empiezan con una letra en minúscula.
- No se especifica el tipo de valor que una variable contiene, está implícito al momento de asignar un valor. No se necesita poner ; al final de cada instrucción.
- Mantener las indentaciones al momento de escribir código.

Cadenas

Las cadenas pueden ser definidas usando comilla simple (') o comilla doble ("). Una característica especial de las cadenas es que son inmutables, esto quiere decir que no se pueden cambiar los caracteres que contiene. El carácter \ sirve para escapar caracteres como \n o \t.

Desarrollo:

Variables y tipos:

```
In [1]: #Iniciando variables
```

```
In [2]: x = 10
```

```
In [3]: print(x)  
10
```

```
In [4]: cadena = "Hola Mundo"
```

```
In [5]: print(cadena)  
Hola Mundo
```

```
In [8]: type(x)  
Out[8]: int
```

```
In [9]: type(cadena)  
Out[9]: str
```

Cadenas

```
1 #Iniciando cadenas  
2 cadena1='Hola'  
3 cadena2="Mundo"  
4 print(cadena1)  
5 print(cadena2)  
6 concat_cadenas=cadena1+cadena2  
7 print(concat_cadenas)
```

```
In [10]: runfile('C:/Users/jmaga/.spyder-py3/temp.py', wdir='C:/
Users/jmaga/.spyder-py3')
Hola
Mundo
HolaMundo
```

```
#Para concatenar un número y una cadena se debe usar la función str()
num_cadena = concat_cadenas + ' ' + str(3) #Se agrega una cadena vacía para agregar un espaci
o
print(num_cadena)
```

Hola Mundo 3

```
#El valor de la variable se va a imprimir en el lugar donde se encuentre {} en la cadena
num_cadena = "{} {} {}".format(cadena1, cadena2, 3)
print(num_cadena)
```

Hola Mundo 3

```
#Cuando se agrega un número dentro de {}, el valor la variable que se encuentra en esa posic
ión
#dentro de la función format(), será impreso.
num_cadena = "Cambiando el orden: {1} {2} {0} #".format(cadena1, cadena2, 3)
print(num_cadena)
```

Cambiando el orden: Mundo 3 Hola #

Operadores

```
#Para el exponente se puede utilizar asterisco
print( 1 + 5 )
print( 6 * 3 )
print( 10 - 4 )
print( 100 / 50 )
print( 10 % 2 )
print( ((20 * 3) + (10 +1)) / 10 )
print( 2**2 )
```

6
18
6
2.0
0
7.1
4

```
print (7 < 5) #Falso

print (7 > 5) #Verdadero

print ((11 * 3)+2 == 36 - 1) #Verdadero

print ((11 * 3)+2 >= 36) #Falso

print ("curso" != "CuRsO") #Verdadero
```

False

True

True

False

True

Ejercicio adicional:

```
1 #Programa que calcula perimetro y area de un triangulo
2 a=5 #altura
3 b=5 #base
4 area=b*a/2
5 print(area)
6 perimetro=a+b
7 print(perimetro)
```

```
In [11]: runfile('C:/Users/jmaga/.spyder-py3/temp.py', wdir='C:/Users/jmaga/.spyder-py3')
12.5
10
```

```
1 #Programa que calcula el perimetro y area de un circulo
2 pi=3.14
3 radio=10
4 area=pi*radio**2
5 print(area)
6 perimetro=2*pi*radio
7 print(perimetro)
8
```

```
In [12]: runfile('C:/Users/jmaga/Desktop/CODIGOS EN PYTHON/Sin título0.py', wdir='C:/Users/jmaga/Desktop/CODIGOS EN PYTHON')
314.0
62.800000000000004
```

```
1 #Programa que da el area de un trapecio
2 h=10
3 b=5
4 c=1
5 area=(b+c)*h/2
6 print(area)
```

```
In [15]: runfile('C:/Users/jmaga/Desktop/CODIGOS EN PYTHON/Sin título1.py', wdir='C:/Users/jmaga/Desktop/CODIGOS EN PYTHON')
```

```
30.0
```

```
1 #Programa que da el area y perimetro de un rectangulo
2 base=10
3 altura=10
4 area=base*altura
5 print(area)
6 perimetro=2*(altura+base)
7 print(perimetro)
```

```
In [16]: runfile('C:/Users/jmaga/Desktop/CODIGOS EN PYTHON/Sin título2.py', wdir='C:/Users/jmaga/Desktop/CODIGOS EN PYTHON')
```

```
100
```

```
40
```

Conclusiones:

Python parece ser un lenguaje mucho más sencillo en la parte de sintaxis que C, lo cual hace mas sencillo el programar, la práctica funcionó para ver lo básico y darnos una idea de como es este lenguaje, no me pareció complicada, pero si necesito mucho más práctica.