



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Adrian Ulises Mercado Martinez

*Asignatura:* Estructura de Datos y Algoritmos I

*Grupo:* 13

*No de Práctica(s):* 10

*Integrante(s):* Méndez Bernal Luis Alberto

*No. de Equipo de  
cómputo empleado:*

*No. de Lista o Brigada:* 13

*Semestre:* 2020-2

*Fecha de entrega:* 7 de Junio del 2020

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

## Introducción

Esta práctica se llama “Introducción a Python (II)” tiene el objetivo de que apliquemos el lenguaje de Python en el entorno de notebook.

Python es uno de los programas de programación que se puede usar a muchas cosas, ya que puede ser orientado a objetos, programación imperativa, y también puede servir como programación funcional.

La practica espera que podamos usar mejor este código, conocer librerías y aprender a graficar usando este programa.

## Desarrollo

### IF

Aquí el comando IF sirve exactamente igual que en C.

```
1 //Mendez Bernal Luis Alberto
2
3 def obtenerMayor (param1, param2);
4     if param1 < param 2;
5         print('{} es mayor que {}'.format(param2, param 1))
6
```

Aquí definimos una nueva función que se llamará “obtener mayor” y lo que hará será comparar dos números que pongamos y es el primero es mayor que el segundo entonces imprimirá algo

Ejemplo:

```
obtenerMayor(24, 56)

56 es mayor que 24
```

En caso de que esto sea negativo, el programa simplemente no imprime nada y se acaba.

### IF-ELSE

Este tipo de declaración funciona igual que en C. Si una condición no se cumple. Entonces el programa pasara al segundo caso.

```
11 def obtenerMayorV2(param1, param2):
12     if param1 < param2:
13         return param2
14     else:
15         return param1
16
```

Eso sirve para que el programa siempre te devuelva el numero mayor de dos que le damos.

### IF-ELIF-ELSE

Este tipo de declaración tiene el fin de funcionar igual que un Switch en programación tipo C. Este es muy útil para hacer un menú.

```

18 def numeros(num):
19     if num==1:
20         print("El numero es 1")
21     if num==12:
22         print("El numero es 2")
23     if num==3:
24         print("El numero es 3")
25     if num==4:
26         print("El numero es 4")
27     else:
28         print("Opcion invalida")

```

Aquí creamos varios casos que actuaran dependiendo de que numero le pongas. El ese es para los casos en los que se pone un valor que no estaba planeado ser recibido.

## CICLO WHILE

Un ciclo while permite que un fragmento de código se repita tantas veces como la condición lo marque.

```

31 def cuenta(limite):
32     i = limite
33     while True:
34         print(i)
35         i = i-1
36         if i == 0:
37             break
38

```

## ITERACION EN DICCIONARIOS

Las iteraciones sirven para renombrar números, valores u otras cosas para que sea más fácil mostrarlas en otros lados.

```

39 elementos = { 'hidrogeno': 1, 'helio': 2, 'litio': 3}
40
41 for llave, valor in elementos.items():
42     print (llave, '=', valor)
43

```

Para mostrar ambos valores:

```

for llave, valor in elementos.items():
    print (llave, '=', valor)

helio = 2
litio = 3
hidrogeno = 1

```

Para mostrar solo los elementos

```

for llave in elementos.keys():
    print(llave)

helio
litio
hidrogeno

```

Para mostrar solo el numero:

```
for valor in eementos.value():
    print(valor)

2
3
1
```

## BIBLIOTECAS

Las bibliotecas son las bases para que Python pueda trabajar, sin estas Python se volvería inútil. Para usar estas bibliotecas se debe poner “import” antes de la biblioteca que usaremos.

```
63 import math
64
65 x = math.cos(math.pi)
66
67 print x
68 -1.0
```

Aquí notamos que siempre se tiene que poner el nombre de la biblioteca antes de la función que usaremos de ella, pero esto se puede solucionar

```
70 from math import *
71
72 x =cos(pi)
73
74 print (x)
75
76 -1.0
```

Así ya no tendremos que poner “math” cada vez que usemos alguna función de aquí.

Python nos permite muchas cosas, una de ellas es ver todas las funciones de una biblioteca

```
print(dir(math))
```

También podemos ver como se usa una función

```
help(math.log)
```

Y podemos definir la biblioteca para facilitarnos las cosas

```
import math as Ma
```

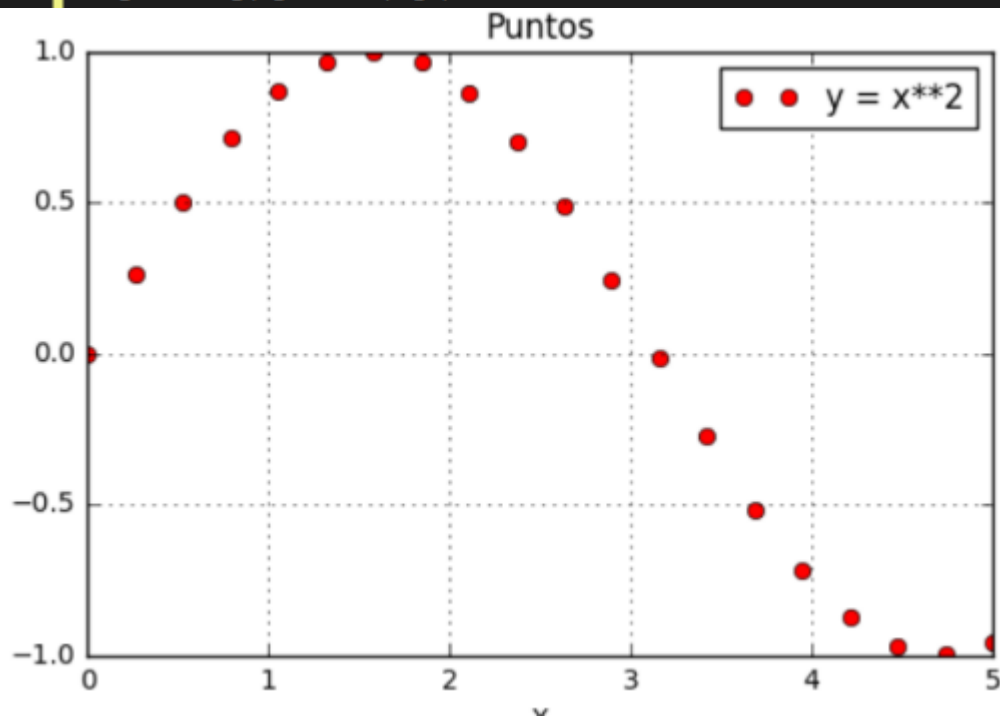
## GRAFICACION

Para graficar se usa la biblioteca “Matplotlib”, lo que hace es darte todas las funciones necesarias para que se cree una grafica en la forma que la pidas.

```

85 import matplotlib.pyplot as plt
86 from mpl_toolkits.mplot3d import Axes3D
87
88 x = linspace (0, 5, 20)
89
90 fig, ax = plt.subplots(facecolor='w', edgecolor='k')
91 ax.plot (x, sin(x), marker='o', color="r", linestyle='None')
92
93 ax.grid(True)
94 ax.set_xlabel('X')
95 ax.set_ylabel('Y')
96 ax.grid(True)
97 ax.legend(["y = x**2"])
98
99 plt.title('Puntos')
100 plt.show()
101
102 fig.savefig("grafica.png")

```



## Conclusiones

En lo personal siento que el lenguaje de programación Python ha sido mucho más fácil de usar que C, aquí te permiten mas facilidades que en el otro. Además de que Python te permite descargar y meter nuevas bibliotecas que pueden facilitar mucho tu trabajo.