



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Adrian Ulises Mercado Marínez

*Asignatura:* Estructura de Datos y Algoritmos I

*Grupo:* 13

*No de Práctica(s):* 10

*Integrante(s):* Martínez Salgado Elean Jim

*No. de Equipo de  
cómputo empleado:* -

*No. de Equipo de  
cómputo empleado:*

*No. de Lista o Brigada:*

*Semestre:* 2020-2

*Fecha de entrega:* 7 de junio de 2020

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

# Introducción

En esta practica se continua una instrucción al lenguaje de programación Python y en esta ocasión se nos enseña a como utilizar bibliotecas, estructuras de iteración y además a como graficar.

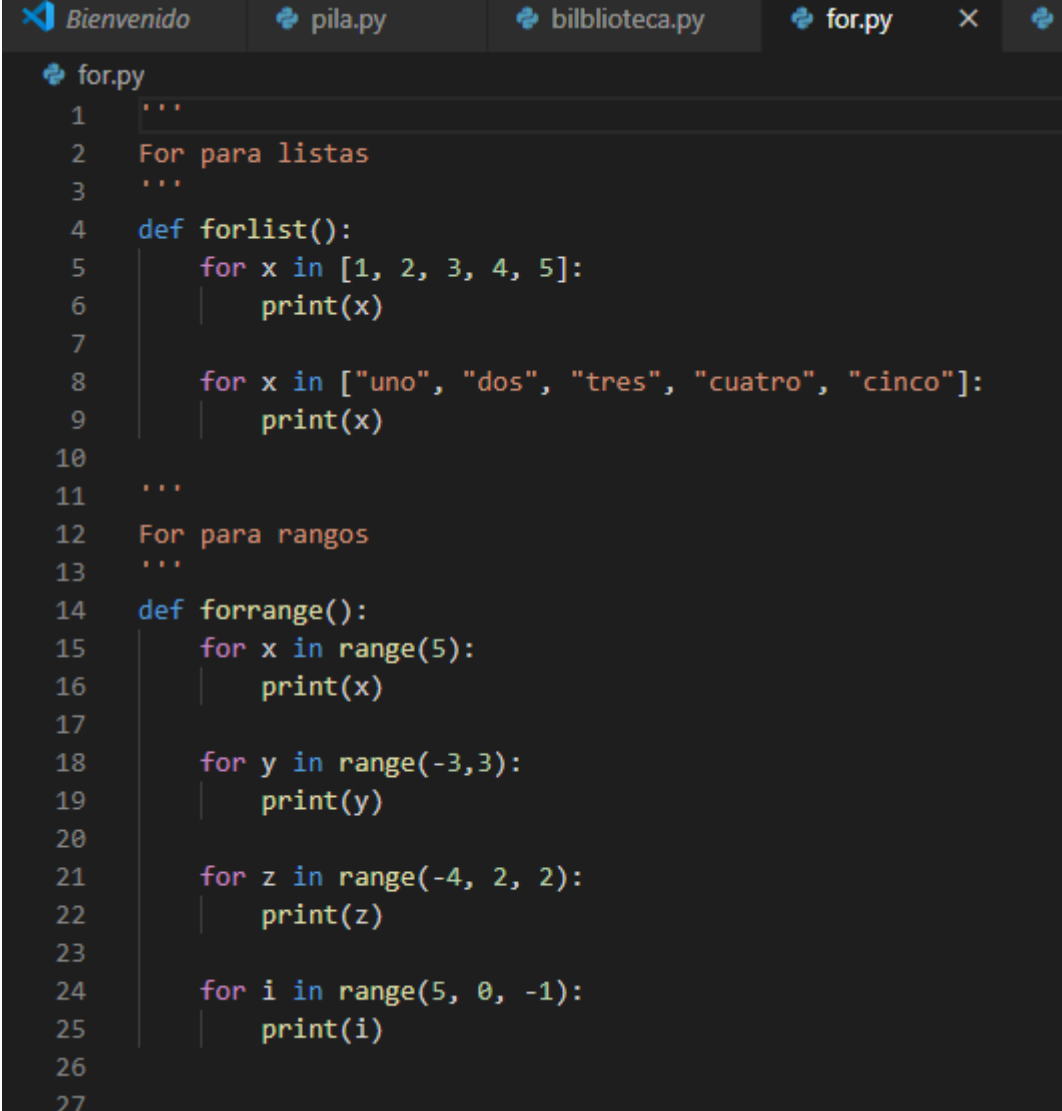
## Desarrollo

- 1- Creamos un archivo py sobre el cual utilizaremos la biblioteca math para poder hacer uso de funciones matemáticas además damos un nombre a las funciones para poder llamarlas más fácilmente (math.cos como cos), además de apoyarnos en la función help que nos ayuda a conocer el trabajo de cada función.

```
bilblioteca.py ×  for.py  grafica
bilblioteca.py
1  #import math
2  #from math import *
3  from math import cos, pi
4  #x = math.cos(math.pi)
5  x = cos(pi)
6  print(x)
```

```
while.py  bilblioteca2.py ×
bilblioteca2.py
1  #import math
2
3  #print(dir(math))
4  #help(math.log)
5
6  import math as ma
7
8  x = ma.cos(ma.pi)
```

- 2- Comenzamos con las estructuras de iteración la primera en ser analizada fue la estructura for así como sus aplicaciones (diccionarios, listas, uso de arreglos) además de implementar un else en un for.



The image shows a code editor with a dark theme. At the top, there are four tabs: 'Bienvenido', 'pila.py', 'biblioteca.py', and 'for.py'. The 'for.py' tab is active. The code in the editor is as follows:

```
1  '''
2  For para listas
3  '''
4  def forlist():
5      for x in [1, 2, 3, 4, 5]:
6          print(x)
7
8      for x in ["uno", "dos", "tres", "cuatro", "cinco"]:
9          print(x)
10
11  '''
12  For para rangos
13  '''
14  def forrange():
15      for x in range(5):
16          print(x)
17
18      for y in range(-3,3):
19          print(y)
20
21      for z in range(-4, 2, 2):
22          print(z)
23
24      for i in range(5, 0, -1):
25          print(i)
26
27
```

```

...
For para diccionarios
...
def fordic():
    diccionario = {'manzana':1, 'pera':3, 'uva':10}
    for clave, valor in diccionario.items():
        print(clave, " = ", valor)

    for clave in diccionario.keys():
        print(clave)

    for valor in diccionario.values():
        print(valor)

    for idx, x in enumerate(diccionario):
        print("El indice {} del elemneto {}".format(idx,x))

```

```

...
Else de For
...

def elsefor():
    for x in range(5):
        print(x)
    else:
        print("La cuenta se termino")

def elsefor2():
    for x in range(5):
        print(x)
        if x == 2:
            break
    else:
        print("La cuenta se termino")

if __name__ == "__main__":
    forlist()
    forrange()
    fordic()
    elsefor()
    elsefor2()

```

### 3- Estructura if

```
if.py
1  def numeroMayor(a, b, c):
2      if a > b and a > c:
3          print("El numero mayor es {}".format(a))
4      elif(b > c and b > a):
5          print("El nuemro mayor es {}".format(b))
6      else:
7          print("El numero mayor es {}".format(c))
8
9
10 if __name__ == "__main__":
11     a = int(input())
12     b = int(input())
13     c = int(input())
14     numeroMayor(a,b,c)
```

### 4- While

```
while.py
1  def factorial(n):
2      i = 2
3      temp = 1
4      while i <=n:
5          temp = temp * i
6          i = i + 1
7      return temp
8
9  if __name__ == "__main__":
10     a = int(input("Ingresa un numero: "))
11     print(factorial(a))
```

- 5- Vimos como aplicar lo anterior a distintos tipos de programas en este caso en la pila en donde las funciones esenciales ya están definidas en Python mientras que en c no.

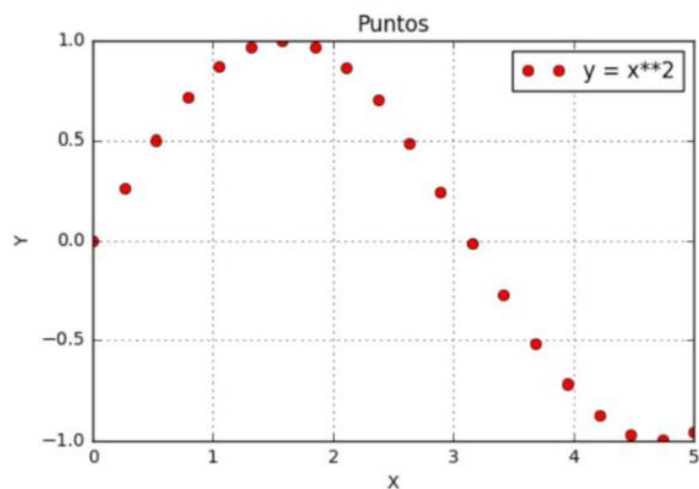
```
pila.py
1  def insertar(lista, dato):
2      lista.append(dato)
3
4
5  def borrar(lista):
6      dato = lista.pop()
7      return dato
8
9  def imprimir_pila(lista):
10     lista.reverse()
11     for x in lista:
12         print(x)
13     print()
14     lista.reverse()
15
16
17 def main():
18     pila = []
19     insertar(pila, "lista1")
20     insertar(pila, 2)
21     imprimir_pila(pila)
22     print()
23     print(borrar(pila))
24     imprimir_pila(pila)
25
26
27 if __name__ == "__main__":
28     main()
```

6- Realizamos una grafica utilizando la biblioteca matplotlib.

```

Bienvenido  pila.py  biblioteca.py  for.py  grafica.p
grafica.py
1  import matplotlib.pyplot as plt
2  from mpl_toolkits.mplot3d import Axes3D
3  from math import sin
4  import numpy as np
5
6
7  x = 5
8
9
10 fig, ax = plt.subplots(facecolor='w', edgecolor='k')
11 ax.plot(x, sin(x), marker="o", color="r", linestyle='None')
12
13 ax.grid(True)
14 ax.set_xlabel('X')
15 ax.set_ylabel('Y')
16 ax.grid(True)
17 ax.legend(["y = x**2"])
18
19 plt.title('Puntos')
20 plt.show()
21
22 fig.savefig("Grafica.png")

```



## **Conclusiones**

A mi parecer Python es un lenguaje mas sencillo que c por las sus funciones con las que cuenta, sin embargo es bastante distinto de c la hora de escribir código por lo que hay que practicarlo con el fin de familiarizarse y dominarlo.