



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Marco Antonio Martínez Quintana

*Asignatura:* Estructura de Datos y Algoritmos I

*Grupo:* 17

*No de Práctica(s):* 10

*Integrante(s):* Pacheco Barbosa Angel David

*No. de Equipo de  
cómputo* 24

*Brigada:*

*Semestre:* 2020-2

*Fecha de entrega:* 19 de abril de 2020

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

## Objetivo:

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

### EJERCICIO 1

```
def obtenerMayor(param1,param2):  
    if param1 < param2:  
        print('{} es mayor que {}'.format(param2, param1))  
obtenerMayor(5,7)
```

```
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ python EJ1.py  
7 es mayor que 5  
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$
```

### EJERCICIO 2

```
x=y=z=3  
if x==y==z:  
    print(True)
```

```
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ python EJ2.py  
True  
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$
```

### EJERCICIO 3

```
def obtenerMayorv2(param1,param2):  
    if param1<param2:  
        return param2  
    else:  
        return param1  
print("El mayor es {}".format(obtenerMayorv2(4,20)))  
print("El mayor es {}".format(obtenerMayorv2(11,6)))
```

```
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ python EJ3.py  
El mayor es 20  
El mayor es 11  
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$
```

### EJERCICIO 4

```
def obtenerMayor_idiom(param1,param2):  
    #La variable valor va a tener el valor de param2 si es verdadero  
    #de lo contrario el valor de param1  
    valor = param2 if (param1 < param2) else param1  
    return valor  
  
print("El mayor es {}".format(obtenerMayor_idiom(11,6)))
```

```
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ python EJ4.py
El mayor es 11
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$
```

## EJERCICIO 5

```
def numeros(num):
    if num==1:
        print("Tu numero es 1")
    elif num==2:
        print("Tu numero es 2")
    elif num==3:
        print("Tu numero es 3")
    elif num==4:
        print("Tu numero es 4")
    else:
        print("No hay opcion")

numeros(2)
numeros(5)
```

```
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ python EJ5.py
Tu numero es 2
No hay opcion
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$
```

## EJERCICIO 6

```
def numeros_idiom(num):
    #La tupla tiene las opciones validas
    if num in (1,2,3,4):
        print("tu numero es {}".format(num))
    else:
        print("{} no es una opcion".format(num))

numeros_idiom(2)

numeros_idiom(5)
```

```
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ python EJ6.py
tu numero es 2
5 no es una opcion
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$
```

## EJERCICIO 7

```
def obtenerMasGrande(a,b,c):  
    if a > b:  
        if a > c:  
            return a  
        else:  
            return c  
    else:  
        if b > c:  
            return b  
        else:  
            return c  
  
print("El mas grande es {}".format(obtenerMasGrande(7,13,1)))|
```

```
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ python EJ7.py  
El mas grande es 13  
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ █
```

## EJERCICIO 8

```
def cuenta(limite):  
    i= limite  
    while True:  
        print(i)  
        i = i -1  
        if i == 0:  
            break #rompiendo el ciclo  
  
cuenta(10)|
```

```
angeldpb2@angeldpb2-ZX4970:~$ cd 'EDA I'/Python  
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ python EJ8.py  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1  
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ █
```

## EJERCICIO 9

```
def factorial (n):  
    i = 2  
    tmp = 1  
    while i < n+1:  
        tmp = tmp*i  
        i = i + 1  
    return tmp  
print(factorial (4))  
  
print(factorial (6))|
```

```
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ python EJ9.py  
24  
720  
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ █
```

## EJERCICIO 10

```
for x in [1,2,3,4,5]:  
    print(x)|
```

```
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ python EJ10.py  
1  
2  
3  
4  
5  
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ █
```

## EJERCICIO 11

```
for x in range(5):  
    print(x)|
```

```
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ python EJ11.py  
0  
1  
2  
3  
4  
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ █
```

## EJERCICIO 12

```
for x in range(-5,2):  
    print(x)
```

```
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ python EJ12.py  
-5  
-4  
-3  
-2  
-1  
0  
1  
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$
```

## EJERCICIO 13

```
for num in ["uno", "dos", "tres", "cuatro"]:  
    print(num)
```

```
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ python EJ13.py  
uno  
dos  
tres  
cuatro  
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$
```

## EJERCICIO 14

```
#Creando un diccionario  
elementos = { 'hidrogeno': 1, 'helio': 2, 'carbon': 6 }  
  
for llave, valor in elementos.items():  
    print llave, " = ", valor
```

```
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ python EJ14.py  
carbon = 6  
hidrogeno = 1  
helio = 2  
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$
```

## EJERCICIO 15

```
#Creando un diccionario  
elementos = { 'hidrogeno': 1, 'helio': 2, 'carbon': 6 }  
  
for llave in elementos.keys():  
    print llave
```

```
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ python EJ15.py
carbon
hidrogeno
helio
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$
```

## EJERCICIO 16

```
#Creando un diccionario
elementos = { 'hidrogeno': 1, 'helio': 2, 'carbon': 6 }

for valor in elementos.values():
    print valor
```

```
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ python EJ16.py
6
1
2
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$
```

## EJERCICIO 17

```
elementos = { 'hidrogeno': 1, 'helio': 2, 'carbon': 6 }

#Si se necesita iterar utilizando un indice
for idx, x in enumerate(elementos):
    print("El indice es: {} y el elemento: {}".format(idx,x))
```

```
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ python EJ17.py
El indice es: 0 y el elemento: carbon
El indice es: 1 y el elemento: hidrogeno
El indice es: 2 y el elemento: helio
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$
```

## EJERCICIO 18

```
def cuenta_idiom(limite):
    for i in range(limite, 0, -1):
        print(i)
    else: #Corresponde al for, NO al IF
        print("Cuenta finalizada")
cuenta_idiom(5)
```

```
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ python EJ18.py
5
4
3
2
1
Cuenta finalizada
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$
```

## EJERCICIO 19

```
def cuenta_idiomv2(limite):  
    for i in range(limite,0,-1):  
        print(i)  
        if i == 3:  
            break  
    else:  
        print("Cuenta finalizada")  
cuenta_idiomv2(5)
```

```
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ python EJ19.py  
5  
4  
3  
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$
```

## EJERCICIO 20

```
import math  
x= math.cos(math.pi)  
  
print(x)
```

```
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ python EJ20.py  
-1.0  
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$
```

## EJERCICIO 21

```
from math import *  
  
x= cos (pi)  
  
print(x)
```

```
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ python EJ21.py  
-1.0  
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$
```

## EJERCICIO 22

```
from math import cos,pi  
x=cos(pi)  
print(x)
```

```
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$ python EJ22.py  
-1.0  
angeldpb2@angeldpb2-ZX4970:~/EDA I/Python$
```



## EJERCICIO 23

---

```
%pylab inline
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

x = linspace(0,5,20)

fig, ax = plt.subplots(facecolor='w', edgecolor= 'k')
ax.plot(x, sin(x), marker="o", color="r", linestyle='None')

ax.grid(True)
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.grid(True)
ax.legend(["y= x**2"])

plt.title('Puntos')
plt.show()

fig.savefig("grafica.png")
```

## CONCLUSIÓN

La práctica permite familiarizarse con más partes del lenguaje Python, además de que muestra la extensión y diferencia conforme al lenguaje C, pues Python permite hacer cosas más rápido que en C.

## Bibliografía

Guía práctica de estudio 10:  
Introducción a Python (II).