



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

---

*Profesor: Martínez Quintana Marco Antonio*

*Asignatura: Estructuras de Datos y Algoritmos I*

*Grupo: 17*

*No de Práctica(s): 9*

*Integrante(s): De León Arias Emiliano*

*No. de Equipo de cómputo empleado: 37*

*No. de Lista o Brigada: 13*

*Semestre: 2020-2*

*Fecha de entrega: 31 marzo 2020*

*Observaciones:*

CALIFICACIÓN: \_\_\_\_\_

# Objetivo

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

## Introducción

Python es un lenguaje de programación que te permite trabajar más rápido e integrar tus sistemas de manera mas efectiva. Puedes aprender a usar Python de manera fácil y rápida y observar como ganar productividad casi de inmediato.

En esta práctica se estará abordando acerca del lenguaje de programación Python. Este lenguaje, como cualquier otra, tiene palabras reservadas y su propia sintaxis las cuales como podemos observar son fáciles de manejar ya que son muy parecidas a las de otros lenguajes. Tiene ligeras excepciones dentro de las cuales la mas destacada es la de que ya no es necesario insertar el punto y coma al final de cada línea o no declarar librerías y función principal dentro del código.

# Desarrollo

códigos:

a) Ejercicio 1

#variables y tipos

x= 10

print(x)

cadena= "Hola Mundo"

print(cadena)

x=y=z=10

print(x,y,z)

print(type(x))

print(type(cadena))

x="Hola Mundo"

cadena=10

print(x)

print(cadena)

print(type(x))

print(type(cadena))

SEGUNDOS\_POR\_DIA=60\*60\*24

PI=3.1416

print(SEGUNDOS\_POR\_DIA)

print(PI)

#Cadenas

cadena1="Hola"

cadena2="Mundo"

print(cadena1)

print(cadena2)

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

```
num_cadena=concat_cadenas+" "+str(3)
print(num_cadena)
```

```
num_cadena="{ } { } { }".format(cadena1,cadena2,3)
print(num_cadena)
```

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

#Operadores

#Aritmeticos

```
print(1+5)
print(6*3)
print(10-4)
print(100/50)
print(10%2)
print(((20*3)+(10+1))/10)
print(2**2)
```

#Booleanos

```
print(7<5)
print(7>5)
print((11*3)+2==36-1)
print((11*3)+2>=36)
print("curso" != "CuRsO")
```

#Listas

```
lista_diasDelMes=[31,28,31,30,31,30,31,31,30,31,30,31]
print(lista_diasDelMes)
print(lista_diasDelMes[0])
print(lista_diasDelMes[6])
print(lista_diasDelMes[11])
```

```
lista_numeros=[["cero",0],["uno",1,"UNO"],["dos",2],["tres",3],["cuatro",4],["x",5]]
print(lista_numeros)
print(lista_numeros[0])
print(lista_numeros[1])
print(lista_numeros[2][0])
print(lista_numeros[2][1])
print(lista_numeros[1][0])
```

```
print(lista_numeros[1][1])
print(lista_numeros[1][2])
lista_numeros[5][0]="cinco"
print(lista_numeros[5])
```

#Tuplas

```
tupla_diasDelMes=(31,28,31,30,31,30,31,31,30,31,30,31)
```

```
print(tupla_diasDelMes)
print(tupla_diasDelMes[0])
print(tupla_diasDelMes[3])
print(tupla_diasDelMes[1])
```

```
tupla_numeros=(("cero",0),("uno",1,"UNO"),("dos",2),("tres",3),("cuatro",4),("x",5))
print(tupla_numeros)
```

```
print(tupla_numeros[0])
print(tupla_numeros[1])
```

```
print(tupla_numeros[2][0])
print(tupla_numeros[2][1])
```

```
print(tupla_numeros[1][0])
print(tupla_numeros[1][1])
print(tupla_numeros[1][2])
```

```
print("valor actual {}".format(lista_diasDelMes[0]))
lista_diasDelMes[0]=50
print("valor cambiado {}".format(lista_diasDelMes[0]))
tupla_diasDelMes[0]=50
```

#Tuplas con nombres

```
planeta=namedtuple("planeta",["nombre","numero"])
planeta1=planeta("Mercurio",1)
print(planeta1)
planeta2=planeta("Venus",2)
print(planeta1.nombre,planeta1.numero)
print(planeta2[0],planeta2[1])
print("campos de la tupla: {}".format(planeta1._fields))
```

#Diccionarios

```
elementos={"hidrogeno": 1, "Helio": 2,"Carbon": 6}
```

```
print(elementos)
print (elementos["hidrogeno"])
```

```
elementos["litio"]=3
elementos["nitrogeno"]=8
print(elementos)
```

```
elementos2={}
elementos2["H"]={"name": "Hydrogen", "number": 1, "weight": 1.000794}
elementos2["He"]={"name": "Helium", "number": 2, "weight": 4.002602}
print(elementos2)
```

```
print(elementos2["H"])
print(elementos2["H"]["name"])
print(elementos2["H"]["number"])
elementos2["H"]["weight"]=4.30
print(elementos2["H"]["weight"])
```

```
elementos2["H"].update({"gas noble": True})
print(elementos2["H"])
```

```
print(elementos2.items())
print(elementos2.keys())
```

#Funciones

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

```
imprime_nombre("JJ")
```

```
def cuadrado(x):
    return x**2
x=5
print("El cuadrado de {} es {}: ".format(x,cuadrado(x)))
```

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

```
def cuadrado_default(x=3):
    return x**2
cuadrado_default()
```

```

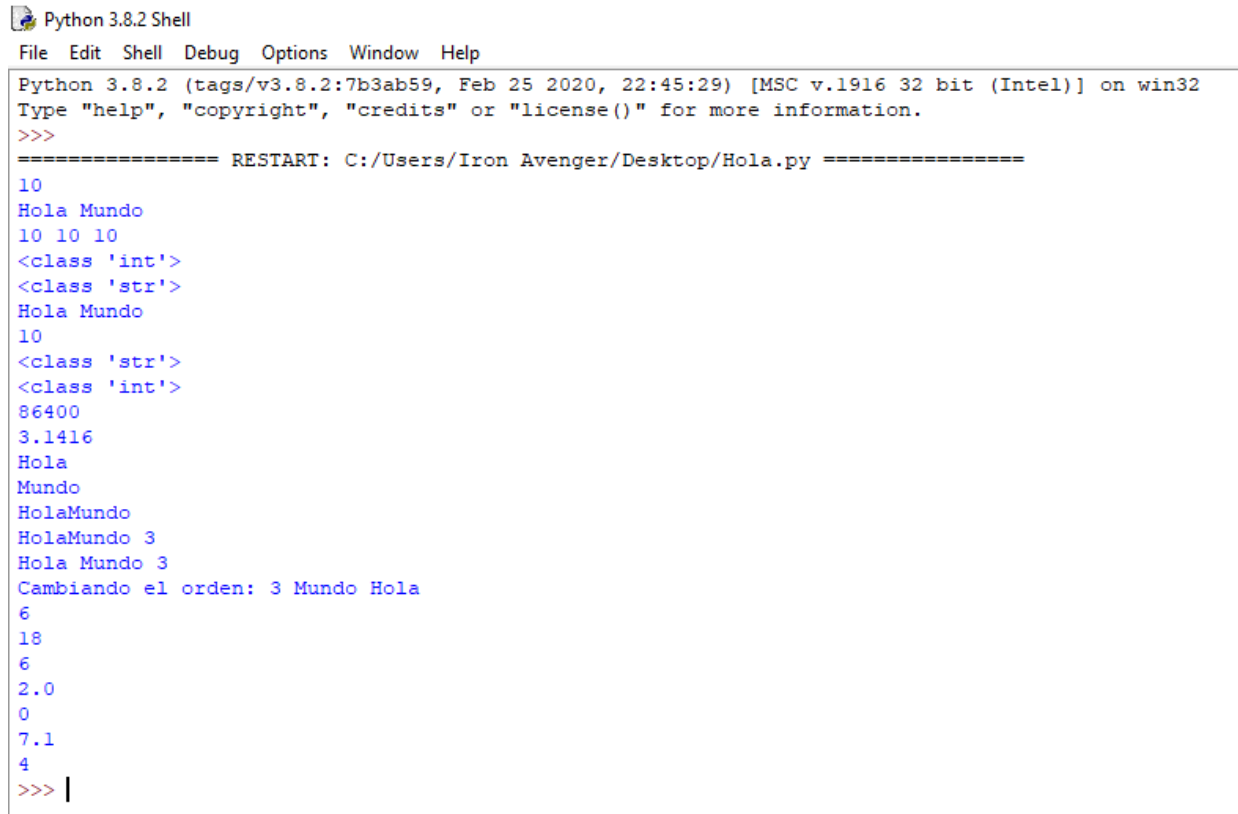
val4, _, val5=varios(2)
print("{} {} {}".format(val4, val5))

```

```

vg="Global"
def funcion_v1():
    print(vg)
    funcion_v1()
    print(vg)
    def funcion_v2():
        vg="Local"
        print(vg)
        funcion_v2()
        print(vg)
        def funcion_v3():
            print(vg)
            vg="Local"
            print(vg)
            funcion_v3()
            def funcion_v4():
                global vg
                print(vg)
                vg="Local"
                print(vg)
                funcion_v4()

```



The screenshot shows a Python 3.8.2 Shell window with a menu bar (File, Edit, Shell, Debug, Options, Window, Help) and a status bar. The main text area displays the output of a script named 'Hola.py'. The output includes the following lines:

```

Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Iron Avenger/Desktop/Hola.py =====
10
Hola Mundo
10 10 10
<class 'int'>
<class 'str'>
Hola Mundo
10
<class 'str'>
<class 'int'>
86400
3.1416
Hola
Mundo
HolaMundo
HolaMundo 3
Hola Mundo 3
Cambiando el orden: 3 Mundo Hola
6
18
6
2.0
0
7.1
4
>>> |

```

```

HolaMundo
HolaMundo 3
Hola Mundo 3
Cambiando el orden: 3 Mundo Hola
6
18
6
2.0
0
7.1
4
False
True
True
False
True
[31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
31
31
31
[['cero', 0], ['uno', 1, 'UNO'], ['dos', 2], ['tres', 3], ['cuatro', 4], ['x', 5]]
['cero', 0]
['uno', 1, 'UNO']
dos
2
uno
1
UNO
['cinco', 5]
(31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31)
31
30
28
(('cero', 0), ('uno', 1, 'UNO'), ('dos', 2), ('tres', 3), ('cuatro', 4), ('x', 5))
('cero', 0)
('uno', 1, 'UNO')
dos
2
uno
1
UNO

```

```

*
uno
1
UNO
['cinco', 5]
(31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31)
31
30
28
(('cero', 0), ('uno', 1, 'UNO'), ('dos', 2), ('tres', 3), ('cuatro', 4), ('x', 5))
('cero', 0)
('uno', 1, 'UNO')
dos
2
uno
1
UNO
{'hidrogeno': 1, 'Helio': 2, 'Carbon': 6}
1
{'hidrogeno': 1, 'Helio': 2, 'Carbon': 6, 'litio': 3, 'nitrogeno': 8}
{'H': {'name': 'Hydrogen', 'number': 1, 'weight': 1.000794}, 'He': {'name': 'Helium', 'number': 2, 'weight': 4.002602}}
{'name': 'Hydrogen', 'number': 1, 'weight': 1.000794}
Hydrogen
1
4.3
{'name': 'Hydrogen', 'number': 1, 'weight': 4.3, 'gas noble': True}
dict_items([('H', {'name': 'Hydrogen', 'number': 1, 'weight': 4.3, 'gas noble': True}), ('He', {'name': 'Helium', 'number': 2, 'weight': 4.002602})])
dict_keys(['H', 'He'])
holaJJ

```



## b) Ejercicio 2

### #Triangulo

#Area

b=5

h=7

area=((b\*h)/2)

print("El area del triangulo es: "+str(area))

#Perimetro

Lu=3

Ld=6

p=b+Lu+Ld

print("El perimetro del triangulo de lados {}, {}, {} es: {}".format(b, Lu, Ld, p))

#Circulo

#Area

PI=3.1416

r=3

area=PI\*(r\*\*2)

print("El area del circulo es: "+str(area))

#Perimetro

p=PI\*(r\*2)

print("EL perimetro del circulo es: "+str(p))

#Rectangulo

#Area

ba=5

ha=10

area=b\*h

print("El area del rectangulo de base {} y altura {} es: {}".format(ba, ha, area))

#Perimetro

p=((ba\*2)+(ha\*2))

print("EL perimetro es: "+str(p))

#Trapezio

#Area

x=5

y=2

z=7

area=(z\*((x+y)/2))

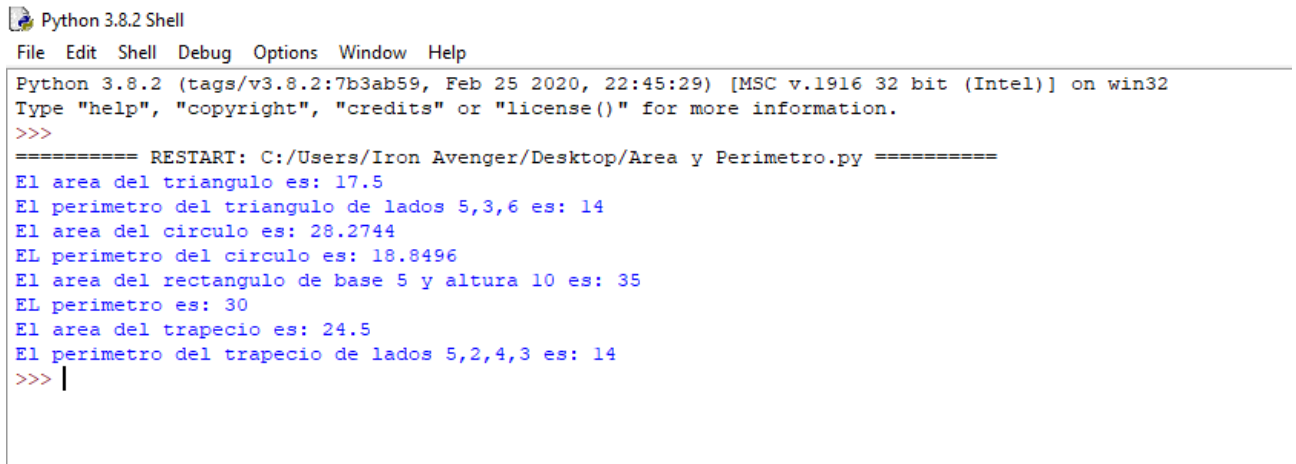
print("El area del trapezio es: "+str(area))

#Perimetro

e=4

f=3

```
p=(x+y+e+f)
print("El perimetro del trapecio de lados {}, {}, {}, {} es: {}".format(x,y,e,f,p))
```



The screenshot shows a Python 3.8.2 Shell window with a menu bar (File, Edit, Shell, Debug, Options, Window, Help). The command prompt displays the Python version and build information. The user enters a command to run a script, and the output shows the results of calculations for the area and perimeter of various shapes: a triangle, a circle, a rectangle, and a trapezoid. The script is identified as 'Area y Perimetro.py'.

```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Iron Avenger/Desktop/Area y Perimetro.py =====
El area del triangulo es: 17.5
El perimetro del triangulo de lados 5,3,6 es: 14
El area del circulo es: 28.2744
El perimetro del circulo es: 18.8496
El area del rectangulo de base 5 y altura 10 es: 35
El perimetro es: 30
El area del trapecio es: 24.5
El perimetro del trapecio de lados 5,2,4,3 es: 14
>>> |
```

## Conclusiones

Python es solo otro lenguaje de programación del cual se requiere conocer la sintaxis para su correcto funcionamiento. Además de identificar todas las palabras reservadas que posee, además de saber identificar y declarar el tipo de dato que queremos dentro de nuestro código y mandar a imprimir en pantalla.

## Bibliografía

Laboratorios A y B, Practica 9 introducción a Python, consultado el 31 marzo 2020, de file: <http://lcp02.fi-b.unam.mx/>

Python, About Python, consultado el 31 de marzo de 2020, de <https://www.python.org/about/>