



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación  
Salas A y B

*Profesor:* Martínez Quintana Marco Antonio

*Asignatura:* Estructura de Datos y Algoritmos 1

*Grupo:* 17

*No de Práctica(s):* 9

*Integrante(s):* Ruiz Godoy Franco

*No. de Equipo de  
cómputo empleado:* 7

*No. de Lista o Brigada:* 34

*Semestre:* 2020-2

*Fecha de entrega:* 31 de Marzo de 2020

*Observaciones:*

CALIFICACIÓN:

## **Objetivo.**

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

## **Introducción.**

Python se define como un lenguaje de programación interpretado en donde la filosofía de trabajo hace énfasis en que la sintaxis siempre asegure que el código pueda ser legible. Es así mismo un lenguaje de programación multiparadigma, que puede soportar la orientación a objetos, una programación imperativa y en una medida menor la programación funcional.

Destaca por la versatilidad del lenguaje, sus plantillas, módulos, paquetes, frameworks, bibliotecas, sistemas de gestión y más. Se lo puede utilizar para:

- Cálculos científicos o de ingeniería.
- Desarrollo web.
- Videojuegos o similares.
- Programas gráficos.

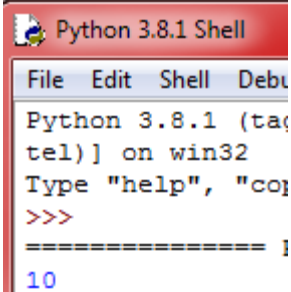
Son varios los estilos que se pueden implementar:

- Programación imperativa.
- Programación orientada a objetos.
- Programación funcional.
- Otros paradigmas al incluir extensiones.

## Desarrollo.

### Declarando Variables

```
x = 10
print(x)
```

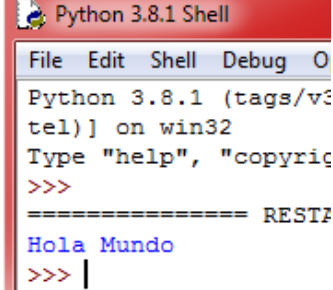


Python 3.8.1 Shell

File Edit Shell Debug

Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MS tel)] on win32  
Type "help", "copyright", "credits" or "license()" for more  
>>>  
10

```
cadena="Hola Mundo"
print(cadena)
```



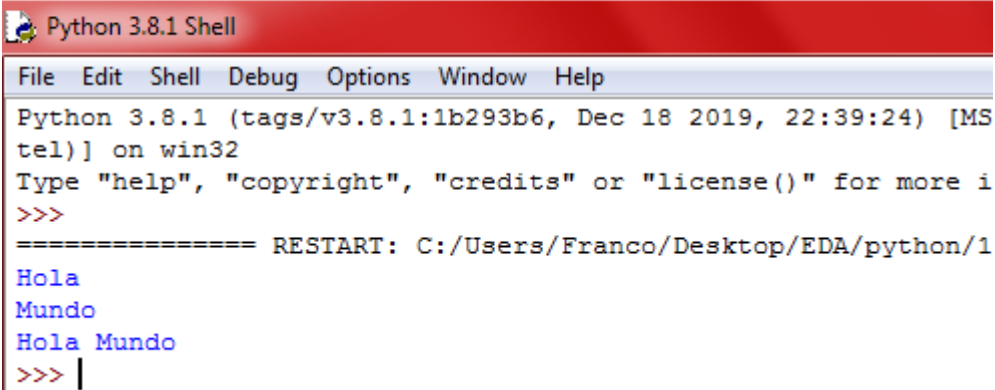
Python 3.8.1 Shell

File Edit Shell Debug Options Window Help

Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MS tel)] on win32  
Type "help", "copyright", "credits" or "license()" for more  
>>>  
Hola Mundo  
>>> |

### Declarando cadenas.

```
cadena1 = 'Hola '
cadena2 = "Mundo"
print(cadena1)
print(cadena2)
concat_cadenas = cadena1 + cadena2 #Concatenación de cadenas
print(concat_cadenas)
```



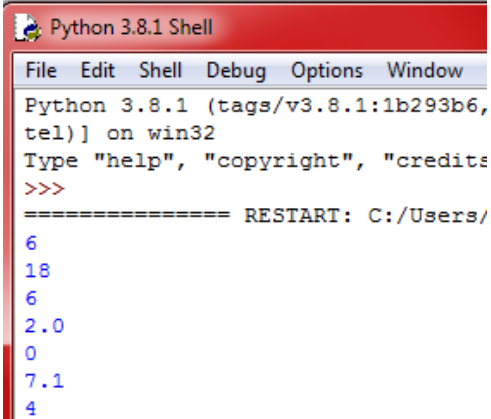
Python 3.8.1 Shell

File Edit Shell Debug Options Window Help

Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MS tel)] on win32  
Type "help", "copyright", "credits" or "license()" for more  
>>>  
===== RESTART: C:/Users/Franco/Desktop/EDA/python/1  
Hola  
Mundo  
Hola Mundo  
>>> |

## Operadores.

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

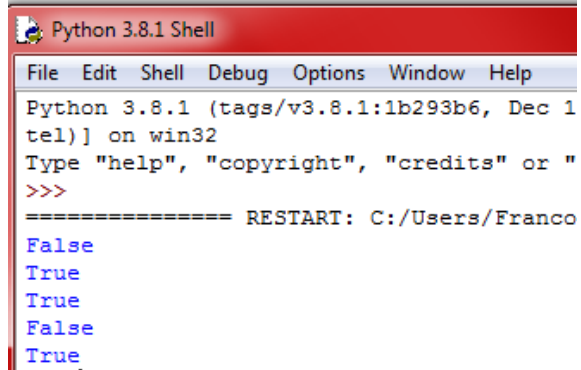


Python 3.8.1 Shell

File Edit Shell Debug Options Window

Python 3.8.1 (tags/v3.8.1:1b293b6, tel)] on win32  
Type "help", "copyright", "credits" or "restart()" for more  
>>>  
===== RESTART: C:/Users/  
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
```



Python 3.8.1 Shell

File Edit Shell Debug Options Window Help

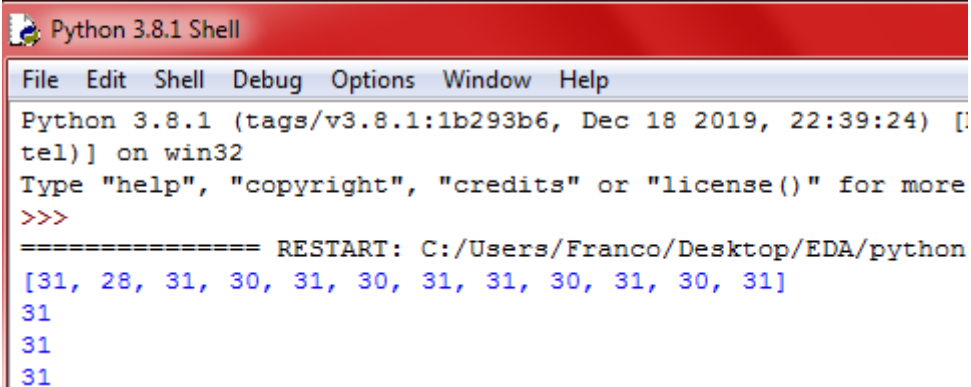
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 1 tel)] on win32  
Type "help", "copyright", "credits" or "restart()" for more  
>>>  
===== RESTART: C:/Users/Franco  
False  
True  
True  
False  
True

## Listas, tuplas y diccionarios.

### Lista simple.

```
lista_diasDelMes=[31,28,31,30,31,30,31,31,30,31,30,31]
```

```
print( lista_diasDelMes ) #imprimir la lista completa
print( lista_diasDelMes[0] ) #imprimir elemento 1
print( lista_diasDelMes[6] ) #imprimir elemento 7
print( lista_diasDelMes[11] ) #imprimir elemento 12
```



Python 3.8.1 Shell

File Edit Shell Debug Options Window Help

Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [ tel)] on win32  
Type "help", "copyright", "credits" or "license()" for more  
>>>  
===== RESTART: C:/Users/Franco/Desktop/EDA/python  
[31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]  
31  
31  
31

## Lista anidada.

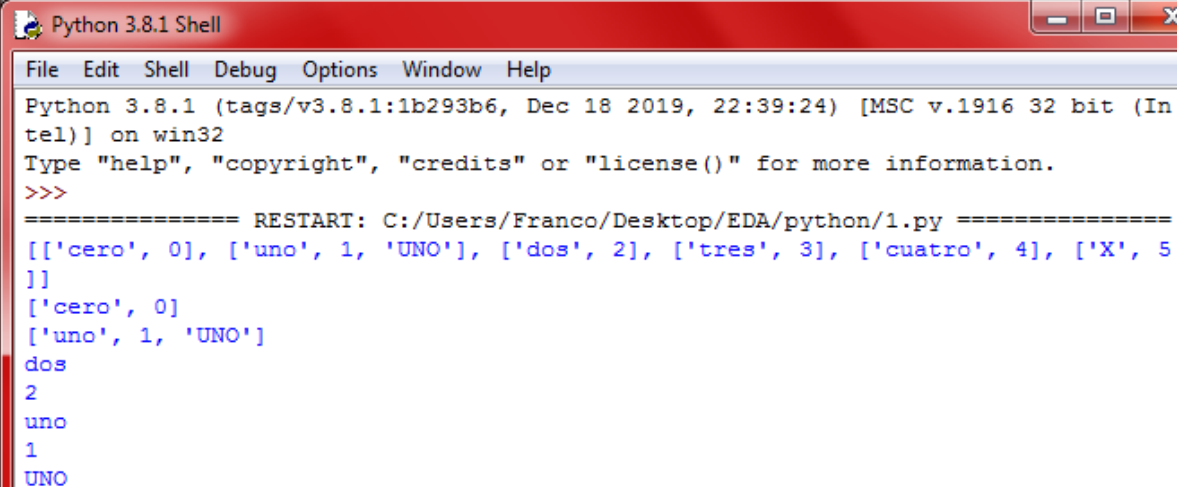
```
lista_numeros=[['cero', 0],['uno',1, 'UNO'], ['dos',2], ['tres', 3], ['cuatro',4]

print (lista_numeros)          #imprimir lista completa

print (lista_numeros[0])       #imprime el elemento 0 de la lista
print (lista_numeros[1])       #imprime el elemento 1 de la lista

print (lista_numeros[2][0])    #imprime el primer elemento de la lista en la posici
print (lista_numeros[2][1])    #imprime el segundo elemento de la lista en la posic

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



Python 3.8.1 Shell

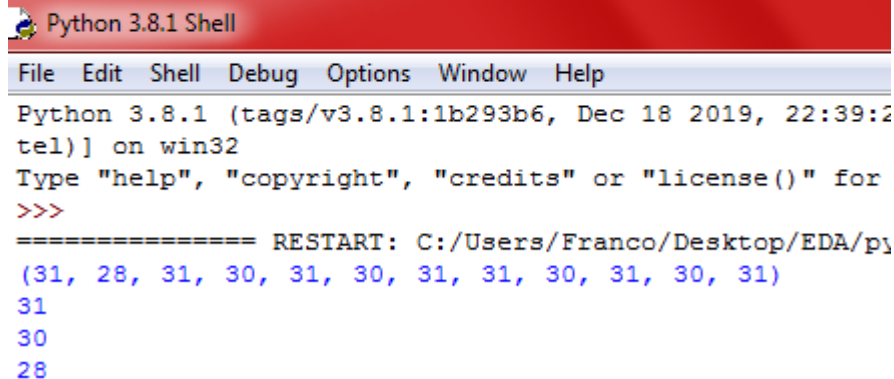
File Edit Shell Debug Options Window Help

Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: C:/Users/Franco/Desktop/EDA/python/1.py =====  
[['cero', 0], ['uno', 1, 'UNO'], ['dos', 2], ['tres', 3], ['cuatro', 4], ['X', 5]]  
[['cero', 0]  
['uno', 1, 'UNO']  
dos  
2  
uno  
1  
UNO

## Tupla

```
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])
```



Python 3.8.1 Shell

File Edit Shell Debug Options Window Help

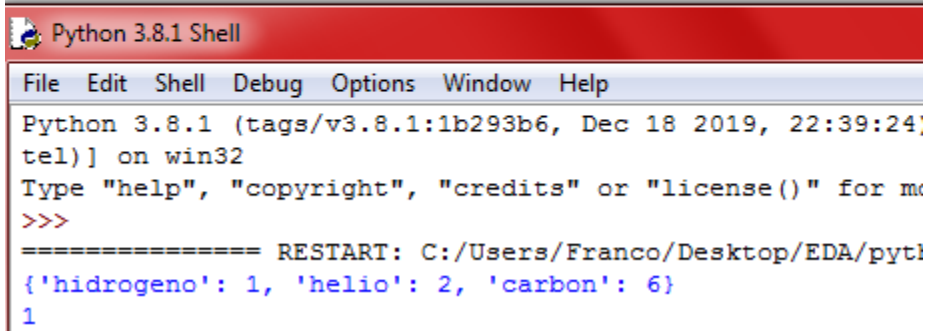
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: C:/Users/Franco/Desktop/EDA/python/1.py =====  
(31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31)  
31  
30  
28

## Diccionarios.

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

print (elementos)

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



The screenshot shows a Python 3.8.1 Shell window with a red title bar. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The command prompt shows the Python version and build information, followed by the user's input to print the dictionary 'elementos'. The output displays the dictionary contents: {'hidrogeno': 1, 'helio': 2, 'carbon': 6}.

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24;
tel)] on win32
Type "help", "copyright", "credits" or "license()" for m
>>>
===== RESTART: C:/Users/Franco/Desktop/EDA/pytl
{'hidrogeno': 1, 'helio': 2, 'carbon': 6}
1
```

## (Extra) Calculo de áreas y perímetros.

```
print("*****")
#Calculo del area y perimetro de un rectangulo.
x = 65 #Base
y = 30 #Altura

area_r = x*y
perimetro_r = 2*x+2*y

print ("El area del rectangulo es: "+str(area_r))
print ("El perimetro del rectangulo es: "+str(perimetro_r))

print("*****")

#Calculo del area y perimetro del triangulo isosceles.

a = 10 #base
b = 25 #Altura
i = 12 #lado 1
j = 12 #lado 2

area_t = (a*b)/2
perimetro_t = i+j+a

print ("El area del triangulo isosceles es: "+str(area_t))
print ("El perimetro del triangulo isosceles es: "+str(perimetro_t))

print("*****")

#Calculo del area y perimetro del circulo.
r = 38 #Radio
pi = 3.1415926535897931

area_c = pi*(r**2)
perimetro_c = (2*pi)*r

print ("El area del circulo es: "+str(area_c))
print ("El perimetro del circulo es: "+str(perimetro_c))
```

```

print("*****")

#Calculo del area y perimetro de un trapecio.

w = 3 #base
W = 9 #BASE
h = 5 #Altura
o = 6 #Lado1
p = 6 #lado2

area_tra = ((w+W)*h)/2
perimetro_tra = (2*o)+w+W

print ("El area del trapecio es: "+str(area_tra))
print ("El perimetro del trapecio es: "+str(perimetro_tra))

print("*****")

```

## Resultado.

```

*****
El area del rectangulo es: 1950
El perimetro del rectangulo es: 190
*****
El area del triangulo isosceles es: 125.0
El perimetro del triangulo isosceles es: 34
*****
El area del circulo es: 4536.459791783661
El perimetro del circulo es: 238.76104167282426
*****
El area del trapecio es: 30.0
El perimetro del trapecio es: 24
*****

```

## Conclusiones.

Sin duda el lenguaje de programación de Python en comparación a otros lenguajes como lo es C, python tiene más sencillez en toda la “sintaxis”, es decir, de fácil uso. El lenguaje cuenta con ventajas de facilidad de escritura de código en diferentes dispositivos y una gran abundancia de bibliotecas. Además de que este ofrece el poder utilizarlo para la creación de Cálculos científicos o de ingeniería, desarrollo web, videojuegos, programas gráficos, entre otros.

## Referencias.

García Cano, E. E. G. C., & Solano Gálvez, J. A. S. G. (2019, 20 enero). Guía práctica de estudio 09: Introducción a Python (I).. Recuperado 31 marzo, 2020, de [http://lcp02.fi-b.unam.mx/static/docs/PRACTICAS\\_EDA1/eda1\\_p9.pdf](http://lcp02.fi-b.unam.mx/static/docs/PRACTICAS_EDA1/eda1_p9.pdf)

R, J. L. R. (2019, 22 octubre). PYTHON (informática) | Qué es, para qué sirve y características. Recuperado 1 abril, 2020, de <https://247tecno.com/python-para-que-sirve-por-que-usarlo/>