



## 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): 10*

*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: 14 abril 2020*

*Observaciones:*

CALIFICACIÓN: \_\_\_\_\_

# Objetivo

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

## Introducción

En esta práctica se continuará con el aprendizaje para poder utilizar el lenguaje de programación Python. Ahora se abordará las estructuras de repetición o ciclos comunes como for y while, las cuales funcionan de igual manera que en otros lenguajes solo que la sintaxis cambia para poder llevar a cabo las operaciones.

También se podrá observar estructuras de selección como lo es if-elif-else una variante del switch, e if else.

Por último conoceremos algunas de las bibliotecas más útiles y comunes en este lenguaje de programación que nos permitirá hacer múltiples operaciones, por ejemplo una de ella es la graficación de datos.

# Desarrollo

*a) código*

```
#if
def obtenerMayor(param1,param2):
    if param1<param2:
        print("{} es mayor que {}".format(param2,param1))
```

```
obtenerMayor(5,7)
```

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

```
#if-else
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)))
```

```
def obtenermayor_idiom(param1,param2):
    valor=param2 if (param1<param2)else param1
    return valor
print("El mayor es {}".format(obtenermayor_idiom(11,6)))
```

```
#if-elif-else
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)
```

```
def numeros_idiom(num):  
    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)
```

```
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)))
```

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

```
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))
```

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

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

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

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

```
#diccionario
elementos={"hidrogeno":1, "Helio":2,"Carbon":6}
for llave,valor in elementos.items():
    print(llave, "=", valor)
```

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

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

```
for idx, x in enumerate(elementos):
    print("El indice es: {} y el elemento: {}".format(idx,x))
```

```
def cuenta_idiom(limite):
    for i in range(limite,0,-1):
        print(i)
    else:
        print("Cuenta finalizada")
cuenta_idiom(5)
```

```
def cuenta_idiomv2(limite):
    for i in range(limite,0,-1):
        print(i)
```

```
    if i==3:
        break
    else:
        print("Cuenta finalizada")
cuenta_idiomv2(5)
```

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

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

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

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

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

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

```
7 es mayor que 5
True
El mayor es 20
El mayor es 11
El mayor es 11
tu numero es 2
no hay opcion
tu numero es 2
5 no es una opcion
El mas grande es 13
10
9
8
7
6
5
4
3
2
1
24
720
1
2
3
4
5
0
1
2
3
4
-5
-4
-3
-2
-1
0
1
uno
dos
---
```

Python 3.8.2 Shell

File Edit Shell Debug Options Window Help

```
uno
dos
tres
cuatro
hidrogeno = 1
Helio = 2
Carbon = 6
hidrogeno
Helio
Carbon
1
2
6
El indice es: 0 y el elemento: hidrogeno
El indice es: 1 y el elemento: Helio
El indice es: 2 y el elemento: Carbon
5
4
3
2
1
Cuenta finalizada
5
4
3
-1.0
-1.0
-1.0
['__doc__', '__loader__', '__name__', '__package__', '__spec__', 'acos', 'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'comb', 'copysign', 'cos', 'cosh',
'degrees', 'dist', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf', 'isclose', 'isfinite', 'i
sinf', 'isnan', 'isqrt', 'ldexp', 'lgamma', 'log', 'log10', 'logip', 'log2', 'modf', 'nan', 'perm', 'pi', 'pow', 'prod', 'radians', 'remainder', 'sin', 'sinh', 'sqrt',
'tan', 'tanh', 'tau', 'trunc']
Help on built-in function log in module math:

log(...)
    log(x, [base=math.e])
    Return the logarithm of x to the given base.

    If the base not specified, returns the natural logarithm (base e) of x.

-1.0
```

## Conclusiones

Como podemos observar, este lenguaje es fácil de manejar, solo varia en la sintaxis que se debe emplear para cada función lo que la vuelve un poco más compleja a comparación de otros lenguajes. Sin embargo, su extensa librería de bibliotecas nos permite realizar varias operaciones o funciones que no encontramos en otros lenguajes lo que lo hace un lenguaje muy completo.

## Bibliografía

Laboratorios A y B, Practica 10 introducción a Python II, consultado el 14 de abril 2020, de file: <http://lcp02.fi-b.unam.mx/>