



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Ing. Marco Antonio Martínez Quintana

Asignatura: Estructura de datos y algoritmos I

Grupo: 17

No de Práctica(s): 10

Inteарante(s): Vega Gutierrez Ricardo Daniel

*No. de Equipo de
cómputo empleado:* 9

No. de Lista o 40

Semestre: 2020-2

Fecha de entrega: 21 de abril del 2020

Observaciones:

CALIFICACIÓN: _____

Objetivo

Aplicar las bases del lenguaje Python en el editor de texto (Notepad++).

Introducción

Python es un lenguaje de alto nivel (LAN), el cual tiene grandes aplicaciones, para usar Python se necesita usar una sintaxis especial, como se muestra a continuación:

- El código se tiene que indentar.
- Las sentencias de código no llevan ;
- Posee palabras reservadas (chechar practica 9).
- No es necesario declarar las variables ni su tipo

Posee distintas estructuras de selección y cuenta con una variedad de librerías, dependiendo la librería que ocupes es lo que podrás realizar en el código.

Desarrollo

El ciclo if es una estructura de selección que permite realizar una acción dependiendo si la sentencia lógica es verdadera. El ejemplo muestra una comparación de dos números y solo imprime cuando el primer numero (n1) es mayor.

```
#if
def mayor(n1, n2):
    if n1 > n2:
        print("{} es mayor que {}".format(n1, n2))
        print(str(n1) + " es mayor que " + str(n2))
mayor(15, 10)
```

```
15 es mayor que 10
15 es mayor que 10
```

El ciclo if-else funciona similar al if, la diferencia esta en: se encuentra otro caso (else), que se cumple siempre y cuando la sentencia lógica sea falsa.

```
#if-else
def mayor_1_2(n1, n2):
    if n1 > n2:
        print("El mayor es: {}".format(n1))
    else:
        print("El numero mayor es " + str(n2))
mayor_1_2(60, 4)
```

```
El mayor es: 60
```

El ciclo if-elif-else funciona con múltiples condiciones, solo se ejecuta una parte del código según corresponda a la sentencia lógica que resulto verdadera. Funciona como la sentencia «switch», que dependiendo un valor realiza ciertas sentencias.

```
#if-elif-else
def mayNum(num):
    if (num%2) == 0:
        print("{} es multiplo de 2" .format(num))
    elif (num%3) == 0:
        print(" {} es multiplo de 3" .format(num))
    elif (num%5) == 0:
        print(" {} es multiplo de 5" .format(num))
    else:
        print("Opción invalida")
mayNum(50)
```

```
50 es multiplo de 2
```

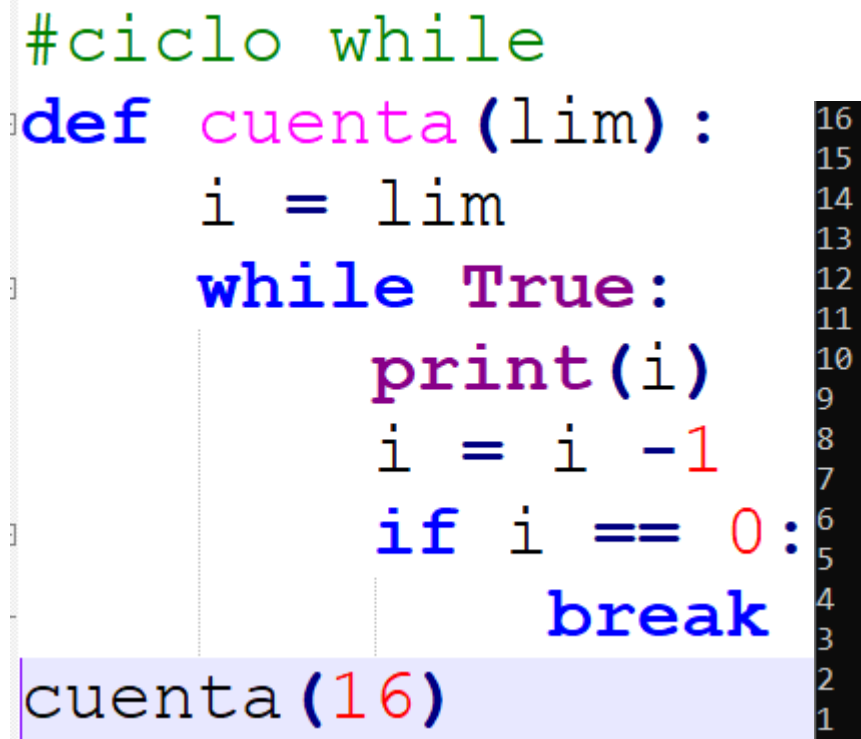
La estructura anidada usa los if, para comparar el valor de tres números y escribir por pantalla al mayor de ellos.

```
#estructura anidada
def mayor_tres(n1, n2, n3):
    if n1>n2:
        if n1>n3:
            print("El numero mayor es: " +str(n1))
        else:
            print("El numero mayor es: "+str(n3))
    elif n2>n3:
        print("El numero mayor es: "+str(n2))
    else:
        print("El mayor es: "+str(n3))
mayor_tres(10, 50, 200)
```

```
El mayor es: 200
```

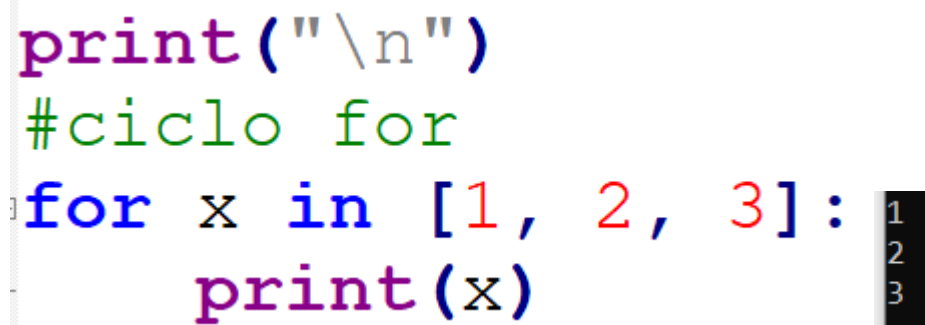
El ciclo «while» se manejó siempre que fuera verdad, el usuario pone el limite y hasta que el numero no sea 0, se escribe el número, cuando el numero es 0, la función corta el proceso y ahí termina de escribir.

```
#ciclo while
def cuenta(lim):
    i = lim
    while True:
        print(i)
        i = i - 1
        if i == 0:
            break
cuenta(16)
```



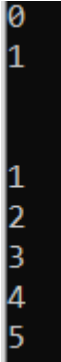
El ciclo for, funciona imprimiendo por índices, imprime lo que posee el índice 0.....hasta terminar con el último elemento.

```
print("\n")
#ciclo for
for x in [1, 2, 3]:
    print(x)
```



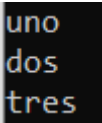
El ciclo «for» con rangos, te permite poner un rango inicial -no importa si es negativo- y un rango final, el cual va a manejar en el tramo de los rangos puestos.

```
print("\n")
#for con rangos
for x in range (2):
    print(x)
print("\n")
for x in range (1, 6):
    print(x)
```



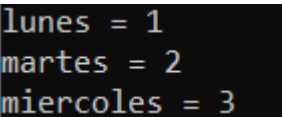
El ciclo «for» también acepta recorrer elementos distintos a números y por lo tal también los puede imprimir.

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



Como se observa el ciclo «for» también funciona en diccionarios, es capaz de imprimir la definición del elemento y su valor.

```
print("\n")
#diccionarios
dias = {"lunes": 1, "martes":2, "miercoles":3}
for llave, valor in dias.items():
    print(llave, "=", valor)
```



Se importo la librería «math», para poder trabajar con funciones trigonométricas y obtener un valor al meterle un parámetro a la función.

```
#math
y = math.cos(math.pi)
print(y)
```

```
-1.0
```

Bibliotecas

NumPy es una biblioteca que Python brinda para manejar operaciones matemáticas, por ejemplo: vectores, matrices, funciones de algebra lineal y se puede integrar con Fortran, C y C++

Scipy es una biblioteca que usa Numpy con el propósito de realizar operaciones más avanzadas.

Matplotlib dicha biblioteca permite hacer uso de un formato 2D y 3D, tiene funciones -se pueden modificar- para generar gráficas y tiene puente con LaTeX por si se desea nombrar las gráficas con lenguaje matemático.

Conclusiones

Python es un lenguaje muy poderoso el cual es capaz de poder importar distintas librerías muy útiles, algunas para cálculos muy avanzados (probabilidad, eficiencia) y otras nos permiten graficar en 2D y 3D, lo cual es muy interesante, tanto que hasta tu puedes manipular las herramientas que necesites para la gráfica. El poder de Python es inmenso conforme lo voy conociendo me agrada más, no tiene nada que pedirles a otros lenguajes.

Bibliografía

Tutorial oficial de Python: <https://docs.python.org/3/tutorial/>

Galería de notebooks: <https://wakari.io/gallery>

Matplotlib: <http://matplotlib.org/>