



Caratula para entrega de Prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: _____ Ing. Marco Antonio Martinez Quintana _____

Asignatura: __ Estructura de Datos y Algoritmos I (1227) __

Grupo: _____ 17 _____

No. de Práctica(s): _____ 10° _____

Integrante(s): _____ Avila Laguna Ricardo _____

No. de Equipo de
cómputo empleado: _____ 10 _____

No. Lista o Brigada: _____ 6 _____

Semestre: _____ 2° _____

Fecha de entrega: _____ Abril del 2020 _____

Observaciones: _____

CALIFICACIÓN: _____

1° Objetivos

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

2° Introducción

En esta práctica se iniciara un repaso del lenguaje de programación Python, por lo que veremos algunos temas como estructuras de control selectivas, estructuras de control repetitivas e iteración de diccionarios.

3° Desarrollo y Resultados

Actividades:

- **Aplicar estructuras de control selectivas**
- **Aplicar estructuras de control repetitivas**
- **Usar las bibliotecas estándar**
- **Generar una gráfica**
- **Ejecutar un programa desde la ventana de comandos**
- **Pedir datos al usuario al momento de ejecutar un programa**

Código:

```
Practica10.py - C:/Users/Marbella/Documents/ProjetsPython/Practica10.py (3.8.2)
File Edit Format Run Options Window Help

#Estructuras de control selectivas
#If
def obtenerMayor (parametro1,parametro2):
    if parametro1 < parametro2:
        print('{1} es mayor que {0} '.format(parametro1,parametro2))
    return 0
obtenerMayor(5,7)

#Se puede dar mas de un parametro de condicion, lo unico es que tenemos que utilizar
a=b=c=10
if a == b == c:
    print(True)
print('\n')

#If-else
def obtenerMayor2 (parametro1,parametro2):
    if parametro1 < parametro2:
        return parametro2
    else:
        return parametro1
print('El mayor es {}'.format(obtenerMayor2(4,20)))
print('El mayor es {}'.format(obtenerMayor2(11,6)))

#Para if-else de una linea de codigo
def obtenerMayor_idiom(parametro1,parametro2):
    valor = parametro2 if(parametro1 < parametro2)else parametro1
    #Valor = parametro2 si parametro1 es menor a parametro2 sino se cumple sera parametro1
    return valor
print('El numero mas grande es {}'.format(obtenerMayor_idiom(6,7)))
print('El numero mas grande es {}'.format(obtenerMayor_idiom(8,7)))
print('\n')

#If-elif-else es parecido al switch
#If seria como el switch, elif seria case y else seria default
def numeros(num):
    if num == '1':
        print('Tu numero es {}'.format(num))
    elif num == '2':
        print('Tu numero es {}'.format(num))
    elif num == '3':
        print('Tu numero es {}'.format(num))
    elif num == '4':
        print('Tu numero es {}'.format(num))
    else:
        print('No es un numero valido')
```

Ln: 209 Col: 57

En esta práctica revisamos las estructuras de control selectivas conformadas por if, el cual sirve para decir si algo es verdadero o no para luego decidir si realizar algunas lineas de codigo, else sirve para realizar instrucciones tal si la condición no se cumple y por último elif que sirve como un switch para la elección de varios casos dentro de un if.

```
Practica10.py - C:/Users/Marbella/Documents/ProjetsPython/Practica10.py (3.8.2)
File Edit Format Run Options Window Help
#Estructura de control selectiva anidada
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("INGRESA TRES NUMEROS")
a = int(input('Opcion:'))
b = int(input('Opcion:'))
c = int(input('Opcion:'))
#a = input()
#b = input()
#c = input()
print("El mas grande es {}".format(obtenerMasGrande(a,b,c)))
print('\n')

#CICLOS
#While
#Ejemplo 1
def cuenta(limite):
    i = limite
    while True:
        print(i)
        i = i - 1
        if i == 0:
            break #Rompiendo el ciclo
    return 0
cuenta(10)
#Ejemplo 2
def factorial(n):
    a = 2
    tmp = 1
    while a < n+1:
        tmp = tmp * a
        a = a + 1
    return tmp
```

Al igual revisamos las estructuras de control selectivas anidadas donde dentro de una función introducimos un if- else de selección y dentro de cada de uno de estos tenemos otro if para que el paso de información se simplifique, al igual en esta parte iniciamos con lo ciclos.

```
Practica10.py - C:/Users/Marbella/Documents/ProjetsPython/Practica10.py (3.8.2)
File Edit Format Run Options Window Help

#CICLOS
#While
#Ejemplo 1
def cuenta(limite):
    i = limite
    while True:
        print(i)
        i = i - 1
        if i == 0:
            break #Rompiendo el ciclo
    return 0
cuenta(10)
#Ejemplo 2
def factorial(n):
    a = 2
    tmp = 1
    while a < n+1:
        tmp = tmp * a
        a = a + 1
    return tmp
print("El factorial es: {}".format(factorial(5)))
print('\n')

#For
#Ejemplo 1
for x in [1,2,3,4,5]:
    print(x)
#Ejemplo 3
#Range sirve para generar una lista
for x in range(5): #En este caso es de (0,4)
    print(x)
#Ejemplo 2
#Tambien se puede para numeros negativos
for x in range(-5,2): #En este caso es de (-5,1)
    print(x)
#Ejemplo 4
for num in ['uno','dos','tres','cuatro']:
    print(num)
print('\n')

#Iteración en diccionarios
#Creamos un diccionario
elementos = {'hidrogeno':1,'helio':2,'carbono':6}
```

Ln: 209 Col: 57

Para la tercera parte del programa estudiamos a fondo los ciclos, los más convenientes para utilizar en Python son el while y el for pero mas recomendable el for, cada uno de ellos estudiamos sus declaraciones y a funcionalidad que le puede dar a un programa..

```
Practica10.py - C:/Users/Marbella/Documents/ProjetsPython/Practica10.py (3.8.2)
File Edit Format Run Options Window Help

#BIBLIOTECAS o MODULOS
#Como utilizar e importar una biblioteca
import math #importamos toda la biblioteca
x = math.cos(math.pi)
print(x)
#Como importar solo las funciones de una biblioteca sin utilizar su prefijo
from math import * #Importamos todas las funciones
x = cos(pi)
print(x)
#Importar solo las funciones necesarias
from math import cos,pi #Importamos las funciones que vamos a utilizar
x = cos(pi)
print(x)
#Despues de importar una biblioteca podemos conocer las funciones que contiene
print(dir(math)) #print(dir(<Biblioteca>))
#Para conocer como utilizar las funciones
help(math.log)
#Se puede definir un ALIAS para llamar a las funciones de la biblioteca
import math as ma #A la biblioteca "math" le ponemos un alias "ma"
x = ma.cos(ma.pi)
print(x)
print('\n')

#GRAFICACIÓN
#Esta linea se ocupa para que las graficas que se generen queden embebidas dentro
#%pylab inline
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
#Datos de entrada
x = linspace(0,5,20) #Generando 20 puntos de grafica entre 0 y 5
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') #Guardando la grafica

Ln: 209 Col: 57
```

Por último vimos cómo se invocan las bibliotecas o llamadas también como módulos, las cuales es muy diferente invocar algunas funciones de la biblioteca que invocar toda la biblioteca, también una funcionalidad muy buena es saber cuáles son las funciones y cómo funcionan cada una de ellas solo con la introducción de 2 líneas de código. Como adicional vimos como crear un gráfico que a mi experiencia le falta mucha más información a la página..

4° Conclusiones

Avila Laguna Ricardo :

Los objetivos se cumplieron porque aprendimos como son las estructuras de control selectivas dentro del lenguaje de programación Python, así mismo también vimos estructuras de control anidadas, los diferentes tipos de ciclos(while,for), como invocar bibliotecas y cómo graficar.

Bibliografía

<http://lcp02.fi-b.unam.mx/>