



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Ing. Adrian Ulises Mercado Martinez

Profesor:

Estructura de Datos y Algoritmos I

Asignatura:

13

Grupo:

No de Práctica(s): PRÁCTICA NO. 10

Integrante(s): Hurtado Rodriguez Nestor Rafael

11,

No. de Lista o Brigada:

2020-2

Semestre:

Fecha de entrega:

Observaciones:

CALIFICACIÓN: _____

Objetivo:

Aplicar las bases del lenguaje de programación Python

INTRODUCCIÓN

Python es un lenguaje de programación Interpretado, dinámico y multiplataforma.

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional.



Primero realizamos unos programas para introducirnos al lenguaje Python

HOLAMUNDO

```
#esto es un comentario de una sola línea
"""
Esto es un comentario de varias
líneas
"""

print("Hola Mundo")
x = 10
print(type(x))
print(x)
x = y = z = 2.3
print(x, y, z )
print(type(x))
x = "cadena"
print(x)
print(type(x))
"""
Concatenar 1
"""
c1= "hola"
c2= "Nestor"
saludo = c1 +" "+ c2
```

Primero hicimos el conocido “hola mundo”, para usar las partes básicas del lenguaje Python, y se hizo énfasis en la forma de imprimir los datos .

CALCULADORA

```
def sumar(a,b):  
    c = a+b  
    return c  
  
def restar(a,b):  
    return a -b  
  
def multiplicar(a,b):  
    return a*b  
  
def div_entera(a,b):  
    if b==0:  
        print("error división sobre cero")  
        return  
    return a//b #division entera  
  
def div_entera(a,b):  
    if b==0:  
        print("error división sobre cero")  
        return  
    return a//b  
  
def division(a,b):  
    if b==0:  
        print("error division sobre cero")  
        return  
    return a/b  
  
def modulo(a,b):  
    if b==0:  
        print(" ")  
        return  
    return a%b  
  
def potencia(a,b):  
    return a**b  
  
def main():  
    print("Ingresa dos valores")  
    x = int(input())  
    y = int(input())  
    print("1.- Sumar \n 2.- Restar\n 3.- División entera")  
    print("4.- División\n 5.- Modulo\n 6.- Potencia\n 7.- Multiplicar")  
    op = int(input())  
  
    if op == 1:  
        print(sumar(x,y))  
    elif op ==2:  
        print(restar(x,y))
```

```
if op == 1:
    print(sumar(x,y))
elif op ==2:
    print(restar(x,y))
elif op ==3:
    print(div_entera(x,y))
elif op ==4:
    print(division(x,y))
elif op ==5:
    print(modulo(x,y))
elif op == 6:
    print(potencia(x,y))
elif op == 7:
    print(multiplicar(x,y))
elif op == 8:
    break
else:
    print("opcion no valida")

if __name__ == "__main__":
    main()
```

Se hizo un programa que puede resolver operaciones básicas, aquí aprendimos a introducir valores a operarlos con la estructura de control “IF” “ELIF”. Además de como operar valores y de los dos tipos de divisiones.

ENCONTRAR EL NÚMERO MAYOR

```
def nnumeroMayor(a,b,c):  
    if a>b and a>c:  
        print("El número es {}".format(a))  
    elif(b>c and b>a):  
        print("El número es {}".format(b))  
    else:  
        print("El número es {}".format(c))  
  
if __name__ == "__main__":  
    a= int(input())  
    b= int(input())  
    c= int(input())  
    nnumeroMayor(a,b,c)
```

Dicho programa usa if/elif para encontrar el número mayor de un conjunto de tres.

MATH

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

Nos da el coseno de pi en la variable "x" y después imprime dicha variable.

Estructuras de control selectivas

Este tipo de declaraciones se usan para dar una opción en el caso de que la condición no se cumpla.

FOR EN PYTHON

```
"""
For para listas
"""
def forlist():
    for x in [1,2,3,4,5]:
        print(x)

"""
For para rangos
"""
def forrange():
    for x in range(5):
        print(x)

    for y in range(-3,3):
        print(y)

    for z in range(-4,2,2):
        print(z)

    for i in range(5, 0, -1):
        print(i)

if __name__ == "__main__":
    forlist()
    forrange()

"""
For para diccionarios
"""
def fordic():
    diccionario = {'manzana':1, 'pera':3, 'uva':10}
    for clave, valor in diccionario.items():
        print(clave, " = ", valor)

    for clave in diccionario.keys():
        print(clave)

    for valor in diccionario.values():
        print(valor)

    for idx, x in enumerate(diccionario):
        print("el indice {} del elemento {}".format(idx, x))
```

```
"""
Else de for
"""
def elsefor():
    for x in range(5):
        print(x)
    else:
        print("la cuenta se termino")

def elsefor2():
    for x in range(5):
        print(x)
        if x== 2:
            break
    else:
        print("la cuenta se termino")

if __name__=="__main__":
    forlist()
    forrange()
    elsefor()
    elsefor2()
```

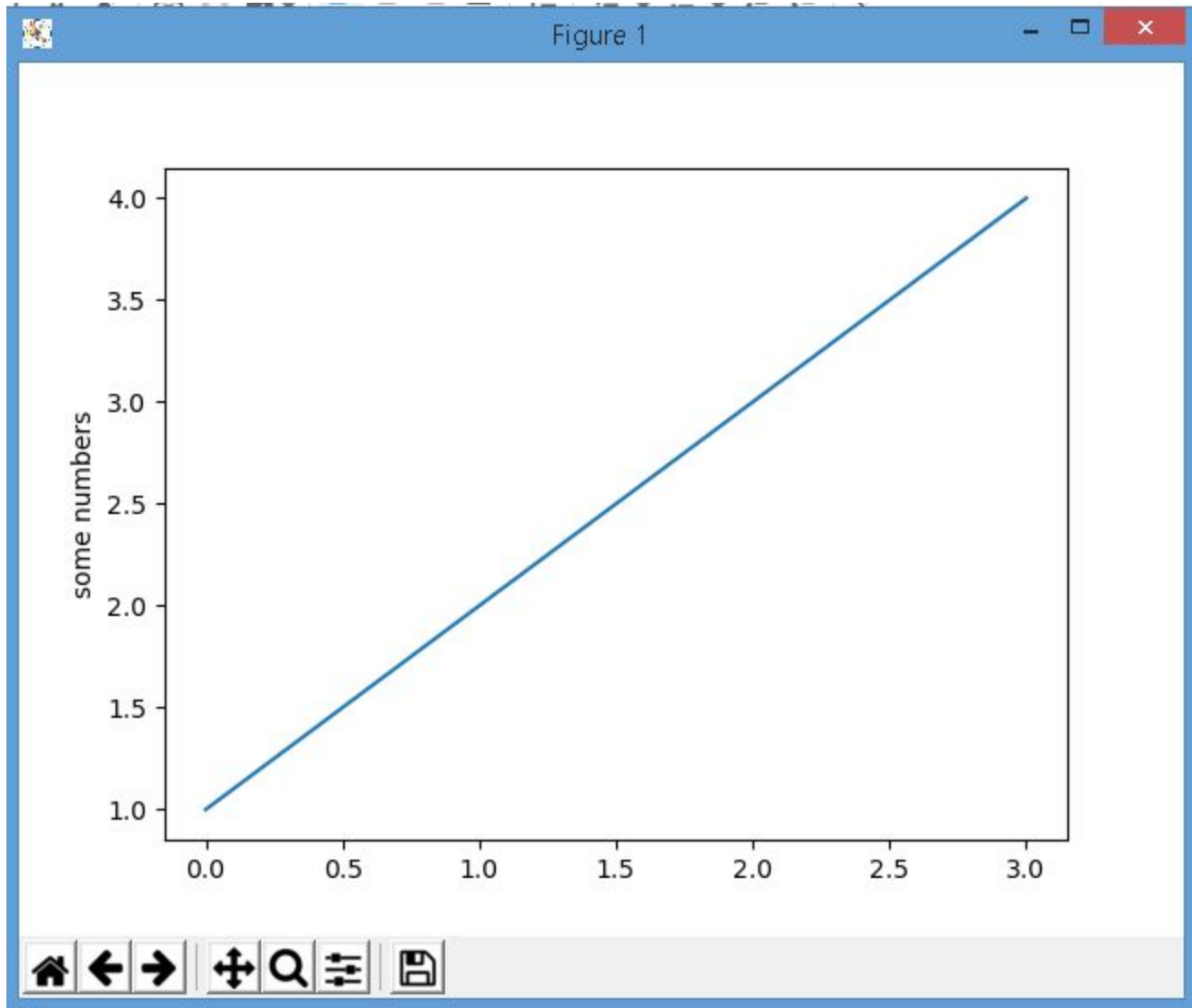
En dicho programa vimos que el for en lenguaje c y en python es parecido pero no es exactamente igual, ya que según la forma en la que utilicemos el for hará cosas distintas. Se usa para:

- Rangos
- Listas
- Para usar diccionarios
- Elsefor

Usar gráficas en python

```
import matplotlib.pyplot as plt
plt.plot([1, 2, 3, 4])
plt.ylabel('some numbers')
plt.show()
```

En este programa le doy unos valores (1,2,3,4) y le pido que al eje de las “y” lo llame “some numbers” e imprime dichos valores.



FACTORIAL

```
def factorial(n):  
    i=2  
    temp = 1  
    while i<=n:  
        temp = temp*i  
        i=i+1  
    return temp  
  
if __name__ == "__main__":  
    a= int(input("Ingresa un número: "))  
    print(factorial(a))
```

Programa para calcular el factorial usando
"while"

CONCLUSIONES

Esta práctica fue una introducción a un nuevo lenguaje de programación, de alto nivel, vimos ciertas semejanzas con el lenguaje que hemos trabajado antes (lenguaje c) y diferencias entre ellos. Además vimos las funciones que nos proporcionan para el uso de python.

REFERENCIAS

https://es.wikipedia.org/wiki/Python#Caracter%C3%ADsticas_y_paradigmas

<https://matplotlib.org/tutorials/introductory/pyplot.html>