

Trabajo Práctico N° 1

Operaciones con variables y gráficos en python. Este cuaderno se encuentra en un [repositorio de github](#)

Ejercicio 1

Calculadora

In [1]:

```
def menu():

    print('1 - Sumar',
          '2 - Restar',
          '3 - Multiplicar',
          '4 - Dividir',
          '5 - Iterativo',
          '6 - Producto punto',
          '7 - Graficar', sep = '\n')

    op = int(input('Elija una opcion: '))

    if (op == 1):
        suma()

    elif (op == 2):
        resta()

    elif (op == 3):
        mult()

    elif (op == 4):
        dividir()

    elif (op == 5):
        iterativo()

    elif (op == 6):
        producto_p()

def suma():

    a,b = input('Ingrese dos numeros para sumar: ').split()
    print('Resultado = ', int(a) + int(b) )

def resta():

    a,b = input('Ingrese dos numeros para restar: ').split()
    print('Resultado = ', int(a) - int(b) )

def mult():

    a,b = input('Ingrese dos numeros para multiplicar: ').split()
    print('Resultado = ', int(a) * int(b) )

def dividir():

    a,b = input('Ingrese dos numeros para dividir: ').split()
    print('Resultado = {:.2f}'.format(int(a) / int(b)))
def iterativo():

    step = int(input('Ingrese el paso: '))
    i = int(input('Ingrese las iteraciones: '))
```

```
a=0
b=0
c=1
```

```
while(i>0):
    a += step
    b -= step
    c *= step
    i-=1
print('a = {}  b = {}  c = {} \n '.format(a,b,c))
```

Ejercicio 2

Se consulta si se desea continuar realizando operaciones y se agrega la operacion de producto punto de matrices

In [2]:

```
import numpy as np

def producto_p():
    op =1
    while(op):
        F1 = int(input("Numero de filas de matriz 1 : "))
        C1 = int(input("Numero de columnas matriz 1 : "))

        print("Ingresar matriz por filas, separado por espacio ")
        elementos = list(map(int, input().split()))
        matriz1 = np.array(elementos).reshape(F1, C1)

        F2 = int(input("Numero de filas de matriz 2 : "))

        C2 = int(input("Numero de columnas matriz 2:  "))
        print("Ingresar matriz por filas, separado por espacio ")
        elementos = list(map(int, input().split()))
        matriz2= np.array(elementos).reshape(F2, C2)

        if (matriz1.shape != matriz2.shape):

            print("No se puede realizar la operacion",
                  "las matrices deben tener el mismo tamaño")
        else:
            break

    res = matriz1*matriz2
    guardar_archivo(matriz1,matriz2,res)
    print("Resultado = ", res)
```

In [3]:

```
def main():
    while(1):
        print(' 1 - Calcular \n',
              '2 - Terminar ')
        cc = int(input(' Elija una opcion: '))
        if (cc == 1):
            menu()
        else:
            return
```

Ejercicio 3

Se agrega una opción de graficar datos ingresados por el usuario y datos extraídos de un archivo

In [4]:

#Definicion de funciones particular para cada grafico

```
import numpy as np
import matplotlib.pyplot as plt

def graficar_plot():
    print("Graficar plot")
    vec = list(map(int, input("Ingrese el vector por elemento separado por espacio: ").split()))
    plt.figure()
    plt.plot(vec, linewidth=1.0, label='Vector ingresado' )
    plt.ylabel('Valor')
    plt.xlabel('Muestra')
    plt.title('Plot Input Vector')
    plt.legend()
    plt.grid()
    plt.show()

def graficar_stem():
    print("Grafico Stem")
    vec = list(map(int, input("Ingrese el vector por elemento separado por espacio: ").split()))
    plt.figure()
    plt.stem(vec, label='Vector ingresado' )
    plt.ylabel('Valor')
    plt.xlabel('Muestra')
    plt.title('Stem')
    plt.legend()
    plt.grid()
    plt.show()

def graficar_hist():
    print("Grafico histograma")
    vec = list(map(int, input("Ingrese el vector por elemento separado por espacio: ").split()))
    plt.figure()
    plt.hist(vec, label='Vector ingresado' )
    plt.ylabel('Valor')
    plt.xlabel('Muestra')
    plt.title('Stem')
    plt.legend()
    plt.grid()
    plt.show()

def graficar_fft():
    print("Grafico FFT ")
    vec = list(map(int, input("Ingrese el vector por elemento separado por espacio: ").split()))
    vec = np.array(vec)
    VEC = np.fft.fft(vec)
    plt.figure()
    plt.plot(VEC, label='Vector ingresado' )
    plt.ylabel('Valor')
    plt.xlabel('Muestra')
    plt.title('FFT')
    plt.legend()
    plt.grid()
    plt.show()

def graficar_filePlot():
    print("Grafico de archivo ")
    vec = np.fromfile('./tp1/sine.log', sep=',')
    plt.figure()
    plt.plot(vec, label='Vector ingresado' )
    plt.ylabel('Valor')
    plt.xlabel('Muestra')
    plt.title('FFT')
    plt.legend()
    plt.grid()
    plt.show()
```

In [5]:

```
def graficar():
    print( ' a - Plot',
           ' b - Stem',
           ' c - Histograma',
           ' d - FFT plot',
           ' e - File plot', sep = '/n')

    op = input('Elija una opcion')

    if ( op == 'a'):
        graficar_plot()
        return
    elif (op == 'b'):
        graficar_stem()
        return
    elif (op == 'c'):
        graficar_hist()
        return
    elif (op == 'd'):
        graficar_fft()
        return
    elif (op == 'e'):
        graficar_filePlot()
        return
```

Ejercicio 4

Almacenar las matrices ingresadas y el resultado de salida de la opción de producto punto

In [6]:

```
def guardar_archivo(a,b,c):

    fd = open("./matrix.log", "w")
    fd.write("Matriz a \n")
    for fila in a:
        np.savetxt(fd,fila)

    fd.write("Matriz b \n")
    for fila in a:
        np.savetxt(fd,fila)

    fd.write("Matriz resultado \n")
    for fila in a:
        np.savetxt(fd,fila)

    fd.close()
```

Verificación

Ejecución de programa completo

In [25]:

```
main()
```

```
1 - Calcular
2 - Terminar
Elija una opcion: 2
```