

# Tutorial de Python con Jupyter Notebook

Nombre : Fernando Sanchez

## Objetivo:

Aprender a realizar programas simples en Python utilizando cuadernos de Jupyter.

## Conocimientos previos:

Conocimientos de programación básica: variables, estructuras de control, funciones, matrices y clases.

## Pregunta 1

```
In [1]: 1 #Intentá imprimir el string "IMAGENES":  
        2 print('IMAGENES')
```

IMAGENES

## Ejercicio 1

Pasar a escala de grises el color codificado en los elementos de la lista `pixel`

```
In [2]: 1 from statistics import mean
2 total= []
3 pixel=[0.6, 0.3, 0.4]
4 #promedioR = mean(pixel)
5 #print('Intesidad :', promedioR)
6 def calcularPromedio(valor):
7     suma = valor
8     longitud = 2
9     resultado= suma/longitud
10    total.append(resultado)
11    promedioRojo= calcularPromedio(pixel[0])
12    promedioVerde= calcularPromedio(pixel[1])
13    promedioAzul= calcularPromedio(pixel[2])
14    #print(total)
15    sumaIntesidad=0
16    for c in total:
17        sumaIntesidad+=c
18
19    print ('La intesidad es:')
20    print (sumaIntesidad)
21
22
23
```

La intesidad es:  
0.6499999999999999

## Ejercicio 2

Pasar a blanco y negro el valor de intensidad codificado en la variable intensidad

```
In [3]: 1 blanco = []
2 negro = []
3 for c in total:
4
5     if c >= 0.5:
6         blanco.append(c)
7         print('Blanco')
8     elif c <= 0.5:
9         negro.append(c)
10        print('Negro', negro)
```

Negro [0.3]  
Negro [0.3, 0.15]  
Negro [0.3, 0.15, 0.2]

## Ejercicio 3

Escribir un for para buscar el máximo de la lista e imprimirlo

```
In [4]: 1 lista=[44,11,15,29,53,12,30]
2 #maximo=max(lista)
3 maximo=0
4 for mayor in lista:
5     if mayor>maximo:
6         maximo = mayor
7         resultado=lista.index(maximo)
8 print('Numero maximo en la lista:', maximo, 'su posicion es:',
9
10
```

Numero maximo en la lista: 53 su posicion es: 4

## Ejercicio 4:

Escribir un for para buscar el minimo elemento de la lista e imprimir su *posición*

## Ejercicio 5 :

Ordenar la lista de forma ascendente

```
In [5]: 1 lista=[44,11,15,29,53,12,30]
2 #minimo= min(lista)
3
4 #print('El numero minimo es:', minimo, 'Su posicion en la lista
5 mini=lista[0]
6 #print(mini)
7
8 for mayor in lista:
9     if mayor<mini:
10         mini=mayor
11         resultado=lista.index(mini)
12         print('Numero menor:', mini, 'posicion en la list es:',
13
14 print('\n')
15
16 print('la lista de forma ascendente es:',sorted(lista) )
17
```

Numero menor: 11 posicion en la list es: 1

la lista de forma ascendente es: [11, 12, 15, 29, 30, 44, 53]

## Ejercicio 7

Crear una funcion en donde me permita enviar como parametro el numero de elementos y devolver un listado de la serie fibonassi con el numero de elementos ingresado.

```
In [6]: 1 def fibonnaci (valor):  
2     a, b = 0,1  
3     for i in range(valor):  
4         print('\n',a, end='\n')  
5         a, b = b, a + b  
6 numero= int(input('Ingrese el valor de la serie fibonnaci a cal  
7 fibonnaci(numero)  
8
```

Ingrese el valor de la serie fibonnaci a calcular: 6

0

1

1

2

3

5

## Ejercicio 8

Escribir una función que reciba una lista y un valor, y devuelva la cantidad de veces que aparece ese valor en la lista

```
In [7]: 1 #Ejercicio 8
2 # Escribir una función que reciba una lista y un valor,
3 #y devuelva la cantidad de veces que aparece ese valor en la li
4
5 def ocurrencias(lista,valor):
6     cont=0
7     for numero in lista:
8         if numero == valor:
9             cont= cont+1
10    print('La cantidad de ocurrencias del valor es:',cont)
11    l=[1,4,2,3,5,1,4,2,3,6,1,7,1,3,5,1,1,5,3,2]
12    v=2
13    ocurrencias(l,v)
14
15
16
```

La cantidad de ocurrencias del valor es: 3

## Ejercicio 9

Generar un CRUD (Crear, Leer, Actualizar y Eliminar) de una agenda de telefono.

```

In [9]: 1 # Generar un CRUD (Crear, Leer, Actualizar y Eliminar) de una a
        2
        3 diccionario = {'0102101010': ['Pepito', 'Perez', '0999999999'],
        4
        5 class agendaTelefono(object):
        6     def __init__(self, nombre, apellido, celular, correo, cedula):
        7         self.nombre = nombre
        8         self.apellido = apellido
        9         self.celular = celular
       10         self.correo = correo
       11         self.cedula = cedula
       12
       13     def crear(self):
       14         diccionario.update({str(self.cedula):[str(self.nombre),
       15
       16     def actualizar(self):
       17         diccionario.update({str(self.cedula):[str(self.nombre),
       18
       19     def leer(self):
       20         valor = int(len(diccionario[str(self.cedula)][0:4]))
       21         if valor > 3:
       22             return str(self.nombre) + " " + str(self.apellido)+
       23         else:
       24             return str('No existe usuario')
       25     def eliminar(self):
       26         valor = int(len(diccionario[str(self.cedula)][0:4]))
       27         if valor > 3:
       28             del diccionario[str(self.cedula)]
       29             print('Usuario eliminado')
       30         else:
       31             print('No existe usuario')
       32

```

```

In [10]: 1 #Primera Persona ingresada
        2 persona1 = agendaTelefono("Fernando", "Sanchez", "0996214346", "f
        3 #Segunda Persona ingresada
        4 persona2 = agendaTelefono("Mateo", "Abad", "072900295", "mateo@ho
        5 #Crear Persona
        6 persona1.crear()
        7 persona2.crear()
        8 #Leer Persona
        9 persona1.leer()
       10 persona2.leer()
       11 #Actualizar Persona
       12 persona1 = agendaTelefono("Luis", "Sanchez", "0996214346", "ferna
       13 persona1.actualizar()
       14
       15 #Eliminar Persona
       16 persona2.eliminar()

```

Usuario eliminado

