

Fundamentos de Python

1- Tipos de datos:

- a. Integer (int): Se utiliza cuando la variable va a tomar un valor de **número entero**.
- b. Float: Se utiliza cuando la variable va a tomar un valor de **número real**.
- c. String: Son cadenas de caracteres, generalmente para uso de texto.
- d. Bool: Una variable booleana va a tomar un valor de Verdadero o Falso.
- e. Listas: Es una colección de elementos del mismo tipo de datos que se encuentran ubicados mediante índices. Es posible modificar los datos que ingresamos a ellas.
- f. Tuplas: Ídem listas a excepción que las tuplas son inmutables, por lo que no se pueden modificar los datos ingresados.

2- Operadores:

- a. Operadores aritméticos: +, -, *, /, //, %, ** (Suma, resta, multiplicación, división, división entera, resto, potencia)
- b. Operadores lógicos: and, or, not
- c. Operadores de asignación: =, +=, -=
- d. Igualación: ==

3- Condicionales

- a. Los condicionales son una estructura de control de flujo que permite tomar decisiones.
 - i. IF-ELIF-ELSE

4- Bucles o Ciclos

- a. Definido: Se utilizan cuando sabemos cuántas veces se va a repetir el código que queremos. Dentro de Python se utiliza el ciclo FOR, el cual se puede utilizar de dos modos: "For elemento in lista:" "For índice in range()"
- b. Indefinido: Se utilizan cuando no sabemos cuántas veces vamos a repetir el código. Si embargo vamos a necesitar una condición de corte o salida, ya que la memoria de nuestras computadoras no es infinita.

5- Funciones

- a. print(): esta función permite mostrar texto por pantalla. Se puede imprimir una cadena que deseamos, el valor de una variable e incluso ambas en simultaneo.
- b. input(): esta función nos permite obtener datos del usuario. Cuando se ejecuta, el programa se frena hasta que el usuario ingrese algo. La función devuelve una cadena de caracteres (string).
- c. int(): transforma una variable en un numero entero (integer). La variable debe ser una cadena numérica (string) o un numero flotante (float), en cualquier otro caso lanzara error.
- d. float(): transforma una variable en un numero flotante (float). La variable debe ser una cadena numérica (string) o un numero entero (integer), en cualquier otro caso lanzara error.

Tipos de datos y operadores

```
#Tipos de datos

#Numeros enteros (int)

num1 = 10
num2 = 20

#Numeros flotantes (Float)

num3 = 3,14
num4 = 26,112011

suma = num2 + num1 #Usamos el operador "+" para sumas dos numeros

resta = num2 - num1 #Usamos el operador "-" para restar dos numeros

multiplicacion = num2 * num1 #Usamos el operador "*" para multiplicar dos numeros

division = num2 / num1 #Usamos el operador "/" para dividir dos numeros y nos de su resultado completo

divisionEntera = num2//num1 #Usamos el operador "/" para dividir dos numeros y que nos de la parte entera de la division

resto = num2%num1 #Usamos el operador "%" para obtener el resto de la division de dos numeros

potencia = num2 ** num1 #Usamos el operador "**" para realizar la potencia (num2 = base, num1 = exponente)

"D:\FIUBA\Algoritmos y Programacion I\venv\Scripts\python.exe" "D:\FIUBA\Algoritmos y Programacion I\asd.py"
la suma es 30
la resta es 10
la multiplicacion es 200
la division es 2.0
la division entera es 2
el resto es 0
el potencia es 10240000000000

Process finished with exit code 0
```

Cadenas de caracteres

```
#Cadenas de caracteres (Strings)

cadena1 = "marianonicolasdiaz"

#Slices o cortes

corte1 = cadena1[0] #--> Corte 1 dara como resultado "m" ya que el numero ingresado permite obtener el caracter segun su indice.
corte2 = cadena1[-1] #--> Corte 2 dara como resultado "z" ya que el valor -1 indica el ultimo caracter de la cadena.
corte3 = cadena1[0:7] #--> Corte 3 dara como resultado "mariano", el primero valor es la posicion inicial, el segundo, hasta donde realizara el corte
corte4 = cadena1[7:] #--> Corte 4 dara como resultado "nicolasdiaz", al no colocar un indice final tomara hasta el ultimo caracter.

"D:\FIUBA\Algoritmos y Programacion I\venv\Scripts\python.exe" "D:\FIUBA\Algoritmos y Programacion I\asd.py"
m
z
mariano
nicolasdiaz

Process finished with exit code 0
```

```
#metodos de strings

cadena2 = "Aquí me pongo a cantar al compás de la vigüela, que el hombre que lo desvela una pena estrordinaria, como la ave solitaria con el cantar se consuela."

buscarCadena = cadena2.find("ave") #--> buscarCadena guardara la primera posicion donde encuentre la palabra "ave".

splitCadena = cadena2.split(",") #--> splitCadena creara una lista donde cada elemento de esta sera lo que esta separado por el caracter ingresado.

minuscula = cadena2.lower() #--> Convierte toda la cadena2 en minusculas

mayuscula = cadena2.upper() #--> Convierte toda la cadena2 en mayusculas

c1 = cadena2[0].islower() #--> devuelve verdadero si el caracter o cadena es una letra en minuscula, falso en caso contrario

c2 = cadena2[0].isalpha() #--> devuelve verdadero si el caracter o cadena pertenece al alfabeto, falso en caso contrario

c3 = cadena2[0].isnumeric() #--> devuelve verdadero si el caracter o cadena es numerico, falso en caso contrario

c4 = cadena2[0].isalnum() #--> devuelve verdadero si el caracter o cadena es alfanumerico, falso en caso contrario

"D:\FIUBA\Algoritmos y Programacion I\env\Scripts\python.exe" "D:\FIUBA\Algoritmos y Programacion I\asd.py"
109
['Aquí me pongo a cantar al compás de la vigüela', ' que el hombre que lo desvela una pena estrordinaria', ' como la ave solitaria con el cantar se consuela.
aquí me pongo a cantar al compás de la vigüela, que el hombre que lo desvela una pena estrordinaria, como la ave solitaria con el cantar se consuela.
AQUÍ ME PONGO A CANTAR AL COMPÁS DE LA VIGÜELA, QUE EL HOMBRE QUE LO DESVELA UNA PENA ESTRORDINARIA, COMO LA AVE SOLITARIA CON EL CANTAR SE CONSUELA.
False
True
False
True

Process finished with exit code 0
```

Listas

```
#Listas

listaVacia = []
listaNumerica = [1,2,3,4,5,6]

listaNumerica.append(2) #-->  agregara al final de la lista el elemento ingresado

listaNumerica.count(4) #--> devuelve la cantidad de veces que se encuentra en la lista el elemento ingresado

listaNumerica.index(4) #--> devuelve el indice donde encuentra el caracter ingresado, si no lo encuentra lanzara error

listaNumerica.pop(0) #--> remueve el elemento que se encuentra en el indice indicado, si no lo encuentra lanzara error. Permite guardar el elemento eliminado

listaNumerica.remove(4) #--> remueve el caracter ingresado, lanzara error si no lo encuentra

listaNumerica.sort() #--> ordenara la lista de menor a mayor

listaNumerica.sort(reverse=True) #--> ordenara la lista de mayor a menor

listaNumerica.clear() #--> borrara todos los elementos de la lista dejandola vacia

"D:\FIUBA\Algoritmos y Programacion I\env\Scripts\python.exe" "D:\FIUBA\Algoritmos y Programacion I\asd.py"
[1, 2, 3, 4, 5, 6, 2] --> Se agrega el elemento 2
1 --> cantidad de veces que esta el numero 4
3 --> indice donde se encuentra el 4
[2, 3, 4, 5, 6, 2] --> se remueve el elemento con indice 0
[2, 3, 5, 6, 2] --> se remueve el elemento 4
[2, 2, 3, 5, 6] --> lista ordenada de menor a mayor
[6, 5, 3, 2, 2] --> lista ordenada de mayor a menor
[] --> lista vacia

Process finished with exit code 0
```

Tuplas

```
#Tuplas

tuplaVacia = ()

tuplaCaracteres = ("A","B","C","D","E")

tuplaCaracteres.index() #--> idem index de la lista

tuplaCaracteres.count("A") #--> idem count de la lista
```

Condicionales

```
#Condicional

if condicion1: #--> condicion debe ser un booleano, debe responder a verdadero o falso, si es verdadero entra en la estructura, sino pasa a la siguiente condicion
    # #--> se realiza algo si se cumple la condicion
elif condicion2: #el programa chequeara esta linea si no se cumple la condicion 1, en caso de cumplirse, entra en la estructura, sino pasa a la siguiente condicion
    # #-->Se realiza algo si se cumple la condicion2
else: #llegara a esta situacion cuando no se cumplan ninguna de las condiciones anteriores
    # #Se realiza algo si no se cumplieron ninguna de la anteriores condiciones

#ejemplo

num1 = 10

if num1 == 10:
    print("El numero 1 es igual a 10")
elif num1 == 20:
    print("El numero 1 es igual a 20")
else:
    print("El numero 1 no es 10 ni 20")

#En este caso la condicion del if se cumple por lo que se imprimira el texto "El numero 1 es igual a 10"
```

Bucle For

```
#Bucle For
for elementoLista in lista:
    #se realiza algo

for valor in range(10):
    #se realiza algo

for valor in range(len(lista)):
    #se realiza algo
```

```
#ejemplos

lista = ["mariano", "nicolas", "jose", "juan", "jorge", "julian", "roman"]

for elemento in lista: #elemento tomara por cada repeticion un elemento de la lista empezando por el primero hasta finalizar la lista
    if elemento == "juan":
        print("Juan fue encontrado")
    else:
        print("Este no es Juan")

for numero in range(10):
    print(numero) #-> en este ciclo se va a imprimir los numeros del 0 al 9. 10 es el numero de repeticiones

for numero in range(5,10):
    print(numero) #-> en este ciclo se va a imprimir los numeros del 5 al 9

for indice in range(len(lista)):
    print(lista[indice]) #-> en este ciclo indice tomara los valores del 0 a la cantidad de elementos que tenga la lista
```

Ejemplo 1:

```
"D:\FIUBA\Algoritmos y Programacion I\venv\Scripts\python.exe" "D:\FIUBA\Algoritmos y Programacion I\asd.py"
Este no es Juan
Este no es Juan
Este no es Juan
Juan fue encontrado
Este no es Juan
Este no es Juan
Este no es Juan
```

Ejemplo 2:

```
"D:\FIUBA\Algoritmos y Programacion I\venv\Scripts\python.exe" "D:\FIUBA\Algoritmos y Programacion I\asd.py"
0
1
2
3
4
5
6
7
8
9

Process finished with exit code 0
```

Ejemplo 4:

```
"D:\FIUBA\Algoritmos y Programacion I\venv\Scripts\python.exe" "D:\FIUBA\Algoritmos y Programacion I\asd.py"
mariano
nicolas
jose
juan
jorge
julian
roman

Process finished with exit code 0
```

Bucle While

```
#Bucle while

while condicion: #--> Se debe cumplir una condicion para que el bucle siga funcionando, una vez que esta condicion deje de ser verdadera el bucle se terminara
    #se realiza algo, en algun momento debe cambiar la condicion para salir del bucle

#Ejemplo

numero = 10

while numero < 20:
    print(numero)
    numero += 1

#En este caso se imprimiran los numeros del 10 al 19, ya que al llegar al numero 20 la condicion sera falsa y se terminara el bucle
```

```
"D:\FIUBA\Algoritmos y Programacion I\venv\Scripts\python.exe" "D:\FIUBA\Algoritmos y Programacion I\asd.py"
10
11
12
13
14
15
16
17
18
19

Process finished with exit code 0
```