Lec 1

Basic tips of Python

By : Sondos Mohamed Ali

- **Python is considered a dynamic language which means the types and values can by changed .**

so you don't have to defines the types of variables , for example:

```
age = 10
```

in python , that means you have a variable named "age" has an integer value = 10

- **Python is a high level language ,it's easy to learn and understand , it use "an interpreter" to run the code .**

Data types in python :-

There are many data types in python and you could think by somehow some of this can do the same task but no ,there are some difference in data types we will learn about..

- **String [str]**

To define a text that means it is string, it can be just a text or maybe numbers "it is treated like a word text not for using operations" or etc..

```
name = "Sondos"
name = 'Sondos'
```

LOOK, You can use double or single quotations

- **Integer [int] , Float [float] , Complex numbers [complex]**

Integer -→ integer numbers

Float -→ numbers with decimal

Complex -→ contain imaginary numbers

```
age = 18 #interger
height = 55.8 #float
com_num = 17j #complex
```

We use '**#**' to make a single line comment .

- **Boolean [bool]**

boolean is for something either true or false

true = 1 , false = 0

if you here after learn cpp , so take care that if condition is true and you want to print it , it will be "true" not "1" ,and the same if it is false .

```
age = 22
print(age<=26)
#output true
```

- **List [list] , Tuple [tuple]**

list and tuple both of them store variables with different data but there are many difference

List..

```
arr = ["cat", "dog", 10 , 12.3]
print(arr[0])
#output cat
```

okay now we have a variable called “arr” its data type -→ list ,its index start from 0 and end to numbers of values-1

```
arr = ["cat", "dog", 10 , 12.3]
print(arr[0] + " " + arr[1])
#output cat dog
```

you can use “**+**” to combine to elements with the same data type

```
arr = ["cat", "dog", 10 , 12.3]
print(arr[0] + " " + arr[2])
TypeError: can only concatenate str (not "int") to str
```

you now have an error

- **Lists provides the flexibility to add ,remove and change its elements on the other hand you can't do it with tuples**

```
arr = ["cat", "dog", 10 , 12.3]
arr[1] = "hello"
print(arr[1])
#output hello (instead of dog)
```

you can change the elements of the list

```
arr = ["cat", "dog", 10 , 12.3]
arr.insert(1,"Hi")
print(arr)
#output ['cat', 'Hi', 'dog', 10, 12.3]
```

`arr.insert(1,"Hi")`

in function insert : 1→ is the index where you want to put the element you want to add which is here “Hi”

```
arr = ["cat", "dog", 10 , 12.3]
arr.append("soso")
print(arr)
#output ['cat', 'dog', 10, 12.3, 'soso']
```

`arr.append("soso")`

append -→ for add an element in last list

```
arr = ["cat", "dog", 10 , 12.3]
arr.remove("cat")
print(arr)
#output ['dog', 10, 12.3]
```

`arr.remove("cat")`

remove the element by its value , **watch out** , python is so sensitive with lower and upper cases , so take care when you write or you face errors

```
arr = ["cat", "dog", 10 , 12.3]
arr.remove("cAt")
print(arr)
ValueError: list.remove(x): x not in list
```

See!

```
arr = ["cat", "dog", 10 , 12.3]
arr.pop(0)
print(arr)
#output ['dog', 10, 12.3]
```

`arr.pop(0)`

pop() do the same job like remove() but instead of writing the value of element , we write its index

Tuple..

Like a list but it's strict ,you can't change anything and more secured and more efficient than list

```
tup = ("cat", "dog", 10 , 12.3)
```

parenthesis are optional, but look, if you want to insert just one item in tuple ,you have to add comma like the following example :

```
tup = "hi",
```

- **in list and tuple we can use negative numbers to define its index , we you call the index [-1] ,that means the last element , [-2] the second last one, until [-number of elements] you will reach the first element.**

> *0   1   2   3   positive index*
>
> *arr = ["cat", "dog", 10 , 12.3]*
>
> *-4   -3   -2   -1   negative index*

- **Range [range]**

it is used with numbers and we often use it with loops

```
num = range(1,100)
```

now “num” has numbers from 1 to 99, 100 is out of range.

Note : To make multiple lines comment we use triple quotes

```
'''
comment
'''

or

"""
comment
"""
```

There are few data types weren't mentioned now but we will discuss it in another time, Good luck !