**PYTHON(Day 1) Aditya Giri**

2 types of programming languages: Dynamic and static.

**Dynamic:** Python/ JS/ Php

**Static:** C/ C++

**Dynamic**: where the data type of a variable does not need to be explicitly assigned.

eg: in python

a=10 and a=nepal can be done.

**Static**: where data type needs to be assigned and cannot be changed.

PYthon:

* + List
  + Dictonary
  + Tuple
  + Set

**Mathematical operations**

addition subtraction multiplication.

**divide:**

two types

interger division and float division

eg:

10/3 =3.333(float division)

10//3=3(integer division)

**Power**

10\*\*2=100

**STRING**

to represent string " " or ' ' but use ""

'' is not mostly used because ' i don't like cat'

solution is 'i don\'t like cat'.

**ARRAY**

x= "i love nepal"

012345 6789 here o=i and 1= space.

**Indexing**: it is the process of retreiving single character from given string expression.

**Syntax**: variable\_name[position]

**2 types**: positive indexing eg a[1] is i

negative indexing eg a[-2] is a this is done so that 'a' can be retreived easily that is in the second last position.

**Slicing**: it is the process of extracting multiple characters from given string expression.

Syntax: variable\_name[start\_position:end\_position:step\_size(optional)]

eg : a[2:5] but the character in 5 is not included in python. so to include character of 5 use 6 i.e [2:6]

step\_size defines jumping of character. eg [2:5:2] means jump 2 characters.

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Day 2**

**DATA TYPES:**

There are two types of data :

**Mutable and Immutable data types**

**Mutable:** Those data types that can be changed directly.

eg: a: {10,34,45,56}

>>a[0]=10

a[0]=90 then 10 is replaced by 90.

**Immutable:** Cannot be changed directly.

eg: strings are immutable and cannot be changed directly

**NOTE: Here 'change' not only means to replace but also add, delete, shift etc.**

**Concatenation:**

"hello"+"Nepal"

>>hello Nepal

Eg:

x=10 y=30 z=30

Three ways to concatenate:

**First**

"The valuce of x is"+str(x)+"y is"+str(y)+"z is"+str(z).

**Second:**

"The value of x is {a}, y is {b}, and z is {c}" .format(a=x, b=y, c=z)

Here format function is used. Mostly used in Python2 but rarely in P3.

**Third:**

"The value of x is {x}, y is {y} and z is {z}" but we must add 'f ' in the front.

f "The value of x is {x}, y is {y} and z is {z}" where f denotes the formatted string.

**Note: '#' is used for single line comment and ''' is used for multi line comment.**

**LIST: Lists are mutable**

Note: Strings are immutable but string inside a list are mutable but again letters in a string inside a list are immutable.

-List is order of collection of data which can be arranged in ascending or decending order.

-It can store any type of data (like integer,float, string, boolean & list itself or dictionery etc).

-It can be declared by list() function or [ ]

eg;

To declare a list a

a=list( )

>>a

or

a=[1,2,3,4]

>>type(a)

<class 'list'>

>>a=["nepal", 12.12, True, 12, [34,32,31]]

**Global Function : len**

len(a);

>>len(a)

5

**Indexing:**

Same as that of a string.

>>a[-1]

>>a[1]

**Nesting Indexing:**

--------------------------------------------------------------------------------------------------------------------------------------------------

**Day 3**

To remove data from the list

**Pop()** // removes based on position

eg;

x=[1,2,3]

>>x.pop()

x=[1,2]

**X.pop(3)**

here 3 is the position.

**remove()** // Removes based on value but only removes the first occurence.

x.remove(2)

here 2 is the actual value

**clear()** // clears all data.

**index()** // gives the position of the value

x.index(1) gives output 0 position.

**count()** // counts the no. of occurence of a value

**reverse** // reverses the list

**sort()**// sorts list in ascending order

sort(reverse=True)// in desending order.

**Note: to sort the list must contain same data type no int char or int float.**

a=x // here list of x is taken by a but when we change value of list of x changes occur in a also.

because they share the same memory address.

x= a. copy()// to overcome the problem where we need the previous list. her new address is allocated.

**DICTIONARY:**

IS a data type to replace list.

unordered collection of data which stores the data in key value pair so cannot be sort.

does not support slicing

only supports indexing and should pass a key

uses { } for dictioneary.

**It is a mutable**.

eg:

price={'shirt':1000} // here shirt is a key and 1000 is a value.

**to add data in dictionary**

**price.update**({'cap':4343})

Note: to add only a single data as above

**price['cap']=3434 no need to use update**

**to remove data**

**price.pop('cap')** /// her key needs to be passed.

**price.popitem()** is a funtion that removes data based of LIFO meaning the data enter latest is removed. Here no key is needed.

**Some other functions**

**price.keys()** // returns list of keys present

**price.value()** // returns list of values

**price.items()** // makes a pair of key and value and returns.

**price.copy()** // same as list.

**TUPLES:**

one of the most unused data type.

**Immutable data type**

slicing and indexing is similar to list.

is an ordered list.

uses ()

eg:

x=(1,2,3)

>>x[0]

1

but we cannot do x[0]=0 /// because it is immutable.

**only uses two functions**

**x.index**

**x.count**

**IF CONDITION**

**Asking Input**

In [2]:



x**=** input("Enter Your Name")

Enter Your Name DarkHorse

In [3]:



x

Out[3]:

'Sushek'

In [4]:



y**=**int(input("Enter Your Age"))

Enter Your Age21

In [5]:



y

Out[5]:

21

In [6]:



y**=**float(input("Enter your weight"))

Enter your weight60.6

In [7]:



'''

if condition

print("this inside the if condition")

print("This is Out side thge if condition")

'''

Out[7]:

'\nif condition\n print("this inside the if condition")\nprint("This is Out side thge if condition")\n'

In [8]:



'''

if condition

print("THis ts inside the if condition")

else:

print("this is inside else")

print("This outside all the condition")

'''

Out[8]:

'\nif condition\n print("THis ts inside the if condition")\nelse:\n print("this is inside else")\nprint("This outside all the condition")\n'