PYTHON BASIC

(DAY 3)

2. LIST :-

Lists are the alternative of array in python.

Lists are mutable (means we can manipulate list)

Syntax : list\_name = [var1, var2, var3,.....]

Variable can be int, char, str, float, double.

*# array = collection of similar data and fixed size*

*# list = collection of data (no similar data) and dynamic size*

*# list is a mutable data type (changable)*

ls = [1,56,37.9,'Abhishek'] *#having same indexing same as string*

print(ls)

print(type(ls))

print(ls[1-len(ls):2-len(ls)])

print(ls[3][2]) *#we can also find out any element from string inside list item*

*# <<<<<<<<<< replace, insert, manupilate, overwrite>>>>>>>>*

li =['Abhishek','Ankit','Abhimanyu','Vicky']

li[1]='Didu' *# update or manipulate*

print(li)

li.append('CP') *#add in the last*

print(li)

*# li.pop(1) #pop out item at index 1*

li.pop() *#pop out last element*

print(li)

li.insert(1,'CP') *# it will insert 'cp' at 1 and shift other elements*

print(li)

li.remove('CP') *#remove 'cp' and shift other items*

print(li)

del li[3] *#remove item with the help of index*

print(li)

print(li.count('Didu')) *#count iteration of item*

print(len(li))

ls1 =['a','b','c','d','z','e','e']

ls2 =[12,46,93,32,45]

ls1.reverse() *#ls1[::-1]*

print(ls1)

ls1.sort()

ls1.sort(reverse = True) *#ls1[::-1]*

print(min(ls2))

print(max(ls2))

print(sum(ls2))

full\_list = ls1 + ls2

print(full\_list)

ls3 =[34,56,86,23]

ls1.append(ls3)

print(ls1)

ls1.extend(ls3)

print(ls1)

ls2 = [0,3,3.1,'Upflairs pvt limited']

ls2[2]=100

print(ls2)

ls2[3][0:9]

print(ls2)

var = ls2[3]

print(var.replace('Upflairs','flipkart'))

3. TUPLE :-

Syntax : tuple\_name = (var1, var2, var3,.....)

Variable can be int, char, str, float, double.

# <<<<<<<<<<<<<<<<<<<< TUPLES >>>>>>>>>>>>>>>>>>>>

# only difference between tuples and list is that tuples are immutables(unchangable)

tp1 = (2,5,76,23,'Abhishek')

# can't perform any operation which manipulate data in tuples

print(tp1[2])

print(tp1.index(23))

print(tp1.count(2))

4. Set :-

1. SET does not show duplicate items.
2. sets are immutable but support insertion and deletion operations.
3. sets do not have any order that's why no indexing methods support.
4. # <<<<<<<<<<<<<<<<<<< SET >>>>>>>>>>>>>>>>>>>

# SET does not show duplicate items

# sets are immutable but support insertion and deletion operations

# sets does not have any order that's why no indexing methods support

st = {12,45,6,723,1,'abhi'}

print(st)

# print(st[2]) give error because no indexing method allowed

st.add('ankit') #insert item

print(st)

st.remove(1)

print(st)

# remove and discard both use for remove item but if item is not present in set and we use "remove" then interpreter through error but not "discard"

# st.remove(456)

st.discard(456)

print(st)

st2={23,56,45,12,347}

st.update(st2) #insert another set (union)

st.update('didu') #it will add "d" "i" "d" "u" as different items.

print(st)

# intesection of sets

print(st.intersection(st2))

1. Dictionary :-

IT IS DIFFERENT FROM SET IN ONLY ONE MANNER THAT IT HAS VARIABLE AND IT'S VALUE SO THAT WE CAN DIFFERENTIATE VALUES AND VARIABLES

# <<<<<<<<<<<<<<<<<<<<<<<<<<<<< DICTIONARY >>>>>>>>>>>>>>>>>>>>>>

# IT IS DIFFERENT FROM SET IN ONLY ONE MANNER THAT IT HAS VARIABLE AND IT'S VALUE SO THAT WE CAN DIFFERENTIATE VALUES AND VARIABLES

# eg:- we need construct a set a student with their numbers of maths subject then we can't substitute value of maths marks in set that's why we use dictionary

d = { 'mohit':23,'Abhishek kuntal':65,'Aditya':56,65:'abhishek'}

print(d['Abhishek kuntal'])

print(d[65])