PYTHON DATA STRUCTURES

LIST class

Sr.No.	Syntax/command	Output	Remarks
1	l = [1,2,3,4, "sudh", 45.56, True]		Indexing of list function works same as string function i.e.0,1,2,3.....(forward) or -1,-2,-3,-4.....(backward)
2	len(l)	7	Gives the no of elements in list
3	del(l[1])		It will remove element at that index. Therefor list are mutable
4	l[0:5]	[1, 2, 3, 4, 'sudh']	**Slicing operation** with subset of the list variable (excludes upper limit)
5	l[:3]	[1, 2, 3]	Without lower bound/limit specified. It will consider default 0
6	l + ["string"]	[1, 2, 3, 4, 'sudh', 45.56, True, 'string']	**Concatenation operation -** will add element to the list
7	L * 2		It will repeat the list continuous
8	2 in l	True	It will detect whether element 2 is in the list.

9	l1 = [1,2,3,4,5]
	max(l1)
		5	Will give max value within list and shall perform function only if list contains only 1 type of elements i.e either int/str/float/bool.

10	l2 = ['sudh', 'ineuron', 'gates']
	max(l2)
		sudh'	In case of string elements in list it will give max value depending on alphabetical order.

11	l.append("kumar")	[1, 2, 3, 4, 'sudh', 45.56, True, 'kumar']	It will add the element to the very end of the list variable.
12	l.append([6,7,8,9])	[1, 2, 3, 4, 'sudh', 45.56, True, 'kumar', [6, 7, 8, 9]]	You can also append list inside a list
13	l.pop(0)	1	It will pop up the data of said index and remove the element from the list.
14	l.pop()		It will pop default last element and remove it from the list.
15	l.reverse()	['kumar', True, 45.56, 'sudh', 4, 3, 2,1]	it will reverse the entire list and will keep the change to list.
16	l[::-1]	['kumar', True, 45.56, 'sudh', 4, 3, 2,1]	it will reverse the entire list and will not maintain the change to list unless you reassign the list.
17	l1.sort()	[1,2,3,4,5]	It will sort the elements in ascending order and shall perform function only if list contains only 1 type of elements i.e either int/str/float/bool.
18	l1.sort(reverse = True)	[5, 4, 3, 2, 1]	It will sort the elements in descending order and shall perform function only if list contains only 1 type of elements i.e either int/str/float/bool.
19	l4 = [l1,l2,l3]		Provide list inside a list
20	l4[2][1]		Extract element from a list inside a list where 2 reperesents list l3 and 1 represents index 1 of l3
21	l.count(2)		Will give the no of occurrence of element 2 in the list.
22	l.extend("add")	........, 'a','d','d']	it will add the element specified to the end of list in seperated form

23	l3 = [7, 8, 9]
	l3.extend([45,65,77])
		[7, 8, 9, 45, 65, 77]	The elements to be added should be provided in list form to get added to the eisting list and the same will be included in existing list without separate list.

24	l3.append([45,65,77])	[7, 8, 9, [45, 65, 77]]	It will add list inside a list.

25	l3.index('a')	3	It will mention the index of the element you mention.

26	l7 = [1,2,3,4,5,6,7,7,7]
	l7.index(7)
		6	Even if there are repeated elements, It will mention the index of the very first element

27	l5 = [1, 2, 3, 4, 5, 6, 45]
	l5.insert(1,66)
		[1, 66, 2, 3, 4, 5, 6, 45]	To add the required element at the position or index as required.

28	l5.pop()	45 [1, 66, 2, 3, 4, 5, 6]	It will remove the element which is at the very end. To remove specific element you need to mention the index of that element.
29	l7.remove(7)	[1, 2, 3, 4, 5, 6, 7, 7]	In case there are recurring elements of a value, it will remove first occurance of the value. You need to mention the element you want to remove. The same can be applied to non recurring elements as well.
30	l1[2] = "nio"	[1, 2, 'nio', 4, 5]	It will replace the element at index position 2 with 'nio'

31	s = 'sudhanshu'
	s[3] = 'u'
		Error	Item reassignment cannot be done inside a itterable as **string elements are immutable. Therefore list has mutable property.**

		TUPLES class	
32	t = (33,455,'adit') t1 = (1,2,3,4)		Tuples similar to list with difference in brackets
33	t[0]	33	Follows indexing of elements similar to list variable.
34	t[0:2]	(33, 455)	Extract range of elements.
35	t[0:2:2]	-33	
36	t[2] = 'u'	**Error**	Item reassignment cannot be done inside a itterable as **tuple elements are immutable. List has mutable property.**
37	t + t1	(33,455,'adit',1,2,3,4)	**Concatenation operation** - will add element of both tuples to create new tuple
38	t.count(1)		No of occurance of an element in tuple
39	list(t)		Convert tuple to list
40	tuple(l)		Convert list to tuple
41	Tuple within tuple		Nested tuple
42	str(t)		Tuple can be converted to string but ull not be able to extract individual elements. So not advisable
		DICTIONARY class	
43	d2 = {'key1' : 'value', 234 : 'xyz', 'name' : 'sudh', 'no' : 111111}		You need to create a key : value pair to create a dict variable
44	d2[234]	'xyz'	To get the value you need to mention the key.
45	'name' in d2	True	It will check whether the said key in in the dict
46	del(d2[0])		This will delete key value pair at the said key index.
47	d3 = {'name' : 'sudh', 'tech' : ['python', 'ML', 'DL', 'CV', 'Big data']}		You can also provide list/tuple inside the dict for value.
48	d3['tech']	['python', 'ML', 'DL', 'CV', 'Big data']	
49	'python' in d3['tech']	TRUE	It will show whether the required element is within value of the key mentioned.

50			All the keys provided should be unique and there should be no repetition. Values can be repeated
51	d4['k1'] = 'rrr'		To add key value to the dict. Incase you try to add a same name key then the value will get replaced.
52	d4['k2']['no']	1111111	You can also add dict inside a dict and get the value of key of dict inside a dict
53	d4 = {'name': 'rrr', 'contact_info': 9990768990, 'k1': 'rrr', 'name11': 'rrr', 'k2': {'key1': 'value', 234: 'xyz', 'name': 'sudh', 'no': 111111}}		Special symbols cannot be used as a key except undescore (_)
54	d6 = {(34,55,66) : "hair"} d6[(34,55,66)]	'hair'	You can provide tuple as key but list/dict not allowed as key
55	del(d4["name])		It will delete that key from the dict
56	d2 + d3	Error	Concadination operations not possible here
57	d2.keys()		Will list out all the keys from the dict
58	d2.values()		Will list out all the values from the dict
59	d2.items()		Will list out all the key values pairs from the dict in tuples
SET class			
60	S = {1,2,3,,5,3,3,3}	{1,2,3,5}	It will keep only unique entities and remove any repeative values and will also align the values in increasing order. Any string value will be aligned at the end.
61	s[0]	Error	Index operations not applicable in set functions. You can perform slicing and concardination operations after converting set to a list/tuple.
62	list(set(l))		To remove repeated elements in a list, first convert it to set and the convert back to list.
63	s3.add('dd')		To add element to a set
64	s4 = {(23,3,34,5,3), 34, 67, 67, 234, 23}		Mutable entities you can keep inside the set but no immutable entities.
65	s4.remove(34)	{(23, 3, 34, 5, 3), 23, 234, 67}	To remove any element from the set and will show error if that element is not present in the set.
66	s4.discard(23)	{(23, 3, 34, 5, 3), 234, 67}	To remove any element from the set and will **not** show error even if that element is not present in the set.
67	chr(65)	A	All the representation for alphabets starts from 65 and above i.e A = 65, B =66 and so on.
68	set1=[1,2,3,4]	{3,4}	It will give intersection of the two sets.
69	set2=[3,4,5,6]		
70	set3=set1 & set2		
71	set1.intersection(set2)	{3,4}	It will give intersection of the two sets.
72	set1.union(set2)	{1,2,3,4,5,6}	This will give union of the two sets
73	Set4=[1,2]	True	Check for subset and return boolean value
74	set4.issubset(set1)		
75	set4.issuperset(set1)	FALSE	Check if super set and return boolean value
76	set1.difference(set4)	{3,4}	It will remove elements from set1 which are common in set4