Exercise 10: List, Tuple, Sets and Dictionary [01/10/2019]

List, Tuple, Set n Dictionary

Lists

--

Create a list of your choice.

Iterate over the list and display the elements of the list.

Concatnate two lists,

Perform multiplication on a list with a no.

myLst = [1, "A", 'MSU', 123, 909, 78.88, 77, 6544, 78.50, 7+9j, True, False, 900000, "XYZ", 90.99999999999]

myLst

myLst[6]

myLst[-2]

myLst[-10]

myLst[0:6]

myLst[1:6]

myLst[6:10]

myLst[:]

myLst[::]

myLst[::-1]

myLst[:10]

myLst[::2]

myLst[0::1]

myLst[2:13:4]

Delete an ite from the list.

Delete elements available at odd indices from list

Delete elements available at even indices from list

Delete all the elements of the list

Detete the definion of the list and verify it

Demonstrate that a list is mutable object.

Undate the value of list at index 5 to 200000.

Create a nested list consisting of various elements along with one or two lists.

Clone alist into another list.

Demonstrate the use of various functons on a list.

min()

mx()

len()

sum()

all() # Returns true if all elements of the list are true.

any() # Returns true if any element of the list are true.

list() # Convets an iterable (tuple, string, set, dictionary) to list.

sorted() # Returns a new sorted list. The original one is not sorted.

in and not in operators

--

check whether certain elements are there or not in a list.

Demonstrate the use of various functons on a tuple.

append()	Add Single Element to The List
extend()	Add Elements of a List to Another List
insert() Inserts Element to The List	
remove()	Removes Element from the List
index()	returns smallest index of element in list
count()	returns occurrences of element in a list
pop()	Removes Element at Given Index
reverse()	Reverses a List
sort()	sorts elements of a list
copy()	Returns Shallow Copy of a List
clear()	Removes all Items from the List
enumerate()	Returns an Enumerate Object. Returns values along with the index.

```
myList = [1,2,3,4,5,6,0,7]

##print(myList)
##print(type(myList))
##
##myTpl = tuple(myList)
##
##print(myTpl)
##print(type(myTpl))

##print(myList[:])
##print(myList[0:])
##print(myList[0::])
##print(myList[::])
```

```
##print(myList[0::1])
##print(myList[0:len(myList):])
##print(myList[0:len(myList):1])
##print(myList[:: -1]) # Reverse the list
##print(myList[::2])
##print(myList[2:7:])
##
##list1 = ["abc","2",34,89,"ZZ",4.534]
##list2 = [1,89,"Krish", "Python", True, 2+8j, (1,2,3), [3,4,5], {3,4,5}, {1:"A"}]
####print(list1)
####print(list2)
##list1[2] = "234"
##del list1[3]
##del list2
##del list1[:] # Deletes all the elements of the list1
##del list1 # Deletets the definition of the list



##list1 = ["abc","2",34,89,"ZZ",4.534]
##list2 = [1,89,"Krish", "Python", True, 2+8j]
##list3 = ["abc","2",34,89,"ZZ",4.534, [2,3,4], 809]
####print(list1,list2,list3, sep =",")
##
##list4 = list1+list2
##print(list4)

##
### Program to insert a list in another list using slice operation
##
##myL =  [1,2,3]
##myL2 = [1,2,3]
##myL[1] =myL2
##print(myL)
##
##
### Cloning list
##list1 = ["abc","2",34,89,"ZZ",4.534]
##list2 = list1
##print(list1)
##print(list2)
##
```

```
##list1,list2 =[1,2,3], [4,3,2,1]
##print(list1)
##print(list2)
##



#List Opeartions
##list1=[1,2,3]
##print(len(list1))
##print(list1*3)
##
##print(3*list1)
##print(1 in list1)
##print(4 in list1)
##print(4 not in list1)
##print(1 not in list1)
##
##print(max(list1))
##print(min(list1))

##list4 = ["A", "a", "Ab", "ZZZ", "zzz"]
##print(max(list4))
##print(min(list4))
##
##list5 = [1,89,89.9,34,90,100.1,100.0]
##print(max(list5))
##print(min(list5))



##list1=[1,2,3]
##print(sum(list1))
##
##
##print(all(list1))
##list2 = [0,1,-3]
##print(all(list2))
##
##print(any(list2))
##sortedL1 = sorted(list1)
##print(sortedL1)
```

```
###List Methods and Functions
num_list = [6,3,7,6,0,1,2,4,9]
print(num_list)
print(num_list.count(6))
num_list.append(10)
print(num_list)
num_list.pop()
print(num_list)

num_list.pop(1)
print(num_list)

##num_list.remove(3)
##print(num_list)

num_list.remove(7)
print(num_list)
num_list.reverse()
print(num_list)



num_list.sort()
print(num_list)

str_list = ["A", "B"]
num_list.extend(str_list)
print(num_list)

num_list.insert(2,3)
print(num_list)

print(num_list.index(3)) # Returns the index of value 3
print(num_list)
```

Create a list that is populated with the all even numbers between 1-1000

Create a list that is populated with the all odd numbers between 1-1000

Create a list that is populated with the all numbers divisible by 5 between 1-1000

Create a list that is populated with squares values of numbers between 1-1000

Write a program to find sum and mean of elements in a list consitiong only numerical values.

Implement a list to behave like a Stack

Implement a list to behave like a Queue.

Tuple

--

Create a tuple of your choice.

Iterate over the list and display the elements of the tuple.

Concatnate two tuple,

Perform multiplication on a tuple with a no.

myTpl = (1, "A", 'MSU', 123, 909, 78.88, 77, 6544, 78.50, 7+9j, True, False, 900000, "XYZ", 90.99999999999)

myTpl

myTpl[6]

myTpl[-2]

myTpl[-10]

myTpl[0:6]

myTpl[1:6]

myTpl[6:10]

myTpl[:]

myTpl[::]

myTpl[::-1]

myTpl[:10]

myTpl[::2]

myTpl[0::1]

myTpl[2:13:4]

Demonstrate that a tuple is mutable object.

Detete the definion of the list and verify it.

Write a program to assign multple varibales using tuple.

Write a program to swap two no. using tuple assignment.

Demonstrate the use of various functons on a tuple.

min()

mx()

len()

tuple()

zip()

Demonstrate the use of various methods on a tuple.

index()	returns smallest index of element in tuple
count()	returns occurrences of element in a tuple

in and not in operators

--

check whether certain elements are there or not in a tuple.

Create a nested tuple.

Iterate over a nested tuple.

Write a program to use divmod() function and print the results of division and modulus by storing the returned values into a tuple.

Sets

Create a tuple of your choice.

print the set.

Iterate over the set.

any()	Checks if any Element of an Iterable is True
all()	returns true when all elements in iterable is true
len()	Returns Length of an Object
max()	returns largest element
min()	returns smallest element
set()	returns a Python set
sorted()	returns sorted list from a given iterable
sum()	Add items of an Iterable
zip()	Returns an Iterator of Tuples

Demonstrate the use of various methods on a set.

remove()	Removes Element from the Set
add()	adds element to a set
copy()	Returns Shallow Copy of a Set
clear()	remove all elements from a set
difference()	Returns Difference of Two Sets

difference_update()	Updates Calling Set With Intersection of Sets
discard()	Removes an Element from The Set
intersection()	Returns Intersection of Two or More Sets
intersection_update()	Updates Calling Set With Intersection of Sets
isdisjoint()	Checks Disjoint Sets
issubset()	Checks if a Set is Subset of Another Set
issuperset()	Checks if a Set is Superset of Another Set
pop()	Removes an Arbitrary Element
symmetric_difference()	Returns Symmetric Difference
symmetric_difference_update()	Updates Set With Symmetric Difference
union()	Returns Union of Sets
update()	Add Elements to The Set.

in and not in operators

--

check whether certain elements are there or not in a set.

```
#Set Operations/Functions and Methods
##my_set = {1,2,3}
##print(my_set)
##
##
##my_set = {1.0, "Hello", (1, 2, 3)}
##print(my_set)



##my_set = {1,2,3,4,3,2}
##print(my_set)
##
##my_set = {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1} # singlton set
##print(my_set)
##
##my_set = {} # empty dictionary, not a set
##print(my_set)
##print(type(my_set))



##mySet =set() # will create an empty set
```

```
##print(type(mySet))



### we can make set from a list
##myList = [1,2,3,4,5]
##mySet = set(myList)
##print(mySet)
##
##my_set = set([1,2,3,2])
##print(my_set)



# we can make set from a list
##myTpl = (1,2,3,4,5,5,5,5,5)
##mySet = set(myTpl)
##print(mySet)

##
##thisset = {"apple", "banana", "cherry"}
##print(thisset)



##thisset = {"apple", "banana", "cherry"}
##
##for x in thisset:
##  print(x)



##thisset = {"apple", "banana", "cherry"}
##
##print("banana" in thisset)




##thisset = {"apple", "banana", "cherry"}
##
##thisset.add("orange")
##
##print(thisset)
```

```
thisset = {"apple", "banana", "cherry"}

print(len(thisset))



thisset = {"apple", "banana", "cherry"}

thisset.remove("banana")

print(thisset)


thisset = {"apple", "banana", "cherry"}

thisset.discard("banana")

print(thisset)


thisset = {"apple", "banana", "cherry"}

x = thisset.pop()

print(x)

print(thisset)


thisset = {"apple", "banana", "cherry"}

thisset.clear()

print(thisset)


thisset = {"apple", "banana", "cherry"}

del thisset

print(thisset)
```

```
set1 = {"a", "b" , "c"}
set2 = {1, 2, 3}

set3 = set1.union(set2)
print(set3)
```

```
set1 = {"a", "b" , "c"}
set2 = {1, 2, 3}

set1.update(set2)
print(set1)
```

```
thisset = set(("apple", "banana", "cherry")) # note the double round-brackets
print(thisset)
```

Dictionary

--

Create a dictionary of roll no and names of your class.

Create a dictionary of months and print the value of a no indicating day of month.

Create a dictionary of weeks and print the value of a no. indicating day of week.