lists and tuples are two commonly used data structures that allow you to store collections of items.

**List:**

A list is a collection of ordered elements. Lists can be of different data types such as integers, floats ,strings etc. various operations such as adding or removing elements of searching and sorting for elements can be performed in a list in python

# create list  
  
my\_list=[1,5.5,"santosh","peddinti",True,False]  
  
print(my\_list) # opt: [1,55,14,'santosh','peddinti',True,False]  
  
# 1. Append – add the new items into the list  
  
my\_list.append("gopinadh")  
  
print(my\_list) # opt: [1,5.5,'santosh','peddinti',True,False,'gopinadh']  
  
# 2. Extend – extend the list to add new items  
  
my\_list.extend([5,3])  
  
print(my\_list)  
  
# opt: [1,5.5 , 'santosh', 'peddinti' , True, False 'gopinadh' 5, 3]  
  
# 3. insert(index, item) – insert the data at particular (index) position.  
  
my\_list.insert(2,"jeevan")  
  
print(my\_list)  
  
# opt: [1,5.5, 'jeevan','santosh','peddinti',True,'False','gopinadh',5,3]  
  
# 4. remove(item) – remove the item in the list  
  
my\_list.remove("peddinti")  
  
print(my\_list)  
  
# opt: [1,5.5, 'jeevan', 'santosh', True, False, 'gopinadh',5,3]  
  
# 5.pop(item) – remove last item in the list  
  
my\_list.pop()  
  
print(my\_list)  
  
# opt: [1,5.5, 'jeevan' 'santosh', True , False, 'gopinadh', 5]  
  
# 6. sort () – sort the list items  
  
my\_list1=[1,43,5,2,535,21,444,20]  
  
my\_list1.sort()  
  
print(my\_list1) # opt: [1, 2, 5, 20, 21, 43, 444, 535]

Applications of list:

* Lists can be used to store and access sequential data.
* We can modify elements in a list because lists are mutable.

**Tuple:**

a tuple is a sequence of elements separated by commas and enclosed in parentheses. tuples are similar to lists, but they cannot be modified once created. This means that you cannot add. Remove or modify elements of a tuple.

# create the list of numbers

list\_tuple = (1,2,3,4,5)

#print first three elements of tuple

print(list\_tuple[:3])

# opt: (1,2,3)

# deleting a tuple

del\_tuple = (1,2,3,4,5)

# delete tuple using del

tuple\_del=(1,2,3,4,5)

# delete tuple using del

del tuple\_del

print(tuple\_del)

Applications of Tuple:

* Unlike lists , tuples are immutable and can be used to store dictionary keys.
* Tuples can return multiple values from a function such as you can use a tuple to return minimum and maximum values from a list.

**Set:**

A set is a built-in data structure in Python that represents a collection of unique elements. We can perform different operations on sets, such as union, intersection, difference, and symmetric difference.

# create the set  
my\_set = {1,44.4,15,7,51}  
  
print(my\_set) # opt: {1, 51, 7, 44.4, 15}  
  
# add – add items into the set  
  
my\_set.add(5)  
  
print(my\_set) # opt: {1, 51, 5, 7, 44.4, 15}  
  
# remove – remove the item in a set  
  
my\_set.remove(44.4)  
  
print(my\_set) # opt: {1, 51, 5, 7, 15}  
  
# union set – Returns a new set with all unique items from both sets.   
my\_set1={43,22,67,643}  
  
union\_set=my\_set.union(my\_set1) # opt: {1, 67, 643, 5, 7, 43, 15, 51, 22}  
  
print(union\_set)  
  
# intersection\_set - Returns a set of items common to both sets.   
  
intersection\_set=my\_set | my\_set1  
  
print(intersection\_set) # opt: {1,67, 643, 5, 7, 43, 15, 51, 22}

**Applications**

* Sets can be used to perform various operations such as union, intersection and difference.
* Sets can be used to remove duplicates from a list.

**Dictionary :**

A dictionary in Python is a collection of key-value pairs, where each key is unique and associated with a value. We can change the values of a dictionary. They are useful for storing and accessing data.

# create the dictionary  
  
my\_dict={"name":"santosh","last\_name":"peddinti","address":"vizag"}  
  
print(my\_dict)  
  
# opt: {'name':'santosh','last\_name':'peddinti','address':'vizag'}  
  
# get a key – get only specified key name in the dictionary  
  
get\_key=my\_dict.get("name")  
  
print(get\_key) # opt: santosh  
  
# get keys – get all the keys in dictionary   
  
only\_keys=my\_dict.keys()  
  
print(only\_keys) # opt: dict\_keys( ['name', 'last\_name','address'] )  
  
# get values – get all the values in dictionary  
  
only\_values=my\_dict.values()  
  
print(only\_values) # opt: dict\_values(['santosh', 'peddinti', 'vizag'])  
  
# items – git all items in a dictionary like a list  
  
only\_values=my\_dict.items()  
  
print(only\_values)  
  
# opt: ([('name', 'santosh'),('last\_name','peddinti'),('address','vizag')])  
  
# pop(key) – remove the item in a dictionary with key name  
  
only\_pop=my\_dict.pop("name")  
  
print(only\_pop) # opt: santosh

**Applications of Dictionary**

* Dictionaries can be used to store key-value pairs.
* Dictionaries are also useful for counting the frequency of elements in a list.

**Difference Between List, Tuple, Set and Dictionary**

Now we will draw a comparison table between a list, tuple, set and dictionary in Python and discuss about the characteristics of each of them

|  |  |  |  |
| --- | --- | --- | --- |
| **List** | **Tuples** | **Set** | **Dictionary** |
| A list is a collection of ordered elements. | A tuple is a sequence of elements separated by commas and enclosed in parentheses. | A set is a built-in data structure in Python that represents a collection of unique elements. | A Dictionary is a collection of key-value pairs, where each key is unique and associated with a value. |
| Lists maintain the order of the elements they contain. | Tuples maintain the order of the elements they contain | Sets do not maintain the order of the elements they contain | Dictionaries do not  maintain the order of the elements they contain. |
| Lists can be accessed by index | Tuples can be accessed by index | Sets cannot be accessed by index | Dictionaries cannot be accessed by index |
| Lists can be modified by adding or removing elements | Tuples cannot be modified by adding or removing elements | Sets can be modified by adding or removing elements | Dictionaries cannot be modified by adding or removing elements |
| Lists can contain duplicate elements | Tuples can contain duplicate elements | Sets cannot contain duplicate elements | Dictionaries cannot contain duplicate elements |
| Lists can be accessed by index | Tuples can be accessed by index | Sets cannot be accessed by index | Dictionaries cannot be accessed by index |
|  |  |  |  |