There are four collection data types in the Python programming language:

**List** is a collection which is ordered and changeable. Allows duplicate members.

**Tuple** is a collection which is ordered and unchangeable. Allows duplicate members.

**Set** is a collection which is unordered, unchangeable\*, and unindexed. No duplicate members.

**Dictionary** is a collection which is ordered\*\* and changeable. No duplicate members.

**Dictionary:**

Accessing Items:

* You can access the items of a dictionary by referring to its key name, inside square brackets**:** thisdict["model"]
* There is also a method called get() that will give you the same result**:** thisdict.get("model")
* The keys() method will return a list of all the keys in the dictionary**:** thisdict.keys()
* The values() method will return a list of all the values in the dictionary**:** thisdict.values()
* The items() method will return each item in a dictionary, as tuples in a list**:** thisdict.items()

Check if a key exists in a dictionary: in operator - ('key1' in d)

Check if a value exists in a dictionary: in operator, values() - ('val1' in d.values())

Check if a key-value pair exists in a dictionary: in operator, items() - (('key1', 'val1') in d.items())

You can change the value of a specific item by referring to its key name: thisdict["year"] = 2018 or thisdict.update({"year": 2020})

* The pop() method removes the item with the specified key name: thisdict.pop("model")
* The popitem() method removes the last inserted item (in versions before 3.7, a random item is removed instead):thisdict.popitem()
* The del keyword removes the item with the specified key name: del thisdict["model"]
* The del keyword can also delete the dictionary completely: del thisdict
* The clear() method empties the dictionary: thisdict.clear()

**Set:**

You cannot access items in a set by referring to an index or a key.

But you can loop through the set items using a for loop, or ask if a specified value is present in a set, by using the in keyword.

* To add one item to a set use the add() method: thisset.add("orange")
* To add items from another set/list/tuple/dict into the current set, use the update() method: thisset.update(set/list..)
* To remove an item in a set, use the remove(), or the discard() method:
* thisset.remove("banana") - If the item to remove does not exist, remove() will raise an error.
* thisset.discard("banana") - If the item to remove does not exist, discard() will NOT raise an error.
* You can also use the pop() method to remove an item, but this method will remove a random item, so you cannot be sure what item that gets removed.The return value of the pop() method is the removed item.
* The clear() method empties the set: thisset.clear()
* The del keyword will delete the set completely: del thisset