**Data Structures in Python - 2**

### **Topics Covered -**

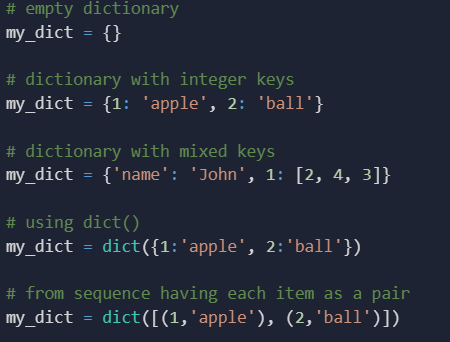
* Dictionary
* Set

### **Dictionary**

* Python dictionary is an unordered collection of items. Each item of a dictionary has a key/value pair.
* Dictionaries are optimized to retrieve values when the key is known.

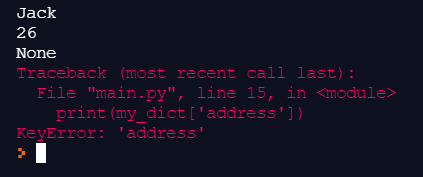
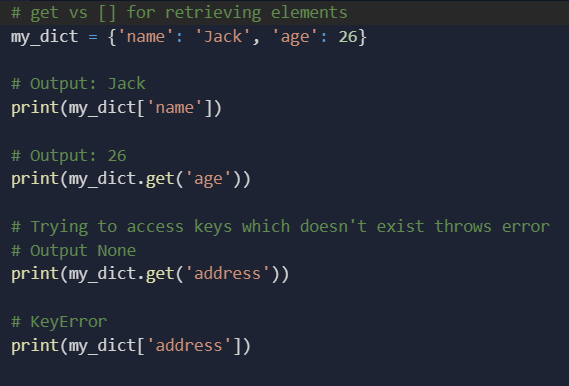
**Creating a dictionary**

* Creating a dictionary is as simple as placing items inside curly braces {} separated by commas.
* An item has a key and a corresponding value expressed as a pair (key: value).
* While the values can be of any data type and can repeat, keys must be of immutable type (string, number, or tuple with immutable elements) and must be unique.
* We can also create a dictionary using the built-in dict() function.



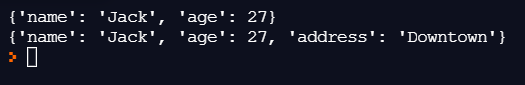
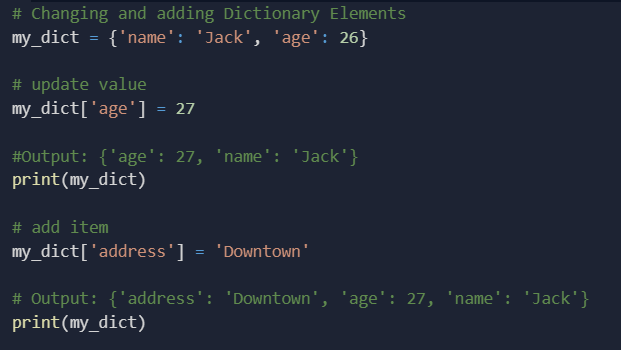
**Accessing Elements from Dictionary**

* While indexing is used with other data types to access values, a dictionary uses keys. Keys can be used either inside square brackets [] or with the get() method.
* If we use the square brackets [], KeyError is raised if a key is not found in the dictionary. On the other hand, the get() method returns None if the key is not found.



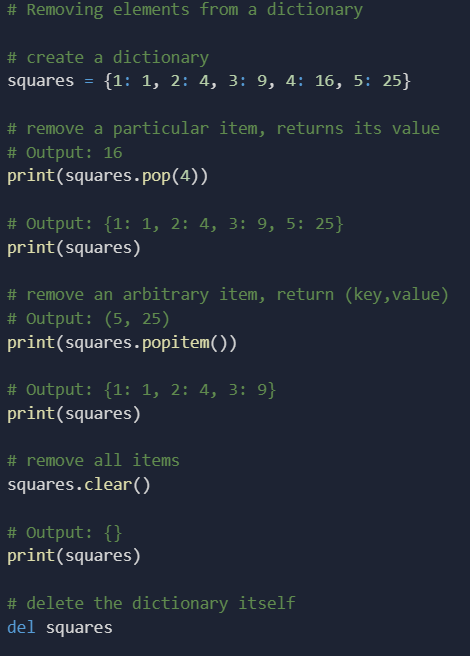
**Changing and Adding Dictionary elements**

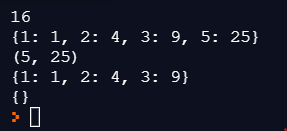
* Dictionaries are mutable. We can add new items or change the value of existing items using an assignment operator.
* If the key is already present, then the existing value gets updated. If the key is not present, a new (key: value) pair is added to the dictionary.



**Removing elements from Dictionary**

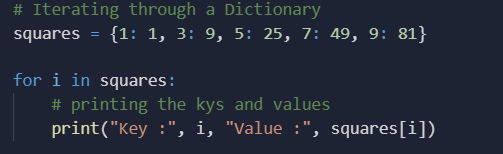
* We can remove a particular item in a dictionary by using the pop() method. This method removes an item with the provided key and returns the value.
* The popitem() method can be used to remove and return an arbitrary (key, value) item pair from the dictionary. All the items can be removed at once, using the clear() method.
* We can also use the del keyword to remove individual items or the entire dictionary itself.

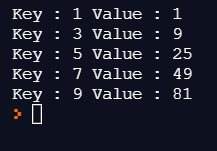




**Iterating Through a Dictionary**

* We can iterate through each key in a dictionary using a for loop.



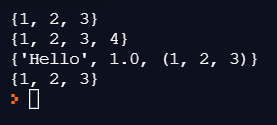
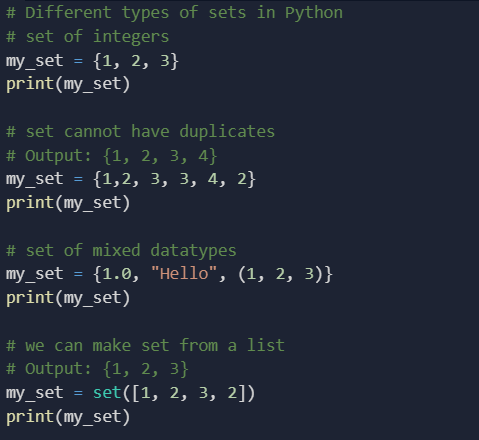


### **Set**

* A set is an unordered collection of items. Every set element is unique (no duplicates) and must be immutable (cannot be changed).
* However, a set itself is mutable. Therefore, we can add or remove items from it.
* Sets can also perform mathematical set operations like union, intersection, symmetric difference, etc.

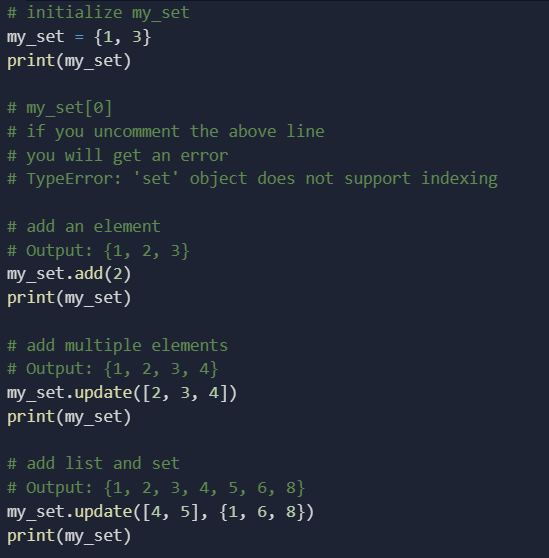
**Creating Python Sets**

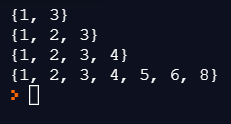
* A set is created by placing all the items (elements) inside curly braces {}, separated by a comma, or by using the built-in set() function.
* It can have many items, and they may be of different types (integer, float, tuple, string, etc.). But a set cannot have mutable elements like lists, sets, or dictionaries as its elements.



**Modifying a set in Python**

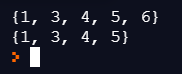
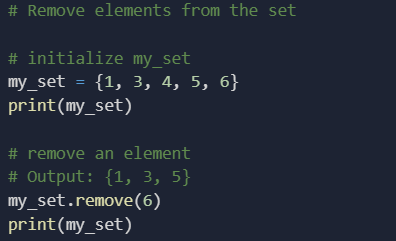
* Sets are mutable. However, since they are unordered, indexing has no meaning.
* We cannot access or change an element of a set using indexing or slicing. This is because the set data type does not support it.
* We can add a single element using the add() method and multiple elements using the update() method. The update() method can take tuples, lists, strings, or other sets as its argument.
* In all cases, duplicates are avoided.





**Removing elements from a set**

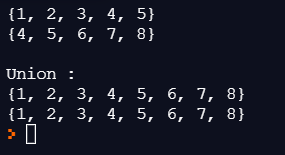
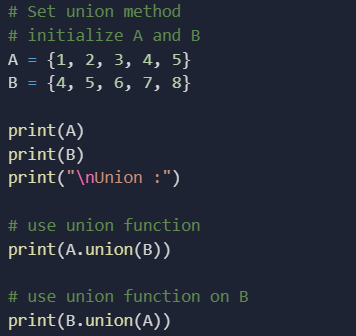
* A particular item can be removed from a set using the method remove().



**Python Set Operations**

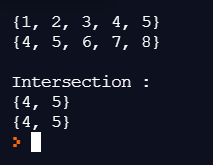
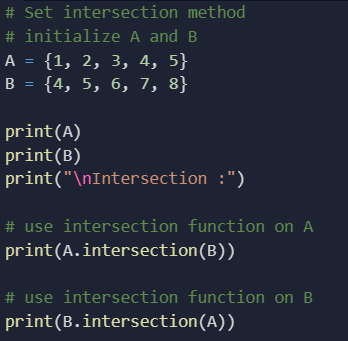
* Sets can carry out mathematical set operations like union, intersection, difference, and symmetric difference. We can do this with operators or methods.
* **Set Union**

1. Union of A and B is a set of all elements from both sets.
2. Union is performed using the union() method.



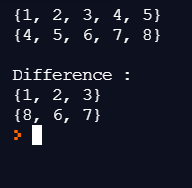
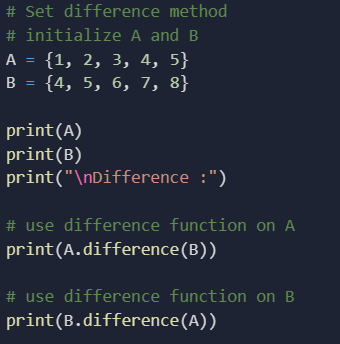
* **Set Intersection**

1. The intersection of A and B is a set of elements that are common in both sets.
2. The intersection is performed using the intersection() method.



* **Set Difference**

1. The difference of set B from set A(A - B) is a set of elements that are only in A but not in B. Similarly, B - A is a set of elements in B but not in A.
2. The difference is performed using the difference() method.



* **Set Symmetric Difference**

1. The Symmetric Difference of A and B is a set of elements in A and B but not in both (excluding the intersection)
2. The symmetric difference is performed using the method symmetric\_difference().

