Day: 12

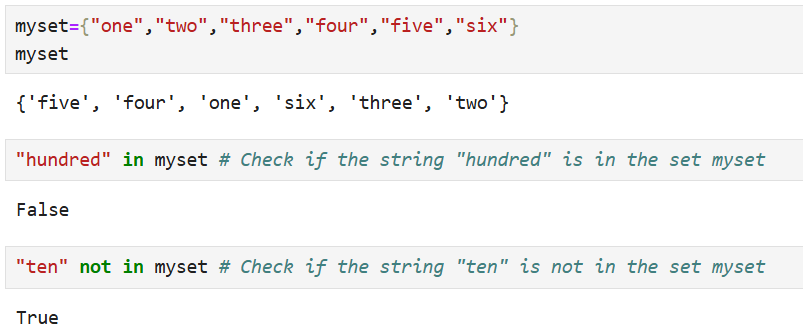
21 November, 2023

**Set Membership:**

Set membership is performed using **in** and **not** **in** keyword.

**in** is used to check if any element is present in the set. **not in** is used to check is any element is not present in the set.

Example:



**Add and Remove Items:**

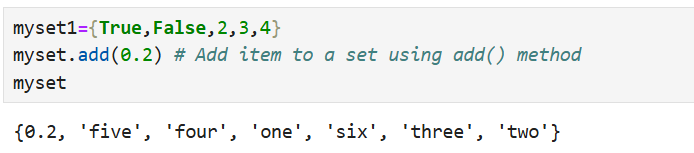
Set is mutable, meaning, we can add, update and remove items or elements in set data type.

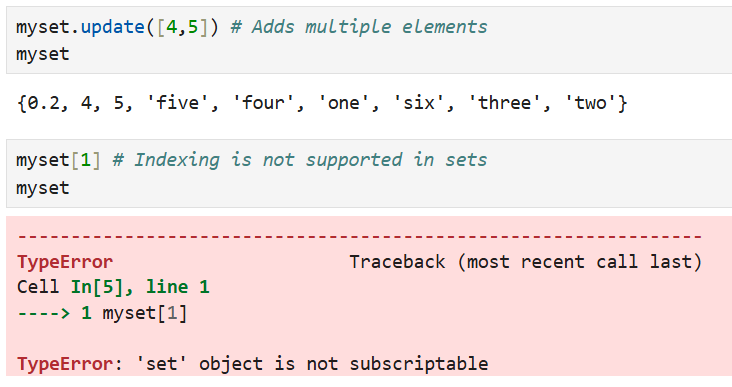
**Adding Elements:**

**.add()** method is used to add a single element to the set.

**.update()** method is used to add multiple elements(from another list, set, or any iterable).

Example:





**Removing Elements:**

**.remove()** is used to remove a specific element from set. This method raises a **keyError** if the element doesn’t exist in the set.

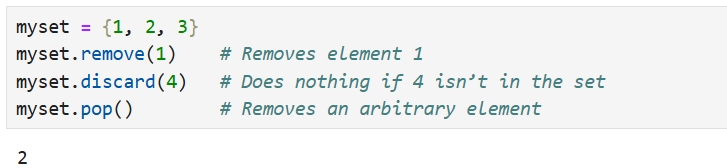
**.discard()** is used to remove an element without raising an error if the element isn’t found.

**.pop()** is used to remove and return an arbitrary element, which can be useful when clearing a set one item at a time.

An arbitrary element is any random element from the set, with no assurance as to which one will be selected, in the context of Python's pop() method for sets.

**.clear()** is used to delete all elements in a set.

Example:



**Differentiation between Set, List and Tuple:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Set** | **List** | **Tuple** |
| **Definition** | Unordered collection of unique elements. | Ordered collection of elements. | Ordered collection of elements. |
| **Syntax** | {element1, element2,…}  or set() | [element1, element2,…] | (element1, element2,…) |
| **Mutable** | Yes (elements can be added/removed.) | Yes (elements can be added/removed.) | No (elements cannot be changed.) |
| **Use Case** | Ideal for unique items and set operations. | Ideal for ordered, mutable collections. | Ideal for ordered, immutable collections. |

**Set Operations:**

There are 3 types of set operations:

1. Union (AUB or A|B):

It returns all the elements available in 2 or more sets.

Let’s suppose,

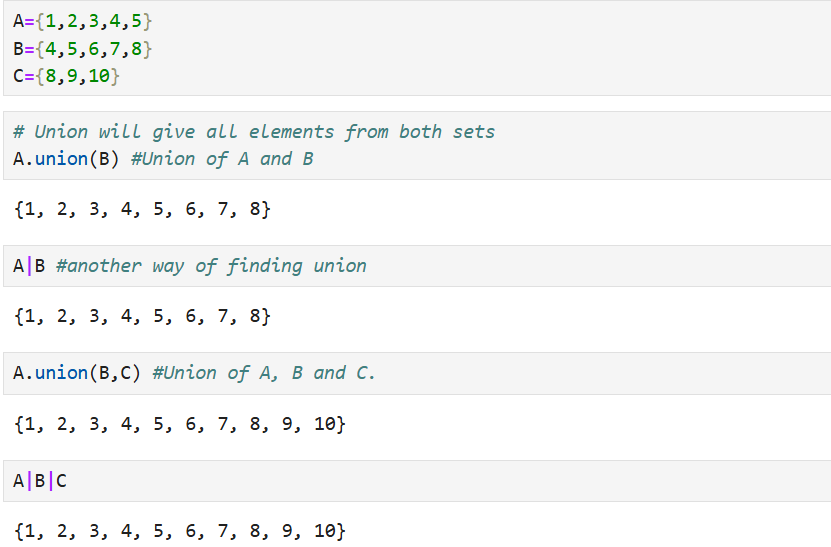
A= {1,2,3,4}

B= {4,5,6,1}

A.union(B)

The result would be all the elements available in set A and B ignoring the duplicates

#Output: {1,2,3,4,5,6}



1. Intersection (A**∩**B or A&B):

It returns common element between two or more sets.

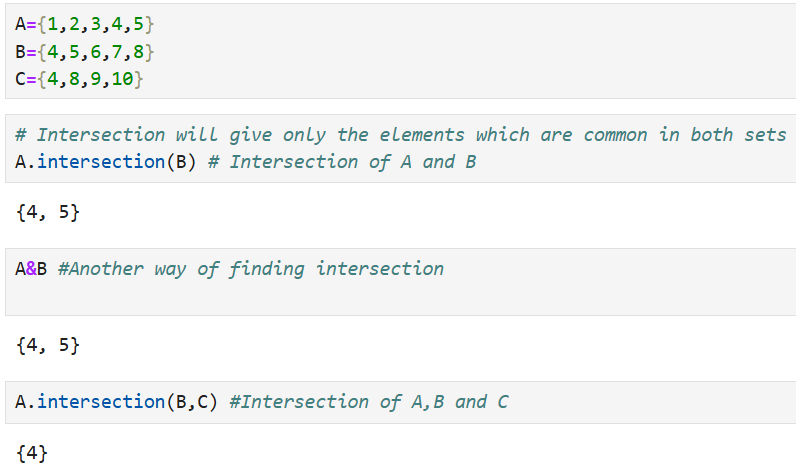
Example:

a= {1,2,3,4}

b= {4,5,6}

a.intersection(b)

#Output:{4}



1. Difference:

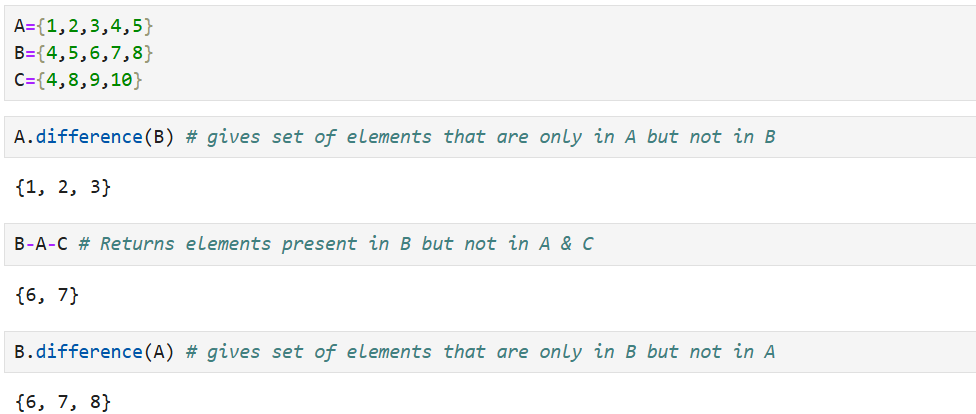
Difference returns the element present in the first set only.

Example:

a= {1,2,3,4}

b= {4,5,6}

a-b #returns element in a but not in b

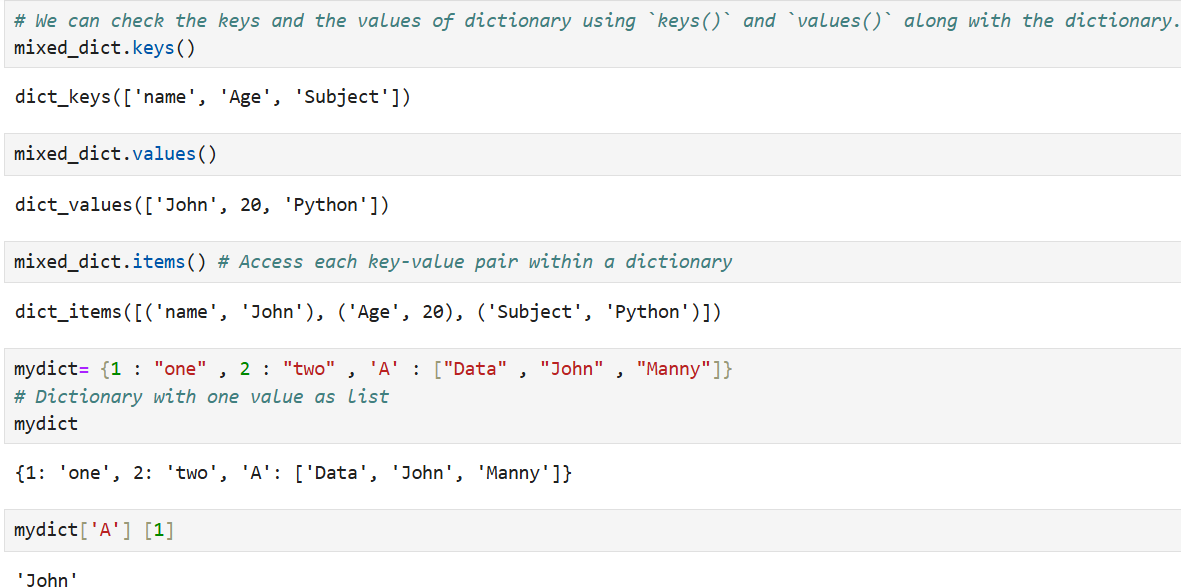
#Output: {1,2,3}

**Dictionary:**

A dictionary in Python is a data structure that stores the objects in **key-value** pairs. The retrieval of values is through the unique keys and thus helps in faster lookups. Dictionary is created using curly braces and key value pair within the braces.

Example:

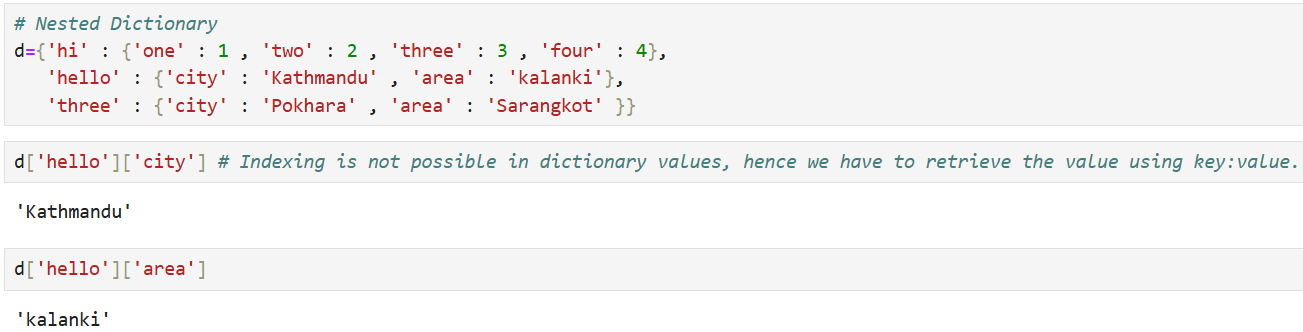




**Nested Dictionary:**

Dictionaries that are kept as values inside another dictionary are known as nested dictionaries.

Example:



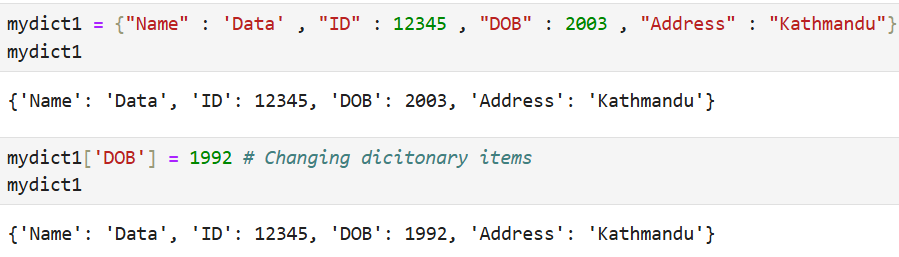
**Add, Change and Remove Items in Dictionary:**

**Update** function is used to add a new key : value pair in a dictionary.

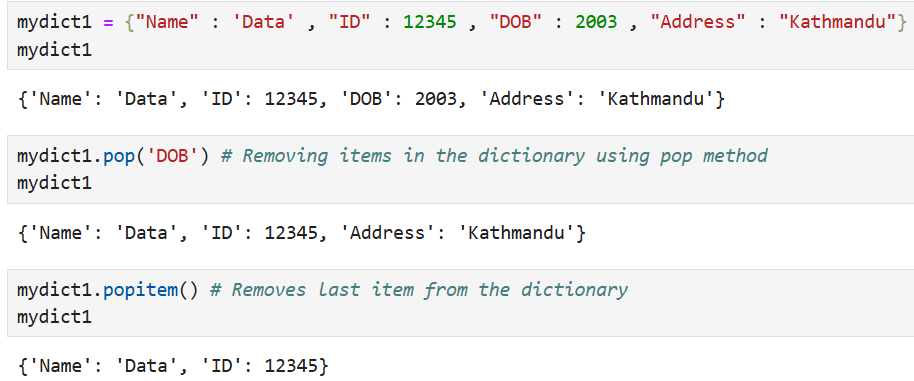
Example:

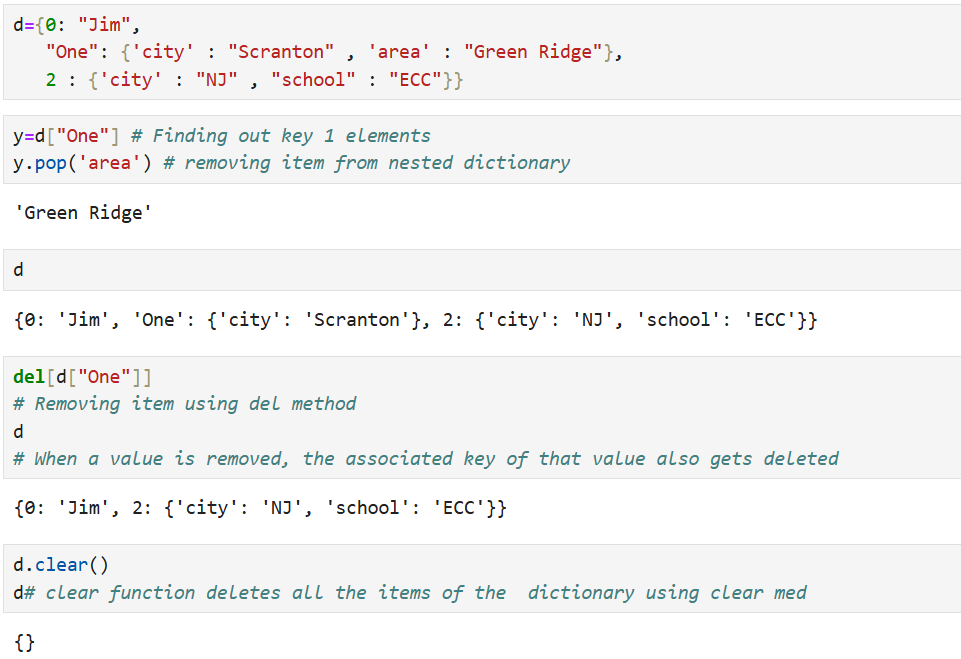
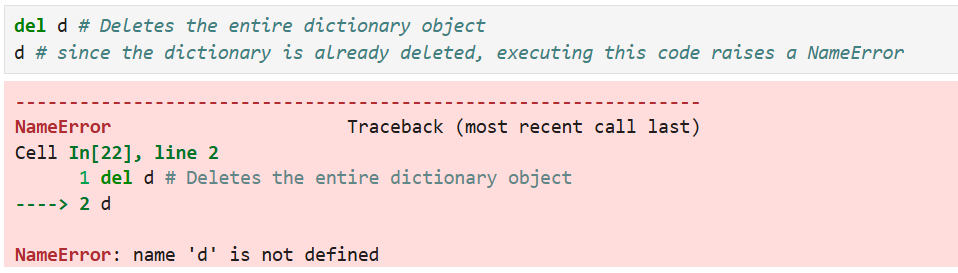


To **Change** the present data, following rules are followed as shown in an example:



To **Remove** certain data from the dictionary, **pop** is used.



To remove data from the nested dictionary,