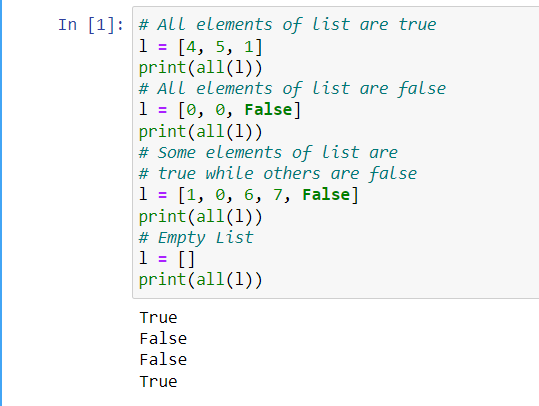
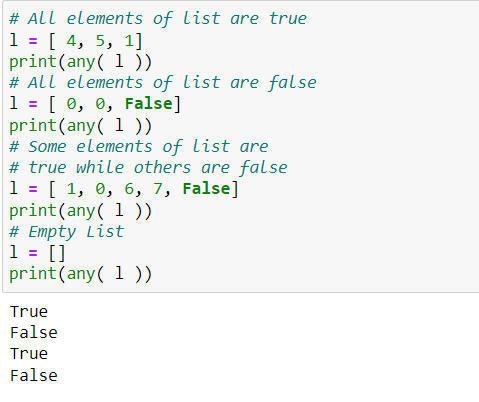
Built-in Functions & Methods in Python:

* List:

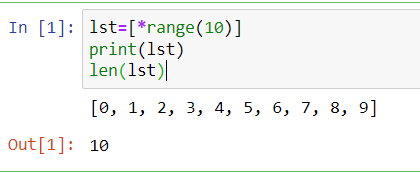
1. Python – all() function: The all() function is an inbuilt function in Python which returns true if all the elements of a given iterable( List, Dictionary, Tuple, set, etc) are True else it returns False. It also returns True if the iterable object is empty.



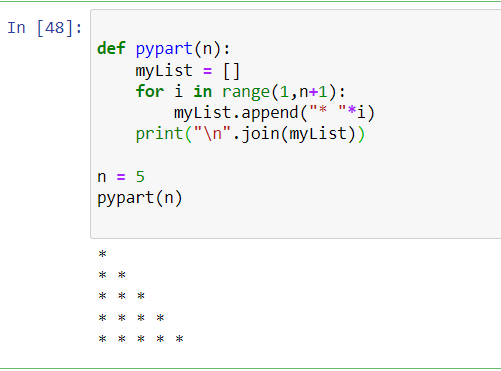
1. Python any() function: **Python any() function** returns True if any of the elements of a given iterable( List, Dictionary, Tuple, set, etc) are True else it returns False.



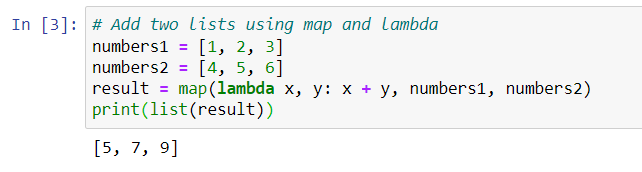
1. Python len() function: Python len() function returns the length of the string.



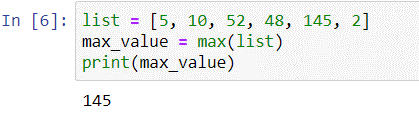
1. Python List append() : The append() method appends an element to the end of the list.



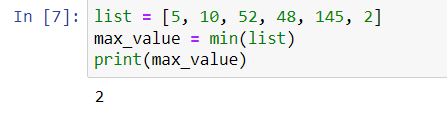
1. Python map() function: map() function returns a map object(which is an iterator) of the results after applying the given function to each item of a given iterable (list, tuple etc.)



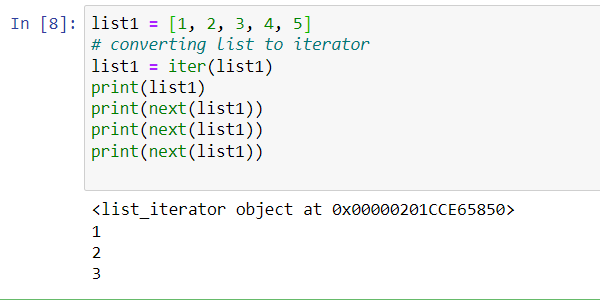
1. Python max() function: Python max() function returns the largest item in an iterable or the largest of two or more arguments.



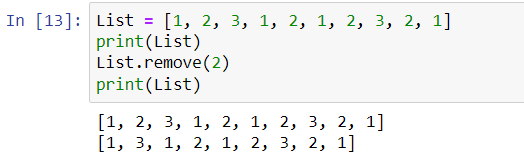
1. Python min() Function: Python min() function returns the smallest of the value or smallest item in an iterable passed as its parameter.



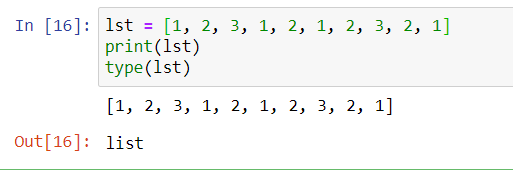
1. Python next() method: Python next() function returns the next item of an iterator. In this article, we will cover next() syntax, next() parameters, next() returns.



1. remove() : Element to be deleted is mentioned using list name and element.

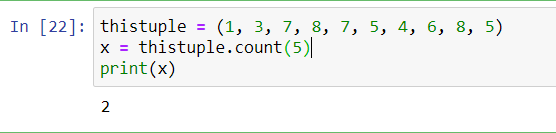


1. type() : type() method returns class type of the argument(object) passed as parameter. type() function is mostly used for debugging purposes.

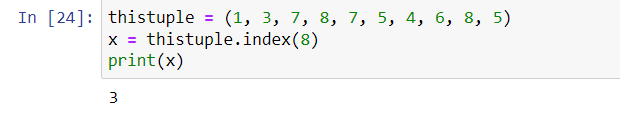


* Tuples:

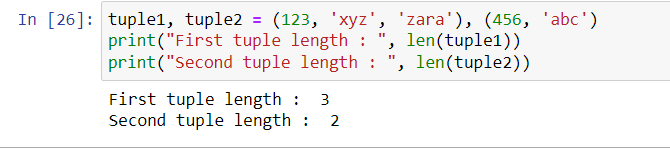
1. count() : The count() method returns the number of times a specified value appears in the tuple.



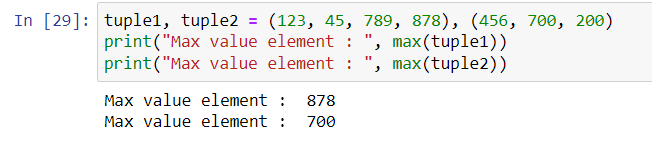
1. index() : The index() method finds the first occurrence of the specified value. The index() method raises an exception if the value is not found.



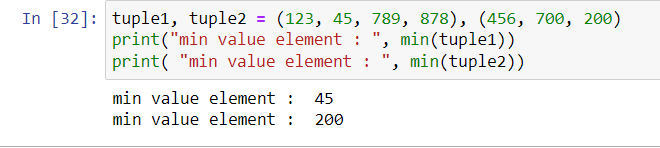
1. len() : Python tuple method len() returns the number of elements in the tuple.



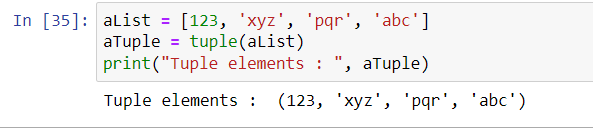
1. max() : Python tuple method max() returns the elements from the tuple with maximum value.



1. min() : Python tuple method min() returns the elements from the tuple with minimum value.

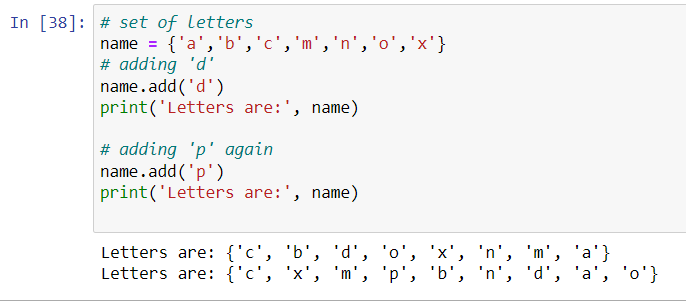


1. tuple() : Python tuple method tuple() converts a list of items into tuples.

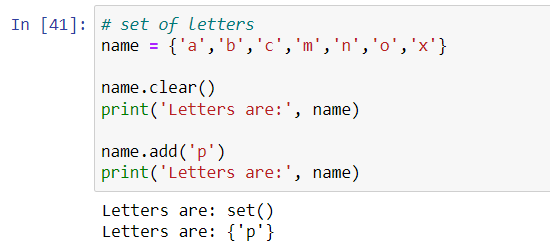


* Set:

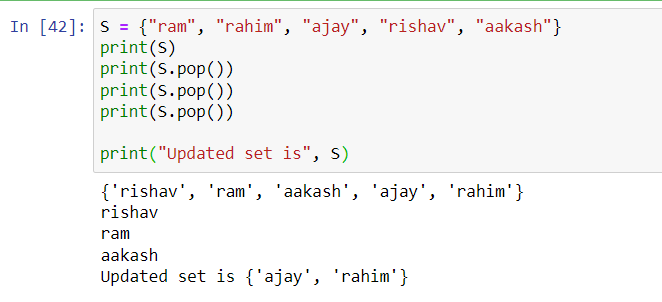
1. add() : The add() method adds an element to the set. If the element already exists, the add() method does not add the element.



1. clear() : The clear() method removes all elements from the set.



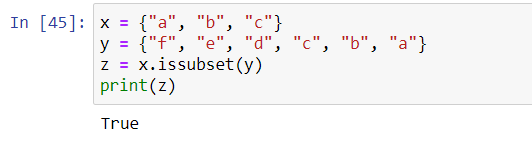
1. pop() : This in-built function of Python helps to pop out elements from a set just like the principle used in the concept while implementing Stack. This method removes a top element from the set but not the random element and returns the removed element.



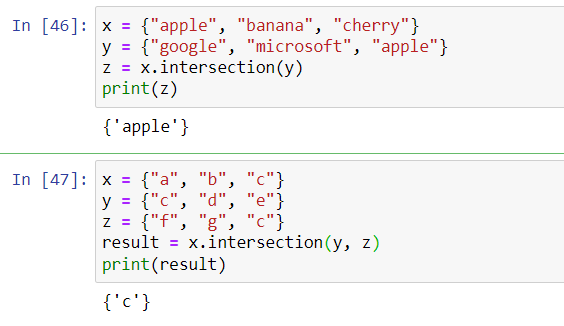
1. union() : The union() method returns a set that contains all items from the original set, and all items from the specified set(s).



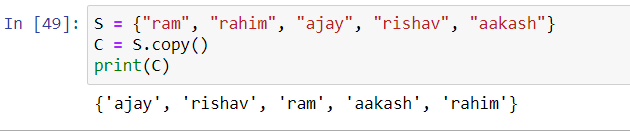
1. issubset() : The issubset() method returns True if all items in the set exists in the specified set, otherwise it retuns False.



1. intersection() : The intersection() method returns a set that contains the similarity between two or more sets.

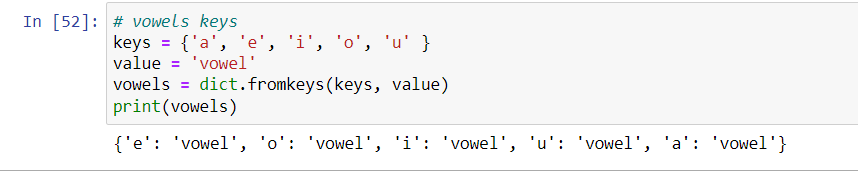


1. copy() : The copy() method copies the set.

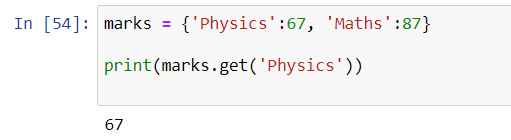


* Dictionary:

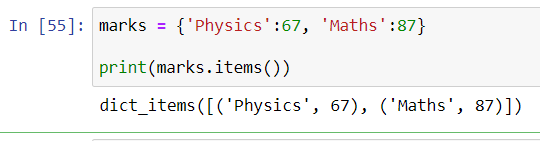
1. fromkeys() : The fromkeys() method returns a dictionary with the specified keys and the specified value.



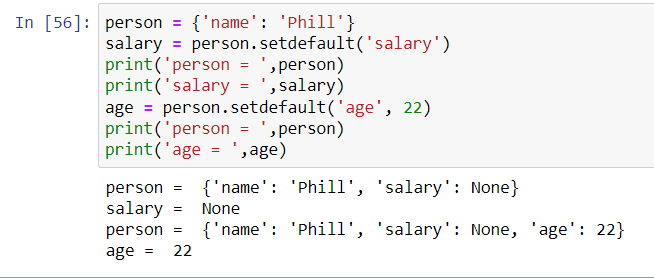
1. get() : The get() method returns the value of the item with the specified key.



1. items() : The items() method returns a view object. The view object contains the key-value pairs of the dictionary, as tuples in a list.



1. setdefault() : The setdefault() method returns the value of the item with the specified key.



|  |  |  |  |
| --- | --- | --- | --- |
| List | Tuple | Set | Dictionary |
| List is a non-homogeneous data structure that stores the elements in single row and multiple rows and columns | Tuple is also a non-homogeneous data structure that stores single row and multiple rows and columns | Set data structure is also non-homogeneous data structure but stores in single row | Dictionary is also a non-homogeneous data structure which stores key value pairs |
| List can be represented by  [ ] | Tuple can be represented by ( ) | Set can be represented by  { } | Dictionary  can be represented by { } |
| List allows duplicate elements | Tuple allows duplicate elements | Set will not allow duplicate elements | Set will not allow duplicate elements and dictionary doesn’t allow duplicate keys. |
| List can use nested among all | Tuple can use nested among all | Set can use nested among all | Dictionary can use nested among all |
| Example: [1, 2, 3, 4, 5] | Example: (1, 2, 3, 4, 5) | Example: {1, 2, 3, 4, 5} | Example: {1, 2, 3, 4, 5} |
| List can be created using list() function | Tuple can be created using tuple() function. | Set can be created using set() function | Dictionary can be created using dict() function. |
| List is mutable i.e we can make any changes in list. | Tuple  is immutable i.e we can not make any changes in tuple | Set is mutable i.e we can make any changes in set. But elements are not duplicated. | Dictionary is mutable. But Keys are not duplicated. |
| List is ordered | Tuple is ordered | Set is unordered | Dictionary is ordered (Python 3.7 and above) |
| Creating an empty list  l=[] | Creating an empty Tuple  t=() | Creating a set  a=set()  b=set(a) | Creating an empty dictionary  d={} |

**Difference Between -** List, Tuple, Set, Dictionary :