

List Methods

append()	Adds an element at the end of the list
clear()	Removes all the elements from the list
copy()	Returns a copy of the list
count()	Returns the number of elements with the specified value
extend()	Add the elements of a list (or any iterable), to the end of the current list
index()	Returns the index of the first element with the specified value
insert()	Adds an element at the specified position
pop()	Removes the element at the specified position
remove()	Removes the item with the specified value
reverse()	Reverses the order of the list
sort()	Sorts the list

Tuple Methods

count()	Returns the number of times a specified value occurs in a tuple
index()	Searches the tuple for a specified value and returns the position of where it was found

Set Methods

add()	Adds an element to the set
clear()	Removes all the elements from the set
copy()	Returns a copy of the set
difference()	Returns a set containing the difference between two or more sets
difference_update()	Removes the items in this set that are also included in another, specified set
discard()	Remove the specified item

intersection()	Returns a set, that is the intersection of two other sets
intersection_update()	Removes the items in this set that are not present in other, specified set(s)
isdisjoint()	Returns whether two sets have a intersection or not
issubset()	Returns whether another set contains this set or not
issuperset()	Returns whether this set contains another set or not
pop()	Removes an element from the set
remove()	Removes the specified element
symmetric_difference()	Returns a set with the symmetric differences of two sets
symmetric_difference_update()	inserts the symmetric differences from this set and another
union()	Return a set containing the union of sets
update()	Update the set with the union of this set and others

Operation	Mathematical Notation	Python Syntax	Result Type	Meaning
Union	$A \cup B$	<code>A B</code>	set	Elements in A or B or both
Intersection	$A \cap B$	<code>A & B</code>	set	Elements common to both A and B
Set Difference	$A - B$	<code>A - B</code>	set	Elements in A but not in B
Symmetric Difference	$A \oplus B$	<code>A ^ B</code>	set	Elements in A or B , but not both
Set Membership	$x \in A$	<code>x in A</code>	bool	x is a member of A
Set Membership	$x \notin A$	<code>x not in A</code>	bool	x is not a member of A
Set Equality	$A = B$	<code>A == B</code>	bool	Sets A and B contain exactly the same elements
Subset	$A \subseteq B$	<code>A <= B</code>	bool	Every element in set A also is a member of set B
Proper Subset	$A \subset B$	<code>A < B</code>	bool	A is a subset B , but B contains at least one element not in A

Dictionary Methods

<code>clear()</code>	Removes all the elements from the dictionary
<code>copy()</code>	Returns a copy of the dictionary
<code>fromkeys()</code>	Returns a dictionary with the specified keys and value
<code>get()</code>	Returns the value of the specified key
<code>items()</code>	Returns a list containing a tuple for each key value pair
<code>keys()</code>	Returns a list containing the dictionary's keys
<code>pop()</code>	Removes the element with the specified key
<code>popitem()</code>	Removes the last inserted key-value pair
<code>setdefault()</code>	Returns the value of the specified key. If the key does not exist: insert the key, with the specified value
<code>update()</code>	Updates the dictionary with the specified key-value pairs
<code>values()</code>	Returns a list of all the values in the dictionary-