**Inbuilt Functions for Different Datatypes**

**1. String Methods:**

1. capitalize() - Converts the first character to upper case
2. casefold() - Converts string into lower case
3. center() - Returns a centred string
4. count() - Returns the number of times a specified value occurs in a string
5. encode() - Returns an encoded version of the string
6. endswith() - Returns true if the string ends with the specified value
7. expandtabs() - Sets the tab size of the string
8. find() - Searches the string for a specified value and returns the position of where it was found
9. format() - Formats specified values in a string
10. format\_map() - Formats specified values in a string
11. index() - Searches the string for a specified value and returns the position from where it was found
12. isalnum() - Returns true if all characters in a string are alphanumeric
13. isalpha() - Returns true if all the characters in a string are in the alphabets
14. isascii() - Returns true if all the characters in the string arte ascii characters
15. isdecimal() - Returns true if all characters in the string are decimal
16. isdigit() - Returns true if all the characters in the string are digits
17. isidentifier() - Returns true if the string is an identifier
18. islower() - Returns true if all characters in a string are lower case
19. isnumeric() - Returns true if all characters in the string are numeric
20. isprintable() - Returns true if all the characters in the string are printable
21. isspace() - Returns true if all the characters in the string are whitespaces
22. istitle() - Returns true if the string follows the rules of a title
23. isupper() - Returns true if all the characters in the string are upper case
24. join() - converts the elements of an iterable into a string
25. ijust() - Returns a left justified version of the string
26. lower() - Converts a string into lowercase
27. istrip() -Returns a left trim version of the string
28. maketrans() - Returns a translation table to be used in transdlations
29. partition() - Returns a tuple where the string is parted into three parts
30. replace() - Returns a string where a specified value is replaced with a specified value
31. rfind() - Searches the string for a specified value and returns the last position of where it was found
32. rindex() - Searches the string for a specified value and returns the last position of where it was found
33. rjust() - Returns a right justified version of the string
34. rpartition() - Returns a tuple where the string is parted in three parts
35. rsplit() - Split the string at a specified separator, and returns a list
36. rstrip() - Returns a right trim version of the string
37. split() - Splits the string at a specified separator, and returns a list
38. splitlines() - Splits the string at line breaks and returns a list
39. startswith() - Returns true if the string starts with the specified value
40. strip() - Returns a trimmed version of the string
41. swapcase() - Swaps cases, lower case becomes upper case and vice versa
42. title() - Converts the first character of each word to upper case
43. translate() - Returns a translated string
44. upper() - Converts a string into uppercase
45. zfill() - Fills the string with a specified number of 0 values at the beginning

**2. List Methods:**

1. Append() - Adds an element at the end of the list
2. clear() - Removes all the elements from the list
3. copy() - Returns a copy of the list
4. count() - Returns a number of elements with the specified value
5. extend() - Add the elements of the list, to the end of the current list.
6. index() - Returns the index of the first element with the specified value
7. insert() - Adds an element at the specified position
8. pop() - Removes an element at a specified position
9. remove() - Removes the first item with the specified value
10. reverse() - Reverses the order of the list
11. sort() - Sorts the list

**3. Tuple Methods:**

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

**4. Set Methods:**

1. add() - Adds an element to the set
2. clear() - Removes all the elements from the set
3. copy() - Returns the copy of the set
4. difference() - Returns a set containing the difference between two or more sets.
5. difference\_update() - Removes the items in this set that are also included in another specified set.
6. discard() - removes the specified items.
7. intersection() - Returns a set that is the intersection of two or more sets.
8. intersection\_update() - Removes the items in this set
9. That are not present in other specified sets.
10. isdisjoint() - Returns whether two sets have a intersection or not.
11. issubset() - Returns whether another set contains this set or not.
12. isuperset() - Returns whether this set contains another set or not.
13. pop() - Removes an element from the set.
14. remove() - Removes the specified element
15. symmetric\_difference() - Returns a set with the symmetric differences of two sets.
16. symmetric\_diffference\_update() - Inserts the symmetric differences from this set and another.
17. union() - Return a set containing the union of sets.
18. update() - Update the set with another set, or any other iterable.

**5. Dictionary Methods:**

1. clear() - Removes all the elements from the dictionary
2. copy() - Returns a copy of the dictionary
3. fromkeys() - Returns a dictionary with the specified keys and value
4. get() - Returns the value of the specified key
5. items() - Returns a list containing a tuple for each key value pair
6. keys() - Returns a list containing dictionary’s keys
7. pop() - Removes the element with the specified key
8. popitem() - Removes the last inserted key-value pair
9. setdefault() - Returns the value of the specified key. If the key does not exist: inserts a key with the specified value
10. update() - Updates the dictionary with the specified key-value pairs
11. values() - Returns a list of all values in the dictionary