**PYTHON ESSENTIALS BATCH-7**

**//Q1 //**

* sort(): Sorts the list in ascending order.
* type(list): It returns the class type of an object.
* append(): Adds a single element to a list.
* extend(): Adds multiple elements to a list.
* index(): Returns the first appearance of the specified value.
* max(list): It returns an item from the list with max value.
* min(list): It returns an item from the list with min value.
* len(list): It gives the total length of the list.
* list(seq): Converts a tuple into a list.
* cmp(list1, list2): It compares elements of both lists list1 and list2

**//Q2 //**

clear(): Removes all items from the dictionary.

copy(): Returns a shallow copy of the dictionary.

fromkeys(seq[, v]): Returns a new dictionary with keys from seq and value equal to v (defaults to None).

get(key[,d]): Returns the value of the key. If the key does not exist, returns d (defaults to None).

items(): Return a new object of the dictionary's items in (key, value) format.

keys(): Returns a new object of the dictionary's keys.

pop(key[,d]): Removes the item with the key and returns its value or d if key is not found. If d is not provided and the key is not found, it raises KeyError.

popitem(): Removes and returns an arbitrary item (key, value). Raises KeyError if the dictionary is empty.

setdefault(key[,d]): Returns the corresponding value if the key is in the dictionary. If not, inserts the key with a value of d and returns d (defaults to None).

update([other]): Updates the dictionary with the key/value pairs from other, overwriting existing keys.

values(): Returns a new object of the dictionary's values

**//Q3 //**

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

**//Q4//**

cmp(tuple1, tuple2): Compares elements of both tuples.

len(tuple): Gives the total length of the tuple.

max(tuple): Returns item from the tuple with max value.

min(tuple): Returns item from the tuple with min value.

tuple(seq): Converts a list into tuple.

**//Q5 //**

capitalize(): Converts the first character to upper case

casefold(): Converts string into lower case

center(): Returns a centered string

count(): Returns the number of times a specified value occurs in a string

encode(): Returns an encoded version of the string

endswith(): Returns true if the string ends with the specified value

expandtabs(): Sets the tab size of the string

find(): Searches the string for a specified value and returns the position of where it was found

format(): Formats specified values in a string

format\_map(): Formats specified values in a string

index(): Searches the string for a specified value and returns the position of where it was found

isalnum(): Returns True if all characters in the string are alphanumeric

isalpha(): Returns True if all characters in the string are in the alphabet

isdecimal(): Returns True if all characters in the string are decimals

isdigit(): Returns True if all characters in the string are digits

isidentifier(): Returns True if the string is an identifier

islower(): Returns True if all characters in the string are lower case

isnumeric(): Returns True if all characters in the string are numeric

isprintable(): Returns True if all characters in the string are printable

isspace(): Returns True if all characters in the string are whitespaces

istitle(): Returns True if the string follows the rules of a title

isupper(): Returns True if all characters in the string are upper case

join(): Joins the elements of an iterable to the end of the string

ljust(): Returns a left justified version of the string

lower(): Converts a string into lower case

lstrip(): Returns a left trim version of the string

maketrans(): Returns a translation table to be used in translations

partition(): Returns a tuple where the string is parted into three parts

replace(): Returns a string where a specified value is replaced with a specified value

rfind(): Searches the string for a specified value and returns the last position of where it was found

rindex(): Searches the string for a specified value and returns the last position of where it was found

rjust(): Returns a right justified version of the string

rpartition(): Returns a tuple where the string is parted into three parts

rsplit(): Splits the string at the specified separator, and returns a list

rstrip(): Returns a right trim version of the string

split(): Splits the string at the specified separator, and returns a list

splitlines(): Splits the string at line breaks and returns a list

startswith(): Returns true if the string starts with the specified value

strip(): Returns a trimmed version of the string

swapcase(): Swaps cases, lower case becomes upper case and vice versa

title(): Converts the first character of each word to upper case

translate(): Returns a translated string

upper() : Converts a string into upper case

zfill(): Fills the string with a specified number of 0 values at the beginning

**SUBMITTED BY,**

**SUTHIR S.**