1. **A=A+1** evaluates to finding A, adding 1 to it. Then storing the value again in variable A. This expression makes Python to look for memory holder of a twice. But A+=1 simply means value of A is to be incremented by 1. As memory address has to be identified once, += leads to faster operation.

2. Minimum number of lines required to write above code in languages other Python will be 4, two for assigning initial values for variables a and b, and two for reassignment i.e. a=a+b and b=a.

3. The Most effective way to set a list of 100 integers to 0 in python is by using repetition operator (\*) or by using list comprehension.

Example:

list\_zero**=** [0]**\***100

print(list\_zero)

4. my\_list **=** [1,2,3]**\***33

print(my\_list)

5. my\_list **=** [[1,1],[2,2],[3,3],[4,4],[5,5]] *# 2 dimensional List*

**for** x **in** range(len(my\_list)):

**for** y **in** range(len(my\_list[x])):

print(my\_list[x][y],end**=**" ")

Output: 1 1 2 2 3 3 4 4 5 5

6. List comprehension with string is possible.

Example:

my\_list **=** [ele **for** ele **in** 'iNeuron']

print(my\_list)

Output: ['i', 'N', 'e', 'u', 'r', 'o', 'n']

7. **Get support with a user-written Python Programme:** Start a command prompt (Windows) or terminal window (Linux/Mac). If the current working directory is the same as the location in which you saved the file, you can simply specify the filename as a command-line argument to the Python interpreter.

**Get support with a User-written Python Program from IDLE:** You can also create script files and run them in IDLE. From the Shell window menu, select **File → New File**. That should open an additional editing window. Type in the code to be executed. From the menu in that window, **select File → Save or File → Save As…** and save the file to disk. Then **select Run → Run Module**. The output should appear back in the interpreter.

8. The tasks which can be performed with the functions in python are:

* A function is an instance of the Object type.
* You can store the function in a variable.
* You can pass the function as a parameter to another function.
* You can return the function from a function.
* You can store them in data structures such as hash tables, lists.

9. Wrappers around the functions are known as Decorators.

10. Generator functions are a special kind of function that return a **lazy iterator**. These are objects that you can loop over like a list. However, unlike lists, lazy iterators do not store their contents in memory.

11. Generator is a written as normal function but uses **yield** keyword to return values instead of **return** keyword.

12. **return** statement sends a specified value back to its caller whereas **yield** statement can produce a sequence of values. We should use generator when we want to iterate over a sequence, but don’t want to store the entire sequence in memory.