1. += is fatser and easy to write for us. runtime of a += 5 is slightly less than a = a + 5 but, the difference is really minor.

2. Two

3. l = [0] \* 100

print(l)

4. l1 = [1,2,3] # create a list containing 1,2,3

count = int(101/len(l1)) # repeat count

l = l1 \* count # \* operator is used to repeat the list

print(l)

5. We can print multidimentional list by simply writing the name of the variable in a cell

l = [[1,2,3,4],['A','B'],[5,6,7,8]]

l

6. Yes, it is possible as string is an iterable object. Explained below::

s = 'akash'

[i\*2 for i in s]

7. Steps for command line:

1. Open command prompt.

2. Go to the directory containing python program file(.py)

3. Type "python file\_name.py", press enter to run the file.

Steps for python IDLE:

1. Open python IDLE.

2. Click on file>open>file\_name.py

3. Press F5 key to run the program.

8. In Python, functions are like any other object, such as an int or a list. That means that you can use functions as arguments to other functions, store functions as dictionary values, or return a function from another function. This leads to many powerful ways to use functions

9. A wrapper, a wrapped feature, and a decorator are all same. They can be thought of as functions which modify the functionality of another function. They help to make your code shorter and more "Pythonic".

10. If a function is a generator function, it does not return any values, instead, it return an iterator object, which we can iterate through to get values.

11. We can use "yield" keyword instead of "return" to make function a generator.

def even\_nos(n):

for i in range(n):

if i % 2 == 0:

yield i

for i in even\_nos(10):

print(i)

12. The main advantage here is that instead of having to compute an entire series of values up front, the generator computes one value and then suspends its activity awaiting the next instruction. This feature is known as state suspension.