**Python Advanced Assignment 14**

Q1. Is an assignment operator like += only for show? Is it possible that it would lead to faster results

at the runtime?

Ans-) The += assignment operator is not only for show; it can result in faster results at runtime in some cases. This is because it performs the addition in place, modifying the existing object, rather than creating a new object.

Q2. What is the smallest number of statements you’d have to write in most programming languages to replace the Python expression a, b = a + b, a?

Ans-) In most programming languages, it would take three statements to replace the Python expression a, b = a + b, a. This is because you would need to use a temporary variable to store the value of a + b before reassigning it to a and b separately.

Q3. In Python, what is the most effective way to set a list of 100 integers to 0?

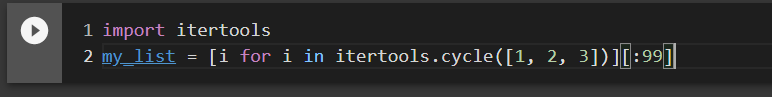
Ans-) The most effective way to set a list of 100 integers to 0 in Python is to use a list comprehension:

**my\_list = [0] \* 100**

Q4. What is the most effective way to initialise a list of 99 integers that repeats the sequence 1, 2, 3?

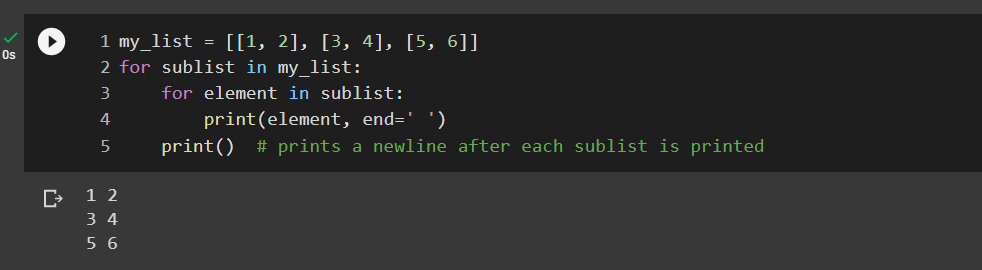
S If necessary, show step-by-step instructions on how to accomplish this.

Ans-) The most effective way to initialise a list of 99 integers that repeats the sequence 1, 2, 3 is to use the itertools library's cycle function in combination with a list comprehension:



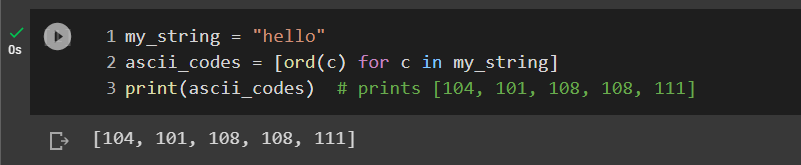
Q5. If you’re using IDLE to run a Python application, explain how to print a multidimensional list as efficiently?

Ans-) To print a multidimensional list efficiently in IDLE, you can use a nested loop to iterate over each element in the list and print it:



Q6. Is it possible to use list comprehension with a string? If so, how can you go about doing it?

Ans-) Yes, it is possible to use list comprehension with a string in Python. For example, you could use list comprehension to create a list of the ASCII codes for each character in a string:



Q7. From the command line, how do you get support with a user-written Python programme? Is this

possible from inside IDLE?

Ans-) To get support with a user-written Python program from the command line, you can use the -h or --help flag to display the program's help message and options. For example:

**python my\_program.py --help**

Q8. Functions are said to be “first-class objects” in Python but not in most other languages, such as

C++ or Java. What can you do in Python with a function (callable object) that you can&#39;t do in C or C++?

Ans-) In Python, functions are considered first-class objects because they can be assigned to variables, passed as arguments to other functions, and returned as values from functions. This allows for more flexible and dynamic programming than is possible in languages like C or C++, where functions are treated as lower-level constructs.

Q9. How do you distinguish between a wrapper, a wrapped feature, and a decorator?

Ans-) In Python, a wrapper is a function that takes another function as input and returns a modified version of that function. The wrapped feature is the original function that is being modified by the wrapper. A decorator is a special type of wrapper that uses the @decorator\_name syntax to apply the wrapper to a function automatically.

Q10. If a function is a generator function, what does it return?

Ans-) If a function is a generator function in Python, it returns a generator object. This object can be iterated over using a for loop, and it generates a series of values on the fly rather than returning them all at once like a regular function.

Q11. What is the one improvement that must be made to a function in order for it to become a

generator function in the Python language?

Ans-) The one improvement that must be made to a function to become a generator function in Python is to replace the return keyword with the yield keyword. This allows the function to generate a series of values rather than returning a single value.

Q12. Identify at least one benefit of generators.

Ans-) One of the main benefits of generators is that they allow for lazy evaluation, which means that they generate values only when they are needed. This can lead to significant improvements in memory usage and performance for large data sets or when working with infinite sequences. Additionally, generators can be more expressive and easier to read than equivalent code using lists or other data structures.