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

**Ans:** += adds a number to a variable, changing the variable itself in the process (whereas + would not). ... \*=, multiplies the variable and a value, making the outcome the variable. /=, divides the variable by the value, making the outcome the variable.

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

**Ans:** A simple statement is comprised within a single logical line. Several simple statements may occur on a single line separated by semicolons.

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

**Ans:** 1. Using range. The range () function returns a sequence of numbers, starting from 100 by default, and decrements by 1 ending at a specified number.

1. Using Rand range. The random module can also generate a random number between in a similar way as above.
2. With NumPy. arrange.

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:** You can use a for loop to create a list of elements in three steps: Instantiate an empty list. Loop over an iterable or range of elements. Append each element to the end of the list.

1. Using range. The range () function returns a sequence of numbers, starting from 100 by default, and decrements by 1 ending at a specified number.
2. Using Rand range. The random module can also generate a random number between in a similar way as above.
3. With NumPy. arrange.

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

**Ans:** In Python, you have different ways to specify a multiline string. You can have a string split across multiple lines by enclosing it in triple quotes. Alternatively, brackets can also be used to spread a string into different lines. Moreover, backslash works as a line continuation character in Python.

1. Approach 1: Accessing with the help of loop.
2. Approach 2: Accessing using square brackets. ...
3. append (): Adds an element at the end of the list. ...
4. extend (): Add the elements of a list (or any iterable), to the end of the current list.
5. reverse (): Reverses the order of the list.

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

**Ans:** We can use an if-else in a list comprehension in Python. Since in a comprehension, the first thing we specify is the value to put in a list, this is where we put our if else. This code stores in a list, for each integer from 0 to 7, whether it is even or odd. List comprehensions provide a concise way to create lists. Common applications are to make new lists where each element is the result of some operations applied to each member of another sequence or iterable, or to create a subsequence of those elements that satisfy a certain condition.

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

**Ans:** To run Python scripts with the python command, you need to open a command-line and type in the word python, or python3 if you have both versions, followed by the path to your script, just like this: $ python3 hello.py Hello World!

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 cannot do in C or C++?

**Ans:** In Python, you can take advantage of special capabilities of strings and other fundamental types (lists, dictionaries) that are built deeply into the language itself. Therefore, you can utilize special syntax such as “slicing” that you could not express as succinctly in C++, even if you went to great length to try to support all the functionality yourself.

You can get information about all Python objects at run time, which includes functions in the program — considered first-class objects. Although C++ has function pointers as well as “callable objects,” the ability to inspect functions (and rename them!) at run time does not exist in C++ the way it does in Python.

To achieve anything like that in C++, you would have to essentially build a whole Python-like environment yourself. I suppose anything is possible… but certain problems have been solved for you in Python, so you do not have to solve them yourself.

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

**Ans:** The two are not necessarily different things. Decorators are a kind of wrapper. The other kind are adapters. So basically, wrapper is simply a function that calls the original function with some extra neat stuff added to it.

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

**Ans:** Python generators are a simple way of creating iterators. All the work we mentioned above are automatically handled by generators in Python. Simply speaking, a generator is a function that returns an object (iterator) which we can iterate over (one value at a time).

Q11. What is the one improvement that must be made to a function for it to become a generator function in the Python language?

**Ans:** If a function contains at least one yield statement (it may contain other yield or return statements), it becomes a generator function. Both yield and return will return some value from a function.

Q12. Identify at least one benefit of generators.

**Ans:** Generators have been an important part of Python ever since they were introduced with PEP 255. Generator functions allow you to declare a function that behaves like an iterator. They allow programmers to make an iterator in a fast, easy, and clean way.