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

ANSWER.

No, an assignment operator like `+=` is not just for show; it serves a practical purpose in Python and can indeed lead to faster results at runtime in certain scenarios.

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?

ANSWER.

In most programming languages, you would typically need three statements to replace the Python expression `a, b = a + b, a`. These three statements would involve temporary variables to store intermediate values during the assignment.

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

ANSWER.

The most effective way to set a list of 100 integers to 0 in Python is to use list multiplication with a singleton list containing the value 0. This approach leverages Python's list multiplication feature to create a new list with the desired number of elements, each initialized to the specified value.

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.

ANSWER.

The most effective way to initialize a list of 99 integers that repeats the sequence 1, 2, 3 is to use list comprehension with the modulo operator (`%`). This approach allows you to generate the desired sequence dynamically in a concise and efficient manner.

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

ANSWER.

Printing a multidimensional list efficiently in IDLE can be accomplished by using nested loops to iterate over each dimension of the list and print each element individually. Additionally, you can use formatting options to control the appearance of the output.

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

ANSWER.

Yes, it is possible to use list comprehension with a string in Python. List comprehension can be used to create a list from the characters of a string or to transform each character of a string based on certain conditions.

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

ANSWER.

From the command line, you can get support with a user-written Python program by including help messages, documentation strings (docstrings), or usage instructions within the code itself. These messages provide guidance on how to use the program, its functionalities, command-line arguments, and any other relevant information.

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't do in C or C++?

ANSWER.

In Python, functions are considered first-class objects, which means they can be treated like any other object in the language. This provides several capabilities that are not typically available in languages like C or C++. Here are some things you can do with functions in Python that you can't do in C or C++:

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

ANSWER.

A wrapper is a construct that encapsulates or contains another function or object, a wrapped feature is the original function or object contained within a wrapper, and a decorator is a specific pattern in Python for dynamically extending or modifying the behavior of functions or methods. Decorators are a type of wrapper that use a specific syntax and pattern for applying additional functionality to functions or methods.

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

ANSWER.

A generator function in Python returns a generator object when called. Unlike regular functions that return a single value, generator functions return an iterator called a generator. This generator can then be iterated over to yield a sequence of values one at a time, lazily, on demand.

When you call a generator function, it doesn't execute the function body immediately. Instead, it returns a generator object, which is an iterator that can be iterated over using a loop, or by calling the `next()` function on it manually.

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?

ANSWER.

The one improvement that must be made to a function in order for it to become a generator function in Python is to use the `yield` statement instead of the `return` statement to yield values lazily.

Q12. Identify at least one benefit of generators.

ANSWER.

Generators offer a powerful and flexible tool for working with sequences of data in a memory-efficient and performance-conscious manner, making them invaluable in a wide range of applications, including data processing, algorithm design, and concurrent programming.