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

+= operator is not for it actually combines the assignment operator and addition operator in a single operator and hence it is faster than using x = x + 1.

**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?**

Temp = a

a=b

b += Temp

so minimum 3 steps are required to replace **a, b = a + b, a.**

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

[ 0 for x in range(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.**

X = [ 1,2,3 ]\*33 will create a list of integers of 99 integers which repeats the sequence 1,2,3

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

We can use list comprehension or loop to create multidimensional list

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

Yes it is very much possible to use list comprehension with string. We can get every character out of the string and perform the operation on it.

x = [i for i in "sample"]

x

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

We can use help() function or we can access the docstring by using \_\_doc\_\_

**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++?**

Python function are called as first class function because

* Function can be assigned to variables
* We can pass functions as argument to other functions
* Function can return other function
* Create lambda functions

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

* a wrapper is a general concept of encapsulating functionality with an additional layer.
* a wrapped feature is the original functionality being wrapped, and
* a decorator is a specific implementation pattern in Python for wrapping functions or classes to modify their behavior.
* While wrappers and decorators share the idea of wrapping functionality, decorators are a more specialized form of wrapper typically used in Python for modifying or extending functions or classes.

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

In generator function we use yield keyword instead of return keyword. It returns a generator object which can be iterated by using next() method or by using loop.

**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?**

In generator function we use yield keyword instead of return keyword. It returns a generator object which can be iterated by using next() method or by using loop.

**Q12. Identify at least one benefit of generators.**

We use generators to save memory. Instead of saving all the elements of upfront generators functions provides one value at a time. Hence save a lot of memory and complexity.