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

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

The += operator is an in-place addition assignment operator that performs addition and assignment in a single step.
```

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=a+b
b=temp
```

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

In [3]:

```
v=0
l=[v]*100
print(l)
```

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0]
```

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.

In [4]:

```
l=list(range(1,100))
print(l)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 2
1, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39,
40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 5
8, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76,
77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 9
5, 96, 97, 98, 99]
```

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

In [7]:

```
l=[[1,2,3],[4,5,6]]

for i in l:
    print(*i)
```

```
1 2 3
4 5 6
```

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

In [8]:

```
# yes it is possible

s="ineuron"
l=[i for i in s]
print(l)
```

```
['i', 'n', 'e', 'u', 'r', 'o', 'n']
```

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

```
Yes we can do this
python -c "import file.py; help(file.py)"
```

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

or Java. What can you do in Python with a function (callable object) that you can't do in C orC++?

```
In C or C++, defining functions inside other functions is not supported. where as we can do it in pyhton.
wherer as in phton when pass an object dietly that object is passeed.in Other lanh=guages the copy of that object is passed
```

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

```
Wrapper: A wrapper is a function or class that wraps around another function or object, providing additional
    functionality or modifying the behavior of the wrapped entity

Wrapped Feature: The wrapped feature refers to the original function or object that is being wrapped or modified
    by the wrapper

Decorator: A decorator is a special kind of wrapper that allows you to modify the behavior of a function or class
    using a specific syntax. In Python, decorators are denoted by the @decorator_name syntax placed above the definition
    of the function or class being decorated
```

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

```
A generator object which is iterable with for loop like list
and eery element canbe accessed using next()
```

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?

```
Inplace of return just use yield keyword
```

Q12. Identify at least one benefit of generators.

```
It do not return entire sequence .
It will return wlements on demand which reduce the time comolexity of code
```