1) How does break, continue and pass work?

Using loops in Python automates and repeats the tasks in an efficient manner. But sometimes, there may arise a condition where you want to exit the loop completely, skip an iteration or ignore that condition. These can be done by loop control statements. Loop control statements change execution from its normal sequence. When execution leaves a scope, all automatic objects that were created in that scope are destroyed. Python supports the following control statements.

Break statement - The break statement is used to terminate the loop or statement in which it is present. After that, the control will pass to the statements that are present after the break statement, if available. If the break statement is present in the nested loop, then it terminates only those loops which contains break statement. Syntax - break

Continue statement - Continue is also a loop control statement just like the break statement. continue statement is opposite to that of break statement, instead of terminating the loop, it forces to execute the next iteration of the loop. Syntax - continue

As the name suggests the continue statement forces the loop to continue or execute the next iteration. When the continue statement is executed in the loop, the code inside the loop following the continue statement will be skipped and the next iteration of the loop will begin.

Pass statement - As the name suggests pass statement simply does nothing. The pass statement in Python is used when a statement is required syntactically but you do not want any command or code to execute. It is like null operation, as nothing will happen is it is executed. Pass statement can also be used for writing empty loops. Pass is also used for empty control statement, function and classes. Syntax – pass

2) What is the difference between list and tuples in Python?

List	Tuples
Lists are mutable	Tuples are immutable
Implication of iterations is Time-consuming	The implication of iterations is comparatively Faster
The list is better for performing operations, such as insertion and deletion.	Tuple data type is appropriate for accessing the elements
Lists consume more memory	Tuple consume less memory as compared to the list
Lists have several built-in methods	Tuple does not have many built-in methods.
The unexpected changes and errors are more likely to occur	In tuple, it is hard to take place.

3) What are functions in Python?

Python Functions is a block of related statements designed to perform a computational, logical, or evaluative task. The idea is to put some commonly or repeatedly done tasks together and make a function so that instead of writing the same code again and again for different inputs, we can do the function calls to reuse code contained in it over and over again. Functions can be both built-in or user-defined. It helps the program to be concise, non-repetitive, and organized.

- We can create a Python function using the def keyword.
  # A simple Python function

  def fun():
    print("Welcome to GFG")

- After creating a function, we can call it by using the name of the function followed by parenthesis containing parameters of that particular function.
  # A simple Python function

  def fun():
    print("Welcome to GFG")
  # Driver code to call a function
  fun()

Arguments of a Function

Arguments are the values passed inside the parenthesis of the function. A function can have any number of arguments separated by a comma.

A simple Python function to check

whether x is even or odd

def evenOdd(x):

  if (x % 2 == 0):

    print("even")

  else:

    print("odd")

Driver code to call the function

evenOdd(2)

evenOdd(3)

4) What is a lambda function?

Python Lambda Functions are anonymous function means that the function is without a name. As we already know that the def keyword is used to define a normal function in Python. Similarly, the lambda keyword is used to define an anonymous function in Python.

Python Lambda Function Syntax:

lambda arguments: expression

This function can have any number of arguments but only one expression, which is evaluated and returned. One is free to use lambda functions wherever function objects are required. You need to keep in your knowledge that lambda functions are syntactically restricted to a single expression. It has various uses in particular fields of programming besides other types of expressions in functions.

5) How can you generate random numbers in Python?

Python defines a set of functions that are used to generate or manipulate random numbers through the random module. Random number generation is the process of producing random numbers whenever required. These numbers can be generated with or without giving conditions. For example, setting the range of numbers from 1 to 100, asking for only even numbers, etc.

Random number generation is one of the most frequently used techniques in programming when we need to input a large number of entities. Python is known for its collection of standard libraries and functions, and similarly, we also have a set of functions that can be used to generate random numbers.

choice() - choice() is an inbuilt function in the Python programming language that returns a random item from a list, tuple, or string.

random() - This method is used to generate a float random number less than 1 and greater or equal to 0.

shuffle() - The shuffle() function is used to rearrange all the values present in the provided list randomly.

randint(beginning, end) - The randint() function takes two arguments, beginning, and end, that denote the starting and ending numbers. It produces random integer type numbers between the given starting and ending numbers.

randrange(beginning, end, step) - The randrange() function has three arguments- beginning, end, and step, denoting the starting number, ending number, and the steps between numbers. It generates random numbers between the range of beginning and ending number input, in a periodic interval equal to the provided steps.

6) What is the difference between range & xrange?

The range() and xrange() are two functions that could be used to iterate a certain number of times in for loops in Python. In Python 3, there is no xrange, but the range function behaves like xrange in Python 2. If you want to write code that will run on both Python 2 and Python 3, you should use range().

range() – This returns a range object (a type of iterable).

xrange() – This function returns the generator object that can be used to display numbers only by looping. The only particular range is displayed on demand and hence called “lazy evaluation“.

range	Xrange
Returns a list of integers.	Returns a list of integers.
Execution speed is slower	Execution speed is faster.
Takes more memory as it keeps the entire list of elements in memory.	Takes less memory as it keeps only one element at a time in memory.
All arithmetic operations can be performed as it returns a list.	Such operations cannot be performed on xrange().
In python 3, xrange() is not supported.	In python 2, xrange() is used to iterate in for loops.

7) How do you write comments in Python?

Comments in Python are the lines in the code that are ignored by the compiler during the execution of the program. Comments enhance the readability of the code and help the programmers to understand the code very carefully. There are three types of comments in Python: Single line Comments, Multiline Comments, Docstring Comments

Eg:

```
# Python program to demonstrate comments

# sample comment

name = "geeksforgeeks"

print(name)
```

Output:

```
geeksforgeeks
```

In the above example, it can be seen that comments are ignored by the compiler during the execution of the program.

Comments are generally used for the following purposes:

- Code Readability
- Explanation of the code or Metadata of the project
- Prevent execution of code
- To include resources