1. **What is __init__?**

__init__" is a reseved method in python classes. It is known as a constructor in object oriented concepts. This method called when an object is created from the class and it allow the class to initialize the attributes of a class.

2. **What is self in Python?**

self represents the instance of the class. By using the "self" keyword we can access the attributes and methods of the class in python. It binds the attributes with the given arguments.

3. **How can you randomize the items of a list in place in Python?**

The method shuffle() can be used to randomize the items of a list in place. It should be noted that this function is not accessible directly and therefore we need to import or call this function using random static object.

Syntax: shuffle (lst)

Here, 'lst' is passed as a parameter which could be a list or tuple. The shuffle() returns a reshuffled list of items.

4. **What are Python iterators?**

Iterator in python is an object that is used to iterate over iterable objects like lists, tuples, dicts, and sets. The iterator object is initialized using the iter() method. It uses the next() method for iteration.

__iter(iterable)__ method that is called for the initialization of an iterator. This returns an iterator object.

next (__next__ in Python 3) The next method returns the next value for the iterable. When we use a for loop to traverse any iterable object, internally it uses the iter() method to get an iterator object which further uses next() method to iterate over. This method raises a StopIteration to signal the end of the iteration.

5. **What is pickling and unpickling?**

Pickling and unpickling are very important when we have to transfer Python objects from one machine to another and vice versa.

In Python, pickling is the process by which Python objects are converted to byte streams. Pickling is about serializing the object structure in python.

Unpickling is the process of retrieving original python objects from the stored string representation i.e from the pickle file. It is the process of converting a byte stream into the python object.

6. **What are the generators in python?**

Generators have been an important part of python ever since they were introduced with PEP 255.

Generator in python are special routine that can be used to control the iteration behaviour of a loop. A generator is similar to a function returning an array. A generator has parameter, which we can called and it generates a sequence of numbers. But unlike functions, which return a whole array, a generator yields one value at a time which requires less memory.

Any python function with a keyword "yield" may be called as generator. A normal python function starts execution from first line and continues until we got a return statement or an exception or end of the function however, any of the local variables created during the function scope are destroyed and not accessible further. While in case of generator when it encounters a yield keyword the state of the function is frozen and all the variables are stored in memory until the generator is called again.

7. How will you capitalize the first letter of a string?

str.capitalize() to capitalize the first letter of a string in python:

Syntax: string.capitalize()

Parameters: no parameters

Return Value: string with the first capital first letter

8. Difference between process and thread?

Process

Process are basically the programs which are dispatched from the ready state and are scheduled in the CPU for execution. PCB (Process Control Block) holds the concept of process. A process can create other processes which are known as Child Processes. The process takes more time to terminate and it is isolated means it does not share the memory with any other process. The process can have the following states like new, ready, running, waiting, terminated, suspended.

Thread

Thread is the segment of a process means a process can have multiple threads and these multiple threads are contained within a process. A thread has three states: Running, Ready, and Blocked. Thread takes less time to terminate as compared to process but unlike process threads do not isolate.

Process	Thread
A process is a program under execution i.e an active program.	A thread is a lightweight process that can be managed independently by a scheduler.
Processes require more time for context switching as they are more heavy.	Threads require less time for context switching as they are lighter than processes.
Processes are totally independent and don't share memory.	A thread may share some memory with its peer threads.
Communication between processes requires more time than between threads.	Communication between threads requires less time than between processes.
If a process gets blocked, remaining processes can continue execution.	If a user level thread gets blocked, all of its peer threads also get blocked.

Processes require more resources than threads.	Threads generally need less resources than processes.
Individual processes are independent of each other.	Threads are parts of a process and so are dependent.
Processes have independent data and code segments.	A thread shares the data segment, code segment, files etc. with its peer threads.
All the different processes are treated separately by the operating system.	All user level peer threads are treated as a single task by the operating system.
Processes require more time for creation.	Threads require less time for creation.
Processes require more time for termination.	Threads require less time for termination.

9. How to comment multiple lines in python?

Comments in programming languages are used to explain code and provide context. They often help make your code more readable by explaining the why of your code (i.e., why you are writing your code the way you are). Sometimes comments are also used to prevent code from running, and this is often referred to as “commenting code out”. You have a number of different options of where you place the pound symbol:

If you place the # symbol at the beginning of a line, the entire line becomes a comment

If you place the # somewhere along the line of code, then everything that follows it becomes a comment