1) What is the difference between enclosing a list comprehension in square brackets and parentheses?

Ans: If we enclose a list comprehension in square brackets the output will be returned as list datatype and if it is enclosed in the parentheses the output will be returned as set data type.

2) What is the relationship between generators and iterators?

Ans: A generator-function is defined like a normal function, but whenever it needs to generate a value, it does so with the yield keyword rather than return. If the body of a def contains yield, the function automatically becomes a generator function. Generator functions return a generator object. Generator objects are used either by calling the next method on the generator object or using the generator object in a “for in” loop. So a generator function returns an generator object that is iterable, i.e., can be used as an 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\_\_(): The iter() method is called for the initialization of an iterator. This returns an iterator object.

\_\_next\_\_(): 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 the next() method to iterate over. This method raises a StopIteration to signal the end of the iteration.

3) What are the signs that a function is a generator function?

Ans: A function is known as a generator function if it consists of Yield statement instead of return or print statement.

4) What is the purpose of a yield statement?

Ans: Yield statement will return the generator object. Using yield keyword is highly memory efficient, since the execution happens only when the caller iterates over the object. As the variables states are saved, we can pause and resume from the same point, thus saving time.

If you want to return multiple values from a function, you can use generator functions with yield keywords. The yield expressions return multiple values. They return one value, then wait, save the local state, and resume again.

5) What is the relationship between map calls and list comprehensions? Make a comparison and contrast between the two.

Ans:

1.List comprehension is more concise and easier to read as compared to map.

2.List comprehension allows filtering. In map, we have no such facility. For example, to print all even numbers in range of 100, we can write [n for n in range(100) if n%2 == 0]. There is no alternate for it in map.

3.List comprehension are used when a list of results is required as map only returns a map object and does not return any list.

4.List comprehension is faster than map when we need to evaluate expressions that are too long or complicated to express.

5.Map is faster in case of calling an already defined function (as no lambda is required).