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

Answer: Enclosing a list comprehension in square brackets will give function results as a list and enclosing in parentheses will become a generator.

l = [n\*2 for n in range(1000)] # List comprehension

g = (n\*2 for n in range(1000)) # Generator expression

2) What is the relationship between generators and iterators?

Answer: Python generator saves the state of local variables every time yield pauses the loop.

An iterator does not make use of local variables, all it needs iterable to iterate on. A generator can have any number of yield statements. Every generator object is an iterator but not vice-versa.

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

Answer: If a function contains at least one yield function then it’s a generator and it can have other yield or return statements.

4) What is the purpose of a yield statement?

Answer: Yield statement works similar as a return statement and used for returning values or objects of a function. Yield statement returns a generator object when called in a function.

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

Answer: List comprehension are concise, easier compared to map.

List comprehensions allow filtering but in map we don’t have such facility.

List comprehensions is used when a list of results is required whereas map function only returns a map object.

List comprehensions are faster compared to Map when we execute long/complicated expressions.

Map is faster while calling an already defined function.