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

**The difference between the two kinds of expressions is that the List comprehension in square brackets provides a list while in plain parentheses provides a Generator expression**

2) What is the relationship between generators and iterators?

**A generator is a construct in Python which typically lazily generates data (i.e. the next value is generated only when needed).**

**A generator is either a callable which uses yield to ‘return’ the next value**

**An iterator is any construct which implements the iterator protocol - i.e. an object which implements the `\_\_iter\_\_` and `next` method;**

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

**If a function contains at least one yield statement (it may contain other yield or return statements), it becomes a generator function**

4) What is the purpose of a yield statement?

**The yield keyword in python works like a return with the only difference is that instead of returning a value, it gives back a generator object to the caller.**

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

**The map function takes an Expression and an Iterable. The output will be an Iterable object where the expression will work on each element of the given Iterable. The output of each expression will be an element of the resultant Iterable. On the other hand, Python List Comprehension is used for creating a list where each element is generated by applying a simple formula on the given list.**