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

The difference is that a generator is defined using parentheses, while list comprehensions are defined using square brackets.

2) What is the relationship between generators and iterators?

iterators are objects that can be iterated on, and generators are special functions that leverage lazy evaluation. Implementing your own iterator means you must create an \_\_iter\_\_() and \_\_next\_\_() method, whereas a generator can be implemented using the yield keyword in a Python function.

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

We can only tell it's a generator function by the presence of a yield statement. A yield statement turns a regular function into a generator function.

4) What is the purpose of a yield statement?

The yield statement returns a generator object to the one who calls the function which contains yield, instead of simply returning a value.

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

List comprehension returns a list, whereas the map function returns an object of Iterable. List comprehension execution is faster than that of map function when the formula expression is huge and complex. Map function is faster than list comprehension when the formula is already defined as a function earlier.