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

Enclosing a list comprehension in square brackets creates a list, while enclosing it in parentheses creates a generator expression. The list comprehension evaluates immediately and creates a list in memory, while the generator expression produces an iterator that lazily computes values on demand.

2) What is the relationship between generators and iterators? Enclosing a list comprehension in square brackets creates a list, while enclosing it in parentheses creates a generator expression. The list comprehension evaluates immediately and creates a list in memory, while the generator expression produces an iterator that lazily computes values on demand.

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

A generator is a type of iterator that produces a sequence of values lazily, as they are requested, instead of generating them all at once. In other words, a generator is a type of iterator that is defined using a generator function. A generator function is a function that contains a yield statement, which is used to produce the next value in the sequence when the generator is iterated.

A function is a generator function if it contains at least one yield statement. When a generator function is called, it returns a generator object, which can be used to iterate over the sequence of values produced by the generator function.

4) What is the purpose of a yield statement?

The purpose of a yield statement is to produce a value from a generator function and suspend the function's execution until the next value is requested. When a yield statement is encountered, the function's state is saved and the yielded value is returned to the caller. When the generator is iterated again, the function's state is restored and execution continues from where it left off.

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

Map calls and list comprehensions are both used to transform one iterable into another by applying a function to each element of the iterable. The main difference between the two is that map returns an iterator that lazily computes the transformed values on demand, while a list comprehension creates a new list in memory that contains all the transformed values at once. Another difference is that map can only apply a single function to each element, while a list comprehension can apply multiple operations or conditions to each element. List comprehensions can also be more readable and easier to write in some cases, but generators can be more memory efficient and performant when dealing with large datasets.