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

ANSWER.

- Square brackets create a list object containing the results of the comprehension, while parentheses create a generator expression object.

- List comprehensions are evaluated eagerly and produce a list containing all the elements, while generator expressions are evaluated lazily and produce elements on demand.

2) What is the relationship between generators and iterators?

ANSWER.

Generators are a type of iterator that allows you to generate values lazily using generator functions or expressions. They provide a convenient and efficient way to work with iterable data streams in Python, making it easier to write readable and memory-efficient code.

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

ANSWER.

def my\_generator():

yield 1

yield 2

yield 3

gen = my\_generator()

4) What is the purpose of a yield statement?

ANSWER.

The `yield` statement is a powerful tool for creating generator functions and expressions in Python, allowing for lazy evaluation, memory-efficient processing, stateful iteration, and seamless integration with the iteration protocol. It is commonly used in situations where efficient handling of large or infinite data streams is required.

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

ANSWER.

While map calls and list comprehensions are both useful for transforming data, list comprehensions are generally preferred for their readability, versatility, and flexibility in handling complex transformations and filtering operations. However, map calls may still be useful in certain situations, particularly for applying simple transformations to large data sets.