**4. What are Python generators and why are they useful in data analysis?**

**Answer:**  
Generators return items one at a time, saving memory (lazy loading). Useful for large datasets and data streams.

A generator function in Python is a special type of function that returns an iterator (specifically, a generator object) rather than a single value. Unlike regular functions that execute a final entirely and return result, generator functions "yield" values one at a time, pausing their execution state and resuming from where they left off upon the next request for a value.

Key characteristics of generator functions:

* **Uses yield keyword:**

Instead of return, generator functions use the yield keyword to produce a value. When yield is encountered, the function's state is saved, the yielded value is returned, and the function pauses.

* **Lazy evaluation:**

Generators produce values on demand, meaning they don't compute and store all values in memory at once. This makes them highly memory-efficient, especially when dealing with large datasets or infinite sequences.

**Benefits of using generator functions**:

* **Memory efficiency:**

Ideal for processing large datasets as they don't require loading all data into memory simultaneously.

* **Lazy evaluation:**

Values are generated only when needed, which can save computation time if not all values are required.

* **Cleaner code:**

Can simplify code that would otherwise require managing iteration states manually.

**Example:**

def read\_data():

for i in range(10000):

yield i