

### 1. What is the difference between a list and a tuple in Python?

List: - List is a Data structure in python which contains or store multiple item in a single variable.

List are ordered, changeable and allow duplicates.

List are enclosed with square bracket [].

List consumes more memory.

Tuple: - Tuple is a collection of objects which is ordered and immutable.

Tuple allows duplicate values

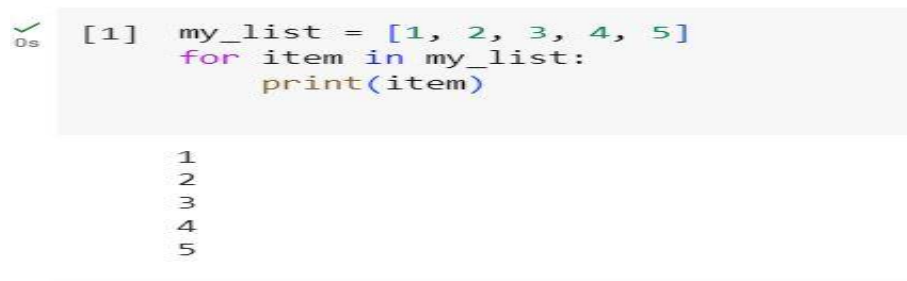
Tuples are enclosed with parentheses. ()

Tuple consumes less memory.

### 2. How can you iterate through a list in Python?.

There are several methods are used to iterate through a list such as

Using for loop:

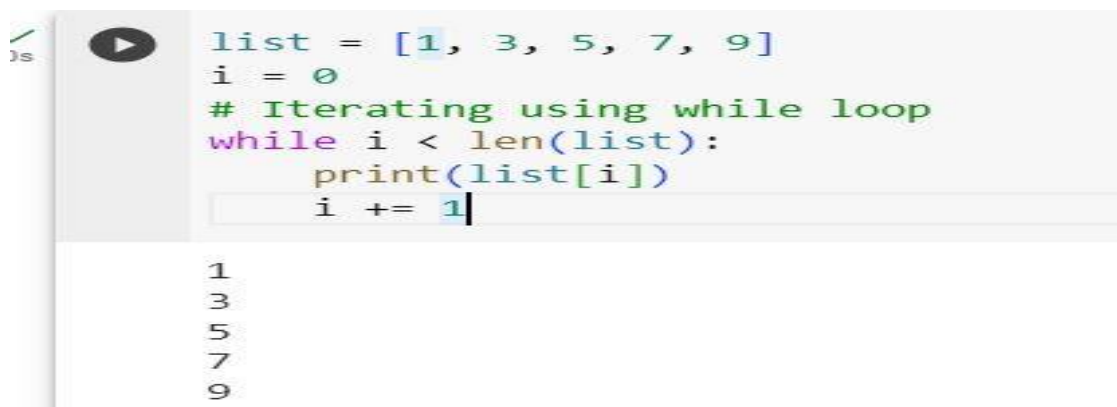


```
✓ 0s [1] my_list = [1, 2, 3, 4, 5]
      for item in my_list:
          print(item)
```

1  
2  
3  
4  
5

---

Using while loop:-



```
✓ 0s [1] list = [1, 3, 5, 7, 9]
      i = 0
      # Iterating using while loop
      while i < len(list):
          print(list[i])
          i += 1
```

1  
3  
5  
7  
9

Like this we can iterate through numpy, list comprehension and map function etc.

### 3. How do you handle exceptions in Python?

In Python, exceptions are used to handle errors and unexpected situations that may occur during the execution of the code. To handle exceptions, you can use try, except, and optionally else and finally blocks.

example:-

```
✓ [8] try:
    result = 10 / 0
except Exception as e:
    print(e,type(e))
else:
    print("No exception occurred.")
finally:
    print("This block always runs.")

division by zero <class 'ZeroDivisionError'>
This block always runs.
```

4.What are list comprehensions in Python?

List comprehensions are a concise and readable way to create lists in Python. They provide a compact syntax for creating new lists by applying an expression to each item in an existing iterable (e.g., a list, tuple, or range) and optionally filtering the items based on a condition. List comprehensions are a powerful and efficient way to work with lists in Python.

Syntax: newList = [ expression(element) for element in oldList if condition ]

Example:-

```
✓ [9] ## List comprehension
num=[1,2,3,4,5,6]
square_num=[i**2 for i in num]
print(square_num)

[1, 4, 9, 16, 25, 36]
```

5. What is the purpose of the if `__name__ == "__main__"` statement?

The if `__name__ == "__main__"` statement is commonly used in Python scripts to determine whether the script is being run as the main program or if it is being imported as a module into another script. This construct allows you to write code that can be used both as a standalone script and as a module that can be imported into other scripts.

6. What is the purpose of the `with` statement in Python?

- The with statement in Python is used to simplify the management of resources, such as files, network connections, and database connections, by ensuring that they are properly acquired and released. It is often used in the context of context managers, which define the behaviour of entering and exiting a "context."
- The primary purpose of the with statement is to ensure that certain operations are performed before and after a block of code, typically related to resource management, like opening and closing a file.

## 7. What are the key features of Spark?

Apache Spark is an open-source, distributed computing framework designed for big data processing and analytics. The key features are speed, ease of use, and versatility and it has gained popularity due to this.

## 8. What are Resilient Distributed Datasets (RDDs) in Spark?

Resilient Distributed Datasets (RDDs) are a fundamental data structure in Apache Spark. RDDs are the building blocks of Spark's distributed data processing framework, and they represent a distributed collection of data that can be processed in parallel across a cluster of machines.

## 9. What is the difference between a DataFrame and an RDD in Spark?

DataFrame	RDD
<ul style="list-style-type: none"><li>• DataFrame is a higher-level abstraction built on top of RDDs. It represents structured data with a schema, similar to a table in a relational database.</li><li>• DataFrames provide a more structured way to work with data, making them suitable for structured data like CSV, JSON, or Parquet files.</li><li>• DataFrames provide a more user-friendly API that is similar to SQL. This makes them easier to work with and requires less code for typical data operations.</li></ul>	<ul style="list-style-type: none"><li>• RDD is a lower-level, fundamental data structure in Spark. It is a distributed collection of data with no specific schema.</li><li>• RDDs are more flexible and suitable for unstructured or semi-structured data.</li><li>• RDDs require more low-level coding, which can be less user-friendly. You have to write more code for common operations.</li></ul>

## 10. What is Spark's ecosystem?

Apache Spark has a rich and extensive ecosystem of libraries, tools, and extensions that complement and extend its capabilities. The Spark ecosystem includes a wide range of projects and components, each serving specific needs in big data processing, analytics, and machine learning.

The Spark ecosystem is continuously evolving, with new projects and extensions being developed to address specific use cases and to enhance Spark's capabilities.