1. **What are the types of Applications?**

**Practical Applications of Python**

* Operating Systems.
* Web and Internet Development.
* Game Development.
* Scientific and Numeric Computing.
* Artificial Intelligence and Machine Learning.
* Desktop GUI.
* Business Applications.
* Console-based Application.

1. **What is programing?**

Programming is the process of creating instructions that a computer can follow to perform specific tasks. These instructions are written in a programming language, which is a set of rules and symbols used to communicate with the computer.

1. **What is Python?**

Python is a programming language that is used for many purposes, including:

* Web development: Python is used to build websites and web applications
* Software development: Python is used to create software
* Data science: Python is used to analyze data
* Machine learning: Python is used for machine learning
* Scientific computing: Python is used for scientific computing
* Automation: Python is used to automate tasks

1. **How memory is managed in Python?**

Python's memory management is handled automatically by its built-in garbage collector. This means that programmers don't need to explicitly allocate or deallocate memory for variables. The garbage collector periodically identifies objects that are no longer in use and reclaims their memory.

Here's a breakdown of how Python's memory management works:

**Reference Counting:** Python uses a reference counting system to track the number of references to an object. When an object's reference count reaches zero, it's considered garbage and is eligible for collection.

**Garbage Collection:** The garbage collector runs periodically to identify objects with a reference count of zero. It then reclaims the memory occupied by these objects.

**Cyclic Reference Detection:** While reference counting is generally effective, it can't handle cyclic references (where objects refer to each other in a circular manner). Python uses a cycle-detecting garbage collector to identify and break such cycles.

1. **What is the purpose continuing statement in python?**

In Python, a continuation statement is used to indicate that a statement is not complete and will be continued on the next line. This is typically achieved using the backslash character

Ex. long\_string = "This is a very long string that needs to be continued on the next line" \

"Because it won't fit on a single line."

1. **What are negative indexes and why are they used?**

In Python, negative indexes are used to access elements from the end of a sequence (like strings, lists, or tuples). Instead of starting from 0 at the beginning, negative indexes start from -1 at the end.

Why Use Negative Indexes?

* **Convenience:** When you know the relative position of an element from the end, using a negative index can be more concise and readable than calculating the positive index.
* **Flexibility:** Negative indexes allow you to access elements from both the beginning and end of a sequence without needing to know the exact length of the sequence.
* **Dynamic Access:** If the length of the sequence changes, negative indexes can still be used reliably to access elements based on their relative positions.

1. **What is List? How will you reverse a list?**

A **list** in Python is a collection of ordered, mutable elements. It is one of the most versatile and commonly used data structures in Python. Lists can store items of different types (integers, strings, other lists, etc.), and the elements can be accessed via indexing.

**Syntax**: A list is defined using square brackets [], with items separated by commas.

Example:

python

Copy code

my\_list = [1, 2, 3, 'apple', 4.5]

**Using reverse() Method**

The reverse() method reverses the elements of the list **in place**, meaning it modifies the original list and does not return a new list.

python

Copy code

# Example

my\_list = [1, 2, 3, 4, 5]

my\_list.reverse()

print(my\_list)

# Output: [5, 4, 3, 2, 1]

1. **How will you remove last object from a list?**

**Using the pop() Method**

The pop() method removes and returns the last item from the list. If no index is provided, it will remove the last item by default.

**Example:**

python

Copy code

# Example list

my\_list = [10, 20, 30, 40, 50]

# Remove the last item

my\_list.pop()

# Print the modified list

print(my\_list)

# Output: [10, 20, 30, 40]

# Remove the last item and capture the removed element removed\_element = my\_list.pop() print("Removed element:", removed\_element) # Output: 50 print("Updated list:", my\_list) # Output: [10, 20, 30, 40]

1. **Differentiate between append () and extend () methods?**

**append()**: Adds a single element at the end (the element could be a list or any other type, and it will be added as a single item).

**extend()**: Adds each element from an iterable to the list, effectively "unpacking" the iterable and adding each element individually.

43)What is tuple? Difference between list and tuple.

1. How Many Basic Types of Functions Are Available in Python?

**Built-in functions**: Predefined functions available in Python.

**User-defined functions**: Functions created by the user to perform specific tasks.

These two categories cover the basic types of functions in Python.