### What is Python and what are the advantages and disadvantages f using python?

### What is Python?

* High-level, interpreted programming language
* Created by Guido van Rossum, first released in 1991
* Supports multiple programming paradigms
* Comprehensive standard library and active community

### Advantages of Using Python

1. **Readability and Simplicity**
2. **Extensive Standard Library**
3. **Cross-Platform Compatibility**
4. **Wide Range of Applications**
5. **Strong Community Support**
6. **Integration Capabilities**
7. **Rapid Development**

### Disadvantages of Using Python

1. **Performance Limitations**
2. **Memory Consumption**
3. **Mobile Development**
4. **Global Interpreter Lock (GIL)**
5. **Less Suitable for Low-Level Programming**
6. **Dependency Management**
7. **Dynamic Typing**
8. What are the key features of Python?

### Key Features of Python

1. **Simple and Readable Syntax**
2. **Interpreted Language**
3. **Dynamically Typed**
4. **High-Level Language**
5. **Extensive Standard Library**
6. **Object-Oriented**
7. **Portable**
8. **Extensible and Embedded**
9. **Support for Multiple Programming Paradigms**
10. **Automatic Memory Management**
11. **Robust Frameworks and Libraries**
12. **Interactive Mode**
13. **Strong Community Support**
14. **Integrated Development Environments (IDEs)**
15. What is variables in python? how do you declare a variable ? Give example.

### What is a Variable in Python?

A variable in Python is a symbolic name that references or points to an object. The variable stores data values and can be used to hold various types of data, such as numbers, strings, lists, or even functions.

### How to Declare a Variable in Python?

Variables in Python are created by simply assigning a value to a name using the = operator. There is no need to declare the variable type explicitly, as Python is dynamically typed.

### Example

Here are some examples of declaring variables in Python:

# Integer variable

age = 25

# Float variable

height = 5.9

# String variable

name = "Alice"

# Boolean variable

is\_student = True

# List variable

scores = [85, 90, 78]

# Dictionary variable

student = {"name": "Alice", "age": 25, "is\_student": True}

4. What are the different data types available in Python? Give an example.

### Different Data Types Available in Python

**Integer (int)**

* 1. Represents whole numbers.
  2. x=2

**Float (float)**

* 1. Represents decimal numbers.
  2. k=3.56
  3. Exampl

**String (str)**

* 1. Represents a sequence of characters.
  2. name = "Alice"

**Boolean (bool)**

* 1. Represents True or False values.
  2. is\_student = True

**List (list)**

* 1. Represents an ordered collection of items.
  2. scores = [85, 90, 78]:

**Tuple (tuple)**

* 1. Represents an ordered, immutable collection of items.
  2. coordinates = (10.0, 20.0)

**Dictionary (dict)**

* 1. Represents a collection of key-value pairs.
  2. student = {"name": "Alice", "age": 25, "is\_student": True}

**Set (set)**

* 1. Represents an unordered collection of unique items.
  2. unique\_numbers = {1, 2, 3, 4, 5}

**None Type**

* 1. Represents the absence of a value or a null value.
  2. data = None

5. What is a function in Python? How do you define a function? Give an example.

### What is a Function in Python?

* A function in Python is a reusable block of code that performs a specific task.
* Functions help to modularize code, making it more readable and maintainable.

### How to Define a Function in Python?

* Use the def keyword followed by the function name and parentheses.
* Define the function body with an indented block of code.

### Example

def greet(name):

# Function to greet a person with their name

print(f"Hello, {name}!")

# Calling the function

greet("Waseem")

* def greet(name): - This line defines a function named greet that takes one parameter name.
* The function body prints a greeting message.

1. Explain the concept of loops in Python. What are the different types of loops? Give an example.

### Different Types of Loops in Python with Examples

#### for Loop

Example:

fruits = ["apple", "banana", "cherry"]

for fruit in fruits:

print(fruit)

While loop

Example

count = 0

while count < 5:

if count == 3:

break

print(count)

count += 1

1. What is a list in Python? How do you create a list? Give an example

### What is a List in Python?

* A list in Python is a collection of items that are ordered and mutable (modifiable).
* Lists can contain elements of different data types, including other lists.

### How to Create a List in Python?

* Use square brackets [] to define a list and separate elements with commas.

1. What is a dictionary in Python? How do you create a dictionary? Give an example

### What is a Dictionary in Python?

* A dictionary in Python is an unordered collection of key-value pairs.
* Each key is unique and associated with a value.
* Dictionaries are mutable (modifiable) and can store heterogeneous data.

### How to Create a Dictionary in Python?

* Use curly braces {} to define a dictionary.
* Specify key-value pairs separated by colons :. Keys are usually strings or numbers, and values can be of any data type.

1. What is a module in Python? How do you import a module? Give an example

### What is a Module in Python?

* A module in Python is a file containing Python code that defines functions, classes, and variables.
* Modules help organize code into manageable sections and allow for code reuse across different programs.

### How to Import a Module in Python?

* Use the import statement to bring a module into your current script.

10. What is exception handling in Python? How do you handle exceptions? Give an example

### What is Exception Handling in Python?

* Exception handling in Python is a mechanism to handle runtime errors, ensuring that the program can continue its execution or terminate gracefully.
* Exceptions are raised when an error occurs during the execution of a program. These can be caught and handled using try, except, else, and finally blocks.

### How to Handle Exceptions in Python?

* try **block:** Contains the code that might raise an exception.
* except **block:** Contains the code that executes if an exception occurs.
* else **block:** Contains the code that executes if no exception occurs.
* finally **block:** Contains the code that executes no matter what, typically for cleanup actions.
  1. Example: