

1. Who developed python programming language.

Python was developed by Guido van Rossum. He started working on Python in the late 1980s, and the language was first released in 1991.

2. Which type of programming does python support ?

Procedural programming

Object oriented programming

Function programming.

Imperative programming

Reflective programming.

3. Is python case sensitive with identifiers ?

Yes, This means that variable names, function names, and other identifiers must be used consistently with regard to their case.

4. What is correct extension of python file ?

The correct extension is .py

5. Is python code compiled or interpreted ?

Python code is primarily interpreted. When you run a Python script, the Python interpreter converts the source code into bytecode, which is then executed by the Python virtual machine (PVM). This process allows Python to be an interpreted language, although it also involves a compilation step to bytecode.

6. Name a few blocks of code used to define in python language .

In Python, blocks of code are defined using indentation rather than braces or keywords.

Here are a few types of code blocks commonly used in Python:

```
def my_function():  
    # Function block  
    print("Hello, World!")
```

Classes

```
class MyClass:  
    # Class block  
    def __init__(self, value):  
        self.value = value
```

Conditional Statement

```
if condition:  
    # If block  
    print("Condition is True")  
elif another_condition:  
    # Elif block  
    print("Another condition is True")  
else:  
    # Else block  
    print("Condition is False")
```

7. state a character used to give single line comment in python.

‘#’ gives a single line comment.

8. mentions functions which helps us find version of the python.

```
python --version
```

# or

```
python -V
```

9. python supports anonymous function at runtime , using a construct called----- ?

Lambda , lambda functions are often used for short, throwaway functions, typically as arguments to higher-order functions like `map()`, `filter()`, and `sorted()`.

10. what does pip stands for in python ?

pip stands for "Pip Installs Packages" or "Preferred Installer Program." It is the package installer for Python and allows you to install and manage additional libraries and dependencies that are not included in the standard Python library.

You can use pip to install packages from the Python Package Index (PyPI) and other indexes. Here are some common pip commands.

`Abs()`, `len()`, `max()`, `min()` etc.

11. Max possible length of an identifier in python ?

In Python, there is no explicit limit on the maximum length of an identifier (variable name). The length is only constrained by the limits of your memory and the underlying system. However, it is generally advisable to keep identifiers reasonably short and meaningful for readability and maintainability.

For practical purposes, excessively long identifiers can make code difficult to read and maintain.

### 13. What are the benefits of using python ?

**Ease of Learning and Use** simple and readable syntax.

**Versatility:** Python is a general-purpose language used for a wide range of applications, including web development, data analysis, machine learning, automation, scripting, and more.

**Large Standard Library:** Python comes with a comprehensive standard library that provides many modules and packages for performing common tasks, reducing the need for additional code.

**Extensive Ecosystem:** Python has a rich ecosystem of third-party libraries and frameworks that extend its capabilities, including popular libraries like NumPy, pandas, Django, Flask, and TensorFlow.

**Cross-Platform Compatibility:** Python is available on various operating systems, including Windows, macOS, and Linux, making it easy to write code that works across different platforms.

**Strong Community Support:** Python has a large and active community that contributes to a wealth of resources, including documentation, tutorials, forums, and open-source projects.

**Integration Capabilities:** Python can easily integrate with other languages and technologies. It can interface with C/C++, Java, and other languages, and work with databases and web services.

**Readability and Maintainability:** Python's emphasis on code readability and a clean syntax helps maintain and understand code more easily, which is crucial for long-term projects.

**Productivity and Speed:** Python's high-level nature and extensive libraries can significantly boost development speed and productivity, allowing developers to focus more on solving problems than on coding details.

**Support for Modern Development Practices:** Python supports modern development practices such as object-oriented, functional, and procedural programming. It also supports test-driven development and continuous integration.

**Strong Data Science and Machine Learning Support:** Python has become a dominant language in data science and machine learning, thanks to libraries like pandas, NumPy, SciPy, scikit-learn, and TensorFlow.

**Scalability:** While Python is often associated with scripting and rapid development, it can also be used for large-scale applications, especially with frameworks like Django for web development or tools like Celery for task queues.

14. How is memory managed in python ?

Python manages memory using:

**Reference Counting:** Each object keeps track of the number of references to it. When the count drops to zero, the object's memory is freed.

**Garbage Collection:** Python's garbage collector handles cyclic references that reference counting cannot, using a generational approach.

**Memory Pools:** Python uses memory pooling and free lists to efficiently allocate and deallocate memory for small objects.

**Dynamic Resizing:** Built-in data structures like lists and dictionaries dynamically adjust their size as needed.

**Explicit Memory Management:** Developers can manually manage memory using `del`, the `gc` module, and profiling tools.

These mechanisms work together to efficiently manage and optimize memory usage in Python.

15. How to install python on windows and set path variables ?

Manually Set PATH Variable (If Needed)

If you didn't check the box to add Python to PATH during installation, or if you need to manually set the PATH, follow these steps:

Find the Python Installation Directory:

By default, Python is installed in  
C:\Users\<YourUsername>\AppData\Local\Programs\Python\Python<version>.

Add Python to PATH:

16. Is indentation required in python ?

Yes, indentation is required in Python. Unlike many other programming languages that use braces or other delimiters to define blocks of code, Python uses indentation to indicate the structure and scope of code blocks. This includes defining the body of functions, loops, conditionals, and classes.

## Examples

```
def greet(name):  
    print(f"Hello, {name}!") # Indented block is the function body
```

## Conditional Statement

```
if age >= 18:  
    print("You are an adult.") # Indented block is executed if the condition is true  
else:  
    print("You are a minor.")
```

Right-click on “This PC” or “Computer” on your desktop or in File Explorer, and select “Properties”.

Click “Advanced system settings” on the left.

Click the “Environment Variables” button.

In the “System variables” section, find and select the “Path” variable, then click “Edit”.

Click “New” and add the paths to the Python installation directory and the Scripts directory (e.g., C:\Users\<YourUsername>\AppData\Local\Programs\Python\Python<version> and C:\Users\<YourUsername>\AppData\Local\Programs\Python\Python<version>\Scripts).

Click “OK” to close all dialog boxes.