

# Python vs C++

Python is a general-purpose, high-level programming language. Python is used for web development, Machine Learning, and other cutting-edge software development. Python is suitable for both new and seasoned C++ and Java programmers. **Guido Van Rossum** has created Python in 1989 at Netherlands' National Research Institute. Python was released in 1991.

**C++** is a middle-level, case-sensitive, object-oriented programming language. Bjarne Stroustrup created C++ at Bell Labs. C++ is a platform-independent programming language that works on Windows, Mac OS, and Linux. C++ is near to hardware, allowing low-level programming. This provides a developer control over memory, improved performance, and dependable software.

Read through this article to get an overview of C++ and Python and how these two programming languages are different from each other.

## What is Python?

Python is currently one of the most widely used programming languages. It is an interpreted programming language that operates at a high level. When compared to other languages, the learning curve for Python is much lower, and it is also quite straightforward to use.

Python is the programming language of choice for professionals working in fields such as **Artificial Intelligence**, **Machine Learning (ML)**, **Data Science**, the **Internet of Things (IoT)**, etc., because it excels at both scripting applications and as standalone programmes.

In addition to this, Python is the language of choice because it is easy to learn. Because of its excellent syntax and readability, the amount of money spent on maintenance is decreased. The modularity of the programme and the reusability of the code both contribute to its support for a variety of packages and modules.

Using Python, we can perform –

- Web development
- Data analysis and machine learning
- Automation and scripting
- Software testing and many more

## Features

Here is a list of some of the important features of Python –

- **Easy to learn** – Python has a simple structure, few keywords, and a clear syntax. This makes it easy for the student to learn quickly. Code written in Python is easier to read and understand.
- **Easy to maintain** – The source code for Python is pretty easy to keep up with.
- **A large standard library** – Most of Python's library is easy to move around and works on UNIX, Windows, Mac.
- **Portable** – Python can run on a wide range of hardware platforms, and all of them have the same interface.

## Python Example

Take a look at the following simple Python program –

```
a = int(input("Enter value for a"))
b = int(input("Enter value for b"))

print("The number you have entered for a is ", a)
print("The number you have entered for b is ", b)
```

In our example, we have taken two variables "a" and "b" and assigning some value to those variables. Note that in Python, we don't need to declare datatype for variables explicitly, as the PVM will assign datatype as per the user's input.

- The **input()** function is used to **take input from the user** through keyboard.
- In Python, the return type of **input()** is string only, so we have to convert it explicitly to the type of data which we require. In our example, we have converted to **int** type explicitly through **int( )** function.
- **print()** is used to display the output.

## Output

On execution, this Python code will produce the following output –

```
Enter value for a 10
Enter value for b 20

The number you have entered for a is 10
The number you have entered for b is 20
```

## What is C++?

C++ is a statically typed, compiled, multi-paradigm, general-purpose programming language with a steep learning curve. Video games, desktop apps, and embedded systems use it extensively. C++ is so compatible with C that it can build practically all C source code without any changes. Object-oriented programming makes C++ a better-structured and safer language than C.

## Features

Let's see some features of C++ and the reason of its popularity.

- **Middle-level language** – It's a middle-level language since it can be used for both systems development and large-scale consumer applications like Media Players, Photoshop, Game Engines, etc.
- **Execution Speed** – C++ code runs quickly. Because it's compiled and uses procedures extensively. Garbage collection, dynamic typing, and other modern features impede program execution.
- **Object-oriented language** – Object-oriented programming is flexible and manageable. Large apps are possible. Growing code makes procedural code harder to handle. C++'s key advantage over C.
- **Extensive Library Support** – C++ has a vast library. Third-party libraries are supported for fast development.

## C++ Example

Let's understand the syntax of C++ through an example written below.

```
#include
using namespace std;

int main() {
    int a, b;
    cout << "Enter The value for variable a \n";
    cin >> a;
    cout << "Enter The value for variable b";
    cin >> b;
    cout << "The value of a is "<< a << "and" << b;
    return 0;
}
```

In our example, we are taking input for two variables "a" and "b" from the user through the keyboard and displaying the data on the console.

## Output

On execution, it will produce the following **output** –

```
Enter The value for variable a
10
Enter The value for variable b
20
The value of a is 10 and 20
```

Learn **Python** in-depth with real-world projects through our **Python certification course**. Enroll and become a certified expert to boost your career.

## Comparison Between Python and C++ across Various Aspects

Both Python and C++ are among the most popular programming languages. Both of them have their advantages and disadvantages. In this tutorial, we shall take a closure look at their characteristic features which differentiate one from another.

### Compiled vs Interpreted

Like C, C++ is also a compiler-based language. A compiler translates the entire code in a machine language code specific to the operating system in use and processor architecture.

Python is interpreter-based language. The interpreter executes the source code line by line.

### Cross platform

When a C++ source code such as hello.cpp is compiled on Linux, it can be only run on any other computer with Linux operating system. If required to run on other OS, it needs to be compiled.

Python interpreter doesn't produce compiled code. Source code is converted to byte code every time it is run on any operating system without any changes or additional steps.

### Portability

Python code is easily portable from one OS to other. C++ code is not portable as it must be recompiled if the OS changes.

## Speed of Development

C++ program is compiled to the machine code. Hence, its execution is faster than interpreter based language.

Python interpreter doesn't generate the machine code. Conversion of intermediate byte code to machine language is done on each execution of program.

If a program is to be used frequently, C++ is more efficient than Python.

## Easy to Learn

Compared to C++, Python has a simpler syntax. Its code is more readable. Writing C++ code seems daunting in the beginning because of complicated syntax rule such as use of curly braces and semicolon for sentence termination.

Python doesn't use curly brackets for marking a block of statements. Instead, it uses indents. Statements of similar indent level mark a block. This makes a Python program more readable.

## Static vs Dynamic Typing

C++ is a statically typed language. The type of variables for storing data need to be declared in the beginning. Undeclared variables can't be used. Once a variable is declared to be of a certain type, value of only that type can be stored in it.

Python is a dynamically typed language. It doesn't require a variable to be declared before assigning it a value. Since, a variable may store any type of data, it is called dynamically typed.

## OOP Concepts

Both C++ and Python implement object oriented programming concepts. C++ is closer to the theory of OOP than Python. C++ supports the concept of data encapsulation as the visibility of the variables can be defined as public, private and protected.

Python doesn't have the provision of defining the visibility. Unlike C++, Python doesn't support method overloading. Because it is dynamically typed, all the methods are polymorphic in nature by default.

C++ is in fact an extension of C. One can say that additional keywords are added in C so that it supports OOP. Hence, we can write a C type procedure oriented program in C++.

Python is completely object oriented language. Python's data model is such that, even if you can adapt a procedure oriented approach, Python internally uses object-oriented methodology.

## Garbage Collection

C++ uses the concept of pointers. Unused memory in a C++ program is not cleared automatically. In C++, the process of garbage collection is manual. Hence, a C++ program is likely to face memory related exceptional behavior.

Python has a mechanism of automatic garbage collection. Hence, Python program is more robust and less prone to memory related issues.

## Application Areas

Because C++ program compiles directly to machine code, it is more suitable for systems programming, writing device drivers, embedded systems and operating system utilities.

Python program is suitable for application programming. Its main area of application today is data science, machine learning, API development etc.

## Difference Between Python and C++

The following table summarizes the differences between Python and C++ –

Criteria	Python	C++
Execution	Python is an interpreted-based programming language. Python programs are interpreted by an interpreter.	C++ is a compiler-based programming language. C++ programs are compiled by a compiler.
Typing	Python is a dynamic-typed language.	C++ is a static-typed language.
Portability	Python is a highly portable language, code written and executed on a system can be easily run on another system.	C++ is not a portable language, code written and executed on a system cannot be run on another system without making changes.
Garbage collection	Python provides a garbage collection feature. You do not need to worry about the memory management. It is automatic in Python.	C++ does not provide garbage collection. You have to take care of freeing memories. It is manual in C++.
Syntax	Python's syntaxes are very easy to read, write, and understand.	C++'s syntaxes are tedious.

Performance	Python's execution performance is slower than C++'s.	C++ codes are faster than Python codes.
Application areas	Python's application areas are machine learning, web applications, and more.	C++'s application areas are embedded systems, device drivers, and more.