Python vs. C++: Two strong contenders in the world of programming

Whether you've only just begun to learn programming or have been using different languages for quite some time, sooner or later you'll come across Python and C++. The two are among the best-known and most widely used programming languages and have been inspiring their respective communities for decades. While fans would certainly find numerous arguments for the language of their choice, neutral observers wonder which option is better.

What are the pros and cons of Python?

Python is an interpreted high-level programming language that was created by Dutch developer Guido van Rossum in 1991. Today, the open-source language is maintained by the Python Software Foundation and is available for free. Python runs on all major operating systems and is platform independent. The focus is on creating a syntax that's clear and as short as possible. Python is used in websites, web applications and software development.

Advantages of Python

Learning curve: The focus on simple and easy-to-understand code means that Python is easier to learn than most other programming languages. Even newcomers experience quick learning success and are able to write their first lines of code fairly quickly.

User-friendliness: Python is renowned for its user-friendliness, offering an easy-to-learn codebase that facilitates rapid comprehension of external projects. Its logically structured syntax requires minimal dialects, enabling collaborative work without requiring extensive training beforehand.

Open Source: Python's enduring success is rooted in its commitment to the open-source philosophy. As a free and open-source programming language, Python fosters widespread adoption, fueling continuous development and adaptations by a diverse community of contributors.

Flexibility: Python stands out as an exceptionally flexible language on multiple fronts. Firstly, its versatility extends to its wide range of applications, making it a practical and sensible solution for diverse projects. Secondly, Python's platform independence ensures maximum portability, enabling seamless execution across various operating systems. Lastly, its compatibility with numerous other languages facilitates effortless integration, simplifying the development process and fostering interoperability.

Community: Python's vast and dedicated community offers tremendous advantages for newcomers. Access to extensive libraries and comprehensive documentation enables targeted approaches to new projects or problem-solving. The community further provides valuable support, assisting with inquiries and swiftly identifying and resolving any bugs that users encounter.

Users: Python has gained significant traction among numerous prominent tech companies, many of which rely on the language either partially or entirely for their operations. For example, notable industry leaders like Google and Mozilla

use Python.

Disadvantages of Python

Speed: Python can have performance disadvantages compared to some other programming languages, especially for large and complex applications. Although new packages have been developed to address this issue, some other languages still tend to exhibit better performance in terms of speed.

Mobile Devices: While Python excels in the domains of desktop and server applications, its performance in the mobile space has room for improvement. Mobile apps are rarely written with the programming language.

Memory consumption: Python's relatively high memory consumption is one of its major drawbacks, particularly for memory-intensive projects. In such scenarios, alternative programming languages often offer better choices and more efficient memory utilization.

Runtime errors: As a dynamically-typed language, Python poses a higher risk of errors, which may only become apparent during runtime. In order to detect and address issues, thorough testing is needed.

Learning other languages: Python focuses on the essentials, which can make the switch to other languages that are more complex somewhat difficult. Starting with a more intricate language first will make it easier to learn other languages.

What are the difference between Python and C++?

Let's delve into a direct comparison of Python and C++. Below we'll highlight the main differences between the two programming languages.

Differences in syntax

When it comes to syntax, there are significant differences between Python and C++. Python uses indentations, while C++ relies on semicolons and curly braces for delimitation. Python offers more options for Boolean expressions, while C++ only interprets values that correspond to 0 as false.

Learning curve: Python is easier to learn than C++ and has more flexible and concise code. In contrast, C++ demands closer attention, as mistakes are swiftly penalized.

Performance: C++ outperforms Python in terms of performance, making it advantageous for large projects. Python's dynamic typing introduces overhead, resulting in slower execution compared to C++.

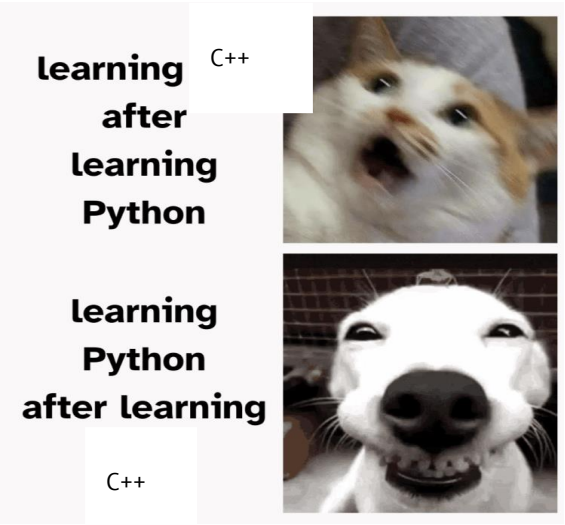
Use: C++ is commonly employed in large-scale projects and low-level hardware programming. It is also well-suited for building extensive games. On the other hand, Python finds its strength in software development and web applications. It is widely used in machine learning and artificial intelligence.

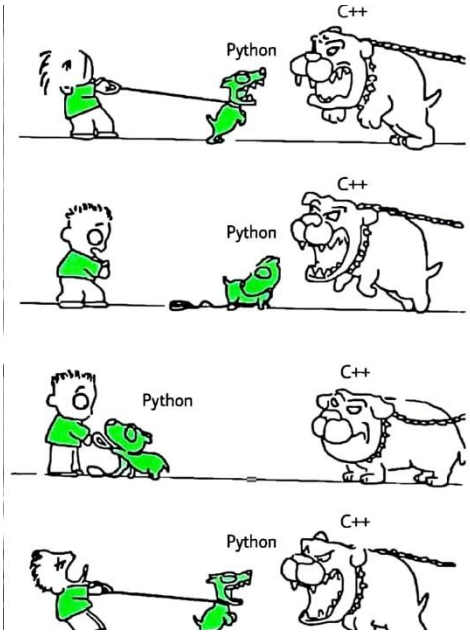
Python vs. C++: Which language is better?

The Python vs. C++ duel lacks a clear winner, as the better choice depends on individual preferences and project requirements. Python excels in quick learning and the rapid development of small programs. In contrast, C++ is suitable for large projects and exploring multiple languages, although it requires more time to master. With influential companies

like Google favoring Python and Netflix relying on C++, both languages are poised to have significant roles in the future.

1. Explain how these memes are related to the text:





2. Complete these sentences with information from the text. Highlight the lines that justify your answer:
   a. Python is ideal for programming beginners, because _____.
   b. Python is a good option when you want your code to work in different devices, due to the fact that _____
   c. Being an open source language means that resources to learn Python _____ and its users are encouraged to _____.
   d. _____, which may cause problems when something has a high memory requirement.
   e. The reason why it's necessary to be extremely careful when testing code written in Python is _____.
   f. C++ and Python have different uses, while C++ _____, Python_____.
3. Match the expressions with the ones that replace them (highlighted in yellow):

a. Projects that demand a lot of memory
b. Python and C++
c. C++
d. Tech companies
e. Having performance problems when dealing with large applications
f. Python
g. People who are new to programming

4. Find the head noun and translate the noun phrase
    a. the best-known and most widely used programming languages
    b. an interpreted high-level programming language that was created by Dutch developer Guido van Rossum in 1991.
    c. a syntax that's clear and as short as possible
    d. seamless execution across various operating systems
    e. a practical and sensible solution for diverse projects
    f. low-level hardware programming
5. Write the Spanish equivalent of the expressions that are underlined
6. Choose four expressions form 5) and write a sentence using each one, related to programming languages.
7. Have a look at the words in **bold**. What are they comparing?
    a. Python is **easier** to learn than C++ and has **more flexible** and concise code
    b. Python's relatively high memory consumption is one of its major drawbacks, particularly for memory-intensive projects. In such scenarios, alternative programming languages often offer **better** choices and **more efficient** memory utilization.
    c. The two are among **the best-known** and **most widely used** programming languages.
8. What extra layer of meaning do the highlighted words add to the main verb?
    a. Python poses a higher risk of errors, which may only become apparent during runtime.
    b. Python focuses on the essentials, which can make the switch to other languages that are more complex somewhat difficult.