Modern Object-Oriented Languages

This lesson is about the most popular Object-Oriented Programming languages and how convenient Python is.

WE'LL COVER THE FOLLOWING

- Languages Round-Up
- Comparing Code in Multiple Languages

Languages Round-Up

Nowadays, there are a lot of object-oriented programming languages being used around the globe and Python is one of them.

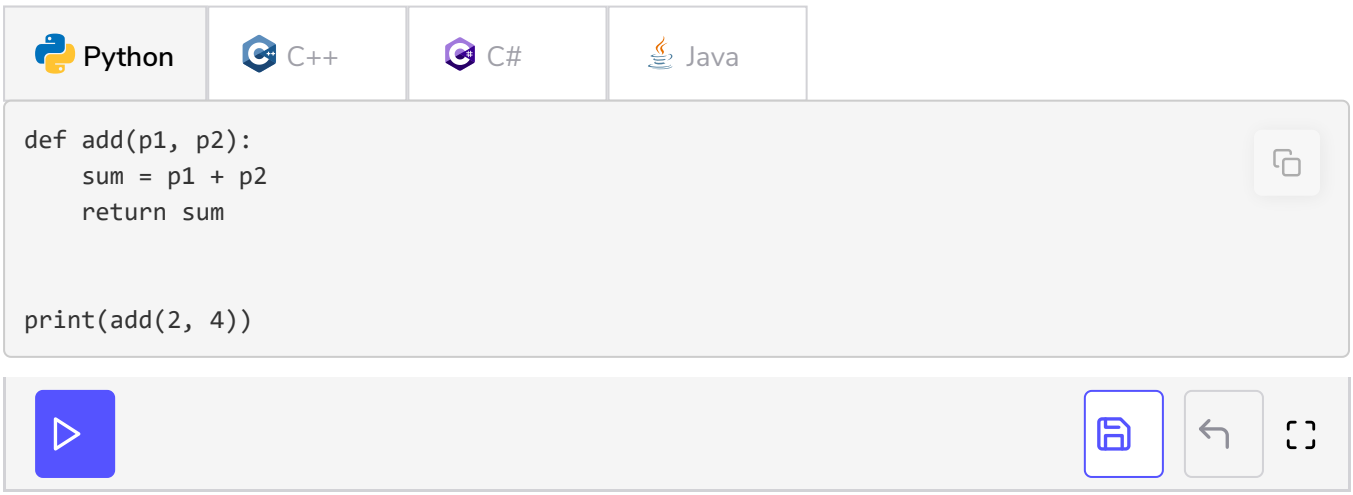
Each language has its own pros and cons. A language is chosen depending on the nature of the task to be performed. Below, we can find some of the most popular programming languages with their core strengths and common applications:

- **Python:** Data science, machine learning, and artificial intelligence.

- **Java:** Mostly used for enterprise-level and Android software development.
- **JavaScript:** Rapid and productive web development.
- **Ruby:** Web application development.
- **C++:** System software development.
- **C#:** Game development, web forms, and web applications development.

Comparing Code in Multiple Languages

Suppose we want to write a simple function that adds two numbers and returns their sum. We'll write this function in four commonly used languages:



As you can see in the codes above, for **Python** we simply wrote the `add` function without making any class or importing any libraries, and called it in our main function just like in a procedural language.

However, when you look at the code for **C++**, **Java**, and **C#**, you'll see that all three are a bit more complex and extensive. If you tried removing some keywords like `Class`, `static`, and `int` in these codes, it would generate an error and the codes won't compile. So, if you intend to write even a simple add or sum function in **C++**, **C#**, or **Java**, you'll always face a base level of complexity.

Hence the choice of language is programmer and problem dependent.

Java and **C#** have emerged as the dominant object-oriented languages, but **Python** is now asked for at a huge number of jobs, in addition to being a primary building block in academics.

Having said this, there are some people who are skeptical about the use of

object-oriented language. They believe it makes the overall program size more complex. As we've shown in our example above, even writing a simple sum function can become a complicated task. They also argue that the details of object-oriented programming, its syntax, and peculiarities, are difficult for the programmer to learn, resulting in a steep learning curve.

Nonetheless, the object-oriented paradigm is a well-established programming practice and this course is designed to make you familiar with it in one of the most user-friendly languages, Python.