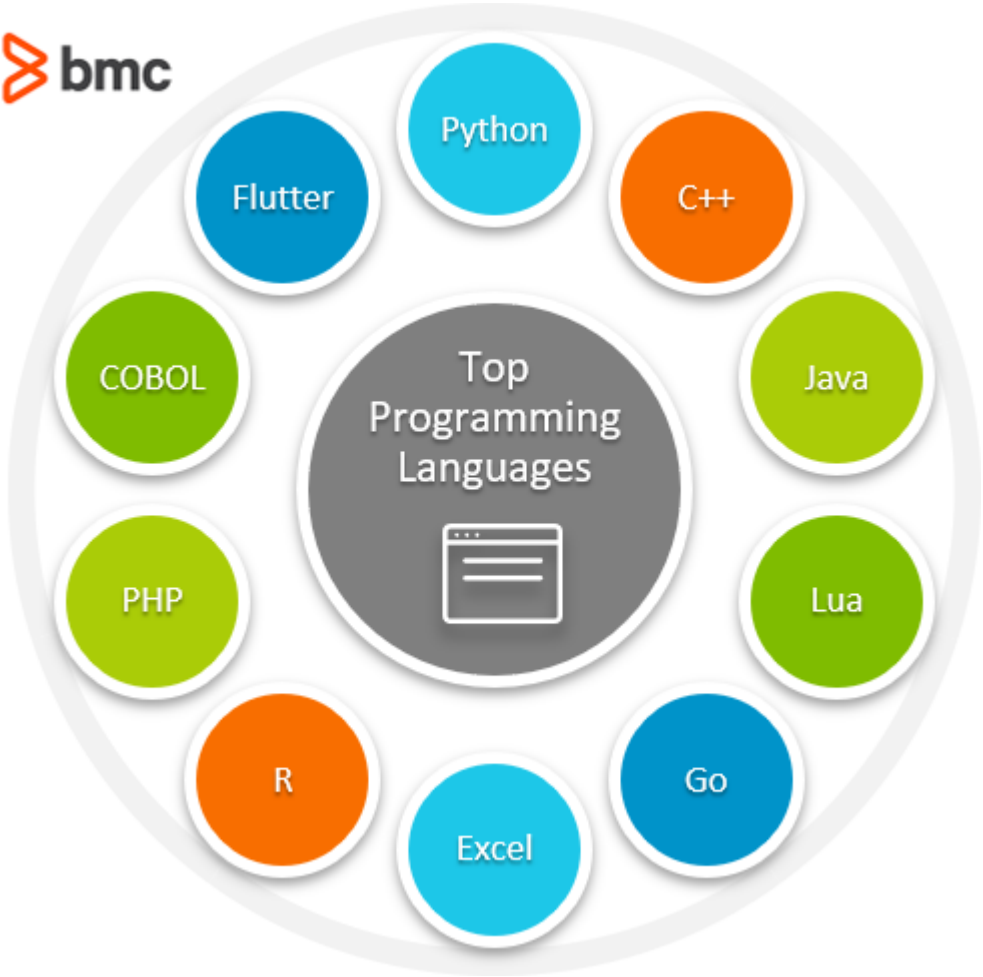
TOP PROGRAMMING LANGUAGES TO LEARN TODAY

If you want a field that is constantly changing, programming is up there. Making your way in the field can be difficult, especially if you're starting out and unsure which languages to learn.

That's why I've put together this list of the best and most common programming languages to learn, including their strengths, weaknesses, and typical use cases.

Python, C++, and other big players are included in this roundup as well as some other languages that might be new to you.



What is a programming language?

Programming languages are the code that is used to create websites and applications, manage and analyze data, and support a whole range of other uses.

The good news with learning programming languages is that there are plenty of resources for learning and getting support. It's not at all like learning a natural language, where you need to speak it well enough for others to understand you.

In fact, a developer can generally pick a new language up quickly once they really understand at least one language—you'll better understand the structures. Still, each language has its own quirks.

For most heavy software, especially enterprise apps, a number of languages can be used. Some languages are better for certain use cases. And, of course, trends come and go for which languages people use.

(Learn how to become a developer or data scientist.)

Top programming languages for 2021

Now, let's look at the top programming languages to learn today.

Python

A fantastic option for newcomers to the world of programming, Python is easy to read, needs fewer lines of code to function, and has huge library support. A number of releases have made Python more powerful and more flexible. Today, you can use Python to create a wide range of software.

With many uses in data science, Python is becoming more and more popular every year. Developed specifically to remove many difficulties associated with programming languages, Python is both perfect for coding beginners and a useful tool for experienced developers who need an interpreted, dynamic language.

When do I use Python?

- Data science
- Artificial intelligence (AI) and machine learning (ML)
- Anything else that you need a flexible language with

When do I not use Python?

- Mobile and browser computing
- Some design restrictions

Read more about Python:

- Python vs Java: What's The Difference?
- Using Python for Big Data & Analytics
- Python vs Go: What's The Difference?
- Python Development Tools: Your Python Starter Kit
- Python & Spark ML Guide

C++

Portable, object-oriented, and fully scalable, C++ is a powerful general-purpose programming language that is capable of developing operating systems, games, and web browsers.

C++ is used in countless applications and technologies every year, delivering excellent performance and business usefulness.

With its expansive user base, C++ is always evolving. That's makes it challenging to learn, but it's still extremely useful—and likely worth the effort. If you can master pointers and the lack of garbage collector, you will have the skills to work on a range of C++ projects.

When do I use C++?

- Building video games
- Writing web scripting
- Building operating systems

When do I not use C++?

- Building websites

Lua

Game developers should be familiar with Lua.

Dynamic and designed to catch faults, Lua is user-friendly. Despite being built on C/C++, Lua is a lot easier to learn. A number of features in Lua make it a useful language for any developer, even beginners.

The main strength of Lua is how it allows end-user customization. Allowing your users—gamers, app users—to create mods and customize the product does not expose the backend and risk causing damage to your work.

Trends in programming have meant that Lua is less popular than it used to be, but it is still an effective tool.

When do I use Lua?

- Game development
- Situations where you don't want someone to edit your C++ code

When do I not use Lua?

- Web development
- Web scripting
- Operating system writing

Go

Also known as Golang, Go is a programming language created by Google that is perfect for cloud-centric projects. Behind projects such as Docker and Kubernetes, Go has plenty to offer developers today. Google has created a tool to build cloud technology that could start to change the way we look at other classic languages.

Best used in distributed network services and cloud technology, Go offers memory safety, platform-agnostic deployment, and clean code. That's why many developers have started rewriting old, buggy code in Go.

When do I use Go?

- Building cloud-centric projects
- Distributed network infrastructures
- Utilities and independent tools

When do I not use Go?

- Memory-sensitive projects
- Enterprise applications (due to a lack of effective garbage collection)

Read more about Go:

- Go vs Python: What's The Difference?
- Go vs Java: Comparing Languages

Java

Many developers create applications to run on a small-scale or enterprise-level through Java. It's perfect for mobile, desktop, and IoT applications thanks to its easy-to-learn and platform-independent nature.

Because Java is not designed to develop attractive GUIs, it is not perfect for everyone. But creating flexible apps is easy with this useful tool.

When do I use Java?

- Game development
- Application development for all platforms

When do I not use Java?

- When you need an attractive GUI
- Verbose or complex systems

Read more about Java:

- Python vs Java: What's The Difference?
- Java on the Mainframe: z/OS vs Linux
- The State of Java Today
- 10 Must-Read Books for Java Developers
- Java Developer Roles & Responsibilities
- Top Java Interview Questions—Answered

Microsoft Excel

You might be surprised to see Excel on this list—but it is indeed a programming language!

Used by programmers and non-programmers alike, Excel is easy to use but difficult to master. Automation, efficiency, and greater security are all key uses for the Excel Macro.

Data scientists have many uses for Excel, but its GUI offers a simple pathway for would-be programmers to learn the ropes. A project is easier to manage with a visual interface, especially compared to other denser programming languages.

When do I use Excel?

- Data entry and visualization
- Task management
- Financial analysis

When do I not use Excel?

- Most other uses that we would expect of programming languages (software development, gaming, etc.)

R

If you've worked in statistical modelling, you will have come across R. Beneficial for scientists and data analysts, R makes it easy to visualize data and create statistical models.

Platform-independent and capable of machine learning, R is a powerful tool for data analysts and those in related fields. But don't expect to build dynamic programs or 3D graphics with it.

When do I use R?

- Data analysis
- Data visualization
- Statistical modeling

When do I not use R?

- When you need a fast data analysis tool
- When you aren't uncomfortable with using packages

(Download our free hands-on guide to learning ML.)

PHP

Despite current competition from newer coding languages like Node.js, PHP has been used to build web applications since long ago. It comes with a wide range of packages, is powerful, and can perform unit testing efficiently.

With so much community support for PHP, it can take a long time to learn how to do exactly what you need to know. But that support means that it is possible to use a wide range of tools to increase PHP's utility.

When do I use PHP?

- Web applications
- Website creation
- Database management

When do I not use PHP?

- When you need faster than fast websites or apps
- When you need flexibility in your programming

COBOL

You might be surprised to see COBOL on this list, but for a particular set of engineers, COBOL continues to be a go-to language. One of the oldest programming languages—it just turned 60 years old—there are an estimated 220 billion lines of COBOL code still in production today.

COBOL remains a crucial language, particularly in the financial and banking industries. Mainframe computers—the ones that companies rely on for daily transactions—run on COBOL. It's probably not necessary to learn unless you are a mainframe sysadmin.

Read more:

- COBOL Trends
- Planning Your Migration to COBOL V6
- It's Older (COBOL) Code, But It Checks Out

Flutter

Flutter has risen quickly as an app development tool. Originally released by Google in May 2017, Flutter has been used by more than two million developers since.

Built on the Dart programming language (another Google product), Flutter is an open-source tool for building UIs, particularly on mobile. An essential concept to Flutter is its widgets, says Jonathan Johnson. From building layouts with Scaffold and Material App widgets, to BLoC patterns and Provider Widgets, Flutter is built of widgets.

Read more:

- How To Create Your First Flutter App
- Flutter for Desktop: Get Started with Cross-Platform Development

Learning programming languages

Learning a programming language isn't easy, but with dedicated time to practice and endless online tutorials and support groups, it's a valuable way to get ahead in your developing career.

Related reading

- BMC DevOps Blog
- 5 Best Practices for Software Development
- What Is Extreme Programming (XP)?
- Asynchronous Programming: A Beginner's Guide
- API/Developer Portals: How To Create Great API Portals
- Top Conferences for Programming & Software Development