



DeepLearning.AI

A Roadmap to Programming Languages for AI



A BEGINNER'S GUIDE

Which AI Programming Languages Should I Learn?

by DeepLearning.AI

Coding is a must-have skill for anyone building AI products. It enables you to bring your machine learning ideas to life. Learning to code is fun and empowering, but it also requires time and effort. The last thing you want to do is start learning a language only to realize weeks or months later that the job you want actually calls for a different language.

Simply put: Deciding which language to start with can be intimidating.

Not to worry! This article will explain the basics behind the most popular programming languages used in AI and help you decide which to learn first. For each language, we will describe its basic features, what it does well, where it falls short, and which sorts of jobs use it most.



A Basic Roadmap to Programming Languages for AI

The five most important programming languages in AI are Python, C++, R, MATLAB, and Java. Before we dive deep into each of them, let's explore which to learn first.

For most people, **the first programming language to learn is Python**. It's easy to learn, extremely adaptable, and has numerous libraries specifically for machine learning. For those reasons and more, it is the de facto coding language in AI. What to learn next depends on your career goals.



A Basic Roadmap to Programming Languages for AI

After Python you should learn:

- **C++** — If you want to work in robotics, self-driving cars, or hardware.
- **R** — If you want to work in academia or the financial industry.
- **MATLAB** — If you wind up working at a company that still uses MATLAB. (You should convince that company to switch to Python.)
- **Java** — If you want to build scalable AI infrastructure.

One last bit of advice: Don't try to learn two languages at once. Focus on getting good at Python first. After you have reached its limits, branch out depending on your career goals.





Python

The best all-around programming language for AI.

What is it? [Python](#) is a popular, general purpose programming language that is relatively easy to learn. Its simplicity lends itself to AI development, and the AI community has adopted Python as its de facto language.

What does it do well? Python is popular for several important reasons.

- **It's easy:** Compared to other coding languages, Python has a simple syntax (the words, symbols, and expressions you'll type to create programs). This means you'll have more time to devote to the stuff that matters: Looking at data and tuning your models.





Python

The best all-around programming language for AI.

- **It's versatile:** Your operating system supports Python, whether you use iOS, Windows, or Linux. What's more, you don't need to modify a Python program much to get it to run across platforms.
- **It's open source:** Anybody can adapt, update, or add to the code that underlies Python. As a result, many members of the Python community have built frameworks and libraries that make it adaptable to nearly any machine learning or data science task.





Python

The best all-around programming language for AI.

What are the downsides? Python has limitations when it comes to performing complex mathematical and statistical functions. It also runs slowly compared to languages like C++ and Java (see below).

Who is it for? We'll say it one more time: Python is the most popular programming language in machine learning and data science. If your job involves building machine learning models and working with lots of data, Python is for you. However, you may want to specialize in a second language if you work in data analysis, AI infrastructure, or plan on doing more intensive programming outside of your core AI work. Keep reading to learn more.





C++

The best programming language for AI infrastructure.

What is it? C++ is one of the most popular languages for general-purpose applications. It is the backbone of operating systems like Windows, iOS, and Linux; apps like Spotify and Photoshop; sites like YouTube; along with video games, banking systems, and more. It is also an essential language for anyone working in self-driving cars or robotics.

What does it do well? C++ is a compiled language: It doesn't require an interpreter program, which adds processing overhead. In practical terms, programs written in C++ are *fast* and *efficient*.





C++

The best programming language for AI infrastructure.

What are the downsides? C++ programs may be efficient to run, but writing them is complex — with a capital C and two pluses. Writing a program in C++ takes time, debugging it often takes even more time, and rewriting it every time you adjust your hyperparameters takes more time than you think it will. C++ is also notoriously difficult to learn. If your goal is to be a data scientist, learning C++ is like learning to fly a helicopter and then using it only to shop at the grocery store.

Who is it for? C++ isn't well suited for most data scientists or machine learning engineers, but it is essential for some disciplines within AI.





C++

The best programming language for AI infrastructure.

- If your role involves building or maintaining AI infrastructure — the core software libraries that others use to deploy models or analyze data — then you should absolutely learn C++.
- Most teams working on robotics and self-driving cars use C++ for their production code due to its speed and efficiency.
- Most jobs in the chip and hardware industry call for C++.

It's important to learn C++ if you want to get involved in the open source community. Many of the most popular Python frameworks and toolkits are programmed in C++.





R

The best programming language for data analysis.

What is it? R was built specifically for statistical analysis.

What does it do well? R was developed by statisticians, for statisticians. It excels at finding patterns in data and deriving insights from model outputs. For obvious reasons, R also appeals to machine learning engineers and data scientists who use it for statistical analysis, data visualizations, and similar projects. Like Python, it is open-source, and the community has created a number of frameworks and libraries for AI tasks.

What are the downsides? In terms of complexity, beginners will find R more difficult to learn than Python. R has more built-in features for crunching numbers than Python, but it also tends to lag when processing projects that use too much data at once. R? More like Arghhh!





R

The best programming language for data analysis.

What are the downsides? In terms of complexity, beginners will find R more difficult to learn than Python. R has more built-in features for crunching numbers than Python, but it also tends to lag when processing projects that use too much data at once. R? More like Arghhh!

Who is it for? R is a great tool for data analysis, data science, and adjacent professions, but it's often used by academics. You might also be required to learn R if you get a job working in finance, and for teams that use it in their legacy software.





MATLAB

A once-popular AI programming language that has been mostly eclipsed by Python.

What is it? MATLAB is more than a programming language, it's a five-part system that consists of a language, development environment, graphics visualizer, math library, and interface for writing programs in other languages. MATLAB focuses on matrix computation. If you aren't familiar, matrices are arrays of numbers, and the ability to compute them well is central to many machine learning and data science applications.





MATLAB

A once-popular AI programming language that has been mostly eclipsed by Python.

What does it do well? MATLAB is excellent for working with numerical arrays. It is also excellent for many other mathematical operations, and it has built-in features for implementing machine learning models. These features give it a leg up over Python, which requires add-on toolkits and frameworks for both mathematical functions and model implementation. In terms of speed, it is fast and easily outpaces Python in many operations.

What are the downsides? The biggest drawback to MATLAB is the cost: You have to pay a fee to access the system and possibly for additional functionality depending on your goal.





MATLAB

A once-popular AI programming language that has been mostly eclipsed by Python.

This barrier to entry has partly contributed to MATLAB's declining popularity among AI builders; Python, after all, is free and open-source. Lastly, MATLAB's syntax is difficult to learn compared to Python's.

Who is it for? Some employers and AI teams prefer MATLAB either because they are legacy users or their goals require more mathematical oomph. Fun fact: Andrew Ng's original Machine Learning course was [taught](#) using MATLAB. When Andrew and his teams at DeepLearning.AI and Stanford University modernized the Machine Learning Specialization in 2022, one of the key upgrades was switching to [Python](#).





Java

A fast, versatile programming language that is useful for building scalable AI infrastructure.

What is it? [Java](#) is similar to Python in many ways: It is popular, open-source, and has many frameworks and toolkits specifically for machine learning and data science. Java is much older, and therefore it has legacy buy-in from many organizations. It is also more technically complex than Python. This means it is more difficult to learn than Python and R, but it can execute programs much more efficiently.

What does it do well? Everything Python can do, Java can do just as well — maybe better, in some cases. For instance, it has frameworks for data science, classification, deep learning, and more. Java has more rules than Python, which makes it more difficult to break or misuse the code.





Java

A fast, versatile programming language that is useful for building scalable AI infrastructure.

It is useful for building full stack, back-end, large scale infrastructure for deploying machine learning models.

What are the downsides? Java has a steeper learning curve than Python (though not as steep as C++). Writing programs in Java also takes more time than Python, and those programs often require more debugging due to their complexity. These are speed bumps that slow down rapid prototyping of machine learning models. Finally, Java's community isn't as active as Python's when it comes to developing AI-focused tools, and as a result Java is useful for a narrower range of machine learning and data science tasks.





Java

A fast, versatile programming language that is useful for building scalable AI infrastructure.

Who is it for? If not for its complexity, Java might be the dominant AI coding language. However, you probably only need to learn it if you are targeting a role building AI infrastructure or deploying machine learning products. For beginners, Python is a safer bet. In the meantime, you can still enjoy java in your mug each morning.



Tips from Andrew Ng on Learning to Code

One of the best ways to learn to code is by taking on a coding project. Start small, Andrew Ng [advises](#), with a project you can finish over a week or two in your spare time. The goal isn't to build a world-changing app, it's to put your knowledge into practice and learn from your mistakes.

When working on projects, it's common for coders to use Google or Stack Overflow to find pre-written lines of code that suit their needs. This is an efficient way to work, and you should feel free to do the same. However, don't just copy and paste what you find, Andrew Ng [said](#). Instead, retype it yourself. The physical act of doing so will build [muscle memory for your brain](#), helping you to internalize the concept and syntax. Keep doing this, and *you'll* be the one posting bits of code that other people copy!



Tips from Andrew Ng on Learning to Code

In a 2020 [letter](#) to readers of his newsletter, [The Batch](#), Andrew Ng wrote:

When you're trying to master a programming technique, consider these practices:

- *Read a line of code, then type it out yourself. (Bonus points for doing it without looking at the reference code while typing.)*
- *Learn about an algorithm, then try to implement it yourself.*
- *Read a research paper and try to replicate the published result.*

Learn a piece of math or a theorem and try to derive it yourself starting with a blank piece of paper.



Conclusion

Coding is an essential skill for AI builders. In fact, Andrew Ng has [compared](#) coding to literacy:

Code is the deepest form of human-to-machine communication. As machines become more central to daily life, that communication becomes ever more important.

So, which language should you learn? For most machine learning engineers and data scientists early in their careers, the best choice is Python. It is easy to learn, quick to implement, and has a ton of add-ons that are tailor-made for AI. You may be tempted to learn a bit of Python, then learn a bit of R, a bit of Java, and so on in order to be more versatile. We recommend against this. Focus on getting good at Python before you switch things up.



Ready to Start?

Are you a **Python beginner**?

Consider [AI For Everyone](#) or [Generative AI for Everyone](#)!

Neither assume a background in any particular programming language.

Already **familiar with Python**?

Try the [Machine Learning Specialization](#) to learn the fundamentals of AI.

Are you an **experienced coder** looking to get up to speed on large language models?

Check out [Generative AI with LLMs](#) to learn about the latest developments and use cases.



Ready to Start?

Not sure **where to start?**

Try one of our [short courses!](#)

- [Pair Programming with a Large Language Model](#)
- [Building Generative AI Applications with Gradio](#)
- [Building Systems with the ChatGPT API](#)
- [LangChain for LLM Application Development](#)
- [Large Language Models with Semantic Search](#)
- [ChatGPT Prompt Engineering for Developers](#)
- And more!

