# Parents and teachers: how are you introducing AI to younger children?

Across the world since the release of ChatGPT, we have seen younger and younger people make use of artificial intelligence chatbots and image generators for their [schoolwork](https://www.theguardian.com/technology/2023/oct/31/educators-teachers-ai-learning-classrooms-misuse) and their lives at home. First, [students in universities](https://www.theguardian.com/technology/2024/feb/01/more-than-half-uk-undergraduates-ai-essays-artificial-intelligence) adopted AI, then teenagers in secondary schools, often with their parents unaware of it. Now, we’d like to know how the youngest among us are learning about AI.

AI companies typically prohibit children under 13 from using their products in terms of service agreements. Some parents and teachers, however, are introducing ChatGPT and other AI to their children and their students in the hopes that an early familiarity with the technology will prepare them to succeed in the future.

With this in mind, we would like to hear from parents and teachers about introducing AI to younger children and instructing them how to use it. In what ways are you teaching them about the technology, and why?

**Share your experience**

You can tell us about how you are teaching your students or your children about AI using this form

Culled from <https://www.theguardian.com/technology/2025/mar/03/parents-and-teachers-how-are-you-introducing-ai-to-younger-children>

# **Best Programming Languages to Learn**

No single programming language is the key to a thriving career in the tech sector. Take a look at the best programming languages to learn and how they can work for you.

If you're just getting started in programming, you face a tough call — with potentially thousands of languages available, which one do you learn?

The answer depends on what you want to accomplish. Do you want to do [front-end or back-end development](https://www.computerscience.org/bootcamps/resources/frontend-vs-backend/)? Create games or build mobile apps? Do systems programming or construct desktop applications?

More importantly, now that AI has learned to code, is learning a programming language even a good idea?

This guide helps answer these questions. We explore whether programming languages are still viable (they are) and focus on top languages for beginners and experts alike to learn. We also explore how programming languages are used and look at where you can get started in the fast-changing, powerful world of [computer programming](https://www.computerscience.org/careers/computer-programmer/).

Culled from https://www.computerscience.org/resources/best-programming-languages-to-learn/

## **What Is the Difference Between Programming and Coding?**

In short, coding is a [part of programming](https://www.computerscience.org/bootcamps/guides/coding-jargon/). Coding involves using programming languages to translate human instructions into a language machines understand.

Creating code works like creating a novel: it's all about the words you use and how you combine them to form sentences that are easy to read. Programming is the art of coming in after the coder and making sure those sentences work together. To continue the metaphor: Coders create the draft. Programmers are the editors that make sure the draft flows.

Coders design, write, and test code using specific symbols, keywords, and syntax specific to the coding language they are using.

Programmers plan, develop, and maintain programs that use code. Programmers often know how to code, but their skills to organize and structure programs are more crucial.

They use a multi-stage process, known as the program development life cycle, that includes creating outlines, designing algorithms and flowcharts, applying complex rules and syntax, and testing and documenting software performance.

https://www.computerscience.org/resources/best-programming-languages-to-learn/

## **Top Programming Languages to Learn**

We prepared a list of the best programming languages to learn that incorporates factors such as field demand, skill level, and overall usefulness. There are additional programming languages beyond this list, but these are the top languages to learn for a [career in the tech sector](https://www.computerscience.org/resources/job-in-tech-hub/).

Jump to a programming language:  
**[Python](https://www.computerscience.org/resources/best-programming-languages-to-learn/" \l "python) | [JavaScript](https://www.computerscience.org/resources/best-programming-languages-to-learn/" \l "javascript) | [HTML/CSS](https://www.computerscience.org/resources/best-programming-languages-to-learn/" \l "html) | [C](https://www.computerscience.org/resources/best-programming-languages-to-learn/" \l "c) | [C++](https://www.computerscience.org/resources/best-programming-languages-to-learn/" \l "cplus) | [Go](https://www.computerscience.org/resources/best-programming-languages-to-learn/" \l "go) | [Swift](https://www.computerscience.org/resources/best-programming-languages-to-learn/" \l "swift) | [Java](https://www.computerscience.org/resources/best-programming-languages-to-learn/" \l "java) | [C#](https://www.computerscience.org/resources/best-programming-languages-to-learn/" \l "cpound) | [SQL](https://www.computerscience.org/resources/best-programming-languages-to-learn/" \l "sql) | [Rust](https://www.computerscience.org/resources/best-programming-languages-to-learn/" \l "rust) | [R](https://www.computerscience.org/resources/best-programming-languages-to-learn/" \l "r)**

### **Python**

Python debuted in 1991 and remains one of the most popular programming languages in the world. Great for beginners, Python's syntax is similar to English, requires fewer lines than many other languages, and is known for its overall readability.

Python can help create web applications, connect to database systems, and handle big data and complex mathematics. Apple, Google, and Meta use Python.

Python is free, and supported by a large library and [thriving community of users](https://www.computerscience.org/bootcamps/rankings/python/). Potential negatives of Python include slow speed and high memory consumption.

### **JavaScript**

JavaScript operates alongside HTML and CSS as one of the core languages of the internet. In use since 1996, JavaScript is fast, [simple for beginners](https://www.computerscience.org/bootcamps/rankings/javascript/), and has cross-platform uses. Originally designed just for web use, JavaScript is now part of some servers and applications.

Coders can write JavaScript as a separate file or embed it in other code. You can find JavaScript in multimedia applications and animation, creating an interactive experience for users.

JavaScript is a scripting language. This means it automates the execution of operations in a runtime environment, rather than compiling operations. In short, programs read and execute JavaScript.

Companies like Amazon, Apple, and Google use JavaScript.

### **HTML/CSS**

[HyperText Markup Language](https://www.computerscience.org/resources/computer-programming-languages/html/) (HTML) helps create and structure webpages and other online content. Released in 1993, HTML is often assisted by Cascading Style Sheets (CSS) or scripting languages such as JavaScript.

HTML uses tags and attributes alongside elements as the standard language of the internet. CSS describes elements, determining how they render online. CSS allows for web content to appear with color, different fonts, spacing, and other stylistic features. HTML and CSS are essential for all web developers to learn.

### **C**

Developed as a general-purpose language during the 1970s, C writes the operating system UNIX. C has a simple set of keywords, a formal structure, and clear syntax. Languages like C++ and JavaScript borrow heavily from C, with C++ specifically functioning as a superset of the language.

C has features that allow for use in low-level programming to script drivers, and high-level programming for software applications. Although a complicated language to learn because it communicates directly with hardware, C is portable, fast, and easy to write and understand. You can implement C in games, graphics, and applications needing calculations.

### **C++**

Object-oriented and portable, C++ works on multiple platforms. In 1985, C++ was developed as an extension of the C programming language. C++ is popular among game developers and other [expert programmers](https://www.computerscience.org/bootcamps/rankings/c-plus-plus/). You can use C++ to create browsers, operating systems, and banking applications. Twitter, YouTube, and Google use C++.

A large standard template library for C++ accompanies its speed, making it useful for large systems. However, C++ lacks security, contains pointers that can take up a lot of memory, and may be difficult to debug.

### **Go (or Golang)**

Go, also called Golang, is a general-purpose, open-source language Google developed in 2009. It resembles C and has built-in abilities and library support for writing concurrent programs. Go features asynchrony and lets slow operations run alongside other programs.

A good starting language for beginners, Go is easy to use, has a standard library, and has built-in security. Cloud and network services use Go, as do [web development](https://www.computerscience.org/careers/web-developer/) and game development.

The tenth most popular programming language in the world, Go has support from Google — one of its major users — Netflix, Meta, and PayPal.

### **Swift**

Apple developed Swift in 2014 as a replacement for all languages in C. [Swift is general-purpose](https://www.computerscience.org/resources/computer-programming-languages/swift/) and meant to make writing and maintaining software easy. Swift is comparable to C-based languages, but differs in efficiency, safety, and overall speed.

Good for newcomers and experts, Swift employs user-friendly features like concise syntax and inferred type, making program development safe, fast, and nearly error-free. Swift is likely to replace [Objective-C](https://www.computerscience.org/resources/computer-programming-languages/objective-c/) as the major language of iOS. Unlike Objective-C, Swift's code lacks brackets, parentheses, and semicolons in certain contexts.

You can use Swift in [mobile and desktop applications](https://www.computerscience.org/bootcamps/rankings/mobile-app-development/), cloud services, and systems programming.

### **Java**

Java is a class-based, object-oriented programming language created in 1995. Java embraces the general-purpose programming language goal of "write once, run anywhere." This means compiled Java code can run on any and all platforms that support Java.

Perfect for [beginners and experts](https://www.computerscience.org/bootcamps/rankings/java/) alike, Java features syntax similar to C and C++. As a server-side language, Java functions on the back end of development.

You can use Java for desktop, mobile, and numerical computing alongside game development and big data projects.

Due to its computing abilities, Financial and e-commerce industries use Java. Uber, Airbnb, and Microsoft use Java, as do IBM, Google, and Amazon.

### **C#**

As a member of the family of C languages, C# debuted in 2000. Microsoft developed C# for use on [the .NET platform](https://www.computerscience.org/bootcamps/rankings/net/). It continues to integrate well with Windows.

You can use C# to develop web, desktop, and mobile applications. Because it is intricately connected to the .NET platform, this may limit its usefulness.

C# resembles C, C++, and Java in its object orientation, but differs from the latter with its common language runtime.

C# has memory access capabilities and a large community of users. Although C# is harder to learn than C and C++ and has a slower runtime than the latter, it is possible for a dedicated beginner to tackle this language.

### **SQL**

[Structured Query Language](https://www.computerscience.org/resources/computer-programming-languages/sql/) (SQL) communicates between relational databases. Developed during the 1970s, SQL can update, retrieve, or delete database information.

SQL is portable, fast, and relatively user-friendly. This makes it [easy to learn](https://www.computerscience.org/bootcamps/guides/online-coding-courses/), requiring little in the way of coding abilities.

MySQL is an open-source database built-in SQL. You can use SQL on relational database management systems like Oracle, Microsoft SQL Server, and Microsoft Access.

In 1986, SQL became a standard of the American National Standards Institute. The following year, it became a standard of the International Organization for Standardization.

### **Rust**

Developed by Mozilla in 2010 to solve the Elevator Problem, which refers to the concentration of specific operations in various areas of a disk, Rust was considered an improvement on C languages for building high-programming applications.

An advanced language, Rust works well for developing Internet of Things (IoT) applications. It is frequently used to build operating systems, gaming engines, and scalable browsers.

Rust's zero-cost abstraction allows developers to hide code details, and the automatic memory management feature means it works without a garbage collector.

Atlassian, Meta, Microsoft, and several cryptocurrencies, including BitFury and Libra, use Rust.

### **R**

Influenced by the language S and developed by statisticians at the University of Auckland, R provides a free system for graphic and statistical computation. It can conduct statistical procedures such as clustering, nonlinear regression models, linear models, and time series analysis.

Compatible with C, C++, or FORTRAN languages, R is a highly efficient system that can run on Unix-like, Windows, and Mac operating systems.

Statisticians, researchers, academics, and data scientists primarily use R for data exploration, predictive modeling, and developing statistical software.

Although a complicated language for beginners, R is a valuable tool for programming experts.