<https://www.britannica.com/science/computer-science/Programming-languages>

[Programming languages](https://www.britannica.com/technology/computer-programming-language) are the languages with which a programmer [implements](https://www.merriam-webster.com/dictionary/implements) a piece of [software](https://www.britannica.com/technology/software) to run on a computer. The earliest programming languages were [assembly languages](https://www.britannica.com/technology/assembly-language), not far removed from the [binary](https://www.britannica.com/science/binary-number-system)-encoded instructions directly executed by the computer. By the mid-1950s, programmers began to use higher-level languages.

Two of the first higher-level languages were [FORTRAN](https://www.britannica.com/technology/FORTRAN) (Formula Translator) and [ALGOL](https://www.britannica.com/technology/ALGOL-computer-language) (Algorithmic Language), which allowed programmers to write algebraic expressions and solve scientific computing problems. As learning to program became increasingly important in the 1960s, a stripped-down version of FORTRAN called [BASIC](https://www.britannica.com/technology/BASIC) (Beginner’s All-Purpose Symbolic Instruction Code) was developed at [Dartmouth College](https://www.britannica.com/topic/Dartmouth-College). BASIC quickly spread to other academic institutions, and by 1980 versions of BASIC for [personal computers](https://www.britannica.com/technology/personal-computer) allowed even students at elementary schools to learn the fundamentals of [programming](https://www.britannica.com/dictionary/programming). Also, in the mid-1950s, [COBOL](https://www.britannica.com/technology/COBOL) (Common Business-Oriented Language) was developed to support business programming applications that involved managing information stored in records and files.

The trend since then has been toward developing increasingly abstract languages, allowing the programmer to communicate with the [machine](https://www.britannica.com/technology/machine) at a level ever more remote from [machine code](https://www.britannica.com/technology/machine-language). COBOL, FORTRAN, and their descendants ([Pascal](https://www.britannica.com/technology/Pascal-computer-language) and [C](https://www.britannica.com/technology/C-computer-programming-language), for example) are known as imperative languages, since they specify as a sequence of explicit commands how the machine is to go about solving the problem at hand. These languages were also known as procedural languages, since they allowed programmers to develop and reuse procedures, subroutines, and functions to avoid reinventing basic tasks for every new application.

Other high-level languages are called [functional languages](https://www.britannica.com/technology/functional-language), in that a program is viewed as a collection of (mathematical) functions and its semantics are very precisely defined. The best-known functional language of this type is [LISP](https://www.britannica.com/technology/LISP-computer-language) (List Processing), which in the 1960s was the mainstay programming language for [AI](https://www.britannica.com/technology/artificial-intelligence) applications. Successors to LISP in the AI [community](https://www.merriam-webster.com/dictionary/community) include Scheme, Prolog, and C and [C++](https://www.britannica.com/technology/C-computer-language) (see below). Scheme is similar to LISP except that it has a more formal mathematical definition. [Prolog](https://www.britannica.com/technology/PROLOG) has been used largely for [logic programming](https://www.britannica.com/technology/logic-programming-language), and its applications include natural language understanding and expert systems such as MYCIN. Prolog is notably a so-called [nonprocedural](https://www.britannica.com/technology/declarative-language), or declarative, language in the sense that the programmer specifies what goals are to be accomplished but not how specific methods are to be applied to [attain](https://www.britannica.com/dictionary/attain) those goals. C and C++ have been used widely in [robotics](https://www.britannica.com/technology/robotics), an important application of AI research. An extension of logic programming is constraint logic programming, in which pattern matching is replaced by the more general operation of constraint satisfaction.

Another important development in programming languages through the 1980s was the addition of support for data encapsulation, which gave rise to [object-oriented languages](https://www.britannica.com/technology/object-oriented-language). The original object-oriented language was called [Smalltalk](https://www.britannica.com/technology/Smalltalk), in which all programs were represented as collections of objects communicating with each other via message-passing. An object is a [set](https://www.britannica.com/topic/set-mathematics-and-logic) of data together with the methods (functions) that can transform that data. Encapsulation refers to the fact that an object’s data can be accessed only through these methods. Object-oriented programming has been very influential in computing. Languages for [object-oriented programming](https://www.britannica.com/technology/object-oriented-programming) include C++, Visual BASIC, and [Java](https://www.britannica.com/technology/Java-computer-programming-language).

Java is unusual because its applications are translated not into a particular machine language but into an intermediate language called Java Bytecode, which runs on the Java Virtual Machine (JVM). Programs on the JVM can be executed on most contemporary computer platforms, including [Intel](https://www.britannica.com/topic/Intel)-based systems, [Apple](https://www.britannica.com/plant/apple-fruit-and-tree) Macintoshes, and various [Android](https://www.britannica.com/technology/Android-operating-system)-based smartphones and tablets. Thus, [Linux](https://www.britannica.com/technology/Linux), iOS, [Windows](https://www.britannica.com/technology/Microsoft-Windows), and other [operating systems](https://www.britannica.com/technology/operating-system) can run Java programs, which makes Java ideal for creating distributed and Web-based applications. Residing on [Web-based servers](https://www.britannica.com/technology/server), Java programs may be downloaded and run in any standard [Web browser](https://www.britannica.com/technology/browser) to provide access to various services, such as a client interface to a game or entry to a [database](https://www.britannica.com/technology/database) residing on a server.

At a still higher level of [abstraction](https://www.britannica.com/dictionary/abstraction) lie declarative and scripting languages, which are strictly interpreted languages and often drive applications running in Web browsers and mobile devices. Some declarative languages allow programmers to conveniently access and retrieve information from a database using “queries,” which are declarations of what to do (rather than how to do it). A widely used database [query language](https://www.britannica.com/technology/query-language) is [SQL](https://www.britannica.com/technology/SQL) (Structured Query Language) and its variants (e.g., MySQL and SQLite). Associated with these declarative languages are those that describe the layout of a Web page on the user’s screen. For example, [HTML](https://www.britannica.com/technology/HTML) (HyperText Markup Language) supports the design of Web pages by specifying their structure and content. Gluing the Web page together with the database is the task of a [scripting language](https://www.britannica.com/technology/computer-scripting-language) (e.g., PHP), which is a vehicle for programmers to [integrate](https://www.merriam-webster.com/dictionary/integrate) declarative statements of HTML and MySQL with [imperative](https://www.merriam-webster.com/dictionary/imperative) actions that are required to effect an interaction between the user and the database. An example is an online book order with [Amazon.com](https://www.britannica.com/topic/Amazoncom), where the user queries the database to find out what books are available and then initiates an order by pressing buttons and filling appropriate text areas with his or her ordering information. The software that underlies this activity includes HTML to describe the content of the Web page, MySQL to access the database according to the user’s requests, and PHP to control the overall flow of the transaction.

Computer programs written in any language other than machine language must be either interpreted or translated into machine language (“compiled”). As suggested above, an [interpreter](https://www.britannica.com/technology/interpreter) is software that examines a [computer program](https://www.britannica.com/technology/computer-program) one instruction at a time and calls on [code](https://www.britannica.com/topic/code-communications) to execute the machine operations required by that instruction.

A [compiler](https://www.britannica.com/technology/compiler) is software that translates an entire computer program into machine code that is saved for subsequent execution whenever desired. Much work has been done on making both the [compilation](https://www.merriam-webster.com/dictionary/compilation) process and the compiled code as efficient as possible. When a new language is developed, it is usually interpreted at first. If it later becomes popular, a compiler is developed for it, since compilation is more efficient than interpretation.

There is an intermediate approach, which is to compile code not into machine language but into an intermediate language (called a virtual machine) that is close enough to machine language that it is efficient to interpret, though not so close that it is tied to the machine language of a particular computer. It is this approach that provides the Java language with its computer platform independence via the JVM.

Языки программирования - это языки, с помощью которых программист реализует часть программного обеспечения для работы на компьютере. Самыми первыми языками программирования были языки ассемблера, недалеко ушедшие от двоично-кодированных инструкций, непосредственно выполняемых компьютером. К середине 1950-х годов программисты начали использовать языки более высокого уровня.

Двумя из первых языков высокого уровня были FORTRAN (транслятор формул) и ALGOL (алгоритмический язык), которые позволяли программистам писать алгебраические выражения и решать научные вычислительные задачи. Поскольку в 1960-х годах обучение программированию становилось все более важным, в Дартмутском колледже была разработана сокращенная версия языка FORTRAN под названием BASIC (Beginner's All-Purpose Symbolic Instruction Code). BASIC быстро распространился в других учебных заведениях, а к 1980 году версии BASIC для персональных компьютеров позволили даже ученикам начальных школ изучать основы программирования. Кроме того, в середине 1950-х годов был разработан язык COBOL (Common Business-Oriented Language) для поддержки приложений бизнес-программирования, связанных с управлением информацией, хранящейся в записях и файлах.

С тех пор наблюдается тенденция к разработке все более абстрактных языков, позволяющих программисту общаться с машиной на уровне, все более удаленном от машинного кода. COBOL, FORTRAN и их потомки (например, Pascal и C) известны как императивные языки, поскольку они определяют в виде последовательности явных команд, как машина должна решать поставленную задачу. Эти языки также известны как процедурные языки, поскольку они позволяют программистам разрабатывать и повторно использовать процедуры, подпрограммы и функции, чтобы не изобретать базовые задачи для каждого нового приложения.

Другие языки высокого уровня называются функциональными, поскольку в них программа рассматривается как набор (математических) функций, а ее семантика очень точно определена. Наиболее известным функциональным языком такого типа является LISP (List Processing), который в 1960-х годах был основным языком программирования для приложений ИИ. Преемниками LISP в сообществе ИИ являются Scheme, Prolog, а также C и C++ (см. ниже). Scheme похож на LISP, за исключением того, что имеет более формальное математическое определение. Prolog использовался в основном для логического программирования, и его приложения включают понимание естественного языка и экспертные системы, такие как MYCIN. Пролог является так называемым непроцедурным, или декларативным, языком в том смысле, что программист указывает, какие цели должны быть достигнуты, но не то, какие конкретные методы должны быть применены для достижения этих целей. Языки C и C++ широко используются в робототехнике - важном приложении исследований в области ИИ. Расширением логического программирования является логическое программирование с ограничениями, в котором сопоставление шаблонов заменяется более общей операцией удовлетворения ограничений.

Другим важным направлением развития языков программирования в 1980-х годах стало добавление поддержки инкапсуляции данных, что привело к появлению объектно-ориентированных языков. Оригинальный объектно-ориентированный язык назывался Smalltalk, в котором все программы были представлены как коллекции объектов, взаимодействующих друг с другом посредством передачи сообщений. Объект - это набор данных вместе с методами (функциями), которые могут преобразовывать эти данные. Инкапсуляция означает, что доступ к данным объекта возможен только через эти методы. Объектно-ориентированное программирование оказало большое влияние на вычислительную технику. Языки для объектно-ориентированного программирования включают C++, Visual BASIC и Java.

Java необычен тем, что его приложения переводятся не на конкретный машинный язык, а на промежуточный язык под названием Java Bytecode, который выполняется на виртуальной машине Java Virtual Machine (JVM). Программы на JVM могут выполняться на большинстве современных компьютерных платформ, включая системы на базе Intel, Apple Macintoshes и различные смартфоны и планшеты на базе Android. Таким образом, Linux, iOS, Windows и другие операционные системы могут запускать Java-программы, что делает Java идеальной для создания распределенных и веб-приложений. Расположенные на веб-серверах, Java-программы могут быть загружены и запущены в любом стандартном веб-браузере для обеспечения доступа к различным сервисам, таким как клиентский интерфейс для игры или вход в базу данных, расположенную на сервере.

На еще более высоком уровне абстракции находятся декларативные языки и языки сценариев, которые являются строго интерпретируемыми языками и часто служат основой для приложений, запускаемых в веб-браузерах и на мобильных устройствах. Некоторые декларативные языки позволяют программистам удобно обращаться к базе данных и извлекать из нее информацию с помощью "запросов", которые представляют собой декларации того, что нужно сделать (а не как это сделать). Широко используемым языком запросов к базам данных является SQL (Structured Query Language) и его разновидности (например, MySQL и SQLite).

С этими декларативными языками связаны те, которые описывают расположение веб-страницы на экране пользователя. Например, HTML (HyperText Markup Language) поддерживает проектирование веб-страниц, определяя их структуру и содержание. Склеивание веб-страницы с базой данных - задача языка сценариев (например, PHP), который позволяет программистам объединить декларативные утверждения HTML и MySQL с императивными действиями, необходимыми для осуществления взаимодействия между пользователем и базой данных. Примером может служить онлайн-заказ книг на Amazon.com, где пользователь запрашивает базу данных, чтобы узнать, какие книги имеются в наличии, а затем инициирует заказ, нажимая кнопки и заполняя соответствующие текстовые области своей информацией о заказе. Программное обеспечение, лежащее в основе этой деятельности, включает HTML для описания содержимого веб-страницы, MySQL для доступа к базе данных в соответствии с запросами пользователя и PHP для управления общим ходом транзакции.

Компьютерные программы, написанные на любом языке, отличном от машинного, должны быть либо интерпретированы, либо переведены на машинный язык ("скомпилированы"). Как было сказано выше, интерпретатор - это программное обеспечение, которое рассматривает компьютерную программу по одной инструкции за раз и вызывает код для выполнения машинных операций, требуемых этой инструкцией.

Компилятор - это программное обеспечение, которое переводит всю компьютерную программу в машинный код, сохраняемый для последующего выполнения в любое время. Была проделана большая работа над тем, чтобы сделать процесс компиляции и скомпилированный код как можно более эффективным. Когда разрабатывается новый язык, он обычно сначала интерпретируется. Если впоследствии он становится популярным, для него разрабатывается компилятор, поскольку компиляция более эффективна, чем интерпретация.

Существует промежуточный подход, который заключается в компиляции кода не в машинный язык, а в промежуточный язык (называемый виртуальной машиной), который достаточно близок к машинному языку, чтобы его можно было эффективно интерпретировать, но не настолько близок, чтобы привязать его к машинному языку конкретного компьютера. Именно этот подход обеспечивает языку Java независимость от компьютерной платформы через JVM.