**Java** is a [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [general-purpose](https://en.wikipedia.org/wiki/General-purpose_programming_language), [memory-safe](https://en.wikipedia.org/wiki/Memory_safety), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) [programming language](https://en.wikipedia.org/wiki/Programming_language). It is intended to let [programmers](https://en.wikipedia.org/wiki/Programmer) *write once, run anywhere* ([WORA](https://en.wikipedia.org/wiki/Write_once,_run_anywhere)),[[18]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-18) meaning that [compiled](https://en.wikipedia.org/wiki/Compiler) Java code can run on all platforms that support Java without the need to recompile.[[19]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-design_goals-19) Java applications are typically compiled to [bytecode](https://en.wikipedia.org/wiki/Java_bytecode" \o "Java bytecode) that can run on any [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) regardless of the underlying [computer architecture](https://en.wikipedia.org/wiki/Computer_architecture). The [syntax](https://en.wikipedia.org/wiki/Syntax_(programming_languages)) of Java is similar to [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B), but has fewer [low-level](https://en.wikipedia.org/wiki/Low-level_programming_language) facilities than either of them. The Java runtime provides dynamic capabilities (such as [reflection](https://en.wikipedia.org/wiki/Reflective_programming) and runtime code modification) that are typically not available in traditional compiled languages.

Java gained popularity shortly after its release, and has been a popular programming language since then.[[20]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-20) Java was the third most popular programming language in 2022 according to [GitHub](https://en.wikipedia.org/wiki/GitHub" \o "GitHub).[[21]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-21) Although still widely popular, there has been a gradual decline in use of Java in recent years with [other languages using JVM](https://en.wikipedia.org/wiki/List_of_JVM_languages) gaining popularity.[[22]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-:0-22)

Java was designed by [James Gosling](https://en.wikipedia.org/wiki/James_Gosling) at [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems). It was released in May 1995 as a core component of Sun's [Java platform](https://en.wikipedia.org/wiki/Java_(software_platform)). The original and [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) Java [compilers](https://en.wikipedia.org/wiki/Compiler), virtual machines, and [class libraries](https://en.wikipedia.org/wiki/Library_(computing)) were released by Sun under [proprietary licenses](https://en.wikipedia.org/wiki/Proprietary_license). As of May 2007, in compliance with the specifications of the [Java Community Process](https://en.wikipedia.org/wiki/Java_Community_Process), Sun had [relicensed](https://en.wikipedia.org/wiki/Software_relicensing) most of its Java technologies under the [GPL-2.0-only](https://en.wikipedia.org/wiki/GNU_General_Public_License) license. [Oracle](https://en.wikipedia.org/wiki/Oracle_Corporation), which bought Sun in 2010, offers its own [HotSpot](https://en.wikipedia.org/wiki/HotSpot_(virtual_machine)" \o "HotSpot (virtual machine)) Java Virtual Machine. However, the official [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) is the [OpenJDK](https://en.wikipedia.org/wiki/OpenJDK" \o "OpenJDK) JVM, which is open-source software used by most developers and is the default JVM for almost all Linux distributions.

[Java 24](https://en.wikipedia.org/wiki/Java_version_history) is the version current as of March 2025. Java 8, 11, 17, and 21 are [long-term support](https://en.wikipedia.org/wiki/Long-term_support) versions still under maintenance.

History

[James Gosling](https://en.wikipedia.org/wiki/James_Gosling), Mike Sheridan, and [Patrick Naughton](https://en.wikipedia.org/wiki/Patrick_Naughton) initiated the Java language project in June 1991.[[23]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-23) Java was originally designed for interactive television, but it was too advanced for the digital cable television industry at the time.[[24]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-24) The language was initially called [*Oak*](https://en.wikipedia.org/wiki/Oak_(programming_language)) after an [oak](https://en.wikipedia.org/wiki/Oak) tree that stood outside Gosling's office. Later the project went by the name *Green* and was finally renamed *Java*, from [Java coffee](https://en.wikipedia.org/wiki/Java_coffee), a type of coffee from [Indonesia](https://en.wikipedia.org/wiki/Indonesia).[[25]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-25) Gosling designed Java with a [C](https://en.wikipedia.org/wiki/C_(programming_language))/[C++](https://en.wikipedia.org/wiki/C%2B%2B)-style syntax that system and application programmers would find familiar.[[26]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-26)

Sun Microsystems released the first public implementation as Java 1.0 in 1996.[[27]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-oraclejavahistory-27) It promised [write once, run anywhere](https://en.wikipedia.org/wiki/Write_once,_run_anywhere) (WORA) functionality, providing no-cost run-times on popular [platforms](https://en.wikipedia.org/wiki/Computing_platform). Fairly secure and featuring configurable security, it allowed network- and file-access restrictions. Major [web browsers](https://en.wikipedia.org/wiki/Web_browser) soon incorporated the ability to run [Java applets](https://en.wikipedia.org/wiki/Java_applet) within web pages, and Java quickly became popular. The Java 1.0 compiler was re-written [in Java](https://en.wikipedia.org/wiki/Bootstrapping_(compilers)) by [Arthur van Hoff](https://en.wikipedia.org/wiki/Arthur_van_Hoff) to comply strictly with the Java 1.0 language specification.[[28]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-28) With the advent of Java 2 (released initially as J2SE 1.2 in December 1998 – 1999), new versions had multiple configurations built for different types of platforms. [J2EE](https://en.wikipedia.org/wiki/Java_2_Enterprise_Edition) included technologies and APIs for enterprise applications typically run in server environments, while J2ME featured APIs optimized for mobile applications. The desktop version was renamed J2SE. In 2006, for marketing purposes, Sun renamed new J2 versions as [*Java EE*](https://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition), [*Java ME*](https://en.wikipedia.org/wiki/Java_Platform,_Micro_Edition), and [*Java SE*](https://en.wikipedia.org/wiki/Java_Platform,_Standard_Edition), respectively.

In 1997, Sun Microsystems approached the [ISO/IEC JTC 1](https://en.wikipedia.org/wiki/ISO/IEC_JTC_1) standards body and later the [Ecma International](https://en.wikipedia.org/wiki/Ecma_International" \o "Ecma International) to formalize Java, but it soon withdrew from the process.[[29]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-29)[[30]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-30)[[31]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-31) Java remains a [*de facto* standard](https://en.wikipedia.org/wiki/De_facto_standard), controlled through the [Java Community Process](https://en.wikipedia.org/wiki/Java_Community_Process).[[32]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-32) At one time, Sun made most of its Java implementations available without charge, despite their [proprietary software](https://en.wikipedia.org/wiki/Proprietary_software) status. Sun generated revenue from Java through the selling of licenses for specialized products such as the Java Enterprise System.

On November 13, 2006, Sun released much of its Java virtual machine (JVM) as [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software) (FOSS), under the terms of the [GPL-2.0-only](https://en.wikipedia.org/wiki/GNU_General_Public_License) license. On May 8, 2007, Sun finished the process, making all of its JVM's core code available under [free software](https://en.wikipedia.org/wiki/Free_software)/open-source distribution terms, aside from a small portion of code to which Sun did not hold the copyright.[[33]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-33)

Sun's vice-president Rich Green said that Sun's ideal role with regard to Java was as an *evangelist*.[[34]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-34) Following [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation)'s acquisition of Sun Microsystems in 2009–10, Oracle has described itself as the steward of Java technology with a relentless commitment to fostering a community of participation and transparency.[[35]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-35) This did not prevent Oracle from filing a lawsuit against Google shortly after that for using Java inside the [Android SDK](https://en.wikipedia.org/wiki/Android_SDK) (see the [*Android*](https://en.wikipedia.org/wiki/Java_(programming_language)#Android) section).

On April 2, 2010, James Gosling resigned from [Oracle](https://en.wikipedia.org/wiki/Oracle_Corporation).[[36]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-36)

In January 2016, Oracle announced that Java run-time environments based on JDK 9 will discontinue the browser plugin.[[37]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-37)

Java software runs on most devices from laptops to [data centers](https://en.wikipedia.org/wiki/Data_center), [game consoles](https://en.wikipedia.org/wiki/Video_game_console) to scientific [supercomputers](https://en.wikipedia.org/wiki/Supercomputer).[[38]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-38)

[Oracle](https://en.wikipedia.org/wiki/Oracle_Corporation) (and others) highly recommend uninstalling outdated and unsupported versions of Java, due to unresolved security issues in older versions.[[39]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-39)

### Principles

There were five primary goals in creating the Java language:[[19]](https://en.wikipedia.org/wiki/Java_(programming_language)" \l "cite_note-design_goals-19)

1. It must be simple, [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming), and familiar.
2. It must be [robust](https://en.wikipedia.org/wiki/Robustness_(computer_science)) and secure.
3. It must be architecture-neutral and portable.
4. It must execute with high performance.
5. It must be [interpreted](https://en.wikipedia.org/wiki/Interpreter_(computing)), [threaded](https://en.wikipedia.org/wiki/Thread_(computing)), and [dynamic](https://en.wikipedia.org/wiki/Dynamic_programming_language).

### Versions

*Main article:*[*Java version history*](https://en.wikipedia.org/wiki/Java_version_history)

As of November 2024, Java 8, 11, 17, and 21 are supported as [long-term support](https://en.wikipedia.org/wiki/Long-term_support) (LTS) versions, with Java 25, releasing in September 2025, as the next scheduled LTS version.[[40]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-40)

Oracle released the last zero-cost public update for the [legacy](https://en.wikipedia.org/wiki/Legacy_system) version [Java 8](https://en.wikipedia.org/wiki/Java_8) LTS in January 2019 for commercial use, although it will otherwise still support Java 8 with public updates for personal use indefinitely. Other vendors such as [Adoptium](https://en.wikipedia.org/wiki/Adoptium" \o "Adoptium) continue to offer free builds of OpenJDK's long-term support (LTS) versions. These builds may include additional security patches and bug fixes.[[41]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-41)

## Execution system

### Java JVM and bytecode

*Main articles:*[*Java (software platform)*](https://en.wikipedia.org/wiki/Java_(software_platform))*and*[*Java virtual machine*](https://en.wikipedia.org/wiki/Java_virtual_machine)

One design goal of Java is [portability](https://en.wikipedia.org/wiki/Software_portability), which means that programs written for the Java platform must run similarly on any combination of hardware and operating system with adequate run time support. This is achieved by compiling the Java language code to an intermediate representation called [Java bytecode](https://en.wikipedia.org/wiki/Java_bytecode), instead of directly to architecture-specific [machine code](https://en.wikipedia.org/wiki/Machine_code). Java bytecode instructions are analogous to machine code, but they are intended to be executed by a [virtual machine](https://en.wikipedia.org/wiki/Virtual_machine) (VM) written specifically for the host hardware. [End-users](https://en.wikipedia.org/wiki/End_user) commonly use a [Java Runtime Environment](https://en.wikipedia.org/wiki/Java_virtual_machine) (JRE) installed on their device for standalone Java applications or a web browser for [Java applets](https://en.wikipedia.org/wiki/Java_applet).

Standard libraries provide a generic way to access host-specific features such as graphics, [threading](https://en.wikipedia.org/wiki/Thread_(computing)), and [networking](https://en.wikipedia.org/wiki/Computer_network).

The use of universal bytecode makes porting simple. However, the overhead of [interpreting](https://en.wikipedia.org/wiki/Interpreter_(computing)) bytecode into machine instructions made interpreted programs almost always run more slowly than native [executables](https://en.wikipedia.org/wiki/Executable). [Just-in-time](https://en.wikipedia.org/wiki/Just-in-time_compilation) (JIT) compilers that compile byte-codes to machine code during runtime were introduced from an early stage. Java's Hotspot compiler is actually two compilers in one; and with [GraalVM](https://en.wikipedia.org/wiki/GraalVM" \o "GraalVM) (included in e.g. Java 11, but removed as of Java 16) allowing [tiered compilation](https://en.wikipedia.org/w/index.php?title=Tiered_compilation&action=edit&redlink=1).[[51]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-51) Java itself is platform-independent and is adapted to the particular platform it is to run on by a [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine) (JVM), which translates the [Java bytecode](https://en.wikipedia.org/wiki/Java_bytecode) into the platform's machine language.[[52]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-52)

#### Performance

*Main article:*[*Java performance*](https://en.wikipedia.org/wiki/Java_performance)

Programs written in Java have a reputation for being slower and requiring more memory than those written in [C++](https://en.wikipedia.org/wiki/C%2B%2B).[[53]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-Jelovic-53)[[54]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-54) However, Java programs' execution speed improved significantly with the introduction of [just-in-time compilation](https://en.wikipedia.org/wiki/Just-in-time_compilation) in 1997/1998 for [Java 1.1](https://en.wikipedia.org/wiki/Java_version_history),[[55]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-55) the addition of language features supporting better code analysis (such as inner classes, the StringBuilder class, optional assertions, etc.), and optimizations in the Java virtual machine, such as [HotSpot](https://en.wikipedia.org/wiki/HotSpot_(virtual_machine)" \o "HotSpot (virtual machine)) becoming Sun's default JVM in 2000. With Java 1.5, the performance was improved with the addition of the java.util.concurrent package, including [lock-free](https://en.wikipedia.org/wiki/Lock-free) implementations of the [ConcurrentMaps](https://en.wikipedia.org/wiki/Java_ConcurrentMap" \o "Java ConcurrentMap) and other multi-core collections, and it was improved further with Java 1.6.

### Non-JVM

Some platforms offer direct hardware support for Java; there are micro controllers that can run Java bytecode in hardware instead of a software Java virtual machine,[[56]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-56) and some [ARM](https://en.wikipedia.org/wiki/ARM_architecture_family)-based processors could have hardware support for executing Java bytecode through their [Jazelle](https://en.wikipedia.org/wiki/Jazelle" \o "Jazelle) option, though support has mostly been dropped in current implementations of ARM.

### Automatic memory management

Java uses an [automatic garbage collector](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)) to manage memory in the [object lifecycle](https://en.wikipedia.org/wiki/Object_lifetime). The programmer determines when objects are created, and the Java runtime is responsible for recovering the memory once objects are no longer in use. Once no references to an object remain, the [unreachable memory](https://en.wikipedia.org/wiki/Unreachable_memory) becomes eligible to be freed automatically by the garbage collector. Something similar to a [memory leak](https://en.wikipedia.org/wiki/Memory_leak) may still occur if a programmer's code holds a reference to an object that is no longer needed, typically when objects that are no longer needed are stored in containers that are still in use.[[57]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-FOOTNOTEBloch201826-28%C2%A7Item_7:_Eliminate_obsolete_object_references-57) If methods for a non-existent object are called, a [null pointer](https://en.wikipedia.org/wiki/Null_pointer) exception is thrown.[[58]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-58)[[59]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-59)

One of the ideas behind Java's automatic memory management model is that programmers can be spared the burden of having to perform manual memory management. In some languages, memory for the creation of objects is implicitly allocated on the [stack](https://en.wikipedia.org/wiki/Stack_(abstract_data_type)) or explicitly allocated and deallocated from the [heap](https://en.wikipedia.org/wiki/Memory_management#DYNAMIC). In the latter case, the responsibility of managing memory resides with the programmer. If the program does not deallocate an object, a [memory leak](https://en.wikipedia.org/wiki/Memory_leak) occurs.[[57]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-FOOTNOTEBloch201826-28%C2%A7Item_7:_Eliminate_obsolete_object_references-57) If the program attempts to access or deallocate memory that has already been deallocated, the result is undefined and difficult to predict, and the program is likely to become unstable or crash. This can be partially remedied by the use of [smart pointers](https://en.wikipedia.org/wiki/Smart_pointer), but these add overhead and complexity. Garbage collection does not prevent [logical memory](https://en.wikipedia.org/wiki/Logical_address) leaks, i.e. those where the memory is still referenced but never used.[[57]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-FOOTNOTEBloch201826-28%C2%A7Item_7:_Eliminate_obsolete_object_references-57)

Garbage collection may happen at any time. Ideally, it will occur when a program is idle. It is guaranteed to be triggered if there is insufficient free memory on the heap to allocate a new object; this can cause a program to stall momentarily. Explicit memory management is not possible in Java.

Java does not support C/C++ style [pointer arithmetic](https://en.wikipedia.org/wiki/Pointer_(computer_programming)),[[60]](https://en.wikipedia.org/wiki/Java_(programming_language)" \l "cite_note-60) where object addresses can be arithmetically manipulated (e.g. by adding or subtracting an offset). This allows the garbage collector to relocate referenced objects and ensures type safety and security.

As in C++ and some other object-oriented languages, variables of Java's [primitive data types](https://en.wikipedia.org/wiki/Primitive_data_type) are either stored directly in fields (for objects) or on the [stack](https://en.wikipedia.org/wiki/Stack-based_memory_allocation) (for methods) rather than on the heap, as is commonly true for non-primitive data types (but see [escape analysis](https://en.wikipedia.org/wiki/Escape_analysis)). This was a conscious decision by Java's designers for performance reasons.

Java contains multiple types of garbage collectors. Since Java 9, HotSpot uses the [Garbage First Garbage Collector](https://en.wikipedia.org/wiki/Garbage-first_collector) (G1GC) as the default.[[61]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-61) However, there are also several other garbage collectors that can be used to manage the heap, such as the Z Garbage Collector (ZGC) introduced in Java 11, and Shenandoah GC, introduced in Java 12 but unavailable in Oracle-produced OpenJDK builds. Shenandoah is instead available in third-party builds of OpenJDK, such as [Eclipse Temurin](https://en.wikipedia.org/wiki/Adoptium#Eclipse_Temurin). For most applications in Java, G1GC is sufficient. In prior versions of Java, such as Java 8, the [Parallel Garbage Collector](https://docs.oracle.com/javase/8/docs/technotes/guides/vm/gctuning/parallel.html) was used as the default garbage collector.

Having solved the memory management problem does not relieve the programmer of the burden of handling properly other kinds of resources, like network or database connections, file handles, etc., especially in the presence of exceptions.