

History

On 23 May 1995, John Gage, the director of the Science Office of the Sun Microsystems along with Marc Andreessen, co-founder and executive vice president at Netscape announced to an audience of SunWorld™ that Java technology wasn't a myth and that it was a reality and that it was going to be incorporated into Netscape Navigator.^[1]

At the time the total number of people working on Java were less than 30 people.^[1] This team would shape the future in the next decade and no one had an idea as to what was in store. From being the mind of an unmanned vehicle on Mars to the operating environment on most of the consumer electronics, e.g., cable set-top boxes, VCR's, toasters, and also for personal data assistants (PDAs).^[2] Java has come a long way from its inception. Let's see how it all began.

The Green team

Behind closed doors, a project was initiated in December of 1990, whose aim was to create a programming tool that could render obsolete the C and C++ programming languages. Engineer Patrick Naughton had become extremely frustrated with the state of Sun's C++ and C APIs (application programming interfaces) and tools. While he was considering to move towards NeXT, he was offered a chance to work on new technology and the "Stealth Project" was started, a secret nobody but he knew.

This Stealth Project was later named the "Green Project" when James Gosling and Mike Sheridan joined Patrick.^[1] Over the period of time that the Green Project teethed, the prospects of the project started becoming clearer to the engineers working on it. No longer was it's aim to create a new language far superior to the present ones, but it aimed to target the language to devices other than the computer.



James Gosling, architect and designer of the compiler for the Java technology

Staffed at 13 people, they began work in a small office on Sand Hill Road in Menlo Park, California. This team would be called "Green Team" henceforth in time. The project they underwent was chartered by Sun Microsystems to anticipate and plan for the "next-wave" in computing. For the team, this meant at least one significant trend, that of the convergence of digitally controlled consumer devices and computers.^[1]

Reshaping thought

The team started thinking of replacing C++ with a better version, a faster version, a responsive version. But the one thing they hadn't thought of, as of yet, was that the language they were aiming for, had to be developed for an embedded system with limited resources. An **embedded system** is a computer system scaled to a minimalistic interface demanding only a few functions from its design. For such a system, C++ or any successor would seem too large as all the languages at the time demanded a larger footprint than what was desired. And, other than this, the language lacked some other important features as well. The team thus had to think in a different way to go about solving all these problems.

Co-founder of Sun Microsystems, Bill Joy, envisioned a language combining the power of Mesa and C in a paper he wrote for the engineers at Sun named *Further*. Gathering ideas, Gosling began work on enhancing C++ and named it "C++ ++ --", a pun on the evolutionary structure of the language's name. The ++ and -- meant, *putting in* and *taking out stuff*. He soon abandoned the name and called it **Oak**^[1] after the tree that stood outside his office.

Table 1: Who's who of the Java technology^[1]
Has worked for GT (Green Team), FP (FirstPerson) and JP (Java Products Group)

Name	GT	FP	JP	Details
Lisa Friendly		✓ Yes	✓ Yes	FirstPerson employee and member of the Java Products Group
John Gage				Science Office (Director), Sun Microsystems
James Gosling	✓ Yes	✓ Yes	✓ Yes	Lead engineer and key architect of the Java technology
Bill Joy				Co-founder and VP, Sun Microsystems; Principal designer of the UC Berkeley, version of the UNIX® OS
Jonni Kanerva		✓ Yes		Java Products Group employee, author of The Java FAQ1
Tim Lindholm		✓ Yes	✓ Yes	FirstPerson employee and member Java Products Group
Scott McNealy				Chairman, President, and CEO of Sun Microsystems
Patrick Naughton	✓ Yes	✓ Yes		Green Team member, FirstPerson co-founder
George Paolini				Corporate Marketing (Director), Sun's Java Software Division
Kim Polese		✓ Yes		FirstPerson product marketing
Lisa Poulson				Original director of public relations for Java technology (Burson-Marsteller)
Wayne Rosing		✓ Yes		FirstPerson President
Eric Schmidt				Former Sun Microsystems Chief Technology Officer
Mike Sheridan	✓ Yes			Green Team member

The demise of an idea, birth of another

By now, the work on Oak had been significant but come the year 1993, people saw the demise of set-top boxes, interactive TV and the PDAs. A failure that completely ushered the inventors' thoughts to be reinvented. Only a miracle could make the project a success now. And such a miracle awaited anticipation.

National Center for Supercomputing Applications (NCSA) had just unveiled its new commercial web browser for the internet the previous year. The focus of the team, now diverted towards where they thought the "next-wave" of computing would be – the internet. The team then divulged into the realms of creating the same embeddable technology to be used in the web browser space calling it **an applet** – *a small application*. The team now needed a proper identity and they decided on naming the new technology they created Java ushering a new generation of products for the internet boom. A by-product of the project was a cartoon named "Duke" created by Joe Parlang which became its identity then.

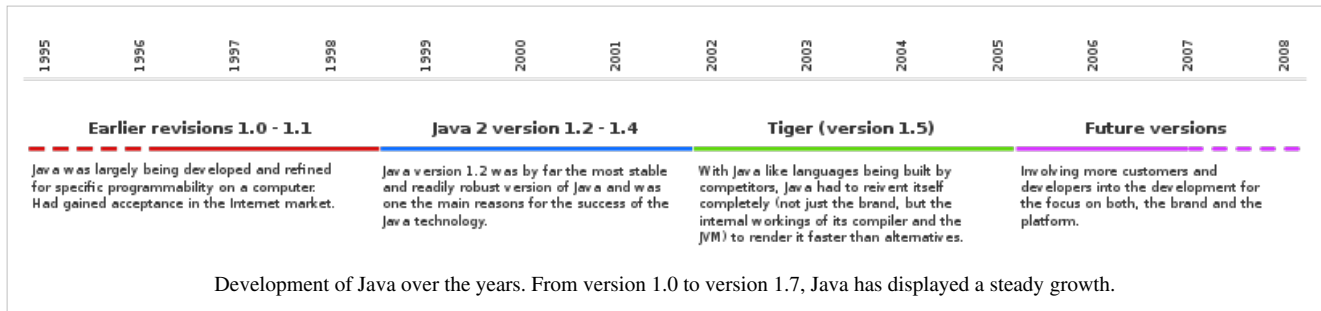
Finally at the SunWorld™ conference, Andreessen unveiled the new technology to the masses. Riding along with the explosion of interest and publicity in the Internet, Java quickly received widespread recognition and expectations grew for it to become the dominant software for browser and consumer applications.^[2]

Recent history

Initially Java was owned by Sun Microsystems, but later it was released to open source;WB:OR the term Java was a trademark of Sun Microsystems. Sun released the source code for its HotSpot Virtual Machine and compiler in November 2006, and the most of the source code of the class library in May 2007. Some parts were missing because they were owned by third parties, not by Sun Microsystems. The released parts were published under the terms of the GNU General Public License, a free software license.

Versions

Unlike C and C++, Java's growth is pretty recent. Here, we'd quickly go through the development paths that Java took with age.



Initial Release (versions 1.0 and 1.1)

Introduced in 1996 for the Solaris, Windows, Mac OS Classic and Linux, Java was initially released as the Java Development Kit 1.0 (JDK 1.0). This included the Java runtime (the virtual machine and the class libraries), and the development tools (e.g., the Javac compiler). Later, Sun also provided a runtime-only package, called the Java Runtime Environment (JRE). The first name stuck, however, so usually people refer to a particular version of Java by its JDK version (e.g., JDK 1.0).

Java 2 (version 1.2)

Introduced in 1998 as a quick fix to the former versions, version 1.2 was the start of a new beginning for Java. The JDKs of version 1.2 and later versions are often called *Java 2* as well. For example, the official name of JDK 1.4 is *The Java(TM) 2 Platform, Standard Edition version 1.4*.

Major changes include:

- Rewrite the event handling (Add Event Listeners)
- Change Thread synchronizations
- Introduction of the JIT-Just in time compilers

Java 2 Extensions (versions 1.3 [2000] to 1.4.0 [2002])

The language as such has been stable since JDK 1.0; the class libraries that come with the JDK got larger and have changed in some parts. Extensions and architectures closely tied to the Java programming language include: J2EE, J2ME, JNDI, JSML, JDBC, JAIN, JDMK, Jini, Jiro, JXTA, JavaSpaces, JMI.

Tiger (version 1.5.0; Java SE 5)

Released in September 2004

Major changes include:

- Generics - Provides compile-time type safety for collections and eliminates the drudgery of casting.
- Autoboxing/unboxing - Eliminates the drudgery of manual conversion between primitive types (such as int) and wrapper types (such as Integer).
- Enhanced for - Shorten the for loop with Collections use.
- Static imports - Lets you import all the static part of a class.
- Annotation/Metadata - Enabling tools to generate code and deployment descriptors from annotations in the source code. This leads to a "declarative" programming style where the programmer says what should be done and tools emit the code to do it. Annotations can be inspected through source parsing or by using the additional