

**Introduction**  
**1.1 What is Haxe?**

Haxe consists of a high-level, open source programming language and a compiler. It allows compilation of programs, written using an ECMAScript-oriented syntax, to multiple target languages. Employing proper abstraction, it is possible to maintain a single code-base which compiles to multiple targets.

Haxe is strongly typed but the typing system can be subverted where required. Utilizing  
type information, the Haxe type system can detect errors at compile-time which would only be noticeable at run-time in the target language. Furthermore, type information can be used by the target generators to generate optimized and robust code.  
Currently, there are nine supported target languages which allow for different use-cases:

|  |  |  |
| --- | --- | --- |
| Name | Output type | Main usages |
| Javascript | Sourcecode | Browser, Desktop, Mobile, Server |
| Neko | Bytecode | Desktop, Server |
| PHP | Sourcecode | Server |
| Python | Sourcecode | Desktop, Server |
| C++ | Sourcecode | Desktop, Mobile, Server |
| Actionscript 3 | Sourcecode | Browser, Desktop, Mobile |
| Flash | Bytecode | Browser, Desktop, Mobile |
| Java | Sourcecode | Desktop, Server |
| C# | Sourcecode | Desktop, Mobile, Server |
|  |  |  |

**Haxe** is not a high level framework. It's a toolkit that can be used to build cross-platform tools and frameworks

**Who uses Haxe?**

**Haxe** is used by thousands of developers worldwide to build games, apps, tools, and frameworks. Several high profile companies are using Haxe, such as*Nickelodeon*, *TiVo*, *Zynga* and *Prezi*.

**Open Source**

**Haxe** is an open source technology, free to use and modify. If you are looking for Support or technical partnership, the **Haxe Foundation** provides such services

**The Haxe Foundation**

After years of open source development, the **Haxe Foundation** was created to fund long term Haxe development and provide support to companies using Haxe.

The goals of the Haxe Foundation are:

* To support the whole Haxe ecosystem by funding core technologies
* To provide a single point of contact for companies that wish to evaluate Haxe as a potential solution
* To offer Paid Support Plans ensuring that somebody will always be available to answer the phone when you need help
* To help the Haxe Open Source Community by organizing events and user groups

**1.3 Hello World**

The following program prints “Hello World” after being compiled and run:

|  |
| --- |
| class HelloWorld { static public function main():Void { trace("Hello World");  }  } |

This can be tested by saving the above code to a file named HelloWorld.hx and invoking the Haxe Compiler like so: haxe -main HelloWorld --interp. It then generates the following output: HelloWorld.hx:3: Hello world. There are several things to learn from this:

*•* Haxe programs are saved in files with an extension of .hx.  
*•* The Haxe Compiler is a command-line tool which can be invoked with parameters such as -main HelloWorld and --interp.  
*•* Haxe programs have classes (HelloWorld, upper-case), which have functions (main, lower-case).

**1.4 History**

The Haxe project was started on 22 October 2005 by French developer *Nicolas Cannasse* as a successor to the popular open-source ActionScript 2 compiler *MTASC* (Motion-Twin Action Script Compiler) and the in-house *MTypes* language, which experimented with the application of type inference to an object oriented language. Nicolas’ long-time passion for programming language design and the rise of new opportunies to mix different technologies as part of his game developer work at *Motion-Twin* led to the creation of a whole new language.

Being spelled *haXe* back then, its beta version was released in February 2006 with the first supported targets being AVM-bytecode and Nicolas’ own *Neko* virtual machine.

Nicolas Cannasse, who remains leader of the Haxe project to this date, kept on designing  
Haxe with a clear vision, subsequently leading to the Haxe 1.0 release in May 2006. This first major release came with support for Javascript code generation and already had some of the features that define Haxe today such as type inference and structural sub-typing.

Haxe 1 saw several minor releases over the course of two years, adding the Flash AVM2 target along with the *haxelib*-tool in August 2006 and the Actionscript 3 target in March 2007. During these months, there was a strong focus on improving stability, which resulted in several minor bug-fix releases.  
Haxe 2.0 was released in July 2008, including the PHP target, courtesy of *Franco Ponticelli*. A similar effort by *Hugh Sanderson* lead to the addition of the C++ target in July 2009 with the Haxe 2.04 release.

Just as with Haxe 1, what followed were several months of stability releases. In January 2011 Haxe 2.07 was released with the support of *macros*. Around that time, *Bruno Garcia* joined the team as maintainer of the Javascript target, which saw vast improvements in the subsequent 2.08 and 2.09 releases.

After the release of 2.09, *Simon Krajewski* joined the team and work towards Haxe 3 began. Furthermore, *Cauˆe Waneck*’s Java and C# targets found their way into the Haxe builds. It was then decided to make one final Haxe 2 release, which happened in July 2012 with the release of Haxe 2.10.

In late 2012, the Haxe 3 switch was flipped and the Haxe Compiler team, now backed by the newly established *Haxe Foundation*, focused on this next major version. Haxe 3 was subsequently released in May 2013.