Compiler

A compiler is a software program that [compiles](https://techterms.com/definition/compile) program [source code](https://techterms.com/definition/sourcecode) files into an executable program. It is included as part of the integrated development environment [IDE](https://techterms.com/definition/ide)with most programming software packages.

The compiler takes source code files that are written in a high-level language, such as [C](https://techterms.com/definition/cplusplus), [BASIC](https://techterms.com/definition/basic), or [Java](https://techterms.com/definition/java), and compiles the code into a low-level language, such as machine code or assembly code. This code is created for a specific processor type, such as Intel Pentium or PowerPC. The program can then be recognized by the processor and run from the [operating system](https://techterms.com/definition/operating_system).

After a compiler compiles source code files into a program, the program cannot be modified. Therefore, any changes must be made in the source code and the program must be recompiled. Fortunately, most modern compilers can detect what changes were made and only need to recompile the modified files, which saves programmers a lot of time. This can help reduce programmers' 100 hour work weeks before project deadlines to around 90 or so.

Interpreter

An interpreter is a program that reads and executes code. This includes source code, pre-compiled code, and scripts. Common interpreters include Perl, Python, and Ruby interpreters, which execute Perl, Python, and Ruby code respectively.

Interpreters and compilers are similar, since they both recognize and process source code. However, a compiler does not execute the code like and interpreter does. Instead, a compiler simply converts the source code into machine code, which can be run directly by the operating system as an executable program. Interpreters bypass the compilation process and execute the code directly.

Since interpreters read and execute code in a single step, they are useful for running scripts and other small programs. Therefore, interpreters are commonly installed on Web servers, which allows developers to run executable scripts within their webpages. These scripts can be easily edited and saved without the need to recompile the code.

While interpreters offer several advantages for running small programs, interpreted languages also have some limitations. The most notable is the fact that interpreted code requires to run. Therefore, without an interpreter, the source code serves as a plain text file rather than an executable program. Additionally, programs written for an interpreter may not be able to use built-in system functions or access hardware resources like compiled programs can. Therefore, most software applications are compiled rather than interpreted.