

## Low Level vs High Level Programming

A **low-level language** is a [programming language](#) that provides little or no abstraction of programming concepts and is very close to writing actual machine [instructions](#) (very close to how the computer works). Two good examples of low-level languages are [assembly](#) and [machine code](#).

Machine language, or machine code, is the lowest level of computer languages. It contains [binary](#) code, often generated by compiling high-level source code for a specific [processor](#). Most developers never need to edit or even look at machine code. Only programmers who build software compilers and [operating systems](#) need to view machine language. (These programs are written in 1's and 0's).

Assembly language is one step closer to a high-level language than machine language. It includes commands such as MOV (move), ADD (add), and SUB (subtract). These commands perform basic operations, such as moving values into [memory](#) registers and performing calculations. Assembly language can be converted to the machine language using an [assembler](#).

Machine code (low level)	Assembly (approx. 120 commands total)	Python/Java (high level)
101001110101	add mov sub	while – this is created by making a function out of a bunch of assembly commands

Languages Low level to higher level

- C
- C++
- Java
- C#
- Perl
- Lisp
- JavaScript
- Python
- Ruby
- SQL