

Difference Between Assembly Language and High-Level Language

The difference between assembly language and high-level language is that assembly language is more efficient than high-level language. Let us check the difference between assembly language and high-level language based on various parameters.

Assembly Language vs High-Level Language	
Assembly Language	High-Level Language
For the conversion process, an assembler is needed for the assembly language.	An interpreter or compiler is necessary for the conversion process of a high-level language.
It is machine-dependent.	It is machine-independent.
This makes accessing hardware components simple.	It makes challenging to access hardware components like this.
This has a more compact code.	There is no compact code in high-level language.
It is more accurate.	It is less accurate.
Length of code is short in assembly language.	Length of code is larger in high-level language.

What is an Assembly Language?

A low-level computer language called assembly language enables users to create a variety of programmes by using alphanumeric codes rather than numeric codes to represent a set of instructions. A notable example of a modern, extensive assembly language is IBM PC DOS.

Every type of CPU requires a separate assembly language coding scheme. For the same tasks, X86 and x64 processors use different assembly language codes. The commands in assembly language are the same as those in machine language, but names are used in place of 0 and 1.

Examples of assembly languages are given below.

- ARM
- MIPS
- x86
- Z80
- 68000
- 6502
- 6510

What is a High-Level language?

A high-level language is a machine-independent language. A high-level language is one that is user-friendly, uses variables and functions, and is not dependent on the architecture of the computer. The programmer doesn't bother about the hardware integration component when writing general-purpose code. A high-level language programme needs to be translated into machine code before being processed by a computer.

Examples of high-level language are given below.

- C
- Fortran
- Basic
- Algol
- Ada
- C++
- C#
- PHP
- Perl
- Python
- Javascript
- Pharo

Key Difference Between Assembly Language and High-Level Language

The key difference between assembly language and high-level language is given below.

- Programs written in assembly language won't work on processors of a different type. Programs operate independently of processor type in high-level languages.
- The performance of assembly language is more accurate than in high-level language.
- High-level languages require additional instructions in order to execute code on a computer.
- Several assembly language codes are generated from one or two high-level language statements.
- Unlike high-level languages, assembly language allows us to read pointers at a physical address directly.
- Assembly language is more efficient than high-level language.

