

# Difference Between High Level Language and Low Level Language

[www.differencebetween.com](http://www.differencebetween.com)

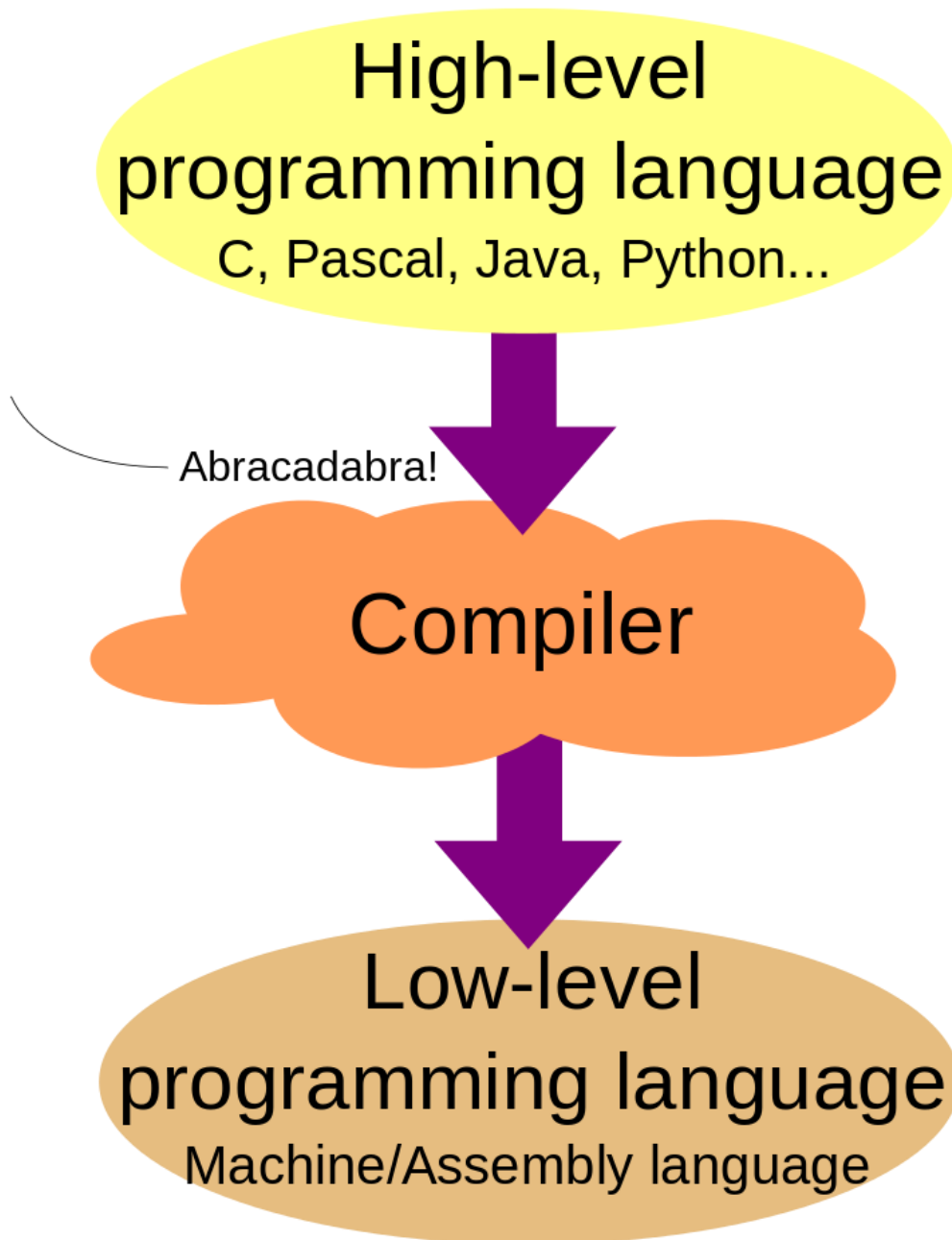
## Key Difference - High Level Language vs Low Level Language

A computer functions according to the instructions provided by the user. A set of instructions written to perform a certain task is a computer program. A collection of computer programs is known as [software](#). Computer programs or software are written using Computer programming language. There is a large number of programming languages in the world. Computer programming languages can be divided into two main categories. They are High Level Languages and Low Level Languages. The **key difference** between High Level Language and Low Level Language is that **High Level Language is a programmer friendly language that provides a high level of abstraction from the hardware whereas Low Level Language is the language that is machine friendly and provides no or less abstraction from the hardware**. High Level Languages are useful for building desktop, web and mobile [applications](#) and Low Level Languages are useful for developing a hardware-related application such as device drivers, operating systems and embedded systems.

## What is High Level Language?

High Level Language is close to human or the programmer. Some examples of High Level Languages are [Java](#), [C#](#), [Python](#). These programming languages are easy for humans to understand and allows to develop programs to perform various tasks. Each programming language has a unique set of keywords and syntax for writing programs. They are machine independent and portable.

High Level Languages have a syntax similar to the English Language so uses a [compiler or an interpreter](#) to convert the human readable program to computer readable machine code. These languages do not interact directly with hardware. Therefore, High Level Languages takes time to execute. High Level Languages are also not memory efficient. They might require specific runtime environments.



**Figure 01: High Level Languages and Low Level Languages**

There are plenty of advantages in using High Level Languages. The programmer can easily understand the language. They are programmer friendly, easy to debug and maintain. Overall, High Level Languages are useful for building various applications.

# What is Low Level Language?

A Low Level Language is a machine-friendly language. It can interact directly with registers and memory. Low Level Language does not require a compiler or an interpreter to convert the program to machine code, so the Low Language is faster than a High Level Language. Those programs are machine dependent and not portable. The most common Low Level Languages are [Machine Language and Assembly Language](#).

Machine Language is the closest language to hardware. The CPU directly executes Those instructions. A machine language consists of zeros and ones. Machine Language programs are machine dependent. Assembly language is one step ahead of Machine Language. The programmer should have a good understanding of the computer architecture and CPU to program using Assembly Language. An Assembly language program is converted to machine language using an assembler. Assembly Language has mnemonics that are low-level instructions. Some commands of Assembly language are MOV and ADD.

Overall, Low Level Languages are used to build applications that execute fast. They can also be used to develop hardware-related applications such as device drivers and operating systems. Learning Low-level programming languages is hard. It requires a good knowledge of computer architecture.

# What is the Similarity Between High Level Language and Low Level Language?

- Both give instructions to a computer to perform a specific task.

# What is the Difference Between High Level Language and Low Level Language?

High Level Language vs Low Level Language	
High Level Language is a programmer friendly language that provides a high level of abstraction from the hardware.	Low Level Language is the language that is machine friendly and provides no or less abstraction from the hardware.
Execution Speed	
A High Level Language is slower than a Low Level Language.	A Low Level Language is faster than a High Level Language.
Memory Efficiency	

A High Level Language is not memory efficient.	A Low Level Language is more memory efficient.
<b>Translation</b>	
A High Level Language requires a compiler or an interpreter to convert the program into machine code.	Assembly Language requires an assembler to convert the program to machine code while machine language is executed by the computer directly.
<b>Comprehensibility</b>	
A High Level Language is easily understandable by the programmer.	A Low Level Language is easily understandable by the computer.
<b>Machine Dependency</b>	
A High Level Language is machine independent.	A Low Level Language is machine dependent.
<b>Portability</b>	
A High Level Language can run on multiple platforms, so it is portable.	A Low Level Language is not portable.
<b>Debugging and Maintenance</b>	
A program written using a High Level Language is easy to debug and maintain.	A program written using a Low Level Language is hard to debug and maintain.
<b>Support</b>	
High Level Languages have more community support.	Low Level Languages do not have much community support.

## Summary - High Level Language vs Low Level Language

Computers perform various functionalities depending on the instructions provided by the user. These instruction sets are programs and are written using a specific programming language. A programming language is a formal constructed language designed to communicate with the computer. Programming languages can be categorized into High Level Languages and Low Level Languages. Low Level Languages are capable of

handling hardware efficiently. High Level Languages are more popular among programmers because they are easy to learn, read, debug and test. The difference between High Level Language and Low Level Language is High Level Language is a programmer friendly language which provides a high level of abstraction from the hardware while Low Level Language is the language which is machine friendly and provides no or less abstraction from the hardware.

#### **Reference:**

1. "Classification of programming languages." Codeforwin, 4 Sept. 2017. [Available here](#)
2. "Difference between Low Level and High-Level Programming languages." Difference between Low Level and High-Level Programming languages - IncludeHelp. [Available here](#)
3. Buxoo Abdullah, Educator Follow. "Computer languages." LinkedIn SlideShare, 22 Apr. 2014. [Available here](#)
4. BESOR ACADEMY Follow. "Computer languages." LinkedIn SlideShare, 20 Oct. 2015. [Available here](#)

#### **Image Courtesy:**

1. 'High level to low level diagram' By Kayau - Own work, ([CC BY-SA 3.0](#)) via [Commons Wikimedia](#)

#### **How to Cite this Article?**

APA: Difference Between High Level Language and Low Level Language.(2018 January 24). Retrieved (date), from <http://differencebetween.com/difference-between-high-level-language-and-vs-low-level-language/>

MLA: "Difference Between High Level Language and Low Level Language" Difference Between.Com. 24 January 2018. Web.

Chicago: "Difference Between High Level Language and Low Level Language." Difference Between.Com. <http://differencebetween.com/difference-between-high-level-language-and-vs-low-level-language/> accessed (accessed [date]).



Copyright © 2010-2017 Difference Between. All rights reserved

