**Evolution of Programming Language**

* Procedural programming – languages such as Pascal, Fortran, Cobol, etc
* Object Oriented Programming – OOP programming languages such as Java, Visual Basic, Python, etc
* Scripting languages are programming languages that don't require an explicit compilation step.

For example, in the normal case, you have to compile a C program before you can run it. But in the normal case, you don't have to compile a JavaScript program before you run it. So JavaScript is sometimes called a "scripting" language.

**Syntax of a programming language**

Syntax refers to the spelling and grammar of a programming language. Computers are inflexible machines that understand what you type only if you type it in the exact form that the computer expects. The expected form is called the syntax.

**Programming languages - what they have in common**

* e.g. strict syntax rules, data storage, input statements, output statements, branching, looping.
* Planning Phase – Design Phase – Testing Phase

**Programming languages - their differences**

* e.g. different syntax, different structures, different focus
* procedural versus object oriented (OOP)

Different types of programming languages <https://www.youtube.com/watch?v=qmksVfulV0o>

**Low-level programming language**

In computer science, a low-level programming language is a programming language that provides little or no abstraction from a computer's instruction set architecture—commands or functions in the language map closely to processor instructions. Generally this refers to either machine code or assembly language.

Low-level languages are useful because written in them can be crafted to run very fast and with a very small memory footprint. However, they are considered more difficult to utilize because they require a deeper knowledge of machine language.

Languages such as C and C++ are considered "lower-level" — they provide a minimal amount of abstraction at the smallest possible cost to performance and efficiency. These abstractions, such as classes, lambda functions and macros, allow programmers to use complex functionality without writing overly complex code. For this reason, lower-level languages are used in projects where abstractions are necessary to keep code highly readable and maintainable, but where maximum performance is still paramount. Many operating systems and high-frame rate computer games are a good example of this.

* Learning - Low-level languages are difficult to learn.
* 2 Understanding - Low-level languages are far from human languages.
* Execution - Programs in low-level languages are fast in execution.
* Modification - Programs in low-level languages are difficult to modify.
* Facility at hardware level - Low-level languages provide facility to write programs at hardware level.
* Knowledge of hardware Deep - Deep knowledge of hardware is required to write programs.
* Uses - These languages are normally used to write hardware programs.

**High level programming languages**

A high-level language is a computer programming language that isn't limited by the computer, designed for a specific job, and is easier to understand. It is more like human language and less like machine language. However, for a computer to understand and run a program created with a high-level language, it must be compiled into machine language.

The first high-level languages were introduced in the 1950's. Today, there are many high-level languages in use, including BASIC, C, C++, Cobol, FORTRAN, Java, Pascal, Perl, PHP, Python, Ruby, and Visual Basic.

* Learning - High-level languages are easy to learn.
* Understanding – High level languages are near to human languages.
* Execution - Programs in high-level languages are slow in execution.
* Modification - Programs in high-level languages are easy to modify.
* Facility at hardware level - High-level languages do not provide much facility at hardware level.
* Knowledge of hardware Deep - Knowledge of hardware is not required to write programs.
* Uses - These languages are normally used to write application programs.