Scripting Vs. Programming | Difference between Scripting and Programming

Most Often, people think of the terms scripting and programming as similar to each other and use them synonymously, even though they are very different from each other. However, due to the advancements in computer languages, these differences are becoming blurred and less important, but to be a good programmer, it is very important to understand the basic differences between the two.

Both scripting and programming are computer languages, with all the scripting languages being programming languages, but all programming languages are not scripting languages. The basic difference between a scripting language and a programming language is that scripting languages do not need an additional step of compilation and rather they are interpreted, whereas programming languages are compiled and hence need a compilation step to convert the high-level language to machine code.

To better understand the differences, we should understand why scripting languages are evolved? In earlier days, programming languages were built to create software and applications such as Microsoft Excel, MS Word, Internet Explorer, etc. But as time passed, the demand to upgrade the programming languages has increased, and programs needed a way to add new functionalities, and hence scripting languages come into existence.

## **What is a Programming Language?**

A [programming language](https://www.javatpoint.com/programming-language)

is a combination of words and symbols that is used to write programs, and these programs are set of instructions. Therefore, we can say, "**A programming language is a way by which programmers communicate with computers through the set of instructions known as code/program.**" Programming languages are compiled languages, which means the source code is compiled to convert it to machine code.

As we know, computers work on bits (0 and 1) and cannot understand human languages such as English; hence programming languages are implemented. Programming languages are the computer languages that are used in computers to provide instruction and implement algorithms. Each programming language contains its own set of rules for writing the code, and such rules are known as **Syntax**. Thus, to learn and write code in one programming language, we need to know its syntax. These languages enable the developers to create **desktop applications, web applications, mobile applications,** implement machine learning algorithms, and many more tasks. Some **popular programming languages are [C++](https://www.javatpoint.com/cpp-tutorial)**

**, [C](https://www.javatpoint.com/c-programming-language-tutorial)**

**, Pascal, [COBOL](https://www.javatpoint.com/cobol)**

**, [Java](https://www.javatpoint.com/java-tutorial)**

(But java is compiled and interpreted as firstly its source code is compiled into byte-code, and then interpreted at runtime).

### **Advantages**

* These are building blocks for other computer languages.
* These are well suited for large projects.

### **Applications of Programming languages**

* Programming languages are mainly used to create different software and applications such as **MS Excel, PowerPoint**, etc.
* These are used for transforming the data, for example, solving a set of equations from a set of conditions.

## **What is a Scripting Language?**

**"A scripting language is a type of programming language which does not require explicit compilation step, and it is designed for a runtime system to automate the execution of tasks**." For example, a JavaScript program is not needed to be compiled before we run it. These are also known as very high-level programming languages because of working at a high level of abstraction.

Scripting languages support "**script**," which is small program written for a specific runtime environment. These are interpreted at runtime rather than compiled. It means, to convert the source code to machine code, scripting languages use an interpreter, not the compiler. As scripting language is not compiled so as we write something meaningful, we can run it immediately.

The scripting language refers to dynamic high-level, general-purpose interpreted languages such as Python, Perl, etc. Thus, a scripting language can automate different environments such as **application softwares, webpages, text editors, operating system shells, computer games, etc**.

### **Advantages**

* It is an easy and quick process to learn coding in Scripting language, and for this, much knowledge of web technology is not needed.
* In scripting languages, a wide variety of libraries is available that enable the developers to develop new applications.
* With the help of scripting languages, we can add visualization interfaces and combinations to web pages. Most of the latest web pages need scripting languages for creating enhanced web pages, fascinating UI, and many more.
* There are less number of data structures and variable to be used, which make it highly efficient.
* These are Less code-intensive as compared to traditional programming languages.

### **Applications of Scripting Language**

* These are used to automate a specific task in a program.
* These are useful to extract information from a dataset.

## **Key differences between Programming and Scripting language**

* **Definition**

A programming language is a computer language which is used to communicate with computers using a set of instructions.

A scripting language is a type of programming language that supports scripts, which are small programs mainly used to automate the execution of a specific function in a specific runtime environment.

* **Interpretation**

Programming languages use compiler and do not require to be interpreted by another language or application; hence these languages run independently and do not depend on the parent program.

In contrast, scripting languages are interpreted within another program; for example, JavaScript has to be combined within HTML, then interpreted by the web browser.

* **Design**

Programming languages are specifically designed to facilitate the developer with complete code and software development, whereas scripting languages are specifically designed to make programming faster and simpler.

* **Development**

Development of software/Application or coding using programming languages is difficult as lots of lines of code is needed for a task. Whereas in scripting languages, coding is easier as it needs only a few lines of code to perform a task.

Therefore, ***development time in programming languages is high due to more coding, whereas development time in a scripting language is less due to less coding.***

* **Types/Categorisation**

Programming languages are categorized into mainly five categories:

1. **The first generation,**
2. **Second generation,**
3. **Third generation,**
4. **Fourth generation,**
5. **and Fifth generation languages.**

On the other hand, Scripting languages are categorized into two categories

1. **Server-side scripting languages and**
2. **client-side scripting languages.**

* **Conversion into machine code**

As programming languages use a compiler, hence the complete program is converted into machine code in one shot. Whereas Scripting languages use an interpreter, hence the program is converted into machine code line by line.

* **Speed**

The programming languages are faster in speed because of using a compiler, which usually runs faster as it finds all the errors at once after analyzing the program.

In contrast, Scripting languages are slow as they use an interpreter that analyses a program line by line. Every time it detects an error, it stops further execution until the error gets removed.

* **Examples**

Some popular examples of programming languages are **C, C++, Java, [Scala](https://www.javatpoint.com/scala-tutorial)**

**, COBOL**, etc. Some popular examples of Scripting languages are **[Perl](https://www.javatpoint.com/perl-tutorial)**

**, [Python](https://www.javatpoint.com/python-tutorial)**

**, [JavaScript](https://www.javatpoint.com/javascript-tutorial)**

**, [PHP](https://www.javatpoint.com/php-tutorial)**

**, [Ruby](https://www.javatpoint.com/ruby-tutorial)**

**, etc.**

## **Comparison table between Programming Language and Scripting Language**

|  |  |
| --- | --- |
| **Programming Language** | **Scripting Language** |
| A programming language is a computer language that is used to communicate with computers using a set of instructions. | A scripting language is a type of programming language designed for a runtime system to automate the execution of tasks. |
| It is compiled language or compiler-based language. | It is interpreted language or interpreter-based language |
| It is used to develop an application or software from scratch. | It is used to combine existing components and automate a specific task. |
| It runs or executes independently and does not depend on the parent (exterior) program. | It runs or executes inside another program. |
| It uses a compiler to convert source code into machine code. | It uses an interpreter to convert source code into machine code. |
| As it uses a compiler, hence the complete program is converted into machine code in one shot. | As it uses an interpreter, hence the program is converted into machine code line by line. |
| These languages are required to be compiled. | There is no need for compilation. |
| It is comparatively difficult to write code in a programming language, and it requires numerous lines of code for each task. | It is comparatively easy to write code in the scripting language, and it requires few lines of code for each task. |
| The development time in programming languages is high as more lines are required. | The development time in a scripting language as a smaller number of lines are required. |
| There is the high maintenance cost. | There is less maintenance cost. |
| All programming languages are not scripting languages | All scripting languages are programming languages |
| It generates a .exe file. | It does not create a .exe file. |
| Usually, programming languages do not support or provide very little support for user interface designing, data types, and graphic designing. | Scripting languages provide great support to user interface design, data types, and graphic design. |
| Some popular examples are C, C++, Java, Scala, COBOL, etc. | Some popular examples are Perl, Python, JavaScript, etc. |

## **Conclusion**

As per the above discussion, we can conclude that the primary differentiating factor between programming and scripting is the **execution process and environment**. We can also say that scripting languages extend the existing program (written in a programming language). Most applications or software are made up of programming languages but use scripting languages to define functionality. For example, an MS excel is written in Visual C++, but we need to use Visual Basic to add functionalities to it. However, the differences between both are getting negligible as advancements and improved features exist in each computer language.