

Scripting Languages

Scripting languages allow a coding professional to create scripts that execute tasks. Often, this is a useful method for automating tasks that don't require human interaction or interpretation so that you can reduce the workload of your staff. If you're interested in a career as a coding professional, it can be beneficial to learn about common and practical scripting languages.

In this article, we discuss what a scripting language is, share the difference between scripting and programming languages, list 14 of the top scripting languages and offer tips for learning how to script.

What is a scripting language?

A scripting language is a coding language that offers a method of creating commands that do not require compilation. Instead, the scripting language runs through an interpreter that translates the script into actions. Scripting languages often facilitate automation within an organization to increase efficiency. Information that anyone enters in a scripting language executes sequentially from the top of the code to the bottom, following any protocols of the scripting language.

Scripting languages can fall into two broad categories:

- **Client-side:** Client-side scripting languages focus on the parts of a website or web application that the end user (or client) interacts with directly.

- **Server-side:** Server-side scripting languages interact directly with and run on the server. These scripts are typically invisible to the end user.

What's the difference between scripting and programming languages?

If your career involves coding, it's important to understand how scripting languages and programming languages differ. Although some use the two terms interchangeably, there are important differences between the two, including:

- **Compiled versus interpreted:** One key difference between scripting and programming languages is whether they're compiled or interpreted. Programming languages are compiled, allowing the machine that's accessing them to translate them directly with no intermediary interpreter.
- **Execution speed:** Programming languages often provide faster execution speeds for task completion than a scripting language. This occurs because of the benefits of executing a compiled program and not relying on an interpreter.
- **Purposes:** The purposes of the two types of languages can overlap, but there are also differences. Generally, professionals use programming languages for a variety of applications, while they use scripting languages mainly for web application development.
- **Simplicity:** Although scripting allows a coder to complete tasks in many situations, professionals often use it for simpler tasks. This can help make it easier to learn than the average programming language.
- **Amount of code:** The simplicity of scripting languages can be beneficial when seeking to create shorter pieces of code. Completed scripts are often

shorter than completed programs, allowing a coder to complete them more quickly.

- **Portability:** Because scripting languages rely on an interpreter and do not need to be compiled prior to execution, they are often transferable across multiple operating systems. Programming languages, conversely, often require specific compilations for each operating system.

14 scripting languages (just learn python+c#+java)

If you're interested in a career in coding, learning about these languages may help:

1. JavaScript

JavaScript is one of the most common scripting languages. JavaScript is a high-level, text-based scripting language that can operate on either the client side or the server side. Web developers can use JavaScript, which works with HTML and CSS, to add interactive elements to websites.

2. PHP

PHP is another popular scripting language. PHP stands for Hypertext Preprocessor, which is a general-purpose, open-source scripting language. While PHP has become less prominent in web design as new developers have created new languages, many developers still use it because of its broad application and strong security.

3. Python

Another widely used scripting language is Python, which professionals can use for both scripting and programming. Because of its simple syntax, Python is known for being relatively easy to learn and understand, making it ideal for beginners. Some applications of Python include artificial intelligence, web development, mobile application development and operating systems.

4. Perl

Perl is a general-purpose, back-end scripting language that professionals frequently use to process text files, and they can also use it for web development and database management. Perl shares many structural similarities to programming in C languages. This can make it an excellent option for transitioning to scripting languages.

5. Ruby

Ruby is an object-oriented language known for its simple syntax. Developers can use Ruby to create web applications, and the language is also useful for data analysis and other purposes. It is a popular scripting language in the professional world, making it a valuable option for an aspiring code professional to learn.

6. Bash

Computer programmers can also create scripts with Bash, which is a Unix command language. One of the most common uses of Bash is accessing files and completing tasks through the command line. Because the syntax of Bash is simple and intuitive, this scripting language is fairly easy to learn.

7. R

R is a scripting language that's especially popular in statistics and data analysis. You can also use R to graph data, making this language popular with data scientists, statisticians and other professionals who work with data. The R language works in the R environment as well as in other development environments and platforms.

8. Lua

Lua is an embeddable scripting language. It has a diverse range of programming methods and applications, making it a versatile scripting language to learn. Lua is a popular scripting choice for use in the development of game engines, which allow video game developers to create systems in which they can design their games.

9. Emacs Lisp

Emacs Lisp is a scripting language originally designed for use with eMac computers. Although eMac computers are no longer in production, the language can still be a valuable learning aid because of its relationship with other scripting languages. It is particularly relevant to coders with an interest in working on Unix machines and with the command line.

10. Groovy

Groovy is a scripting language with a syntax similar to Java. Greedy makes use of a dot-separated notation and can carry out complex tasks when needed. Professionals often use Groovy for web development projects, and it can complete a range of tasks within a web design.

11. PowerShell

PowerShell is a command-line scripting language that can work with many platforms. Although you may use it for other tasks, PowerShell primarily serves to automate computer tasks. This frees up time for yourself or others in the organization to work on other projects.

12. VBA

VBA is a domain-specific scripting language and stands for Visual Basic for Applications. Professionals primarily use it as a scripting addition to Microsoft applications. For example, you may create a macro in Excel that completes multiple tasks on a single button press, with VBA providing the scripting to execute the actions.

13. GML

GML is a scripting language for the Game Maker Studios line of development software. Game Maker Studio is an engine for creating computer games that combine both visual layout and scripting. Although developers do not require scripting skills to use Game Maker Studio due to the inclusion of plug-and-play buttons, learning to script in GML significantly expands your options when creating a game in Game Maker Studios.

14. VBScript

VBScript is a scripting language based on the Visual Basic suite of computer programs. VBScript can create code for both online and client-side implementation. Although VBScript has decreased in prominence in recent years, it is a foundational scripting language for some developers.

Tips for learning scripting languages

If you're interested in learning a scripting language, these tips can help you do so effectively:

- **Think about your goals.** Consider what your intentions are as a coding professional when choosing which scripting languages to learn. Adapting your language studies to match those which align with your objectives can make your study more effective.
 - **Use online guides.** When learning to code, online resources and courses can be an excellent place to begin. Often, these courses are available for free and provide a comprehensive understanding of the language and potentially a certification or credentials as well.
 - **Consider a formal education.** Although a degree in computer science is not a requirement when seeking to work as a coding professional, it may still be beneficial. Earning a degree may be useful when applying to coding positions, and it allows you to gain a high-quality education from a college or university.
 - **Practice your scripting skills.** Practical learning is an effective way to learn a new scripting language for most coding professionals. By writing practice programs, you learn how to apply your scripting knowledge, and the process of troubleshooting can help you further strengthen your understanding of the language and how it works.
 - **Search when you are stuck.** If you encounter a problem during your scripting and you do not know the solution, online resources may help. Most scripting languages have active communities with a simple search for your problem, providing you with links to multiple articles or forum posts discussing the issue which you may apply to fix your script.
-

Revision #3

Created 2023-05-29 13:19:25 UTC by naruzkurai

Updated 2023-05-29 14:13:00 UTC by naruzkurai