Server side PHP, Python, Node JS, Perl, Ruby

Types of Scripting Languages

client side HTML, JS, CSS

There are two main types of scripting languages: server-side and client-side.

They differ on where the code is run from, which affects not only the actual languages chosen but also the performance and their capabilities.

Server-side scripting language

The term server-side scripting language refers to those that run off a web server. Since it performs from the back-end side, the script is not visible to the visitor. Because of that, it is a more secure approach.

They are often used to create dynamic websites and platforms, handle user queries, and generate and provide data and others. A famous example of server-side scripting is the use of PHP in WordPress.

Examples: PHP, Python, Node.js, Perl, and Ruby.

Client-side scripting language

Unlike the above, client-side scr ipting languages run off the user's browser.

It is usually performed at the front-end, which makes it visible to visitors and makes it less vulnerable to exploits and leaks. As such, it is often used to build user interfaces and lighter functionality such as that.

Since it runs locally, they usually provide better performance and, therefore, do not strain your server.

Examples: HTML, CSS, jQuery, and JavaScript.

5 Scripting Languages Examples

Now that you know the types, let's take a closer look at some of the different scripting languages out there to choose from. Each has its own unique advantages and potential uses.

1. JavaScript

JavaScript website

Also sometimes abbreviated as simply "JS", JavaScript is probably the best-known scripting language, as it's a pillar of the web as we know it (right along with HTML and CSS).

In fact, about 98 percent of websites currently on the web use JavaScript.

2. Python

Python website

After JavaScript, Python is easily the most popular, best-known scripting language in use today.

Programmers love it for its sheer ease of use and concise syntax systems, as they can create code significantly more quickly and with less actual typing involved.

Python is also free and open-source, making it a highly accessible scripting language. Features supported by Python include functional programming paradigms, object-oriented programming, and more.

3. PHP

PHP website

PHP is an open-source scripting language commonly used by backend web developers.

The name began as an acronym for “personal home page” — a callback to PHP’s origins as a way to make static HTML pages more functional and dynamic.

However, modern PHP is very much its own standalone scripting language.

PHP features object-oriented programming options and can be easily embedded into HTML documents of all types. Input is also fairly loose and easy, meaning programmers don’t need to declare variable data types.

4. Ruby

Ruby website (scripting languages)

Ruby’s claim to fame is its flexibility, making it a favorite among web developers of all types.

Among other things, it takes so much of the guesswork out of creating truly innovative software. It’s also incredibly easy to learn, thanks to its clean syntax, making Ruby an especially popular pick for beginning coders.

Ruby is a strictly object-oriented scripting language, so everything becomes an object when working with it. This is even the case for factors like integers or Booleans that are usually more primitive.

5. Perl

Perl website (scripting languages)

Perl is a general scripting language that’s been around a very long time — since December of 1987, to be exact.

It started out as a UNIX language primarily used to process reports. (Its name even originates from the phrase “practical extraction and reporting language”.)

Perl really began to gain traction throughout the 1990s when coders used it heavily for CGI (common gateway interface), a specification most often seen today on legacy websites.

However, it remains fairly popular because of its innovation and suitability for text manipulation tasks.

Applications of Scripting Languages:

1. Task automation locally or remotely

As with any other kind of programming language, scripting helps you automate repetitive tasks based on patterns.

You can use languages like Python to optimize long processes within the applications or platforms you or your business already use.

2. Beginner-friendly short scripts

The use of scripting languages does not require the same time commitment of more complex programming languages.

As such, they are beginner-friendly, especially when dealing with one such as Python. Despite this, it is able to power collaboration platforms such as Instagram, Pinterest and even Google.

3. Create and manage dynamic content

Scripting languages help make the web more dynamic than its traditional static pages.

This is the case of displaying different data depending on the decisions made by the visitor, thus customizing the experience. An example of this is WordPress, which uses PHP embedded with HTML code to serve content hosted in its databases.

4. Back-end programming for complex systems

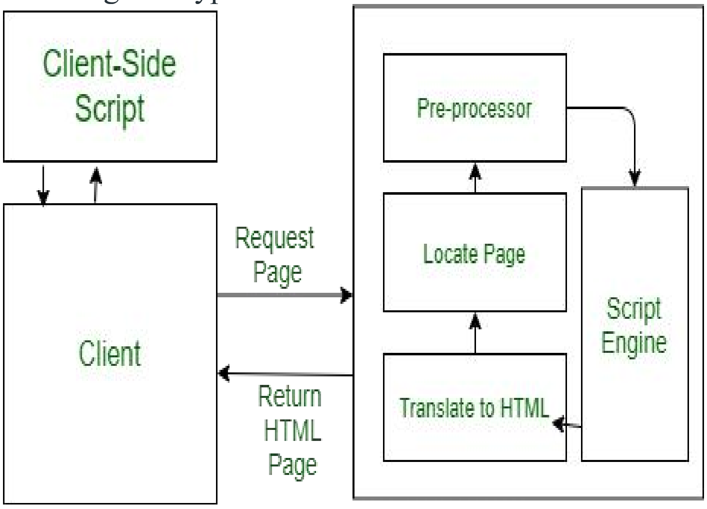
Most modern web applications such as YouTube, Gmail, and social network sites run on a series of scripting languages in the back-end. Among those is JavaScript, which is the most popular language across websites according to W3Techs.

The process of creating and embedding scripts in a web page is known as **web-scripting**. A script or a computer-script is a list of commands that are embedded in a web-page normally and are interpreted and executed by a certain program or scripting engine.

- Scripts may be written for a variety of purposes such as for automating processes on a local-computer or to generate web pages.
- The programming languages in which scripts are written are called scripting language, there are many scripting languages available today.
- Common scripting languages are VBScript, JavaScript, ASP, PHP, PERL, JSP etc.

Types of Script :

Scripts are broadly of following two type :



Web Scripting

Client-Side Scripts :

1. Client-side scripting is responsible for interaction within a web page. The client-side scripts are firstly downloaded at the client-end and then interpreted and executed by the browser (default browser of the system).

2. The client-side scripting is browser-dependent. i.e., the client-side browser must be scripting enables in order to run scripts
3. Client-side scripting is used when the client-side interaction is used. Some example uses of client-side scripting may be :
    - To get the data from user’s screen or browser.
    - For playing online games.
    - Customizing the display of page in browser without reloading or reopening the page.
4. Here are some popular client-side scripting languages VBScript, JavaScript, Hypertext Processor(PHP).

Server-Side Scripts :

1. Server-side scripting is responsible for the completion or carrying out a task at the server-end and then sending the result to the client-end.
2. In server-side script, it doesn’t matter which browser is being used at client-end, because the server does all the work.
3. Server-side scripting is mainly used when the information is sent to a server and to be processed at the server-end. Some sample uses of server-scripting can be :
    - Password Protection.
    - Browser Customization (sending information as per the requirements of client-end browser)
    - Form Processing
    - Building/Creating and displaying pages created from a database.
    - Dynamically editing changing or adding content to a web-page.
4. Here are some popular server-side scripting languages PHP, Perl, ASP (Active Server Pages), JSP (Java Server Pages).