Server-Side scripting lets web developers generate dynamic contents per client request.

**Server-side script basics**

* Runs on a server, embedded in the site’s code
* Designed to interact with back-end permanent storage, like databases, and process information from the server to access the database – like a direct line from user to database.
* Facilitates the transfer of data from server to browser, bringing pages to life in the browser, processing and then delivering a field that a user request or submits in a form
* Runs on-call. When a webpage is called up, or when parts of pages are posted back to the server with AJAX. Server-side scripts process and return dta
* Power functions in dynamic web applications, such as user validation, saving and retrieving data, and navigating between other pages.
* Plays a big role in how a database is built from the ground up and managed afterwards—an example of how roles often overlap in all aspects of development
* Build application programming interfaces (APIs), which control what data and software a site shares with other apps

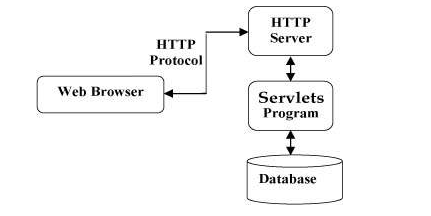
Some Server-side scripting languages

1. Java via JavaServlets
2. PhP
3. Node.js
4. ASP.Net
5. Python
6. Ruby on Rails

Server-side scripting using Java

* Servlets provide a component-based, platform-independent method for building Web-based applications.
* Java Servlets are programs that run on a Web or Application server and act as a middle layer between a request coming from a Web browser or other HTTP client and databases or applications on the HTTP server.

Servlets Architecture



Servlets Task

* Read the explicit data sent by the clients (browsers). This includes an HTML form on a Web page or it could also come from an applet or a custom HTTP client.
* Read implicit HTTP request data sent by the clients (browsers). This includes cookies, media type and compression schemes the browser understands and so forth.
* Process the data and generate the results. This process may require talking to a database, executing an RMI or CORBA call, invoking a Web service, or computing the response directly.
* Send the explicit data to the clients (browsers). This document can be sent in a variety of formats.
* Send the implicit HTP response to the clients (browsers). This includes telling the browsers or other clients what type of document is being returned setting cookies and caching parameters, and other such tasks.

<https://www.tutorialspoint.com/servlets/servlets_overview.htm>

Server side scripting using PHP

The most popular server-side language on the web, PHP is designed to pull and edit information in the database. It’s most commonly bundled with databases written in the SQL language. PHP was designed strictly for the web and remains one of the most widely used languages around.

https://www.upwork.com/hiring/development/server-side-scripting-back-end-web-development-technology/

Server-side scripting is the most traditional and main target field of PHP. You need three things to make this work: the PHP parser, a web server and a web browser.

<http://php.net/manual/en/intro-whatcando.php>

Superglobals are built-in variables that are always available in all scopes

The PHP superglobal variables:

* $GLOBALS
* $\_SERVER
* $\_REQUEST
* $\_POST
* $\_GET
* $\_FILES
* $\_ENV
* $\_COOKIE
* $\_SESSION

What PHP can do?

* PHP can generate dynamic page content
* PHP can create, open, read, write, delete, and close files on the server
* PHP can collect form data
* PHP can send and receive cookies
* PHP can add, delete, modify data in your database
* PHP can be used to control user-access
* PHP can encrypt data

**Node.js** is an [open-source](https://en.wikipedia.org/wiki/Open-source_software), [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) [JavaScript](https://en.wikipedia.org/wiki/JavaScript" \o "JavaScript)[run-time environment](https://en.wikipedia.org/wiki/Runtime_system) for executing JavaScript code [server-side](https://en.wikipedia.org/wiki/Server-side). Historically, JavaScript was used primarily for [client-side scripting](https://en.wikipedia.org/wiki/Client-side_scripting), in which scripts written in JavaScript are embedded in a webpage's HTML, to be run client-side by a JavaScript engine in the user's web browser. Node.js enables JavaScript to be used for [server-side scripting](https://en.wikipedia.org/wiki/Server-side_scripting), and runs scripts server-side to produce [dynamic web page](https://en.wikipedia.org/wiki/Dynamic_web_page) content *before* the page is sent to the user's web browser. Consequently, Node.js has become one of the foundational elements of the "JavaScript everywhere" paradigm,[[4]](https://en.wikipedia.org/wiki/Node.js" \l "cite_note-4) allowing [web application](https://en.wikipedia.org/wiki/Web_application) development to unify around a single programming language, rather than rely on a different language for writing server side scripts.

Node.js allows the creation of [Web servers](https://en.wikipedia.org/wiki/Web_server) and networking tools using [JavaScript](https://en.wikipedia.org/wiki/JavaScript) and a collection of "modules" that handle various core functionality.[[27]](https://en.wikipedia.org/wiki/Node.js#cite_note-b3-27)[[30]](https://en.wikipedia.org/wiki/Node.js#cite_note-b1-30)[[44]](https://en.wikipedia.org/wiki/Node.js#cite_note-b2-44)[[45]](https://en.wikipedia.org/wiki/Node.js#cite_note-b4-45)[[46]](https://en.wikipedia.org/wiki/Node.js#cite_note-b5-46) Modules are provided for [file system](https://en.wikipedia.org/wiki/File_system) I/O, networking ([DNS](https://en.wikipedia.org/wiki/Domain_Name_System), [HTTP](https://en.wikipedia.org/wiki/HTTP), [TCP](https://en.wikipedia.org/wiki/Transmission_Control_Protocol), [TLS/SSL](https://en.wikipedia.org/wiki/Transport_Layer_Security), or [UDP](https://en.wikipedia.org/wiki/User_Datagram_Protocol)), [binary](https://en.wikipedia.org/wiki/Binary_file) data (buffers), [cryptography](https://en.wikipedia.org/wiki/Cryptography) functions, [data streams](https://en.wikipedia.org/wiki/Stream_(computing)) and other core functions.[[30]](https://en.wikipedia.org/wiki/Node.js#cite_note-b1-30)[[45]](https://en.wikipedia.org/wiki/Node.js#cite_note-b4-45)[[47]](https://en.wikipedia.org/wiki/Node.js#cite_note-47)Node.js's modules use an API designed to reduce the complexity of writing server applications.[[30]](https://en.wikipedia.org/wiki/Node.js#cite_note-b1-30)[[45]](https://en.wikipedia.org/wiki/Node.js#cite_note-b4-45)

https://en.wikipedia.org/wiki/Node.js