**Server-Side Web Scripting**

- Server-side scripting serves as an interface to limit client access to resources on the web server.

- Server-Side Scripting concerns itself more on the back-end side of Web development or any occurrences inside the server. SSWS is used in creating dynamic websites, any scripting or programming that runs on a server is referred to as server-side scripting.

- Earliest forms of SSWS was done through CGI or Common Gateway Interface and it was a method of running scripts from programming languages like C+ or Perl on websites. This was soon followed by multitudes of programming languages and frameworks like PHP, Java, .NET, ColdFusion, Node.js and others have given developers more options to use for development of Dynamic Websites.

- The inner workings of SSWS between these scripting languages although varied is mostly similar in process, in server-side scripting if a user wishes to access contents from the server, the SSWS makes this possible by making a request from the server, the webserver will first process the script before the page is served to the browser. Processing the script can involve either pulling information from a database, making simple calculations or choosing content to display on the page, once the script has been processed the resulting content is then returned to the browser and then it is rendered. Since the scripts run on the web server the scripts will remain hidden from the end-user providing a layer of security for both the data and the source code itself.

**Server-Side Scripting Disadvantages**

* Increase of Server load.
  + Since most of the processing Is done on the server it places the burden of running the application on the webserver rather than on the client machine.
* Displaying of new content
  + Processing of data is done through page refreshes to display new content.

**Passing Instructions to the Web Server**

1. HTTP Get
   * Query strings are sent on the URL of a GET request
   * Request can be cached
   * Request can be bookmarked
   * Is not to be used handling sensitive data
   * Length restrictions
   * Should only be used in data retrieval
2. HTTP Post
   * Query strings are sent in the HTTP message body
   * Never cached
   * Cannot be bookmarked
   * No restrictions or data limit
   * On post back, it should be verified if the request is a post back, followed by taking the user’s data that was sent over, store it in a database and return a validation to the user validating whether the data sent was correct.

Server-Side Languages:

* Active Server Pages(.ASP) and ASP.NET (.ASPX)
* Perl and Common Gateway Interface (. CGI, IPL, .PL)
* Personal Home Page Tools PHP (. PHP)
* Java Server Pages (. JSP)
* Ruby (. RB, .RBW) or Python (. PY)
* Cold Fusion (. CFM)