Unit-6
Introduction to Server Side and Client-Side Scripting

(Marks: 5)

CTEVT Diploma in Computer Engineering
Subject : Web Technology I
Prepared By: Er. Abinash Adhikari

Shree Sarwajanik Secondary Technical School

What is Server-side Scripting?

Server-side scripting is a programming technique for creating code that may run software on the server side. In other words, server-side scripting is any scripting method that may operate on a web server. At the server end, actions such as website customization, dynamic changes in website content, response creation to user requests, database access, and many more are carried out.

Server-side scripting creates a communication channel between a server and a client. Previously, CGI (Common Gateway Interface) scripts were used to implement server-side scripting, and CGI was created to execute scripts written in computer languages such as C++ or Perl on websites.

The server-side is made up of three parts: the database, the server, the APIs, and the backend web software written in the server-side scripting language. When a browser requests a page with server-side scripting, the web server evaluates the script before delivering the page to the browser. In this case, script processing may entail collecting information from a database, performing simple computations, or selecting the relevant material to be shown on the client end. The output is provided to the web browser when the script is processed. The web server hides the scripts from the end user until the content is delivered, making the data and source code safer.

There are various server-side scripting languages. Some main server-side scripting languages are as follows: Python,PHP, Ruby

Features of Server-side Scripting

Server-side scripting is a powerful technique used in web development to create dynamic and interactive web applications. Below are the key features of server-side scripting:

1. **Dynamic Content Generation**
   - ➢ Enables the creation of dynamic web pages by processing requests on the server and generating HTML, CSS, or other content based on user inputs, database queries, or logic.
2. **Database Integration**
   - ➢ Server-side scripts can interact with databases to retrieve, modify, or store data, making them essential for applications like e-commerce, blogs, and user management systems.
3. **Enhanced Security**
   - ➢ Server-side code is executed on the server, hidden from the client, which makes it harder to tamper with sensitive operations like authentication, data processing, and encryption.
4. **Session Management**
   - ➢ Supports maintaining user state and session data across requests, which is critical for personalized user experiences like shopping carts or user-specific dashboards.
5. **File Handling**
   - ➢ Facilitates operations like uploading, downloading, reading, and writing files on the server, enabling features like file storage and management systems.

6. **Scalability**
   - ➢ Allows for handling multiple user requests simultaneously, making it suitable for large-scale applications with high traffic.
7. **Integration with APIs**
   - ➢ Enables the server to act as a middleman between the client and external APIs, providing processed and customized data to the client.
8. **Platform Independence**
   - ➢ Most server-side scripting languages (e.g., PHP, Python, Node.js) are platform-independent and can run on different operating systems with minimal adjustments.
9. **Customization and Personalization**
   - ➢ Tailors the output based on user preferences, roles, or behaviors, improving the user experience.
10. **Error Handling**
    - ➢ Handles errors robustly on the server, providing meaningful error messages or fallback mechanisms without exposing vulnerabilities to the client.
11. **Reduced Client-Side Load**
    - ➢ Offloads complex computations and operations to the server, reducing the load on the client-side and enhancing performance on less powerful devices.
12. **Cross-Browser Compatibility**
    - ➢ Since the processing occurs on the server, the output is sent as standard HTML or other formats that are compatible with any browser.

Examples of Server-Side Scripting Languages

- **Python**
  - It is an open-source language that is very powerful and easy to learn. It is suitable for beginners because it is simple to learn and read. It is believed to be used by Google and YouTube. It is a OOPs language with dynamic typing and data structures. It has grown to be one of the most popular languages for both quick application development and web development.
- **PHP**
  - It is an open-source server-side scripting programming language mainly designed for web apps and is the most utilized scripting language. It allows you to retrieve and manipulate data from a database and is utilized along with SQL to query the database. It is a fast and simple language to learn and develop, and Facebook, Wikipedia, and WordPress utilize it.

- **Ruby**
  - It is a free and open-source programming language that was developed and firstly introduced in the early 1990s. It is a dynamic language that is simple to read and write and an OOPs language that is interpreted as it runs. It has evolved continuously since its development and is one of the most utilized web development languages.
- **Node.js**
  - Node.js is an open-source, cross-platform JavaScript runtime environment built on Chrome's V8 JavaScript engine. It enables developers to use JavaScript for server-side scripting, making it possible to build scalable and high-performance web applications. Node.js is asynchronous and event-driven, which makes it ideal for handling concurrent requests efficiently without blocking the server. It has a rich ecosystem of modules available through npm (Node Package Manager), simplifying the development of various functionalities. Companies like Netflix, LinkedIn, and PayPal extensively use Node.js for its speed and scalability.
- **ASP.NET**
  - ASP.NET is an open-source web framework developed by Microsoft for building modern web applications and services. It is a part of the .NET platform and allows developers to use multiple programming languages, including C# and VB.NET, for server-side development. ASP.NET supports Object-Oriented Programming (OOP) principles and is well-known for its robustness, security features, and ability to build enterprise-grade applications. It provides tools for rapid application development, seamless database integration, and a powerful library for handling web development needs. Popular platforms like Stack Overflow and Microsoft’s own services use ASP.NET for their web development needs.

What is Client-side Scripting?

Client-side scripting generates code that may be executed on the client end without needing server-side processing. These scripts are typically embedded into HTML text. Client-side scripting may be utilized to check the user's form for problems before submitting it and to change the content based on the user input. The web needs three components to function: client, database, and server.

The client-side scripting may significantly reduce server demand. It is intended to be utilized as a scripting language with a web browser as the host program. The HTML and CSS are delivered as plain text when a user uses a browser to request a webpage from the server, and the browser understands and renders the web content at the client end.

Features of Client-side Scripting

Client-side scripting is a critical aspect of web development, enabling interactivity and responsiveness directly in the user's browser. Below are the primary features of client-side scripting:

1. **Executes in the Browser**
   - Client-side scripts are downloaded and executed in the user's web browser, enabling fast responses to user actions without requiring communication with the server.
2. **Interactivity and Dynamic Behavior**
   - Enhances user experience by enabling interactive elements like form validation, dynamic content updates, sliders, dropdown menus, and more without reloading the page.
3. **Asynchronous Communication**
   - Works with technologies like AJAX to fetch or send data to the server asynchronously, allowing for smoother user experiences (e.g., loading new data without refreshing the page).
4. **Real-time User Feedback**
   - Provides instant feedback to users for actions such as form validation (e.g., checking if an email format is correct or if a password is strong).
5. **Improved Performance**
   - Reduces server load by handling some operations on the client-side, such as animations, calculations, and user input validations.

6. **Interactive User Interfaces**
   - Enables the creation of rich user interfaces with interactive elements, such as modal windows, drag-and-drop features, and dynamic animations.

7. **Accessibility of Source Code**
   - Scripts are visible in the browser's developer tools, making them accessible for debugging, learning, or even tampering (a security concern).

8. **Quick Rendering of Content**
   - Allows immediate manipulation of the Document Object Model (DOM), enabling real-time updates to web pages without reloading.

9. **Reduces Server Load**
   - Offloads tasks like form validation or user interface management to the client, allowing the server to focus on critical operations.

10. **Integration with Third-party Libraries**
    - Client-side scripting often leverages libraries and frameworks like jQuery, React, Angular, or Vue.js to simplify development and add advanced features.

11. **Event-driven Programming**
    - Listens to user actions (e.g., clicks, mouse movements, keystrokes) and responds dynamically, enabling real-time interactions.

12. **Security Considerations**
    - Limited in scope to avoid exposing sensitive data or operations, as scripts can be inspected and altered by users.

13. **Offline Functionality**
    - Works with technologies like Service Workers and Web Storage (localStorage/sessionStorage) to enable offline access and caching of resources.

Examples of Client-side Scripting Languages

There are various client-side scripting languages. Some main client-side scripting languages are as follows:

- **HTML**
  - It is not a scripting language; it is a markup language. However, it serves as the basic language for client-side web development, also referred to as front-end. The presence of hypertext on a page denotes its hyperlinks. The markup language uses tags to define the structure and layout. It is a programming language that is mainly used to design a web page's structure and layout.
- **CSS**
  - CSS is an abbreviation for Cascading Style Sheets. It provides a technique for creating graphic elements that help a web application's appearance look more appealing. A style tag in a web page defines all the specifics regarding the web page's presentation, including its border styles, image styles, colour, font styles, borders, format, font size, margins, padding, etc.
- **JavaScript**
  - It is a client-side scripting language designed for a specific purpose, but several JavaScript frameworks are already utilized as server-side scripting technologies.
- **VBScript**
  - VBScript is based on Visual Basic, which was created by Microsoft in 1996. It is a scripting programming language that is lightweight, fast, and easy to learn. It is not a OOPs language but is similar to JavaScript.

Server-side Scripting vs Client-side Scripting

Feature	Server-side Scripting	Client-side Scripting
Primary Function	The main function of this scripting is to manipulate and grant access to the requested database.	The main purpose of this scripting is to give the requested output to the end-user.
Uses	It is employed at the backend, where the source code is invisible or concealed on the client side.	It is utilized at the front end, which users may view through the browser.
Processing	It needs server interaction.	It doesn't need any server interaction.
Security	It is more secure while working on a web app.	It is less secure than server-side scripting due to the code accessibility offered to the client.
Running	It executes on the web server.	It executes on the remote computer system.
Dependability	It doesn't depend on the client.	It depends on the user's browser version.

Feature	Server-side Scripting	Client-side Scripting
File Access	It offers complete access to the file that is stored in the web database server.	It doesn't offer any access to the files on the web servers.
Code Allowance	It enables the backend developer to hide the source code from the user.	The user is given access to the written code after confirming their requirements.
Occurrence	It only responds after the user begins the browsing request.	It happens when the browser processes all of the codes and then acts according to the client's needs.
Affect	It may reduce the server load.	It may effectively customize web pages and offer dynamic websites.
Languages Involved	The server-side scripting programming languages, such as PHP, ColdFusion, Python, ASP.net, Java, C++, Ruby, C#, etc.	Its programming languages are HTML, CSS, and JavaScript.