BIYANI
GROUP OF COLLEGES

THE COMPLETE HTML

Concept by :
Dr. Sanjay Biyani
Director (Acad.) Biyani Group of Colleges
www.sanjaybiyani.com

Written by :
Mr. Rahul Agarwal
Asso. Prof.
Biyani Girls College

The Internet: Basics and Early Technologies

The Internet is a global network of interconnected computers and servers that communicate using standardized protocols. It has revolutionized how we access information, communicate, and transfer data. Here are the foundational technologies that helped shape the modern Internet.

1. File Transfer (FTP - File Transfer Protocol)

FTP is a protocol for transferring files over a TCP/IP network.

Purpose: It allows users to upload, download, and manage files on remote servers.

Modes: Active Mode: The client opens a connection to the server to initiate the file transfer.

Passive Mode: The server opens a connection, and the client connects to it.

Ports: FTP usually operates on ports 21 (for commands) and 20 (for data transfer).

2. Telnet

Telnet is an old Internet protocol that allows users to remotely access and manage devices over a text-based interface.

Usage: It was widely used for remote system administration, accessing remote files, and running commands on a server.

Security: Telnet transmits data in plain text, which makes it insecure. It has been largely replaced by SSH (Secure Shell), which provides encryption and better security.

3. Usenet

Usenet is a decentralized network of discussion groups or newsgroups where users can post messages and participate in discussions.

Structure: Usenet was organized into newsgroups, which were divided into categories like technology, news, and entertainment.

Decline: It was a precursor to modern-day forums, chat groups, and social media but declined as the web became more popular.

4. Gopher

Gopher was an early Internet protocol and system that allowed users to search for and retrieve text-based documents stored on remote servers.

Functionality: It used a hierarchical, menu-driven interface, similar to a file system.

Decline: It was eventually replaced by the World Wide Web, which provided more multimedia and interactive features.

5. WAIS (Wide Area Information Server)

WAIS was an early system and protocol for searching and retrieving information from distributed databases on the Internet.

Usage: It allowed users to perform keyword searches across a network of data sources, similar to modern search engines.

Decline: WAIS was eventually overshadowed by the development of the web and search engines like Google.

6. Archie

Archie was one of the first search engines, created to index FTP servers to make it easier for users to find and retrieve files.

Functionality: Archie indexed publicly available FTP sites, allowing users to search for specific files on these servers.

Legacy: Archie was one of the precursors to modern search engines, but its scope was limited to FTP sites.

7. Veronica

Veronica (Very Easy Rodent-Oriented Netwide Index to Computerized Archives) was a search system for the Gopher network.

Functionality: It provided a search interface for users to find specific documents or information within the Gopher space, similar to how modern search engines index the web.

Introduction to Internet Protocols

The Internet operates using a suite of protocols that define how data is transmitted between devices. These protocols ensure interoperability and enable the functioning of online services.

1. **HTTP (Hypertext Transfer Protocol)**
   HTTP is the primary protocol used for transferring web pages on the World Wide Web.
   Functionality: HTTP defines how browsers request web pages from servers and how servers respond to those requests.
   Request: A client (browser) sends a request for a resource (like a web page) to a web server.
   Response: The server responds with the requested content or an error message (like "404 Not Found").
   HTTPS (HTTP Secure): A secure version of HTTP that uses SSL/TLS encryption to protect data during transfer, ensuring confidentiality and integrity of data.
2. **FTP (File Transfer Protocol)**
   FTP is used to transfer files between computers or from a server to a client (and vice versa) over the Internet.
   Functionality: It allows users to upload and download files, manage directories, and perform file operations remotely on a server.
   Ports: Typically uses port 21 for command communication and port 20 for data transfer.
3. **SMTP (Simple Mail Transfer Protocol)**
   SMTP is the protocol used to send email messages across the Internet.
   Functionality: SMTP is responsible for the transmission of emails from the sender's email client or server to the recipient's email server.
   Commonly Used Together: SMTP is often paired with IMAP (Internet Message Access Protocol) or POP3 (Post Office Protocol) for retrieving and managing received emails.
   World Wide Web (WWW)

The World Wide Web is a vast collection of documents, multimedia, and services accessible via browsers over the Internet. It enables users to view and interact with web pages and services.
Elements of the Web
Web Pages: HTML-based documents containing text, images, links, and other multimedia content.
URLs (Uniform Resource Locators): The address used to locate a resource on the web (e.g., `https://www.example.com`).
Web Resources: Files such as images, scripts, stylesheets, or videos that are loaded when viewing a web page.
Web Browser and Its Architecture
Web Browser: A software application that allows users to access and view content on the web.

Architecture:

1. User Interface (UI): The front-end that users interact with (address bar, buttons, etc.).
2. Rendering Engine: Interprets HTML, CSS, and JavaScript to display web pages. For example, Chrome uses Blink, and Firefox uses Gecko.
3. JavaScript Engine: Executes JavaScript to add interactivity. Chrome uses V8, while Firefox uses SpiderMonkey.
4. Networking: Manages HTTP requests and responses.
5. Data Storage: Stores browsing history, cookies, cache, and other data for improved performance and user experience.

Web Server: A web server is a system that stores web content and serves it to users when requested. Web Server Software: Examples include Apache HTTP Server, Nginx, and Microsoft IIS.

Functionality: It listens for incoming requests, processes them, and sends back the requested files (e.g., HTML pages, images).

Proxy Server: A proxy server acts as an intermediary between a user's device (client) and a web server.

Uses: Privacy: A proxy can hide a user's IP address, providing anonymity.

Security: It can filter content, block access to harmful websites, or cache content to improve performance.

Access Control: It's often used in corporate environments to restrict access to certain websites.

Microsoft Internet Explorer (IE)

Internet Explorer (IE) was once the dominant web browser on Windows but was phased out and replaced by Microsoft Edge in 2015.

Features: It provided basic browsing features but lacked support for modern web standards and security practices, leading to its decline.

Viewing Pages with a Browser : Browser Functionality: Web browsers interpret HTML, CSS, and JavaScript to render and display web pages.

HTML: Provides the structure of a webpage.

CSS: Styles the webpage, controlling layout, colors, and fonts.

JavaScript: Adds interactivity and dynamic behavior (e.g., forms, animations).

Using a Browser for Mail, News, and Chat

Mail: Browsers provide access to web-based email services like Gmail, Outlook, and Yahoo Mail.

News: Browsers allow users to access news websites or RSS feeds for real-time news updates.

Chat: Browsers support web-based messaging services, like WebRTC, live chat on websites, and social media platforms (e.g., Facebook, Twitter).

Security and Privacy Issues : As the Internet grew, so did concerns about security and privacy. Here are the main issues:

1. **Cookies :** Cookies are small text files stored on a user's device that track information such as login status, preferences, and browsing activity.
   Privacy Concerns: While cookies are useful, they can also be used to track users across different websites, leading to privacy issues.
   Third-Party Cookies: Used by advertisers to track users across websites, raising privacy concerns.
2. **Firewalls :** A firewall is a network security system that monitors and controls incoming and outgoing traffic based on security rules.
   Purpose: Firewalls are used to protect networks from unauthorized access, threats, and malicious attacks.
3. **Data Security :** Data Security involves safeguarding sensitive information from unauthorized access or corruption.
   Encryption: Protocols like SSL/TLS encrypt data during transmission, ensuring that sensitive information remains confidenstial.

1. Introduction to HTML

- **HTML (HyperText Markup Language)** is the standard language used to **create and design webpages**.
- It defines the **structure and layout** of a web document by using a variety of tags and attributes.
- HTML is not a programming language; it is a **markup language** that tells the browser how to display content.
- HTML documents are saved with the .html or .htm extension.

Basic Structure of an HTML Page:

```
<!DOCTYPE html>
<html>
<head>
<title>My First Web Page</title>
</head>
<body>
<h1>Welcome to My Website</h1>
<p>This is a sample paragraph.</p>
</body>
</html>
```

2. HTML Elements

An HTML element represents a piece of content on a web page enclosed by opening and closing tags.

Example:

```
<p>This is a paragraph.</p>
```

Elements can be nested inside each other.

Some elements are self-closing, such as <br>, <img>, and <hr>.

3. HTML Semantics

Semantic HTML means using tags that have meaningful names describing their purpose.

Semantic elements improve readability, accessibility, and SEO.

Examples of Semantic Elements :

Tag	**Meaning**
<header>	Header section of a page or article
<nav>	Navigation bar or links
<section>	Thematic grouping of content
<article>	Self-contained article or blog post
<aside>	Sidebar or supplementary content
<footer>	Footer information
<main>	Main content area of a page

4. HTML5 Doctype

The doctype declaration tells the browser which HTML version is used.

The HTML5 doctype is simple and universal:

```
<!DOCTYPE html>
```

5. New Structure Tags in HTML5

HTML5 introduced new **structural tags** for better document organization:

Tag	Description
<section>	Groups related content together
<nav>	Defines a navigation area for menus/links
<article>	Represents independent content like blog posts
<aside>	Defines content related to the main content (sidebar)
<header>	Represents introductory content or navigation links
<footer>	Represents footer information of a document or section

6. HTML Attributes

Attributes provide **additional information** about HTML elements.

They are placed **inside the start tag** and usually come in name/value pairs.

Example:

```
<img src="image.jpg" alt="Sample Image" width="200" height="150">
```

Common Attributes:

id – unique identifier for an element.

class – specifies one or more class names for an element.

src – specifies image or media source.

href – defines link destination.

style – inline styling.

alt – alternative text for images.

7. Headings and Paragraphs

HTML provides six heading tags (<h1> to <h6>) where <h1> is the largest and <h6> the smallest.

Headings define the structure and hierarchy of the content.

Paragraphs are created using the <p> tag.

Example:

```
<h1>Main Heading</h1>
<h2>Subheading</h2>
<p>This is a paragraph of text.</p>
```

8. Styles

HTML elements can be styled using the style attribute, internal CSS, or external CSS.

Inline Style Example:

```
<p style="color: blue; font-size: 18px;">This is a styled paragraph.</p>
```

9. Quotations

HTML provides specific tags for quotes:

<blockquote> – for long quotations (block level).

<q> – for short, inline quotations.

<cite> – to cite the source of a quote.

Example:

```
<blockquote>"HTML makes web pages come alive."</blockquote>
<p>He said, <q>HTML is easy to learn.</q></p>
```

10. Blocks, Classes, and Layout

Block elements: Start on a new line and take full width (e.g., <div>, <p>, <section>).

Inline elements: Do not start on a new line (e.g., <span>, <a>, <img>).

Classes: Allow multiple elements to share the same styling using CSS.

Example:

```
<div class="content">
<p>This is a paragraph inside a block.</p>
</div>
```

11. Iframes

An iframe embeds another web page within a page.

Example:

```
<iframe src="https://www.wikipedia.org" width="600" height="400"></iframe>
```

12. Creating HTML Pages

Steps:

1. Open a text editor (VS Code, Sublime, Notepad).
2. Write HTML code using tags.
3. Save the file with a .html extension.
4. Open the file in a web browser.

13. Horizontal Rules and Graphical Elements

<hr> tag: Inserts a horizontal line used to separate content sections.

Graphical elements:

- <img> – display images.
- <canvas> – draw graphics with JavaScript.
- <svg> – scalable vector graphics.

Example:

```
<img src="logo.png" alt="Logo" width="150">
<hr>
```

14. Hyperlinks

Used to connect one page to another or to external sites using <a> tag.

Example:

```
<a href="https://www.example.com" target="_blank">Visit Example</a>
```

Attributes:

href: link destination

target="_blank": opens in new tab

title: tooltip on hover

15. Creating HTML Tables

Used to present data in tabular form.

Example:

```
<table border="1">
<tr>
<th>Name</th>
<th>Age</th>
</tr>
<tr>
<td>John</td>
<td>25</td>
</tr>
</table>
```

Common Tags:

<table> – defines table

<tr> – defines row

<th> – table header

<td> – table cell

16. Creating HTML Forms

Used to collect user input.

Example:

```
<form action="submit.php" method="post">
<label for="name">Name:</label>
<input type="text" id="name" name="username"><br><br>
<input type="submit" value="Submit">
</form>
```

Form Elements:

<input> – text, password, radio, checkbox, submit

<textarea> – multi-line text input

<select> – dropdown list

<button> – clickable button

17. HTML and Image Techniques

Images are added using <img> tag.

Always include alt text for accessibility.

Use CSS for image styling and responsiveness.

Example:

```
<img src="nature.jpg" alt="Beautiful Nature" width="400" height="250">
```

18. HTML and Page Development

a. Planning

Define the purpose and target audience of the website.

Create wireframes and a content outline.

b. Navigation and Themes

Ensure easy navigation using <nav> bars.

c. Elements of a Web Page

Header

Navigation Bar

Main Content

Sidebar / Aside

Footer

d. Steps of Creating a Website

1. Plan structure and design.
2. Create pages with HTML and CSS.
3. Add interactivity with JavaScript.
4. Test website on different devices and browsers.
5. Optimize for performance and SEO.

e. Publishing and Publicizing the Site

Buy a **domain name** and **web hosting.**

Upload site files via **FTP** or a CMS.

Promote via **SEO, social media,** and **online directories**.

f. Structuring a Website

Maintain a clear folder structure:

/index.html

/about.html

/contact.html

/css/style.css

/js/script.js

/images/

Summary

HTML provides the **foundation** of web development.

It defines the **structure, content, and semantics** of web pages.

Combined with **CSS** and **JavaScript**, it creates modern, interactive, and responsive websites.

Proper planning, structuring, and design principles ensure a **professional and user friendly** web experience.

Cascading Style Sheets (CSS)

1. Understanding Style Sheets

CSS (Cascading Style Sheets) is a style language used to control the presentation and layout of HTML documents.

It separates content (HTML) from design (CSS), making web development easier and more efficient.

CSS defines how HTML elements are displayed on screen, paper, or other media.

Benefits of CSS:

Improves design consistency.

Enables easier site-wide style updates.

Reduces code redundancy.

Enhances webpage loading speed and maintainability.

2. CSS Syntax

The CSS syntax consists of:

```
selector {
property: value;
}
```

Example :

```
p {
color: blue;
font-size: 16px;
}
```

Selector – targets an HTML element (e.g., p, h1, .class, #id).

Property – defines the aspect to be styled (e.g., color, margin, font-size).

Value – specifies the style setting.

3. Applying Style Sheets to HTML Documents

CSS can be applied to HTML in three ways:

a) Inline CSS

Applied directly to an element using the style attribute.

```
<p style="color: red; font-size: 18px;">This is inline styling.</p>
```

b) Internal (Embedded) CSS

Defined inside the <style> tag in the HTML <head> section.

```
<head>
<style>
body { background-color: lightblue; }
h1 { color: navy; }
</style>
</head>
```

c) External CSS

Stored in a separate .css file and linked using <link> tag.

```
<link rel="stylesheet" href="style.css">
```

External CSS is recommended for large websites for better management.

4. Developing Style Sheets

Identify common design elements (fonts, colors, layouts).

Create an external stylesheet (style.css).

Link it to all HTML files.

Use consistent class and ID names for reusability.

5. CSS Selectors

Selectors are used to target HTML elements for styling.

Types of CSS Selectors :

Selector Type	Example	Description
Element	p {}	Selects all <p> elements
Class	.title {}	Selects elements with class="title”
ID	#main {}	Selects the element with id="main”
Universal	* {}	Selects all elements
Grouping	h1, h2, p {}	Styles multiple elements at once
Descendant	div p {}	Targets <p> inside <div>
Pseudo-class	a:hover {}	Styles element in specific state

6. The <div> Tag

<div> (division) is a block-level container used to group HTML elements.

Commonly used with CSS for layout and design.

Example:

```
<div class="container">
<h2>Welcome</h2>
<p>This is inside a div container.</p>
</div>
```

7. **Using Class and ID**

Class Selector (.classname)

Used to style multiple elements with the same class name.

```
<p class="highlight">Highlighted text.</p>
.highlight { color: red; }
```

ID Selector (#idname)

Used to style a unique element.

```
<p id="unique">Unique paragraph.</p>
#unique { font-size: 20px; }
```

8. **Styling BackgroundsCSS allows customization of element backgrounds.**

Properties:

background-color
background-image
background-repeat
background-size
background-position

Example:

```
body {
background-color: #f2f2f2;
background-image: url('bg.jpg');
background-repeat: no-repeat;
background-size: cover;
}
```

9. **Styling Borders**

Properties:

border-style (solid, dotted, dashed)
border-width
border-color
border-radius (for rounded corners)

Example:

```
div {
border: 2px solid black;
border-radius: 10px;
}
```

10. Styling Text

Common Properties:

color

text-align

text-decoration

text-transform

letter-spacing

Example:

```
h1 {
color: darkblue;
text-align: center;
text-transform: uppercase;
}
```

11. Styling Fonts

Properties:

font-family – sets the font type.

font-size – defines text size.

font-style – italic, normal, etc.

font-weight – bold, light, normal.

Example:

```
p {
font-family: 'Arial', sans-serif;
font-size: 16px;
}
```

12. Styling Links

Different link states can be styled using pseudo-classes:

```
a:link { color: blue; }
a:visited { color: purple; }
a:hover { color: red; }
a:active { color: green; }
```

13. Styling Lists

Example:

```
ul {
list-style-type: square;
padding-left: 30px;
}
```

14. Styling Tables

Example :

```
table {
border-collapse: collapse;
width: 100%;
}
th, td {
border: 1px solid #ccc;
padding: 10px;
}
th {
background-color: #f2f2f2;
}
```

15. Margins

Margins create space outside an element's border.

Syntax:

```
margin: 10px 20px 10px 20px; /* top right bottom left */
```

Use margin: auto; for centre alignment.

16. Flex and Grid Layouts

a) Flexbox

Flexible box layout for aligning and distributing space among items.

Example:

```
.container {
display: flex;
justify-content: space-around;
align-items: center;
}
```

b) Grid Layout

Provides two-dimensional layout control (rows and columns).

Example:

```
.container {
display: grid;
grid-template-columns: 1fr 2fr 1fr;
gap: 10px;
}
```

17. Bootstrap & Web Page Design

Bootstrap is a popular CSS framework used to design responsive websites quickly.

It includes ready-made classes for layout, forms, buttons, navigation & components.

Advantages:

Mobile-first design

Prebuilt responsive grid system

Consistent styling and faster development

Example:

```
<link rel="stylesheet”
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
<button class="btn btn-primary">Click Me</button>
```

18. CMS (Content Management Systems)

A CMS is software that allows users to create, edit, and manage websites without deep coding knowledge.

Examples of CMS :

WordPress

Joomla

Drupal

Wix, Squarespace

Benefits:

User-friendly interface

Theme and plugin support

SEO and security features

JavaScript

1. Introduction to Scripting Language

A **scripting language** is a type of programming language that is used to write scripts—small programs that automate tasks and control software applications. Unlike compiled languages such as C or Java, scripting languages are **interpreted,** meaning that the code runs directly without the need for compilation.

JavaScript is the most popular **client-side scripting language** used to add interactivity, dynamic behavior, and logic to web pages. It works hand-in-hand with HTML (for structure) and CSS (for presentation).

Features of JavaScript:

Lightweight and easy to learn.

Interpreted and dynamically typed.

Object-based and event-driven.

Platform-independent (runs in all modern browsers).

Integrated with HTML and CSS.

Supports asynchronous programming (using promises and async/await).

Example:

```
<script>
document.write("Welcome to JavaScript!");
</script>
```

2. Client-Side Scripting

Client-side scripting means the script is executed on the user's browser rather than on the server.

JavaScript is the primary client-side scripting language used to make web pages interactive.

Functions of Client-Side Scripting:

Validate form inputs before sending data to the server.

Create dynamic content (e.g., image sliders, dropdown menus).

Respond to user events (e.g., clicks, mouseovers).

Modify HTML and CSS dynamically through the DOM (Document Object Model).

Example:

```
<button onclick="alert('Hello!')">Click Me</button>
```

3. Memory Concepts

JavaScript uses automatic memory management through garbage collection, meaning developers do not manually allocate or free memory.

When an object is no longer referenced, it is automatically deleted from memory.

Types of Memory:

1. **Stack Memory:**

   Used for primitive values like strings, numbers, booleans.

2. **Heap Memory:**

   Used for objects, arrays, and functions.

Garbage Collection:

JavaScript's garbage collector periodically identifies and removes objects that are no longer accessible.

4. Arithmetic and Decision Making

Arithmetic operations are the foundation of calculations in JavaScript.

Operators:

+, -, *, /, % – Basic arithmetic.

++, -- – Increment/decrement.

Example:

```
let x = 10;
let y = 5;
console.log(x + y); // 15
```

Decision Making:

JavaScript supports conditional statements to make decisions during program execution.

Example:

```
if (x > y) {
console.log("x is greater");
} else {
console.log("y is greater");
}
```

5. JavaScript Control Structures

Control structures guide the flow of execution of a program.

They include conditional statements and loops.

Conditional Statements:

if, if...else, if...else if...else, and switch.

Example:

```
switch(day) {
case 1: console.log("Monday"); break;
case 2: console.log("Tuesday"); break;
default: console.log("Invalid day");
}
```

Loops:

for, while, do...while, for...in, and for...of.

Example:

```
for (let i = 1; i <= 5; I++) {
console.log("Number: " + I);
}
```

6. JavaScript Functions

Functions are reusable blocks of code designed to perform specific tasks. They make programs modular and easier to maintain.

Syntax:

```
function functionName(parameters) {
// code block
return value;
}
```

Example:

```
function add(a, b) {
return a + b;
}
console.log(add(5, 10)); // 15
```

Types of Functions:

User-defined functions

Built-in (global) functions

Arrow functions (ES6)

7. JavaScript Popup Boxes

JavaScript provides three types of popup boxes to interact with users.

Type	Purpose	Example
alert()	Displays a message boxalert	("Welcome!");
confirm()	Asks for user confirmationconfirm	("Are you sure?");
prompt()	Accepts user input	prompt("Enter your name:");

8. Events in JavaScript

Events are actions or occurrences (like clicks, key presses, or page loads) that can be detected and handled by JavaScript.

Common Events:

Event	Description
onclick	Triggered when an element is clicked
onmouseover	Triggered when the mouse is over an element
onload	When a webpage finishes loading
onchange	When an input value changes
onsubmit	When a form is submitted

Example:

```
<button onclick="changeText()">Click Me</button>
<p id="demo">Hello!</p>
<script>
function changeText() {
document.getElementById("demo").innerHTML = "You clicked the button!";
}
</script>
```

9. Program Modules in JavaScript

JavaScript code can be split into modules for organization and reusability.

Modules help divide a large program into smaller, manageable files.

Example (ES6 Modules):

```
// file: math.js
export function add(a, b) { return a + b; }
// file: main.js
import { add } from './math.js';
console.log(add(5, 10));
```

10. Function Definitions, Duration of Identifiers, Scope Rules

Scope:

Determines the visibility and accessibility of variables.

Global Scope: Accessible everywhere.

Local/Function Scope: Accessible only inside a function.

Block Scope: Created by let and const inside {}.

Duration of Identifiers :

Variables exist as long as their scope is active.

After exiting the scope, they are destroyed automatically.

Example:

```
let globalVar = "Global";
function demo() {
let localVar = "Local";
console.log(globalVar); // accessible
}
```

11. Controlling Programming Flow

JavaScript controls execution using **loops, conditional statements**, and **jump statements** (break, continue, return).

This ensures logical flow, repetition, and selective execution of code blocks.

12. Recursion

Recursion is a technique where a function calls itself until a condition is met.

Example:

```
function factorial(n) {
if (n === 0) return 1;
else return n * factorial(n - 1);
}
console.log(factorial(5)); // 120
```

13. JavaScript Global Functions

JavaScript provides built-in global functions for common operations:

Function	Description
parseInt()	Converts string to integer
parseFloat()	Converts string to floating-point
isNaN()	Checks if a value is not a number
eval()	Executes a string as JavaScript code
encodeURI() / decodeURI()	Encodes/decodes URLs

14. Arrays Handling in JavaScript

Arrays store multiple values under one name.

Example:

```
let fruits = ["Apple", "Banana", "Cherry"];
console.log(fruits[1]); // Banana
```

Common Methods:

push(), pop(), shift(), unshift()

concat(), slice(), splice(), sort(), reverse()

Looping through Arrays:

```
for (let fruit of fruits) {
console.log(fruit);
```

15. The JavaScript Object Model

Everything in JavaScript is an object.

An object is a collection of properties (key-value pairs).

Example:

```
let person = {
name: "John",
age: 25,
greet: function() {
console.log("Hello " + this.name);
}
};
person.greet();
```

Objects can represent real-world entities and help organize code logically.

16. Developing Interactive Forms

JavaScript adds interactivity and dynamic validation to web forms.

Example:

```
<form onsubmit="return validateForm()">
<input type="text" id="username">
<input type="submit" value="Submit">
</form>
<script>
function validateForm() {
let name = document.getElementById("username").value;
if (name === "") {
alert("Name cannot be empty!");
return false;
}
return true;
}
/script>
```

17. Validation of Forms

Form validation ensures user inputs are correct before submission.

Types of Validation:

Client-side: Done using JavaScript before submission.

Server-side: Performed on the server after submission.

Example :

```
if (!email.includes("@")) {
alert("Invalid email address!");
}
```

18. Cookies and JavaScript Security

Cookies are small text files stored on the client browser to store data such as usernames, preferences, or session IDs.

Creating and Reading Cookies:

```
document.cookie = "username=John; expires=Fri, 31 Dec 2025 12:00:00 UTC";
alert(document.cookie);
```

Security Measures:

Use secure and HttpOnly flags for cookies.

Validate user inputs to prevent Cross-Site Scripting (XSS).

Avoid storing sensitive data in cookies.

19. Controlling Frames in JavaScript

Frames allow displaying multiple HTML documents in one browser window.

JavaScript can control frames through the window or parent object.

Example:

```
parent.frames[0].location = "page2.html";
```

20. Client-Side JavaScript Customization

Client-side JavaScript allows developers to customize user experiences by modifying page content and style dynamically.

Uses :

Changing content based on user input.

Dynamic themes and layouts.

Real-time data updates (AJAX).

Animation and multimedia control.

Example :

```
document.getElementById("welcome").innerHTML = "Hello, User!";
```

Summary

JavaScript is the backbone of modern web interactivity.

It enables:

Dynamic content updates

Form validation and user interaction

Complex logic and modular programming

Secure, responsive, and personalized websites

OUR GROUP OF COLLEGES

Girls Colleges

Biyani Girls College
www.biyanicolleges.org
Affiliated to University of Rajasthan
B.B.A.| B.Com. (Pass Course/Hons.) | CA/CS
BCA
B.A.| BVA (Visual Arts)
B.Sc. (Biology/Maths/Biotech.)
M.Com. (ABST) | M.Sc. (Biotech/Physics/Maths/Chemistry)
Zoology (Botany/Environmental Science) | M.A. (Geography)
English Literature/Economics)
IAS/RAS

Biyani Institute of Science & Management for Girls
www.bisma.in
Affiliated to Rajasthan Technical University, Kota
MBA | Ph.D

Biyani Girls B.Ed. College
www.biyanigirlscollege.com
Affiliated to University of Rajasthan
Approved by National Council of Teachers' Education
B.Ed.|M.Ed.|B.Sc.-B.Ed|M.Ed|D.El. Ed.

Biyani School of Nursing & Paramedical Sciences
Biyani Institute of Science & Management (Nurs.)
www.biyaninursingcollege.com
Affiliated to RUHS, Jaipur
Approved by INC and RNC
G.N.M.|B.Sc.Nursing

Biyani Institute of Skill Development for Girls
www.bisd.in
Affiliated to Rajasthan ILD Skill Development University
B.Voc. - Fashion Designing
B. Voc. - Journalism & Mask Communication

Biyani Institute of Yoga & Naturopathy for Girls
www.biyanicolleges.org
Affiliated to Jagadguru Ramanandacharya
Rajasthan Sanskrit University, Jaipur
PGDYT

Co-Ed Colleges

Biyani College of Science & Mgmt. (Co-Ed.)
www.bcsmjaipur.com/edu
Affiliated to University of Rajasthan
B.A.|B.Sc.| B. Com.|BCA-B.Ed.|B.Sc.- B.Ed.

Biyani Law College (Co-Ed.)
www.biyanilawcollege.com
Affiliated to Dr. Bhimrao Ambedkar Law University, Jaipur
Approved by Bar Council of India, New Delhi
BA-LL.B.| LLB.| LLM.| PGD.LL.

Biyani Institute of Pharmaceutical Sciences (Co.Ed.)
www.biyanipharmacycollege.com
Affiliated to RUHS, Jaipur
Approved by Pharmacy Council of India, New Delhi
D. Pharma | B. Pharma

Biyani Institute of Architecture & Design
www.biyanicolleges.org
Affiliated to Rajasthan Technical University, Kota
Approved by AICTE, New Delhi
B. Arch.

Biyani Private ITI (Co-Ed.)
www.biyaniiti.com
Approved by Quality Council of India, New Delhi
ITI Trade- Elecrician

Biyani Institute of Physical Education (Co-Ed.)
www.bcsmjaipur.com
Affiliated to University of Rajasthan
B.P. Ed.

Jaipur Institute of Yoga & Naturopathy
www.biyanicolleges.org
Affiliated to Jagadguru Ramanandacharya
Rajasthan Sanskrit University, Jaipur
PGDYT

Department of Information Technology
Biyani Girls College
Sector-3, Vidhyadhar Nagar, Jaipur,Rajasthan
+91-8696218218,+91-8290638942
acad@biyanicolleges.org