
HTML, CSS, JavaScript1. HTML: Structures web content (headings, paragraphs, links).2. CSS: Styles and layouts web pages (colors, fonts, responsiveness).3. JavaScript: Adds interactivity (dynamic behavior, events, DOM manipulation)

Introduction to Web DevelopmentWeb development is the process of building and maintaining *websites and web applications* for the internet. Key Aspects1. *Front-end*: Client-side development focusing on user interface and experience (HTML, CSS, JavaScript).2. *Back-end*: Server-side logic, databases, and APIs (languages like Node.js, Python, Ruby).3. *Full-stack*: Combination of front-end and back-end development.Core Technologies- *HTML*: Structure of web pages.- *CSS*: Styling and layout.- *JavaScript*: Interactivity and dynamic behavior.Web Development Process1. *Planning*: Define goals and requirements.2. *Design*: Create wireframes and visual designs.3. *Development*: Coding (front-end, back-end).4. *Testing*: Ensure functionality and compatibility.5. *Deployment*: Publish to a web server. Career Paths- Front-end Developer: Focuses on UI/UX.- Back-end Developer: Works on server-side logic.- *Full-stack Developer*: Handles both front-end and back-end.Tools and Trends- *Code Editors*: VS Code, Sublime Text.- *Frameworks*: React, Angular, Vue.js (JavaScript).- *Responsive Design*: Building sites for various devices.Web development is a dynamic field enabling creation of *interactive and accessible online experiences*.

HTML BasicsWhat is HTML?HTML (Hypertext Markup Language) is the standard markup language used to create the *structure and content of web pages*. HTML StructureA basic HTML document has the following structure:<!DOCTYPE html><html> <head> <title>Page Title</title> </head> <body> <!– Content goes here □ </body></html>

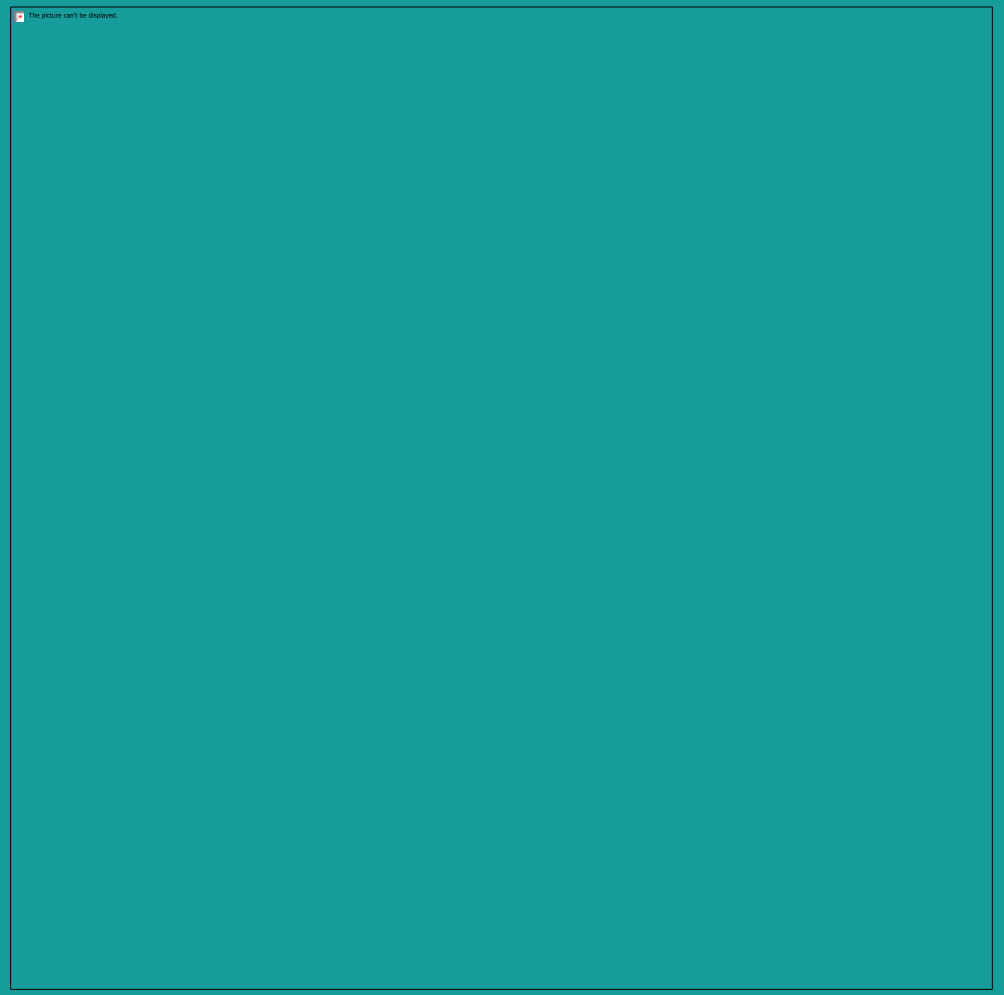
Key HTML Tags1. *`<html>`*: Root element of an HTML document.2. *`<head>`*: Contains meta-information like title, charset.3. *`<title>`*: Sets the title shown in browser tab.4. *`<body>`*: Contains the visible page content.5. *`<h1>` to `<h6>`*: Headings of different levels.6. *`<p>`*: Paragraph.7. *`<a>`*: Hyperlink.8. *`<img>`*: Image.9. *`<div>`*: Division/block container.10. *`<span>`*: Inline container. Example HTML Document<!DOCTYPE html><html> <head> <title>My First HTML Page</title> </head> <body> <h1>Welcome</h1> This is a paragraph of text. <a href="https: <img src="image.jpg" alt="An example image"> </body></html> Attributes- HTML tags can have *attributes* providing additional info.- Example: `<img src="image.jpg" alt="Description">` (`src`, `alt` are attributes). Semantic HTML- Uses tags like `<header>`, `<nav>`, `<main>`, `<footer>` for meaning.- Improves *accessibility and SEO*.HTML is foundational for building web pages and works with CSS for styling and JavaScript for interactivity.

HTML Structure

- !DOCTYPE html><html lang="en"><head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>My First Webpage</title></head><body > <h1>Hello, World!</h1> This is my first webpage using HTML structure. </body></html>

CSS (Cascading Style Sheets) What is CSS?CSS is used to control the *visual styling and layout* of HTML elements on web pages.CSS SyntaxA CSS rule consists of a *selector* and *declarations*:selector { property: value;}Example:h1 { color: blue; font-size: 24px;}Types of CSS1. *Inline CSS*: Styles applied directly in HTML tags via `style` attribute.<h1 style="color: red;">Heading</h1>2. *Internal CSS*: Styles defined in `<style>` tag within HTML `<head>`.<head> <style> h1 { color: green; } </style></head>3. *External CSS*: Styles in a separate `.css` file, linked to HTML.<link rel="stylesheet" href="styles.css">

Selectors- *Element Selector*: `h1`, `p`, `div`.- *Class Selector*: `.classname`.- *ID Selector*: `#idname`.- *Attribute Selector*: `[type=“text”]`.

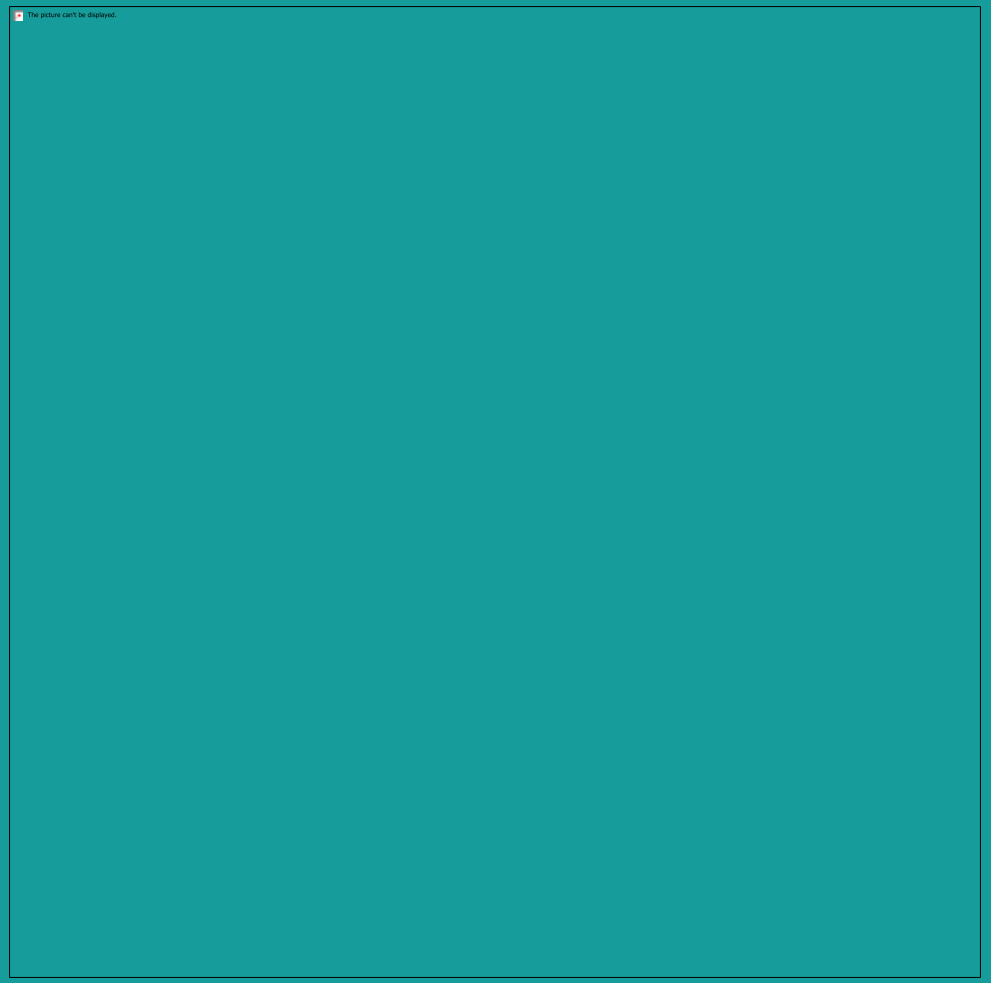


Stylish Examples1. *Colorful Heading*:h1 { color: #3498db; font-family: ‘Arial’, sans-serif; text-align: center;}2. *Styled Button*:.button { background-color: #2ecc71; color: white; padding: 10px 20px; border: none; border-radius: 5px; cursor: pointer;}.button:hover { background-color: #27ae60;}3. *Responsive Image*:img { max-width: 100%; height: auto;} Key CSS Concepts- *Box Model*: Margin, Border, Padding, Content.- *Flexbox*: Layout technique for flexible arrangements.- *Grid*: Powerful layout system for 2D arrangements.- *Media Queries*: For responsive design adapting to screen sizes.### CSS Properties- *Text*: `color`, `font-size`, `text-align`.- *Layout*: `margin`, `padding`, `display`.- *Visuals*: `background`, `border`, `box-shadow`.CSS enables *creative and responsive styling* of web pages, enhancing user experience.

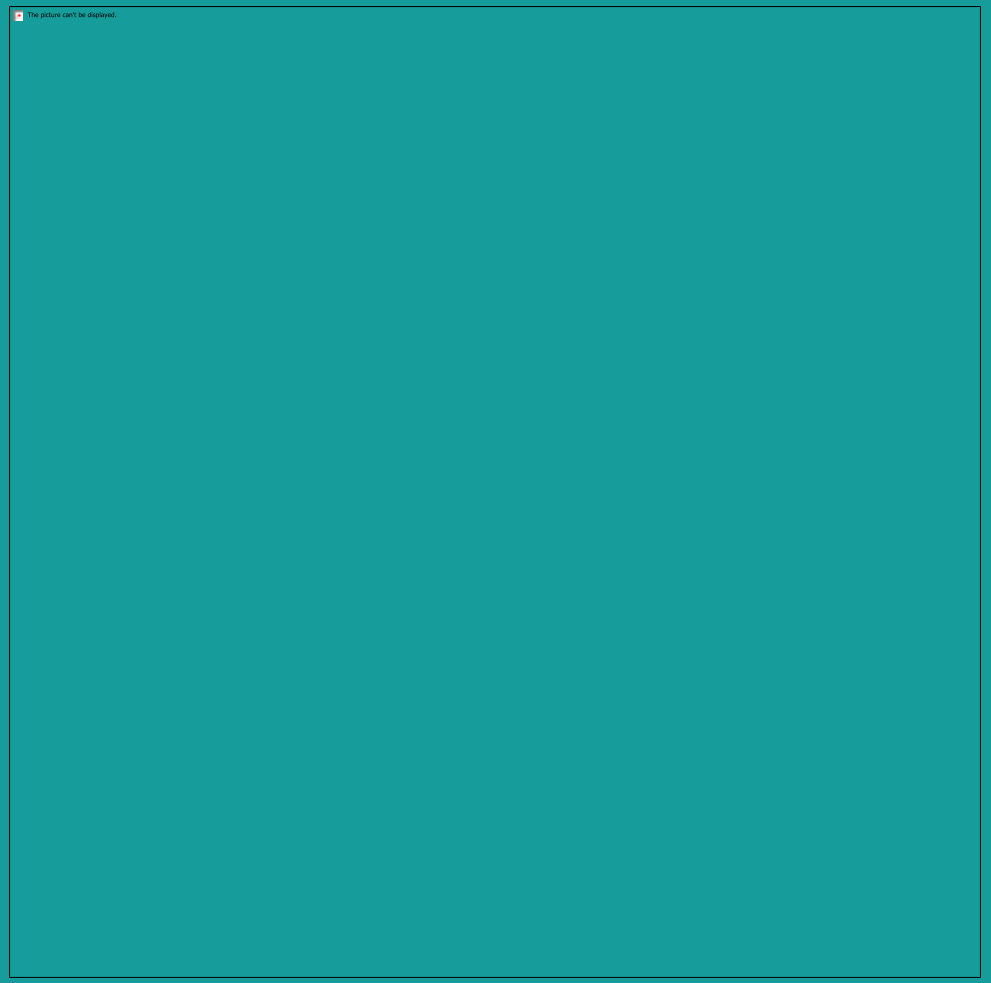
JavaScript IntroductionWhat is JavaScript?JavaScript is a *high-level, dynamic, interpreted programming language* primarily used for adding *interactivity to web pages*.

Key Features of JavaScript1. *Client-side scripting*: Runs in web browsers to create dynamic web pages.2. *Event-driven*: Responds to user interactions like clicks, form submissions.3. *DOM Manipulation*: Can interact with and modify the Document Object Model (DOM).4. *Asynchronous programming*: Handles tasks like API calls, timers.5. *Cross-platform*: Used in web browsers, Node.js (server-side), mobile apps.Uses of JavaScript1. *Web Development*: Front:** For interactive websites (DOM manipulation, events).2. *Web Applications*: Complex apps with frameworks like React, Angular, Vue.3. *Server-side*: Node.js for backend development.4. *Mobile/Desktop Apps*: Using frameworks like React Native, Electron.

Simple JavaScript Examples1. *Hello World*:console.log(“Hello, World!”);2. *Variable Declaration*:let name = “John”;console.log(name);3. *Function*:function greet(name) { return `Hello, ${name}!`;}console.log(greet(“Alice”));4. *Event Handling*:document.getElementByld(“myButton”).addEventListener(“click”, function() { alert(“Button clicked!”);}); Core JavaScript Concepts- *Variables*: `let`, `const`, `var`.- *Data Types*: Strings, Numbers, Booleans, Arrays, Objects.- *Functions*: Reusable blocks of code.- *Events*: Handling user interactions.- *DOM Manipulation*: Changing HTML/CSS via JavaScript.Popular JavaScript Frameworks/Libraries- *React*: For building user interfaces.- Node.js: For server-side JavaScript.- *jQuery*: Simplified DOM manipulation (older but still used). JavaScript in Browsers- Executes in web browsers to make pages *interactive and dynamic*.- Can *validate forms*, create *animations*, handle *AJAX requests*.JavaScript is *versatile and widely used* for both front-end and back-end development, enabling rich interactive experiences.



Here’s a clear comparison of HTML, CSS, and JavaScript so you can understand how they work individually and together in web development:---1. PurposeHTML (HyperText Markup Language):Structure of a webpage.Defines elements like headings, paragraphs, images, links, tables, forms, etc.Think of it as the skeleton of a website.CSS (Cascading Style Sheets):Styling of a webpage.Controls colors, fonts, layouts, spacing, animations, responsiveness, etc.Think of it as the skin and design that makes the skeleton look attractive.JavaScript (JS):Interactivity and logic of a webpage.Adds dynamic behavior (sliders, pop-ups, form validation, data fetching, etc.).Think of it as the brain that makes the webpage respond to user actions.---2. Role in the Web Development ProcessAspect HTML CSS 🎨 JavaScript ⚡Role Structure Style BehaviorFile Extension .html .css .jsPosition Backbone (content) Decoration (presentation) Functionality (logic)Example Use Add text, images, forms Make text red, center images, responsive layouts Validate a form, show alerts, fetch data---3. ExampleHere’s how they work together:<!DOCTYPE html><html><head> <title>Comparison Example</title> <style> h1 { color: blue; text-align: center; } p { font-size: 18px; } </style></head><body> <!– HTML Structure <h1>Hello World</h1> This is styled with CSS and interactive with JavaScript. <button onclick=“showMessage()”>Click Me</button> <!– JavaScript <script> function showMessage() { alert(“You clicked the button!”); } </script></body></html>HTML: Provides <h1>, , <button>.CSS: Styles heading color, text alignment, font size.JavaScript: Adds button functionality (shows an alert



The importance of HTML, CSS, and JavaScript in web development in a structured way:◆ 1. HTML (HyperText Markup Language) – The FoundationImportance:Provides the structure and content of a webpage.Defines elements such as text, images, forms, links, and multimedia.Ensures semantic meaning, which improves SEO and accessibility.Serves as the base layer upon which CSS and JavaScript build.👉 Without HTML → there's no webpage content or skeleton.◆ 2. CSS (Cascading Style Sheets) – The DesignImportance:Controls the presentation and layout of a webpage.Improves user experience (UX) through styling, responsiveness, and readability.Enables consistency in design across multiple pages.Supports modern UI/UX practices like animations, grid systems, and media queries.👉 Without CSS → websites would be plain, text-heavy, and unattractive.---◆ 3. JavaScript – The InteractivityImportance:Adds functionality and interactivity (form validation, dynamic content, pop-ups, sliders).Enables real-time updates (e.g., live chat, notifications, updating content without reloading).Powers modern web applications through frameworks (React, Angular, Vue).Bridges communication with backend services via APIs.👉 Without JavaScript → websites would be static and non-interactive.◆ Overall Importance in Web DevelopmentThese three technologies work together as the core of front-end web development:HTML = Content & Structure CSS = Styling & Layout 💅JavaScript = Interactivity & Logic ⚡

Here’s a neat conclusion with references for your comparison of HTML, CSS, and JavaScript: --- Conclusion HTML, CSS, and JavaScript are the three pillars of web development, each playing a distinct but interconnected role: HTML provides the structure and semantic meaning of content. CSS handles the presentation, making web pages visually engaging and responsive. JavaScript powers the interactivity and logic, allowing users to interact dynamically with web pages. Together, they form the foundation of modern web design and development. Without