

### Lesson 1: Introduction to Web Development

#### What is Web Development?

Web development is the process of creating and maintaining websites. It involves various tasks, including web design, web content development, client-side/server-side scripting, and network security configuration.

#### Key Components of Web Development

Web development is generally divided into two main areas:

##### Front-End Development (Client-Side):

- This is the part of web development that deals with the visual and interactive aspects of a website. It's everything users see and interact with in their web browsers.

##### Languages and Technologies:

- **HTML (HyperText Markup Language):** The backbone of any web page, HTML provides the structure and content of the web page.
- **CSS (Cascading Style Sheets):** CSS controls the presentation, layout, and design of HTML elements. It makes the website look good by styling fonts, colors, and layouts.
- **JavaScript:** A programming language that adds interactivity to web pages. It can be used for dynamic content updates, form validations, animations, and more.

##### Back-End Development (Server-Side):

- This involves the server-side of web development, where the data is processed and managed. It's responsible for handling data and server communication.

##### Languages and Technologies:

- **Server-Side Languages:** These include languages like Python, Ruby, PHP, Java, and Node.js. They handle data processing, server logic, and database interactions.
- **Databases:** Systems like MySQL, PostgreSQL, MongoDB, and SQLite are used to store and manage data. They allow web applications to retrieve and manipulate data.

**Server Management:** Involves configuring and managing servers where websites are hosted. This could involve web servers like Apache or Nginx.

### Introduction to Wireframing

- A wireframe, also known as a wire, is a simplified black and white version of designs that work as a bridge between developers, designers, project managers, and clients. They ensure everyone is on the same page while discussing the overall structure and functionality of the future product.
- Wireframes define dimensions and placement of page elements, information architecture, intended behaviors, and navigation for your website.

## Introduction to HTML

Element	Description	Attributes	Example
<b>Paragraphs</b>	Defines a block of text; automatically adds space before and after.	align	<code>&lt;p align="center"&gt;This is a centered paragraph.&lt;/p&gt;</code>
<b>Headings</b>	Defines headings; ranges from <code>&lt;h1&gt;</code> (largest) to <code>&lt;h6&gt;</code> (smallest).	align	<code>&lt;h1 align="right"&gt;This is a right-aligned heading 1&lt;/h1&gt;</code>
<b>Comments</b>	Adds comments in the code; not displayed in the browser.	N/A	<code>&lt;!-- This is a comment --&gt;</code>
<b>Table</b>	Defines a table structure. Includes <code>&lt;table&gt;</code> , <code>&lt;tr&gt;</code> ,	border, cellpadding, cellspacing, width, align	<code>&lt;table border="1" cellspacing="5" cellpadding="10" width="80%" align="center"&gt;&lt;tr&gt;&lt;th&gt;Header&lt;/th&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;Data&lt;/td&gt;&lt;/tr&gt;&lt;/table&gt;</code>

	<td>, and <th>.		
<b>List</b>	A generic list tag; often used for nested lists.	N/A	<ul style="list-style-type: none"> <li>Item 1</li> <li>Item 2</li> </ul> <ol style="list-style-type: none"> <li>Item 1</li> <li>Item 2</li> </ol>
<b>Unordered List</b>	Creates a bulleted list.	type	<ul style="list-style-type: none"> <li>Item 1</li> <li>Item 2</li> </ul>
<b>Ordered List</b>	Creates a numbered list.	type, start, reversed	<ol style="list-style-type: none"> <li>Item 1</li> <li>Item 2</li> </ol> <ol reversed="" start="3" style="list-style-type: none"> <li>Item 3</li> </ol>
<b>Description List</b>	Defines a list of terms and their descriptions.	N/A	<dl> <dt>Term</dt> <dd>Description</dd> </dl>
<b>Line Break</b>	Inserts a single line break.	N/A	<p>Line 1 Line 2</p>
<b>Pre Tag</b>	Displays text in a fixed-width font, preserving whitespace and line breaks.	width, height, class, id	<pre>class="preformatted"&gt; Preformatted text</pre>

<b>Horizontal Line</b>	Inserts a horizontal rule (line).	width, size, color, align, class, id	<code>&lt;hr width="50%" size="2" color="blue" align="center" class="horizontal-line"&gt;</code>
------------------------	-----------------------------------	--------------------------------------	--

## Introduction to HTML:

- **Paragraphs (<p>):** Defines blocks of text.
  - **Attributes:** align (e.g., <p align="center">).
- **Headings (<h1> to <h6>):** Specifies headings in different sizes.
  - **Attributes:** align (e.g., <h1 align="right">).
- **Comments (<!-- -->):** Adds non-displayed comments in the code.
- **Table (<table>):** Defines tables for displaying data.
  - **Attributes:** border, cellpadding, cellspacing, align, width.
- **Unordered List (<ul>):** Creates a bulleted list.
  - **Attributes:** type (e.g., disc, circle).
- **Ordered List (<ol>):** Creates a numbered list.
  - **Attributes:** type (e.g., 1, A), start, reversed.
- **Description List (<dl>):** Creates a list of terms with descriptions.
- **Line Break (<br>):** Inserts a single line break.
- **Preformatted Text (<pre>):** Displays text in fixed-width font, preserving spaces and line breaks.
- **Horizontal Line (<hr>):** Inserts a horizontal line.
  - **Attributes:** width, size, color, align, class, id.

## Lesson 2: HTML Block & Inline, Head

### HTML Inline

HTML Element	Description	Attributes	Example
HTML Block Elements	Elements that take up the full width of their container and start on a new line.	N/A	<code>&lt;div&gt;This is a block-level element.&lt;/div&gt;</code> <code>&lt;header&gt;This is a header.&lt;/header&gt;</code>
HTML Inline Elements	Elements that do not start on a new line and only take up as much width as necessary.	N/A	<code>&lt;span&gt;This is an inline element.&lt;/span&gt;</code> <code>&lt;a href="#"&gt;This is a link.&lt;/a&gt;</code>
HTML Links	Defines a hyperlink that links to another page or resource.	href, target, title, rel	<code>&lt;a href="https://www.example.com" target="_blank" title="Example Website"&gt;Visit Example&lt;/a&gt;</code>
HTML Images	Embeds an image in the document.	src, alt, width, height, title	<code>&lt;img src="image.jpg" alt="Description of image" width="300" height="200" title="Image Title"&gt;</code>
HTML Bold	Makes text bold.	N/A	<code>&lt;b&gt;This text is bold.&lt;/b&gt;</code> <code>&lt;strong&gt;This text is strong (also bold).&lt;/strong&gt;</code>
HTML Italic	Makes text italicized.	N/A	<code>&lt;i&gt;This text is italic.&lt;/i&gt;</code> <code>&lt;em&gt;This text is emphasized (also italic).&lt;/em&gt;</code>
HTML Superscript and Subscript	Displays text as superscript or subscript.	N/A	<code>&lt;sup&gt;This is superscript.&lt;/sup&gt;</code> <code>&lt;sub&gt;This is subscript.&lt;/sub&gt;</code>

<b>HTML Formatting</b>	Various tags for formatting text.	N/A	<u>This text is underlined.</u> <span style="background-color: yellow;">This text is highlighted.</span> <small>Small text</small>
------------------------	-----------------------------------	-----	--

## HTML Inline

### HTML Block Elements:

**<div>**: Defines a division or section.

**<header>**: Represents introductory content or a group of introductory or navigational aids.

**<footer>**: Defines the footer for a document or section.

**<section>**: Represents a generic section of a document.

**<article>**: Represents a self-contained composition in a document.

### HTML Inline Elements

**<span>**: Used to group inline elements and apply styles or attributes.

**<a>**: Defines a hyperlink.

**<strong>**: Represents strong importance (typically bold).

**<em>**: Represents emphasized text (typically italic).

### HTML Links:

**href**: Specifies the URL of the page the link goes to.  
**target**: Specifies where to open the linked document (\_blank opens in a new tab).

**title**: Provides additional information about the link (displayed as a tooltip).

**rel**: Specifies the relationship between the current document and the linked document.

### HTML Images:

**src**: Specifies the path to the image.  
**alt**: Provides alternative text for the image if it cannot be displayed.  
**width**: Specifies the width of the image.

**height**: Specifies the height of the image.  
**title**: Provides additional information about the image (displayed as a tooltip).

### HTML Bold:

**<b>**: Makes text bold; not semantically important.

**<strong>**: Makes text bold; indicates that the text is of strong importance.

### HTML Italic:

**<i>**: Makes text italic; not semantically important.

**<em>**: Makes text italic; indicates emphasized text.

### HTML Superscript and Subscript

**<sup>**: Renders text as superscript.

**<sub>**: Renders text as subscript.

### HTML Formatting:

**<u>**: Underlines text.

**<mark>**: Highlights text.

**<small>**: Renders text smaller than the surrounding text.

### HTML Head

HTML Element	Description	Attributes	Example
HTML Head	Contains meta-information about the HTML document.	N/A	<code>&lt;head&gt;&lt;title&gt;Document Title&lt;/title&gt;&lt;/head&gt;</code>
HTML Title	Sets the title of the document, shown in the browser's title bar or tab.	N/A	<code>&lt;title&gt;Page Title&lt;/title&gt;</code>
HTML Style	Defines CSS styles for the document.	type (default text/css), media	<code>&lt;style type="text/css"&gt;body { background-color: lightblue; }&lt;/style&gt;</code>
HTML Meta Elements	Provides metadata about the HTML document, such as description, keywords, author, and more.	name, content, charset, http-equiv	<code>&lt;meta name="description" content="A brief description of the page."&gt; &lt;meta charset="UTF-8"&gt;</code>
HTML Favicon	Specifies the icon to be displayed in the browser tab or bookmark list.	href, rel, type	<code>&lt;link rel="icon" href="favicon.ico" type="image/x-icon"&gt;</code>

## HTML Head:

**<head>:** Contains all meta-information and linked resources like stylesheets and scripts.

## HTML Title:

**<title>:** Sets the title for the HTML document, which appears on the browser tab or window title.

## HTML Head

### HTML Style:

**<style>:** Contains CSS rules for styling the document.

**type:** Specifies the MIME type of the style (default is text/css).

**media:** Specifies the media for which the styles are intended (e.g., screen, print).

### HTML Meta Elements:

**<meta>:** Provides metadata about the document.

**name:** Defines the type of metadata (e.g., description, keywords, author).

**content:** Provides the value for the name attribute.  
**charset:** Specifies the character encoding for the document (e.g., UTF-8).

**http-equiv:** Provides an HTTP header for the value of the content attribute (e.g., Content-Type, X-UA-Compatible).

### HTML Favicon:

**<link>:** Links to external resources, such as stylesheets or icons.

**rel:** Specifies the relationship between the current document and the linked resource (e.g., icon).

**href:** Specifies the path to the favicon file.

**type:** Specifies the MIME type of the favicon (e.g., image/x-icon).



### Lesson 3: HTML Form & Semantic HTML

#### HTML Form & Semantic HTML

Element	Description	Attributes	Example
<code>&lt;form&gt;</code>	Defines an HTML form for user input	action, method, enctype, target	<code>&lt;form action="/submit" method="post"&gt;</code>
<code>&lt;input&gt;</code>	Defines an input control	type, name, value, placeholder, required, readonly, disabled, maxlength, list	<code>&lt;input type="text" name="username" placeholder="Enter your username" list="username-list"&gt;</code>
<code>&lt;label&gt;</code>	Defines a label for an <code>&lt;input&gt;</code> element	for	<code>&lt;label for="username"&gt;Username:&lt;/label&gt;</code>
<code>&lt;textarea&gt;</code>	Defines a multi-line text input control	name, rows, cols, placeholder	<code>&lt;textarea name="message" rows="4" cols="50"&gt;&lt;/textarea&gt;</code>
<code>&lt;button&gt;</code>	Defines a clickable button	type, name, value	<code>&lt;button type="submit"&gt;Submit&lt;/button&gt;</code>
<code>&lt;select&gt;</code>	Defines a drop-down list	name, multiple	<code>&lt;select name="country"&gt;&lt;option value="usa"&gt;USA&lt;/option&gt;&lt;option value="canada"&gt;Canada&lt;/option&gt;&lt;/select&gt;</code>
<code>&lt;option&gt;</code>	Defines an option in a drop-down list	value	<code>&lt;option value="usa"&gt;USA&lt;/option&gt;</code>

<b>&lt;fieldset&gt;</b>	<b>Groups related elements in a form</b>	-	<b>&lt;fieldset&gt;&lt;legend&gt;Personal Information&lt;/legend&gt;...&lt;/fieldset&gt;</b>
<b>&lt;legend&gt;</b>	<b>Defines a caption for a &lt;fieldset&gt; element</b>	-	<b>&lt;legend&gt;Personal Information&lt;/legend&gt;</b>
<b>&lt;datalist&gt;</b>	<b>Contains a list of pre-defined options for an &lt;input&gt; element</b>	id	<b>&lt;datalist id="username-list"&gt;&lt;option value="admin"&gt;&lt;option value="user"&gt;&lt;/datalist&gt;</b>
<b>&lt;output&gt;</b>	<b>Represents the result of a calculation or user action</b>	name, for	<b>&lt;output id="result" name="result"&gt;&lt;/output&gt;</b>

## HTML Form & Semantic HTML

1. **<form>**: Defines an HTML form for user input.
2. **<input>**: Defines an input control for various types of data entry.
3. **<label>**: Defines a label for an `` element.
4. **<textarea>**: Defines a multi-line text input control.
5. **<button>**: Defines a clickable button.
6. **<select>**: Defines a drop-down list.
7. **<option>**: Defines an option in a `` or `` element.
8. **<fieldset>**: Groups related elements in a form.
9. **<legend>**: Defines a caption for a `` element.
10. **<datalist>**: Contains a list of predefined options for an `` element.
11. **<output>**: Represents the result of a calculation or user action.

HTML Form & Semantic HTML

Element	Description	Example
<div>	Defines a generic container for grouping and styling content.	<div class="container">Content</div>
<aside>	Represents content indirectly related to the main content (e.g., a sidebar).	<aside><p>Related links</p></aside>
<section>	Represents a thematic grouping of content, typically with a heading.	<section><h2>About Us</h2><p>Content</p></section>
<footer>	Represents the footer of a section or page, often containing author information, copyright, etc.	<footer><p>© 2024 Company Name</p></footer>
<main>	Represents the dominant content of the <body>, excluding headers, footers, and sidebars.	<main><h1>Main Content</h1></main>
<figure>	Represents any content that is referenced from the main content, such as images, diagrams, or code snippets.	<figure><figcaption>Image caption</figcaption></figure>
<figcaption>	Provides a caption or legend for a <figure> element.	<figcaption>Image caption</figcaption>
Accessibility	Ensures content is accessible to all users, including those using assistive technologies.	Use of semantic elements like <header>, <nav>, <main>, <footer> improves accessibility.

HTML Form & Semantic HTML

1. **<div>**: Defines a generic container for grouping and styling content.
2. **<aside>**: Represents content that is indirectly related to the main content, such as a sidebar or additional information.
3. **<section>**: Represents a thematic grouping of content, usually with a heading, which is a distinct section of a document.
4. **<footer>**: Represents the footer of a section or page, often containing information like author details, copyright, or contact info.
5. **<main>**: Represents the dominant content of the ``<body>``, excluding headers, footers, and sidebars.
6. **<figure>**: Represents content that is referenced from the main content, such as images, diagrams, or illustrations.
7. **<figcaption>**: Provides a caption or description for a ``<figure>`` element.
8. **Accessibility**: Refers to practices that ensure web content is usable by all people, including those with disabilities, often involving semantic HTML to improve navigation and understanding.

## Lesson 4: CSS Selector & Keyframes

### CSS Selectors

**CSS selectors are patterns used to select and style elements in an HTML document. They determine which elements the CSS rules apply to. By using selectors, you can target specific elements based on their type, class, ID, attributes, and their relationship to other elements.**

#### Types of CSS Selectors

1. **Universal Selector**: Selects all elements in the document.
2. **Type Selector**: Selects all elements of a specified type (e.g., all ``<p>`` elements).
3. **Class Selector**: Selects all elements with a specified class, indicated by a dot (`.`) before the class name.
4. **ID Selector**: Selects a single element with a specified ID, indicated by a hash (`#`) before the ID name.
5. **Grouping Selector**: Selects multiple specified elements, allowing you to apply the same styles to different elements.
6. **Descendant Selector**: Selects elements that are descendants of a specified element, regardless of their depth.
7. **Child Selector**: Selects direct child elements of a specified element.

8. **Adjacent Sibling Selector:** Selects the next sibling element immediately following a specified element.
9. **General Sibling Selector:** Selects all sibling elements that follow a specified element.
10. **Pseudo-class Selector:** Selects elements based on their state or position, such as when an element is hovered over.
11. **Pseudo-element Selector:** Selects a specific part of an element, such as the first line or first letter.

The primary purpose of CSS selectors is to apply styles to specific elements, allowing for greater control over the design and layout of web pages.

### CSS Selectors

Selector Type	Selector Syntax	Definition	Example	CSS Code
Universal	*	Selects all elements.	All elements in the document.	<code>* { margin: 0; padding: 0; }</code>
Type	element	Selects all elements of a specified type.	All <code>&lt;p&gt;</code> elements.	<code>p { color: blue; }</code>
Class	.class	Selects all elements with a specified class.	All elements with the class "active".	<code>.active { font-weight: bold; }</code>
ID	#id	Selects a single element with a specified ID.	The element with ID "header".	<code>#header { background: grey; }</code>
Grouping	element1, element2	Selects multiple specified elements.	All <code>&lt;h1&gt;</code> and <code>&lt;h2&gt;</code> elements.	<code>h1, h2 { margin: 10px; }</code>
Descendant	element element	Selects descendant elements.	All <code>&lt;span&gt;</code> elements inside <code>&lt;div&gt;</code> .	<code>div span { color: red; }</code>
Child	element > element	Selects direct child elements.	All <code>&lt;li&gt;</code> elements that are direct children of <code>&lt;ul&gt;</code> .	<code>ul &gt; li { list-style: none; }</code>

<b>Adjacent Sibling</b>	<b>element + element</b>	<b>Selects the next sibling element.</b>	<b>The first &lt;p&gt; after an &lt;h2&gt;.</b>	<b>h2 + p { margin-top: 0; }</b>
<b>General Sibling</b>	<b>element ~ element</b>	<b>Selects all sibling elements after a specified element.</b>	<b>All &lt;p&gt; elements after an &lt;h2&gt;.</b>	<b>h2 ~ p { color: green; }</b>
<b>Pseudo-class</b>	<b>:hover</b>	<b>Selects an element when hovered over.</b>	<b>Changes color on hover.</b>	<b>a:hover { color: orange; }</b>
<b>Pseudo-class</b>	<b>:not(selector)</b>	<b>Selects all elements that do not match a specified selector.</b>	<b>All &lt;p&gt; elements except those with class "highlight".</b>	<b>p:not(.highlight) { color: grey; }</b>

### CSS Keyframes

- CSS keyframes are a way to define animations in CSS. They allow you to create complex animations by specifying styles at various points (or "keyframes") in the animation sequence. This enables smooth transitions between different states of an element.

### Key Concepts

1. **@keyframes Rule:** This is the syntax used to define the animation. Inside the rule, you specify the styles at various points in the animation.
2. **Percentage Values:** You can define styles at specific percentages (from 0% to 100%), representing the start and end of the animation, as well as any intermediate steps.
3. **Animation Properties:** Once you define keyframes, you can use properties like animation-name, animation-duration, and animation-timing-function to apply the animation to elements.

### CSS Keyframes

<b>Property</b>	<b>Definition</b>	<b>Example</b>	<b>CSS Code</b>
<b>@keyframes</b>	Defines the keyframes for an animation.	Used to create animations for elements.	@keyframes slide { ... }
<b>from</b>	Represents the starting point of the animation.	The initial state of the animation.	from { transform: translateX(0); }

<b>to</b>	Represents the ending point of the animation.	The final state of the animation.	to { transform: translateX(100px); }
<b>percentage</b>	Defines intermediate points in the animation.	Allows for multiple stages in the animation.	0% { opacity: 0; } 100% { opacity: 1; }
<b>animation-name</b>	Specifies the name of the @keyframes to be applied.	Links the animation to an element.	animation-name: slide;
<b>animation-duration</b>	Sets the duration of the animation.	Defines how long the animation takes to complete.	animation-duration: 2s;
<b>animation-timing-function</b>	Defines the speed curve of the animation.	Controls the pace of the animation.	animation-timing-function: ease-in-out;
<b>animation-delay</b>	Sets a delay before the animation starts.	Specifies a wait time before the animation.	animation-delay: 1s;
<b>animation-iteration-count</b>	Defines how many times the animation will repeat.	Can be set to a specific number or infinite.	animation-iteration-count: infinite;
<b>animation-direction</b>	Sets whether the animation should alternate direction.	Controls the flow of the animation.	animation-direction: alternate;
<b>animation-fill-mode</b>	Defines how styles are applied before/after the animation.	Determines whether styles are retained.	animation-fill-mode: forwards;