

The Foundation of the Web: An Introduction to HTML

Welcome to Week 1 of our journey into web development! The World Wide Web is the most pervasive and influential communication platform in human history. Understanding how it works starts with its foundational language: HTML. This presentation will cover the fundamental concepts of the web, differentiate between static and dynamic content, and provide a deep dive into the Hypertext Markup Language (HTML).

Our Core Mission

To provide beginner web development students with clear, instructional, and detailed learning material on the core concepts of the web and HTML structure.

Key Takeaways

Understand the role of the WWW, distinguish static from dynamic websites, and master the basic syntax and structure of an HTML document.

Understanding the World Wide Web and Website Types

What is the World Wide Web (WWW)?

The World Wide Web (WWW) is an information space where documents and other web resources are identified by URLs, interconnected by hypertext links, and can be accessed via the Internet. It is a system of interconnected, publicly accessible web pages and websites.

Static Websites

A static website is composed of fixed, pre-built files (like HTML, CSS, and JavaScript) that are delivered to the user's browser exactly as they are stored. The content does not change based on user input or server-side processing. Think of it as an online brochure.

- Simple and fast to load.
- Easy to host and maintain.
- Content remains the same for every visitor.

Dynamic Websites

A dynamic website generates content in real-time when a user requests it. This content can change based on various factors, such as the time of day, user location, database information, or previous user interactions. This requires server-side processing using languages like Python, PHP, or Node.js.

- More complex functionality (e.g., e-commerce, user accounts).
- Content is personalized and interactive.
- Relies on databases and server-side logic.



1

Fixed Content

Static sites are perfect for portfolios or simple informational pages where content rarely changes.

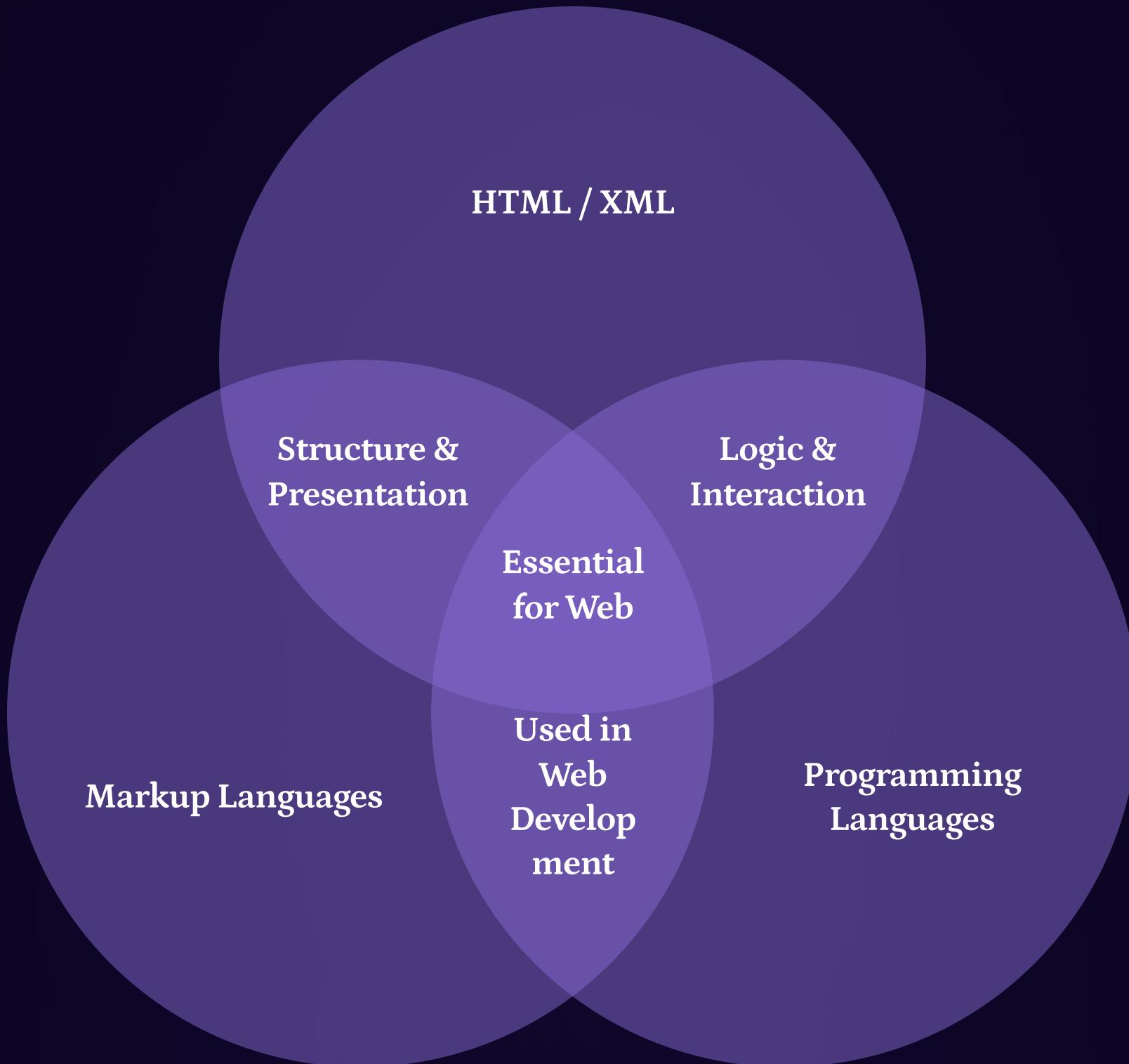
2

Interactive Content

Dynamic sites power applications like social media, blogs, and online stores.

Markup Languages vs. Programming Languages

It is crucial to understand the difference between markup and programming languages, as HTML falls into the former category. Both are essential for web development but serve distinct purposes.



Markup Languages (e.g., HTML)

Markup languages are designed to annotate text in a way that is syntactically distinguishable from the text. They define the structure and presentation of documents. HTML dictates **what** content is on the page (headings, paragraphs, images).

Key Role: Structural foundation and presentation.

Programming Languages (e.g., JavaScript)

Programming languages consist of a vocabulary and set of grammatical rules for instructing a computer to perform specific tasks. They handle logic, computation, and complex behavior. JavaScript dictates **how** the page behaves (user interactions, data fetching).

Key Role: Functionality, logic, and interactivity.

Hypertext Markup Language (HTML): Tags and Attributes

Definition of HTML

HTML is the standard markup language for documents designed to be displayed in a web browser. It is the backbone of any web page, providing the structure and meaning (semantics) to the content.

The "Hypertext" part refers to the non-linear way of organizing information (links), and "Markup Language" refers to the use of tags to define elements.

HTML Elements: Tags and Attributes

1 Tags

Tags are the fundamental components of HTML. They usually come in pairs: an opening tag (e.g., `<p>`) and a closing tag (e.g., `</p>`). Tags contain the content and define the element's purpose, such as a paragraph, heading, or list.

Example: `<h1>This is a main title</h1>`

2 Attributes

Attributes provide additional information about HTML elements, modifying their appearance or behavior. Attributes are always specified in the opening tag and usually come in name/value pairs (e.g., `name="value"`).

Example: `Link` (where `href` is the attribute)

- Remember: All HTML elements are essentially defined by a tag, and most tags can accept attributes to customize their functionality.

HTML Standards, Structure, and Tools

HTML Standards and Enhancements

HTML standards are maintained by the World Wide Web Consortium (W3C). The current stable version, HTML5, introduced significant enhancements, including new semantic tags (like `<article>`, `<nav>`), multimedia support (audio and video tags), and APIs for complex features like drag-and-drop and local storage.

HTML File Structure

Every HTML document must follow a basic, hierarchical structure to be valid and interpreted correctly by the browser:

- The `<!DOCTYPE html>` declaration defines the document type.
- The `<html>` element is the root element.
- The `<head>` contains meta-information (title, character set, etc.).
- The `<body>` contains the visible page content.

Web Pages File Extensions

HTML files must be saved with either the `.html` or `.htm` extension. Using `.html` is the standard and preferred convention.

Creating Your First HTML Document

The simplest way to create an HTML document is using a basic text editor and saving it with the correct file extension.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>My First Webpage</title>
</head>
<body>
  <h1>Hello World!</h1>
  <p>This is my first paragraph of web content.</p>
</body>
</html>
```

Recommended HTML Editors

While Notepad or TextEdit work, specialized editors offer syntax highlighting, auto-completion, and debugging tools to boost efficiency:



Structuring Content: Paragraphs, Headings, and Formatting Text

Paragraphs and Headings: Defining Hierarchy

HTML uses `<p>` tags for paragraphs and six levels of heading tags (`<h1>` through `<h6>`) to structure text and establish a clear information hierarchy. `<h1>` is the most important (usually the main title), and `<h6>` is the least important. You should only use one `<h1>` per page.

Code Example: Headings and Paragraphs

```
<h1>Main Article Title (Most Important)</h1>
<p>This is the introductory paragraph for the section.</p>
<h2>Section Subheading</h2>
<p>This is a second paragraph with more details.</p>
```

Formatting Text for Emphasis

HTML provides tags to format text, conveying semantic meaning to the browser and accessibility tools:

1

Bold & Strong

`` (bold) for visual style; `` for important content.

2

Italic & Emphasis

`<i>` (italic) for alternative voice; `` for emphasized content.

3

Underline

`<u>` can be used, but use sparingly as it can be confused with links.

Formatting Example

```
<p>This text is <strong>very important</strong> and should be emphasized.</p>
<p>We should <em>always</em> follow semantic rules.</p>
```

Linking and Imagery: Making the Web Hypertext



The Anchor Tag (`<a>`)

The anchor tag is the core of "Hypertext." It creates links to other web pages or resources. The critical attribute is `href` (Hypertext Reference), which specifies the destination URL.

A common optional attribute is `target="_blank"`, which opens the link in a new browser tab.

```
<a href="https://www.w3.org" target="_blank">Visit W3C</a>  
<a href="about.html">About Us Page</a>
```



The Image Tag (``)

The image tag is used to embed an image into an HTML document. Unlike most tags, `` is a self-closing tag (it doesn't have a separate closing tag).

It requires two key attributes: `src` (Source) for the file path, and `alt` (Alternate Text) for accessibility and when the image fails to load.

```

```



Data Presentation: Tables

HTML tables are used to display data in a row-and-column format. They are specifically for tabular data, not for general page layout (which should be handled by CSS).

Basic Table Tags

Tag	Description
<table>	Defines the overall table structure.
<tr>	Defines a Table Row.
<td>	Defines a Table Data (a standard cell).
<th>	Defines a Table Header cell (automatically bold and centered).

Code Example: Simple Table

```
<table border="1">
<tr>
  <th>Product</th>
  <th>Price</th>
</tr>
<tr>
  <td>Laptop</td>
  <td>$1200</td>
</tr>
<tr>
  <td>Monitor</td>
  <td>$300</td>
</tr>
</table>
```

Best Practice: Use `<thead>`, `<tbody>`, and `<tfoot>` for large tables to clearly separate the different parts of the table structure, improving accessibility and maintainability.

User Interaction: Forms and Input

HTML Forms are essential for dynamic websites, allowing users to submit data to a server (e.g., login information, comments, search queries). The `<form>` element acts as a container for all input elements.



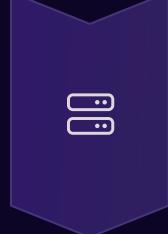
The `<form>` Tag

The key attributes are `action` (where to send the data) and `method` (GET or POST, defining how the data is sent).



The `<input>` Tag

The `<input>` tag is versatile, with the `type` attribute defining its function (e.g., `text`, `password`, `checkbox`, `submit`).



Sending Data

When the user clicks a submit button, the data entered into the form fields is bundled up and sent to the server script specified in the `action` attribute.

Forms Code Example

```
<form action="/submit-data.php" method="POST">
  <label for="username">Username:</label>
  <input type="text" id="username" name="username"><br><br>
  <label for="password">Password:</label>
  <input type="password" id="password" name="password"><br><br>
  <input type="submit" value="Login">
</form>
```

Metadata and Review: The Head of the Document

Meta Tags: Data About Data

Meta tags are placed inside the `<head>` section and provide metadata about the HTML document. This information is typically not visible to the user but is essential for browsers, search engines, and other web services.

Character Set

`<meta charset="UTF-8">` - Defines the character encoding for the document, ensuring proper display of international characters.

Description

`<meta name="description" content="...">` - A short summary of the content, often used by search engines.

Viewport

`<meta name="viewport" content="...">` - Crucial for responsive design, instructing the browser how to control the page dimensions and scaling.

Week 1 Key Takeaways

Structure First

HTML provides the structure (`<h1>`, `<p>`, `<table>`) for all web content. Without structure, there is no web page.

Semantics Matter

Use the correct tags (e.g., `` vs. ``) to give meaning to your content, aiding accessibility and search engines.

Tags & Attributes

Tags define elements, and attributes (like `href` or `src`) modify or provide supplemental information about them.

Next week, we will move on to Cascading Style Sheets (CSS) to transform this structure into a beautiful, styled, and responsive layout!