

Web Application Development

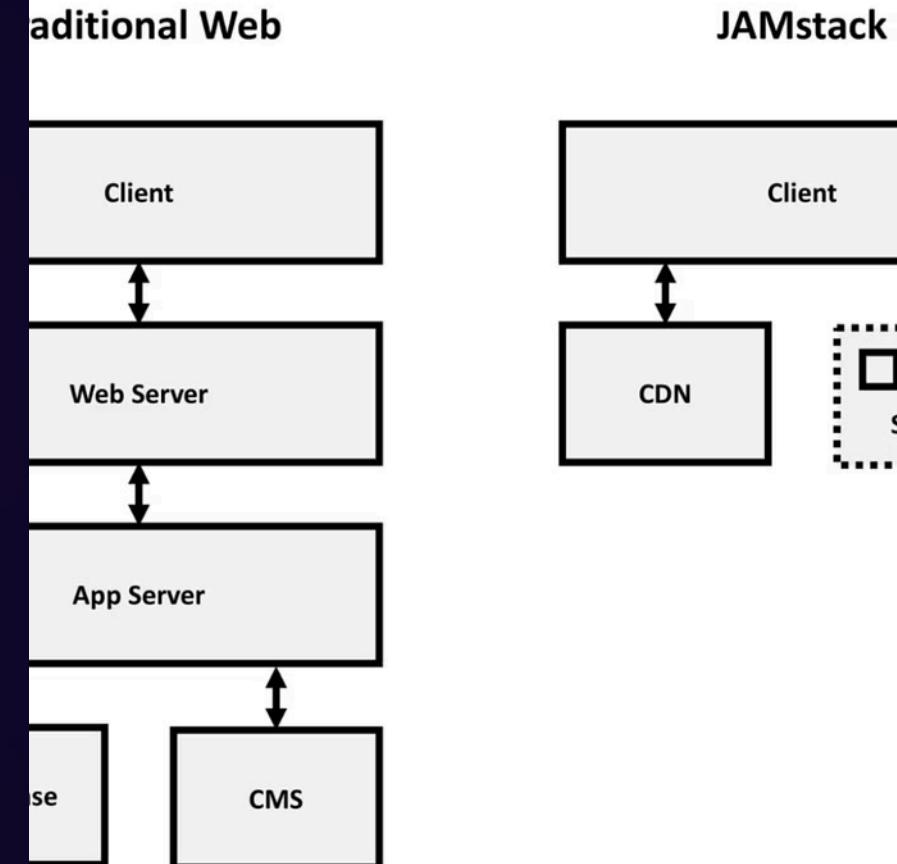
Week 1: Getting Started with HTML

Welcome to the exciting world of web development! In this first week, you'll learn the foundation of every website you've ever visited: HTML. This presentation will guide you through the essential building blocks that power the internet.

HTML is the backbone of every website. Combined with CSS for styling and JavaScript for interactivity, HTML creates the structure that holds everything together. By the end of this week, you'll understand how to create your first web pages and structure content properly.

How it works

Traditional Web



The World Wide Web: Why It Matters

The World Wide Web, invented by Tim Berners-Lee in 1989, revolutionized how we share and access information globally. Today, understanding web development is one of the most valuable skills you can learn.

Why Learn Web Development?

- Create opportunities in a booming tech industry
- Build projects that impact millions of people
- Develop skills that are always in demand
- Express creativity through interactive experiences
- Earn competitive salaries as you grow

The Internet Today

Over 5.3 billion people use the internet. Every single website they visit runs on the same core technologies you're about to learn. From social media platforms to streaming services, all of it starts with HTML, CSS, and JavaScript working together.

Web development spans everything from building simple blogs to complex applications that process millions of transactions daily.

Static vs. Dynamic Websites

Before diving into HTML, it's crucial to understand that websites come in two main flavors: static and dynamic. Each serves different purposes and requires different technologies.

Static Websites

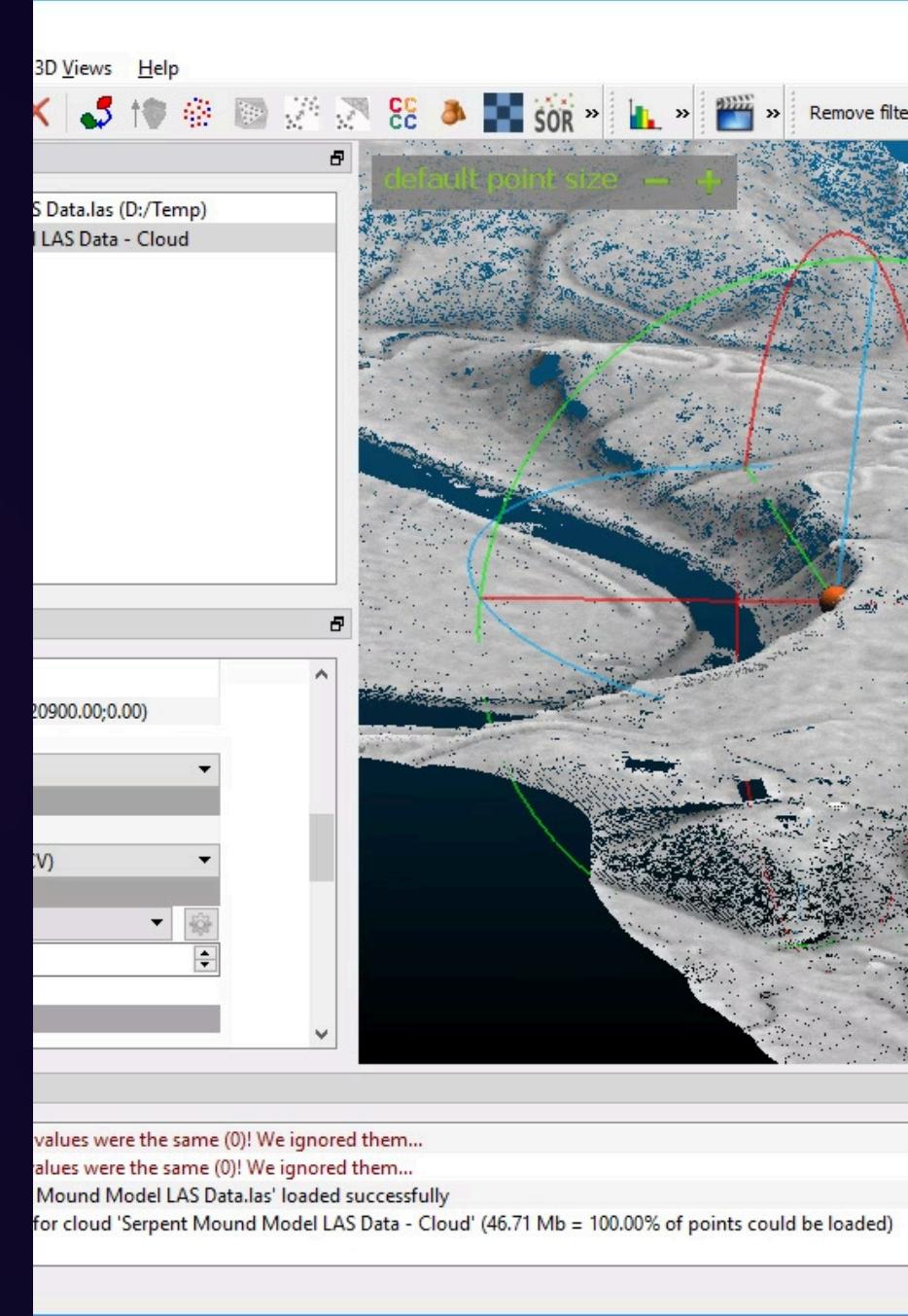
Display the same content to every visitor. Think of them as digital brochures—the information doesn't change based on who's viewing it or what they do. Every visitor sees identical HTML files.

Examples: Portfolio sites, company information pages, blogs with no user interactions

Dynamic Websites

Generate unique content for each visitor based on their actions, preferences, or data. The content changes in real-time. These sites use programming languages to process information and respond to user input.

Examples: Facebook, Gmail, Netflix, Instagram, online banking



Markup Languages vs. Programming Languages

Understanding the difference between markup and programming languages is fundamental. They serve completely different purposes, though they often work together.

Markup Languages

HTML (Hypertext Markup Language) is a markup language. It uses tags to describe and structure content. Think of markup like labeling a box: you're telling the browser "this is a heading," "this is a paragraph," "this is an image."

Key point: Markup languages describe what content IS, not how to make things happen.

Other markup languages: XML, XHTML, SVG

Programming Languages

JavaScript, Python, Java are programming languages. They contain logic, make decisions, perform calculations, and control behavior. They can respond to user actions and create dynamic experiences.

Key point: Programming languages tell computers how to DO things.

How they work together: HTML structures the page, JavaScript makes it interactive and responsive.

Introduction to HTML: Tags & Attributes

HTML stands for Hypertext Markup Language. At its core, HTML uses a system of tags—special keywords wrapped in angle brackets—to tell web browsers how to display content.

What Are Tags?

Tags are instructions that tell your browser how to display content. Most tags come in pairs: an opening tag and a closing tag. The content goes between them.

```
<tagname>Content goes here</tagname>
```

What Are Attributes?

Attributes provide additional information about tags. They're always placed in the opening tag and modify how that element behaves or looks.

Attributes follow the pattern: name="value"

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

In this example, href tells the browser where the link goes, and target="_blank" tells it to open in a new tab. These attributes give the tag more information and control.

HTML Standards & Document Structure

Web browsers need to understand HTML consistently. That's why the World Wide Web Consortium (W3C) maintains standards. Following these standards ensures your websites work across all browsers and devices.

01

DOCTYPE Declaration

The very first line tells the browser which version of HTML you're using.

For modern web development, use: `<!DOCTYPE html>`

03

Head Section

The `<head>` contains metadata about your page: title, character encoding, links to stylesheets, and information the browser needs but users don't see directly.

02

The HTML Root Element

Everything else goes inside the `<html>` tag. This wraps your entire document.

04

Body Section

Everything users see goes in `<body>`: text, images, links, forms, and all visible content.

```
<!DOCTYPE html>
<html>
  <head>
    <title>My First Webpage</title>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

Essential HTML Elements: Text Formatting & Structure

HTML provides semantic tags that give meaning to your content while making it easy for browsers and screen readers to understand your page structure. Let's explore the most fundamental elements you'll use constantly.

Headings

`<h1>` through `<h6>` organize content hierarchically. Use one `<h1>` per page as your main title. Proper heading structure helps with SEO and accessibility.

```
<h1>Main Title</h1>
<h2>Subheading</h2>
```

Paragraphs

The `<p>` tag defines paragraphs. Browsers automatically add space before and after paragraphs. This is how most text content is structured on web pages.

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

Text Formatting

Use `` for important text (displays bold), `` for emphasis (displays italic). These are better than `` and `<i>` because they convey meaning, not just appearance.

```
<strong>Important</strong> and
<em>emphasized</em>
```

Links, Images, Tables & Forms

These four elements are the workhorses of web development. They enable navigation, visual content, organized data, and user interaction. Understanding them is essential for building functional websites.

Links: Connecting the Web

```
<a href="https://example.com" target="_blank">Click here</a>
```

The `href` attribute specifies the destination. `target="_blank"` opens in a new tab.

Images: Adding Visual Content

```

```

The `alt` attribute describes the image—crucial for accessibility and when images fail to load.

Tables: Organizing Data

```
<table>
  <tr><th>Name</th><th>Age</th></tr>
  <tr><td>Sarah</td><td>21</td></tr>
</table>
```

`<tr>` = row, `<th>` = header cell, `<td>` = data cell.

Forms: Collecting User Input

```
<form>
  <input type="text" name="username" placeholder="Enter your name">
  <input type="submit" value="Submit">
</form>
```

Forms let users send data back to servers. Different input types handle different data: text, email, password, checkboxes, radio buttons, and more.

Meta Tags & SEO Essentials

Meta tags in the `<head>` section provide important information to browsers and search engines. They don't display on the page but deeply impact how your site functions and how people find it.



Character Encoding

`<meta charset="UTF-8">` tells the browser how to read your text. UTF-8 supports all languages and symbols worldwide.



Viewport

`<meta name="viewport" content="width=device-width, initial-scale=1">` makes pages responsive on mobile devices by controlling how content scales.



SEO Description

`<meta name="description" content="...">` provides a summary for search engines. This text often appears in search results beneath your page title.



Keywords

`<meta name="keywords" content="...">` helps search engines understand your page topic, though its impact on rankings has diminished over time.

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="description" content="Learn web development basics">
  <title>Web Development Course</title>
</head>
```

Your HTML Journey Begins

Congratulations! You've learned the foundation of web development. HTML is your starting point for building anything on the web. As you continue, you'll combine HTML with CSS for beautiful styling and JavaScript for interactivity.

Master HTML Basics

Practice writing semantic HTML with proper structure. Build simple pages with headings, paragraphs, links, and images. Focus on clean, organized code.

Add JavaScript Interactivity

JavaScript brings your pages to life with dynamic behavior, animations, form validation, and user interactions. This is where your websites become truly powerful.

Learn CSS Styling

Once HTML gives your content structure, CSS will make it beautiful. Learn colors, fonts, layouts, and responsive design to create professional-looking websites.

Build Real Projects

Apply your knowledge by creating actual projects: personal portfolios, small business websites, or simple applications. Real-world practice cements your skills.

Remember: Every expert developer started exactly where you are right now. Be patient with yourself, practice consistently, and don't hesitate to experiment. The web development community is incredibly supportive. Welcome to an exciting journey of creating experiences millions of people will use and enjoy!