07-16 Class Session: Web Applications, HTML Structure, and CSS Specificity

> Date & Time: 2025-07-16 13:37:28
> Location: [Insert Location]
> Course Name: [Enter Course Name]

Keywords

`Document Object Model (DOM)` `CSS Specificity` `HTML Structure`

Key Learnings

1. Difference between Desktop Applications and Web Applications: Explains the fundamental difference between desktop applications and web applications, focusing on delivery, installation, and usage.
2. HTTP (Hypertext Transfer Protocol): Covers the role, history, and function of HTTP as the foundational protocol for web communication.
3. Web Browsers and Developer Tools: Discusses the use of different browsers and their developer tools to observe HTTP requests and responses.
4. Essential Components of a Web Application: Identifies the three core components required for a web application to function.
5. HTML (Hypertext Markup Language): Introduces HTML as the language used to structure and describe the contents of web pages.
6. HTML Structure: Head and Body: Explains the division of an HTML document into head and body sections, and the purpose of each.
7. HTML Tags and Their Functions: Details common HTML tags and their purposes in structuring web content.
8. HTML Structure and Semantic Grouping: Understanding how HTML elements are grouped semantically using tags like div, and how these groupings affect the structure and rendering of a web page.
9. Div and Span Elements in HTML: Understanding the use and differences between div and span elements in HTML, including their nesting rules and semantic purposes.
10. Document Object Model (DOM): Understanding the Document Object Model (DOM) as the key data structure representing the structure of a web page, and its role in web development.

11. CSS (Cascading Style Sheets): Introduction to CSS, its purpose in separating content from presentation, and how CSS rules are applied to HTML elements.
12. CSS Rules and the DOM Tree: CSS rules are applied to the DOM (Document Object Model) tree, and the formatting is inherited by all descendants unless overridden by more specific rules.
13. Multiple CSS Rules and Inheritance: When multiple CSS rules are present, inheritance and specificity determine which styles are applied to elements and their descendants.
14. CSS Selectors: Tag, Class, and ID: CSS selectors can target elements by tag, class, or ID. IDs are unique, classes can be shared, and tags refer to element types.
15. CSS Specificity and Precedence: When multiple CSS rules could apply to an element, the most specific rule takes precedence: ID > class > tag. Direct rules override inherited rules.
16. Cascading and Overriding in CSS: CSS rules cascade, meaning later or more specific rules override earlier or less specific ones. Directly applied rules override inherited ones.
17. CSS Activity and Assignment Instructions: Students are assigned an activity involving CSS selectors and rules, with instructions for team formation and assignment submission.

Explanations

1. Difference between Desktop Applications and Web Applications

- **Key Points**
  - Desktop applications require installation and are run locally on a user's machine.
  - Web applications are delivered via the cloud and accessed through a web browser, requiring no upfront purchase or installation.
  - Web applications depend on a network connection to access resources.
- **Explanation**
  The class began by contrasting desktop and web applications. Desktop applications are installed and run on a local computer, while web applications are accessed over the internet via a browser and are delivered by the cloud. This distinction sets the stage for understanding the underlying technologies that enable web applications.
- **Examples of Web Applications Used Daily**
  > Students mentioned YouTube, Google Calendar, Google, and a web application called 'National' as examples of web applications they use daily.

  i. These examples illustrate the variety of web applications available.

ii. All these applications are accessed via a browser and delivered over the network, not installed locally.

2. HTTP (Hypertext Transfer Protocol)

- **Key Points**
  - HTTP stands for Hypertext Transfer Protocol.
  - Invented by Tim Berners-Lee in 1998 or 1999.
  - Standardized by the Internet Engineering Task Force.
  - HTTP defines how clients (browsers) and servers exchange information.
  - Consists of requests (from client) and responses (from server).
  - HTTP is the protocol indicated by http:// or https:// in web addresses.
- **Explanation**
  HTTP is the protocol that enables communication between a client and a server. When a user enters a URL, the browser sends an HTTP request to the server, which responds with the requested resource (such as an HTML page). HTTP was chosen over other protocols like Archie and Gopher due to its simplicity. The protocol is standardized and widely adopted.
- **HTTP Request and Response Example**

  > When a user goes to a URL like google.com, the browser sends a GET request (e.g., GET /index.html) to the server. The server responds with the HTML page and specifies the content type and protocol in the response.

  i. The client initiates a connection and sends a request.
  ii. The server processes the request and sends back a response, typically an HTML page.
  iii. The browser renders the response using the Document Object Model (DOM).

3. Web Browsers and Developer Tools

- **Key Points**
  - Popular browsers include Chrome, Brave, Firefox, Edge, and Safari.
  - Developer tools in browsers allow users to inspect network activity and see HTTP requests and responses.
  - There may be subtle differences in how browsers render pages or format messages.
- **Explanation**
  The instructor demonstrated how to use browser developer tools (specifically in Chrome) to observe HTTP requests and responses when loading a web page like google.com. Students were encouraged to try this in their own browsers, noting that the exact details may vary between browsers.
- **Viewing Network Activity in Chrome Developer Tools**

> The instructor opened google.com in Chrome, accessed the developer tools, and showed the network tab where multiple HTTP requests and responses could be seen as blue bars.

i. Each network request represents a resource being loaded by the browser.
ii. Students can use this feature to see what data is being sent and received.

4. Essential Components of a Web Application

- **Key Points**
  - A server that serves the web page.
  - A client (browser) that receives and displays the content.
  - An HTTP connection that enables client-server interaction.
- **Explanation**
  For a web application to work, there must be a server to provide content, a client to request and display it, and an HTTP connection to facilitate communication. Other elements like JavaScript and data handling are optional but common in modern web applications.

5. HTML (Hypertext Markup Language)

- **Key Points**
  - HTML stands for Hypertext Markup Language.
  - Invented by Tim Berners-Lee.
  - Describes the structure and content of a web page using tags.
  - HTML is hierarchical and organized using tags enclosed in angular brackets.
  - Tags define elements like headings, images, links, paragraphs, lists, tables, forms, and divisions (divs).
    Explanation
    HTML is the foundational language for web pages, using tags to define the structure and content. Tags like