# COMP 266: Unit 2

Hyper Text Markup Language

Richard Sullivan Andison

2026, January 19th.

## Unit Review

This unit was an introduction to the HyperText Markup Language (HTML) for structuring the core elements of a website (metadata, content structure, element classification/identification, page linking, and script-inclusion (CSS and Javascript)). Research included a brief history lesson of markup languages and their subdivisions (procedural, descriptive, and presentational), exploration into the history of HTML's development, and exploring proper HTML syntax via viewing well-constructed HTML sites, viewing tutorials, and visiting reference sites for proper practices in constructing HTML structures [4]-[5]. The methods and resources gathered during research were applied to the 3DS library site; some examples included adding comments, labeling elements with class and unique identifiers, using divs to create logical segments of the website, and setting site metadata to ensure the site is displayed properly on different systems (desktop and New 3DS XL). Other related resources were used in structuring the file systems for the site to fit currently accepted practices [2].

During this development phase of the site, accomplishing tasks typically fell into a particular development cycle; each element of the site was worked on in layers (starting from background to foreground) with a cycle of implementing HTML, CSS, and Javascript elements. In terms of individual page development, an initial homescreen was created and copied; the copy was reduced to basic features and used as a template for other pages; page design templates were divided into three

categories: navigational, informational, and reading. At this stage, CSS and JavaScript elements were used sparingly as basic structure and functional navigation was currently the main priority. Some functionality tests included attempting to play audio (this was unsuccessful) and implementing images (this was successful with GIFs) within the “about.html” page)_. Some limitations were also discovered in implementing GIFs; there is a noticeable slowdown in playback speed with each additional gif added on the New 3DS console browser. For development, the KATE IDE was used for editing and Git repository management while Firefox’s browser tools were used as responsive testing grounds to modify and adjust the site’s CSS elements.

While it was instructed for students to only focus on the HTML of the website at this stage, I did find myself incorporating CSS and Javascript functionalities to my site before the entire HTML structure was completed; this was mostly because I started with a cloned repo for an optimized New 3DS website template [7]. For CSS, background elements and container structures were implemented; areas of focus involved learning correct CSS practices and understanding CSS classes and identification. For Javascript, I learned some basic understanding of how to select elements and add event listeners in order to create basic logic to change the top screen’s text when a menu item is selected; additionally, functionality for detecting 3DS systems/Desktops was implemented to change site settings accordingly. Finally, a GitHub repo for live-testing and tracking significant changes has been regularly updated and commented on to keep track of changes.

There were several challenges in the initial site development. To begin, creating a site framework that had proper proportions to the 3DS screen involved much trial and error (each update could take up to ten minutes to be reflected); some issues included setting proper proportions (especially with borders, margins, and learning general structural syntax, such as tag nesting, classes, and identification), and adjusting the metadata tags to prevent the site from scrolling or scaling [1]. Some major setbacks generally involved implementing a change in HTML structure that had to be reflected in all pages, such as changing tag identification or nesting structure (for example, the upper screen's text and images being moved within their own divs). Future implementations could include a more robust templating system (such as Django, Flask, or some other equivalent) to reduce the need to update every .html file when a change is needed.

In conclusion, This unit provided an overview in HTML and the necessary concepts, frameworks, and methodologies required to create a website that is structured correctly and easy to undertand. The underlying structure provided by HTML allows for the seemless inclusion of styling elements and logic (CSS and Javascript).

## Resources Used

[1] Crocsx, “Force page zoom at 100% with JS,” *Stack Overflow*, Jan. 13, 2014. https://stackoverflow.com/questions/21093570/force-page-zoom-at-100-with-js (accessed Jan 13th, 2026).

[2] GeeksforGeeks, “File and Folder Organization Best Practices for Web Development,” *GeeksforGeeks*, Jun. 16, 2025. https://www.geeksforgeeks.org/javascript/file-and-folder-organization-best-practices-for-web-development/

[3] MDN Contributors - HTML | MDN,” *Mozilla.org*, Oct. 14, 2025. https://developer.mozilla.org/en-US/docs/Web/HTML/Reference/Elements/meta/name/viewport (accessed Jan 15th, 2026).

[4] W3Schools, “HTML Tutorial,” W3schools.com, 2022. https://www.w3schools.com/html/default.asp

This resource was of massive assistance for quickly indexing specific HTML tags and functionalities throughout every stage of development so far.

[5] Wikipedia Contributors, “Markup language,” *Wikipedia*, Mar. 07, 2019. https://en.wikipedia.org/wiki/Markup_language (accessed Jan 12th, 2026).

[6] Wolfyxon, “GitHub - Wolfyxon/3ds-web-stuff: A collection of browser games and other stuff for the Nintendo 3DS,” *GitHub*, 2025. https://github.com/Wolfyxon/3ds-web-stuff (accessed Jan. 13Th, 2026).

[7] simbas, “GitHub - simbas/n3dsite: optimized website for 3DS,” *GitHub*, 2025. https://github.com/simbas/n3dsite