Unit 2: HTML

Content Area: **Computer Science**
Course(s):
Time Period: **Marking Period 1**
Length: **20 Days**
Status: **Published**

Summary

This unit will serve as the backbone for the course. It is in this unit that students will learn about a wide variety of HTML elements such as paragraphs, images, links, ordered and unordered lists, nested lists, font size, font (using font family tag), font color, tables, and using the inline style attribute for a wide variety of purposes. By the end of this unit, students should be able to create a basic one-page website using a text editor such as Notepad++, or any equivalent editor. The instructor may also choose to teach about spans and divs in this unit, or in unit 4, where it is currently placed. Lastly, as this is the unit where students learn how to incorporate images on their webpages, it is here that students should learn about copyright rules, how to search for images with a particular kind of copyright, and what the "Creative Commons License" is.

Revised Date: July 2025

P.4	Demonstrate creativity and innovation.
P.5	Utilize critical thinking to make sense of problems and persevere in solving them.
P.8	Use technology to enhance productivity increase collaboration and communicate effectively.
	The scalability and reliability of the Internet are enabled by the hierarchy and redundancy in networks. Network topology is determined by many characteristics.
	Network security depends on a combination of hardware, software, and practices that protect data while it is at rest, in transit, and in use. The needs of users and the sensitivity of data determine the level of security implemented. Advanced attacks take advantage of common security vulnerabilities.
	Engineering design evaluation, a process for determining how well a solution meets requirements, involves systematic comparisons between requirements, specifications, and constraints.
	Engineers use science, mathematics, and other disciplines to improve technology. Increased collaboration among engineers, scientists, and mathematicians can improve their work and designs. Technology, product, or system redesign can be more difficult than the original design.

Essential Questions/Enduring Understandings

Essential Questions:

- How does HTML allow us to make webpages?
- To what extent does "white space" matter in HTML?
- Why should "alt" and "title" attributes always be specified?
- How can we debug our code when it does not run as intended?
- Why should we adhere to copyright rules?

Enduring Understandings:

- The "ML" in HTML stands for "Markup Language." HTML is a coding language that allows us to "mark up" regular text and media so that we can style it and position it how we wish.
- In this particular language, white space does not effect how the code is interpreted. However, it is important to indent code properly and follow other conventional "rules," so that others who view your code can understand it more quickly. White space also helps to show relationships between different portions of code, and it allows people to more easily change a website's code.
- Screen-reading programs help the visually-impaired to interact with websites. These programs use the alt and title attributes to describe portions of a website to such a person. To make sure your website is as accessible as possible, it is important to include these attributes. The "alt" attribute is also beneficial since its text will display when an image does not load.
- Many text editors offer color coding for different parts of code. When the coloring is not correct, it indicates that there is a problem in your code somewhere in that area. A web designer can also try to "comment out" portions of code to help find out where the error might be. Sometimes, a second pair of eyes is also helpful. It is sometimes easier to spot someone else's mistakes than your own.
- While copyright should be adhered to because it is a "rule," it should primarily be observed so that the creator of content receives due recognition for their work.

Objectives

Students Will Know:

- what each of the following are:
    - elements
    - opening/start tag
    - closing/end tag
    - self-closing tags
    - attributes
    - values
    - parent/child terminology & nested elements

Students Will be Skilled at:

- opening an html file in a text editor and observing how changes affect the website when viewed in a browser.
- organizing code following traditional rules to improve code readability and to display relationships between elements.
- using a color wheel to select colors.
- using online HTML documentation to look up additional elements and how to implement elements discussed in the unit.
- using the elements presented in this unit to create a basic single-file website.

Learning Plan

- Show students how to open a text editor. Start with something simple like Notepad. Students should be able to open a text file, write some basic HTML, save it as an html file, and open the file in a browser. Students should be comfortable opening the HTML file in both a text editor and a browser as needed.
- Show students how to write a paragraph or two. The content is irrelevant. Then, allow students to guide the direction of the class:
    - Ex: "What do you want to learn how to do first?"
    - As the students are exposed to different elements, be sure to give students a chance to mark up their text accordingly.

- Eventually, students will ask you to show them most of the basics in the unit. For any unexplored elements, show the students how to use them.
- Be sure to emphasize proper use of white space. Check for this as you move around to assist students.
- When appropriate, switch students over to a more advanced text editor such as Notepad++. Highlight some of the functionality the new editor offers, such as color-coding.
- Teach students about copyright and the "Creative Commons License"
    - Students should be able to execute a Google Image Search by Usage Rights.

- At the end of the unit, students should complete an "About Me" project where they utilize their knowledge from the unit to create a one-page website about themselves.

Assessment

- Assessments

    - Formative: Daily assessments using examples from class notes and CodeHS.com, AP Classroom/Albert Checks for Understanding

    - Summative: Teacher-created assessments/projects and CodeHS Computer Science Projects, AP Classroom/Albert Unit Assessments

    - Benchmark: Check for understanding benchmark assessments on CodeHS, AP Classroom/Albert/Khan Academy Diagnostics

    - Alternative Assessments: Student-centered activities such as a doorbell coding project, game design projects, and other activities involving real world applications shown below:

        - Students complete an "About Me" project that uses all skils from the unit to make a single-page website. Students may have a "coding buddy" to whom they go for help if something is not working as intended. Suggested timeframe: 4-5 days.

Materials

- Core instructional materials: Core Book List
- Supplemental materials:
    - CodeHS (for remediation and differentiation as deemed appropriate)
    - Internet
    - Computers
    - Projection system for lecture
    - w3schools