

Using Discussion Boards to Teach CSS and Improve Canvas Course Layouts

Craig E. Shepherd, Andrea F. Hall, and Maxine D. Evans, University of Memphis

OVERVIEW

In this three-week lesson, situated in a five-week, shortened summer course, instructional design graduate students learn basic cascading stylesheet (CSS) skills to facilitate page layouts within learning management systems. During Week 1, basic HTML tags (e.g., heading levels, lists, paragraphs, images, divider, span, link tags) and tag attributes, including style (used for inline CSS in Canvas), are introduced. In Week 2, learners are introduced to CSS attributes: background color, font color, borders, margins, and padding. During Week 3, they learn about block position, float, clear, and z-index. Throughout the three weeks, learners leverage low-stakes discussion boards to share ideas and refine skills.

Topics: Coding, CSS, Higher Education, HTML, Learning Management System, Online Learning, Page Layout, Web Design

Time: Approximately one hour a week for three-weeks of content

MATERIALS

- Internet-enabled computer for each student
- Access to Canvas or another learning management system (LMS)
- Learning materials in an [exported course shell](#) (A Canvas imsc file that is compatible with many other LMSs)
- [HTML Intro video](#) (erschelshep, 2022a)
- [HTML Attributes video](#) (erschelshep, 2022c)
- [CSS Intro video](#) (erschelshep, 2022b)
- [CSS positioning video](#) (erschelshep, 2022d)
- Canvas HTML Editor Allowlist (Instructure, 2021)

CONTEXT-AT-A-GLANCE

Setting

This course is for instructional design graduate students in an urban, Southeastern, public university in the United States.

Modality

Asynchronous online

Class Structure

The five-week seminar course focuses on course structure and community development. It is delivered through Canvas, and students use free Canvas accounts to develop and share projects. Students are expected to login to the course daily.

Organizational Norms

The degree program is entirely online. This course is an elective for most learners, focusing on application of theory through curriculum development.

Learner Characteristics

All students were enrolled in an instructional design master's or doctoral program and came from business, PK-12, and higher education sectors. Some students had prior experience designing online courses. Few had prior HTML or CSS experience.

Instructor Characteristics

The instructor had moderate experience with web development and regularly used CSS to design course layouts in D2L Brightspace and Canvas LMSs.

Development Rationale

Many LMSs provide *What You See Is What You Get* (WYSIWYG) editors for basic design layouts. However, advanced layouts require basic knowledge of HTML and CSS. This training sought to introduce the basic knowledge of HTML and CSS to expand student abilities with webpage design.

SETUP

Instruction took place in an LMS. The imsc file is an exported Canvas course that includes two modules: One regarding basic HTML and another regarding CSS (see Figure 1). Each module contains content videos, web resources, and discussion boards to complete instruction. Educators can use a [free Canvas account](#) (Instructure, n.d.-a) if they desire. Instructure (n.d.-b) provides [guidance to import content into Canvas](#). However, this article also describes the resources used, if importing content into an LMS is not possible.

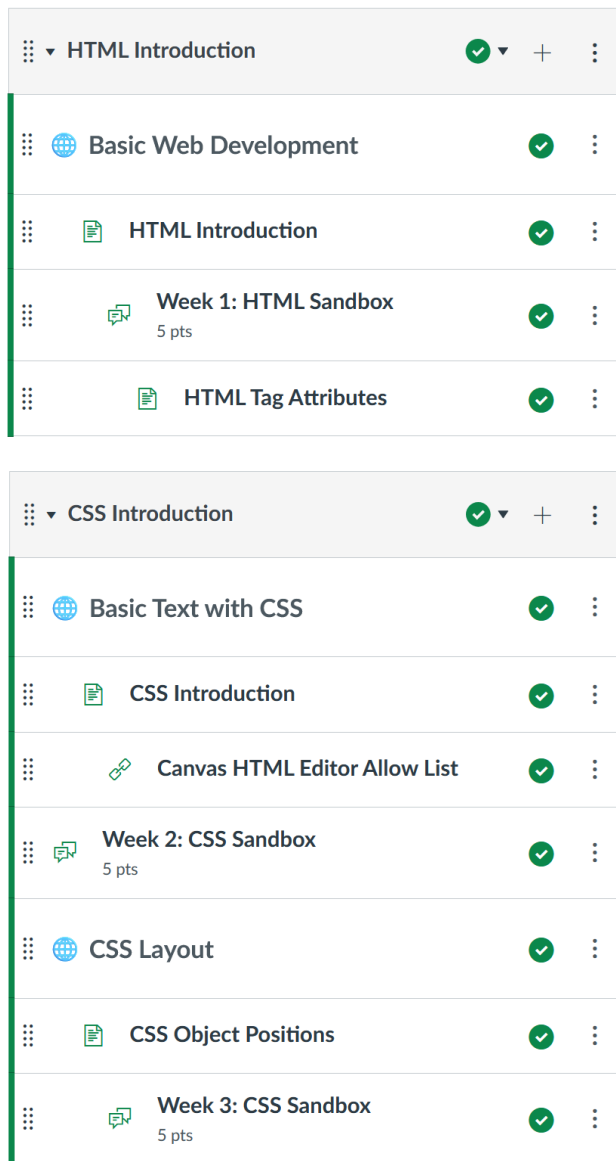


Figure 1. Module structure in Canvas

LEARNING OBJECTIVES

By the end of this three-week lesson, learners will be able to:

- Recognize how HTML tags function within a document.
- Add and remove block and inline HTML tags in Canvas discussion boards.
- Add and alter attributes of those tags in a Canvas discussion board.
- Recognize three ways that CSS elements can be added to a web document.
- Recognize the CSS elements allowed in Canvas.
- Use the style attribute to add CSS elements that influence text properties to discussion threads.
- Use the style attribute to add object placement CSS to HTML tags in Canvas discussion boards.

CONTEXT AND SETTING

In the early 2000s, the lead author taught basic HTML to preservice teachers as part of an introductory technology integration course. However, user-friendly online content creation tools have diminished this need (e.g., Google Sites, WordPress, Wix). Educators can publish basic websites using *What You See Is What You Get* (WYSIWYG) editors.

Learning management systems like Schoology and Canvas also use WYSIWYG editors for content creation. However, these editors have limited page layout functionality. Additionally, WYSIWYG editors sometimes scrambled HTML codes when users copy and paste text from word processing documents or make several color, font, and size adjustments. These scrambled HTML codes create awkward and unwanted formatting that can lead to frustration when users lack the HTML skills to correct the formatting problems.

Some LMSs like D2L Brightspace allow designers to quickly update page layouts with external style sheets. Changing course styles is as simple as updating the external style sheet. Others, like Canvas do not allow this access. CSS styles can be used but must be added to HTML tags within each page. CSS styles allow greater page layout customization but are more time consuming and a skill many people do not have.

LEARNER CHARACTERISTICS

Alumni from an instructional design and technology master's degree program at a medium, urban, southeastern university in the United States indicated they wanted more design and development opportunities in the program. Based on this feedback, faculty members selected several courses to include additional design opportunities. The seminar in online learning, the course where this three-week lesson is housed, was one of these selected courses.

Results from a skills assessment survey completed at the beginning of the course indicated that most students were comfortable developing content in an LMS, though few students had used the Canvas platform. Additionally, fewer students expressed comfort navigating HTML or using CSS to format web documents. Based on course sequencing, most students completed this course after they took the introduction to instructional design, message design, and multimedia development courses. Thus, students had some background in design and development when taking this course.

COURSE STRUCTURE

Based on the skills assessment survey results, the lead author developed content to help students consider structure and community in online course development (Shepherd, 2022). Assignments tasked learners to modify or create online modules to better organize and communicate course expectations, align with established rubrics for quality online learning experiences, and reduce perceptions of transactional distance. Students also developed online content to promote community and foster a sense of belonging. Development activities occurred within free Canvas accounts. Students built course content and shared them with peers and their instructor for feedback.

The lead author also wanted to include HTML and CSS attributes, but the asynchronous seminar course was scheduled in a five-week, shortened summer semester. To learn HTML and CSS attributes, the lead author believed that learners would need practice over time to troubleshoot problems, better conceptualize principles, and gain confidence in their abilities. Yet, the author did not want to devote the entire course to CSS instruction. Rather, he focused on mini lessons, consisting of about one hour each

week over three-weeks, to be included within modules devoted to broader course topics.

DEVELOPMENT RATIONALE

Because Canvas already provided basic HTML commands to developers through WYSIWYG editors, the lead author speculated that learners only needed to familiarize themselves with those commands to begin learning CSS. The author also did not want to focus on web design principles like file management or HTML page structure (e.g., doctype, title, head, body tags) because Canvas handled these functions.

Web pages developed within Canvas provide an option for anyone to edit them. However, the lead author believed that having several learners examine and modify the same HTML document (even asynchronously) would result in frustration. If anyone maladjusted a tag, it might influence the rendering of others' work. Rather than rely on one page for all learners, the lead author desired a space where learners could explore page design principles in their own area and demonstrate ideas to the larger group for feedback. Discussion boards fit this desire.

Discussion boards leverage the WYSIWYG editor of other content pages. They also provide an HTML editor, allowing learners to examine and modify the underlying HTML and CSS tags. Posts in a discussion board are assigned (and labeled) to the content creator. They also provide editing rights to that creator and their instructor (providing opportunities for troubleshooting). Additionally, posts are viewable by other class members and threaded by topic. These features allow other learners to examine tag experimentations, reply to posts with feedback and ideas, and maintain the original post—capturing the evolution of the experience.

With these ideas, the lead author developed mini lessons regarding HTML and CSS during the first 3 weeks of the course. Four short videos (between six and 15 minutes) introduced concepts. Students practiced those concepts in discussion board sandboxes where they were graded solely on participation. This participatory activity allowed learners to primarily focus on other aspects of the course but still gain experience designing content with HTML and CSS. A final assignment leveraged CSS for page layout purposes.

LEARNING REPRESENTATION

The five-week course was organized into five modules, one for each week. The first page of each module introduced module purposes, expected learning outcomes, readings and resources, and activities. Because these broader topics are not the focus of this article, they are not included. The HTML and CSS mini lessons were introduced within these larger modules. *During this lesson, italic text identifies the content or prompts provided to the learners.*

WEEK 1: BASIC HTML RECOGNITION

In the mini lesson during Week 1, learners were introduced to basic HTML coding using block and inline tags. They were encouraged to explore the presented content and ideas through participation-graded discussion boards. To situate the activity within the broader topic of online learning, learners were told that a basic understanding of HTML would help them grasp CSS, which would help them customize LMS page layouts. At the beginning of the Week 1 module, the following learning objectives were provided:

By the end of Week 1, you will be able to:

- *Recognize how HTML tags function within a document.*
- *Add and remove block and inline HTML tags as well as attributes of those tags.*
- *Add and alter attributes of those tags in a Canvas discussion board.*

INTRODUCTION TO HTML VIDEO

Learners then navigated to an [introduction to HTML video](#) (erschelsheph, 2022a) that included the following prompt:

This 9:20 minute video (feel free to speed-up playback) introduces the structure of an HTML document and introduces block and inline tags. It then shows you how to add tags to an HTML document (like a discussion post in Canvas). The content in this presentation is pretty basic. It distinguishes between block and inline tags. It also introduces heading, paragraph, division, unordered list, strong, line break, image, and span

tags. All tags mentioned in this presentation (with the exception of division and span) are available in the WYSIWYG Canvas discussion thread editor. However, knowing how to read HTML tags is integral to understanding CSS attributes and how to use them.

Take some time to play with HTML tags in the HTML Discussion thread this week. Feel free to begin by using the WYSIWYG editor to add headings, bold and italic text, and ordered and unordered lists.

Afterwards, review the tags using the HTML editor </> button at the bottom-right of the discussion post. Look for the beginning and ending tags. Add additional content between those tags. Remove a tag and see what happens to your text.

WEEK 1: HTML SANDBOX DISCUSSION

After learners reviewed the video, they were taken to a discussion board with the following prompt:

Use this space to practice writing HTML code. I realize the WYSIWYG editor will do a lot of this code for you. Begin there! Once you've added a few headings, paragraphs, lists, and bolded or italicized words, enter code view </> at the bottom-right of the editor and examine the tags. Then try to add additional content and tags manually in code-view. Becoming more familiar with HTML tags and how they appear will prepare you for CSS attributes that will be introduced later.

As you play with the codes, reflect on the process in your discussion thread. What makes sense? What do you struggle with in terms of grasping HTML tags and attributes? Mess around with the HTML tags as you write your reflections. Next, try adding a few link and image tags. Begin using the WYSIWYG editor. Then try to alter the size of the image or change the URL by modifying the attributes in the tag.

Remember: You can embed inline tags within block tags, but you cannot embed block tags within inline tags!

In addition to the prompt, the discussion board contained example block and inline tags (provided

below) with links to additional information from the W3schools (n.d.-a) website (see Support Materials; W3schools, n.d.-b to n.d.-l).

BLOCK TAG EXAMPLES

- `<p> </p>` paragraphs
- `<div> </div>` generic block container
- `<h1> </h1>` through `<h6> </h6>` headings
- ` ` ordered (numbered) lists
- ` ` unordered (bulleted) lists
- ` ` list items

INLINE TAG EXAMPLES

- ` ` generic inline container
- `<a> ` link
- `` image
- ` ` emphasis
- `
` line break

Learners spent a couple days adding HTML tags to discussion posts (either within the WYSIWYG editor or in the HTML editor) and then viewed the code to examine it. They also asked questions in the discussion board. The instructor posted ideas, provided feedback and suggestions, and encouraged learners to try new tags. Learners also supported each other by answering questions and troubleshooting problems.

HTML ATTRIBUTES VIDEO

After learners gained basic familiarity with block and inline tag appearances in the HTML editor, they watched an [HTML Attributes video](#) on modifying attributes within HTML tags (erschelsheph, 2022c). The video included the following text introduction:

Watch the HTML Intro Presentation and mess around with the Week 1: HTML Sandbox discussion thread before you watch this video.

This 6:04 minute video continues the HTML introduction by discussing attributes that can be placed within HTML tags. Attributes describe particular features of a tag. They may be used to indicate the size of an image or the web address of a link. They are important for CSS because a special attribute called "style" can be used to create inline CSS formatting in Canvas. After watching this presentation, try to add and modify

a few link and image tags in the HTML Sandbox discussion thread. Reflect on your experience. You do not need to experiment with the style attribute yet. We will begin covering that attribute next week.

Experimentation with these attributes occurred within the Week 1: Html Sandbox Discussion (previously described). The goal was to familiarize learners with HTML tags and attributes and help learners recognize those tags and attributes in the HTML viewer. Gaining this awareness helped learners add CSS attributes to those tags during the second and third weeks of the course.

WEEK 2: CSS INTRODUCTION

During the second week, learners were informed about three ways that CSS elements could be added to web documents: inline, internal, and external. They were then informed that the course would focus on inline CSS because that was the only option allowed in Canvas. Next, learners were introduced to CSS elements that influence text properties and were provided with a resource that documented what CSS elements were allowed in Canvas. Finally, students practiced adding CSS elements to WYSIWYG generated HTML through another participation-graded discussion thread. At the beginning of the Week 2 module, the learning objectives were provided.

By the end of Week 2, you will be able to:

- *Recognize three ways that CSS elements can be added to a web document.*
- *Recognize the CSS elements allowed in Canvas.*
- *Use the style attribute to add CSS elements that influence text properties to discussion threads.*

INTRODUCTION TO CSS VIDEO

Learners navigated to the [CSS Introduction video](#) (erschelsheph, 2022b) that included the following prompt:

Cascading Style Sheets (CSS) are used to format the layout of a webpage. They leverage HTML tags but provide additional structure to manage the look and feel of those tags. This 9:40 minute

video introduces CSS styles. It begins by reviewing the main sections of an HTML document (remember, Canvas only allows course developers to access tags within the body of the HTML page). The presentation then indicates three ways that CSS can be added to a webpage: inline, internal, and external. It focuses predominantly on inline styles because that is what Canvas allows. To add inline styles, you add the style attribute within HTML tags. You can then include various specifications for those styles. The presentation describes background-color, color, font-size, margin, padding, border, border-radius, and a few more.

Once you have watched this presentation, try your hand at basic CSS styles. Navigate to the Week 2: CSS Sandbox Discussion Board and modify the text color, background color, padding and other specifications for headings, regular text, and so forth. Learning how to complete these simple modifications will help prepare you for more difficult layouts in subsequent weeks.

Additionally, learners received a list of the HTML and CSS tags allowed in Canvas (Instructure, 2021). Afterwards, they experimented in a new discussion board.

WEEK 2: CSS SANDBOX DISCUSSION

Use this space to practice writing CSS.

Reflect on your experience learning HTML and CSS and modify those paragraphs and headings with the style attribute and various specifications.

Remember: each element is placed within the style attribute and uses a colon and semi-colon.

For example:

```
<p style="background-color: #aaaaaa; color: blue;">Content</p>
```

In addition to the prompt, the discussion board also contained example elements (provided below) with links to additional information from the W3schools (n.d.-a) website (see Support Materials; W3schools, n.d.-m to n.d.-t).

CSS EXAMPLES

- [background-color](#)
- [color](#)
- [font-size](#)
- [margin](#)
 - *margin-top, margin-bottom, margin-left, margin-right*
- [padding](#)
 - *padding-top, padding-bottom, padding-left, padding-right*
- [border-width](#)
- [border-style](#)
- [border-radius](#) [Links to an external site.](#)
 - *border-radius 0px 10px 0px 10px; (where the first number represents the top-left corner, second is top-right, third is bottom-right, and fourth is bottom-left)*

Learners spent their time altering HTML tags with inline CSS elements using the style attribute. During this time, students and the instructor posted examples and shared their HTML and CSS in discussion threads. Learners also used the space to ask questions, post ideas, and provide feedback (see Figure 2). Learners were encouraged to experiment with their ideas. As with the previous activity, five points of participation credit were awarded at the end of the week (based on the number of posts made and learner attempts to comprehend CSS formatting).

WEEK 3: OBJECT POSITIONING

During Week 3, learners were told that they would continue to examine CSS for layout design and object positioning. At the beginning of the Week 3 module, the learning objectives were provided.

- *Learners will work towards using the style attribute to add object placement CSS to HTML tags in Canvas discussion boards.*

As with the previous weeks, learners were provided content through a video and a sandbox-like discussion board.

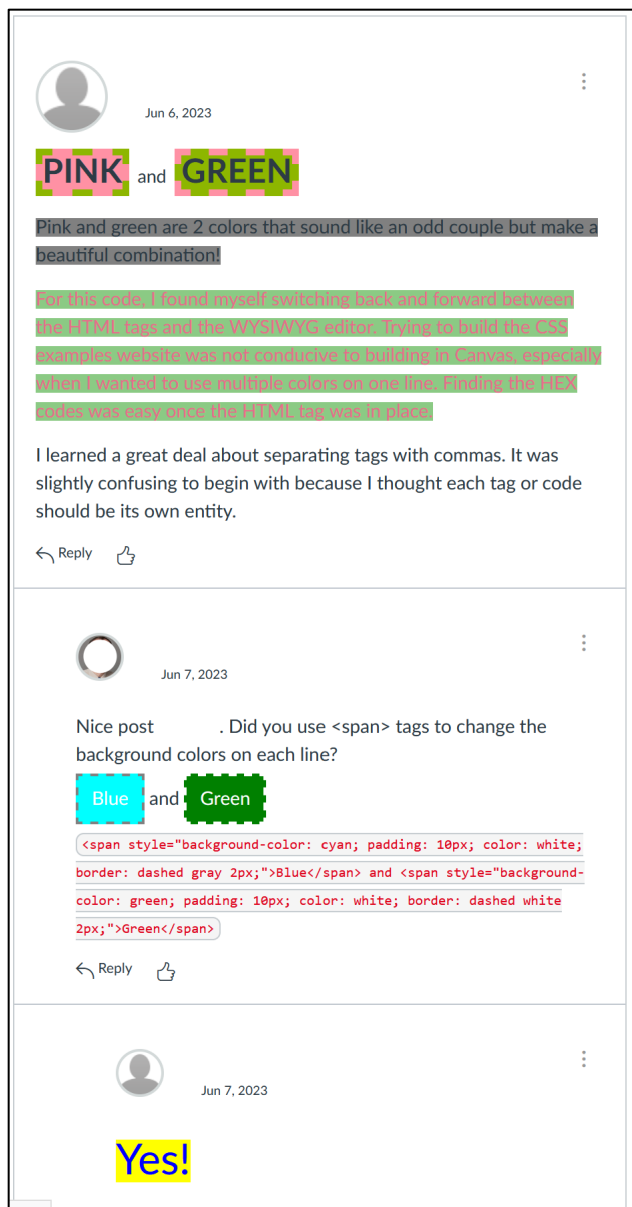


Figure 2. Example of student post with instructor feedback.

CSS OBJECT POSITIONING VIDEO

Learners began CSS instruction this week by navigating to the [CSS Positioning video](#) (erschelsheph, 2022d), which showcased object positioning with CSS. It included the following prompt in Canvas:

This 14:53 minute video describes how to use float, clear, min-height, position, and z-index styles to manipulate the layout of an HTML document.

Learners were reminded of allowable tags in the Canvas LMS (Instructure, 2021) and practiced their skills in a discussion board.

WEEK 3: CSS SANDBOX DISCUSSION

Now that you have some basic experience modifying HTML tags with CSS, use this space to practice modifying tag position on the page.

Reflect on your experience learning and modifying those paragraphs and headings with the style attribute and various specifications.

Remember: Each specification is placed within the style attribute and uses a colon and semi-colon.

For example:

<p style="background-color: #aaaaaa; color: blue;">Content</p> would show the word Content on the screen. It would be blue and have a gray background.

Additionally, learners were provided with the following lists of tags, some repeated from the previous week, with accompanying links to the W3schools (n.d.-a) website (see Support Materials; W3schools, n.d.-m to n.d.-z).

PREVIOUS CSS EXAMPLES

- [background-color](#)
- [color](#)
- [font-size](#)
- [margin](#)
 - Margin-top, margin-bottom, margin-left, margin-right
- [padding](#)
 - Padding-top, padding-bottom, padding-left, padding-right
- [border-width](#)
- [border-style](#)
- [border-radius](#)
 - Border-bottom-left-radius, border-top-right-radius

OBJECT PLACEMENT CSS EXAMPLES

- [Width](#)
- [Float](#)

- [Position](#)
 - *Static*
 - *Relative*
 - *Fixed*
 - *Absolute*
 - *Sticky*
- [z-Index](#)
- [Clear](#)
- [Min-height](#)

Again, learners modified HTML tags with CSS, particularly focusing on float, clear, and min-height, and positioning elements (see Figures 3 and 4).



Figure 3. Learner example and question using float.

The instructor also posted examples in discussion threads. Learners used the space to ask questions, post ideas, and provide feedback. They were awarded five points for participation.

OPTIONAL ASSESSMENT

During the fourth week of the course, learners were tasked to develop two activities within a Canvas course shell that promoted community development. These activities were graded based on adherence to web accessibility guidelines and the use of CSS

principles. Learners were told that their activities would be evaluated on the look and feel of the module (through CSS styles and relevant images), including the following:

- Change the background color of at least one HTML block tag of your choice.
- Use at least two different levels of heading and include different text colors, text sizes, padding, margins, border options or other components to distinguish them from normal text.
- Include at least one object that floats to the right of your window but does not overlap with other page elements.
- Include an image that takes up between 33-50% of the screen and includes a border of your choice.

Because the focus of this assignment was community development (as opposed to CSS), only three of 20 points were allotted to these tasks.

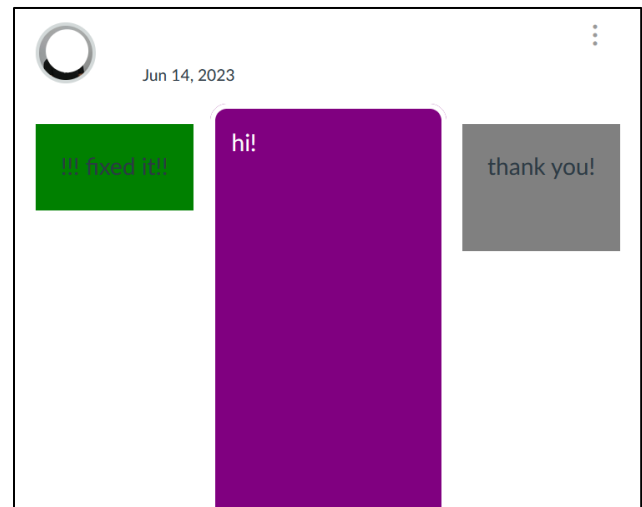


Figure 4. Resolution after instructor feedback.

CRITICAL REFLECTION

This lesson has been implemented three times, once per year during a shortened summer semester. Overall, most learners in the course appear to enjoy the activities. Most learners also gained basic skills with inline CSS. They successfully used these concepts to help stylize Canvas course shells as described in the assessment activity. However, a few struggled to use inline CSS concepts during weeks two and three. Some attempted to leverage ChatGPT and artificial intelligence for help, which often

resulted in CSS styles suitable for external style sheets or within page headings but unsuitable for inline CSS. This section describes these issues in greater detail and provides ideas for future implementation.

LEARNER PERCEPTIONS AND USES

As mentioned, most learners appeared to enjoy CSS sandbox activities. Since these mini lessons were only one part of the content for the whole course, other graded, course-level discussion boards prompted learners to synthesize instructional content, relate content to current practice, post and respond to guiding questions, and other formal practices. The HTML/CSS Sandbox Discussion Boards differed from these formal discussions.

First, sandbox discussion board points were awarded solely on participation. Learners could earn full points whether or not they successfully used inline CSS elements. If learners posted multiple attempts, they obtained participation points. Second, content did not matter; learners did not have to provide insightful comments or well-developed summaries. Rather, they could post whatever came to their mind, including nonsense. Posts focused on aesthetics rather than content. Many students wrote about extracurricular interests or TV/streaming shows they regularly viewed. Some posted Lorem Ipsum and other dummy text. This shift in focus away from posting specific content seemed to provide a light-hearted space to pursue colorful, vibrant, and bombastic posts.

Although the discussion boards continued to emphasize thoughtful participation with original posts related to module activities, the sandbox-like structure and grading of the HTML/CSS discussion forums seemed to lessen feelings of anxiety among many participants. Because discussion threads captured learner attempts over time, they depicted successive approximations that were mined by other learners. Individuals posted their attempts and often wrote whether it appeared as they desired. The posts that deviated from desires typically included learners' code (sometimes code was requested by the instructor or other learners). This allowed the instructor and other learners to provide feedback and trouble-shoot collectively. Suggestions often resulted in additional posts and further attempts. Although these tinkering methods and participatory, troubleshooting culture went well in the discussion

forums, there were some unique challenges in these mini lessons.

LEARNER ANXIETY

Introducing any type of coding can be intimidating. Bringing this content to a fast-paced, asynchronous shortened summer course can add to the anxiety of learning HTML/CSS. Some students were intimidated by the topic and the sandbox. They wondered why coding was being introduced in a seminar course regarding online learning. During the second week, a student posted the following message (see Figure 5):

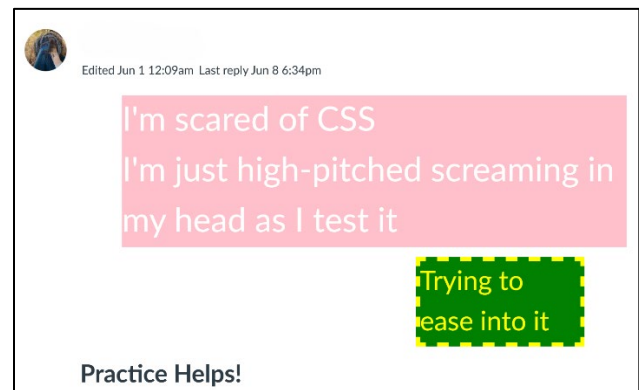


Figure 5. *I'm scared of CSS message.*

A few days later, another learner wrote a response:

I genuinely appreciate your honesty about being uncomfortable and struggling. I was seriously not thrilled when I saw this on the course outline.... This is still something I am struggling with, as I keep putting off playing around in the sandbox, pun intended, because I just know that I am not *good* at it.

The original poster then stated:

You're not alone with the discomfort. It's all new to me, too. To me, the sandbox feels like a safe place to practice. We are all learning, and it's OK to make mistakes. I believe that's the purpose of the sandbox. I know it's easier said than done to feel comfort in the discomfort.

She then concluded the thread two days later with the following post (see Figure 6).



Figure 6. Borrowed/modified content from the sandbox.

CANVAS UPDATES

Additionally, during the Summer 2024 iteration of this course, the university updated Canvas discussion boards to provide greater functionality. This functionality update introduced display challenges for these mini lessons. Previously, a new or edited post would automatically appear in discussion threads. The update removed this functionality for the duration of the course. New posts sometimes required page refreshes to appear. This problem caused confusion and anxiety among some learners who spent considerable time formatting content only to see it disappear when posted. Several reminders to refresh the page reduced anxiety. Yet, prior to the reminders, some students created two or three versions of the same post—believing their previous work had been lost. When implementing this activity, instructors may want to check with their information technology department to determine if any LMS changes may require adjustments.

ARTIFICIAL INTELLIGENCE ASSISTANCE?

Each sandbox area includes sample tags or elements with links to additional information on the W3schools (n.d.-a) website. Learners appreciated these links. However, they often used other sources to continue their exploration, including artificial intelligence (AI). Results varied. Those with prior experience using CSS were able to query AI tools for new elements, tips, and code snippets. They often shared their findings in the discussion thread and indicated that

they used AI tools to help them. However, AI assistance caused frustration for some learners. Often AI tools would generate CSS code that was formatted for an external style sheet or for the head section of an HTML document. These styles, while more sophisticated than what we did in class, often could not function as inline styles. Learners would post the code and wonder why it did not work. Fortunately, other learners and the instructor frequently pointed out challenges with their code and potential modifications to remedy the situation. When implementing this activity, instructors may provide information on the benefits and drawbacks of using AI and reiterate the three ways CSS elements can be added to web documents.

FUTURE PLANS

This lesson attempted to introduce basic HTML and CSS to graduate students in an instructional design and technology program. The instructors knew that a four-to-five-week introduction was insufficient to have learners master the concepts but attempted to have some inclusion as a starting point for learners. The authors have been using CSS styling for several years and continue to gain new insights and ideas as they leverage different elements. The experience seemed to reduce anxiety among learners in leveraging HTML views in LMSs, attempting to modify code in that view, and personalizing page layouts.

However, as mentioned in the AI section of this reflection, the lesson does not cover external style sheets and some of the more powerful cascading features of cascading style sheets. In this sense, this lesson is a very narrow introduction to HTML and CSS. In future semesters, we plan on adding another video that introduces external style sheets and linking those sheets in the head section of an HTML document. The cascading nature of CSS will be further introduced, showcasing how it can simplify and fine tune page layouts for websites and learning management systems (like D2L Brightspace).

REFERENCES

- erschelsheph. (2022a, June 3). *HTML intro* [Video]. YouTube. <https://youtu.be/Mu4iCAWX0Nw?si=p6XYOUeJtaf6mnik>

erschelsheph. (2022b, June 6). *CSS intro* [Video]. YouTube. https://youtu.be/D94BunRgbl?si=c_JHTaU7iTusfONA

erschelsheph. (2022c, June 6). *HTML attributes* [Video]. YouTube. <https://youtu.be/ZaOhFNr6Ys8?si=LIXtgisfAz5deVkW>

erschelsheph. (2022d, June 8). *CSS positioning* [Video]. YouTube. <https://youtu.be/NDErXUiiirV?si=tnukQkZE3rshkrzX>

Instructure. (2021, July 7). *Canvas HTML Editor Allowlist*. Retrieved April 12, 2024, from <https://community.canvaslms.com/t5/Canvas-Resource-Documents/Canvas-HTML-Editor-Allowlist/ta-p/387066>

Instructure. (n.d.-a). *Canvas*. Retrieved November 30, 2024, from <https://canvas.instructure.com/login/canvas>

Instructure. (n.d.-b). *How do I import a Canvas course export package?* Retrieved December 4, 2024 from <https://community.canvaslms.com/t5/Instructor-Guide/How-do-I-import-a-Canvas-course-export-package/ta-p/795>

Shepherd, C. E. (2022). Considering structure in online learning. *Journal of Technology-Integrated Lessons and Teaching*, 1(2), 24-35. <https://doi.org/10.13001/jtilt.v1i2.7383>

W3Schools. (n.d.-a). Retrieved November 30, 2024, from <https://www.w3schools.com/>

- W3schools. (n.d.-d). *HTML <h1> to <h6> tags*. Retrieved November 30, 2024, from https://www.w3schools.com/tags/tag_hn.asp
- W3schools. (n.d.-e). *HTML tag*. Retrieved November 30, 2024, from https://www.w3schools.com/tags/tag_ol.asp
- W3schools. (n.d.-f). *HTML tag*. Retrieved November 30, 2024, from https://www.w3schools.com/tags/tag_ul.asp
- W3schools. (n.d.-g). *HTML tag*. Retrieved November 30, 2024, from https://www.w3schools.com/tags/tag_li.asp

INLINE TAG EXAMPLES

- W3schools. (n.d.-h). *HTML tag*. Retrieved November 30, 2024, from https://www.w3schools.com/tags/tag_span.asp
- W3schools. (n.d.-i). *HTML <a> tag*. Retrieved November 30, 2024, from https://www.w3schools.com/tags/tag_a.asp
- W3schools. (n.d.-j). *HTML tag*. Retrieved November 30, 2024, from https://www.w3schools.com/tags/tag_img.asp
- W3schools. (n.d.-k). *HTML tag*. Retrieved November 30, 2024, from https://www.w3schools.com/tags/tag_strong.asp
- W3schools. (n.d.-l). *HTML
 tag*. Retrieved November 30, 2024, from https://www.w3schools.com/tags/tag_br.asp

CSS EXAMPLES

- W3schools. (n.d.-m). *CSS backgrounds*. Retrieved November 30, 2024, from https://www.w3schools.com/css/css_backgroun_d.asp
- W3schools. (n.d.-n). *CSS colors*. Retrieved November 30, 2024, from https://www.w3schools.com/css/css_colors.asp
- W3schools. (n.d.-o). *CSS font size*. Retrieved November 30, 2024, from https://www.w3schools.com/css/css_font_size_a_sp
- W3schools. (n.d.-p). *CSS margin*. Retrieved November 30, 2024, from https://www.w3schools.com/css/css_margin.asp
- W3schools. (n.d.-q). *CSS padding*. Retrieved November 30, 2024, from https://www.w3schools.com/css/css_padding.as_p

SUPPLEMENTAL MATERIALS

The HTML supplemental materials are presented in order of appearance in the article.

BLOCK TAG EXAMPLES

- W3schools. (n.d.-b). *HTML paragraphs*. Retrieved November 30, 2024, from https://www.w3schools.com/html/html_paragraphs.asp
- W3schools. (n.d.-c). *HTML <div> tag*. Retrieved November 30, 2024, from https://www.w3schools.com/tags/tag_div.asp

- W3schools. (n.d.-r). *CSS border width*. Retrieved November 30, 2024, from https://www.w3schools.com/css/css_border_width.asp
- W3schools. (n.d.-s). *CSS borders*. Retrieved November 30, 2024, from https://www.w3schools.com/css/css_border.asp
- W3schools. (n.d.-t). *CSS rounded borders*. Retrieved November 30, 2024, from https://www.w3schools.com/css/css_border_rounded.asp

OBJECT PLACEMENT CSS EXAMPLES

- W3schools. (n.d.-u). *CSS layout - width and max-width*. Retrieved November 30, 2024, from https://www.w3schools.com/css/css_max_width.asp
- W3schools. (n.d.-v). *CSS layout - float and clear*. Retrieved November 30, 2024, from https://www.w3schools.com/css/css_float.asp
- W3schools. (n.d.-w). *CSS layout - The position property*. Retrieved November 30, 2024, from https://www.w3schools.com/css/css_positioning.asp
- W3schools. (n.d.-x). *CSS layout - The z-index property*. Retrieved November 30, 2024, from https://www.w3schools.com/css/css_z_index.asp
- W3schools. (n.d.-y). *CSS layout - Clear and clearfix*. Retrieved November 30, 2024, from https://www.w3schools.com/css/css_float_clear.asp
- W3schools. (n.d.-z). *CSS min-height property*. Retrieved November 30, 2024, from https://www.w3schools.com/cssref/pr_dim_min-height.php

ABOUT THE AUTHORS

Craig E. Shepherd is an Associate Professor of Instructional Design and Technology at the University of Memphis with research interests regarding technology use in formal and informal settings to foster knowledge acquisition and community. He can be contacted at craig.shepherd@gmail.com.

Andrea F. Hall is originally from Albuquerque, New Mexico and began her career in education as an English teacher, teaching in both special and general education settings. She moved to Nashville to pursue

her M.Ed. in Leadership and Organizational Performance from Vanderbilt University. She continued teaching English in Middle Tennessee and is now a design and content specialist for educational organizations within Tennessee. She is currently pursuing a Master's degree in Instructional Design Technology at the University of Memphis.

Maxine D. Evans is currently enrolled in the doctoral program at the University of Memphis pursuing studies in Instructional Design and Technology. Her research interests include technology integration in public schools. She has engaged with the school system as a parent, as an advocate for children with special needs, and as a substitute teacher. She can be contacted at maxyevans@gmail.com.

SHARING & MODIFICATION PERMISSIONS

Unless otherwise noted, this article and its resources are published under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International license](https://creativecommons.org/licenses/by-nc-sa/4.0/):



You can freely share the article and its resources if you indicate the original authors, identify the Creative Commons license, and use them non-commercially.

You may also make and share modifications by:

- [Identifying the original authors](#).
- Using the resources non-commercially.
- Licensing modifications under the CC BY-NC-SA 4.0 license (and including a link to it).
- Indicating what modifications were made.