Unit 2 - Web Development

In the Web Development unit, students are empowered to create and share the content on their own web pages. They begin by thinking about the role of the web, and how it can be used as a medium for creative expression. As students develop their pages and begin to see themselves as programmers, they are encouraged think critically about the impact of sharing information online and how to be more critical content consumers. They are also introduced to problem solving as it relates to programming, as they learn valuable skills such as debugging, commenting, and structure of language. At the conclusion of the unit, students compile their work to create a personal website they can publish and share.

Chapter 1: Web Content and HTML

Big Questions

- Why do people create websites?
- How can text communicate content and structure on a web page?
- How can I incorporate content I find online into my own webpage?
- What strategies can I use when coding to find and fix issues?

Week 1

Lesson 1: Exploring Websites

Unplugged

This lesson covers the purposes that a website might serve, both for the users and the creators. The class explores a handful of the most-used websites in the United States and discusses how each of those sites is useful for users and how it might also serve its creators.

Lesson 2: Websites for Expression

Unplugged

This lesson introduces websites as a means of personal expression. The class first discusses different ways that people express and share their interests and ideas, then looks at a few exemplar websites made by students from a previous course. Finally everyone brainstorms and shares a list of topics and interests to include, creating a resource for developing a personal website in the rest of the unit.

Lesson 3: Intro to HTML

Web Lab

This lesson introduces to HTML as a solution to the problem of how to communicate both the content and structure of a website to a computer. The lesson begins with a brief unplugged activity demonstrating the challenges of effectively communicating the structure of a web page. The class looks at an HTML page in Web Lab and discusses how HTML tags help solve this problem, then uses HTML to write the first web pages of the unit.

Week 2

Lesson 4: Headings

Web Lab

This lesson continues the introduction to HTML tags, this time with headers. The class practices using header tags to create page and section titles and learns how the different header elements are displayed by default. Next, the class plans how to organize their content on the personal web pages that will be built across the unit and begins the first page of the project.

Lesson 5: Digital Footprint

Unplugged

This lesson takes a step back from creating the personal website to talk about personal information people choose to share digitally. The class begins by discussing what types of information are good to share with other people, then looks at several sample social media pages to see what types of personal information could be shared intentionally or unintentionally. Finally, the class comes up with a set of guidelines to follow when putting information online.

Week 3

Lesson 6: Lists

Web Lab

This lesson introduces ordered and unordered lists and the associated <ul>, <ol>, and <li> HTML tags. The class practices using the tags, then goes back to the personal web page project to add a new HTML page that includes the new tags.

Lesson 7: Intellectual Property and Images

Unplugged | Web Lab

This lesson covers how to use media such as images, video, or music created by others a website while respecting the rights of the creator of that media. After first studying Creative Commons licensing, the class learns how to add images to web pages, and how to give proper attribution when doing so.

Lesson 8: Clean Code and Debugging

Web Lab

This lesson covers common issues that arise when designing web pages in HTML. The class will correct errors in a sequence of increasingly complex web pages found on Code Studio and learn the importance of comments, whitespace, and indentation as tools for making web pages easier to read.

Week 4

Lesson 9: Project - Multi-Page Websites

Web Lab | Project

This lesson covers hyperlinks, which allow web developers to connect pages together into one website. The class will link together all the previous pages into one project, and create navigation bars for each page before publishing the entire site to the Web.

Chapter Commentary

Students use computing as a form of self expression as they design and develop basic web pages. Focusing on the tags, keywords, and syntax used to communicate instructions to the computer, students use HTML to structure the content of a web page. They also explore the privacy and intellectual property implications of publishing their work online.

Chapter 2: Styling and CSS

Big Questions

- How do I modify the appearance and style of my web pages?
- How do I safely and appropriately make use of the content published on the Internet?

Week 5

Lesson 10: Styling Text with CSS

Web Lab

This lesson introduces CSS as a way to style elements on the page. The class learns the basic syntax for CSS rule-sets and then explores properties that impact HTML text elements. Finally, everyone applies text styles to their personal websites.

Lesson 11: Styling Elements with CSS

Web Lab

This lesson continues the introduction to CSS style properties, this time focusing more on non-text elements. The class begins by investigating and modifying the new CSS styles on a Desserts of the World page. Afterwards, everyone applies this new knowledge to their personal websites.

Lesson 12: Sources and Search Engines

After first completing a web search scavenger hunt, the class learns about the inner workings of search engines and has an opportunity to flex their analytical skills in a search for strange and unlikely animals.

Lesson 13: RGB Colors and Classes

Web Lab

This lesson covers classes and custom colors. The class first learns how to specify custom colors using RGB (red, green, blue) values, then applies these colors to a new Four Seasons web page, which uses CSS classes. Using classes, the class adds more styles to the Four Seasons web page, then use them to style their personal websites.

Week 6

Lesson 14: Project - Final Personal Website

Web Lab | Project

In the last few days of the unit, the class finalizes their personal websites, working with peers to get feedback, review the rubric, and put the finishing touches on the site. To cap off the unit, everyone shares their projects and how they were developed.

Chapter Commentary

After covering the basics of HTML, students dive into improving their websites with CSS. They learn to use colors, fonts and margins to create a unique style of their own design. At the end of the chapter students publish a personal portfolio website that demonstrates use of HTML for content and CSS for a personalized.

Lesson 1: Exploring Websites

Unplugged

Overview

Every website has a purpose, a reason someone created it and others use it. In this lesson, students will start to consider the purposes a website might serve, both for the users and the creators. Students will explore a handful of the most-used websites in the United States and try to figure out how each of those sites is useful for users and how they might also serve their creators.

Purpose

Unit 2 is all about the World Wide Web, with a focus on creating websites for self expression. To start students thinking about why they might want to create a website, they must begin to consider how websites are useful tools for both users and creators. The lesson starts with popular websites because they all serve clear purposes for a large number of users, but eventually students are asked to narrow their vision to simpler sites that might serve the needs of a smaller user group.

Assessment Opportunities

1. **Identify the reasons someone might visit a given website**

   On page 2 of the activity guide, check that students have multiple reasons that a user might visit Julia's website, such as getting recipe ideas or connecting with people who have similar interests.

2. **Identify the reasons someone might create a given website**

   On page 2 of the activity guide, check students' reasons for why Julia might make a website. Student explanations may include pragmatic reasons, such as getting feedback and advice from others, or more social reasons such as personal expression.

Agenda

Warm Up (10 minutes)

- **The Last Website you Visited**

Activity (45 minutes)

- **Websites We Use Frequently**
- **Finding the Purpose**

Wrap Up (5 min)

- **Web Development Goals**

View on Code Studio

Objectives

Students will be able to:

- Identify the reasons someone might visit a given website
- Identify the reasons someone might create a given website

Preparation

- [ ] Print a copy of the activity guide for each student.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **The Purpose of Websites** - Exemplar

For the Students

- **The Purpose of Websites** - Activity Guide

Make a Copy

Vocabulary

- **Website** - A collection of interlinked web pages on the World Wide Web

Teaching Guide

Warm Up (10 minutes)

The Last Website you Visited

Set Up: Have students take out their journal. In their journal have students answer the following prompt.

Prompt: What was the last website you visited? Why did you go to that website?

Discuss: Have students share out the last website they visited. Make sure students share why they went to that website and what information they were looking for. In addition you can ask students whether or not the website provided the information they wanted.

Teaching Tip

Websites vs. Apps: You may find that students don't **think** that they actually visit websites very frequently, because they use apps instead. If you encounter this, encourage students to consider whether those apps might also have website versions. Facebook, for example, might be used more frequently from an app, but it originally started as a website, and can still be accessed from the web.

Discussion

Goal: The goal of having this discussion is to pull out that websites share information and that people go to websites for a particular purpose. You only stay on that website if it fulfills your needs.

Activity (45 minutes)

Websites We Use Frequently

Prompt: With a neighbor, come up with a list of what you think are the top 5 most frequently visited websites. These can include websites that you visit often, but also consider which pages the other people in your life use frequently.

Share: Have groups share out their top 5 websites lists, keeping track of unique sites on the board.

Discuss: Why do you think these websites are so commonly used? What purpose do they serve, or what problems do they solve?

Discussion

Goal: The intent of this discussion is to get students thinking about both why people make websites, and why they use websites. Push students to give detailed purposes for some of the websites that the class generated.

Finding the Purpose

Remarks

> We came up with a lot of reasons why people might use some of the most visited websites, but that's only one side of the picture. The people who make websites also have their own reasons for doing so. In the next activity, you're going to start thinking about the purposes that various websites serve for both the user and the creator.

Group: Put students in pairs.

Distribute: Pass out a copy of the activity guide to each student.

Direct students to the "Top Websites" level on Code Studio.

Code Studio levels

- Top Websites
- **Student Overview**

Top Websites in the US

View on Code Studio

The following are some of the most frequently visited websites in the United States. Each includes a brief description of the website's core function.

- Google.com - Search engine for websites, pictures, videos, and other media

- Youtube.com - User-submitted video site with ratings and comments
- Facebook.com - Social media community that allows users to share information with friends
- Amazon.com - Online shopping sites for all kinds of goods
- Yahoo.com - A search engine, email provider, and news source
- Wikipedia.org - A free encyclopedia with content generated by users

The Purpose of Websites

Top Websites

This first page of this activity guide asks students to consider what purposes some of the most visited websites might serve. Using a list of the top ten websites in the US, student pairs select three sites that interest them and for each discuss:

- Why people might use that site (what purpose does it serve users?)
- Why the creators might have made that site (what purpose does it serve the creators?)

Encourage students to be as detailed as possible when brainstorming the purposes of these sites.

Share: Once everyone has had a chance to brainstorm about their sites, give the class a chance to share out their thoughts.

Teaching Tip

The goal of this activity to get students to consider why different websites were created. They don't need to necessarily get it "right" for each site, but they should demonstrate that they have considered the potential purposes that a website might serve for the creator. If students struggle to identify the purpose a site serves its creator, consider asking the following prompts:

- Does the web site sell goods or services?
- Do users contribute anything (images, information, reviews) that might be useful to the creator?
- Does the site include any paid advertising?

Personal Websites

The second page of this activity guide encourages students to transition from thinking about huge sites that serve the needs of millions of people, to the much smaller kinds of sites they could create, which can still effectively serve the needs of a narrower group of users.

Read through the personal website description with the class:

Website Description: Julia wants to become a better home cook, so she started a blog where she can post about the recipes she tries. Each week Julia attempts a new recipe and adds a page about it to her blog. Sometimes the recipes go really well and her blog readers leave her encouraging comments. Occasionally her recipe attempts don't go so well, but she still posts and usually gets some helpful advice from her readers. After starting to blog about the meals she makes, Julia has started to cook more often and is attempting dishes that she never would have tried before.

Discuss: Give students an opportunity to ask questions about this example site. Make sure they understand how a small site that is primarily a tool for individual self expression can also be a useful site for other people.

The final questions on the activity guide ask students to come up with potential purposes this website serves both for its creator and its users.

Share: Ask for volunteers to share the purposes they came up with.

Wrap Up (5 min)

Web Development Goals

Journal: At this point we've just scratched the surface of what websites really are, and we haven't even begun to create websites of our own. In your journal, come up with three goals for yourself this unit. These could be related to improving on one of our class practices, learning how to make a specific kind of web site, or creating a site that serves some purpose for you or others.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- **IC** - Impacts of Computing

Lesson 2: Websites for Expression

Unplugged

Overview

In this lesson students investigate ways to use websites as a means of personal expression and develop a list of topics and interests that they would want to include on a personal website. To begin the lesson students brainstorm different ways that people express and share their interests and ideas. Students then look at a few exemplar websites made by students from a previous CS Discoveries course to identify ways they are expressing their ideas. Finally students brainstorm and share a list of topics and interests they might want to include on a personal website which they can reference for ideas as they progress through the unit.

Purpose

This lesson introduces the central story of the unit, namely that web development, and by extension computer science, is an avenue for self expression. The warm up situates web development as another means of self-expression alongside visual art, dance, music, fashion, social media, and many other ways students are likely already expressing themselves. Seeing student exemplars helps scope students expectations for their skills by the end of the unit. Finally the brainstorm of content for their website provides students an opportunity to consider what messages they want to express as they build their websites.

Assessment Opportunities

1. **Identify websites as a form of personal expression**

   In the wrap up discussion, ensure that students are recognizing the personal touches that each person is bringing to their website design.

Agenda

Warm Up (5 mins)

Express Yourself

Activity (40 mins)

Personal Website Planning

Wrap Up (5 mins)

View on Code Studio

Objectives

Students will be able to:

- Identify websites as a form of personal expression

Preparation

- [ ] Prepare journals or optionally print copies of the activity guide

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Personal Website Sketch** - Exemplar

For the Students

- **Personal Website Sketch** - Activity Guide
  Make a Copy

Vocabulary

- **Website Content** - The raw text, images, and other elements included in a web page

Teaching Guide

Warm Up (5 mins)

Express Yourself

Prompt: What are the ways that you or your friends express yourselves? Think about different ways of communicating, activities, or spaces that you consider a form of self-expression.

Discuss: Students should brainstorm ideas independently, then share with a neighbor, and finally share with the class at large.

Remarks

> We share our thoughts, feelings, and ideas in many different ways. Some people might express their thoughts online, others might do so in the way they dress or by making a song or drawing a picture. Self-expression is a really important part of our lives, and as we're going to see, making websites is another way we can express our ideas, interests, and feelings.

Discussion

Goal: This warm-up should generate a list of ways students express their ideas. If they need help getting started offer them a couple examples from the list below. The goal of this warm-up is to frame web development as another form of self-expression in this list.

Draw a picture, write a song, build something, dance, clothing and fashion, write and share something, talking with friends, posting something online

Activity (40 mins)

Personal Website Planning

Personal Website Planning Guide

Distribute: Have students open a journal or optionally distribute copies of the activity guide.

Brainstorm Content: Read through this section of the activity guide. Then give students several minutes to silently brainstorm content they might want to include in a personal website.

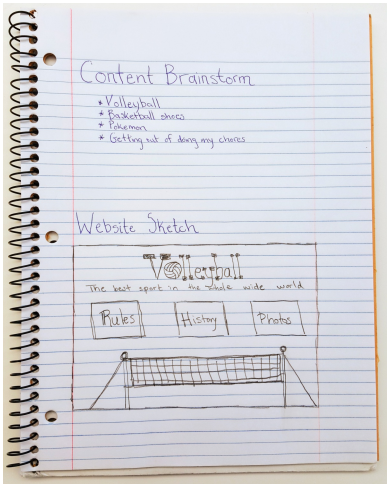
Share: Have students share their content ideas with a neighbor.

Remarks

> When sharing ideas with other people an important thing to consider isn't just what you want to say, but how you want to say it. You just created a list of ideas you might want to share on your personal websites. Let's go look at some personal websites other students have made and think about not only what content they're sharing, but how they are presenting that message.

Teaching Tip

Use Journals: The activity for today's lesson can easily be completed in student journals. Project or otherwise share the prompts and let students complete their work in their journals.



View Personal Websites: Send students to Code.org and have them look through the exemplar websites in pairs. Use these websites as inspiration for their own personal websites. They were created in Web Lab by other CS Discoveries students.

Code Studio levels

Lesson Overview

Teacher Overview | Student Overview

View on Code Studio to access answer key(s) View on Code Studio

Exemplar Personal Websites

Student Overview

Website Sketch: Give students some time to sketch one page from their website. Remind students that they should be thinking of ways to effectively share the content on that page. Use the student exemplars students looked at to help scope the complexity of web pages they should be considering.

Teaching Tip

Limit Time: Sketching a website can easily grow to take an entire class period if student expectations aren't scoped. Explain to students that they'll have plenty of time to refine their designs once they've developed more programming skills. Roughly 10 minutes should be plenty to get a reasonable sketch. Emphasize that their topic lists is much more important at this point but both will be updated throughout the unit.

Wrap Up (5 mins)

Share: Give students a chance to share their website sketches with a neighbor.

Prompt: Now that you've had a chance to share your websites sketches, write down one thing that makes your site special to you, and one thing that makes your neighbor's site special to them.

Assessment Opportunity

As students share different aspects of the websites, makes sure that the discussion highlights the creative aspect of creating a website, and that it can be a form of personal expression unique to each individual.

Remarks

> This unit we'll be working to bring many of your ideas to reality. Along the way you're going to learn some new computer languages and explore the way the websites you use are created. The sketches you made or even the content you want to share might change as we go through the unit, but this goal of using websites for self-expression will be there throughout.

Collect: If students did not put their website sketches in a journal collect them. They will explicitly be referenced again in Lesson 4.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- **AP** - Algorithms & Programming
- **IC** - Impacts of Computing

Lesson 3: Intro to HTML

Web Lab

Overview

In this lesson students are introduced HTML as a solution to the problem of how to communicate both the content and structure of a website to a computer. The lesson begins with a brief unplugged activity demonstrating the challenges of effectively communicating the structure of a web page. Students then look at an exemplar HTML page in Web Lab and discuss with their classmates how HTML tags help solve this problem. Students then write their first HTML. A wrap-up discussion helps to solidify the understanding of content vs. structure that was developed throughout the lesson.

Purpose

This lesson introduces many new concepts and tools to students. They are introduced to HTML, the Web Lab tool, and how to navigate lesson resources on Code.org in general. While the understanding of HTML as a way to communicate the structure of a web page is a critical learning objective, this lesson has students do minimal programming since there are many other new ideas and tools to grasp. In the next lesson students will have more time to spend programming in HTML.

Assessment Opportunities

1. **Explain that HTML allows a programmer to communicate the way content should be structured on a web page**

   In the final reflection prompt, students should give an explanation of how HTML is used to structure content on a page.

2. **Write a simple HTML document that uses opening and closing tags to structure content**

   Level 8 on Code Studio provides a mini-rubric to assess student use of HTML.

3. **Understand how to use lesson resources provided in Web Lab**

   As students progress through the HTML lesson, check that they are looking through the drop down tips on the right hand side of the instructions, clicking into the map levels in the "Help and Tips" tab, and making use of the inspector tool.

Agenda

Warm Up (10 mins)

View on Code Studio

Objectives

Students will be able to:

- Explain that HTML allows a programmer to communicate the way content should be structured on a web page
- Write a simple HTML document that uses opening and closing tags to structure content
- Understand how to use lesson resources provided in Web Lab

Preparation

- [ ] Review the Code Studio levels

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Exemplar Text Website**

Vocabulary

- **HTML** - Hypertext Markup Language, a language used to create web pages
- **HTML Element** - A piece of a website, marked by a start tag and often closed with an end tag
- **HTML Tag** - The special set of characters that indicates the start and end of an HTML element and that element's type
- **Website Content** - The raw text, images, and other elements included in a web page
- **Website Structure** - The purpose of different pieces of content in a web page, used to help the computer determine how that content should be displayed

Introduced Code

-
- <html></html>

The Need for HTML

Activity (30 minutes)

Exploring HTML

Wrap Up (5 minutes)

Reflection

- <head></head>
- <!DOCTYPE>
- <body></body>

Teaching Guide

Warm Up (10 mins)

The Need for HTML

Display: Show the image inside the **Exemplar Text Website**

Prompt: Imagine you wanted to explain to a person over the phone how to draw this web page. Write down as clear instructions so that what they would draw would perfectly match this image.

Discuss: Once students have written their instructions, have them briefly share their instructions with a neighbor.

Demo: Run a quick demo using the instructions below.

- Pick one student to verbally share one of their instructions with you.
- The teacher should act as the person on the phone trying to draw the web page
- Publicly “draw” the website exactly as the students instructions say. For example, if told to "Write bigger", write the word "bigger" on the page. If they don't indicate where text goes then place text in random locations.
- As the student gives you instructions have them tell you if you have drawn it correctly. If you have not drawn it correctly, have them make their directions more specific until you can draw it correctly.
- Change students after a couple instructions to get more students involved.
- Keep track of the instructions students give and the improvements they make to the instructions somewhere visible as well.
- Repeat this process until you have recreated most of the web page.

Discuss: Once you have finished drawing the site, quickly create a list of all the different kinds of information they needed to account for in their instructions. For example, location, size, font, etc.

Teaching Tip

If this site is blocked for students, your IT department may need to whitelist codeprojects.org. This is the same site that students will use to publish their own web pages, so it's important that they have access.

Discussion

Goal: Activities like this one are often used in CS courses to help highlight just how much precision is needed to communicate instructions to a computer. In this instance the goal is similar. You want to highlight the challenge of differentiating the actual content on the page and instructions indicating how it should be structured. This demonstration helps justify the creation of HTML in order to tag pieces of content to help the computer understand what they are and hence how they should look.

Remarks

> There's a lot of information that we need to communicate if we want to create web pages. It's not enough to just know what content you want to put on your page, like the actual words or images. You need to know where things should be and how they should look. Today we're going to start learning the languages used on the web to represent this additional information.

Activity (30 minutes)

Exploring HTML

Remarks

> Today we are going to start working with a lot of resources. As we discover each type of resource, we'll add it to the list here at the front of the room, and at the end of the lesson we'll review how each is used.

Display: At the front of the room, write the heading "Resources" on the board or blank poster paper, leaving space to list the various resources as they appear in the lesson.

Teaching Tip

Using Resources: Below you can find recommendations for using the many resources students are introduced to in the lesson. Wait until after students have seen all of these resources to review at the end of the lesson, but add them to the list and model the correct usage as they appear.

- **Videos:** Watched as a class, but students can always return to them.
- **Map Levels:** Contain text and diagrams explaining content. These are intended as helpful student resources, not class readings. They are a good place to go for review after learning content or when students get stuck in levels.
- **Level Instructions:** Instructions may introduce small pieces of new content. Each level features a "Do This" section explaining what students are supposed to do in that level. Set the expectation early that reading these instructions, not just the "Do This" section, is important.
- **Level Tips:** Students can click these tips in the instruction areas of lessons. Students should use these as a first place to check for help before talking with peers or a teacher.
- **Inspector Tool:** Highlights the code corresponding to a web page element when hovered over in the Preview Area.
- **Bubble Color:** Bubbles may turn green but there is no validation of correctness. Green only means a student clicked Continue or Finish for a level. Set the understanding early that this is more a tool for them than an indication of either completeness or correctness.

Code Studio levels

Lesson Overview

Student Overview

Experiment with Web Lab

2 *(click tabs to see student view)*

Video: Intro to Web Lab - Part 1

Teacher Overview | Student Overview

View on Code Studio

Discussion Goals

While there are no broad learning goals for this video, students should understand that the instructions for Web Lab levels appear in the wide panel at the top, and the bottom three panels, from left to right, include the list of files that they are working on, the area where they type in their code, and the area where they see the results of their code. For now, they do not need to worry about every button, as they will be explained later, but they should understand that the inspector tool helps them to link parts of the web page with code that created them, and the refresh and save button allows them to update their webpage if it does not automatically update when they change the code.

Although it's not part of the video, the instructions pane will also often include drop down questions and answers that students can use when they are completing an activity, and in some levels the tabs at the top of the instructions pane will take students to a "Help and Tips" area that includes reference information relevant to the activities.

Explore HTML

Teacher Overview | Student Overview

View on Code Studio

HTML Discussion

You should bring students back together once they've spent a couple minutes looking through this level. The discussion prompts listed in the level should be used in a standard Think-Pair-Share structure.

- What text is appearing in both the code and the web page?
- How is this language communicating extra information about the way to represent text?

The goal of the discussion is to call out the features of HTML that students are noticing. The two primary takeaways (reinforced in the subsequent video as well) are that HTML uses a system of tags to surround content and indicate what it is and how it should be displayed.

Video: Intro to Web Lab - Part 2

Teacher Overview | Student Overview

View on Code Studio

Discussion Goals

As student discuss HTML tags, make sure they understand that HTML tags are used to structure, or organize, content on the screen. Talking about the organization, structure, or role of the content in the page (heading, paragraph, list, etc.) is more accurate than talking about specific aspects of its appearance (such as size or spacing).

Although certain structural roles may be associated with appearance (headings are larger), HTML does not specify anything in particular about the appearance. This allows users with different needs, such as the visually impaired, to interact with the structured content in a way that makes the most sense for them. Later in the unit, students will learn about styles and CSS, which allows them to specify exact size, color, and spacing to their HTML elements.

The paragraph tag separates text into paragraphs. You may want to follow up this question by asking students how they think a web browser for the blind might deal with paragraphs. For example, while sighted people may use spacing and new lines to separate out paragraphs, what should a screen reader do?

Wrap Up (5 minutes)

Reflection

Prompt: In your own words, how does HTML help solve the problem of telling a computer what a web page looks like, not just what content is on it?

Discuss: Have students write or silently think out their ideas, then share with a partner, then share with the class.

Vocabulary: Introduce the following words Website Content Website Structure

Remarks

> HTML uses tags to help the computer know what different pieces of content in the web page actually are. Right now we've only learned how to tell the computer that some text is a paragraph, or that part of your website is the body. We've already seen how that affects the way our web pages look and are structured. As we move forward we're going to learn more tags and see more examples of how this language helps us add structure to our webpages.

Review: Return to the list of lesson resources you wrote on the board and review as a class how they are supposed to be used. Refer to the teaching tip above for recommended uses.

Assessment Opportunity

Goal: Students answers will vary but will likely center around the fact that using tags helps the computer know what different pieces of content "are". Using these tags helps the computer know what the tags are supposed to look like. If this discussion needs to be returned to after students have seen more tags that's fine as well. In either case use this discussion to motivate the content vs. structure wrap-up point.

Content Corner

Content - Structure - Style: Once students have been through this lesson the definitions provided here should have some context. The content is the literal words that are being typed on the page. Using HTML students are providing stucture to the page, explaining how those pieces of content should be interpreted. Later in the unit students will learn CSS, a language that allows them to individually style elements. For now, however, the styles being applied based on their HTML tags are just the default styles of their web browser. Students don't need to fully understand this difference at this point, as it will be much clearer once they learn CSS in chapter 2. The difference between content and structure, however, will be returned to many times in this chapter.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- **AP** - Algorithms & Programming

Lesson 4: Headings

Web Lab

Overview

In this lesson, students continue to use HTML to structure text on web pages, this time with headings. Students learn how the different heading elements are displayed by default and practice using them to create page and section titles. Students then start to decide how they will organize their content on their own personal web pages. In the last level, students begin the project that they will continue to work on throughout the unit.

Purpose

This lesson introduces the heading tags that students will use in their pages for the rest of the unit. It also reinforces the general structure of HTML (opening tag / content / closing tag) so that students are ready to begin adding their own content in the project.

The personal web page that students work on at the end of the lesson will follow them throughout the unit. As they learn more HTML and CSS, they will improve the page, adding images, colors, and different fonts. This lesson is a chance for them to start the page, knowing that they will continue to improve it over the next few weeks.

Assessment Opportunities

1. **Use heading tags to change the appearance of text on a web page.**

   See the multiple choice question on Level 5 and the rubric on Level 6.

2. **Structure content into headings, subheadings, and paragraphs.**

   See the rubric on Level 6.

Agenda

Warm Up (5 minutes)

- **Tags Poster**

Activity (45 minutes)

- **Pair Programming**
- **Web Lab: Headings**

Wrap Up (5 minutes)

- **Journal**

View on Code Studio

Objectives

Students will be able to:

- Use heading tags to change the appearance of text on a web page.
- Structure content into headings, subheadings, and paragraphs.

Preparation

- [ ] Have student journals (or project sketches) ready to give back.
- [ ] If you will not use journals to track HTML tags, prepare poster paper to do so as a whole class

Vocabulary

- **Heading** - A title or summary for a document or section of a document.

Introduced Code

- <h1></h1>

Teaching Guide

Warm Up (5 minutes)

Tags Poster

Group: Put students into groups of three to four students.

Journal: Have students make a T-chart on a blank page in their journals and label the top "HTML Tags"

Prompt: Yesterday, you learned about HTML, the language of the World Wide Web. HTML uses tags to structure content on web pages. In your groups, think of as many tags as you remember and what they do.

Give students a few minutes to think of as many tags as they can.

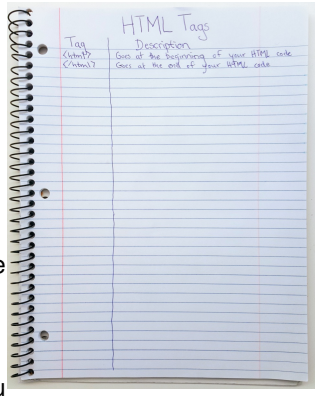
Discuss: Groups should discuss with one another and record their ideas in their journals (or alternately prepare to share on the shared class poster). Afterwards quickly share across the room to make sure all groups have listed all tags.

Remarks

> As we learn more HTML tags, we'll be keeping track of them so we'll have a reference as we make our web pages.

Teaching Tip

Journal or Poster? Throughout this unit students will be keeping track of the tags they learn. This warm up descibes one option for students to record the HTML tags that they learn by writing them in their journals. If you wish you can keep track of the same information on a shared class poster that you update after each lesson. Prompts throughout the unit will tell you when students should update their journals or the class poster.



Discussion

Goal: The goal of this discussion is to review the tags that students saw in the previous lesson.

- <!DOCTYPE html> - Tells the computer that this is a document written in HTML
- <html> - Indicates the beginning of your code written in HTML
- <head> - Contains information (sometimes called "metadata") about your web page
- <body> - Contains all the main content of your web page
- <p> - Defines a paragraph

Activity (45 minutes)

Pair Programming

Group: students into pairs.

Remarks

> We're going to be working on Web Lab again today, but this time we'll be using **pair programming**. Pair programming helps people make better programs by working together, but there are some rules we have to follow to make sure it goes well.

Display: Show the Pair Programming video.

Review: Ensure that students understand the rules for pair programming:

- There is only one computer.
- The driver is the only one to touch the keyboard/mouse.
- The navigator should look for problems in the code and keep track of the high-level plan.
- Both driver and navigator should be communicating constantly.
- Driver and navigator must switch when the teacher indicates, typically every couple minutes.

Teaching Tip

Pair Programming: Pair programming is practiced in education and in industry. Students who pair program are more confident in their abilities and are more likely to continue to study computer science. This practice is most effective if it is introduced early and the rules for switching partners are enforced by the teacher. You can read more about Pair Programming in the **Computer Science Discoveries Curriculum Guide**

Web Lab: Headings

Transition: Have pairs go to Code Studio and both log in using the "Pair Programming" feature.

Prompt: students to switch driver and navigator every three minutes.

Code Studio levels

Lesson Overview

Student Overview

Pair Programming

Teacher Overview | Student Overview

View on Code Studio

Discussion Goals

The goals of this discussion center less around particular answers to this question and more around promoting positive attitudes toward pair programming. As students discuss its potential benefits, make sure they understand that this is an industry standard practice, not just something done by learners in the classroom.

Headings

3 | 4 | 5 | 6 *(click tabs to see student view)*

Headings in HTML

Teacher Overview | Student Overview

View on Code Studio

Updating the Tags List

This map level recaps the tags that students learned throughout this lesson. Encourage students to use this resource to update their HTML tags list or poster.

Problem Solving Process for Programming

Teacher Overview | Student Overview

View on Code Studio

Review the Problem Solving Process

Review the four steps of the Problem Solving Process as a class.

Your Personal Website

9 | 10 *(click tabs to see student view)*

Wrap Up (5 minutes)

Journal

Prompt: Have students update their "HTML Tags" log with the heading tags they learned in this lesson.

Prompt: Have students reflect on their development of the **Five Practices of CS Discoveries** (Problem Solving, Persistence, Creativity, Collaboration, Communication). Choose one of the following prompts as you deem appropriate.

- Choose one of the five practices in which you believe you demonstrated growth in this lesson. Write something you did that exemplified this practice.
- Choose one practice you think you can continue to grow in. What's one thing you'd like to do better?
- Choose one practice you thought was especially important for the activity we completed today. What made it so important?

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▸ **AP** - Algorithms & Programming

Lesson 5: Digital Footprint

Unplugged

Overview

As students have recently spent some time thinking about the actual content that will go into their personal website, this lesson takes a step back from the unit-long project (publication of a personal website) to help students articulate what personal information they choose to share digitally and with whom. It also reinforces the notion that much of the information that they choose to share digitally falls largely out of their control the moment it is released.

Students begin by individually identifying appropriate audiences with whom they would be comfortable sharing given pieces of personal information. They then look at several social media pages to determine what sorts of information people are sharing about themselves or one another. Last, students reflect on what guidelines they think are appropriate for posting information online.

The ultimate point of this lesson is not to scare students, but rather to experientially bring students to realizing precisely what level of control they don't have in releasing information into the web.

Purpose

Now that students are beginning to share information publicly, it's crucial that we instill in them an understanding of the potential consequences of sharing personal information online.

The first activity engages students in considering specific pieces of personal information and the audiences with whom they would or would not want that information to remain private. The goal of this activity is for students to understand that for any given piece of personal information if there is **any** audience they would like to maintain privacy with they should consider not posting that information online.

In a follow up activity students look through several example social media profiles for fake students. Individually these pages contain relatively innocuous information, but when students cross reference information across multiple profiles they see how a detailed picture (or digital footprint) can start to form. Through this activity students see that even information that they may not see as private on its own can share more information than intended when combined with other small details.

Assessment Opportunities

View on Code Studio

Objectives

Students will be able to:

- Understand and explain reasons that it is difficult to control who sees information published online.
- Understand and justify guidelines for safely publishing information online.

Preparation

- [ ] Print out copies of "When Does Your Privacy Matter?" for each student
- [ ] Print out copies of "Social Sleuth" for each student (or one for each group if grouping)

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **When Does Your Privacy Matter?** - Exemplar
- **Social Sleuth** - Exemplar

For the Students

- **When Does Your Privacy Matter?** - Activity Guide **Make a Copy**
- **Social Sleuth** - Activity Guide **Make a Copy**

Vocabulary

- **Digital Footprint** - The collected information about an individual across multiple websites on the Internet.

1. **Understand and explain reasons that it is difficult to control who sees information published online.**

   In page 2 the Social Sleuth activity guide, look at students' explanations of how information was accidentally shared.

2. **Understand and justify guidelines for safely publishing information online.**

   Review students' checklists in the lesson wrap up.

Agenda

Warm Up (10 min)

- **Your Digital Footprint**

Activity (30-40 min)

- **How Much Do You Care About Privacy? Investigating Social Profiles**

Wrap Up (5 minutes)

- **Shaping Your Digital Footprint**

Teaching Guide

Warm Up (10 min)

Your Digital Footprint

Prompt: Ask the students to consider what websites they have given personal information to. In small groups, ask students to come up with a list of websites that they have accounts on (even if they use sign in with Facebook, Google, or similar).

Discuss: Have students silently brainstorm or journal, then share with a partner, and finally share as a full class. Create a comprehensive list of all of the websites that may have their personal information.

Discussion

Goal: This discussion is not about coming up with a comprehensive list of personal information students may have shared with websites, nor is it to frighten them out of sharing information. The goal is for students to start thinking more critically about when and where they share information about themselves.

This will prime them for the main activity of the day, looking at specific pieces of information and audiences to consider when their privacy matters, and whether they are unknowingly giving up their privacy.

Discuss: Ask the class about what kind of information they think these websites might have about them. You can frame it from a few different angles:

- What information do you know you've given to a website (eg. your email address)
- What information might you have unknowingly given to a website (eg. a picture with your home address)
- What information might other people have shared about you without your knowledge or permission (eg. tagging you on Facebook)

Activity (30-40 min)

How Much Do You Care About Privacy?

Distribute: activity guide to "When Does Your Privacy Matter?"

When Does Your Privacy Matter?

Step 1: Individually, have students complete the grid on the front side of the activity guide. Encourage them to think critically about the consequences of each audience having access to each piece of personal information. What might seem like an innocuous piece of information might have far reaching consequences depending on who has access to it.

Step 2: Once students have completed the privacy grid, have them come together into pairs or small groups. They should look for cells in each grid where their partner marked a different answer than they did and discuss their choices. Encourage students to respectfully debate when they disagree on an issue. While there are no right or wrong answers here, it's useful for students to hear how others may have considered consequences (both positive and negative) that they didn't think of.

Step 3: After the small group discussion, students should complete the reflection questions on the next page of the activity guide.

Discussion

In the pair discussion on the Activity Guide, the goal is not to focus on what the actual private information is (like who happens to be “the favorite teacher” or the “secret crush” of a student), but rather about identifying commonalities and differences in what students deem appropriate to share with different audiences.

Students may be tempted to think about specific examples that they would or would not be comfortable with others knowing, but you want to push them towards thinking more generally of the types of information and audiences presented.

Prompt: Were there any places on the grid where your discussion with your partner changed your perspective? If so, what were they?

Discuss: Have students share places where they changed their mind. Focus particularly on blind spots, assumptions, or misconceptions that students had about access to their personal information that may have been revealed through discussion with a partner.

Investigating Social Profiles

Goal: Explore how small pieces of personal information spread across multiple sites on the Internet can produce a fairly detailed picture of a person, known as a digital footprint.

Group: This activity can be done individually, but is better when students are in groups of 2-3

Distribute: activity guide to "Social Sleuth"

Transition: Send students to Code Studio.

Code Studio levels

Lesson Overview

Teacher Overview | Student Overview

View on Code Studio to access answer key(s)

View on Code Studio

Social Sleuth

Teacher Overview | Student Overview

Social Sleuth

View on Code Studio

Step 1: Students look through all of the provided social media pages on Code Studio. The pages represent several unique individuals, each of whom has an account on three social media platforms. They are distributed as follows (do not share this with students):

	FaceSpace	Chirpr	instantframe
John Thomas	John T	@johnsnow	johnsnow
Haley Gutierrez	Haley G	@flyinghail	desertrider
Lizzie Dell	Lizzie Dell	@lillizard	photolizzie

Tip:

Students are asked to choose two users from the available profiles, but if you are short on time, you can ask students to complete just one of the footprints.

Step 2: Once they've looked through the pages, students will attempt to figure out who the two users are and answer some detailed questions about them. The footprint questions are designed to push students towards combining details from multiple social platforms and understanding that together the profiles represent more detailed information than the users probably intend to reveal about themselves. Some questions may not be answerable for all users, or may have different levels of details (full street address for one user, but only a city and state for another).

Share: Have the class share the digital footprints that they developed through the activity.

Discuss:

- Which information was most difficult to find?
- Which details were innocuous on their own, but revealed private information when combined with other details?
- How does this make you think differently about what you choose to post online?

Wrap Up (5 minutes)

Goal: Develop a common understanding of how our choices to publish information online contribute to a digital footprint.

Shaping Your Digital Footprint

Remarks

> The activities today focused on how social media websites contribute to a digital footprint. As we prepare to publish our first web pages, you'll need to think about how those contribute to your digital footprint.

Prompt: With your elbow partner, come up with a checklist that you can use to determine what should, and shouldn't, be posted online.

Share: Allow groups to share out their checklists, using the responses to develop a class-wide web publishing checklist. Consider making a poster of your class-wide checklist that you can refer back to throughout the unit.

Assessment Opportunity

Goal: Use this discussion to create a lasting checklist of best practices for when publishing information online. Possible checks include:

- Could someone identify me with this information?: for example name, address, phone number, etc.
- Do I want everyone to see this?: Don't publish anything you don't want to possibly be published to the world.
- Do I want this to be permanent?: You lose control of information once it's published and it could be around for your entire life.

Students' lists may be different but should cover these principles and additional ones they saw in today's lesson.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- **IC** - Impacts of Computing
- **NI** - Networks & the Internet

Lesson 6: Lists

Web Lab

Overview

Students are introduced to ordered and unordered lists in HTML and work through a few levels in which they use the <ul>, <ol>, and <li> tags. They then go back to their project, where they add a new HTML page. Inside the new page, they write the HTML to display a recipe, top ten list, or any other content that uses the new tags that they have learned.

Purpose

This lesson introduces list elements, which are different from the previous elements in that they involve nested tags. Students should understand that the list item elements (
) go inside the list elements (or).

Students also continue to work on their projects, adding a new HTML page. Students are free to choose the content of this page, but it should include some type of list. Right now there is no way for users to navigate from one project page to another, but students will learn more about how to link the pages together in Lesson 9, right before they publish their project.

Assessment Opportunities

1. **Use the <ol>, <ul>, and <li> tags to create ordered and unordered lists in an HTML page.**

   See levels 5 and 6 in the Code Studio lesson.

2. **Create and name a new HTML page.**

   On the last level of the Code Studio lesson, check that students have created a new web page, given it a name, and put content inside.

Agenda

Warm Up (10 minutes)
- **Quick Share**

Activity (40 minutes)
- **Web Lab: Lists**

Wrap Up (10 minutes)
- **Journal**

View on Code Studio

Objectives

Students will be able to:

- Use the <ol>, <ul>, and <li> tags to create ordered and unordered lists in an HTML page.
- Create and name a new HTML page.

Preparation

- [ ] Have student journals (or project sketches) ready if they will use them to generate ideas for their lists page.

Introduced Code

-
-
-

Teaching Guide

Warm Up (10 minutes)

Quick Share

Remarks

> In the last lesson, we looked at different types of content that we would or would not want to share with the world. Today you'll create an entirely new page for your project, and you'll need to think about the content of that page, but you'll also be using a new structure: lists.

Prompt: Take a few minutes to brainstorm some content that you would like to share with the world that you think could be structured as a list.

Discuss: Students should brainstorm individually, then share with partners, then finally share with the whole class.

Prompt: Sometimes we use numbers to order our lists. Which of these lists do you think should be numbered, and which should not be numbered?

Discuss: Allow students to share their ideas for when numbering is appropriate.

Remarks

> In HTML, there are two types of lists, ordered lists, which have numbers, and unordered lists, which have bullets. We're going to learn how to make both types of lists today so that you can use this structure in your project.

Discussion

Goal: The goal of this discussion is to allow students some time to think about how lists are used, and to think about the difference between ordered and unordered lists before seeing them in HTML. Some possible lists students might come up with might be lists of their favorite things, top 10 lists, a shopping list, etc.

When students are discussing whether lists should be ordered or unordered use examples to highlight the difference. For example a list of finishers in a race should probably be ordered, while a grocery list probably does not need to be. Return to this discussion later in the lesson if need be.

Activity (40 minutes)

Web Lab: Lists

Group: Place students into pairs.

Circulate: Helps students as they work through the first set of levels. Keep track of when pairs make it to Level 5.

Teaching Tip

Pair Programming: More on Pair Programming can be found in the **Computer Science Discoveries Curriculum Guide**.

Code Studio levels

Lesson Overview | Student Overview

Lists | 2 | 3 | 4 | 5 | 6 | *(click tabs to see student view)*

Lists in HTML | Teacher Overview | Student Overview

Review

View on Code Studio

Bring the class back together after all students have completed the previous level. You can track their progress in your Teacher Panel for that level. Review what students have learned about lists in the level progression. The previous level provides an opportunity to review both types of lists and how some common errors can affect how the list is displayed. Students can also use this time to update their HTML tag lists with <ul> , <ol> , and <li> .

Expanding Your Website

8 *(click tabs to see student view)*

Expanding Personal Website

9 *(click tabs to see student view)*

Wrap Up (10 minutes)

Journal

Prompt: Have students update the "HTML Tags" page in their journals with the tags they learned in this lesson.

Prompt: Have students reflect on their development of the**five practices of CS Discoveries** (Problem Solving, Persistence, Creativity, Collaboration, Communication). Choose one of the following prompts as you deem appropriate.

- Choose one of the five practices in which you believe you demonstrated growth in this lesson. Write something you did that exemplified this practice.
- Choose one practice you think you can continue to grow in. What's one thing you'd like to do better?
- Choose one practice you thought was especially important for the activity we completed today. What made it so important?

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▸ **AP** - Algorithms & Programming

Lesson 7: Intellectual Property and Images

Unplugged | Web Lab

Overview

Starting with a discussion of their personal opinions on how others should be allowed to use their work, the class explores the purpose and role of copyright for both creators and users of creative content. They then move on to an activity exploring the various Creative Commons licenses as a solution to the difficulty in dealing with copyright.

Finally, with a common understanding of the restrictions of various Creative Commons licenses, students learn how to add images to their web pages using the <img> tag.

Purpose

Until this point the only content that students have used on their web pages is their own, but it's common, and quite useful, to be able to integrate information and media created by others. Before learning about the technical ways to do this, however, we need to step back and gain a common understanding of the restrictions and purpose of copyright. We use the Creative Commons license as a clearer alternative to the more restrictive standard copyright and guide students through searching for, using, and properly citing Creative Commons licensed media.

Assessment Opportunities

1. **Explain the purpose of copyright.**

   In the wrap up journal prompt, check that students are thinking of reasons that people may want their intellectual property covered under copyright law.

2. **Identify the rights and restrictions granted by various Creative Commons licenses**

   In the activity guide, check that students are choosing licenses and providing explanations consistent with the given scenarios.

3. **Add an image to a web page**

   See Level 5 in the Code Studio lesson.

Agenda

Warm Up (10 min)
- **Use of Your Work**

Activity 1 (20 min)
- **The Creative Commons Solution**

View on Code Studio

Objectives

Students will be able to:

- Explain the purpose of copyright.
- Identify the rights and restrictions granted by various Creative Commons licenses
- Add an image to a web page

Preparation

- Preview **Creative Commons Overview - Video**. You may need to download it before school if Youtube is blocked.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Licensing Your Work** - Exemplar

For the Students

- **Creative Commons Overview** - Video
- **Licensing Your Work** - Activity Guide

Make a Copy

Vocabulary

- **Citation** - A quotation from or reference to a book, paper, or author, especially in a scholarly work.
- **Copyright** - the exclusive legal right to print, publish, perform, film, or record literary, artistic, or musical material, and to authorize others to do the same
- **Creative Commons** - A collection of public copyright licenses that enable the free distribution of an otherwise copyrighted work, used when an author wants to give people the right to share, use, and build upon a work that they have created
- **Intellectual Property** - A work or invention

that is the result of creativity, such as a piece of writing or a design, to which one has rights and for which one may apply for a patent, copyright, trademark, etc.

Introduced Code

- <img >

Teaching Guide

Warm Up (10 min)

Use of Your Work

Remarks

> All of you have been working hard to create a new website that you're going to publish to the world. When you publish something, though, it can be hard to control what other people do with it. Sometimes people use our work in ways that don't seem fair to us. I'm going to describe a few situations for you. If you think what happened was fair, stand up. If you think it was unfair, sit down.

1. You take an awesome picture and someone puts it on their social media account and with your name beside it.
2. You write a story and someone else publishes it and says that they wrote it.
3. You write a song and someone sings it to her friends.
4. You write a song and someone sings it at a concert and makes a lot of money.
5. You take a picture and someone else Photoshops it and puts the new version on his web site.

Teaching Tip

These prompts aim to relate broader questions of copyright to students' own experiences creating and sharing content online. To make this hit close to home you may want to change the prompts to pick particular forms of social media that students are interested in. If you know students don't use a lot of social media, then alter the prompt to reflect another area where your students might think their own creative work was used inappropriately.

Remarks

> It's okay if not everyone agrees how they want their work to be used. Copyright law says that whoever creates new content, such as a picture, a story, or a song, gets to decide how other people are allowed to use it.

Prompt: What rules would you make for people who want to use your creative work?

Discuss: Have students journal individually, then share with a neighbor, and finally discuss as a whole class.

Activity 1 (20 min)

The Creative Commons Solution

Remarks

> Copyright is granted the moment something is created, so unless explicitly told otherwise, we have to assume any picture, video, or other media we find online is fully covered by copyright law, which means that no one else can make copies or post it online without permission.
>
> Sometimes, though, we want people to share our work so that more people can see it. The Creative Commons (CC) license was developed to help content creators have more specific control over how other people can use their work.

Group: Pair students.

Distribute: Give each student a copy of the activity guide.

Licensing Your Work

> **License Components**
>
> The first portion of this activity guide covers the components of a Creative Commons license. Push students to think critically here about both the value each component adds to the creator, but also the ways it might limit the cultural exchange of ideas, connecting the components of the license to the scenarios that they considered earlier in the class.
>
> **Choosing the Right License**
>
> The second portion of this activity provides students with four CC licenses and two content creator scenarios. For each scenario, students are asked to evaluate which of the four provided licenses is the **least restrictive** but still addresses

the concerns and needs of the content creator.

Discuss: Ask several students to share out their responses to the **Choosing the Right License** scenarios. Encourage discussion and debate if students identified different licenses for the same scenarios.

Activity 2 (20 min)

Adding Images

Transition: Send students to Code Studio, where they will learn about the `<img>` tag and have an opportunity to add images to the pages they've already created.

Teaching Tip

Is it Derivative?: Students may struggle with whether the use of a work is derivative or not. The rule of thumb is to consider whether your use of a licensed work would be considered a **new** copyrightable work.

Converting an image from digital to print, or a song from CD to mp3 would not create a derivative work, but animating a drawing or adding new lyrics to a song **would** be creating derivative works.

Discussion

Goal: While there may not be a specific "right" license for each of these scenarios, encourage students to always consider whether the license they chose is more restrictive than it strictly needs to be. One of the design goals of the Creative Commons license is to increase the amount of creative material available to the general public, promoting the sharing of intellectual property for the common good. With this in mind we should only be adding the minimal restrictions needed to meet the content creator's wishes.

Content Corner

As students start to use images, they may have trouble finding the right size to fit their pages. In the second chapter, students will learn how to resize their images with CSS. For now, students can use an online image resizer, such as the one found at **https://resizeimage.net/**, to resize their images. Alternatively, they can use the `height` or `width` attribute inside the `img` tag to resize it as the page loads.

Code Studio levels

Lesson Overview

Teacher Overview | Student Overview

View on Code Studio to access answer key(s)

View on Code Studio

Adding Images

2 | 3 | 4 | 5 *(click tabs to see student view)*

Images in HTML

Student Overview

Creative Commons Search

Teacher Overview | Student Overview

View on Code Studio

Recommended Search Engines

Because we can't know which sites might be blocked in your district, we've avoided pointing students to a specific search engine. Not all search engines make it easy to set Creative Commons filters - some of the easiest include:

- **Creative Commons Search**
- **Google Image Advanced Search**
- **Wikimedia Commons**
- **Flickr Creative Commons**

As with any site with crowdsourced content, search engine results will change from day to day, and some of those results may be inappropriate for the classroom. You may want to check these sites shortly before displaying them to the class, and decide what is best for your classroom.

Attribution

8 | 9 *(click tabs to see student view)*

Wrap Up (5 min)

Your Own CC License

Journal: Have students update their "HTML Tags" list with the tags they learned in this lesson.

- **Journal:** Think of all the personally created items that you've put on your website so far, and those that you may add in the future. Without a clear license all of those of things (and your web pages themselves) are covered under the fully restrictive copyright. Which Creative Commons license would you rather use (if any) and why?

  If there is time you may want to consider having students return to their site and add the Creative Commons license they decided they would like their website to have.

  - Check out the **Creative Commons license chooser**

  Students can add a footer at the bottom of each page with their Creative Commons License.

Assessment Opportunity

As students discuss the advantages and disadvantages of various types of licenses, make srue that they recognize that copyright is intended to protect the rights of content creators, such as themselves. These righs include being recognized as the creator when the work is displayed, or deciding who can use and make money off the work.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▸ **AP** - Algorithms & Programming
- ▸ **IC** - Impacts of Computing

Lesson 8: Clean Code and Debugging

Web Lab

Overview

Students deal with common issues that arise when designing web pages in HTML. Students will correct errors in a sequence of increasingly complex web pages. In the process they will learn the importance of comments, whitespace, and indentation as tools for making web pages easier to read. At the end of the lesson students create a list of strategies for debugging web pages and ensuring they are easy to read and maintain.

Purpose

Bugs in HTML are more forgiving than programming languages such as JavaScript (the language used in Unit 3). However debugging is an explicit problem solving process that students will use repeatedly when working with any language on the computer. When problem solving there are different strategies that a computer scientist can use to find the source of the issue.

In addition to the strategies to fix bugs once they have occurred there are certain styles of writing HTML code that help prevent bugs or make it easier to find bugs. The three main style conventions used are comment, whitespace, and indentation. To motivate students to consider using these conventions in the future, the debugging levels demonstrate that it is easier to debug a program that is written with these style conventions.

Assessment Opportunities

1. **Describe why using whitespace, indentation, and comments makes your code easier to maintain**

   In the final class discussion, make sure students mention the clean code practices as described in the Code Studio lesson.

2. **Develop a set of techniques for preventing bugs in HTML code and finding them when they occur**

   Check the list of tips that students create at the end of the main activity.

Agenda

Warm Up (10 minutes)

- **Previous Experience with Bugs**

Activity (35 minutes)

- **Web Lab: Smash Those Bugs!**

View on Code Studio

Objectives

Students will be able to:

- Describe why using whitespace, indentation, and comments makes your code easier to maintain
- Develop a set of techniques for preventing bugs in HTML code and finding them when they occur

Preparation

- [ ] Prepare poster paper, sticky notes, and markers

Vocabulary

- **Bug** - Part of a program that does not work correctly.
- **Comment** - A note in the source code of a computer program that helps explain the code to people who read it
- **Debugging** - Finding and fixing problems in an algorithm or program.
- **Indentation** - The placement of text farther to the right or left of the surrounding text, making it easier to understand the program's structure
- **Whitespace** - Any character that shows up as a blank space on the screen, such as a space, a tab, or a new line; helps separate different parts of the document to make it easier to read

Introduced Code

- <!-- -->

Wrap Up (10 minutes)

- **Coding Style Conventions**

Teaching Guide

Warm Up (10 minutes)

Previous Experience with Bugs

Set up: Put a poster up on the wall where all students can get to it. Write the title "Class Bugs" at the top but wait to explain the meaning of the term until it is introduced below.

Group: Place students in groups of 3-4.

Distribute: Give each group a handful of sticky notes

Prompt: With your group:

- Come up with at least three specific problems you have encountered while trying to write web pages in HTML.
- What project were you working on?
- How did you ultimately track down and fix the problem?

Discuss: Have students share out the bugs they have faced and strategies they have used for finding them.

Vocabulary: Introduce the concept of bug and debugging explaining them in the context of the problems and strategies students shared.

Share: Ask each group to use the sticky notes they were given to write down at least three of the bugs they've encountered and how they solved them, using one sticky per bug. Each sticky should have:

1. A brief description of the bug
2. Steps taken to solve it
3. Name of student who solved it

Once groups have written down their bugs, have them stick them up on the class poster.

Remarks

> The problems you had when your HTML code did not work correctly are bugs. The process of fixing bugs and strategies used to fix them is called debugging. Today we're going to be working on our debugging skills and learning some strategies to keep our code clean to help avoid them.

Discussion

Goal: Students do not need to get all the bugs or the strategies in this first discussion. They will add to this list throughout class.

Bugs students might share could include:

- Closing and opening tag switched
- Not closing a tag
- Not putting quotes around attributes values
- Spelling a tag name wrong
- Not putting a list item inside a type of list element
- Using the wrong header tag
- Wrong file path for an image
- Forgetting the equals sign between attribute name and value
- Closing tags out of order
- Not putting content inside a tag

Strategies students might share for debugging could include:

- Guess and Check
- Taking out sections to see which section is causing problems
- Looking for missing tags
- Figuring out what section of the code has the problem (Making the problem smaller)
- Asking a friend for help
- Thinking about when the code last worked and what you have added since then

Teaching Tip

Using the Bugs Poster: Encourage students to use this bugs poster as a resource through the rest of the unit. Whenever a student successfully squashes a new bug, have them put it on the poster. When students are get stuck, they can check the bug poster for someone who may have encountered the same issue before.

Use Journals: If you like you can have sutdents individually track their bugs on a new page in their journals, just like the "HTML Tags" page that they have been updating.

Activity (35 minutes)

Web Lab: Smash Those Bugs!

Group: Put students in pairs to work on these Code Studio levels.

Code Studio levels

Lesson Overview | Student Overview

Debugging | 2 | 3 | 4 | 5 | 6 *(click tabs to see student view)*

Formatting HTML | Student Overview

Challenge | 8 *(click tabs to see student view)*

- **Prompt:** As student finish the debugging challenges, have them work in teams to discuss the strategies that they used to debug the code in the activity, as well as the debugging examples they came up with at the beginning of class. They can them write down tips for debugging code in the future, either in their journals or to share as a class poster.

Assessment Opportunity

As students work in groups, circulate to assess the strategies that students have come up with. Check to make sure that the strategies are general enough to be useful ("double check spelling on image file names" vs. "make sure you didn't misspell Mexico") but specific enough to provide guidance ("double check spelling on image names" versus "double check your code").

Wrap Up (10 minutes)

Coding Style Conventions

Prompt: What made it harder or easier to debug the web pages?

- **Discuss:** Have students share out things that made it easier or harder to debug the web pages they encounter.

Remarks

> Code needs to be useful for both people and computers. Code that your computer can run might still be really hard for someone (or even you!) to read and make changes to. From now on it's important that we use these practices to ensure our code is easy to read for people, not just good enough for a computer to use.

Assessment Opportunity

**Students answers will vary but should hopefully include the following:

- Number of bugs
- Use of comments (text between the characters <!-- and -->)
- Separating things onto separate lines (whitespace)
- Grouping together things that are one idea such as a list (whitespace)
- Indenting elements that are inside other elements

If students don't mention some of these things ask them to compare two sites that have different uses of these elements.

Setting Expectations: Use this discussion to motivate the need for making readable code. Highlight that you will expect them to follow the norms they learned today from now onwards, in particular on their projects for this unit.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▸ **AP** - Algorithms & Programming

Lesson 9: Project - Multi-Page Websites

Web Lab | Project

Overview

After learning about how to link web pages to one another, students are finally able to publish the website they have been working on. In this lesson, they link together all the previous pages they have created into one project, create a new page, and add navigation between the pages before publishing the entire site to the Web.

Purpose

In several of the previous lessons, students created web pages that will be included in their websites. In this lesson, they will learn to link all of these pages together and publish them to the Internet.

Assessment Opportunities

Use the project rubric attached to this lesson to assess student mastery of learning goals of this chapter.

Agenda

Warm Up (15 minutes)

- **Content Brainstorm**

Activity

- **Multi-Page Websites**
- **Peer Review**

Wrap Up (10 minutes)

- **Journal: Personal Website**

View on Code Studio

Objectives

Students will be able to:

- Connect multiple web pages into one website using hyperlinks.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Multi-Page Website** - Exemplars

For the Students

- **Multi-Page Website** - Project Guide
  Make a Copy
- **Multi-Page Website** - Rubric
  Make a Copy
- **Multi-Page Website** - Peer Review
  Make a Copy
- **Computer Science Practices** - Reflection
  Make a Copy

Vocabulary

- **Hyperlink** - A link from a HTML file to another location or file, typically activated by clicking on a highlighted word or image on the screen.

Introduced Code

- <a>

Teaching Guide

Warm Up (15 minutes)

Content Brainstorm

Remarks

> In this lesson, you'll be publishing your website to the Internet so that anyone with the address can see it. Before you do that, though, you're going to have a chance to add anything that you think might be missing from your site.

Prompt: What are some things you want on your site that you haven't had a chance to add yet?

Give students a few minutes to brainstorm some ideas, then allow them to share with the class.

Remarks

> In a moment, you'll be able to create new pages using any of these ideas. Before we do, though, we need to make sure that our users will be able to get to all of the pages on our sites. To do that, we'll need to use a hyperlink.

Activity

Multi-Page Websites

Distribute the **Personal Website - Project Guide** and review the first page as a class.

Circulate: Support students as they complete the first two pages of the Project Guide.

Transition: When students are finished designing their pages, have them transition to Code Studio.

Teaching Tip

Adding Multiple Pages: Students who decide to add more than one web page will need extra paper to complete their sketches. Alternatively, you may choose to have students complete their sketches and image tables in their journals.

Teaching Tip

As students are linking their pages together, they may notice that some links appear to be broken and lead only to a blank page in Web Lab, even if they appear to work when shared. If this happens, check carefully for any other errors on the page that you are linking from. For security reasons, Web Lab is a little more strict than the general browser, and it will often prevent pages with errors from linking to other pages.

Code Studio levels

- Lesson Overview
- **Teacher Overview**
- **Student Overview**

View on Code Studio to access answer key(s)

View on Code Studio

- Levels
- **2**
- **3**

View on Code Studio

Student Instructions

View on Code Studio

Making a Multi-page Site

A web site should include more than one web page, and they should connect to each other. Here's an example of a project with multiple pages that are connected by hyperlinks (the blue underlined words that take you to another page when you click on them).

Do This

- Use the inspector tool to see the code that creates the hyperlinks on the top of index.html.
- Discuss with your partner how you think that code works.
- Click on recipe.html in the files list on the left hand side. Add a hyperlink to the top of this page that will take you back to index.html.

Student Instructions

View on Code Studio

Create a Navigation Bar

Here's your project from before. You'll be linking the two pages by creating a **navigation bar** at the top of the page. Websites usually have the same navigation bar across all of their pages so you can get back and forth between things on the site. You can see an example of a navigation bar on **this page**

Do This

- In your home page (index.html), add two hyperlinks at the top of the page, one to your lists page, and one to the home page.
- **Even though it might seem strange to link a page to itself, it's less confusing if the navigation bar is exactly the same on every page.**
- Copy the code for the navigation bar, then add it to the top of your lists page.
- Test your links in the preview area by clicking on them.

- Hyperlinks
- **Student Overview**

- Project Guide
- **Teacher Overview**
- **Student Overview**

View on Code Studio

Planning Checkpoint

View on Code Studio

Check over student plans before allowing them to continue on the rest of the levels.

Your Personal Website

View on Code Studio

In the next few levels, you'll be completing and publishing your personal website.

You've already done a lot of work on your website, but this is your chance to get creative and add a new page of something that you care about.

Here are some ideas for your new page, but you can do anything approved by your teacher.

- A movie, television, or book review
- Description of a favorite holiday tradition
- A "cause" page that explains an issue that is important to you

If you have more than one idea, you can add multiple pages to your site.

Your Project Guide will help you remember everything you need to do. Make sure you have your teacher's approval before moving to the next level.

Examples

- Levels
- 6
- 7
- 8
- 9
- 10

Student Instructions

View on Code Studio

Create and Name Your New Page

Before you start coding, you should make sure that you have all the pages that you need in your project.

Do This

- Create a new HTML page and rename it. (Look at **Lesson 6 Level 8** if you've forgotten how.)
- Repeat this process for any other pages you want to add.

Student Instructions

View on Code Studio

Upload Images

Next, you'll need to upload all of your images into your project.

Do This

- Download all the images you want onto your computer. Visit **Lesson 7 Level 5** for help if you need it.
- Make sure you have the right to use these images.
- Write down all of the attribution information that you need.
- Upload all of the images into your project. Visit **Lesson 7 Level 5** if you've forgotten how.

Student Instructions

View on Code Studio

Add HTML

Next, you should add the HTML to your pages.

Do This

- For each of the new pages that you've created, add HTML to make the page you sketched in your project guide.
- You may also add more HTML to the pages you've already created.

Student Instructions

View on Code Studio

Navigation Bar and Footer

Each of your pages needs a navigation bar and a footer.

Do This

- For each page, make sure it has a navigation bar that links to other pages in your site. For a review of the navigation bar, go to **Lesson 9 Level 4**.
- For each page, make sure it has a footer at the bottom that explains the copyright information for your page. (Go to **Creative Commons** for help in creating this HTML.)

Student Instructions

View on Code Studio

Review Your Project

You're almost ready to publish your project, but you should check one more time that you have fulfilled all the requirements for this project.

Do This

- Check that you are comfortable sharing all the information that's on your site.
- Check that all of your images are correctly attributed, and that you have the right to put them on your site.
- Check that your project includes everything included in the project rubric.
- Use comments and whitespace to make your website readable.
- Ask your partner to help you find any potential problems with your site.

- Publishing Your Website
- **Teacher Overview**
- **Student Overview**

Discussion Goals

View on Code Studio

The video content is fairly straightforward, but this is your last check in with students before they publish, so it's a good time to make sure they have full thought through the implications of publishing to the entire world. They should make sure that they are not sharing any sensitive information, and that they have the rights to use all the content on the page.

As they began to share, they should also think about how they respect the work of other website publishers, in particular their classmates. You may want to take some time to establish classroom norms around how they talk about and give feedback on their classmates' sites, as well as guidelines for using code that others have written.

Questions to Consider

View on Code Studio

- What do you think your responsibilities are as a website publisher?
- Why might you want to see the code or remix someone else's site?

Peer Review

Group: Place students in pairs.

Distribute: Personal Website - Peer Review

Support: Help students as they complete the Peer Review Guide, then allow students to make any final revisions to their websites.

Collect: Project Guides, Peer Review Guides, and student website addresses.

Wrap Up (10 minutes)

Journal: Personal Website

Prompt: After the first day of pulling together your personal website reflect on your experience.

- How did you use the problem solving process to in creating your site?
- What do you like about your site so far?
- What do you still want to know how to do?

Discussion

Goal: Students should reflect on the process so far and keep track of the fact that this is an iterative process where they are constantly improving things as they go.

Send students to Code Studio to complete their reflection on their attitudes toward computer science. Although their answers are anonymous, the aggregated data will be available to you once at least five students have completed the survey.

Code Studio levels

- CSD U2 project share_2019
- **Student Overview**

Share Your Project

View on Code Studio

It's time to share your project with the world!

Do This

- Click the Share button at the top of this page.
- Copy the URL it gives you, and email it to the people you want to see your page.

Discuss: Have students share out the things they still want to learn how to do.

Keep this list and cross it off as they learn more things in the unit.

Journal: Have students add the new tags they learned to the "HTML Tags" section of their journal.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- **AP** - Algorithms & Programming
- **IC** - Impacts of Computing

Lesson 10: Styling Text with CSS

Web Lab

Overview

This lesson introduces CSS as a way to style elements on the page. Students learn the basic syntax for CSS rule-sets and then explore properties that impact HTML text elements. They work on a HTML page about Guinness World Record holders, adding their own style to the provided page. In the last level, students apply what they have learned about styles for text elements to their personal web page.

Purpose

While there are many CSS properties, the properties highlighted in the levels are simple properties that can style text elements. Students will use these properties often. More properties are covered in subsequent lessons.

Assessment Opportunities

1. **Use CSS selectors to style HTML text elements.**

   See Level 5 in Code Studio.

2. **Create and link to an external style sheet.**

   On Level 13, ensure that students have successfully created and linked to a stylesheet.

3. **Explain the differences between HTML and CSS in both use and syntax.**

   See the review at the main activity. You may also want to collect a T-chart from each group.

Agenda

Warm Up (5 minutes)

Journal: HTML Appearance

Activity (40 minutes)

Web Lab: Introduction to CSS

Wrap Up (10 minutes)

Recording CSS Properties

View on Code Studio

Objectives

Students will be able to:

- Use CSS selectors to style HTML text elements.
- Create and link to an external style sheet.
- Explain the differences between HTML and CSS in both use and syntax.

Preparation

- [ ] Create a new poster titled **CSS Properties** if your students will not be tracking new CSS properties in their journals

Vocabulary

- **CSS** - Cascading Style Sheets; a language used to describe how HTML elements should be styled
- **CSS Selector** - the part of a CSS rule-set that defines which HTML elements the style should be applied to

Introduced Code

- text-decoration: value
- text-align: value
- font-size: value
- font-family: value
- color: value

Teaching Guide

Warm Up (5 minutes)

Journal: HTML Appearance

Prompt: In the past few lessons, we've been using HTML to write our web pages. HTML lets us use tags to define the structure of a page. With your partner, come up with a list of all the different HTML tags as you can recall, and what they mean.

Discussion

Goal: Students may mention that the <h1> tag has large font, or that the <li> tag puts a number or bullet before the text. Some other options that they may not consider are that all the text appears in black, and uses the same font. None of the text is underlined or bolded. Prompt students to realize that there are other possible styles that they see on web pages that are not the default appearance when HTML is used.

Discuss: Choose one of the tags that describes text on the screen (<p> , <li> , <h1> , etc.) and have the students describe how the contents of that tag appear on a web page. Ask students whether they would **always** want those elements to appear in that exact way. For example, <p> always means that the content is a paragraph, but should paragraphs always look the same on every page and on every site?

Remarks

> So far we have only made web pages where we control the content and structure, such as which parts of the pages are headers, lists, or paragraphs. We've been using HTML as the language to specify the content and structure of the pages. While HTML allows us some control over how the page looks, it doesn't give developers much control over the specific look and style of each element. To do that, we need a language to express **style**. Web developers use CSS to specify the style of a page.

Activity (40 minutes)

Web Lab: Introduction to CSS

Transition: Send students to Code Studio.

Code Studio levels

Lesson Overview

Student Overview

Exploration: Adding Style

Teacher Overview | Student Overview

Have students explore this level with a partner. Afterwards lead a short share out so that partners can share with the class their responses to the three questions in the instructions.

View on Code Studio

The subsequent video should help reinforce what students discovered so there's no need to lead a lengthy debrief conversation.

Video: Intro to CSS

Teacher Overview | Student Overview

View on Code Studio

Discussion Goals

Students should understand that the structure of the page is there to organize the information in a logical way, but doesn't tell the computer how to display it. So, for example, the structure could include different headings and paragraphs, but would say nothing about the color of text or how big it is displayed. The style of the page is the specifics of its appearance. Without a particular style, each web browser would decide how to display different web page elements on their own.

Styles are important to allow web developers to decide exactly how a web page looks on the screen. Because styles are separate from structure and content, web developers can change the style of an entire page very easily, without having to make any changes to the structure and content of the web page. This means that it's very easy for web pages to have an individual look and feel that is unified across the entire page.

Styling Text

4 | 5 *(click tabs to see student view)*

CSS and Style Sheets

Teacher Overview | Student Overview

View on Code Studio

Introducing Content, Structure, and Style

This map level, and in particular the first section, introduces the Content-Structure-Style paradigm that students will use to help understand the differences between HTML (HyperText Markup Language) and CSS (Cascading Style Sheets). While they have previously been using HTML tags to label and categorize their content, they're now learning how to control the style of the different elements in their page using CSS. Students may have some misconceptions at this point since their browser has used default styles for their different kinds of content (e.g. <h1> tags by default make text larger). As they move through these lessons they should come to see that tagging content really just allows the browser's default styles, or the styles they define using CSS, to be applied to the correct pieces of content in their web pages.

Some students may also be confused by the fact that "Content-Structure-Style" and "Cascading Style Sheets" have the same initial letters. CSS stands for Cascading Style Sheets and refers to the language they will use to style their content. Content-structure-style is a paradigm that they use to separate out the different information that the browser needs to render a page, but should not be abbreviated "C-S-S", to avoid confusion.

Fonts and Style

7 | 8 | 9 | 10 *(click tabs to see student view)*

CSS Text Properties

Teacher Overview | Student Overview

View on Code Studio

Differences in Browsers and Operating Systems

As mentioned in this map, you may find that different combinations of computers and browsers render fonts differently. While there is a common specification for how HTML and CSS should be read and displayed on the screen, each browser does so in a slightly different way, which can lead to the same page looking different on different computers. If your students encounter this, you can reassure them that it's a common challenge that web developers face.

In the case of `font-family: fantasy`, you can actually specify a list of fonts, from most to least specific. That way, if a user's computer has the exact font you specified, that's what they'll see, but if not the browser will try the next font on your list. To try and ensure that users see the font in the example image, you could write the css as `font-family: Papyrus, fantasy`. That way, even if a user's browser has a different default font for "fantasy", it will try to load the specific font "Papyrus" first.

If students want the page to show an exact font, they'll need to use a font from the web, so the browser can download that specific font to use when rendering the page. More information on these fonts can be found at **Google Fonts** and **W3Schools**.

Video: Intro to CSS Part 2

Teacher Overview | Student Overview

View on Code Studio

Discussion Goals

Make sure all the students understand how to link to their stylesheet from each web page.

Styling your Website

13 | 14 *(click tabs to see student view)*

Review: Briefly review the "Content-Structure-Style" paradigm introduced in the map level during the lesson. Draw a T-chart on the board and label one side HTML and one side CSS. Have students work in small groups to think of as many differences they can between the two languages, then come back together as a group and share.

Assessment Opportunity

Make sure students are distinguishing between how HTML indicates the structure of a document and CSS now allows students to set the styles, as well as the differences in how the languages look on the screen and where they are used.

Wrap Up (10 minutes)

Recording CSS Properties

Set Up: Have students create a new page in their journals called CSS Properties where students can easily get to it.

Group: Place students in groups of two to five - you'll need at least one group for each of the properties introduced in this lesson.

Teaching Tip

Journal or Poster? Just as with the "HTML Tags" page in their journals, you may choose to have your class keep track of CSS Properties in a shared class poster.

Jigsaw: Assign each group one of the properties introduced today. Each group needs to come up with the a description and example for their property.

Share: Have groups add their properties to their journals or to the class "CSS Properties" poster.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- **AP** - Algorithms & Programming

Lesson 11: Styling Elements with CSS

Web Lab

Overview

This lesson continues the introduction to CSS style properties, this time focusing more on non-text elements. Students begin this lesson by looking at a website about Desserts of the World. They investigate and modify the new CSS styles on this website, adding their own styles to the page. After working on the Desserts page, students apply their knowledge of new CSS properties to do more styling of their personal websites.

Purpose

This lesson builds on what students previously learned about CSS properties, this time looking at properties that can be used all elements. These new properties impact the visual look of the web page beyond its text and give students more precise control over the layout of their pages. As students expand their knowledge of style properties, many may have questions about how they can do even more with CSS. Curious students can explore more properties at **W3 Schools** .

Assessment Opportunities

1. **Use CSS properties to change the layout properties of elements.**

   See Level 11 in Code Studio.

2. **Create a CSS rule-set for the body element that impacts all elements on the page.**

   See Level 11 in Code Studio

Agenda

Warm Up (5 minutes)

Discuss: What Styles Do You Want?

Activity (40 minutes)

Web Lab: Styling Elements with CSS

Wrap Up (5 minutes)

View on Code Studio

Objectives

Students will be able to:

- Use CSS properties to change the size, position, and borders of elements.
- Create a CSS rule-set for the body element that impacts all elements on the page.

Introduced Code

- width: value
- margin: value
- height: value
- border-width: value
- border-style: value
- border-color: value
- border-radius: value;
- background-color: value
- float: value

Teaching Guide

Warm Up (5 minutes)

Discuss: What Styles Do You Want?

Prompt: Yesterday we styled text elements. What other ways do you wish you could style your page?

Discuss: Have students share styles they would like to be able to add their page.

Remarks

> Today we are going to learn more properties we can use to style our web pages. Hopefully some of them allow you to add some of the styles we listed in this discussion.

Discussion

Goal: Students will likely bring up:

- Adding background colors to things
- Being able to move things around on the page (center, left, right)
- Being able to add borders to things
- Being able to control the size of objects

Activity (40 minutes)

Web Lab: Styling Elements with CSS

Transition: Send students to Code Studio.

Teaching Tip

Pair Programming: Consider using pair programming for some or all of this lesson, in particular as students are learning new CSS properties.

Code Studio levels

Lesson Overview — Student Overview

Exploring more CSS — 2 | 3 *(click tabs to see student view)*

The Body Element — Student Overview

Layout with CSS — 5 | 6 | 7 | 8 | 9 | 10 | 11

(click tabs to see student view)

Layout Properties — Student Overview

Styling your Website — 13 *(click tabs to see student view)*

Wrap Up (5 minutes)

Journal: Have students add the new properties they learned to the CSS Properties page in their journal or the class poster.

Prompt: Have students reflect on their development of the **five practices of CS Discoveries** (Problem Solving, Persistence, Creativity, Collaboration, Communication). Choose one of the following prompts as you deem appropriate.

- Choose one of the five practices in which you believe you demonstrated growth in this lesson. Write something you did that exemplified this practice.
- Choose one practice you think you can continue to grow in. What's one thing you'd like to do better?
- Choose one practice you thought was especially important for the activity we completed today. What made it so important?

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▸ **AP** - Algorithms & Programming

Lesson 12: Sources and Search Engines

Overview

This lesson encourages students to think more critically about how web searches work and how to find relevant and trustworthy information online. After viewing and discussion a video about how search engines work, students will search for information about several unlikely animals. They'll need to analyze the sites they find for reliability in order to identify which of the animals is actually a hoax.

Purpose

As students are finishing up their own websites, this lesson encourages them to also think about their responsibilities as consumers of information. By the end of this lesson, students should have developed strategies for determining which websites are more trustworthy and tie these strategies back to their own role of content producers by looking for ways to make their own sites appear more trustworthy.

Assessment Opportunities

1. **Use basic web searching techniques to find relevant information online**

   In the Strange Yet True Animals activity guide, check that students have filled out the table. It's not necessary that they complete every cell, but they should be able to find the answers to most questions on the web.

2. **Identify elements that contribute to a website's trustworthiness or untrustworthiness**

   Use the students' trustworthiness checklist and discussions around it to get a broader sense of the elements students can identify, as well as the final journal prompt for more individual reflection about how a student could use those elements in a personal website.

Agenda

Warm Up (10 minutes)

Internet Scavenger Hunt

Activity (30 minutes)

Relevant and Trustworthy Sites
The Trustworthiness Checklist

Wrap Up (5 minutes)

Human versus Computer Searching

View on Code Studio

Objectives

Students will be able to:

- Use basic web searching techniques to find relevant information online
- Identify elements that contribute to a website's trustworthiness or untrustworthiness

Preparation

- [ ] Print a copy of the Internet Scavenger Hunt activity guide for each pair of students.
- [ ] Print a copy of the Strange But True Animals Activity Guide for each pair of students.
- [ ] Create a blank poster titled **Trustworthiness Checklist** and place it on the wall.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Strange Yet True Animals** - Exemplar

For the Students

- **How Search Works** - Video (**download**)
- **Internet Scavenger Hunt** - Activity Guide **Make a Copy**
- **Strange Yet True Animals** - Activity Guide **Make a Copy**

Vocabulary

- **Algorithm** - A precise sequence of instructions for processes that can be executed by a computer
- **Relevant** - Closely connected or appropriate to the matter at hand
- **Search Engine** - A program that searches for and identifies items in a database that correspond to keywords or characters

specified by the user, used especially for finding particular sites on the World Wide Web.

- **Trustworthy** - Able to be relied on as honest or truthful

Teaching Guide

Warm Up (10 minutes)

Internet Scavenger Hunt

Group: Place students in pairs.

Distribute: Each group will need a copy of the Internet Scavenger Hunt activity guide.

Transition: Send students to the computers to prepare for the scavenger hunt.

Remarks

> When I start the clock you will have exactly 7 minutes to complete as much of this scavenger hunt as you can. Your goal isn't to answer **every** question on this list, but to find as much **accurate** information as possible in the time you've been given.

Prompt:

- Which things were hardest to find?
- Which were easiest to find?
- What strategies did you use to find things online?
- How did you know the information you found was correct?

Discussion

Goal: The first three prompt questions are intended to help students identify any searching techniques that they currently use. If students **don't** have clear strategies, you may want to spend some time discussion basic search techniques. Keep track of search strategies on the board for reference later.

The last question gets to the focus of the activities in this lesson. It's likely that students didn't consider whether or not their information was accurate, but if they do come up with some ideas, make sure that you track them in a place where they can be referred to later in the lesson.

Discuss: Student pairs share out what they were able to discover about the scavenger hunt and what strategies worked well for finding information.

Activity (30 minutes)

Relevant and Trustworthy Sites

Display: Watch the **How Search Works - Video** with the class.

Remarks

> Now that we know a little bit more about how search engines work, we're going to search for some information about some of the world's strangest animals.

Group: Place students in pairs.

Distribute: Give each pair one copy of the Strange But True Animals activity guide.

Unlikely Animals

> **Search:** Give student pairs some time to search online and fill in the table on the activity guide. They should have little difficulty finding this information by just searching for the names of each animal.
>
> **Share:** Ask student pairs to share out their answers for each of animals. If, during any point in the share out, others in the class disagree with an answer encourage them to discuss where and how they found their information.
>
> **Vocab:** It's likely that students found websites that were relevant to their topics, but how do we know if they are trustworthy?

Teaching Tip

The purpose of this exercise is not to trick the students, but rather to give them an opportunity to explore which aspects of websites can help users to evaluate their trustworthiness. If students have completed similar activities before, encourage them to use what they have learned to complete the activity to more depth, explaining in more detail how they can decide whether a website is trustworthy.

Remarks

> You did a great job finding websites that wererelevant to your search. You found a lot of interesting information about those strange animals, but is any of it real? You and your partner have five minutes to decide which of these animals are real, and which are made up. For each of your decisions you'll need some supporting evidence that shows why your source websites are either trustworthy, or untrustworthy.

Search: Give students five more minutes to look through their source sites to try to figure out whether or not they're trustworthy.

Share: Allow each pair to share out their assessment of trustworthy and untrustworthy sites. Push students to give detailed reasons for why they trust a site or not, and keep track of them on the board. Once all of the pairs have had a chance to share, reveal the that the Tree Octopus was the only fake animal. Depending on how accurate your students were, consider visiting the site as a class to explore some of the clues that it's not quite

Teaching Tip

Students may come upon sites that explain that the pacific northwest tree octopus is in fact a fictional animal. If so, you can ask them to think about why they believe some sites over others, and how they are sure that the other two animals are real. Have them continue filling out the worksheet with these questions in mind, and let them know that they will be able to share their thoughts later in the activity.

You can find more untrustworthy websites in**this forum post**.

Teaching Tip

Just in Case: The Pacific Northwest Tree Octopus is the fake animal.

The Trustworthiness Checklist

Set Up: Start a poster on the wall labeled **Trustworthiness Checklist**.

Group: Place students in groups of 4-5.

Prompt: Now that we've identified some sites that are, and are not, trustworthy, work with your group to come up with a list of things to check for when trying to determine whether or not to believe a website.

Assessment Opportunity

Check students' lists to make sure that they are coming up with criteria that relate to a site's trustworthiness. For example, students might mention that trustworthy sites have clear contact information, cite their sources, tend to have fewer and less invasive ads, and may be better designed than less credible sites. The domains .edu and .gov are reserved for schools and the government, and may be more credible.

Share: Have groups share out their lists. As a class, generate a class Trustworthiness Checklist on a poster. You can refer back to this in the future whenever asking students to find information online.

Wrap Up (5 minutes)

Human versus Computer Searching

Journal: Think about your own websites - do you think people would find your site trustworthy or untrustworthy? What changes could you make to your site to encourage users to trust your content?

Assessment Opportunity

Students should connect the criteria from the lists that they have created to their own sites. While they may not be able to control the web address of their site, they should be able to identify design elements and content features that will make their sites more trustworthy, such as orderly layout and properly citing sources.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- **IC** - Impacts of Computing

Lesson 13: RGB Colors and Classes

Web Lab

Overview

In this lesson, students first learn how to specify custom colors using their RGB (red, green, blue) values. They then apply these colors to a new Four Seasons web page, which uses CSS classes. CSS classes allow web developers to treat groups of elements they want styled differently than other elements of the same type. Using classes, students add more styles to the Four Seasons web page, then use them to style their personal websites.

Purpose

Up until this point, the only styling students have been able to do is styling by element, which means that every element of a particular kind has the same style. Classes allow for web developers to group together a set of elements they want to style. This means students can single out a certain element they want to style or group together elements from one or more types of elements. Once elements are in a class, the class can be used as a selector in a style rule.

Note: Single elements can also be selected by id, but this type of selection is possible with a class applied only to that single element. Because id selection does not add any extra functionality, it is not taught in this course.

Assessment Opportunities

1. **Group elements using classes in order to create more specific styles on their website.**

   See Level 10 on Code Studio.

2. **Apply the rgb() color function to add custom colors to their website**

   In the Activity Guide, check the table at the bottom of the page that the CSS colors are close to the exemplar. They do not need to be exactly the same.

Agenda

Warm Up (5 minutes)
- **RGB Colors Matching**

Activity (40 minutes)
- **RGB and Classes**

Wrap Up (5 minutes)
- **Journal: Reflecting on the Practices**

View on Code Studio

Objectives

Students will be able to:

- Group elements using classes in order to create more specific styles on their website.
- Apply the rgb() color function to add custom colors to their website

Preparation

- [ ] Print a copy of the activity guide for every student.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **RGB Colors** - Exemplar

For the Students

- **RGB Colors** - Activity Guide

  Make a Copy

Vocabulary

- **CSS Class** - An identifier that allows multiple elements in an HTML document to be styled in the same way

Introduced Code

- rgb(red, green, blue)

Teaching Guide

Warm Up (5 minutes)

RGB Colors Matching

Remarks

> So far, you've been using color names to choose colors for your web pages. That means that you can only use the colors that someone else has already thought of. Today, you're going to learn how to mix new colors for your pages.

Distribute: Hand out the activity guide.

Transtion: Send students to Code Studio to use the RGB widget at the beginning of this progression to complete the activity guide. You may want to answer one question together, then allow students to work in pairs to complete the first activity.

Remarks

> On the computer screen, we mix colors by combining red, green, and blue. In the next few levels, you'll have a chance to use the color mixer to choose your own colors, then put them into a web page.

Activity (40 minutes)

RGB and Classes

Transition: Once students have completed the warm up, allow them to continue working through the remaining levels.

Code Studio levels

Lesson Overview | Teacher Overview | Student Overview

View on Code Studio to access answer key(s)

View on Code Studio

RGB Widget | Student Overview

RGB Exploration | 3 | 4 | 5 | 6 *(click tabs to see student view)*

CSS Classes | Student Overview

Styling with Classes | 8 | 9 | 10 | 11 *(click tabs to see student view)*

Wrap Up (5 minutes)

Journal: Reflecting on the Practices

Journal: Have students update their "CSS Properties" page in their journal with the new information they've learned about classes.

Journal: Have students reflect on their development of the five practices of CS Discoveries (Problem Solving, Persistence, Creativity, Collaboration, Communication). Choose one practice you thought was especially important for the activity we completed today. What made it so important?

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- **AP** - Algorithms & Programming

Lesson 14: Project - Final Personal Website

Web Lab | Project

Overview

Students have spent a lot of time throughout the unit working on their Personal Website. In the final couple of days students finalize their websites. They work with peers to get feedback, put the finishing touches on the websites, review the rubric and reflect on their process. To cap off the unit, they will share their projects and also a overview of the process they took to get to that final design.

Purpose

This project emphasizes many of the core practices of this course. Students will need creativity, problem solving skills, and persistence to complete their websites. This final section of the project requires those practices as well as collaboration and communication. Peer review will encourage students to leverage their peers as resources. The final reflection and presentation allow students to practice communication about their work.

Assessment Opportunities

Use the project rubric attached to this lesson to assess student mastery of learning goals of this unit. You may also choose to assign the post-project test through code studio.

Agenda

Day 1

Define: Rubric Review
Plan: Finishing Personal Website
Try: Finalize Personal Website

Day 2

Reflect: Peer Review
Plan and Try: Final Touches
Reflect: Final Reflection

Day 3

Showcase Set Up
Student Website Showcase

Post-Project Test

View on Code Studio

Objectives

Students will be able to:

- Apply CSS styles across an entire website
- Explain the design choices they made on their website to other people
- Prioritize and implement incremental improvements

Preparation

- [ ] Figure out a way to showcase all the students projects and get as many users as possible to check them out
- [ ] Plan out the day breakdown based on your class schedule
- [ ] Print a copy of the peer review guide for each student.
- [ ] Print a copy of the project guide for each student.
- [ ] Print a copy of the rubric for each student.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Final Personal Website** - Exemplars

For the Students

- **Final Personal Website** - Project Guide
  Make a Copy
- **Final Personal Website** - Rubric
  Make a Copy
- **Final Personal Website** - Peer Review
  Make a Copy
- **Computer Science Practices** - Reflection
  Make a Copy

Teaching Guide

Day 1

Define: Rubric Review

Remarks

> Over the course of the whole unit you have been developing a personal website. Over the next couple of classes you will be creating a final version of that project. As we prepare to showcase your work, we will use the problem solving process as our guide to ensure we have the best quality product possible.
>
> The first step of the problem solving process is to define the problem. To help define our problem we need to understand the expectations of the final product. The rubric for your Personal Website will help us. Let's go over it before starting.

Distribute: Give each student a copy of he project guide.

Discuss: As a class, review the rubric portion of the Project Guide and clarify any questions students have.

Plan: Finishing Personal Website

Remarks

> Next you will begin the planning stage of the problem solving process. Keep in mind you have limited time to complete this project so you will need to prioritize the work you need to.

Students should use the project guide to plan out the work they will do to finish their Personal Website.

Try: Finalize Personal Website

Remarks

> Now that you all have plans, it is time to begin implementing your plan. We will be doing a peer review as a class. You should work to make a polished product to share during the peer review.

Transition: Students log onto Code Studio and begin finalizing their websites.

Code Studio levels

- Lesson Overview
- **Teacher Overview**
- **Student Overview**

View on Code Studio to access answer key(s)

View on Code Studio

- Final Personal Website
- **Student Overview**

View on Code Studio

Personal Website

View on Code Studio

This is it! You have been working on your Personal Website for the whole unit. Put the finishing touches on it and submit it to your teacher! Remember to use the problem solving process as your guide in preparing your final product.

Do This

- Define: Read the rubric so you know what is expected
- Plan: Decide what you still need to work on and put those things in priority order
- Try: Implement your improvements to your personal website.
- Reflect: Get feedback from a classmate and decide what of that feedback to act on.

Don't forget to take a screenshot of your home page and save it somewhere on your computer so you can find it later (**Need help with how to screenshot? Check out this website .**)

Day 2

Remarks

> You should now have a polished product you are proud of. An important part of any major project is to get feedback from people not working on that project with you. They may bring some perspective you might have missed. We will spend today giving and getting feedback. Then you will reflect on how to put this feedback into action.

Reflect: Peer Review

Distribute: One copy of the peer review guide to each student.

Group: Pair students up.

Peer Review Process

> Students will:
>
> - Open up their website projects in Web Lab.
> - Fill in the top part of the worksheet, identifying what they would like feedback on.
> - Trade places with their partner so their partner is now looking at their sheet and website.
> - Give feedback on partner's work.
> - Switch back to their sheet and website to review feedback.
> - Make a plan for implementing some of the feedback.

Plan and Try: Final Touches

Transition: Students return to Code Studio and make any improvements that were identified in the peer review session. If they did not get any suggestions from the peer review, the teacher may want to give them some suggestions.

Students should also review the rubric as a final way to check their work.

Reflect: Final Reflection

Setup: Students will need their journals and all of their screenshots of the progress on their home page. You may want printed versions if you are going to display them during the showcase.

Prompts:

- I am most proud of the following aspect(s) of my personal website:
- The biggest challenge that I overcame so far in making my personal website:
- Improvements I still want to make to my website are:
- If I knew at the beginning of this project what I know now, I would have started it differently by doing this:
- I would describe the level of effort I've invested into this project as:

Send students to Code Studio to complete their reflection on their attitudes toward computer science. Although their answers are anonymous, the aggregated data will be available to you once at least five students have completed the survey.

Code Studio levels

- Levels
- 3

Student Instructions

View on Code Studio

Day 3

Showcase Set Up

Setup: Students need:

- A computer to display the website.
- A way to display their website progression screenshots

Student Website Showcase

Students should stand next to their computers and talk to people attending the showcase about their work. If you can't get others to come visit your room for this activity, you can split the class in half and have one half present while the others circulate. Then they can switch.

Post-Project Test

The post-project test is found at the bottom of the Web Development unit overview page on Code Studio (**studio.code.org/s/csd2-2019**).

This test is locked and hidden from student view by default. In order for students to see and take this test, you'll need to unlock it by clicking the "Lock Settings" button and following the instructions that appear.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▸ **AP** - Algorithms & Programming