

Unit 2 - Web Development ('24-'25)

[**▼ Teacher resources**](#)[**▼ Printing Options**](#)

In Web Development, students are put in the shoes of a web developer and empowered to design and create a web page to meet a user's needs. As students develop the pages and begin to see themselves as web developers and designers, they are encouraged to think critically about the impact of sharing information online and how to be more critical consumers of content. They are also introduced to problem-solving related to programming while learning valuable skills such as debugging, using resources, and teamwork. At the conclusion of chapter one, students will design and create a web page for a user. After chapter two, students will have worked with a team to create a multi-page website they can publish and share.

- [Chapter 1 Overview](#)
- [Chapter 2 Overview](#)
- [Implementation Guidance for Web Development](#)

Finished Teaching This Unit?

Answer this [short survey](#) to let the Code.org curriculum team know how the unit went.

| | | | | | |
|---------------|---------------------------------------|--|--|--|---|
| Week 1 | Lesson 1: Exploring Web Pages | Lesson 2: Intro to HTML | Lesson 3: Headings and Lists | Lesson 4: Digital Footprint | Lesson 5: Mini-Project: HTML Web Page |
| Week 2 | Lesson 6: Styling Text with CSS | Lesson 7: Intellectual Property | Lesson 8: Using Images | Lesson 9: Styling Elements with CSS | Lesson 10: Mini-Project: Adding Style to a Web Page |
| Week 3 | Lesson 11: CSS Classes | Lesson 12: Organizing Content with Flexbox | Lesson 13: Flexbox Children for More Control | Lesson 14: Chapter 1 Project | |
| Week 4 | Lesson 15: Team Problem Solving | Lesson 16: Sources and Research | Lesson 17: Linking Pages | Lesson 18: CSS Pseudo-classes | Lesson 19: Planning a Multi-Page Site |
| Week 5 | Lesson 19: Planning a Multi-Page Site | Lesson 20: Project - Website for a Purpose | | Lesson 21: Peer Review and Final Touches | |

| | | | |
|------------|---|--|--|
| Key | <input type="checkbox"/> Instructional Lesson | <input checked="" type="checkbox"/> Assessment | <input checked="" type="checkbox"/> Unplugged Lesson |
|------------|---|--|--|

[**View calendar**](#)**Active section:****Select a section**

▼ Chapter 1: Creating Webpages

▼ Lesson 1: Exploring Web Pages

Question of the Day: How can we choose websites based on the needs of a user?

To kick off a unit devoted to group problem solving and developing websites for other users, students begin by investigating the design of different websites. Students look at a variety of websites and attempt to match each design with a potential user. Then students choose a user and attempt to prototype a website design for them on paper or in a digital template. To conclude the activity, students consider what it means to be a web designer and create resources for other users.

 1

Sample Web Pages

 2

Check-In Survey

▼ Lesson 2: Intro to HTML

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. In this lesson, students are introduced 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. 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.

Question of the Day: How can we tell the computer both *what* to put on the web page, and *how* to organize it?

 1-2

Exploration

1

2

 3-5

Skill Building

3

4

5

 6

Practice

 7-8

Assessment

7 8

9

Challenge

▼ Lesson 3: Headings and Lists

This lesson introduces the core practices of pair programming and debugging. In this lesson, students continue to use HTML to structure text on web pages, this time in pairs, with a focus on working together and debugging problems with their sites. Students learn how to use the different heading and list elements and practice using them to give their web pages more structure.

Question of the Day: How can we work together to fix problems with our websites?

1

Exploration

2-3

Skill Building

2 3

4

Quick Check

5-7

Skill Building

5 6 7

8

Practice

9-10

Assessment

9 10

11

Challenges

▼ Lesson 4: Digital Footprint

Question of the Day: How can you make sure that your private information stays private?

In this lesson, students pause their own on developing web pages so they can develop an understanding of how personal information is surfaced on the internet, sometimes intentionally and sometimes unintentionally. Students look at several fictitious social media pages to see what they can learn about different people purely from publicly available information. They then reflect on what guidelines are appropriate for posting their own information online, especially as they prepare to create more personalized websites.

▼ Lesson 5: Mini-Project: HTML Web Page

In this mini-project, students use what they have learned to create a web page for a user. Students begin their project by using the Problem Solving Process to help them design a web page for a user. As they determine what content to include on the web page for their user, they will identify which tags they will use to implement their design. Students then build their user's web page in Web Lab. Optionally, after engaging in a formal feedback process, they may change their web pages based on feedback before reflecting on their process. The lesson also includes an optional review of the HTML students have learned before building the web page.

Question of the Day: How can I use HTML to design and build a web page for a user?



Review: HTML



Code Your Page



Review Your Page

▼ Lesson 6: Styling Text with CSS

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 an HTML page about Guinness World Record holders, adding their own style to the provided page. While only a few CSS properties are introduced in the core lesson, students are encouraged to use the optional activities at the end of the lesson to explore more ways that they can express themselves using CSS.

Question of the Day: How can we change the style of text on a web page?



Sample Website



Exploration



Skill Building

3

4

5

6

7

8



Practice



Assessment

10 11

12

Challenges

▼ Lesson 7: Intellectual Property

Until this point, the only content that students have used on their web pages is their own, but in the next lesson, they will be adding images to their sites. Before they do so, they need to understand the rules governing how to legally use content they find on the web. 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.

Question of the Day: What kind of rules protect everyone's rights when we use each other's content?

This lesson contains no levels.

▼ Lesson 8: Using Images

This lesson introduces images, which are different from earlier tags in that they are 'self-closing' and include attributes. Students should understand that these tags do not surround content in the same way as other tags. Students start the class by considering the ethical implications of using images on their websites, specifically in terms of intellectual property. They then learn how to add images to their web pages using the `` tag and how to cite the image sources appropriately.

Question of the Day: How can we add images on our websites, while making sure we respect everyone's rights?

1

Exploration

2-6

Skill Building

2

3

4

5

6

7

Practice

8-9

Assessment

8

9

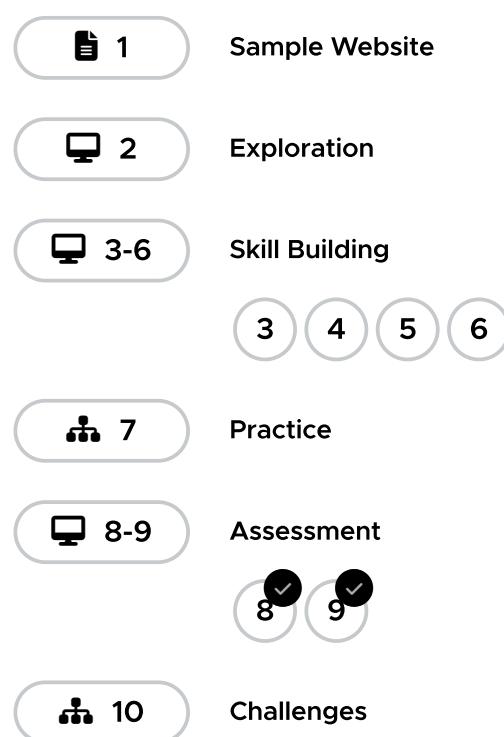
10

Challenges

▼ Lesson 9: Styling Elements with CSS

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 their personal websites.

Question of the Day: How can we style the images and layouts of our pages?



▼ Lesson 10: Mini-Project: Adding Style to a Web Page

In this mini-project, students use what they have learned to style a web page for a user. Students begin their project using the Problem Solving Process to help them determine how they will style a web page for a user. Students then style their users' web pages in Web Lab. Optionally, after engaging in a formal feedback process, they may change their web pages based on feedback before reflecting on their process. The lesson also includes an optional review of the HTML and CSS students learned before styling the web page.

Question of the Day: How can I use CSS to style a web page for a user?



▼ Lesson 11: CSS Classes

Up until this point, the only styling students have been able to do is styling by the element, which means that every element of a particular kind has the same style. This lesson expands on the CSS that students have already learned by introducing classes, which allow web developers to treat groups of elements they want to be styled differently than other elements of the same type. Students first investigate and modify classes on various pages, then create their own classes and use them to better control the appearance of their pages. They then reflect on how they could use this skill to improve their team websites.

Question of the Day: How can we create different styles for the same type of element?

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.

1

Sample Web Page

2

Exploration

3-4

Skill Building

3

4

5

Practice

6-7

Assessment

6

7

8

Challenges

▼ Lesson 12: Organizing Content with Flexbox

This lesson introduces students to two new web development tools: div tags and Flexbox styling. This lesson introduces students to div tags to organize their HTML code into sections. They then learn about Flexbox and how it can be used with their div containers to style and position items. Students practice using Flexbox properties to control the layout of web pages.

Question of the Day: What are div tags and Flexbox and why are they important in web design?

1

Div & Flex Exploration

2-5

Skill Building

2 3 4 5

6

Practice Levels

7-8

Assessment Level

7 8

9

Challenge Levels

▼ Lesson 13: Flexbox Children for More Control

This lesson introduces students to advanced Flexbox child properties including `order`, `align-self`, `flex-grow`, and `flex-shrink` through an interactive warm-up and the various level exercises. They'll deepen their understanding of how these properties affect web page layouts and conclude with a creative group drawing game to reinforce their learning.

Question of the Day: How do Flexbox child properties affect the layout and organization of elements within a webpage?

1

Exploration level

2-5

Skill Building Levels

2 3 4 5

6

Practice Levels

7-8

Assessment Level

7 8

9

Challenge Levels

▼ Lesson 14: Chapter 1 Project

In this chapter project, students use what they have learned to style and organize a web page for a user. Students begin their project by using the Problem Solving Process to help them determine how they will organize the content of a user's web page. Students then use CSS classes and flexbox to organize their user's web page in Web Lab. Optionally, after engaging in a formal feedback process, they may change their web pages based on feedback before reflecting on their process.

Question of the Day: How can I use CSS classes and Flexbox to organize webpage content for a user?

▼ Chapter 2: Multi-page Websites

▼ Lesson 15: Team Problem Solving

This lesson explicitly addresses the challenges students may find working in a group and supports them in crafting a plan to overcome these challenges. Students work together to set group norms and brainstorm what features they would like their websites to have. The class starts by thinking of some popular teams in different contexts, then reflects on what makes teams successful. They then get into their own teams and make a plan for how they will interact and reach success in their own projects. Afterward, the teams begin to brainstorm ideas for their website project.

Question of the Day: How can we work together to make a great team?

This lesson contains no levels.

▼ Lesson 16: Sources and Research

This lesson encourages students to think more about their responsibilities as consumers of information and how to find relevant and trustworthy information online. After viewing and discussing a video about how search engines work, students will search for information relevant to their site. They'll need to analyze the sites they find for credibility to decide which are appropriate to use on their own website. 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 as content producers by looking for ways to make their own sites appear more trustworthy.

Question of the Day: How do we find relevant and trustworthy information on the Internet?

This lesson contains no levels.

▼ Lesson 17: Linking Pages

This lesson will give students practice in using links and introduce them to good navigation practices for their sites. Students begin by looking online for the first web page and discussing how its use of links was what started the web. They then transition to Web Lab where they

learn how to make their own links, as well as good conventions that make it easier for users to navigate on a page. Finally, they reflect on their group project and what their personal goals are for the final stretch.

Question of the Day: How can we combine several different web pages into one website?



Exploration



Skill Building

2

3

4



Practice



Adding Pages

6

7



Challenges

▼ Lesson 18: CSS Pseudo-classes

This lesson introduces students to the dynamic world of CSS pseudo-classes. Students will learn about their role in enhancing web page interactivity and style. They will explore, practice, and apply various pseudo-classes like `link`, `visited`, `hover`, and `active`, understanding how these selectors can transform the user experience on websites.

Question of the Day: What is a CSS pseudo-class, and how does it change the way an element looks or behaves on a web page?



Exploration Level



Skill Building Levels

2

3

4

5



Practice Levels



Assessment Level

7

8



Challenge Levels

▼ Lesson 19: Planning a Multi-Page Site

Students work in teams to plan out their websites and create a sketch of each page. They then download the media that they will need for their sites. At the end of the activity, they decide how the work will be distributed among team members and report whether the entire group agreed to the plan.

Question of the Day: How do we plan a web page as a group?



Upload Your Images

▼ Lesson 20: Project - Website for a Purpose

Teams have spent a lot of time throughout the chapter planning their websites. In this lesson, they are finally able to code their pages. Using the project guide, the team works together and individually to code all of the pages, then puts all of the work together into a single site.

Question of the Day: What skills and practices will help us work together to make a great website?



Your Team Project



Add Content and HTML



Share Your Pages



Add Style



Share Your Stylesheets



Check Your Work

▼ Lesson 21: Peer Review and Final Touches

This lesson focuses on the value of peer feedback. Students first reflect on what they are proud of, and what they would like feedback on. Teams then work with peers to get that feedback through a structured process that includes the project rubric criteria. Afterward, students decide how they would like to respond to the feedback and put the finishing touches on their sites. After a final review of the rubric, they reflect on their process. To cap off the unit, they will share their projects and also an overview of the process they took to get to that final design.

Question of the Day: How can we use feedback to make our websites better?*



1

Incorporate Feedback



2

End-Of-Unit Survey

▼ Post-Project Test

This lesson is locked - you need to become a verified teacher to unlock it. [Learn more.](#)



Lesson 1: Exploring Web Pages

45 minutes

Overview

Question of the Day: How can we choose websites based on the needs of a user?

To kick off a unit devoted to group problem solving and developing websites for other users, students begin by investigating the design of different websites. Students look at a variety of websites and attempt to match each design with a potential user. Then students choose a user and attempt to prototype a website design for them on paper or in a digital template. To conclude the activity, students consider what it means to be a web designer and create resources for other users.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- **IC** - Impacts of Computing

Agenda

Before the Lesson

Preparing for the Unit

Warm Up - Option 1

Problem Solving Word Search

The Problem Solving Process

Warm Up - Option 2 (5 minutes)

Revisit The Problem Solving Process

Activity (35 minutes)

Recommending Websites

Designing a Website

Wrap Up (5 minutes)

Reflection

Objectives

Students will be able to:

- Create a prototype of a web design to meet the needs of a user using the problem-solving process
- Identify features of a web design that match the needs of users
- Understand the steps of the problem-solving process

Preparation

- Print a copy of the activity guide for each student.
- Check the "[Teacher's Lounge](#)" forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our [Virtual Lesson Modifications](#)

Links

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

For the teachers

- [Exploring Web Pages](#) - Slides
▼ Make a Copy

For the students

- [\(Warm Up\) Word Search](#) - Activity Guide
▼ Make a Copy
- [Problem Solving and Design](#) - Resource
▼ Make a Copy
- [The Problem Solving Process](#) - Video ([Download](#))

- [Web Design Template](#) - Resource
▼ Make a Copy
- [Web Design for Users \(A\)](#) - Activity Guide
▼ Make a Copy
- [Web Design for Users \(B\)](#) - Activity Guide
▼ Make a Copy

Teaching Guide

Before the Lesson

Preparing for the Unit

Getting Started with Code.org: Consider watching our [Getting Started with Code.org](#) video series for an overview of how to navigate lesson plans, setup a classroom section, and other important features of the Code.org platform. Each video also has a support article if you'd prefer to read or print instructions - [click here to learn more](#).

Setup a Classroom Section: You can use a class section in Code.org to manage your students, view their progress, and assign specific curriculum - [click here to learn more](#).

If you are using a learning management system, there may be additional steps to sync your classes with Code.org:

- [Click here](#) for steps to setup your classes with Google Classroom
- [Click Here](#) for steps to setup your classes with Clever

Become a Verified Teacher: Lesson plans and levels have additional resources and answer keys for Verified Teachers, which is quick process that verifies your position at an educational institution. [Click here to complete a form](#) and you should have access to verified teacher resources in ~1 business day. Verified teachers also have access to the ["Teacher's Lounge"](#) section of the forums.

Get Inspired: Consider watching our [Teacher Tips video playlist](#), featuring current Code.org teachers.

Technical Requirements: For the very best experience with all Code.org content, we recommend consulting with your school or district's IT department to *ensure specific sites are allowed and are not blocked*. [Click here to see a list of sites to unblock](#).

You can also find mobile and tablet support details, hardware recommendation information such as minimum Internet connection speed, smallest screen size supported, and other hardware recommendations, as well as a list of supported browsers and platforms at the same [technical requirements website](#).

Warm Up - Option 1

💡 Teaching Tip

Two Warm-Up Options: This lesson has two warm-up options, depending on whether or not students have been previously introduced to the Problem Solving Process.

Option 1 is for classrooms that have **not been introduced** to the Problem Solving Process. The warm up is longer than normal because it introduces the Problem Solving Process through a brief activity and a video.

Option 2 is for classrooms that **have been introduced** to the Problem Solving Process in a previous unit. The warm-up is shorter and acts as a quick review of the Problem Solving Process.

Problem Solving Word Search (10 minutes)

Microphone Remarks

In this unit, we'll be learning how to design websites to solve problems. The word "problem" can refer to lots of different situations - I could say I have a problem for homework, a problem with my brother, and a problem with my car, and all three mean very different things.

Prompt: What's an example of a problem you've had to solve recently?

Circulate & Distribute: Have students share their thoughts with a neighbor. Listen for examples that may be worth using as examples throughout the lesson.

As students share, distribute a copy of the **(Warm Up) Word Search** to each student, face down. Make sure students don't start the word search ahead of time.

Microphone Remarks

When we build websites, we'll also face problems to solve. So today, we are going to learn about and practice an effective strategy for when we need to solve problems. We'll start with a simple problem: solving a word search.

Prompt: *What is the problem you are trying to solve when you do a word search?*

Discuss: Allow students to discuss their ideas with a classmate before sharing them with the entire class.

Discussion Goal: Guide students to the idea that the problem of a word search is that there are hidden words throughout the grid of letters and thus the goal of a word search is to find all the words in the list.

Prompt: *What are the different strategies that someone could use for finding words. What are the pros and cons of each strategy?*

Discuss: Allow students to silently record their ideas in writing for a minute. Afterward, invite them to share what they wrote with a neighbor and then finally bring the whole class together to develop a classwide list of strategies.

Discussion Goal: This is a quick, low-stakes brainstorm that asks students to bring in their own experiences with word searches and identify strategies they may have used or could use. Student responses may vary but may include the following strategies:

- Scan each row from left to right followed by scanning each column from top to bottom.
- Check surrounding letters. Once you found the first letter or a key letter in a word you're searching for, check the letters around it to see if you should keep going. For example, if you have a word with Q in it, if there's no U next to it, then you need to keep searching.
- Look for words that have less-common letters in them such as words with a Q, X, J, or Z
- Look for words with letter pairs, such as double letters or QU
- Look for more than one word at a time.
- Turn the puzzle upside down

Do This: Direct students to pick ONE strategy from the list that they want to try and put a "#1" next to it in their journals to indicate it was their first strategy used. Once they pick a strategy, direct students to use that strategy in their word search.

It is important to let the students know that if at any time the strategy they picked is not working, they should feel free to go back to the list of strategies and pick a different one, making sure to put a number next to the new strategy.

Circulate: Give students a few minutes to find as many words as possible. It is okay if they don't find every word. You may bring everyone back together once everyone has found at least several of the words on the list.

Prompt: Display and have students reflect on each of the following prompts:

1. Were you successful in solving this word search problem? How do you know?
2. What strategy did you pick and how did it work for you?
3. Did anyone change strategies at any point ... why or why not?

💡 Teaching Tip

As you lead this discussion, you may choose to do a quick raise of hand poll to find out who was successful, or on track to be successful if they didn't get a chance to finish. Additionally, you can also instruct students to pick two of the questions to answer.

Discussion Goal: This discussion serves as a way to get students to reflect on how they did solving this problem and to give students a moment to recognize if they changed strategies and why or if they would solve a problem like this differently next time.

The Problem Solving Process (5 minutes)

Display: Show [The Problem Solving Process](#) video.

Questions to Consider with Video:

- What are the 4 steps to the Problem-Solving Process?

💡 Teaching Tip

Videos are used throughout the curriculum to spark discussions, supplement key concepts with additional explanations and examples, and expose students to the various roles and backgrounds of individuals in computer science.

While interacting with the video, turn on closed captioning so students can also read along as they watch.

To encourage active engagement and reflection, use one or more of the strategies discussed in the [Guide to Curriculum Videos](#).

Discuss and Display: Briefly discuss with students what parts of the warm-up activity they felt fell into each step of the problem-solving process before displaying the example answers in the slides:

- **Define:** when we put into our own words what the problem of a word search is and what the goal is
- **Prepare:** the brainstorming of different strategies that could be used to solve a word search
- **Try:** when we picked a strategy and tried to find the words
- **Reflect:** when we reflected on how they did at the end

💡 Teaching Tip

This is a good point to remind students that the problem solving process is a cycle and there should be times when they need to repeat steps. For example, if they went back to the list of strategies and picked a different one at any point when solving this problem, they were not only doing the “Reflect” step without realizing it, but going back to the “Prepare” and “Try” steps again.

Microphone Remarks

Now that we're familiar with the problem solving process, let's explore how we can use it when designing websites.

Warm Up - Option 2 (5 minutes)

Question Teaching Tip

Two Warm-Up Options: This lesson has two warm-up options, depending on whether or not students have been previously introduced to the Problem Solving Process.

Option 2 is for classrooms that **have been introduced** to the Problem Solving Process in a previous unit. The warm-up is shorter and acts as a quick review of the Problem Solving Process.

Revisit The Problem Solving Process (5 minutes)

Microphone Remarks

In this unit, we'll be learning how to design websites to solve problems. Today, we are going to remind ourselves about the Problem Solving Process and how it can be applied when we get design websites.

List Prompt: What do you remember about the Problem-Solving Process?

Discussion Goal: Based on student responses, the teacher should lead students through a discussion about the four steps of the Problem-Solving Process and what they remember to be involved in each step.

List Prompt: *Based on what you know about the Problem-Solving Process, what do you think each step could look like when creating a website?*

Discussion Goal: Even though students haven't made websites yet, guide students to imagine the steps involved from having the first inspiration for a website to having it created. Some ideas may include:

- **Define:** Deciding what you want to make a website about or who you want to make it for
- **Prepare:** Looking at similar websites and deciding what kinds of colors or images you want to use, or sketching out the website
- **Try:** Creating a basic version of the website that may not have all of the features
- **Reflect:** Showing it to friends to get feedback

As students finish discussing, transition to the main activity.

Activity (35 minutes)

Recommending Websites

Microphone Remarks

This unit we are going to learn how to create websites, which starts with thinking about who we're creating a website for. Sometimes we make websites about our own personal interests, but we can also think about designing websites for other people. We'll learn how to apply the problem solving process when designing a website to meet the needs of our users and ourselves.

Group: Put students in pairs.

Distribute: Pass out a copy of the Web Design for Users activity guide to each student. There is an A and a B version with different users for the second activity, but it doesn't matter which one students get.

 **Teaching Tip** ▲

Answer Keys & Exemplars: An answer key or exemplar is provided for verified teachers as part of the resources in this lesson plan. If you do not see an answer key or exemplar listed as a resource, [follow these steps](#) to become a verified teacher.

Overview: Have students read through the directions on the first page of the activity guide. The first part of this activity asks students to match different users to a website template that suits their needs. Students will need to visit Code Studio to view the website templates.

Code Studio: Direct students to the "Sample Web Pages" level on Code Studio.

 **1**

Sample Web Pages

Circulate: Have students work through this activity in pairs, encouraging discussion about why exactly they are choosing to make each template. This is a great place to discuss different website designs and the needs of each user.

This activity guide asks students to consider why people make personal web pages and what sorts of interests can be expressed on them.

 **Teaching Tip** ▲

Value Reasoning over Correctness: Each user is designed with a particular template in mind, but students are encouraged to use their own reasoning and logic when recommending a template to users. It's okay for students to not make an "obvious" choice as long as they're thinking carefully about their reasoning and are able to justify their choices.

The sites are intentionally simple, to help set expectations about the websites that students will be creating over the course of the unit.

Share Out: Have students discuss the following questions in small groups before bringing to a full class discussion:

- Which users were the easiest to find matches for?
- Which users were the hardest to find matches for?

Based on your observations while circulating, you should also identify a few specific users to ask the class to discuss, especially users where multiple students chose different websites. Encourage students to hear different justifications for their choices, emphasizing that the rationale is more important than the choice itself.

 **Assessment Opportunity** ▲

As students discuss their reasoning for their choices, check to ensure that they are identifying the particular user's needs and characteristics, rather than general reasons to prefer a certain website. You may want to challenge students to distinguish their own needs and preferences from those of the described users.

Designing a Website

Remarks

Now we've been asked to design a website ourself for a client. We can create our own design, but it should meet the needs of our user.

Teaching Tip

Note on Timing: Depending on how long the warm-up took, especially if the class watched the problem-solving video, you may run out of time to fully complete this activity. That's okay! It can be something students come back to as they progress throughout the unit, or you may decide to extend this activity into a new class period and stretch-out the design and feedback process. This task can be implemented in a variety of ways to match the constraints of the classroom.

Overview: Have students flip to the back of their activity guide and read the overview of the next part of the activity. Students will choose a user to create a basic website design for, following the Problem Solving Process. Students can sketch their design on pen and paper, or use the provided Google Slides template to create their design. These designs are meant to be sketches - students will not have enough time to "perfect" their design.

Distribute: Pass out copies of the **Problem Solving and Design** resource. Students can use this when preparing their design, and it can be a reference throughout the unit.

Circulate: Monitor students as they complete this activity, ensuring that students are creating designs that match the needs of the user they chose. Monitor student conversations as well to make sure both voices are being heard during discussions and decisions. Students will be working in teams later in the unit to create websites, so it can be good to intervene early to reinforce good collaboration habits.

(Optional) Share Out: Depending on how much time is left in class, allow students to trade with other groups and share their designs. Encourage students to share with another group who picked the same user so they can see similarities and differences in their designs.

Assessment Opportunity

Following the Problem Solving Process: You can use student responses on the activity guide to determine how well they followed each step of the problem-solving process. This process will be reinforced throughout the unit, especially on projects.

Wrap Up (5 minutes)

Prompt: Reflect - What was one thing that you think your design does well and really meets the need of the user? What's an area that you think can still be improved to meet the needs of your user?

Circulate: Students can answer these prompts on their activity guide in the Reflect section of the Problem Solving Process.

Share Out: Have students share their responses, highlighting similarities between responses even when the users were different.

Question of the Day: How can we choose websites based on the needs of a user?

Reflection

Code Studio: Have students answer 5 quick survey questions at the beginning of this unit. Once at least 5 students have completed the survey you will be able to view the anonymized results in the Teacher Dashboard. Some of these questions will be asked again at the end of the first project, which can be helpful in seeing student growth and shifts in attitudes throughout the unit.



This work is available under a [**Creative Commons License \(CC BY-NC-SA 4.0\)**](#).

If you are interested in licensing Code.org materials for commercial purposes [**contact us**](#).



Lesson 2: Intro to HTML

45 minutes

Overview

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. In this lesson, students are introduced 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. 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.

Question of the Day: How can we tell the computer both *what* to put on the web page, and *how* to organize it?

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- AP - Algorithms & Programming

Agenda

Warm Up (5 minutes)

The Need for HTML

Activity (35 minutes)

Pair Programming

Exploring HTML

Wrap Up (5 minutes)

Reflection

Objectives

Students will be able to:

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

Preparation

- Review the Code Studio levels
- Check the "Teacher's Lounge" forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our Virtual Lesson Modifications

Links

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

For the teachers

- Exemplar Text Website - Website
- HTML Tags - Resource
- Intro to HTML - Slides
 - ▼ Make a Copy

For the students

- Intro to Web Lab Part 2 - Video (Download)
- Pair Programming - Video

- [Video: Intro to Web Lab Part 1](#) -
Video ([Download](#))

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 text and images on a website
- **Website Structure** - how the content of a website is organized

Introduced Code

- `<!DOCTYPE>`
- `<body></body>`
- `<head></head>`
- `<html></html>`
- `<p></p>`

Teaching Guide

Warm Up (5 minutes)

The Need for HTML

Display: Show the image inside the [Exemplar Text Website](#)

 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.

 **Prompt:** How could you explain to someone over the phone how to draw the following web page?

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

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.

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 student's 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 of 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.

Microphone 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.

Question of the Day: How can we tell the computer both *what* to put on the web page, and *how* to organize it?

Key Vocabulary:

- **website content** - the text and images on a website
- **website structure** - how the content of a website is organized

Activity (35 minutes)

Pair Programming

Group: Put students into pairs.

Microphone Remarks

Today we're going to start using a tool called Web Lab. We're going to explore this tool using something called **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.

Video: Show students the [Pair Programming](#) video in the slides.

Questions to consider with the video:

- Why do you think professional programmers use pair programming?

- How do you think pair programming will help you to program better?

Discussion Goal: 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 fun to do.

💡 Teaching Tip

Videos are used throughout the curriculum to spark discussions, supplement key concepts with additional explanations and examples, and expose students to the various roles and backgrounds of individuals in computer science.

While interacting with the video, turn on closed captioning so students can also read along as they watch.

To encourage active engagement and reflection, use one or more of the strategies discussed in the [**Guide to Curriculum Videos**](#).

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 few minutes.

💡 Teaching Tip

Practicing Communication: Some classes may need more support in communicating and collaborating effectively. If appropriate, consider having your students brainstorm a list of "sentence stems" that they can use for respectful and effective communication before they break into pairs ("Have you considered..." "What about..." "I think the problem might be..."). As students move through the lesson, be attentive to how students are working together - understanding the norms of pair programming is just as important as learning the new HTML code in this lesson!

To read more about pair programming, see the [**Guide to Teaching and Learning Strategies**](#).

Exploring HTML

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

 **Display:** Use the slide to guide your students on how to connect Pair Programming in Code Studio.

- Choose one partner to log into Code Studio.
- Click on your name, and choose "Pair Programming from the menu."
- Choose your partner's name from the list.
- Start coding as a team!

Do This: Remind students to switch driver and navigator every three minutes. You may want to project a digital timer at the front of the room.



1

Exploration

Facilitating Exploration Levels: Exploration levels are a great opportunity for students to think critically about code and engage in class discussions. Consider having students discuss with a partner before and after typing in code. Once they have had a chance to explore the code and discuss with a partner, bring the class together for a full-group discussion to discuss what they did in the Exploration level, what they **noticed**, and what they still **wonder**. Use this as an opportunity to address any misconceptions that students may have had about the code initially.

Digging Deeper: For more tips about programming levels, see the [**CSD Guide to Programming Levels**](#). This document includes strategies and best-practices for facilitating programming levels with students.

Circulate: Give students time to explore the tool and complete the task on the level. Students may also discover different tabs and buttons on the page, such as the Inspector tool or the Documentation button.

Prompt: *What did you notice about the workspace and the preview? What other features did you discover in this tool?*

Discussion Goal: Students should notice that even if they type sentences on multiple lines, the preview will show all of the sentences on a single line. Also highlight any additional features students discover in this tool and make sure all students are aware of it

▀ **Video:** Show students the [**Video: Intro to Web Lab Part 1**](#) video in the slides.

Question to consider with the video:

- What are the different parts of Web Lab, and what are they used for?

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.

Using Resources: Below you can find recommendations for using the many resources students are introduced to in the lesson. You could consider creating a "Resource Chart" to keep track of these options and support students to be self-sufficient as they progress through 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.
- **Help and Tips Tab** This tab contains all of the relevant videos and reference guides for a particular level.
 - **Videos:** Watched as a class, but students can always return to them.
 - **Reference Guides:** 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. You may decide to print these and have them available for students as they work through levels.
- **Documentation** A link to Web Lab's documentation is available in the upper-right corner of the instructions panel. Students can use this to refresh their memory on important tags introduced in the unit.

 **Code Studio:** Have students continue to the next level in Code Studio

2

Exploration

Teaching Tip



Using the Tool As students explore the site, make sure that they understand how to use the inspector tool and the "Help and Tips" tab, which will give them access to the previous video.

Text-to-Speech Options: The instructions panel includes two options that can support comprehension for students.

- **Text to Speech** which reads aloud the instructions for students
- **Microsoft Immersive Reader** which opens a new panel for the instructions and gives controls to change the text size, contrast, or translate to another language.

[**Click here to learn more about these options**](#)

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

- What code makes the text bigger and bolder?
- What code makes text appear as a list?
- What code makes the text appear on separate lines?
- What's a piece of code that doesn't appear to do anything on the screen?

It's okay to not address these questions as a full group or answer them completely - they will be introduced in the following video and covered in-depth throughout the unit. Instead, 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:** Show students the [**Intro to Web Lab Part 2**](#) video in the slides

Questions to consider with the video:

- Why are HTML tags useful?
- What does the paragraph tag do?

Discussion Goals: As students 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).

For the second prompt, there's a direct answer: 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?

 **Key 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

 **Code Studio:** Have students continue to the next level in Code Studio

3-5

Skill Building

3

4

5

Teaching Tip

Facilitating Skill Building Levels: Skill Building levels are designed to continue teaching new skills and blocks through exploration, trial-and-error, and using worked examples from pre-supplied code. Students are still getting familiar with the concepts in the lesson and will need strong support throughout these levels to build confidence, debug their code, and cement their understanding.

Consider having students complete Skill Building levels in pairs using **Pair Programming**, which has students use one computer and trade between being a Driver or a Navigator. This process is highlighted in [this video](#), which you can show to the class. You can have students switch roles based on a timer, or switch every time they complete a level.

Digging Deeper: For more tips about programming levels, see the [CSD Guide to Programming Levels](#). This document includes strategies and best-practices for facilitating programming levels with students.

6

Practice

Teaching Tip

Facilitating Practice Levels: Practice levels are designed for students to apply their knowledge from the previous levels and develop fluency in using the new blocks of code to solve problems. Students can choose which practice levels they would like to complete, and it's not necessary for a student to complete each practice level before continuing.

Students tend to be more engaged and respond better when they have an authentic choice about how to continue their learning. Allow students to choose practice levels according to their interests and level of comfort, and consider providing opportunities for students to demonstrate and explain their solutions to the practice levels they chose to the entire class.

Digging Deeper: For more tips about programming levels, see the [CSD Guide to Programming Levels](#). This document includes strategies and best-practices for facilitating programming levels with students.

7-8

Assessment

7

8

Assessment Opportunity

Formative Assessment: Levels 7 & 8 can be used as a formative assessment.

A rubric is provided in level 7, and written feedback can be given to students. [Click here to learn more about giving feedback to students](#).

Level 8 is a free response reflection that can be used as an opportunity to gain insight into how the students completed the assessment level and the choices they made with the HTML tags. In addition, this reflection can be used to identify any misconceptions shared among students that should be

cleared up.



Challenge

💡 Teaching Tip

Facilitating Challenge Levels: Challenge levels are designed as extensions to the concepts and skills students learn throughout a lesson. Challenge levels tend to focus on more open-ended tasks for students to complete, or opportunities to combine several skills from previous lessons together into one program.

Challenge levels do **not** need to be completed for students to meet the core objectives of a lesson. Instead, every task in a challenge level is meant to supplement and enrich the learning objectives of a lesson, but are not required for future lessons. Students can still demonstrate mastery of the objectives of a lesson without completing any of the challenge levels.

Digging Deeper: For more tips about programming levels, see the [**CSD Guide to Programming Levels**](#). This document includes strategies and best-practices for facilitating programming levels with students.

(Optional) Video: One of the challenge levels invites students to create their own poetry as a webpage, featuring an example poem from Caia Lomeli. Caia is a poet and computer science student who was featured in our Poem Art Hour of Code activity. The lesson [**links to a video**](#) of Caia discussing how computer science and poetry are similar, especially "starting from a blank page". Even though it's not directly related to webpages and HTML, [**you may decide to show this video to students as an inspirational video**](#) on being creative with code, which ties into the projects students will complete in this unit.

Wrap Up (5 minutes)

Reflection

Question of the Day: How can we tell the computer both *what* to put on the web page, and *how* to organize it?

Journal 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?

✓ 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.

As students discuss HTML as a solution, make sure that they are using the key vocabulary of the lesson:

- **website content** - the text and images on a website
- **website structure** - how the content of a website is organized
- **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

The content is the literal words that are being typed on the page. Using HTML, students are providing structure 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 later in the unit.

Discuss: After students have had time to reflect individually in the journal, allow them to discuss with a partner, then share with the class.

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.



This work is available under a [**Creative Commons License \(CC BY-NC-SA 4.0\)**](#).

If you are interested in licensing Code.org materials for commercial purposes [**contact us**](#).

Lesson 3: Headings and Lists

45 minutes

Overview

This lesson introduces the core practices of pair programming and debugging. In this lesson, students continue to use HTML to structure text on web pages, this time in pairs, with a focus on working together and debugging problems with their sites. Students learn how to use the different heading and list elements and practice using them to give their web pages more structure.

Question of the Day: How can we work together to fix problems with our websites?

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- AP - Algorithms & Programming

Agenda

Warm Up (5 minutes)

Tags Poster

Activity (35 minutes)

Web Lab: Headings

Web Lab: Lists

Wrap Up (5 minutes)

Journal

Objectives

Students will be able to:

- Structure content into headings, subheadings, lists and paragraphs.
- Use a structured practice to collaboratively create a digital artifact.
- Use heading and list tags to change the appearance of text on a web page.

Preparation

- Have student journals ready to give back.
- If you want to use an anchor chart, prepare poster paper to do so as a whole class.
- Check the "[Teacher's Lounge](#)" forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our [Virtual Lesson Modifications](#)

Links

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

For the teachers

- [Formatting HTML](#) - Resource
- [Headings and Lists](#) - Slides
 - ▼ Make a Copy
- [Headings and Paragraphs](#) - Resource
- [Lists](#) - Resource

For the students

- [Video: Debugging](#) - Video
[Download](#)
- [Video: Pair Programming](#) - Video
[Download](#)

Vocabulary

- **Heading** - A title or summary for a document or section of a document.
- **List** - allows web developers to group a set of related items

Introduced Code

- `<h1></h1>`
- ``
- ``
- ``

Teaching Guide

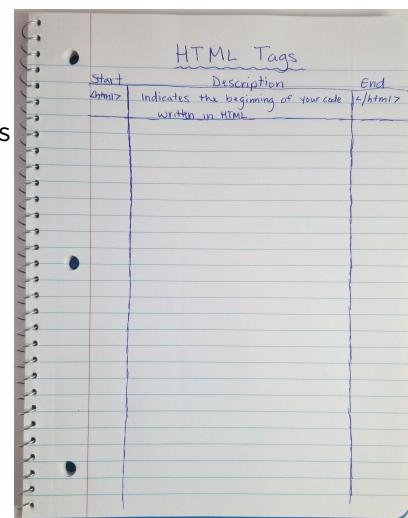
Warm Up (5 minutes)

Tags Poster

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

💡 Teaching Tip

Anchor Chart / Poster: Throughout this unit students will be keeping track of the tags they learn. This warm-up prompts students to record the HTML tags that they learn by writing them in their journals. If you wish, you can also keep track of the same information on a shared class poster or anchor chart that you update after each lesson. Prompts throughout the unit will tell you when students should update their journals, and updating the anchor chart may provide reinforcement of that process.



 **Prompt:** Yesterday, you learned about HTML, the language of the World Wide Web. HTML uses tags to structure content on web pages. Individually, 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.

Remarks

Now that you've had some time to think of your own, share your lists with a partner and see whether there's anything else that you can add.

Discuss: Pairs should discuss and share with one another the HTML tags they can remember and start record their ideas on the HTML Tag Chart.

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

- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<body>`
- `<p>`

Afterward, ask students to share the tags they came up with along with each tag's description.

 **Display:** Use the slide with the HTML Tag Chart table to help students ensure they documented all tags needed at this point along with their descriptions.

`<!DOCTYPE html>` - Tells the computer that this is a document written in HTML - No closing tag

`<html>` - Indicates the beginning of your code written in HTML - `</html>`

`<head>` - Contains information (sometimes called "metadata") about your web page - `</head>`

`<body>` - Contains all the main content of your web page - `</body>`

`<p>` - Defines a paragraph - `</p>`

As you go through the tags, highlight how working together allowed the students to make their lists more comprehensive.

Teaching Tip ▲

If you choose to make a poster or anchor chart, you can use this time to update the chart at this point.

Remarks

Usually we are able to solve problems better when we work with someone else. That's true in programming our web sites, too. Today, we're going to look at some ways that we can work together to solve different problems that our websites might have.

Question of the Day: How can we work together to fix problems with our websites?

Activity (35 minutes)

Teaching Tip ▲

Guide to Programming Levels: Additional guidance for programming levels is provided in the [CSD Guide to Programming Levels](#). This document includes strategies and best-practices for facilitating programming levels with students.

Web Lab: Headings

Group: Group students into pairs

Transition: Have pairs go to Code Studio and both log in using the "Pair Programming" feature, just like yesterday

Do This: Remind students to switch driver and navigator every three minutes. You may want to project a digital timer at the front of the room.

1

Exploration

Circulate: Give students time to explore the headings and discuss the questions on the level with their partner.

Prompt: When would you use each of these headings? For example, which heading would you use for a web page title ... when might you use the H6 heading?

Discussion Goal: Students should notice the H1 is the largest heading tag and should therefore be used for web page titles. The remaining heading tags are used for sub-titles and section/subsection titles.

Code Studio: Have students continue to the next level in Code Studio

2-3

Skill Building

2

3

4

Quick Check

Teaching Tip



[pull-right]



[/pull-right]

Level Types: This multiple choice question that asks students to choose how HTML headings will display on a web page. You can see which students have answered the question correctly by going to that level and pulling out the Teacher Panel from the right hand side. Students who have successfully answered the question will have a green bubble. Students have as many chances as they like to answer correctly, so guessing and checking can also lead to a correct answer. [Click Here to learn more about using the teacher panel with students.](#)

Assessment Opportunity ▲

This level can be used to assess student understanding of using heading tags to change the appearance of text on a web page.

Transition: Have your students pause programming and bring the class back together at this point in order to add heading tags to their HTML Tag Chart.

Prompt: What are the different heading tags? Which one would we use for web page titles? What are the others for?

Discussion Goal: Students should be able to identify six different heading tags, from `<h1>` to `<h6>`. In addition, students should understand that the `<h1>` tag is the largest heading tag, typically used for the main web page title while the others are used for subtitles throughout a web page.

Display: Use the slides and the remarks below to help students add heading tags to their HTML Tag Chart.

`<h1>` - Largest heading tag, typically used for web page titles - `</h1>`

`<h2>` to `<h6>` - Heading tags, typically used for subtitles - `</h2>` to `</h6>`

Remarks

Prompt: It is important to remember and get down in our HTML chart that heading tags, `<h1>` through `<h6>` are used any time we have a title or a subtitle. `<h1>` is normally used for the webpage title, while `<h2>` to `<h6>` are used for subtitles on webpages.

Web Lab: Lists

Prompt: When might someone use a list?

- Think about times when you have made lists ... Why did you make the list? What was the list for? What type of stuff was in your list(s)?

Keep track of the different types of lists students identify.

Discussion Goal: There are many possible answers. Examples could include a **bulleted** list used to make a grocery list and a **numbered** list used to list your favorite movies or the steps in a recipe.

Display: Show the first example of an HTML list and the result. Ask the students "What do you notice? What do you wonder?"

Discussion Goals: Student responses to the first prompt, *What do you notice*, may vary but could include:

- There are two new tags - `` and ``
- The result is a bulleted list
- The closing `` tag comes after all of the `` tags
- The `` tags are indented

Student responses to the second prompt, *What do you wonder*, may also vary but could include:

- What do "ul" and "li" mean?
- Why are there multiple `` elements?
- Why does the `` tag not seem to do anything?
- Why are the `` tags indented?

It is important to note that this prompt is purposefully open-ended which could result in unexpected student responses, which is okay.

Do This: Move on to the second example before answering any questions. Allow their *What do you wonder* questions to linger.

Display: Show the second example of an HTML list and the result. Again, ask the students *What do you notice? What do you wonder?*

Discussion Goals: Student responses for the first prompt, *What do you notice*, may again vary but could include:

- The `` tag is being used again
- `` is being used instead of ``
- This time a numbered list was made
- There is no content next to the starting `` tag, similar to the `` start tag in the first example
 - This last observation is important to draw out of your students. You may consider sharing with students that it is something *you* noticed if there are no students who volunteer that information.

Student responses to the second prompt, *What do you wonder*, for the second example may again vary but could include:

- What does "ol" mean?
- Why did this example make a numbered list?
- Why are both examples using the `` tag?
- Why does making a list involve two different HTML tags?

Discussion Goal: Use student responses from both rounds of *What do you notice? What do you wonder* to guide your discussion, making sure to cover the following key points:

- To make an HTML list, they need both tags - the first tag indicating the **type** of list they want and the second tag indicating a list item.
- The structure of the HTML list element is unlike other HTML elements they have used so far since there is no content that goes right next to the start `` or `` tag.
- We use indentation, or *whitespace*, in front of the `` tags to indicate that they are "children" nested within the type of list tag.
- Once they have finished listing their items using the `` tag, that is when they will finally use the closing `` or `` tag.
- If they want their list to have a title, they need to use one of the heading tags they just learned about.

Display: Use the slide to help students add list tags to their HTML Tag Chart.

`` - Starts an **Ordered List** - ``

`` - Starts an **Unordered List** - ``

`` - Actually creates the list item - ``

Have students continue Pair Programming through the List Skill Building levels.

Transition: Have your students pause programming and bring the class back together at this point in order to watch the [**Video: Debugging**](#) video together.

Questions to consider with the video:

- What is debugging?
- What are the four steps to debugging?

Discussion Goal: Students should identify debugging as the process of finding and fixing problems in their code. The four steps of the debugging process are describing the bug, hunting for the bug, trying out small solutions (or changing your code), and documenting what you have learned.

 Teaching Tip ▲

Normalizing Mistakes and Supporting Debugging: As programming levels become more complex, students may find themselves with bugs in their code that they need to untangle. If this happens frequently, this can be a demoralizing experience for students and can affect their self-perception of how capable they are in class.

To counter this, we recommend normalizing bugs and mistakes as something that happens to everyone - it's just part of the process. Additionally, consider displaying the [**Student Guide to Debugging**](#) for students to reference throughout the unit and having [**Bug Report Quarter-Sheets**](#) available for students to use.

 Teaching Tip ▲

To encourage active engagement and reflection, use one or more of the strategies discussed in the [**Guide to Curriculum Videos**](#).

Do This: Have students continue Pair Programming through the remainder of the levels.

 8

Practice

 9-10

Assessment

9

10

 Assessment Opportunity ▲

Formative Assessment: Levels 9 & 10 can be used as a formative assessment.

A rubric is provided in level 9, and written feedback can be given to students. [**Click here to learn more about giving feedback to students**](#).

Level 10 is a free response reflection that can be used as an opportunity to gain insight into how the students completed the assessment level and the choices they made with the HTML tags. In addition, this reflection can be used to identify any misconceptions shared among students that should be cleared up.

Wrap Up (5 minutes)

Journal

Question of the Day: How can we work together to fix problems with our websites?

Key Vocabulary:

- **heading** - a title or summary for a document or section of a document
- **list** - allows web developers to group a set of related items

Prompt: Today, you learned a lot about debugging, which is an important skill for programmers. What is one way working with a partner helped you to debug today?

 Assessment Opportunity ▲

Check that students are describing effective forms of collaboration.



This work is available under a [Creative Commons License \(CC BY-NC-SA 4.0\)](#).

If you are interested in licensing Code.org materials for commercial purposes [contact us](#).

Lesson 4: Digital Footprint

45 minutes

Overview

Question of the Day: How can you make sure that your private information stays private?

In this lesson, students pause their own on developing web pages so they can develop an understanding of how personal information is surfaced on the internet, sometimes intentionally and sometimes unintentionally. Students look at several fictitious social media pages to see what they can learn about different people purely from publicly available information. They then reflect on what guidelines are appropriate for posting their own information online, especially as they prepare to create more personalized websites.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

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

Agenda

[Warm Up \(5 minutes\)](#)

[Your Digital Footprint](#)

[Activity \(35 minutes\)](#)

[Investigating Social Profiles](#)

[Wrap Up \(5 minutes\)](#)

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 "Social Sleuth" for each student (or one for each group if grouping)
- Check the "[Teacher's Lounge](#)" forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our [Virtual Lesson Modifications](#)

Links

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

For the teachers

- [Digital Footprint](#) - Slides
 - ▼ Make a Copy

For the students

- [Social Sleuth](#) - Activity Guide
 - ▼ Make a Copy

Vocabulary

- **Digital Footprint** - The collected information about an individual

across multiple websites on the Internet.

Teaching Guide

Warm Up (5 minutes)

Your Digital Footprint

Prompt: Think of 2-3 websites you have accounts on. What kind of information could someone learn about you by looking on these websites?

Have students silently brainstorm or journal, then share with a partner, and finally share as a full class.

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. Based on their responses, create a list of all of the websites that may have their personal information.

💡 Teaching Tip

Extending the Discussion: If students are having trouble thinking of information, you may want to follow up with some more specific prompts:

- 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)

🎙️ Remarks

There's a lot of information that we put on the internet, whether we mean to do it or not. As we learn more about creating our own websites and start developing our own personal projects, you may find yourself wanting to include some of your own personal information. That's why today we're going to pause on making websites and instead focus on how we may want to share personal information online and making sure our digital footprint doesn't contain information that could put us in danger or make us unsafe.

📘 Key Vocabulary:

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

📘 Question of the Day: How can you make sure that your private information stays private?

Activity (35 minutes)

Investigating Social Profiles

Group: Decide if you would like to have students complete this activity individually or in groups of 2-3

Distribute: Give each student a copy of the Social Sleuth activity guide

Display: Show students the overview of the activity guide. Have students read the Sleuthing Online section out loud to get an overview of the activity they are about to complete.

Transition: Send students to Code Studio.

1

Social Sleuth

Do This: Have students look through all of the provided social media pages on Code Studio. The pages represent three unique individuals, each of whom has an account on three social media platforms. Then have students choose two users they want to investigate.

Teaching Tip

Short on Time? 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.

Do This: Have students explore the different social media pages and record the information they find in the table on their activity guide.

[pull-right]

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

[/pull-right]

Circulate: 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).

Make sure students have dedicated time before the end of class to answer the questions at the end of the activity guide, even if they don't finish completely sleuthing on the social media websites.

Teaching 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.

Share Out: Invite students to share some of their answers to the last questions on the activity guide

- What was one piece of information about either of your users that they probably did not want to share? How was it accidentally shared?

- Which of the two users do you think was more successful in protecting their privacy? What made them successful?

Discussion Goal: Student answers will vary, and it's okay to not have a class consensus. Instead, focus on the reasoning and justification students use when explaining their answers. These thoughts will segue into the wrap-up activity where students consider a “checklist” of what should or shouldn't be shared online.

Wrap Up (5 minutes)

Microphone Remarks

The activity 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.*

Discussion 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.

(Optional) 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.

Question of the Day: How can you make sure that your private information stays private?



This work is available under a [Creative Commons License \(CC BY-NC-SA 4.0\)](#).

If you are interested in licensing Code.org materials for commercial purposes [contact us](#).

Lesson 5: Mini-Project: HTML Web Page

45 minutes

Overview

In this mini-project, students use what they have learned to create a web page for a user. Students begin their project by using the Problem Solving Process to help them design a web page for a user. As they determine what content to include on the web page for their user, they will identify which tags they will use to implement their design.

Students then build their user's web page in Web Lab. Optionally, after engaging in a formal feedback process, they may change their web pages based on feedback before reflecting on their process. The lesson also includes an optional review of the HTML students have learned before building the web page.

Question of the Day: How can I use HTML to design and build a web page for a user?

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- AP - Algorithms & Programming

Agenda

Project Setup

Warm Up (5 minutes)

Activity (35 minutes)

Designing and Building Web Pages for Users

Step 1: Define - Pick a User and Define Their Needs

Step 2: Prepare - Design the Page Layout

Step 3: Try - Develop your Page

Peer Feedback

Step 4: Reflect

Wrap Up (5 minutes)

Journal

Objectives

Students will be able to:

- Build a webpage using HTML
- Design a webpage based on user needs
- Write readable code

Preparation

- Determine which project guide option your students will complete
- Print out a copy of the project guide for each student.
- Check the "[Teacher's Lounge](#)" forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our [Virtual Lesson Modifications](#)

Links

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

For the teachers

- [Mini-Project: HTML Web Page](#) - Slides ▾ Make a Copy

For the students

- [HTML Web Page Rubric](#) - Rubric ▾ Make a Copy
- [Option #1 - HTML Web Page Project Guide](#) - Activity Guide ▾ Make a Copy
- [Option #2 - HTML Web Page Project Guide](#) - Activity Guide ▾ Make a Copy

- [**Peer Review**](#) - Activity Guide
▼ Make a Copy
- [**Problem Solving Process with Programming**](#) - Resource
▼ Make a Copy

Teaching Guide

Project Setup

This mini-project focuses on assessing knowledge and skills associated with utilizing several basic HTML concepts by having students design a webpage for a user. When designing for users, it can often require the web developer to interview the user to find out what information they would like on their page. This has been replicated by providing detailed user descriptions in the project guide. Depending on the difficulty level and time constraints, there are two project guide options to choose from.

Project Guide #1: This version has detailed user descriptions with the potential topic headings already underlined and information they should include already highlighted for them. The students can look over what has been underlined and highlighted and make their design layout from there. Choose this option if your time is limited or if you have students who might need more support.

Project Guide #2: This version has detailed user descriptions with the first three potential topic headings already underlined as a starting point for the students. The remaining topic headings and the information that should be included for each topic are left up to the student to decide as the web developer. Choose this option if you have at least 2 days (1 day for planning and design and 1 day for building the webpage) or if your students would benefit from the extra challenge and web developer experience.

Make sure you select the HTML Web Page Project Guide that corresponds to the option you choose. You will also need to go into the provided slide deck and delete the slide for the choice you did *not* select (*either slide 9 or slide 10*).

Warm Up (5 minutes)

💡 Remarks

In our first lesson for this unit, we worked to design websites for others. But at the time, we were only able to design them on paper because we didn't know any HTML yet. Today, we get to apply our HTML knowledge and skills to design a web page to meet the needs of a user.

📝 Prompt: As a web developer, you will most likely create web pages for various users. Why might it be useful to have a process, such as the Problem Solving Process, that we follow when designing and creating websites for our users?

Discussion Goal: This quick discussion allows students to recall the steps of the Problem Solving Process and think about how each step is helpful when creating websites for others. Make sure to point out that having a process, such as the Problem Solving Process, will help ensure that you keep your user in mind and meet their needs.

💡 Teaching Tip

Facilitating Mini-Projects: Mini-Projects act as checkpoints in the curricula and cover the subset of skills students have seen so far in the unit. They are designed for 1-2 days of implementation as a way to check-in with how well students understand the course content so far. You may decide to extend these projects as a way to support or challenge students, which could allow you to revisit difficult concepts or

support students who may have missed lessons and are trying to catch up. However, we recommend deciding this ahead of time and being firm with students about how much time they have for each project - otherwise, it's easy for projects to drag-out to multiple days and for student's work to spiral beyond the scope of this project.

Question of the Day: How can I use HTML to design and build a web page for a user?

Activity (35 minutes)

Designing and Building Web Pages for Users

Distribute: Give each student a copy of the HTML Web Page Project Guide.

💡 Teaching Tip

Ensure your students have the HTML Web Page Project Guide corresponding to the project option you chose for them.

☰ Do This: Use the slide to go over the goals of this project

- Design a page to meet a user's needs
- Use HTML to structure a site
- Write readable code

🎤 Remarks

*Before you jump in and get started, you need to read over the different users and their needs in your Project Guide and decide which user you will create a web page for.

As you can see, you also have the option to be the “user” and create a page about how to solve or improve something at our school or in your community.*

Step 1: Define - Pick a User and Define Their Needs

☰ Do This: Direct students to pick a user based on the short user descriptions on the first page of the HTML Web Page Project Guide. Once they pick a user, they should turn to the project guide's last pages to find their user's detailed description. This includes information about what the user would like included in the web page that students will build for them.

Circulate: As you move around the classroom, encourage students to number their main topics directly on their user descriptions. This might make it easier for students to list out the topic headings on the first page of the project guide that they need to include on the web page.

💡 Teaching Tip

Make sure you have determined which project guide option your students will complete. Information for the two options can be found in the Project Setup section above.

If you have chosen Option #1, ensure students know that their user descriptions have potential topic headings already underlined for them and that the information to be included has been highlighted. Their job as web developers is now to read through the annotated user description, jot down those topics, and understand what information would go in each topic's section.

If you have chosen Option #2, ensure students are aware that their user descriptions have the first three potential topic headings already underlined for them, but it is their job as web developers to go through their user's detailed description to determine what the other main topic headings should be.

Step 2: Prepare - Design the Page Layout

Do This: Direct students to complete a sketch of their designs on the second page of the project guide.

Distribute: Give each student a copy of the HTML Web Page Rubric as they are planning the layout and tags they will use so that they can make sure they meet the requirements of the project.

Circulate: As you move around the classroom, encourage students to add numbers in the margins and within the user's description for the information they think needs to be included in each numbered topic on page 1. This might make it easier for students to list the content on the second page of the project guide that they need to include on the web page. Also, ensure your students identify which tags will be needed for different text on the web page. They are given a quick example of this in their project guide.

💡 Teaching Tip

Scoping Student Projects: Students may ideate projects that are beyond their current skills or would take longer than the allotted time to implement. Rather than asking students to choose a different user, consider asking students to imagine a more scaled-down version of their initial idea. As an analogy, if students' initial idea is the "Run" step, imagine a less intense version representing what the "Walk" step would look like. If necessary, you can keep going back further to a "Crawl" step as well.

Digging Deeper: This is sometimes referred to as the Minimal Viable Product - you can learn more about this process and adapt it to your project strategies by reading this article: [**Making Sense of MVP**](#) by Henrik Kniberg

Step 3: Try - Develop your Page

🗣 Remarks

Many of you are ready to start creating your own projects. One thing that could make this challenging is the *blank screen effect*: unlike previous levels, the only starter code you have is the tags needed to start an HTML page, like the body tag. This means you might not be sure what exactly you're supposed to do, or you might run into bugs you need to fix which can be frustrating. Luckily, we can also use the problem-solving process to help with these types of problems as well!

Distribute: Hand out a copy of the [**Problem Solving Process with Programming**](#) to pairs of students. Encourage students to look over the guide.

Display: Show the slide with the Problem Solving Process graphic

🗣 Remarks

If you feel stuck or are unsure what to do next, remember you can always follow the steps of the problem-solving process to help you when you're stuck. Don't forget though, when working with the Problem-Solving Process, not all bullets in the steps will apply to every problem, so pick out one or two from each step to see if they help.

Do This: Direct students to start building the web page for their users. Let students know that if they would like to review the tags they have learned so far before they start on their web pages, they can go to Level 1 to complete these review activities or use them as resources while they work on their projects.

Transition: Send students to Code Studio Lesson 5, Level 2 to build their pages.



Review: HTML



Code Your Page

Circulate: Encourage students to use the steps in the Problem Solving Process for Programming when they get stuck or are unsure of what to do next. Not all bullets in the Problem Solving Process will be applicable to every problem a student has. Instead, encourage them to pick one or two from each category to try each time they are stuck

💡 Teaching Tip

Debugging Strategies: As students design and implement their own project ideas, they may find themselves with new bugs that they need to untangle and you may find yourself looking at completely unfamiliar code as students look for help troubleshooting their errors. To help smooth out the debugging experience, consider the following strategies:

- Review the [**Teacher Guide to Debugging**](#) for some common questions and strategies to help support students in debugging their code
- Have students follow the steps in the [**Student Guide to Debugging**](#) and use the [**Bug Report Quarter-Sheets**](#) as an initial step in the debugging process. This helps students prepare and communicate their issue before asking for help.
- If students haven't seen it yet, consider showing the [**Debugging Video**](#) to the class to reinforce debugging best practices.

Digging Deeper: Consider supplying students with an object to talk to as part of the debugging process. This is sometimes known as Rubber Duck Debugging - you can learn more on the website <https://rubberduckdebugging.com/>

Peer Feedback

As part of this project, you may choose to allow students to give feedback to their peers using the Peer Review form. This will likely extend the project by at least one day but will allow students to reflect and make any needed improvements in Level 3 based on feedback.



Review Your Page

Do This: Direct students to review the page they built for their users. They should use Level 3 to ensure they have included everything their user needs and met the requirements of the project by consulting the project rubric one last time.

Circulate: As you walk around the class, encourage students to refer to the project rubric to ensure they have met all the requirements at a level they are happy with.

Step 4: Reflect

Do This: Direct students to complete the reflection questions on their project guide. These questions allow students to think about what they liked best about their pages, what was challenging, and what they would do differently next time. You may also want to allow students to formally or informally present their

work to the rest of the class.

✓ Assessment Opportunity ▾

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

Wrap Up (5 minutes)

Question of the Day: How can I use HTML to design and build a web page for a user?

Journal

Prompt: *What's one thing that would make your web page better that you don't know how to do yet?*

As students share their ideas, put them on the board or a poster, and try to call out any skills that they will be learning in the next few lessons.

Discussion Goal: Use this share out to get students excited about what they will learn in upcoming lessons. If you create a poster, you may also want to cross off topics as you teach them. For topics not covered in the curriculum, you can point students to similar topics that are covered or, if appropriate, direct them to outside resources that will guide them in learning those skills.



This work is available under a [**Creative Commons License \(CC BY-NC-SA 4.0\)**](#).

If you are interested in licensing Code.org materials for commercial purposes [**contact us**](#).

Lesson 6: Styling Text with CSS

45 minutes

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 an HTML page about Guinness World Record holders, adding their own style to the provided page. While only a few CSS properties are introduced in the core lesson, students are encouraged to use the optional activities at the end of the lesson to explore more ways that they can express themselves using CSS.

Question of the Day: How can we change the style of text on a web page?

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- AP - Algorithms & Programming

Agenda

Warm Up (5 minutes)

Journal: HTML Appearance

Activity (30 minutes)

Web Lab: Introduction to CSS

Wrap Up (10 minutes)

Recording CSS Properties

Objectives

Students will be able to:

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

Preparation

- Create a new poster titled **CSS Properties** if your students will not be tracking new CSS properties in their journals
- Check the "[Teacher's Lounge](#)" forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our [Virtual Lesson Modifications](#)

Links

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

For the teachers

- [Style Sheets](#) - Resource
- [Styling Text with CSS](#) - Slides
 - ▼ Make a Copy
- [Text Properties](#) - Resource

For the students

- [Intro to CSS - Part 1](#) - Video
([Download](#))
- [Intro to CSS - Part 2](#) - Video
([Download](#))

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

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

Teaching Guide

Warm Up (5 minutes)

Journal: HTML Appearance

Do This: Send students to the sample web page in Code Studio or display it on the board.

1

Sample Website

Prompt: Check out the web page on Code Studio. If you wanted to create a page like this, what do you already know how to do? What do you still need to learn how to do?

Discuss: Have students share which parts they know and don't know

Discussion Goal: Students should notice that they can get the structure and size of the text right using headings (e.g. `<h1>`) and paragraphs `<p>`. They may notice that they cannot change the color of the text. Some other styles to notice are that all the paragraphs are in italics, the speech names are underlined, and the citations are much smaller than the paragraphs.

Remarks

So far we have only made web pages where we control the content and structure, such as which parts of the pages are headings 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*.

Question of the Day: How can we change the style of text on a web page?

Activity (30 minutes)

Web Lab: Introduction to CSS

Group: Put students in pairs.

Transition: Send students to Code Studio to explore Level 2 with their partner.

💡 Teaching Tip

Guide to Programming Levels: Additional guidance for programming levels is provided in the [CSD Guide to Programming Levels](#). This document includes strategies and best-practices for facilitating programming levels with students.



Exploration

💡 Teaching Tip

Have students explore this level with a partner. Afterward, lead a short share out so that partners can share with the class their responses to the three questions in the instructions. The subsequent video should help reinforce what students discovered so there's no need to lead a lengthy debrief conversation.

🎬 **Video:** Show students the [Intro to CSS - Part 1](#) video in the slides and discuss the CSS video as a class.

💡 Teaching Tip

To encourage active engagement and reflection, use one or more of the strategies discussed in the [Guide to Curriculum Videos](#).

Questions to Consider with the video:

- How is the style of a web page different from the structure?
- Why might you want your web page to have a certain style?

Discussion Goal: Key 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

💡 Teaching Tip

Images in the Video: Around the 40 second mark, the video briefly mentions that students have learned how to add images to their website. In a previous version of the curriculum when this video was originally recorded, students learned images before learning CSS. In the current version of the curriculum, students haven't seen videos yet. It's a small moment in the video, but if students ask about it, use the moment to build excitement since students will be learning about images in just a few lessons!

🎬 **Display:** Show students the slide that displays the "Content-Structure-Style" paradigm.

Discussion Goal: 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 its 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.

Circulate: Support students as they continue through the first set of skill building levels.

 3-7

Skill Building

3

4

5

6

7

 Teaching Tip



Normalizing Mistakes and Supporting Debugging: As programming levels become more complex, students may find themselves with bugs in their code that they need to untangle. If this happens frequently, this can be a demoralizing experience for students and can affect their self-perception of how capable they are in class.

To counter this, we recommend normalizing bugs and mistakes as something that happens to everyone - it's just part of the process. You can show students our [Debugging Video](#), which includes several students normalizing mistakes and discussing debugging strategies that students can use. Additionally, consider displaying the [Student Guide to Debugging](#) for students to reference throughout the unit and having [Bug Report Quarter-Sheets](#) available for students to use.

 **Video:** Show students the [Intro to CSS - Part 2](#) video in the slides and discuss it as a class.

Question to Consider with Video:

- How does the web page know what stylesheet it should be using?

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

Circulate: Support students as they continue through the remaining levels.

 8

Skill Building

 9

Practice

 10-11

Assessment

10 
11 

Assessment Opportunity ▾

Formative Assessment: Levels 10 & 11 can be used as a formative assessment.

A rubric is provided in level 10, and written feedback can be given to students. [Click here to learn more about giving feedback to students.](#)

Level 1 is a free response reflection that can be used as an opportunity to gain insight into how the students completed the assessment level and the choices they made with the CSS rule sets and properties. In addition, this reflection can be used to identify any misconceptions shared among students that should be cleared up.

 12

Challenges

Teaching Tip ▾

Note on fonts and font families

For a web browser to display font, the **font** must be available on the device the web browser is running on. There's no guarantee that any device has a particular font, so it's much safer to use font **families**, which allows for many different fonts that have the same general look at feel.

If students want to specify 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](#).

 **Review:** Briefly review the "Content-Structure-Style" paradigm found in the "Help and Tips" area of the Code Studio levels. 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 how CSS 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

Question of the Day: How can we change the style of text on a web page?

Set Up: Have students create a new page in their journals called CSS Properties.

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.

| CSS Properties | | |
|----------------------|---|---|
| PROPERTY | WHAT IT CHANGES | EXAMPLE |
| color font-family | The color of text which font is used | color : Maroon; font-family : fantasy; |

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.

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

Share: Have groups add the properties they learned today to their new "CSS Properties" chart or to the class "CSS Properties" poster.

Key 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



This work is available under a [Creative Commons License \(CC BY-NC-SA 4.0\)](#).

If you are interested in licensing Code.org materials for commercial purposes [contact us](#).

Lesson 7: Intellectual Property

45 minutes

Overview

Until this point, the only content that students have used on their web pages is their own, but in the next lesson, they will be adding images to their sites. Before they do so, they need to understand the rules governing how to legally use content they find on the web. 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.

Question of the Day: What kind of rules protect everyone's rights when we use each other's content?

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

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

Agenda

Warm Up (5 minutes)

Use of Your Work

Activity (35 minutes)

The Creative Commons Solution

Wrap Up (5 minutes)

Your Own CC License

Objectives

Students will be able to:

- Explain the purpose of copyright.
- Identify the rights and restrictions granted by various Creative Commons licenses.

Preparation

- Preview Creative Commons Overview - Video. You may need to download it before school if YouTube is blocked.
- Check the "[Teacher's Lounge](#)" forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our [Virtual Lesson Modifications](#)

Links

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

For the teachers

- [Creative Commons Search](#) - Resource
- [Intellectual Property](#) - Slides
 - ▼ Make a Copy

For the students

- [Creative Commons Overview](#) - Video
- [Licensing Your Work](#) - Activity Guide
 - ▼ Make a Copy

Vocabulary

- **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

Teaching Guide

Warm Up (5 minutes)

Use of Your Work

Microphone icon 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 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.

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.

Discussion Goal: This discussion serves to get students thinking about the problem before introducing them to the Creative Commons solution. As students discuss the rules that they would like for their own work, make sure to emphasize and highlight the principles that correspond to the Creative Commons properties that they will explore later in the lesson.

Microphone icon 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.

Key Vocabulary:

- **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

Question of the Day: What kind of rules protect everyone's rights when we use each other's content?

Activity (35 minutes)

The Creative Commons Solution

Microphone 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.

Display: Show students the [**Creative Commons Overview**](#) video. This video is from the McLaughlin Library at the University of Guelph.

💡 Teaching Tip

To encourage active engagement and reflection, use one or more of the strategies discussed in the [**Guide to Curriculum Videos**](#).

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.

💡 Teaching Tip

As students go through the guide, the goal is not for them to memorize all aspects of the various licenses, but to have an understanding that creators can choose how their work is to be used by other people. This activity should prepare them to legally and responsibly use images that they find on the web in their own web sites.

Other Options

The second part of the guide introduces students to other licenses and scenarios that they will likely encounter as they search for images.

Graduation Cap Content Corner

Students may also be curious about how to attribute pictures that they take themselves. You can remind them that as the content creators, they can **choose** the license that they want to use, and list themselves as the creator along with their chosen license.

Choosing the Right License

The second page 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.

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.

Assessment Opportunity ▾

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.

Image Hunt

The last portion of the guide prompts students to find images of their choosing and identify the licenses under which they were published.

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:

- [Wikimedia Commons](#)
- [Google Image Advanced Search](#)
- [Flickr Creative Commons](#)
- [Wordpress Openverse](#) (formerly Creative Commons Search)

Additionally, [**this guide from the Harvard Law School Library**](#) has a variety of search engines that could be used for creative commons images. This may be overwhelming for students, but it could be a good resource for specific situations or to find additional search engines that are available at your school.

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.

Share: Allow students to share the images and licenses that they have found, whether the images could be included on a student web site, and if so, what rules the student would need to follow to use the image.

Wrap Up (5 minutes)

Your Own CC License

Question of the Day: What kind of rules protect everyone's rights when we use each other's content?

Key Vocabulary:

- **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

Journal: Think about some of the photos, drawings, and written work that you have created in the past, or even the web site that you will create in this class. Without a clear license, all of those things are covered under the fully restrictive copyright. Which Creative Commons license would you rather use (if any) and why?

- Check out the [**Creative Commons license chooser**](#)

Assessment Opportunity ▲

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



This work is available under a [**Creative Commons License \(CC BY-NC-SA 4.0\)**](#).

If you are interested in licensing Code.org materials for commercial purposes [**contact us**](#).

Lesson 8: Using Images

45 minutes

Overview

This lesson introduces images, which are different from earlier tags in that they are 'self-closing' and include attributes. Students should understand that these tags do not surround content in the same way as other tags. Students start the class by considering the ethical implications of using images on their websites, specifically in terms of intellectual property. They then learn how to add images to their web pages using the `` tag and how to cite the image sources appropriately.

Question of the Day: How can we add images on our websites, while making sure we respect everyone's rights?

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

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

Agenda

Warm Up (5 minutes)

Using Images

Activity (35 minutes)

Adding Images

Wrap Up (5 minutes)

Using Images

Objectives

Students will be able to:

- Add an image to a web page.
- Follow copyright law, accurately attributing others when using their work.

Preparation

- Check the "[**Teacher's Lounge**](#)" forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our [**Virtual Lesson Modifications**](#)

Links

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

For the teachers

- [**Creative Commons Search**](#) - Resource
- [**Images in HTML**](#) - Resource
- [**Using Images**](#) - Slides
 - ▼ Make a Copy

Introduced Code

- ``

Teaching Guide

Warm Up (5 minutes)

Using Images

Microphone Remarks

Today, we're going to add some images to our sites. This can make the websites a lot more fun, but it also means we'll need to be extra careful about making sure that we are safe and responsible in how we use them.

Prompt: In your journal, think of some guidelines we should keep in mind when using images. How can we make sure that we are safe and respecting the rights of others?

Discussion Goal: As students come up with their guidelines, you may choose to prompt them by reminding them of the recent activities they have done around sharing personal information and respecting copyright. Many classes may also want to add in other guidelines, such as defining what is appropriate in a school setting, or making sure that any people depicted in the image are comfortable with the image being posted online. Regardless, make sure that you are satisfied that the guidelines are sufficient before you move on.

Discuss: Have students journal individually, then share with a neighbor, and finally discuss as a whole class. You may choose to create a poster with the guidelines that students have created, and prompt them if anything seems missing from the list.

Question of the Day: How can we add images on our websites, while making sure we respect everyone's rights?

Activity (35 minutes)

Adding Images

Transition: Send students to Code Studio.

💡 Teaching Tip

Guide to Programming Levels: Additional guidance for programming levels is provided in the [CSD Guide to Programming Levels](#). This document includes strategies and best-practices for facilitating programming levels with students.



Exploration

Circulate: Give students time to explore the image tag and discuss the questions on the level with their partner.

Prompt:

- How do you think they knew what the name of the image should be for the src (source) attribute? Where did they find this information here in Web Lab?
- What did they put as the alternate text? Do you think that is descriptive enough?
- What else do you notice about the image tag?

Discussion Goal: Students should notice the new image file in the left-hand panel of Web Lab. This is where they can find the names of images for their web pages. Students may also notice that the image tag does not have a closing tag. You can explain to students that the image tag is a self-closing tag and is one of the few that do not need a closing tag.

Do This: Use the "Adding Images" slide to go over the three important parts needed when adding images with the `` tag.

Code Studio: Have students continue to the next levels in Code Studio

 2-6

Skill Building

2

3

4

5

6

Level 1: Students may come across some image tags that include a slash at the end, like this:

```

```

These ending slashes are there to indicate that this tag is self-closing, but they are optional and have no effect on the web page.

Level 2: Because of the type of license used, students are not legally required to include attribution, but it is a good habit for them to get into when developing their websites.

Level 5 - 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.

 Teaching Tip

A useful way to have students name images that they find in image repositories online would be to combine *description + image source + Creative Commons license type*.

An example might be: **CatWikimediaCCO.jpg**

This ensures that the students know where they found the image originally, what the search terms might have been, and that they are able to provide attribution to a source if needed.

 7

Practice

 Teaching Tip

Although this level can be used to assess student understanding of the syntax of the `` tag, its main purpose is to make students aware of the most common types of bugs they may encounter when adding images to web pages.

8

9

 Assessment Opportunity ▾

Formative Assessment: Levels 8 & 9 can be used as a formative assessment.

A rubric is provided in level 8, and written feedback can be given to students. [Click here to learn more about giving feedback to students.](#)

Level 9 is a free response reflection that can be used as an opportunity to gain insight into how the students completed the assessment level and the choices they made with adding images. In addition, this reflection can be used to identify any misconceptions shared among students that should be cleared up.

Wrap Up (5 minutes)

Using Images

Question of the Day: How can we add images on our websites, while making sure we respect everyone's rights?

 **Do This:** Have students update their HTML Tags chart in their journals to include the image tag and its attributes.

Journal Prompt: 3-2-1

- What are **3** ways you can use images to make your site better?
- What are **2** challenges in adding images to a website?
- What is **1** way you can respect people's rights when using images?



This work is available under a [Creative Commons License \(CC BY-NC-SA 4.0\)](#).

If you are interested in licensing Code.org materials for commercial purposes [contact us](#).

Lesson 9: Styling Elements with CSS

45 minutes

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 their personal websites.

Question of the Day: How can we style the images and layouts of our pages?

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

► AP - Algorithms & Programming

Agenda

Warm Up (5 minutes)

Discuss: What Styles Do You Want?

Activity (35 minutes)

Web Lab: Styling Elements with CSS

Wrap Up (5 minutes)

Objectives

Students will be able to:

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

Preparation

- Check the ["Teacher's Lounge"](#) forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our [Virtual Lesson Modifications](#)

Links

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

For the teachers

- [Body Styling](#)
- [Layout Properties](#) - Resource Page
- [Styling Elements with CSS](#) - Slides

[Make a Copy](#)

Introduced Code

- `background-color: value;`
- `border-color: value;`
- `border-radius: value;`
- `border-style: value;`
- `border-width: value;`

- **float: value;**
- **height: value;**
- **margin: value;**
- **width: value;**

Teaching Guide

Warm Up (5 minutes)

Discuss: What Styles Do You Want?

Display: Send students to the sample web page in Code Studio or display it on the board.

Prompt: What are two CSS styles on this page that you already know? What are two styles on the page that you don't know how to code yet?

1

Sample Website

Discuss: Have students share different stylings on the page.

Discussion Goal: This discussion serves as a review of the CSS properties students have already learned and showcases some of the new properties they will be learning in this lesson. Some of the new properties showcased are:

- Adding background colors to things
- Being able to move things around on the page (the images to the left of the text)
- Being able to add borders to and round corners of images

Remarks

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

Question of the Day: How can we style the images and layouts of our pages?

Activity (35 minutes)

Web Lab: Styling Elements with CSS

Transition: Send students to Code Studio.

Teaching Tip

Guide to Programming Levels: Additional guidance for programming levels is provided in the [CSD Guide to Programming Levels](#). This document includes strategies and best-practices for facilitating programming levels with students.



2

Exploration



3-6

Skill Building

3

4

5

6



7

Practice



8-9

Assessment

8

9



Assessment Opportunity ▲

Formative Assessment: Levels 8 & 9 can be used as a formative assessment.

A rubric is provided in level 8, and written feedback can be given to students. [Click here to learn more about giving feedback to students.](#)

Level 9 is a free response reflection that can be used as an opportunity to gain insight into how the students completed the assessment level and the choices they made with styling HTML elements with CSS. In addition, this reflection can be used to identify any misconceptions shared among students that should be cleared up.



10

Challenges

Wrap Up (5 minutes)

Question of the Day: How can we style the images and layouts of our pages?

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

Prompt: How might you want to use these new styles in your personal web page?



This work is available under a [Creative Commons License \(CC BY-NC-SA 4.0\).](#)

If you are interested in licensing Code.org materials for commercial purposes [contact us](#).

Lesson 10: Mini-Project: Adding Style to a Web Page

45 minutes

Overview

In this mini-project, students use what they have learned to style a web page for a user. Students begin their project using the Problem Solving Process to help them determine how they will style a web page for a user. Students then style their users' web pages in Web Lab. Optionally, after engaging in a formal feedback process, they may change their web pages based on feedback before reflecting on their process. The lesson also includes an optional review of the HTML and CSS students learned before styling the web page.

Question of the Day: How can I use CSS to style a web page for a user?

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

► AP - Algorithms & Programming

Agenda

Warm Up (5 minutes)

Activity (35 minutes)

Styling Web Pages for Users

Step 1: Define - Define Your User's Needs

Step 2: Prepare - Design the Style

Step 3: Try - Develop your Page

Peer Feedback

Step 4: Reflect

Wrap Up (5 minutes)

Journal

Objectives

Students will be able to:

- Style a webpage based on user needs
- Style a webpage using CSS
- Write readable code

Preparation

- Print a copy of the project guide for each student.
- Check the "[Teacher's Lounge](#)" forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our [Virtual Lesson Modifications](#)

Links

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

For the teachers

- [Mini-Project: Your Personal Style](#) - Slides ▾ Make a Copy

For the students

- [Adding Style to a Webpage - Project Guide](#) - Activity Guide
▾ Make a Copy
- [Adding Style to a Webpage Rubric](#) - Rubric ▾ Make a Copy
- [Peer Review - Adding Style to a Webpage](#) - Activity Guide
▾ Make a Copy

Teaching Guide

Warm Up (5 minutes)

💡 Remarks

In our first mini-project for this unit, we worked to build websites for a user. But we could only build very simple and boring pages because we didn't know how to add images or CSS. Today, we get to apply our CSS knowledge and skills to design that web page to meet our user's needs.

☰ **Distribute:** Give each student a copy of the [Adding Style to a Webpage - Project Guide](#)

☰ **Do This:** Use the slide to go over the goals of this project

- Design a page to meet a user's needs
- Add images to a web page
- Use CSS to style a web page
- Write readable code

💡 Teaching Tip

Facilitating Mini-Projects: Mini-Projects act as checkpoints in the curricula and cover the subset of skills students have seen so far in the unit. They are designed to be implemented over 1-2 days, depending on your schedule, as a way to check-in with how well students understand the course content so far.

You may decide to extend these projects as a way to support or challenge students, which could allow you to revisit difficult concepts or support students who may have missed lessons and are trying to catch up. However, we recommend deciding this ahead of time and being firm with students about how much time they have for each project - otherwise, it's easy for projects to drag-out to multiple days and for student's work to spiral beyond the scope of this project.

Question of the Day: How can I use CSS to style a web page for a user?

Activity (35 minutes)

Styling Web Pages for Users

💡 Remarks

Before you jump in and get started, you need to find the user for whom you built a webpage in Lesson 5 in your Project Guide and read over how they would like their page styled.

If you were not here for Lesson 5, select one of the NEW Users on page 2 and style a webpage for one of those users.

Step 1: Define - Define Your User's Needs

Do This: Direct students to read over their Lesson 5 user's style requests (or NEW User for those that did not complete Lesson 5) and fill in the chart on page 1 of their project guide.

Step 2: Prepare - Design the Style

Do This: Direct students to fill in the information on page 2 to help them plan out the style for each element. When finished with page 2, direct students to go to Code Studio, Lesson 10, Level 2 to see the images provided for them. Based on the images provided, they should sketch out a plan for where the best place to add each image is on page 3 of their project guide.

Distribute: Give each student a copy of the Adding Style to a Webpage Rubric as they are planning the style so that they can make sure they meet the requirements of the project.

Circulate: As you move around the classroom, ensure your students complete pages 2 and 3 of their project guide. An example of how to complete the page 3 sketch can be found in the slides.

Teaching Tip

Scoping Student Projects: Students who may ideate styles for their user's webpage that are beyond their current skills or would take longer than the allotted time to implement. Ask those students to imagine a more scaled-down and very simplified version of their initial idea and assure them that they will learn more about styling in the upcoming lessons. As an analogy, if a student's initial idea is the "Run" step, imagine a less intense version representing what the "Walk" step would look like. If necessary, you can keep going back further to a "Crawl" step as well.

Digging Deeper: This is sometimes referred to as the Minimal Viable Product - you can learn more about this process and adapt it into your project strategies by reading this article: [Making Sense of MVP](#) by Henrik Kniberg

Step 3: Try - Develop your Page

Remarks

Many of you are ready to start styling your user's webpage. Remember, if you run into bugs you need to fix, we have a handy problem solving process to help!

Teaching Tip

New Style, New Project: In order to create opportunities for students to develop a variety of projects and help emphasize their newfound CSS skills, this project is **not** linked to the previous HTML Mini-Project in Lesson 4. If students would like to build off of their previous work, they can return to their previous project and copy-and-paste their HTML Code into this project. Or, they can use this as an opportunity to build a completely new website to suit their personality.

Distribute: Hand out a copy (or ask students to pull out their copy if they were given one during the previous mini-project) of the Problem-Solving Process with Programming to pairs of students. Encourage students to look over the guide.

Display: Show the slide with the Problem Solving Process graphic

Remarks

If you feel stuck or are unsure what to do next, remember you can always follow the steps of the problem-solving process to help you when you're stuck. Don't forget though, when working with the Problem-Solving Process, not all bullets in the steps will apply to every problem, so pick out one or two from each step to see if they help.

Do This: Direct students to start styling the web page for their user. Let students know that if they would like to review the tags they have learned so far before they start on their web pages, they can go to Level 1 to complete these review activities or use them as resources while they work on their projects.

Transition: Send students to Code Studio Lesson 10, Level 2 to style their pages.

 1

Review: CSS

 2-4

Add Style to Your User's Page

2

3

4

 Teaching Tip

The Problem Solving Process: Remind students of the Problem Solving Process for Programming if they find themselves getting stuck or encountering bugs. The guide is available in the Help and Tips section of each level.

Debugging Strategies: As students design and implement their own project ideas, they may find themselves with new bugs that they need to untangle and you may find yourself looking at completely unfamiliar code as students look for help troubleshooting their errors. To help smooth out the debugging experience, consider the following strategies:

- Review the [**Teacher Guide to Debugging**](#) for some common questions and strategies to help support students in debugging their code
- Have students follow the steps in the [**Student Guide to Debugging**](#) and use the [**Bug Report Quarter-Sheets**](#) as an initial step in the debugging process. This helps students prepare and communicate their issue before asking for help.
- If students haven't seen it yet, consider showing the [**Debugging Video**](#) to the class to reinforce debugging best practices.

Digging Deeper: Consider supplying students with an object to talk to as part of the debugging process. This is sometimes known as Rubber Duck Debugging - you can learn more on the website <https://rubberduckdebugging.com/>

Circulate: Encourage students to use the steps in the Problem Solving Process for Programming when they get stuck or are unsure of what to do next. Not all bullets in the Problem Solving Process will apply to every problem a student has. Instead, encourage them to pick one or two from each category to try each time they are stuck.

Peer Feedback

As part of this project, you may choose to allow students to give feedback to their peers using the Peer Review form. This will likely extend the project by at least one day but will allow students to reflect and make any needed improvements in Level 4 based on feedback.

Do This: Direct students to review the page they styled for their user. They should use Level 4 to ensure they have included everything their user needs and met the project's requirements by consulting the project rubric one last time.

Circulate: As you walk around the class, encourage students to refer to the project rubric to ensure they have met all the requirements at a level they are happy with.

Step 4: Reflect

Do This: Direct students to complete the reflection questions on their project guide. These questions allow students to think about what they liked best about their pages, what was challenging, and what they would do differently next time. You may also want to allow students to formally or informally present their work to the rest of the class.



Assessment Opportunity



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

Wrap Up (5 minutes)

Question of the day: How can I use CSS to style a web page for a user?

Journal

Prompt: What's one thing that would make your user's web page better that you don't know how to do yet?

As students share out their ideas, put them on the board or a poster, and try to call out any skills that they will be learning in the next few lessons.

Discussion Goal: Use this share out to get students excited about the things that they will learn in upcoming lessons. If you create a poster, you may also want to cross off topics as you teach them. For topics that are not covered in the curriculum, you can point students to similar topics that are covered or, if appropriate, direct them to outside resources that will guide them in learning those skills.



This work is available under a [Creative Commons License \(CC BY-NC-SA 4.0\)](#).

If you are interested in licensing Code.org materials for commercial purposes [contact us](#).

Lesson 11: CSS Classes

45 minutes

Overview

Up until this point, the only styling students have been able to do is styling by the element, which means that every element of a particular kind has the same style. This lesson expands on the CSS that students have already learned by introducing classes, which allow web developers to treat groups of elements they want to be styled differently than other elements of the same type. Students first investigate and modify classes on various pages, then create their own classes and use them to better control the appearance of their pages. They then reflect on how they could use this skill to improve their team websites.

Question of the Day: How can we create different styles for the same type of element?

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.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

► AP - Algorithms & Programming

Agenda

[Warm Up \(5 minutes\)](#)

[Discuss: What Styles Do You Want?](#)

[Activity \(35 minutes\)](#)

[Classes](#)

[Wrap Up \(5 minutes\)](#)

[Journal](#)

Objectives

Students will be able to:

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

Preparation

- Check the "[Teacher's Lounge](#)" forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our [Virtual Lesson Modifications](#)

Links

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

For the teachers

- [CSS Classes](#) - Slides
 - ▼ Make a Copy
- [Classes](#) - Resource

Vocabulary

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

Teaching Guide

Warm Up (5 minutes)

Discuss: What Styles Do You Want?

Display: Send students to the sample web page in Code Studio or display it on the board.

Prompt: What are two CSS styles on this page that you already know? What is one thing this page does that we haven't learned how to do yet?

Discuss: Have students share their answers to the questions.

Discussion Goal: This discussion serves as a review of the CSS properties students have already learned and prompts them to think about how this page goes beyond that knowledge. Students may not realize it, but this page styles the same types of elements in different ways on different parts of the page. You may want to prompt students to think more deeply about why this is difficult by asking them what the color property of the paragraph element would be in the stylesheet for this page.

Microphone icon Remarks

So far, we've been able to style all of the elements on our page, but there's been a catch. We had to style **every** element of the same type in the same way. For example, if we wanted one paragraph to have green text, they all had to have green text. If we wanted to have one image float to the right, they all had to float to the right. Today, we're going to learn a way to get around this problem.

Question of the Day: How can we create different styles for the same type of element?

Activity (35 minutes)

Classes

Transition: Send students to Code Studio.

💡 Teaching Tip

Guide to Programming Levels: Additional guidance for programming levels is provided in the [CSD Guide to Programming Levels](#). This document includes strategies and best-practices for facilitating programming levels with students.

1

Sample Web Page

2

Exploration

3-4

Skill Building

3

4

 5

Practice

 6-7

Assessment

 6 7 Assessment Opportunity ▲

Formative Assessment: Levels 6 & 7 can be used as a formative assessment.

A rubric is provided in level 6, and written feedback can be given to students. [Click here to learn more about giving feedback to students.](#)

Level 7 is a free response reflection that can be used as an opportunity to gain insight into how the students completed the assessment level and the choices they made with adding classes and styling them in CSS. In addition, this reflection can be used to identify any misconceptions shared among students that should be cleared up.

 8

Challenges

Wrap Up (5 minutes)

Journal

Question of the Day: How can we create different styles for the same type of element?

Prompt: Think about your team website. What are two new ideas you have for your site, now that you have classes?



This work is available under a [Creative Commons License \(CC BY-NC-SA 4.0\).](#)

If you are interested in licensing Code.org materials for commercial purposes [contact us](#).

Lesson 12: Organizing Content with Flexbox

45 minutes

Overview

This lesson introduces students to two new web development tools: div tags and Flexbox styling. This lesson introduces students to div tags to organize their HTML code into sections. They then learn about Flexbox and how it can be used with their div containers to style and position items. Students practice using Flexbox properties to control the layout of web pages.

Question of the Day: What are div tags and Flexbox and why are they important in web design?

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- AP - Algorithms & Programming

Agenda

Warm Up (5 minutes)

Quick Sketch

Activity (35 minutes)

Introduction to Div and Flex

Div and Flex Practice

Wrap Up (5 minutes)

Solving Layout and Design Issues

Objectives

Students will be able to:

- Analyze and solve common web design layout challenges using div tags and Flexbox
- Apply Flexbox properties in CSS
- Identify and explain the purpose and use of div tags in HTML

Preparation

- Print copies of the handout
- If you would like more information on what Flexbox is and how to use it with CSS, [check out this video!](#)
- Check the "[Teacher's Lounge](#)" forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our [Virtual Lesson Modifications](#)

Links

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

For the teachers

- [Div Containers](#) - Resource
- [Flexbox](#) - Resource
- [Organizing My Content](#) - Slides
 - ▼ Make a Copy

For the students

- [Flexbox Cheat Sheet](#) - Handout
 - ▼ Make a Copy

Vocabulary

- **Flexbox** - a layout model in CSS for displaying items
- **div** - the HTML division tag, called “div” for short, is an element that lets you group sets of content together in a container

Introduced Code

- `<div></div>`

Teaching Guide

Warm Up (5 minutes)

Quick Sketch

Remarks

Let's warm up our design skills today with a pencil and paper. I want you to imagine and sketch how you would design a webpage about something you love. You have complete creative freedom!

 **Prompt:** Imagine you are designing a webpage for your favorite hobby, club, pet, animal, food, or something else you love. How would you arrange the following elements:

- Title
- Paragraph of information
- 6-8 images of the topic

 **Do This:** Give students 2 minutes to sketch how they would like to arrange these elements quickly. Encourage creativity and personal expression. After 2 minutes, allow students to quickly share their sketches with a partner before inviting a few volunteers to share their sketches.

Remarks

Now that we've seen some different ways to organize the title, text, and images, let's talk about them real quick.

 **Prompt:** Display the following discussion prompts and quickly discuss one at a time:

- Did you notice any common themes in how we arranged our content? What were they?
- What made you decide to place your title/images/text in that particular spot on the page?

Discussion Goal: The discussion should reveal that organizing webpage content is a crucial part of web design. The tools they will learn today are essential for creating effective layouts, so use examples from the students' sketches to segue into the concepts of div tags and Flexbox. Mention how these tools will help them achieve the layouts they've envisioned. Students should also recognize that there is no “right” way to design a webpage and that different layouts

When discussing the sketches, steer the conversation towards the functionality and layout choices rather than artistic quality. This keeps the focus on web design principles.

Keep the discussion concise and focused. While it is important to hear from several students, be mindful of the time to ensure ample opportunity for the main activity.

Microphone Remarks

Great sketches, everyone! Some great ideas for how to organize webpage content! This is a crucial part of a web developer's job which means you need the right tools to help you do this job. Today, we're going to start learning how we can bring your organization and layout ideas to life using div tags in our HTML and Flexbox in our CSS.

Question of the day: What are div tags and Flexbox and why are they important in web design?

Activity (35 minutes)

Introduction to Div and Flex (10 minutes)

Display: Use the activity slides for this lesson to introduce students to the the `div` tag. Use the Speaker Notes below as a guide to explaining what a `div` tag is, what it does, and why use it.

Look for this symbol on the slides to show when animation plays when presenting the slides:  . Make sure to preview the slides before class.

| Slides | Speaker Notes |
|---|--|
|  | <p>Say:</p> <p>A <code>div</code> tag defines a division or a section in an html document.</p> <p>It allows developers like you to split their webpage into distinct sections, each capable of taking on unique styles through CSS.</p> <p>Think of a <code>div</code> tag as a container for different sections of your html code on your page.</p> |
|  | <p>Say:</p> <p>Divs go in the body section of an html file.</p> <p> Click through animation</p> <p>Say:</p> <p>To create a <code>div</code>, you need to use an open and close tag, just as you would for a heading or paragraph tag.</p> |

| Slides | Speaker Notes |
|---|--|
|  | <p>Say: Now, the <code><div></code> tag doesn't technically do anything.</p> <p>It can help organize an HTML file into sections, but that won't affect how those sections display on our screen.</p> <p> Click through animation</p> <p>Notice how content in a <code><div></code> looks the same as the heading and paragraph not in a <code><div></code>.</p> |
|  | <p>Say: Now that you know a <code><div></code> element doesn't do anything on its own, you might wonder why you'd even use one!</p> |
|  | <p>Say: You'll use the div container to group sections of code that you want to target with CSS</p> |
|  | <p>Say: You can style items in a div to look different from other sections on the page</p> |
|  | <p>Say: ... or to position them in a specific way.</p> <p>Here we have the same HTML code in a div being displayed in two different layouts.</p> <p> Click through animation to show the same HTML code with two different positioning results</p> |

 **Do This:** Use the two “Self Check” slides to gauge student understanding. Answer any questions the students may have.

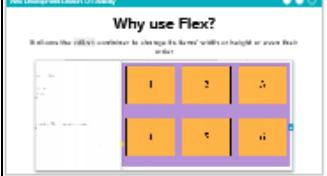
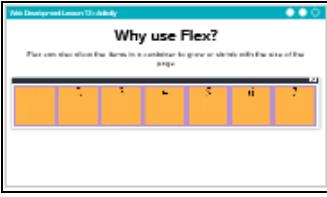
Teaching Tip



You can choose to either have students raise their hands to indicate which option they think is correct or simply have them jot their letter choice down for their own reference.

 **Display:** Continue to use the activity slides for this lesson to introduce students to Flexbox. Use the Speaker Notes below as a guide to explaining what Flexbox is.

Look for this symbol on the slides to show when animation plays when presenting the slides:  . Make sure to preview the slides before class.

| Slides | Speaker Notes |
|---|--|
|  | <p>Say: Now that we know we can use <code>div</code> tags to position the code inside of it in different ways, you are probably wondering how do we do that??</p> |
|  | <p>Say: Flexbox, or Flex for short, is how! Flex provides a more efficient way to lay out, align, and distribute space among items in a container.</p> |
|  | <p>Say: The main idea behind flex is to allow the container to alter its items' width or height and even order to fill the available space best.</p> |
|  | <p>Say: As you can see in this animation, a flex container expands the items in it to fill available free space or shrinks them to prevent overflow. Pretty handy right?</p> |
|  | <p>Say: So, how do we use Flex? Flex is the term used to refer to a whole set of CSS properties. Some are applied to the container and some are applied to the items in a container.</p> <p> Click through animation</p> <p>Today, we'll start with the flex properties for the <code>div</code> container. Who's ready to jump in and get started?!</p> |

Div and Flex Practice (25 minutes)

 **Distribute:** Pass out the [**Flexbox Cheat Sheet**](#) handout to the students.

 Teaching Tip



This handout has a section on the first page for students to write down any notes to themselves. Please encourage students to use this space as they go through the skill-building, practice, and challenge

levels to remind themselves of anything they notice and wish to remember or refer back to later.

Transition: Send students to Code Studio, Lesson 12, Level 1, and have students explore the code. Bring students' attention back to you after a few minutes of exploration.



Div & Flex Exploration

Prompt: What did you discover about the flex properties in the CSS file?

Discussion Goal: The goal of this quick discussion is to allow students to share what they figured out or what they noticed about the properties. It is okay if they didn't figure out exactly what each property does as they will learn in the next skill-building levels. Some possible discoveries might include:

- Without `display: flex;` the images change position
- Without `flex-direction: row-reverse;` the order of the images change
- Without `flex-wrap: wrap;` the images try to fit in a single row
- Without `justify-content: center;` the alignment of items within the container will change

Remarks

These 4 flex properties you just played with are the ones that you will use the most. When applying Flex properties to containers, you will typically follow three steps ...

Display: Use the “4 Basic Flex Properties” slide to explain the 3 steps they will typically follow when applying flex properties to their containers.

Teaching Tip

This information is already in the Flexbox Cheat Sheet handout for students, but you might want to direct students to highlight, circle or star these steps for quick reference later.

Group: Group students into pairs for Pair Programming.

Transition: Send students back to Code Studio and have students connect with their partner using the “Pair Programming” feature to complete the Skill Building and Practice levels.

Do This: Remind students to switch driver and navigator every 3 minutes. You may want to project a digital timer at the front of the room.



Skill Building

2

3

4

5

Teaching Tip

A slide with a timer has been provided for you. Delete this slide if you prefer a different method of keeping track of time for students so that they switch drivers and navigators often.



Practice Levels

7 8

Assessment Opportunity ▾

Formative Assessment: Levels 7 & 8 can be used as a formative assessment.

A rubric is provided in level 7, and written feedback can be given to students. [Click here to learn more about giving feedback to students.](#)

Level 8 is a free response reflection that can be used as an opportunity to gain insight into how the students completed the assessment level and the choices they made with using `div` containers and Flexbox layout. In addition, this reflection can be used to identify any misconceptions shared among students that should be cleared up.

Teaching Tip ▾

It is **highly advisable** to have your **students complete Challenge Level 9e** at least. The gallery product cards they will learn to build using `div` containers and Flexbox properties will come in **very** handy in the Chapter 1 project.

Wrap Up (5 minutes)

Solving Layout and Design Issues

Question of the Day: What are div tags and Flexbox and why are they important in web design?

Do This: Have students include the Flexbox Reference Cheat Sheet in their notebooks if they haven't done so already.

Journal Prompt: In your own words, how do div tags and Flexbox help solve page design and layout issues?

Discussion Goal: The goal of this quick reflection is to ensure students grasp the fundamental roles of div tags (structuring and grouping HTML content in a container) and Flexbox (styling and positioning elements within containers). Student responses may vary but might include design and layout issues like alignment, spacing, and organization.



This work is available under a [Creative Commons License \(CC BY-NC-SA 4.0\)](#).

If you are interested in licensing Code.org materials for commercial purposes [contact us](#).

Lesson 13: Flexbox Children for More Control

45 minutes

Overview

This lesson introduces students to advanced Flexbox child properties including `order`, `align-self`, `flex-grow`, and `flex-shrink` through an interactive warm-up and the various level exercises. They'll deepen their understanding of how these properties affect web page layouts and conclude with a creative group drawing game to reinforce their learning.

Question of the Day: How do Flexbox child properties affect the layout and organization of elements within a webpage?

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- AP - Algorithms & Programming

Agenda

[Warm Up \(5 minutes\)](#)

[Predict the Layout](#)

[Activity \(30 minutes\)](#)

[Introduction to Flex Children](#)

[Div and Flex Practice](#)

[Wrap Up \(10 minutes\)](#)

[Flex Property Pictionary](#)

Objectives

Students will be able to:

- Analyze and solve common web design layout challenges using Flexbox
- Apply Flexbox children properties in CSS

Preparation

- Print out one copy of the activity guide and handout for each student.
- Print and cut out the Pictionary Terms slips.
- If you would like more information on how to use Flexbox Children [check out this video!](#)
- Check the "[Teacher's Lounge](#)" forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our [Virtual Lesson Modifications](#)

Links

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

For the teachers

- [Flex Children for More Control](#) - Slides ▾ Make a Copy
- [Flexbox Children Properties](#) - Resource

- [Pictionary Terms Slips](#) - Resource

[▼ Make a Copy](#)

For the students

- [Flexbox Children Cheat Sheet](#) -

Handout [▼ Make a Copy](#)

- [Layout Prediction](#) - Activity Guide

[▼ Make a Copy](#)

Teaching Guide

Warm Up (5 minutes)

Predict the Layout

 **Distribute:** Pass out the [Layout Prediction](#) Activity Guide to the students.

Remarks

Let's see how well we remember what we learned last class and get a preview of some new Flex properties with a quick layout prediction game. Your challenge is to try to identify which Flex property was applied to each layout and try to predict what some new Flex properties might do! Get through as many as you can in the time provided!

Group: Group students into pairs so they have someone to discuss and share their thought processes and reasoning with.

 **Do This:** Give students 3-4 minutes to work on the activity guide.

 **Do This:** Use the slides to go over the answers. Ask students for volunteer answers.

Teaching Tip

As you review the answers to part 1, use this as an opportunity to ensure students clearly understand how each Flex property from the previous lesson affects the layout. Clarify students' misconceptions or misunderstandings (such as mixing up what `justify-content` and `align-items` do or using `justify-content` to align items horizontally when the direction is a column instead of `align-items`). The discussion should solidify their grasp of properties like `flex-direction`, `flex-wrap`, `justify-content`, and `align-content`.

As you go over student predictions for part 2, ask students about the reasons behind their predictions. Hearing different perspectives deepens understanding and promotes a collaborative learning atmosphere. This will also help foster an environment where students critically analyze why certain properties result in specific layout changes, reinforcing their knowledge and developing their problem-solving skills.

If you have time, you can expand on what each child property does as you go over the answers to part 2.

If any students have incorrect answers from part 1 or part 2, offer a clear explanation or analogy to help the student understand why their answer was incorrect and how the correct property works. For example, "Think of `justify-content` like when you are figuring out the spacing for posters across your bedroom wall and `align-items` like figuring out if you want them close to your ceiling, in the center of your wall or close to the floor" or "Think of `order` like a VIP getting to skip everyone else in line and

move to the front." Reinforce that making mistakes is a natural and valuable part of the learning process. If appropriate, ask if any other students would like to add their understanding or explanation, fostering a collaborative learning atmosphere.

Microphone Remarks

Well done! You're becoming quite the Flexbox detectives. Understanding these properties and predicting their effects is key to mastering web page layouts.

Question of the Day: How do Flexbox child properties affect the layout and organization of elements within a webpage?

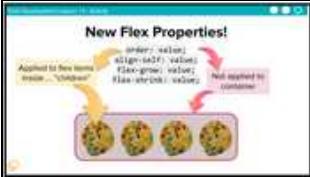
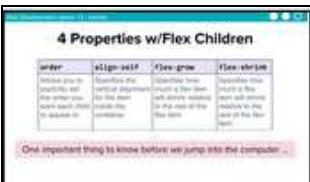
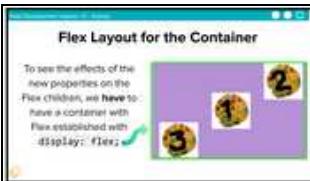
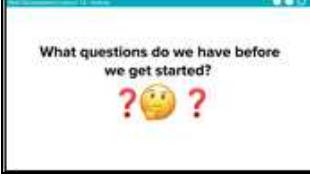
Activity (30 minutes)

Introduction to Flex Children (5 minutes)

Do This: Use the activity slides for this lesson to introduce students to the concept of Flexbox Children. Use the Speaker Notes below as a guide to explaining what Flex Children are and why you would use them.

Look for this symbol on the slides to show when animation plays when presenting the slides: . Make sure to preview the slides before class.

| Slides | Speaker Notes |
|---|---|
|  | <p>Say: After the last class, we know that we can control the layout of items, like images, by putting them inside a <code>div</code> container...</p> <p> Click through animation</p> <p>... and then aligning items in a row or in columns ...</p> <p> Click through animation</p> <p>... we have control over spacing across the page ...</p> <p> Click through animation</p> <p>... and up and down the page and whether items should wrap onto a new line or not.</p> <p>But what if we want even more control?</p> |

| Slides | Speaker Notes |
|---|--|
| | <p>Say:</p> <p>Our warm-up today gave you a preview of the new Flex properties we will work with today that will unleash even more power and control over our web page layouts.</p>  <p>Click through animation</p> <p>But these new properties won't be applied to the <code>div</code> container we learned about yesterday ...</p> <p>Click through animation</p> <p>They'll be applied to flex items <i>inside</i> the <code>div</code> container called "children".</p> |
|  | <p>Say: To have more control over flex items, we can target them directly with a class attribute and Flex children properties in CSS.</p> |
|  | <p>Say:</p> <p>We are going to learn about four properties that can be used with Flex children:</p> <p>Click through animation</p> <p><code>order</code>, <code>align-self</code>, <code>flex-grow</code>, and <code>flex-shrink</code></p> <p>Click through animation</p> <p>You already figured out what some of these do during our warm-up, but there is an important thing we need to know before we jump onto the computer and start building our Flex skills ...</p> |
|  | <p>Say:</p> <p>We have to have a container with the <code>display: flex</code> property established before any of the Flex children properties will work.</p> <p>Since these new properties are Flex children properties, that means the "parent" or the container tag they are nested in must have Flex established in order to see our layout change</p> |
|  | <p>Say: What questions do we have before we jump in and get started?</p> |

Div and Flex Practice (25 minutes)

 **Distribute:** Pass out the **Flexbox Children Cheat Sheet** handout to the students.

Transition: Send students to Code Studio, Lesson 13, Level 1 and have students explore the code.



Exploration level

 **Prompt:** What did you discover about the new flex properties?

Discussion Goal: The goal of this quick discussion is to allow students to share what they noticed or what they noticed about the properties. It is okay if they didn't figure out exactly what each new property does, as they will learn in the next skill building levels. Some possible discoveries might include:

- `flex-grow` seems to increase the size of an element
- `flex-shrink` appears to decrease the size of an element
- `order` rearranged the items
 - Students may also wonder why a `-1` put the item at the front while a `1` put the item at the end. This is because the default order value of all items is 0. A negative number will put that to the left of items with a "0" value and an item with an order value greater than 0 will be placed to the right of items with a 0 value. This information is already noted in their Cheat Sheet but it may be helpful for students to highlight or circle the information.
- `align-self` changes the vertical alignment for that one item

Remarks

These 4 flex children properties you just played with are the ones that you will use the most. Don't forget that when applying Flex properties to children, you need to have an established Flex layout in the container.

 **Display:** Use the “Flex Children Steps” slide to stress establishing a Flex layout in the container before applying Flex Children properties.

Group: Group students into pairs for Pair Programming.

Transition: Send students back to Code Studio and have students connect with their partner using the “Pair Programming” feature to complete the Skill Building and Practice levels.

 **Do This:** Remind students to switch driver and navigator every 3 minutes. You may want to project a digital timer at the front of the room.



Skill Building Levels

2

3

4

5

Teaching Tip



A slide with a timer has been provided for you. Delete this slide if you prefer a different method of keeping track of time for students so that they switch drivers and navigators often.



Practice Levels



Assessment Level

7 8

Assessment Opportunity ▲

Formative Assessment: Levels 7 & 8 can be used as a formative assessment.

A rubric is provided in level 7, and written feedback can be given to students. [Click here to learn more about giving feedback to students.](#)

Level 8 is a free response reflection that can be used as an opportunity to gain insight into how the students completed the assessment level and the choices they made with adding Flexbox children properties. In addition, this reflection can be used to identify any misconceptions shared among students that should be cleared up.



Challenge Levels

Wrap Up (10 minutes)

Flex Property Pictionary

Question of the Day: How do Flexbox child properties affect the layout and organization of elements within a webpage?

Group: Place students into groups of 3-4.

Distribute: Pass out the Pictionary Terms slips already cut up and in containers, blank pieces of paper, and a few markers to each group.

Teaching Tip ▲

You should cut out the slips ahead of time and place them into some type of container for each group, such as a plastic cup. This will save time and give the students a container to pull slips from during the activity.

The blank piece of paper and markers are where the students will be drawing their properties. If you have enough small dry-erase boards for each group, you might consider using those instead.

Optional Scoring: If you would like this activity to have a competitive feature, you can instruct students that whoever guesses correctly keeps the slip of paper. At the end of the activity, students would count the number of slips they had to see who guessed the most properties correctly.

Display: Use the “Flex Property Pictionary” slide to help you quickly review the activity's instructions.

Do This: Direct students to play the Pictionary activity until the time is up or they go through all of their slips.

 **Journal Prompt:** How did playing this game with your group help you understand Flexbox properties better?

Discussion Goal: This quick reflection aims to highlight the value of collaborative learning and peer interaction. This reflection aims to inform students of how sharing ideas, discussing concepts, and working together (even by playing a game) can enhance their comprehension and problem-solving skills, especially in complex subjects like web design.

 **Do This:** Have students include the Flexbox Children Cheat Sheet Reference handout in their notebooks if they haven't done so already.



This work is available under a [Creative Commons License \(CC BY-NC-SA 4.0\)](#).

If you are interested in licensing Code.org materials for commercial purposes [contact us](#).

Lesson 14: Chapter 1 Project

90 minutes

Overview

In this chapter project, students use what they have learned to style and organize a web page for a user. Students begin their project by using the Problem Solving Process to help them determine how they will organize the content of a user's web page. Students then use CSS classes and flexbox to organize their user's web page in Web Lab. Optionally, after engaging in a formal feedback process, they may change their web pages based on feedback before reflecting on their process.

Question of the Day: How can I use CSS classes and Flexbox to organize webpage content for a user?

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

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

Agenda

Project Setup

Warm Up (5 minutes)

Debugging 3-2-1

Activity (75 minutes)

Final Product for Users

Step 1: Define - Define Your User's Needs

Step 2: Prepare - Design the Page

Step 3: Try - Develop your Page

Chapter 1 Project

[Optional] Peer Feedback

Step 4: Reflect

[OPTIONAL] Publishing Your Website

Wrap Up (10 minutes)

Journal

Objectives

Students will be able to:

- Organize webpage content using flexbox
- Style a webpage using CSS
- Write readable code

Preparation

- Determine which project option your students will complete.
- Print copies of the project guide and any accompanying handouts for your students
- Print copies of the reflection and peer feedback documents.
- Check the "[Teacher's Lounge](#)" forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our [Virtual Lesson Modifications](#)

Links

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

For the teachers

- [Web Development - Extra Code in Challenge Levels](#) - Resource
▼ Make a Copy

For the students

- [Option 1 - Project Rubric](#) - Resource
▼ Make a Copy
- [Option 1 Chapter 1 Project Guide](#) - Activity Guide
▼ Make a Copy

- [Option 1 Student Checklist - Chapter 1 Project](#) - Resource
▼ Make a Copy
- [Option 2 - Project Rubric - Chapter 1 Project](#) - Resource
▼ Make a Copy
- [Option 2 Chapter 1 Project Guide](#) - Activity Guide
▼ Make a Copy
- [Option 2 Student Checklist - Chapter 1 Project](#) - Resource
▼ Make a Copy
- [Option 3 - Project Rubric - Chapter 1 Project](#) - Resource
▼ Make a Copy
- [Option 3 Chapter 1 Project Guide](#) - Activity Guide
▼ Make a Copy
- [Option 3 Student Checklist - Chapter 1 Project](#) - Resource
▼ Make a Copy
- [Option 3 User Web Page Content](#) - Handout
▼ Make a Copy
- [Peer Review - Chapter 1 Project](#) - Activity Guide
▼ Make a Copy
- [Problem Solving Process with Programming](#) - Resource
▼ Make a Copy
- [Publishing Your Website](#) - Video
[\(Download\)](#)

Teaching Guide

Project Setup

This chapter project assesses knowledge and skills of utilizing CSS Classes and flexbox concepts. Adding Flexbox concepts to an already created web page can often require the web developer to shift and move around the HTML content for the page to look as expected. As a result, this project has 3 options for you to choose from depending on the difficulty level and time constraints.

Option 1 - Easy: Students add classes and flexbox to web page content that is already organized with `<div>` elements. Choose this option if you don't have time available to allow your students to either alter the code that will be brought over from the Lesson 10 mini-project or build a new web page for a new user from scratch.

Option 2 - Intermediate: Students add `div` tags, classes, and Flexbox to the webpage they styled for Lesson 10. Choose this option if your students are up for the challenge and you have the extra time (possibly 2 days) to allow students to shift their HTML code, iterating on their HTML to make their page look as their user requests.

Option 3 - Challenging: Students create a webpage for a new user from scratch. Choose this option if you have at least 2-3 days to dedicate to this project and would like students to have the experience of building a webpage from beginning to end without breaks in between the process.

Select the Chapter Project Guide, Scoring Rubric, and Presentation Slides corresponding to your chosen option.

Depending on a teacher's specific timing and students, you *could* further differentiate this project for your students by allowing them to pick the option they feel they are ready for. However, remember that students choosing Option 3 may require more time to complete their project than those choosing Option 1.

Warm Up (5 minutes)

Debugging 3-2-1

🎙️ Remarks

In this lesson, you'll be adding CSS Classes and Flexbox concepts to the content of a web page in order to add more style and organization. Once we start adding classes and flexbox properties, it can be easy to end up with some bugs in our code. This is one of the biggest parts of coding ... debugging.

📝 **Prompt:** Use the slide to display the 3 discussion questions:

- What are the top **3** bugs you have seen so far when making web pages?
- What are **2** things you can do while coding to make bugs easier to find and fix?
- What is the **1** piece of advice you'd give someone who has a bug?

Discussion Goal: The purpose of this discussion is both to normalize debugging and to make sure students start the coding part of the project with some good strategies that will help them to debug. The following are possible student responses:

- **Common bugs** they may have seen might include:
 - Spelling errors
 - Forgetting to link the style sheet
 - Syntax errors such as forgetting the "." in the stylesheet when styling a class
 - *This will vary from class to class, but most students will have encountered multiple bugs in the chapter.*
- Things they can do while coding to **make bugs easier to find and fix** might include:
 - Use proper formatting to help them to read their code
 - Code small bits at a time and check as they go
 - Use good naming conventions
 - *Again, this will vary from class to class.*
- For **debugging advice**, students may give the following types of advice:
 - Dispositional advice ("Don't give up!", "Remember that it's normal")
 - Recommended resources ("Ask a friend to look at it", "look at the map levels")
 - Process-based advice ("describe the bug first, then look for it before you change stuff", "look at what's messed up and check the spelling in that part")
 - Tools based advice ("look above where the pink is", "use the inspector tool").
 - *Give students time to share their advice, and make sure that they are referencing the Debugging Process.*

Give students a few minutes to brainstorm some ideas, then allow them to share with the class. You may want to project their ideas and leave the display up as the students code their pages.

Question of the Day: How can I use CSS classes and Flexbox to organize webpage content for a user?

Activity (75 minutes)

Final Product for Users

- **Distribute** Give each student a copy of the Chapter 1 Project Guide.

💡 Teaching Tip

Ensure your students have the Chapter 1 Project Guide corresponding to the project option you chose for them.

- **Do This:** Use the "Project Goals" slide to go over the goals of this project

💡 Teaching Tip

Ensure you are using the Chapter 1 Project Slides corresponding to the project option you chose for your students.

Step 1: Define - Define Your User's Needs

- **Do This:** Direct students to read over and follow the **Define** step instructions on their project guide, which will be somewhat different depending on the project level you chose.

Step 2: Prepare - Design the Page

- **Do This:** Direct students to fill in the **Prepare** section information on the project guide to help them plan out the task ahead.

- **Distribute:** Give each student a copy of the Chapter 1 Project Rubric that corresponds to the project option they are working on. They should have this resource as they are planning so that they can make sure they meet the requirements of the project.

Circulate: As you move around the classroom, ensure that your students are referring to the rubric as they are completing the Prepare section of their project guide

💡 Teaching Tip

Scoping Student Projects: Even at this stage, students may ideate styles for their user's webpage that are beyond their current skills or that would take longer than the allotted time to implement. Ask those students to imagine a more scaled-down and very simplified version of their initial idea and assure them that they will learn about concepts such as hyperlinks and creating buttons in the upcoming lessons. As an analogy, if students' initial idea is the "Run" step, imagine a less intense version that represents what the "Walk" step would look like. If necessary, you can keep going back further to a "Crawl" step as well.

Digging Deeper: This is sometimes referred to as the Minimal Viable Product - you can learn more about this process and adapt it into your project strategies by reading this article: [Making Sense of MVP](#) by Henrik Kniberg

Step 3: Try - Develop your Page

🎤 Remarks

Many of you are ready to start working on your user's webpage. Remember, if you run into bugs you need to fix, we have a handy problem-solving process to help!

Distribute: Hand out a copy (or ask students to pull out their copy if they were given one during the previous mini-project) of the Problem-Solving Process with Programming to students. Encourage students to look over the guide.

Display: Show the slide with the Problem Solving Process graphic

Remarks

If you feel stuck or are unsure what to do next, remember you can always follow the steps of the problem-solving process to help you when you're stuck. Don't forget though, when working with the Problem-Solving Process, not all bullets in the steps will apply to every problem, so pick out one or two from each step to see if they help.

Do This: Direct students to start working on the web page for their user.

Transition: Send students to Code Studio Lesson 14. The level they will work on will correspond to the project option you chose for them.

- If your students are doing project option 1, they will go to Level 1 and pick the option for the user they chose.
- If your students are doing project option 2, they will go to Level 2 and pick the option for the user they've been working with in previous mini-projects.
- If your students are doing project option 3, they will go to Level 3 and *complete each level a-e*.

Ensure students are selecting the correct level for the project option they are working on, Option 1 - Easy, Option 2 - Intermediate, or Option 3 - Challenging.

Chapter 1 Project

 1-3

1 2 3

Circulate: Encourage students to use the steps in the Problem Solving Process for Programming when they get stuck or are unsure what to do next. Not all bullets in the Problem Solving Process will apply to every problem a student has. Instead, encourage them to pick one or two from each category to try each time they are stuck.

 Teaching Tip



New Code and Previous Challenge Levels: students may have learned additional code in the challenge levels of different lessons throughout the unit. As students approach this project, they may want to revisit the code they learned in these levels or visit them for the first time to learn new codes to use in this project. To help guide students back to previous levels, you can use the [Extra Code in Challenge Levels](#) as a resource to quickly find where new codes were introduced in earlier challenge levels.

[Optional] Peer Feedback

As part of this project, you may choose to allow students to give feedback to their peers using the Peer Review form. This will likely extend the project by at least one day but will allow students to reflect and make any needed improvements based on feedback.

Do This: Direct students to review the page they worked on for their user. They should make sure they have included everything their user needs as well as met the requirements of the project by consulting the project guide and project rubric one last time.

Circulate: As you walk around the class, encourage students to refer to the project rubric to ensure they have met all the requirements at a level they are happy with.

💡 Teaching Tip

Rubric and Checklist: Students can use two resources for self-reflection and ensuring they are on the right track: the rubric and the student checklist. We recommend having students use the checklist for their own self-assessment and reflection, since it may be easier to digest and understand when reviewing their own project. However, we recommend teachers use the full rubric for evaluating projects to give more nuanced and specific feedback to students.

Step 4: Reflect

Do This: Direct students to complete the reflection questions on their project guide. These questions allow students to think about what they liked best about their pages, what was challenging, and what they would do differently next time. You may also want to allow students to formally or informally present their work to the rest of the class.

✓ Assessment Opportunity

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

Collect: Project Guides and Peer Review Guides.

[OPTIONAL] Publishing Your Website

Video: Show students the [**Publishing Your Website**](#) video in the slides.

💡 Teaching Tip

To encourage active engagement and reflection, use one or more of the strategies discussed in the [**Guide to Curriculum Videos**](#).

Questions to Consider with Video:

- 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?

Discussion Goal: 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 fully 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 right to use all the content on the page.

In addition, as they began to share, they should also think about how they respect the work of other website publishers, particularly 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.

Wrap Up (10 minutes)

Journal

Question of the Day: How can I use CSS classes and Flexbox to organize webpage content for a user?

Prompt: Reflect on your entire experience as a web developer and creating a webpage for a user.

- How did you use the problem-solving process in creating your user's webpage?
- What skills and practices did you implement as you made your user's webpage?
- What's one thing that would improve your user's webpage that you still don't know how to do yet?

Discussion Goal: Students should reflect on the process so far. Students may mention specific skills such as using documentation, planning, teamwork, and debugging. Prompt students to explain how these practices helped them as they coded. You may consider reminding them of the various HTML and CSS skills they implemented and let me know about the five practices of computer science: problem-solving, persistence, creativity, communication, and collaboration.

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.



Reflection



This work is available under a [Creative Commons License \(CC BY-NC-SA 4.0\)](#).

If you are interested in licensing Code.org materials for commercial purposes [contact us](#).

Lesson 15: Team Problem Solving

45 minutes

Overview

This lesson explicitly addresses the challenges students may find working in a group and supports them in crafting a plan to overcome these challenges. Students work together to set group norms and brainstorm what features they would like their websites to have. The class starts by thinking of some popular teams in different contexts, then reflects on what makes teams successful. They then get into their own teams and make a plan for how they will interact and reach success in their own projects. Afterward, the teams begin to brainstorm ideas for their website project.

Question of the Day: How can we work together to make a great team?

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

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

Agenda

Warm Up (5 minutes)

Activity (35 minutes)

Team Goal

Skills and Goals

Bringing Different Ideas Together

Looking forward

Wrap Up (5 minutes)

Shout-outs!

Objectives

Students will be able to:

- Communicate and collaborate with classmates in order to solve a problem

Preparation

- Check the "[Teacher's Lounge](#)" forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our [Virtual Lesson Modifications](#)

Links

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

For the teachers

- [Team Problem Solving](#) - Slides
▼ Make a Copy

For the students

- [Video: Managing Disagreement](#) - Video ([Download](#))
- [Video: Teamwork](#) - Video ([Download](#))
- [Website Team Plan](#) - Activity Guide
▼ Make a Copy

Teaching Guide

Warm Up (5 minutes)

Game: Top Five Teams

 Teaching Tip ▲

You may also choose to have this game displayed as the class walks in, and allow them to write down their answers at their own pace.

 Remarks

Today is the first day that you get to work in your teams. Before we do that, we're going to think of some of the top teams in the world. When I say a category of team, everyone should write down the team you think **most of the class** will answer as the top team. If you choose the most popular team, you get a point. Remember, it's not the one you think is the top, but the one you think most people will answer.

1. Top Music Team (band or singing group)
2. Top Sports Team
3. Top Superhero Team
4. Top Villain Team
5. Top Kid-only Team

Have students share out their answers after each one, and note the most popular answer, giving "points" to those who guessed correctly.

Prompt: We've mentioned some very successful teams. Write down three things that you think made these teams so successful.

Share: Allow students some time to write down their own answers, then have them share with the rest of the class. Highlight characteristics that will be helpful when they are working with their teams on the project.

Discussion Goal: This game and discussion should prime students to think about strategies for successful teamwork on the project. Feel free to swap out the categories for ones that make more sense for your own classroom.

Question of the Day: How can we work together to make a great team?

Activity (35 minutes)

Group: Place students in their project groups.

 **Display:** Show students the [Video: Teamwork](#) video in the slides.

 Teaching Tip ▲

To encourage active engagement and reflection, use one or more of the strategies discussed in the [Guide to Curriculum Videos](#).

Questions to Consider with Video:

- What are the advantages to working in a team?
- What are some tips for teams to work well together?

Discussion Goal: The goal of this video is to help students come into teamwork with a positive attitude and give them a few suggestions for how to make their teams run more smoothly. Students may have different ideas for the advantages of working on a team, but they should note that people can get more done in a team and that bringing in a diversity of ideas will make a project better.

💡 Teaching Tip

Teamwork Tips: The video includes several tips on how to work as a team. Consider recording these rules on a poster or a small handout that students can refer back to regularly. The tips mentioned in the video are:

- Have a plan
- Give everyone a special role
- Celebrate successes of your teammates
- Have fun!
- Figure out everyone's strengths
- Listen to and respect everyone's ideas
- Be confident in sharing your ideas

Team Goal

Distribute: Hand out the activity guide.

The teams should start by discussing possible topics for their project and eventually, they should make a decision and all be in agreement about the topic.

Skills and Goals

Each student should identify their own personal strengths and areas of growth for the project, then share those out with the group.

💡 Teaching Tip

This type of personal reflection and sharing may be difficult for some students. Depending on the class, you may want to take some time to review how to listen respectfully to others in the group. This first activity is a good chance for students to practice this type of listening. Some groups may need teacher intervention to explain what sorts of specific behaviors promote good group dynamics.

Students use what they've learned to come up with a plan for their group and how they will support each other.

Bringing Different Ideas Together

🎙️ Remarks

Part of working in groups is having lots of different ideas, and team members don't always agree on how to bring their ideas together. We're going to explore some different ways that a team can manage disagreement to make their product even better.

▀ **Video:** Show students the [Video: Managing Disagreement](#) video in the slides.

Question to Consider with Video:

- What are three tips for dealing with disagreement?

Discussion Goal: Students should understand that disagreement is a normal part of teamwork and have strategies for constructively dealing with disagreement in their teams.

Students plan how they will deal with disagreement and come up with strategies for advocating for themselves and their team members.

💡 Teaching Tip



Collaboration and Inclusivity: Fostering an inclusive culture and collaborating around computing are core practices in the CS K12 Framework. Students may need guidance on effective strategies for inclusion and collaboration. For example, some students may suggest a "majority rules" approach to disagreement. Challenge students to think of how that may exclude valuable perspectives and whether there are better strategies that can incorporate the entire team's voice.

Tips for Handling Disagreement: The video includes several tips for handling disagreement. Similar to the last video, consider having a poster or small handout that students can refer to with these tips. Examples include:

- Appreciate that your team has lots of ideas, even if you don't agree with all of them.
- Be curious about the other ideas, and try to understand what is good about them.
- Compromise with your teammates, taking good parts of lots of ideas and putting them together.

Looking forward

Students define what success will look like in the context of the project and anticipate what will be the most challenging and most fun parts of the project.

Circulate: As students are completing or have completed the guide, circulate around the room and ask them about their plans for working together in a group. You may want to point out the advantages and disadvantages of certain strategies.

Share: Have different groups share out aspects of their plans. You may want to call out specific effective strategies that you saw during the activity.

✔️ Assessment Opportunity



Formative Assessment: You can use the short answer responses on the second page of the activity guide to assess student mastery of the learning goals of this lesson, in particular their ability to work as a team.

Brainstorming

Discuss: Give students time to brainstorm ideas for their project in their teams. They can talk about different content or website features that they would like. As they discuss, keep an eye out for potential problems in group dynamics and intervene if necessary. Remind team members of their plans and how they can contribute to a successful team.

Wrap Up (5 minutes)

Shout-outs!

Question of the Day: How can we work together to make a great team?

Prompt: When you work in a team, it's important to recognize when your team members do something great. For each member of your team, write down one "shout-out" that recognizes a special way they contributed to the group today.



This work is available under a [**Creative Commons License \(CC BY-NC-SA 4.0\).**](#)

If you are interested in licensing Code.org materials for commercial purposes [**contact us**](#).

Lesson 16: Sources and Research

45 minutes

Overview

This lesson encourages students to think more about their responsibilities as consumers of information and how to find relevant and trustworthy information online. After viewing and discussing a video about how search engines work, students will search for information relevant to their site. They'll need to analyze the sites they find for credibility to decide which are appropriate to use on their own website. 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 as content producers by looking for ways to make their own sites appear more trustworthy.

Question of the Day: How do we find relevant and trustworthy information on the Internet?

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- **IC** - Impacts of Computing

Agenda

[Warm Up \(10 minutes\)](#)

[Internet Scavenger Hunt](#)

[Activity \(30 minutes\)](#)

[Research and Trustworthiness](#)

[The Trustworthiness Checklist](#)

[How Search Works](#)

[Wrap Up \(5 minutes\)](#)

[Journal](#)

Objectives

Students will be able to:

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

Preparation

- Print a copy of the Internet Scavenger Hunt activity guide for each pair of students, or prepare to project the questions to the class.
- Create a blank poster titled **Trustworthiness Checklist** and place it on the wall.
- Check the "[Teacher's Lounge](#)" forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our [Virtual Lesson Modifications](#)

Links

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

For the teachers

- [Sources and Research](#) - Slides
▼ Make a Copy

For the students

- [How Search Works](#) - Video
([Download](#))

- [Internet Scavenger Hunt](#) - Activity Guide ▾ Make a Copy
- [Links and Research](#) - Activity Guide ▾ Make a Copy

Vocabulary

- **Relevant** - closely connected to a topic
- **Search Engine** - A program that searches for items on the World Wide Web.
- **Trustworthy** - reliable, honest, and truthful

Teaching Guide

Warm Up (10 minutes)

Internet Scavenger Hunt

Group: Place students in pairs. They do not need to be in their project groups for this warm up, but will transition to those groups for the main activity.

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

💡 Teaching Tip

Reducing Printing: Rather than print an activity guide for each student, consider displaying the questions and allowing students to answer in their journals or on scrap paper.

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

🎤 Remarks

You have seven 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 and how did you find them?
- How do you know that the information was accurate?

Discuss: Student pairs share out what they were able to discover during the scavenger hunt and what strategies worked well for finding accurate information. If possible, use the discussion to introduce the key vocabulary for the lesson.

Discussion Goal: These 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 discussing basic search techniques. Keep track of search strategies on the board for reference later.

As students discuss the search engines that they used in the process, you may want to challenge them to think about how the search engines are able to give them relevant results. This can provide a good transition to the video in the next activity.

Key vocabulary:

- relevant - closely connected to the topic (answers the question)
- trustworthy - reliable, honest, and truthful (gives a correct answer)
- search engine - a program that searches for items on the World Wide Web

Question of the Day: How do we find relevant and trustworthy information on the Internet?

Activity (30 minutes)

Research and Trustworthiness

Remarks

As you begin to build your sites, you may want to use information that you find on the web, or link to other sites relevant to the problem that you are trying to solve. As you do that, you'll need to think about whether the information you are using is trustworthy.

Group: Place students in their project groups.

Distribute: Give each pair one copy of the Links and Research activity guide, or hand out the journals.

Search: Give student groups about ten minutes to search online and fill in the table on the activity guide.

Share: Ask student groups to share out the different sites that they found and which they decided were the most and least trustworthy and why. Push students to give detailed reasons for why they trust a site or not, and keep track of them on the board.

Search: Give students a few more minutes to look through their source sites again with the new criteria from their classmates. If there is more class time, students may want to continue to find sources for their site.

The Trustworthiness Checklist

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

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.

Have students record their responses on the bottom of their activity guides

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.

How Search Works

Video: Show students the [**How Search Works**](#) video in the slides.

Teaching Tip

To encourage active engagement and reflection, use one or more of the strategies discussed in the [**Guide to Curriculum Videos**](#).

Wrap Up (5 minutes)

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.

Journal

Prompt: 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?



This work is available under a [**Creative Commons License \(CC BY-NC-SA 4.0\)**](#).

If you are interested in licensing Code.org materials for commercial purposes [**contact us**](#).

Lesson 17: Linking Pages

45 minutes

Overview

This lesson will give students practice in using links and introduce them to good navigation practices for their sites. Students begin by looking online for the first web page and discussing how its use of links was what started the web. They then transition to Web Lab where they learn how to make their own links, as well as good conventions that make it easier for users to navigate on a page. Finally, they reflect on their group project and what their personal goals are for the final stretch.

Question of the Day: How can we combine several different web pages into one website?

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

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

Agenda

Warm Up (5 minutes)

First Web Page Hunt

Activity (35 minutes)

Web Lab

Wrap Up (5 minutes)

Journal

Objectives

Students will be able to:

- Use HTML to link between web pages.

Preparation

- Make sure students have access to their website plans from the previous lesson.
- Check the "[**Teacher's Lounge**](#)" forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our [**Virtual Lesson Modifications**](#)

Links

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

For the teachers

- [**Hyperlinks**](#) - Resource
- [**Linking Pages**](#) - Slides
 - ▼ Make a Copy

Introduced Code

- `<a>`

Teaching Guide

Warm Up (5 minutes)

First Web Page Hunt

Group: Place students in pairs. They do not need to be in their project groups.

Display: Show the two following prompt questions on the board, and direct students to find the answers online by using a search engine of their choice, or whatever methods they might normally use for research.

1. What was the first web page ever created?
2. What's the one feature it has that we haven't learned how to do yet?

💡 Teaching Tip



Although there are many places students could go for the answer to this question, a Google search will most likely take them to an article describing the page that Tim Berners-Lee created on August 6, 1991. The page is comprised of simple text with a few hyperlinks.

World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#) , [Policy](#) , November's [W3 news](#) , [Frequently Asked Questions](#) .

What's out there?

Pointers to the world's online information, [subjects](#) , [W3 servers](#), etc.

Help

on the browser you are using

Software Products

A list of W3 project components and their current state. (e.g. [Line Mode](#) ,[X11 Viola](#) , [NeXTStep](#) , [Servers](#) , [Tools](#) , [Mail robot](#) , [Library](#))

Technical

Details of protocols, formats, program internals etc

Bibliography

Paper documentation on W3 and references.

People

A list of some people involved in the project.

History

A summary of the history of the project.

How can I help ?

If you would like to support the web..

Getting code

Getting the code by [anonymous FTP](#) , etc.

Share: After giving students time to search online, let them share out their answers.

🎙 Remarks

Even though the first web page looks simple, it had something that had never been done before: links. Links are what made the web special, because they let people easily move around the Internet. Today, you'll learn how to put links on your own web pages, so that your team can combine all your pages together into one website.

Question of the Day: How can we combine several different web pages into one website?

Activity (35 minutes)

Web Lab

Group: Allow students to continue in their pairs.

Transition: Send students to Code Studio.

💡 Teaching Tip

Guide to Programming Levels: Additional guidance for programming levels is provided in the [CSD Guide to Programming Levels](#). This document includes strategies and best-practices for facilitating programming levels with students.



Exploration



Skill Building

2

3

4



Practice



Adding Pages

6

7



Assessment Opportunity

Formative Assessment: Levels 6 & 7 can be used as a formative assessment.

A rubric is provided in level 6, and written feedback can be given to students. [Click here to learn more about giving feedback to students.](#)

Level 7 is a free response reflection that can be used as an opportunity to gain insight into how the students completed the assessment level and the choices they made with adding HTML files and hyperlinks. In addition, this reflection can be used to identify any misconceptions shared among students that should be cleared up.

Wrap Up (5 minutes)

Journal

Question of the Day: How can we combine several different web pages into one website?

Do This: Have students update their HTML Tags chart in their journals to include the hyperlink tag and its attribute.

Prompt: Today, you learned how to link pages, which will help your team work together to build one website. Besides all linking to each other, what else will help all your pages feel like a single site?

Discuss: After students have time to write down their thoughts, allow them to share in small groups before taking ideas from the whole class.

Discussion Goal: Students may come up with ideas such as a similar color scheme or styles, or similar content. They may mention a header at the top of the page or navigation bar. The most important thing is that they are thinking of the site as a single project that they are all contributing to, rather than a set of individual web pages.



This work is available under a [Creative Commons License \(CC BY-NC-SA 4.0\)](#).

If you are interested in licensing Code.org materials for commercial purposes [contact us](#).

Lesson 18: CSS Pseudo-classes

45 minutes

Overview

This lesson introduces students to the dynamic world of CSS pseudo-classes. Students will learn about their role in enhancing web page interactivity and style. They will explore, practice, and apply various pseudo-classes like `link`, `visited`, `hover`, and `active`, understanding how these selectors can transform the user experience on websites.

Question of the Day: What is a CSS pseudo-class, and how does it change the way an element looks or behaves on a web page?

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- AP - Algorithms & Programming

Agenda

Warm Up (5 minutes)

Explore Pseudo-classes

Activity (35 minutes)

Pseudo-class Practice

Wrap Up (5 minutes)

Teaching Guide

Warm Up (5 minutes)

Explore Pseudo-classes

🎙 *Remarks*

Objectives

Students will be able to:

- Apply pseudo-classes to selectors in CSS to enhance website interactivity and design
- Explain the function and importance of CSS Pseudo-classes

Preparation

- Check the "[Teacher's Lounge](#)" forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our [Virtual Lesson Modifications](#)

Links

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

For the teachers

- [CSS Pseudo-classes](#) - Slides
 - ▼ Make a Copy
- [Pseudo-classes](#) - Resource

Today, we're going to explore CSS pseudo-classes, a powerful tool that brings interactivity and style to our web pages. Let's dive in on our computers and see firsthand how these pseudo-classes can transform the look and feel of a website with just a few lines of code!

Transition: Send students to Code Studio, Lesson 18, Level 1, and have students explore the code



Exploration Level

Circulate: As students are exploring the code, walk around to ensure students are discussing with their neighbor. Listen for and give positive affirmations to what students are noticing about the selector of the `<a>` tag. This could include the fact that there is a colon after the `<a>` selector, there are multiple selectors for the `<a>` tag, or noticing what effects different rule sets are having on the hyperlink.

Prompt: How can adding these styles impact a user's experience on a website?

Discussion Goal: This quick discussion aims to foster an understanding of how CSS pseudo-classes can enhance user experience and website usability. It aims to make students aware of the practical implications of styling links for aesthetic purposes and improving navigation, accessibility, and overall user engagement on a web page.

Question of the Day: What is a CSS pseudo-class, and how does it change the way an element looks or behaves on a web page?

Activity (35 minutes)

Pseudo-class Practice

Remarks

Before you continue on to the rest of the levels, there's one very important thing you need to know when using pseudo-classes with links - the order matters! You will see this in your first Skill Building level but let's go over it real quick.

Display: Use the "Link Pseudo-classes" slides to explain the order in which those pseudo-classes must be applied for anchor tags.

Teaching Tip

All of these pseudo-classes can apply to a hyperlink; in some cases, more than one will apply. For example, an unvisited link can be hovered and active simultaneously as it's an unvisited link. Since these four rules apply to the hyperlink, and the selectors all have the same specificity, then the last one listed that matches what the user is doing, wins.

If we mix up the order, for example, this means that if "hover" was after "active," the "active" style will never appear because the "hover" style will always override it. For this reason, the recommended order is

`a:link` `a:visited` `a:hover` `a:active`

Here's a fun way for students to remember the order - Just have them remember **LOVE (LV) & HATE (HA)**

Group: Group students into pairs for Pair Programming.

Transition: Send students back to Code Studio and have students connect with their partner using the “Pair Programming” feature to complete the Skill Building and Practice levels.

Do This: Remind students to switch driver and navigator every 3 minutes. You may want to project a digital timer at the front of the room

 2-5

Skill Building Levels

2

3

4

5

Teaching Tip

A slide with a timer has been provided for you. Delete this slide if you prefer a different method of keeping track of time for students so that they switch drivers and navigators often.

 6

Practice Levels

 7-8

Assessment Level

7

8

Assessment Opportunity

Formative Assessment: Levels 7 & 8 can be used as a formative assessment.

A rubric is provided in level 7, and written feedback can be given to students. [Click here to learn more about giving feedback to students.](#)

Level 8 is a free response reflection that can be used as an opportunity to gain insight into how the students completed the assessment level and the choices they made with adding and styling pseudo-classes. In addition, this reflection can be used to identify any misconceptions shared among students that should be cleared up.

 9

Challenge Levels

Wrap Up (5 minutes)

Question of the Day: What is a CSS pseudo-class, and how does it change the way an element looks or behaves on a web page?

Do This: Have students update their CSS Charts in their journals to include the pseudo-classes.

Prompt: How might you use pseudo-classes in your group’s project to enhance the aesthetics and functionality of your website?

Discussion Goal: This quick reflection encourages students to think about the practical application of the concepts they've learned. By considering how to incorporate pseudo-classes into their projects, students move from theoretical understanding to practical usage, which is crucial for deep learning and skill retention. It also considers how different pseudo-classes can enhance the aesthetics and functionality of their project, leading to a more engaging and user-friendly end product.



This work is available under a [**Creative Commons License \(CC BY-NC-SA 4.0\)**](#).

If you are interested in licensing Code.org materials for commercial purposes [**contact us**](#).

Lesson 19: Planning a Multi-Page Site

90 minutes

Overview

Students work in teams to plan out their websites and create a sketch of each page. They then download the media that they will need for their sites. At the end of the activity, they decide how the work will be distributed among team members and report whether the entire group agreed to the plan.

Question of the Day: How do we plan a web page as a group?

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

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

Agenda

[Warm Up \(5 minutes\)](#)

[Activity \(80 minutes\)](#)

[Overview](#)

[Design Your Site](#)

[Describe Your Tag and Class Styles](#)

[Work Plan](#)

[Find Your Images](#)

[Uploading images](#)

[Wrap Up \(5 minutes\)](#)

[Journal](#)

Objectives

Students will be able to:

- Distribute tasks amongst team members.
- Seek and incorporate feedback from team members.

Preparation

- Check the "[Teacher's Lounge](#)" forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our [Virtual Lesson Modifications](#)

Links

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

For the teachers

- [Planning a Multi-Page Site](#) - Slides
▼ Make a Copy

For the students

- [Plan a Multi-Page Website](#)
▼ Make a Copy
- [Website for a Purpose](#) - Rubric
▼ Make a Copy
- [Website for a Purpose - Student Checklist](#) ▼ Make a Copy

Teaching Guide

Warm Up (5 minutes)

Prompt: Write down the top three fun features that you might like to add to your site.

Share: Allow students to share out some of their favorite features.

Discussion Goal: The goal of this discussion is to make sure that the various features that students have learned are top of mind when they begin to design their sites. Make sure students are mentioning the features that they have particularly enjoyed in the unit, and review the tags and css properties that create them.

💡 Remarks

Today, you'll have a chance to design your websites in your teams. You'll get to work together to decide how all of these features can go into your sites.

Question of the Day: How do we plan a web page as a group?

Activity (80 minutes)

Group: Place students into project groups.

Distribute: Hand out the project guides, one for each group. Also hand out a rubric or student checklist for each group so they can self-assess and reflect throughout the project.

Overview

Review the project requirements with the students, emphasizing that part of the project is working together as a group.

Design Your Site

Teams should describe each page in the site, including its purpose. On separate sheets of paper, they should sketch out the various pages. This allows different team members to focus on different pages.

💡 Teaching Tip

You may want to ask groups to run their sketches by you before moving on. Make sure that the pages all appear to be the same style, and that they include a way to navigate from one to another.

Describe Your Tag and Class Styles

Students should decide together how they would like all of the elements to be styled. They may want to refer back to the sketches in order to decide together.

Work Plan

Once the design is set, teams should decide how they will split up the work. There are multiple pages and a style sheet to consider. They should also consider how they will incorporate the different components into a single project.

Check to make sure that all of the students are in agreement about the plan.

Find Your Images

Once students have gotten your approval for their site designs and work plan, allow them to go online to find images for the site.

Uploading images

Once students have gotten their images, allow them to upload them into the project on Code Studio. Every student within each team should have an identical set of images with identical names.



Upload Your Images

Wrap Up (5 minutes)

Journal

Question of the Day: How do we plan a web page as a group?

Prompt: Describe a difference of opinion that members of your team had, and how you resolved it.



Assessment Opportunity



Check that student descriptions of the disagreement and resolution involve effective and collaborative strategies for working together.



This work is available under a [Creative Commons License \(CC BY-NC-SA 4.0\)](#).

If you are interested in licensing Code.org materials for commercial purposes [contact us](#).

Lesson 20: Project - Website for a Purpose

90 minutes

Overview

Teams have spent a lot of time throughout the chapter planning their websites. In this lesson, they are finally able to code their pages. Using the project guide, the team works together and individually to code all of the pages, then puts all of the work together into a single site.

Question of the Day: What skills and practices will help us work together to make a great website?

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- AP - Algorithms & Programming

Agenda

Warm Up (5 minutes)

Team Cheer

Activity (80 minutes)

Web Lab

Wrap Up (5 minutes)

Shout-out!

Objectives

Students will be able to:

- Create a digital artifact
- Distribute tasks among team members and maintain a project timeline

Preparation

- Check the "[Teacher's Lounge](#)" forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our [Virtual Lesson Modifications](#)

Links

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

For the teachers

- [Project - Website for a Purpose](#) - Slides ▾ Make a Copy

For the students

- [Web Development Challenge Levels Extra Code](#) - Resource ▾ Make a Copy
- [Website for a Purpose](#) - Rubric ▾ Make a Copy
- [Website for a Purpose - Student Checklist](#) ▾ Make a Copy

Teaching Guide

Warm Up (5 minutes)

Team Cheer

Microphone Remarks

Your teams have been working hard for these past few lessons, so we're going to take a bit of time to celebrate. You have five minutes to think of a great cheer for your team.

Share: Give students some time to come up with a cheer for themselves, then allow them to share with the class if they'd like.

Question Teaching Tip

Depending on your classroom, you may want to alter the activity. Any activity that gets students in a good mood to persist through coding out this project will work well.

Activity (80 minutes)

Web Lab

Microphone Remarks

Now that we're ready to get going, it is time to begin implementing the plans that you've made.

Distribute: Return the "Prepare" project guide to groups.

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

Question Teaching Tip

New Code and Previous Challenge Levels: Throughout the unit, students may have learned additional code in the challenge levels of different lessons. As students approach this project, they may want to revisit the code they learned in these levels or visit them for the first time to learn new codes to use in this project. To help guide students back to previous levels, you can use the [Extra Code in Challenge Levels](#) as a resource to quickly find where new codes were introduced in earlier challenge levels.



Your Team Project



Add Content and HTML

Question Teaching Tip

Debugging Strategies: As students design and implement their own project ideas, they may find themselves with new bugs that they need to untangle and you may find yourself looking at completely unfamiliar code as students look for help troubleshooting their errors. To help smooth out the debugging experience, consider the following strategies:

- Review the **Teacher Guide to Debugging** for some common questions and strategies to help support students in debugging their code
- Have students follow the steps in the **Student Guide to Debugging** and use the **Bug Report Quarter-Sheets** as an initial step in the debugging process. This helps students prepare and communicate their issue before asking for help.
- If students haven't seen it yet, consider showing the **Debugging Video** to the class to reinforce debugging best practices.

Digging Deeper: Consider supplying students with an object to talk to as part of the debugging process. This is sometimes known as Rubber Duck Debugging - you can learn more on the website <https://rubberduckdebugging.com/>

 3

Share Your Pages

 Teaching Tip



How Students Will Share Their HTML: Due to the functionality of Web Lab, in order to share webpages to create a completed project, students will need to do some copy and pasting of teammates' code into their own project.

This will require students to complete the following steps:

1. Click the "Share" button on their page, and give the link to teammates in a shared document, by email or however you have students collaborate and share with one another.
2. Get the "Share" link of **each** teammate's webpage
3. On each teammate's webpage click the "View Code" link at the bottom of the screen.
4. Copy the HTML from each teammates' page into a new webpage in their project.
5. Copy and paste the correct navigation bar onto each page.
6. Check that all of the links and images work correctly.

Students will repeat steps 3-4 to get teammate's CSS code

 4

Add Style

 5

Share Your Stylesheets

 6

Check Your Work

 Teaching Tip



Rubric and Checklist: Students have two resources they can use for self-reflection and making sure they are on the right track: the rubric and the student checklist. We recommend having students use the checklist for their own self-assessment and reflection, since it may be easier to digest and understand when reviewing their own project. However, we recommend teachers use the full rubric for evaluating projects to give more accurate feedback to students. You can see examples of this with the Sample Marked Rubrics resource at the top of the lesson plan (only visible to verified teachers)

✓ Assessment Opportunity ▲

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

Wrap Up (5 minutes)

Shout-out!

Prompt: Your team just worked really hard! Give one shout-out to each person on the team about something that they did especially well.



This work is available under a [Creative Commons License \(CC BY-NC-SA 4.0\)](#).

If you are interested in licensing Code.org materials for commercial purposes [contact us](#).

Lesson 21: Peer Review and Final Touches

90 minutes

Overview

This lesson focuses on the value of peer feedback. Students first reflect on what they are proud of, and what they would like feedback on. Teams then work with peers to get that feedback through a structured process that includes the project rubric criteria. Afterward, students decide how they would like to respond to the feedback and put the finishing touches on their sites. After a final review of the rubric, they reflect on their process. To cap off the unit, they will share their projects and also an overview of the process they took to get to that final design.

Question of the Day: How can we use feedback to make our websites better?*

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

► AP - Algorithms & Programming

Agenda

Warm Up (5 minutes)

Journal 3-2-1

Feedback and Iteration (40 minutes)

Reflect: Peer Review

Prepare and Try: Final Touches

Reflect: Final Reflection

Teacher End-Of-Unit Survey

Showcase (45 minutes)

Showcase Set Up

Student Website Showcase

After the Lesson

Post-Project Test

End Of Course Survey

Objectives

Students will be able to:

- Give and receive feedback
- Prioritize and implement incremental improvements

Preparation

- Print a copy of the peer review guide for each student.
- Check the "[Teacher's Lounge](#)" forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our [Virtual Lesson Modifications](#)

Links

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

For the teachers

- [Peer Review and Final Touches](#) - Slides [▼ Make a Copy](#)

For the students

- [Computer Science Practices](#) - Activity Guide [▼ Make a Copy](#)
- [Peer Review - Website for a Purpose](#) - Activity Guide [▼ Make a Copy](#)
- [Website for a Purpose](#) - Rubric [▼ Make a Copy](#)
- [Website for a Purpose - Reflect](#) - Activity Guide [▼ Make a Copy](#)
- [Website for a Purpose - Student Checklist](#) [▼ Make a Copy](#)

Teaching Guide

Warm Up (5 minutes)

Journal 3-2-1

Prompt:

- What **3** parts of your website are you most proud of?
- What are **2** things you learned while working on this website?
- What **1** thing would you like an outside opinion on?

Share: Allow students to share their responses with the class.

Discussion Goal: This should be a fun time for students to reflect on the great work that they have done, but also recognize that they can always use an outside perspective. This will also prepare them to fill out the creator portion of the peer review guide.

Question of the Day: How can we use feedback to make our websites better?

Microphone icon 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.

Feedback and Iteration (40 minutes)

Reflect: Peer Review

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

Group: Pair teams up with each other. Alternatively, you could pair individuals from different teams together.

Peer Review Process

Teams will:

- Open up their website projects in Web Lab.
- Fill in the top part of the worksheet, identifying what they would like feedback on. (They should have thought about this at the end of the last lesson)
- Trade places with the other team so each is now looking at the other's sheet and website.
- Give feedback on the other team's work.
- Switch back to their sheet and website to review feedback.
- Make a plan for implementing some of the feedback.

Prepare and Try: Final Touches

Transition: Teams 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, you may want to give them some suggestions.

Teams should also review the rubric and student checklist as a final way to check their work.



Incorporate Feedback



Rubric and Checklist: Students have two resources they can use for self-reflection and making sure they are on the right track: the rubric and the student checklist. We recommend having students use the checklist for their own self-assessment and reflection, since it may be easier to digest and understand when reviewing their own project. However, we recommend teachers use the full rubric for evaluating projects to give more accurate feedback to students.



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

Reflect: Final Reflection

Distribute: Hand out one copy of the project reflection **to each student**.

In the reflection, students will reflect on both the process and the product. They should identify aspects of the page itself that they are proud of, as well as how the group worked together. They also describe what they have learned in the course of the project.

Send students to Code Studio to complete a short end-of-unit survey. Although their answers are anonymous, the aggregated data will be available to you once at least five students have completed the survey.



End-Of-Unit Survey

Teacher End-Of-Unit Survey

We also have a teacher end-of-unit survey to learn more about how the unit went for you and your students. While students take their survey, **please complete this end of unit survey for teachers** as well. Your feedback is valued and appreciated!

Showcase (45 minutes)

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.

After the Lesson

Post-Project Test

Post-Project tests are included at the end of every unit. These include several multiple choice and matching questions as well as open ended reflections on the final project of the unit. These tests are aligned to the **learning framework** of each unit and are designed to assess parts of the framework that may not have been covered by the project rubrics. To holistically assess the learning objectives of the unit, the post-project test should be paired with the end-of-unit project which is the primary student assessment in each unit.

Unlocking The Tests: 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. [**Click here for more information about unlocking and administering assessments**](#)

End Of Course Survey

If this is the last unit of CS Discoveries that you are teaching, also have students take the end-of-course survey. See the [**CSD Instructions resource**](#) for more information about the End-of-Course survey and how to assign and see the results.



This work is available under a [**Creative Commons License \(CC BY-NC-SA 4.0\)**](#).

If you are interested in licensing Code.org materials for commercial purposes [**contact us**](#).