

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.



5

Review Your Page

 **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**.