

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** - 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** - 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.



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**.