

girls who
code

Girls Who Code At Home

Share Your Skillz
Informational Website with HTML + CSS

Activity Overview

During uncertain times, it's important to take a break to focus on and share something that brings you joy! Learn how to use HTML and CSS to build a tutorial that teaches someone how to do something like how to make a TikTok, how to draw a cat, how to make your favorite cookies, how to land a perfect water bottle flip - no skill is too small! Before you draft your story and start writing code we recommend checking out the featured Woman in Tech Spotlight, Janelle Hinds. Janelle founded Helping Hands, a company that uses an app to match volunteers with opportunities in their community.

Materials

- [Glitch](#) or the text editor of your choice
- [Example Tutorial Project](#) and [Extension Example](#)
- Planning Guide
- Pen, Pencil, or Markers

Women in Tech Spotlight: Janelle Hinds



Janelle Hinds founded Helping Hands, a company that uses an app to match volunteers with opportunities in their community. The Helping Hands app focuses on matching high school students with civic-oriented volunteer opportunities. These projects help students gain skills they need to succeed in the workplace, such as how to improve their communication skills, while also helping their communities! Through Helping Hands, she hopes to increase youth civic engagement so that students like you can create change in your own communities!

Before starting Helping Hands, Janelle studied Electrical and Biomedical Engineering. She started DeltaHacks, her university's annual student-run "hackathon for change," before graduating college! She has managed projects, designed apps, coded, and founded or co-founded several companies. Throughout her career, Janelle has kept an eye out for Diversity, Equity, and Inclusion in STEM. Helping Hands helps track volunteer hours so that it's easy for all students to meet their volunteering requirements to graduate! Helping Hands also has a grant model for offering free workshops for underserved communities.

Watch [this video](#) to learn about what Janelle advocates for around youth engagement and skills training.

Reflect

Being a computer scientist is more than just being great at coding. Take some time to reflect on how Janelle and her work relates to the strengths that great computer scientists focus on building - bravery, resilience, creativity, and purpose.



RESILIENCE

Helping Hands aims to increase youth active engagement and skills training within communities.

How might workshops around job skills training help build resilience as students look towards starting their careers?

Share your responses with a family member or friend. Encourage others to read more about Janelle to join in the discussion!

Step 1: Brainstorm your tutorial (10 min)

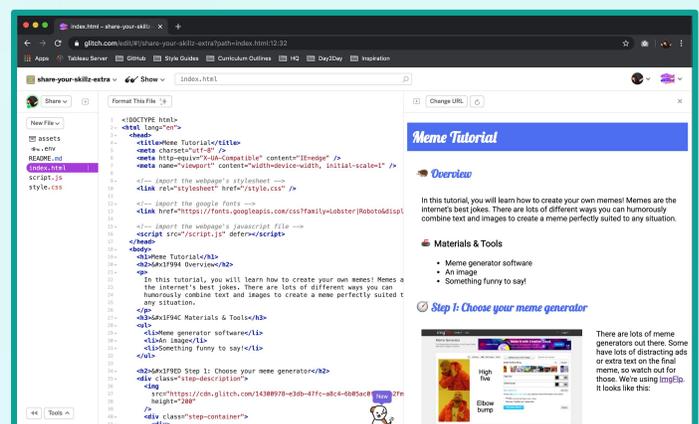
Teaching someone how to do something is a super fun challenge. You need to think about what materials and tools they need, what they need to know to get started, and how to deliver step-by-step information in easily digestible chunks. Before you get started building your tutorial, you need to decide what you want to share. It can be a skill like sewing or a full project like making pizza dough. If you need some inspiration, head on over to [Instructables](#) to see thousands of tutorials on everything from leatherworking to anime drawing. Once you make your decision, keep reading.

Now that you've made your decision, time to plan. We've created a [Planning Guide](#) if you want a place to record your ideas. Let's get started!

- **Overview.** Describe what they will learn in your tutorial. What is the goal? What will this person learn to make or do? Why is it a fun or helpful skill to have? Where might they use it in their lives?
- **Materials and Tools.** What materials and tools do they need to have on hand?
- **Outline.** What are the steps? Draft a list of the steps to learn the skill or complete the project. Try to break the steps down into manageable parts.
- **Steps.** Once you feel good about your outline, it's time to write up each step. Steps should have a heading and short description. You can also include images, links, videos you think would be helpful.
- **Resources.** There is only so much you can teach on one website! Are there any resources that you can point people to if they want to learn more?

Step 2: Create a new project in Glitch (10 mins)

Glitch is a simple tool for creating web apps. It comes with a text editor that allows you to see edits to your webpage in real time. It allows you to publish your project easily for the world to see! If you find a cool project someone else made, you can look at their code and remix it.



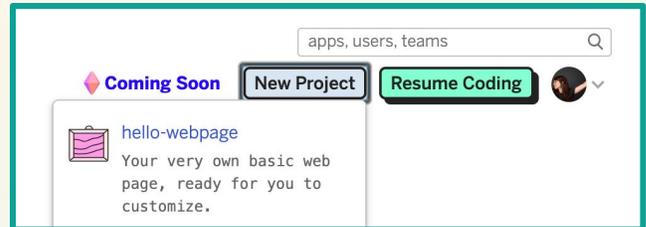
Step 2: Continued

1. Sign in using your Google, Facebook, or GitHub account.

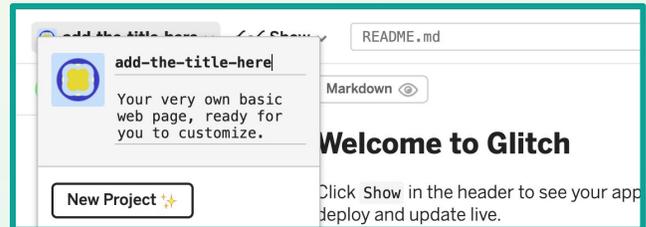
To save and share your work on Glitch, you will need to sign in. Glitch allows you to sign in using a Google, Facebook, or GitHub account. If you are under 13, ask your guardian if you can use one of their accounts to login.

2. Create a new project.

Now that you're signed in, let's get started! Select the *Hello-Webpage* template under the **Start a New Project** menu on the Glitch homepage.

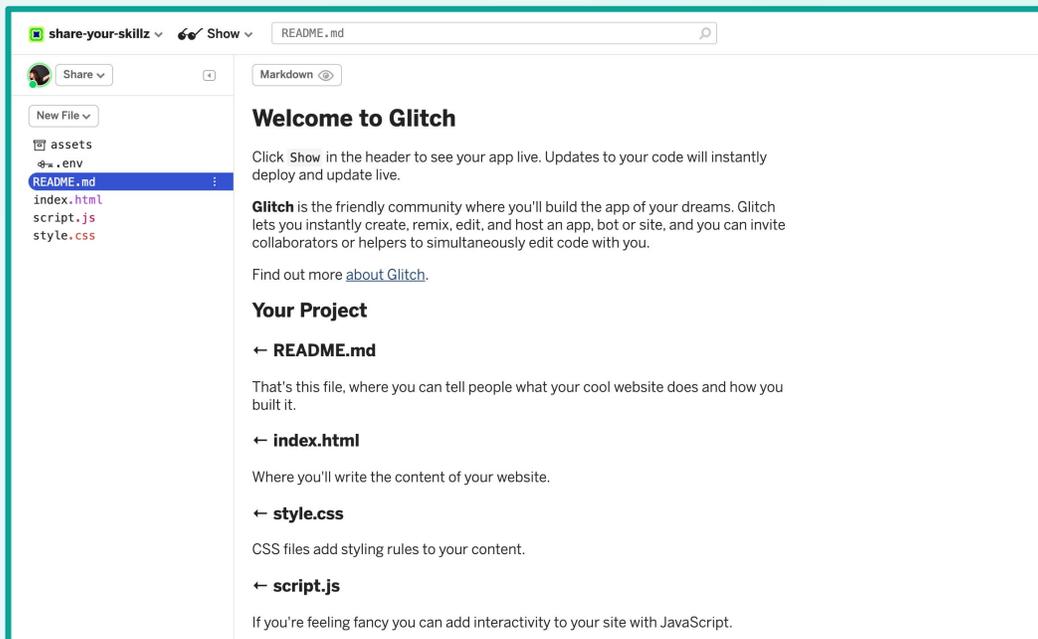


Rename the project space and description. Be sure to give it a descriptive title and clear description. This will be helpful after you make lots of projects and want to look back at them later.



3. Explore the files that Glitch has provided to you!

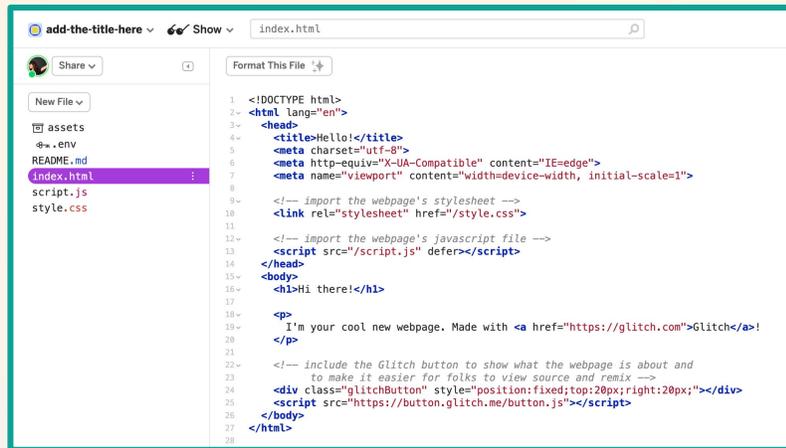
The first thing you will see are the contents of the **README.md** file. READMEs are documents that you will constantly see in other people's projects. They usually contain information about how to navigate through the project's files and how to run the program. This README specifically points you to three different files.



Step 3: Intro to HTML (25 mins)

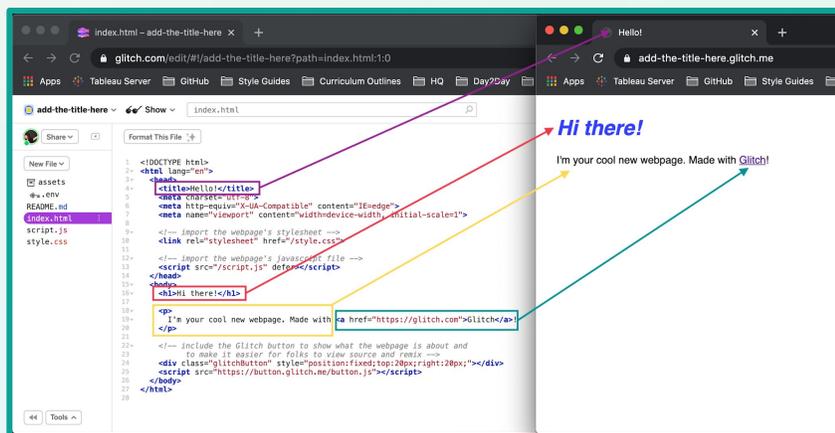
1. Explore HTML

Let's take a look at the file that holds our HTML first. Click on the **index.html** file.



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <title>Hello!</title>
5 <meta charset="utf-8">
6 <meta http-equiv="X-UA-Compatible" content="IE=edge">
7 <meta name="viewport" content="width=device-width, initial-scale=1">
8
9 <!-- import the webpage's stylesheet -->
10 <link rel="stylesheet" href="/style.css">
11
12 <!-- import the webpage's javascript file -->
13 <script src="/script.js" defer</script>
14 </head>
15 <body>
16 <h1>Hi there!</h1>
17
18 <p>
19 I'm your cool new webpage. Made with <a href="https://glitch.com">Glitch</a>!
20 </p>
21
22 <!-- include the Glitch button to show what the webpage is about and
23 to make it easier for folks to view source and remix -->
24 <div class="glitchButton" style="position:fixed;top:20px;right:20px;"></div>
25 <script src="https://button.glitch.me/button.js"></script>
26 </body>
27 </html>
```

Okay so there's a lot of content there. It's okay to not understand it all at once. Click the 'Show' button at the top of the page and see how the project looks on the web. Look for similarities between what is written in your **index.html** file and what appears on the live version of the page!



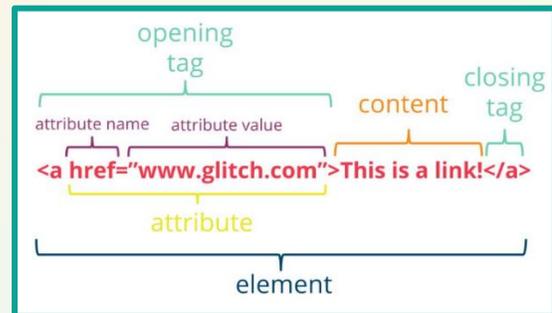
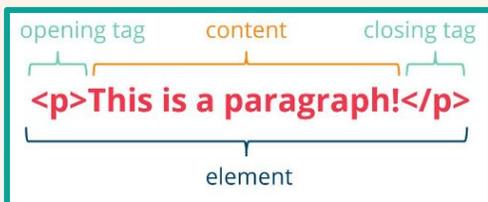
What happens if you begin altering or adding to what's written in **index.html**? Try the following:

- Change line 12 from `<h1>Hi there!</h1>` to `<h1>Hello, Girls Who Code!></h1>`?
- What happens if you add a new line between lines 19 and 20 and type "Created by me! "? Even though the code is written on a new line in your **index.html** file, it doesn't show up in a new line in the live preview. Can you guess why?

Step 3: Continued

2. HTML Tags

All of the content in your **index.html** file is contained in HTML tags. HTML stands for **Hyper Text Markup Language** and is used to add and organize content for a website. Tags are keywords that HTML uses to categorize and organize content. Content might include text, images, video, forms, buttons, or even interactive components like iFrames. All of the content that shows in your live preview are contained within a `<body>` opening tag and `</body>` closing tag.



Tags are part of the anatomy of an HTML element, which are usually made up of the opening tag, content, and a closing tag.

3. Practice using tags

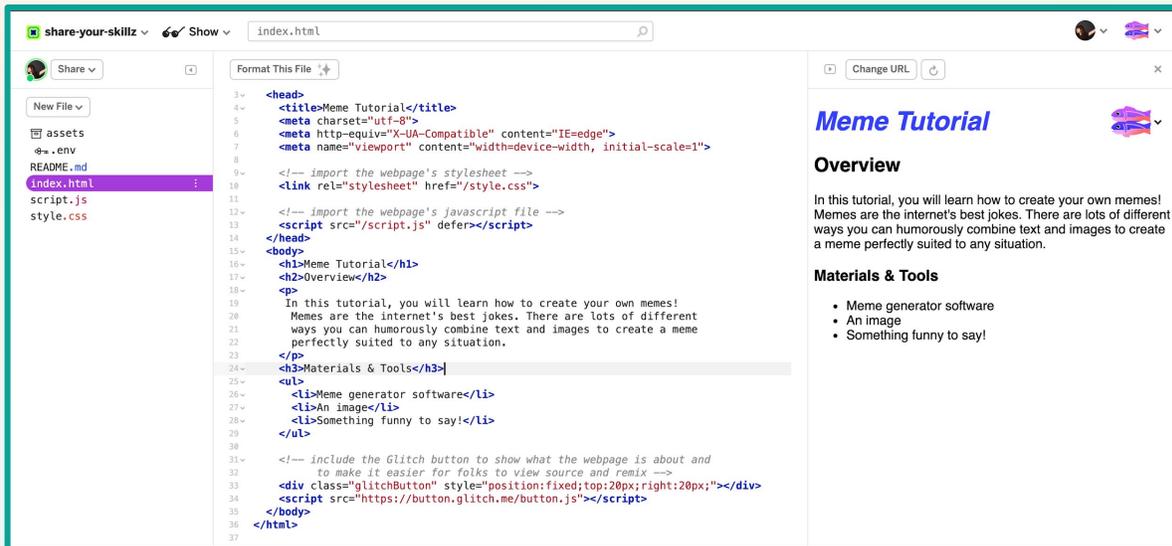
Let's practice using some new tags as we build the overview and first step of our tutorial. We'll walk through them in the next steps, but here's a handy reference:

Tag	Use	Syntax	Example
<code><h1></code> <code><h2></code> <code><h5></code>	Use these tags to create headings that distinguish between information sections.	<code><h1>Overview</h1></code> <code><h3>Materials</h3></code>	Overview Materials
<code></code> <code></code> <code></code>	Create an unordered (bulleted) list with <code></code> and an ordered (numbered) list with <code></code> . Use <code></code> for each item.	<code></code> <code>Item 1</code> <code></code>	<ul style="list-style-type: none"> Item 1
<code><p></code>	Use the <code><p></code> tag for paragraphs of text.	<code><p>Puppy kitty ipsum dolor sit good dog stick canary.</p></code>	Puppy kitty ipsum dolor sit good dog stick canary.
<code><a></code>	Use the <code><a></code> tag for hyperlinks.	This is a <code></code> link <code>!</code>	This is a link!
<code></code>	Use this tag to include images.	<code></code>	

Step 3: Continued

The **Overview** lets people know what they can expect to build, how long it will take, and any materials or tools they need to make it.

- Add `<h2>Overview</h2>` under your name.
- Add a description of what someone will make using your tutorial. Which tag will you use?
- Add a header for your **Materials and Tools** list. Try using a different header element (e.g. h3, h4, h5, or h6)
- Add your materials to the list. To create a bulleted list, we use the `` or unordered list tag with `` or list elements nested inside.



Create your **First Step**.

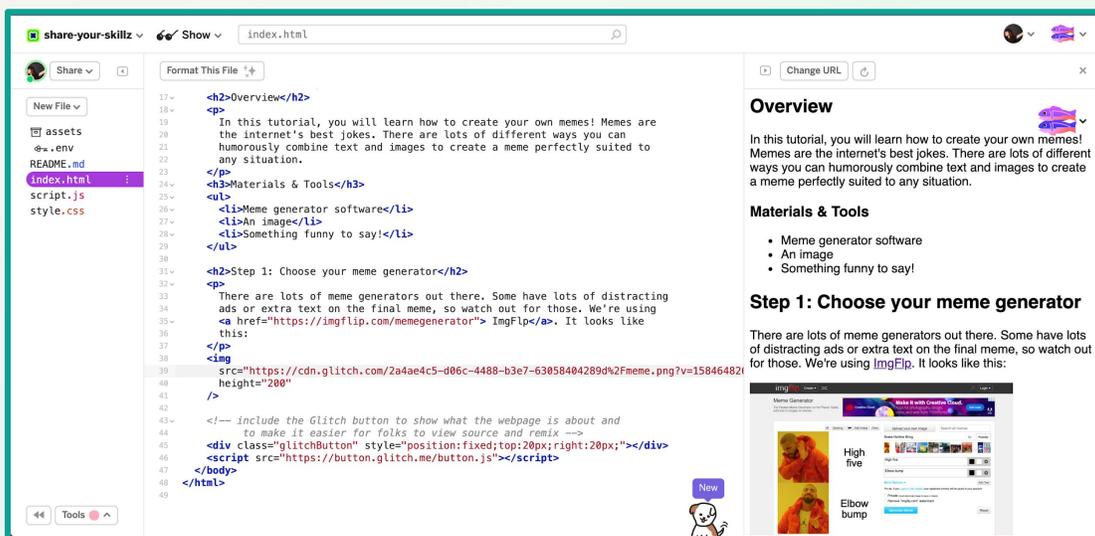
- Add a header for the first step using the `<h2>` tag.
- Add the description of the steps someone should take using the `<p>` tag.
- Maybe you want to tell someone where they can find more resources for your step. Let's add an `<a>` tag for a link.



Step 3: Continued

Add an **image**.

- At this point you might be thinking: Wow, this looks boring. Well, allow us to introduce you to the image tag!
 - Here is the syntax: `` where image.jpg is the URL or file.
 - You can set the width and height like this: ``
- To add an image in Glitch, click on **assets** above all the files. Drop your image in there, then click on the image and copy the URL. Finally, paste the URL in the attribute value of the image element.
- You can also use images from free images on the web on sites like <https://pixabay.com/>. Just copy the image address and paste it in the attribute value of the image element.



Step 4: Styling with CSS (20 mins)

1. Intro to CSS.

Now that we've got the skeleton of our tutorial, it's time to make it look nice! We can do this with CSS! **CSS**, or **Cascading Style Sheets**, describes what kinds of presentation rules – or styles – should be applied to HTML elements. For example, if you wanted to make the text larger or smaller you would do that using CSS. CSS allows you to change the way content on your website looks, including text color, text size, the fonts used, background colors or images, and so much more.

Step 4: Continued

2. Anatomy of a CSS rule-set.

CSS creates rules for presenting content through rule-sets. Click on the **styles.css** file. There are two rule-sets in the file that look like the code below.



A rule-set is made up of:

- a **selector** that determines where the styles are being applied
- a **declaration block**, contained between `{}`s, that lists all of the individual styles to be applied. These individual style declarations have a property name, which corresponds with a value. Each declaration is separated by a `;` or semicolon.

```
1 /* CSS files add styling rules to your content */
2
3 body {
4   font-family: helvetica, arial, sans-serif;
5   margin: 2 em;
6 }
7
8 h1 {
9   font-style: italic;
10  color: #373fff;
11 }
12 |
```

From the pre-written CSS file, we can see that the content in our `<h1>` tags was appearing as blue and italic in our live site because there is a rule-set written for `<h1>` tags that declares their color property should be the value `#373fff` and the font-style property to be italic. The color is written in HEX, or hexadecimal format. If you want to see how HEX values convert to different colors check out this [tool](#) and try entering the HEX code from the starter project.

3. Practice adding some CSS

- What happens if you change the value of the color property? How could you declare the `<h1>` tag be green, for example?
- How would you turn all the `<h2>` tags teal?
- Experiment with adding a new property and value pair for the `<h2>` tags. How might you change their font-size? Background color? Space around the text?

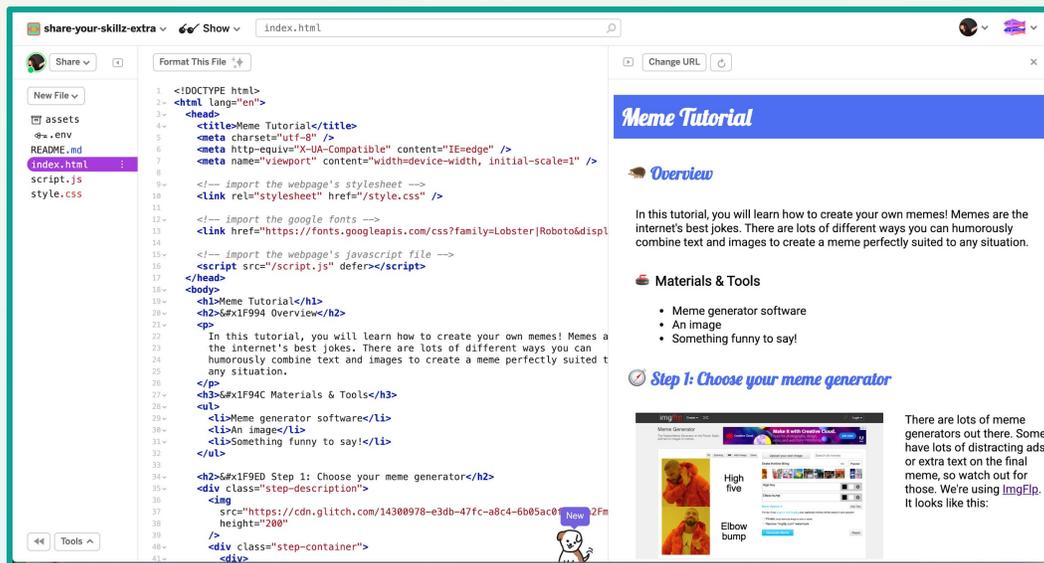
Step 4: Continued

CSS resources

- W3Schools has a great [reference](#) section for CSS.
- Mozilla has a wealth of helpful information in their CSS module, including:
 - [How to style text](#)
 - [CSS Layout](#)
 - [How to use CSS to solve common problems](#)
 - [Debugging your CSS](#)

Step 5: Finish your tutorial (10-20 mins)

Congratulations! Now you have everything you need to finish your tutorial. Add the rest of your steps using the format we created for Step 1. If you want to take it further, try out one or more of the extensions below. Otherwise, move on to Step 6.



Step 6: Take it further (5-10 mins)

- **Make your website more accessible.** Read about accessibility principles outlined by [WAI](#). To begin to make your website more accessible, add alt text for your images, an example is shown below.

```

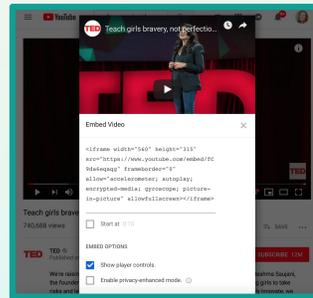
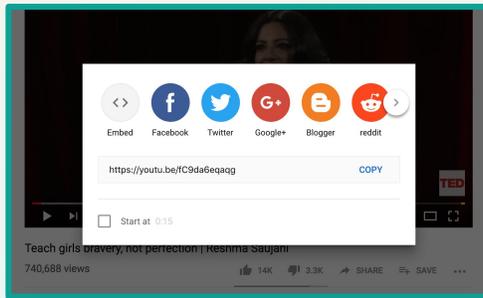
```

Step 6: Continued

Here are some more resources on accessibility:

- Web Accessibility Initiative's [overview of accessibility](#) and [stories of web users](#)
- [Web Accessibility Evaluation Tool](#)
- For more practice with building accessible features, check out [W3C's tutorials](#).

- **Embed YouTube videos into your site.** Embedding YouTube videos is a fun way to make your website interactive.
 - Navigate to your favorite YouTube video
 - Press the "Share" button under the video
 - Select the "Embed" option
 - Copy the `<iframe>` HTML element given by YouTube and paste it into your HTML file. That's it!



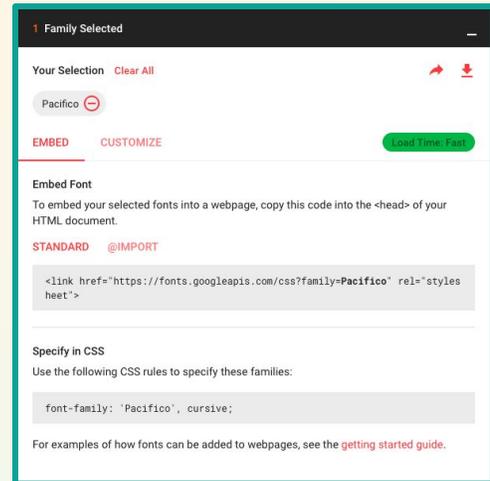
```
<iframe width="560" height="315"
src="https://www.youtube.com/embed/fC9da6eqaag"
frameborder="0"
allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture"
allowfullscreen>
</iframe>
```

- **Add Some Fancy Fonts.** You can easily import and use fancy fonts in your website from [Google Fonts](#).
 - Go to the Google font website and find and select a font you like by clicking the little red plus symbol.



Step 6: Continued

- When you add a font, you can see a menu with the code for inserting your fonts into your website. Put the Embed code for your font between the `<head></head>` tags of your **index.html** page.



- You can now reference this font in your CSS stylesheet. Use the CSS font-family code from Google fonts for the font that you selected during project planning. For example in our website we would use this code to style text:

```
font-family: 'Pacifico', cursive;
```

Step 7: Share Your Creation (5 mins)

Don't forget to share your projects on social media. Tag @girlswhocode and use the hashtag #codefromhome. We might even feature you on our account!

Planning Guide

Use this guide to plan your tutorial and build your website.

- 1. Overview.** Describe what they will learn in your tutorial. What is the goal? What will this person learn to make or do? Why is it a fun or helpful skill to have? Where might they use it in their lives?
- 2. Materials and Tools.** What materials and tools do they need to have on hand? List them below. Include any links if applicable.
- 3. Outline.** What are the steps? Draft a list of the steps to learn the skill or complete the project. Try to break the steps down into manageable parts.

Planning Guide

- Steps.** Once you feel good about your outline, it's time to write up each step. Steps should have a heading and short description. You can also include images (and image address links), links, videos you think would be helpful. For example:

Header

Step 1: Take out all the ingredients

Description

Make sure you have all the ingredients you need to make your pizza dough.

Image

<https://i.pinimg.com/originals/3f/45/3e/3f453eb63aff23a9d6acb51d55628574.jpg>



Links

<https://thepioneerwoman.com/cooking/cpks-bbq-chicken-pizza/>

Planning Guide

- 5. Resources.** There is only so much you can teach on one website! Are there any resources that you can point people to if they want to learn more?