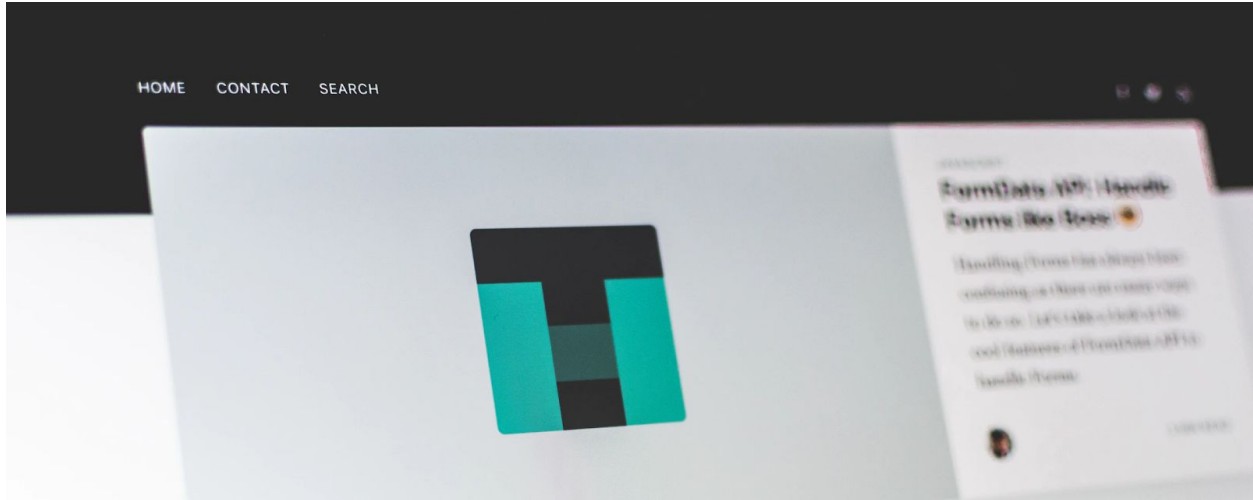


# Web Development Course

## Project 1: Personal Website



**Project:** <https://my-personal-site-example.glitch.me/>

**Code:** <https://glitch.com/edit/#!/my-personal-site-example?path=index.html%3A34%3A9>

**Big Idea:** Students can be creative and express themselves online with HTML and CSS

In this first project, students will get set up and familiar with making websites using the Glitch platform, and make their first website that reflects them and their interests.

This course serves as either an off-ramp from [Vidcode's JavaScript courses](#) into traditional web development, or as a standalone course to teach your students how to make personalized, interactive websites. Your students will be introduced to creating websites with HTML, CSS, and JavaScript.

**Time:** 35-45 minutes

- |                      |                           |
|----------------------|---------------------------|
| <b>5-10 minutes</b>  | Getting set up on Glitch  |
| <b>5 minutes</b>     | Intro to HTML             |
| <b>20-25 minutes</b> | Personal Websites project |
| <b>5 minutes</b>     | Discussion and Sharing    |

## Background

### Introducing HTML

**HTML** stands for Hyper Text Markup Language. It is the standard markup language for web pages, which means that it allows the user to create and structure web pages.

Webpages are broken up into **HTML**, **CSS**, and **JavaScript**.

- **HTML** is the **content** of the site. The text and images on a site are **organized in HTML tags**.
- **CSS styles** sites, and customizes the HTML of the site with fonts, background-colors, placing elements, just to name a few.
- **JavaScript** adds interaction to sites. If you open a menu, or play a game online, JavaScript is probably behind it!

Teacher Reference: [More on HTML](#)

## Personal Site

**Hi! Welcome to Trish's webpage.**

### 3 facts about me:

1. My favorite color is purple
2. I love dogs
3. I love to eat ice cream in the summer

### This is my dog Pluto:



**Example project:** <https://my-personal-site-example.glitch.me/>

**Code:** <https://glitch.com/edit/#!/my-personal-site-example?path=index.html%3A34%3A9>

## Project Checklist for Students

Each project should include:

- At least one header tag
- A paragraph tag
- A list with 3 elements
- At least one picture
- All elements above styled with CSS

## Start Here: Get Set up on Glitch

Students will be moving off of Vidcode in this lesson. On [Glitch](https://glitch.com), they can create real websites that will be hosted on Glitch's platform. They can use Glitch to host portfolios, events, and websites for local businesses.

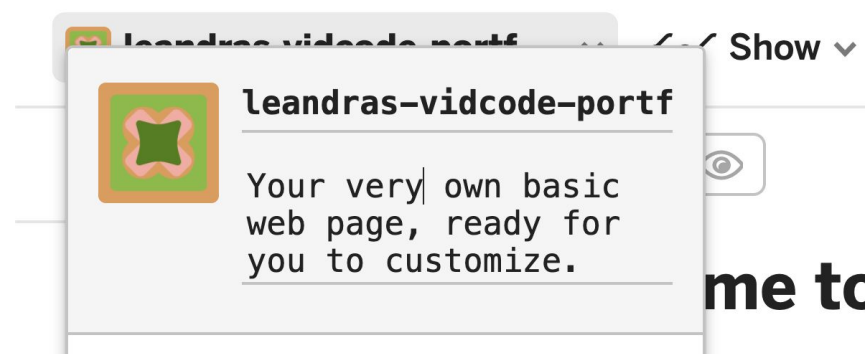
Glitch is not Vidcode! Review their [privacy policy](#), and make the best decision for you and your students.

Students should visit <https://glitch.com/> and create accounts. This will let them save their work.

To start, they should press **"New Project"** and then **"Hello Webpage"** in the top right corner.

This will open their first website! This is what your students will be using in all the projects in this guide. We'll walk through the different pages in the first project.

Students can **rename** their websites by clicking on the top left corner. The name of their project is reflected in the URL.





The **head** contains information that won't appear on the main part of the page. This is where information such as the name of your site lives.

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7
8     <title>Hello!</title>
9
10    <!-- import the webpage's stylesheet -->
11    <link rel="stylesheet" href="/style.css">
12
13    <!-- import the webpage's javascript file -->
14    <script src="/script.js" defer</script>
15  </head>
```

The **body** (lines 16-27 below) includes information that will actually appear on students' webpages.

```
16 <body>
17   <h1>Hi there!</h1>
18
19   <p>
20     I'm your cool new webpage. Made with <a href="https://glitch.com">Glitch</a>
21   </p>
22
23   <!-- include the Glitch button to show what the webpage is about and
24         to make it easier for folks to view source and remix -->
25   <div class="glitchButton" style="position:fixed;top:20px;right:20px;"></div>
26   <script src="https://button.glitch.me/button.js" defer</script>
27 </body>
28 </html>
29
```

**Start by deleting everything between the open and closed body tags** (lines 17-26) - you're going to be starting from scratch with your personal website!

## Elements to Add

The elements below are the **building blocks of every webpage**. Later on in this course, students will make modern-looking websites, but for now, they should just focus on basic elements and styling. Once they recognize these elements, they'll use them on almost every site they make in the future.

### Title

The title of a site will appear in the tab on the top of a web browser, or in the preview when you share the link on certain sites. Update this to "Your Name Site."

```
<title>Trish's First Webpage</title>
```

## Headers

Content has default size and priority on websites. You can add **headers** and **paragraphs**. Headers are reserved for important information like article titles and announcements. Headers are numbered from **h1** to **h6**, in order of importance.

```
<h1>Something really important here</h1>
<h2>Something slightly less important</h2>
```

Try it! Add an **h1 tag** welcoming people to your homepage. Put it directly inside the `<body>` `</body>` tags.

For example:

```
<body>
  <h1>Hi, welcome to Trish's webpage!</h1>
</body>
```

## Paragraphs

Most of the text on a website is generally going to be in paragraph form. Anytime you have big blocks of text, or text you don't think is super important, put it in a paragraph tag.

Paragraphs are also **block** elements, which means they put your content into a new line. If you ever want a new line for your content, you can put it in a paragraph tag.

These look like:

```
<p>Your content here</p>
```

Add a paragraph tag, with the content, **3 facts about me**. You're going to be using a **list element** to list these facts next!

```
<p>3 facts about me:</p>
```

**Preview your website to make sure all your content is loading.**

## Lists

**Lists** are used when you want to create a list of something, like a shopping list or a list of state capitals. These can be either **ordered**, with numbers next to each element, or **unordered**.

### Example of an ordered list:

Code	Output
Colors of the rainbow: <ol> <li>Red</li> <li>Orange</li> <li>Yellow</li> <li>Green</li> </ol>	Colors of the rainbow: 1. Red 2. Orange 3. Yellow 4. Green

### Example of an unordered list:

Code	Output
Things to pack for a picnic: <ul> <li>A basket</li> <li>Cheese</li> <li>Lemonade</li> <li>A blanket</li> </ul>	Things to pack for a picnic: • A basket • Cheese • Lemonade • A blanket

Add 3 facts about you, **inside of the paragraph tag**. It might look like:

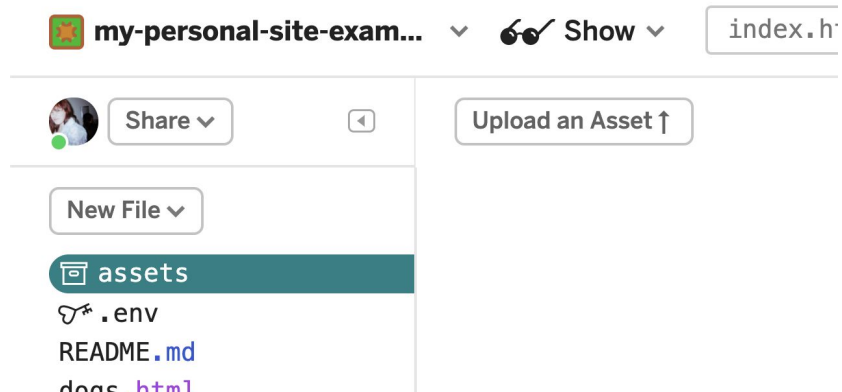
```
<p>  
    3 facts about me:  
    <ol>  
        <li>My favorite color is purple</li>  
        <li>I love dogs</li>  
        <li>My favorite food is pizza</li>  
    </ol>  
</p>
```

## Images

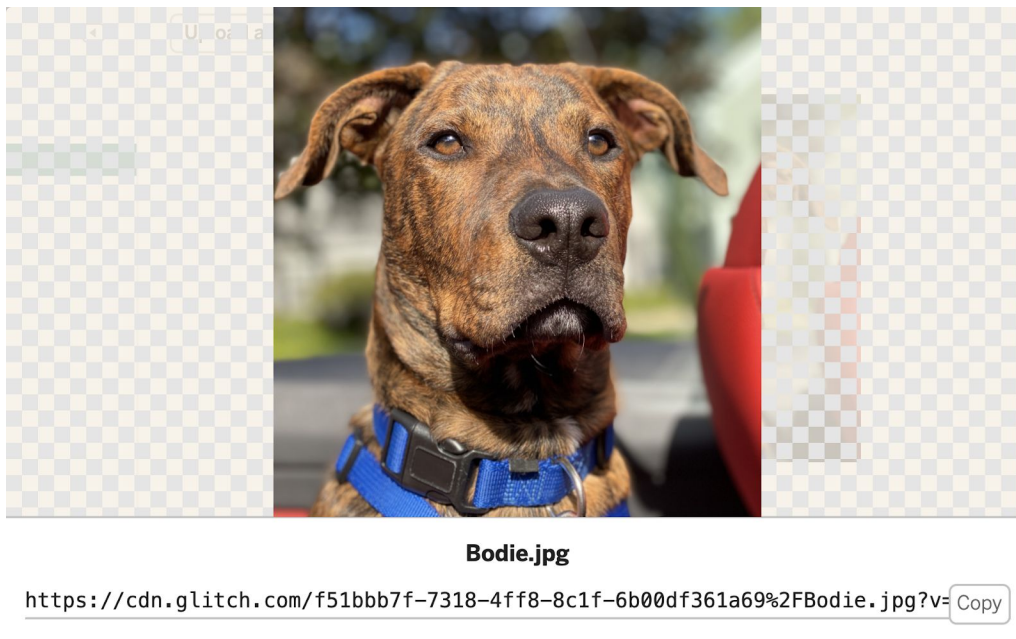
Images can be added to an image with `img` tags, and a **source**.

This is the trickiest element you've added yet!

First, you'll want to **upload** an image to Glitch. Find an image online or on your computer that you want to add to your personal website. One great place to look for pictures is the [Creative Commons Image Search](#).



Click **Assets** on the left and **upload** the image you've chosen from your computer. Once it's uploaded, **copy the image source** by clicking on the image and then pressing "copy" on the bottom right.





From here, go back to your `index.html` file and add an image to your webpage. Images are added with the code:

```

```

Or

```

```

Images **do not** have a closing tag.

**Add images** that support the **facts about you** to your website.

## Links

The last part of your HTML is to add links to some of your favorite websites. These are structured inside `a` tags. These can be included inside paragraph tags, and won't break up your text.

Code:

```
<a href="https://www.vidcode.com/">Visit Vidcode and learn to code!</a>
```

**Add at least one link to your website.**

Here's what your code might look like at this point:

```
17 ~ <body>  
18 ~   <h1>  
19     Hi! Welcome to Trish's webpage.  
20   </h1>  
21 ~   <h2>  
22     3 facts about me:  
23   </h2>  
24 ~   <p>  
25 ~     <ol>  
26 ~       <li>My favorite color is purple</li>  
27 ~       <li>I love dogs </li>  
28 ~       <li>I love to eat ice cream in the summer</li>  
29     </ol>  
30  
31   </p>  
32 ~   <h3>  
33     This is my dog Pluto:  
34   </h3>  
35 ~   <p>  
36     more adorabe dogs</a>  
38   </p>  
39   </body>  
40 </html>  
41
```

## Styling with CSS

The next step is to add style to your website. At first you'll change simple things, like the color of your site background and text, but this will build to letting you have total control over the way your website looks!

To start, click into your style.css file on the left. Delete the content in the file so far, you'll want to start from scratch.

There are a few ways to style your code, but at first you're going to use the types of elements to add style to all of them at once.

CSS styling is added like:

```
element {  
    styleType: details;  
}
```

Pay attention to the small details, the : (**colon**) after the style type and ; (**semicolon**) at the end are both really important, your code won't work otherwise.

Start by adding a color to your **h1** tag.

Add it like:

```
h1 {  
    color: red;  
}
```

But with a color of your choice. You can do this with all your header and paragraph tags. For example:

```
h2 {  
    color: purple;  
}
```

```
p {  
    color: blue;  
}
```

etc.

If you want to change the background color, you can add a style directly to the **body**. For example:

```
body {  
    background-color: cyan;  
}
```

There are a lot of ways to style a website, but we're only going to go over one more here, and it's to change an image size.

You can call an image with the tag name, same as the others:

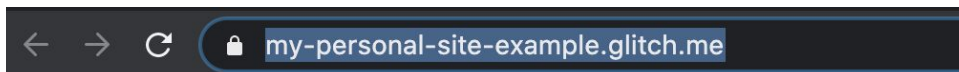
```
img {  
  width: 300px;  
}
```

Px stands for pixels, which are the unit you use for size on websites.

Here's what your CSS code might look like at the end of this project:

```
1~ img {  
2   width: 300px;  
3 }  
4  
5~ h1 {  
6   color: purple;  
7 }  
8  
9~ h2 {  
10  color: pink;  
11 }  
12  
13~ p {  
14  color: blue  
15 }  
16  
17~ h3 {  
18  color: green  
19 }
```

At this point, your student's projects are finished! They can view their projects, and share it with others by copying and sharing the site URL.



# Hi! Welcome to Trish's webpage.

## Discussion Points

- **How do websites work, what are they made up of?**

A website is simply a collection of codes for web pages. The code describes the layout, format, and content on a page.

- **What is HTML?**

HTML stands for Hyper Text Markup Language. It is the standard markup language for web pages, which means that it allows the user to create and structure web pages.

- **What is CSS?**

CSS stands for Cascading Style Sheets. It describes how HTML elements are to be displayed in media and saves a lot of work, by controlling the layout of multiple web pages all at once.

- **What are some places you can find images online?**

Adding images and where to find creative commons media, how to give credit

Creative commons media are media that can be freely distributed, even though they are copyrighted.

You can find creative commons media by doing a simple Google search. There are several different websites where you can find images free for you to use.

Give credit for your image by adding to the readme and linking to it from the site where the image has been found or add the image directly under pictures and link back to the website.

- **Share your website with someone else in your class. How are they similar, how are they different? Can you learn something new from their site that you can use next time?**