Girls Who Code At Home

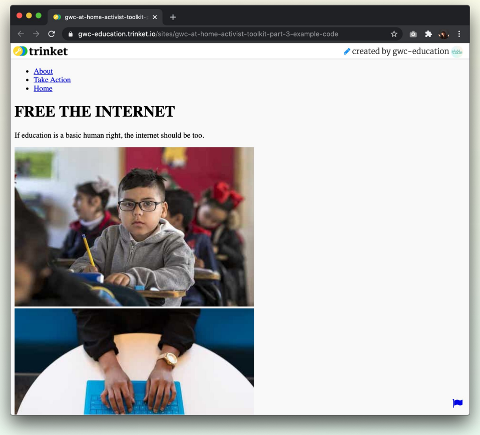
Activist Toolkit Part 3

Building: Intro to HTML

Activity Overview

At the end of Part 2, you created a wireframe and mockup for your website. Now it's time to take those ideas to the screen. In this activity, we will explore the basics of HTML and its role in building a static web page. HTML, which stands for Hypertext Markup Language, helps to organize and categorize content on a website. By content we mean things like text, images, videos, etc.

First you will create your project on Trinket, learn more about HTML tags and elements, then add tags to your wireframe. Next you will use your annotated wireframe to build the HTML for your homepage, Take Action page, and About page.


gwc-at-home-activist-toolkit-p
gwc-education.trinket.io/sites/gwc-at-home-activist-toolkit-part-3-example-code
trinket
created by gwc-education

- About
- Take Action
- Home

FREE THE INTERNET

If education is a basic human right, the internet should be too.

Example Project
It may not look like much now, since HTML just give webpages their structure. We will style your HTML in Part 4.

Materials

➔ Computer
➔ Trinket or the text editor of your choice
➔ Starter Code
➔ Example Project

As a reminder, we will walk you through building a sample project for the remaining activities. You can follow along with our design and customize components like fonts and images or you can create an entirely new design.

PART 1 PLANNING	PART 2 PLANNING	PART 3 BUILDING	PART 4 BUILDING	PART 5 BUILDING
Research & Feedback	**Wireframe & Mockup**	**Intro to HTML**	**Intro to CSS**	**Make it Responsive with CSS**
Identify and research a cause.	Layout the skeleton of your website, then add in your design elements.	Build the framework of your site using HTML.	Style the HTML elements on your site using CSS.	Add styling that adapts your site to different screen sizes.

Women in Tech Spotlight: Miishe Addy

Image Source: Medium

Bartering, the practice of exchanging goods and services for other goods and services without using money or other currency, dates back to 6000 BC. That's about 8,000 years ago!

While bartering may seem ancient, entrepreneur Miishe Addy disrupted the on-demand services market by co-founding and chairing a successful online bartering platform called Skilltapp, Inc. as CEO. The app helps people who do not have a steady flow of cash access the goods and services they need.

"Knowing how to code is like knowing how to read." Miishe Addy

Before working in technology, Miishe worked in law and consulting. However, she sought a role where she could have the most social impact. Technology provided her the best platform for social change. After working as the CEO of Skilltapp for three years, she moved to Ghana and currently advises startups on technology and entrepreneurship.

Watch this video from Blavity to learn more about the inspiration behind Skilltapp and Miishe's journey into the tech world!

Reflect

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

CREATIVITY

Where did Miishe find inspiration for her application Skilltapp? How do you find inspiration for projects?

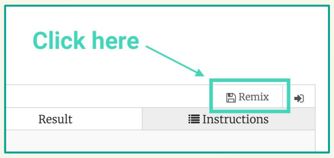
Share your responses with a family member or friend. Encourage others to read more about Miishe to join in the discussion!

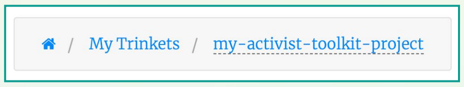
Step 1: Create Your Project in Trinket (5-10 mins)

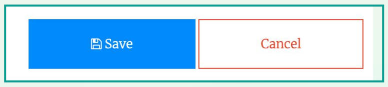
Before we talk about programming, let's talk about the biggest tool we will use in programming: a code editor. There are lots of different code editors you can use to build a website. Some of them are standalone apps that you need to download to use, while others let you program right in the browser window. We are going to use a tool called Trinket for this project. Trinket lets you write and run code in any browser, on any device. You don't need to install anything and it allows you to see edits to your webpage in real time.

Remix the Starter Code

First, create an account in Trinket. Next, go to the Activist Toolkit Starter Code page. You can make your own copy of the starter code by remixing it. Click **Remix**, then rename your project by clicking on the title or URL of the project. Save your project by clicking the blue **Save** button on the right.

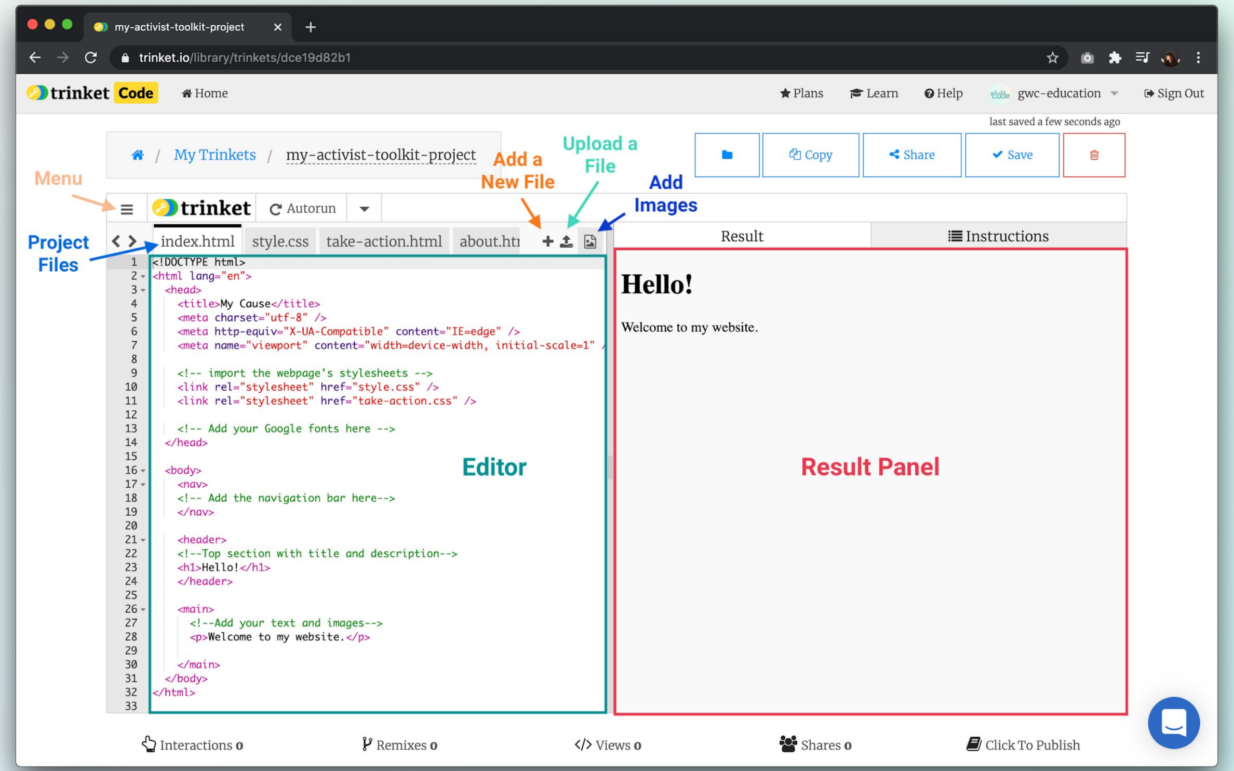






Get to Know the Trinket Environment

Now that you have your very own version, let's examine the Trinket coding environment to see what all of the different spaces, buttons, and tabs do.



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>My Cause</title>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1" /

    <!-- import the webpage's stylesheets -->
    <link rel="stylesheet" href="style.css" />
    <link rel="stylesheet" href="take-action.css" />

    <!-- Add your Google fonts here -->
  </head>

  <body>
    <nav>
    <!-- Add the navigation bar here-->
    </nav>

    <header>
    <!--Top section with title and description-->
    <h1>Hello!</h1>
    </header>

    <main>
      <!--Add your text and images-->
      <p>Welcome to my website.</p>

    </main>
  </body>
</html>
```

Hello!

Welcome to my website.

Review the Files

If you take a look at the starter code, you probably noticed that there are a few tabs above the editor. Websites are really just lots of files stored on a computer remotely that get served up to you when you request them.* Different files do different things. For example, files with the extension **.html** are for HTML and files with the extension **.css** are for our CSS rule sets. This is one way of organizing the content for our website (another way that we will discuss later is how to name things).

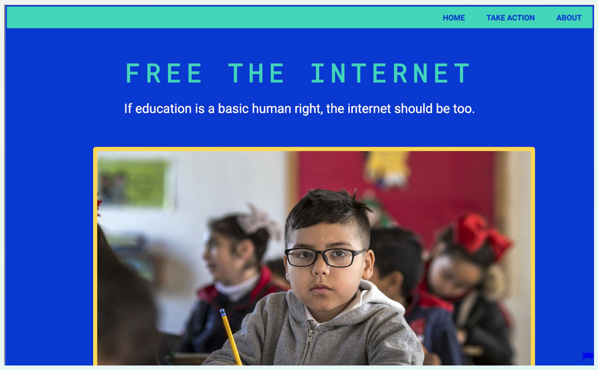
Click on the **Instructions** tab on the right side of the editor, then scroll down to read more about the purpose of each file.

`index.html`	This is your homepage.
`take-action.html`	The checklist lives here.
`about.html`	A page all about you.
`style.css`	Contains the CSS rule sets for all pages.

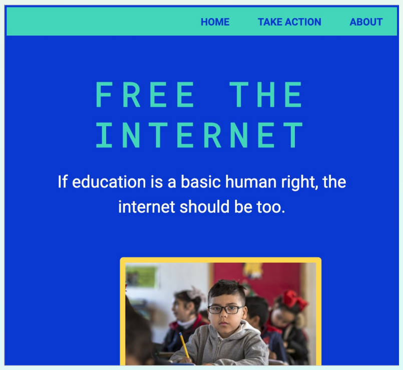
Publish Your Project

Publishing your project does two things: (1) it allows other people to see your website and (2) it allows you to see your full website in a separate tab or window.

Right now, you can only see what your code looks like in the small window of the **Result** panel. This is fine for now, but when we begin styling your website in the next activity, we will want to make design decisions based on how it looks on the full page.


HOME
TAKE ACTION
ABOUT
FREE THE INTERNET
If education is a basic human right, the internet should be too.

Full page view

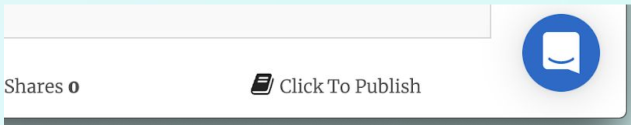

HOME
TAKE ACTION
ABOUT
FREE THE INTERNET
If education is a basic human right, the internet should be too.

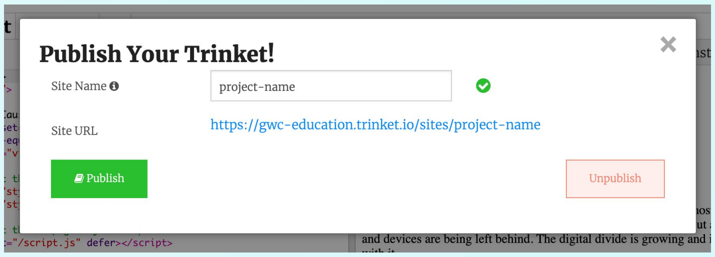
Result panel view

We actually want to make design decisions based on **multiple** screen sizes, but we'll tackle that in Part 5.

Here's how to publish:

- ➔ Find the **Click to Publish** button under the Result panel and click it. You may have to scroll down.
- ➔ Select **Publish** in the next window.
- ➔ Click the **Site URL** to open your project in a new tab.




Publish Your Trinket!

Site Name: project-name

Site URL: https://gwc-education.trinket.io/sites/project-name

** This is a very simple way to think about it. If you are interested in learning more, Mozilla has a great guide to How the Internet Works.*

Step 2: I Spy HTML (15 mins)

We have the start of our project! In this section, we will learn more about HTML by making observations about a webpage and its HTML.

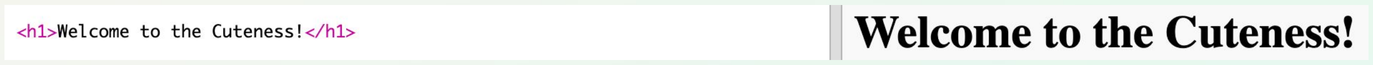
HTML stands for Hyper Text Markup Language.

We use HTML to add information to our content so we can organize a webpage.

Go to this example Trinket project about cute baby animals. Spend a minute or two looking at the HTML and comparing the editor pane to the result pane.

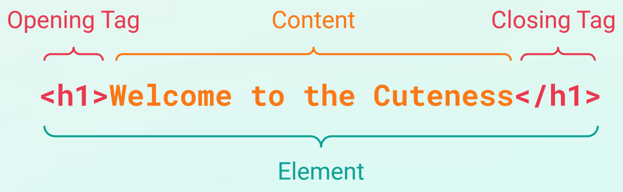
In the next part of the activity, we are going to ask some basic questions and make simple observations to make sense of how HTML works. Read the question first, then try to formulate your own observation before reading the answer.

Question #1: What do you notice about the colors in this chunk of code? Which words show up in the Result panel? Which words don't show up? Why do you think this is?


```
<h1>Welcome to the Cuteness!</h1>
```
Welcome to the Cuteness!

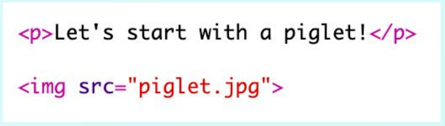
Observation #1: The black text shows up in the Result panel, but the pink words do not. The pink words add information to the black text.

The pink words are called **tags**. HTML uses tags to tell a browser what to do with the content inside it. The tags and content inside the tags are called **elements**.



Elements are the building blocks of HTML. Unlike other programming languages, we don't use variables or conditionals or loops or other core CS fundamentals in HTML. This is because HTML is a markup language. We mark up content with tags so we can organize it into our webpage.

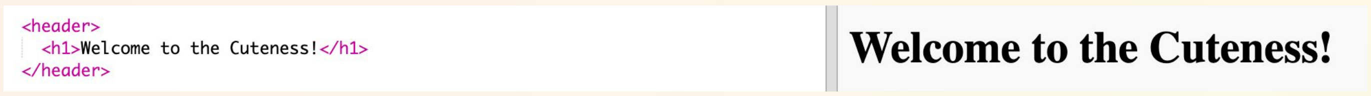
Question #2: What do you notice about the difference between the <p> tag and the <img> tag?


```
<p>Let's start with a piglet!</p>

<img src="piglet.jpg">
```

Observation #2: The <p> tag has a second </p> tag after it, while the <img> tag does not.

Most tags require an opening and closing tag, but some tags don't need them both. For example, the <img> tag is self-closing. It is used to link and display an image on a website. Each HTML tag consistently follows its own rules. You can learn more about different tags in the HTML elements reference from Mozilla Developers Network and the HTML reference from W3Schools.

Question #3: What do you notice about these two tags?



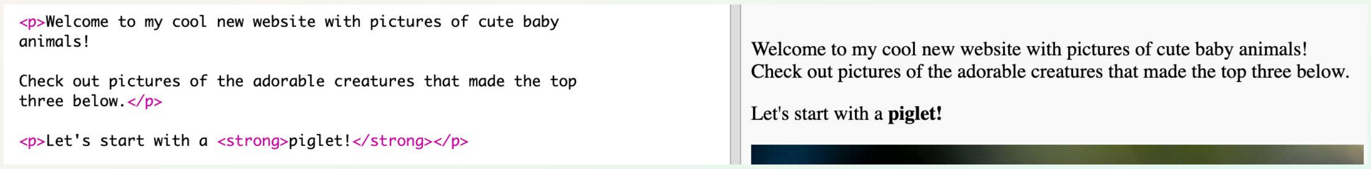
```
<header>
  <h1>Welcome to the Cuteness!</h1>
</header>
```

Welcome to the Cuteness!

Observation #3: The <h1> tag is inside of the <header> tag.

We call putting tags inside of tags **nesting**. Nesting tags inside of each other is a way to group HTML elements together into content sections. This will come in handy later when we want to apply styling to all the HTML elements in a group. *Note: It is important to remember to close your tags - if you get an error while making your project, it might be because you forgot to close your tag.*

Question #4: What do you notice about spacing in these lines of code? Which tag makes a line break and which does not?



```
<p>Welcome to my cool new website with pictures of cute baby
animals!

Check out pictures of the adorable creatures that made the top
three below.</p>

<p>Let's start with a <strong>piglet!</strong></p>
```

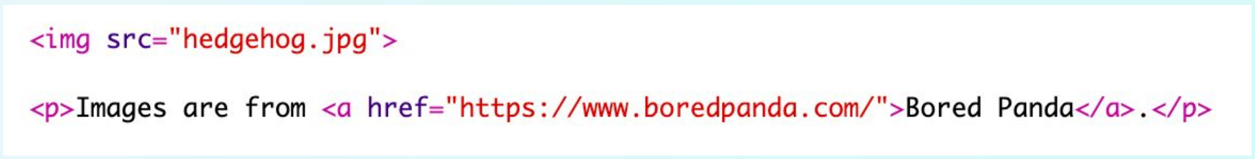
Welcome to my cool new website with pictures of cute baby animals! Check out pictures of the adorable creatures that made the top three below.

Let's start with a **piglet!**

Observation #4: We can only see a line break when we start a new <p> tag, but not when we use the <strong> tag.

Tags like <p> tags are **block elements**. When we use them, they create their own block of space on a webpage by default. They start on a new line and take up the full width available to it. Other block elements include <p>, <h1>...<h6>, <div>, <main>, <body>, <nav>, <ul>, and <ol> tags - but more on those later! Tags like the <strong> tag create **inline elements:** they do not start on a new line and only take up as much width as necessary. Other inline elements include: <a>, <img>, <input>, <em>, <button>, and <span> tags.

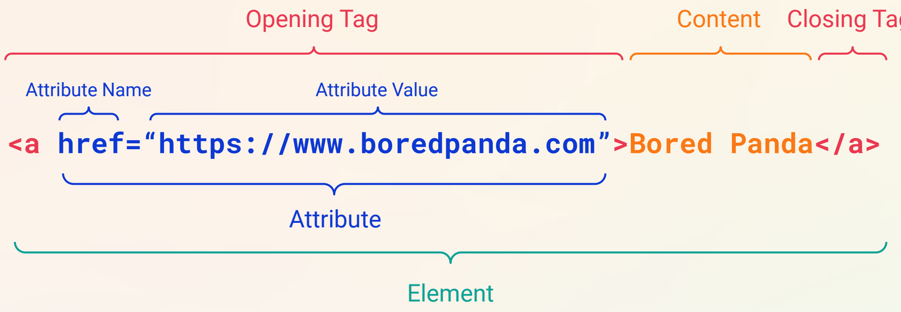
Question #5: What similarities do you notice in the <img> and <a> tags in these lines of code?

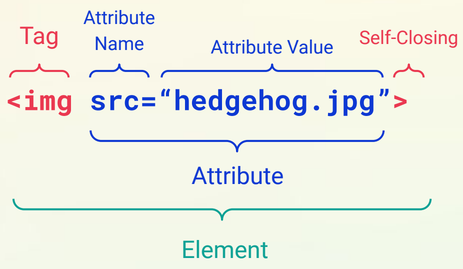


```
<img src="hedgehog.jpg">

<p>Images are from <a href="https://www.boredpanda.com/">Bored Panda</a>.</p>
```

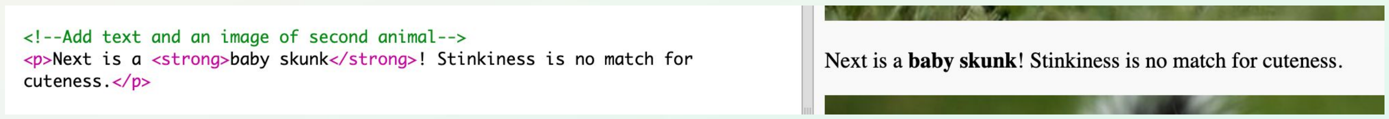
Observation #5: They both have extra information inside the first tag. There is a purple word followed by an = sign, then a URL or file name inside the double quotation marks.

HTML elements can contain additional information, known as an **attribute**. Attributes have a name and a value. This example is known as the anchor tag <a>, which is used to add links to a webpage. The attribute name for an anchor tag is href, which refers to the link reference. The value is the URL inside double quotation marks.





Question #6: What do you notice about the green text in this chunk of code? Does it show up in the right window? Why do you think this is?



```
<!--Add text and an image of second animal-->
<p>Next is a <strong>baby skunk</strong>! Stinkiness is no match for
cuteness.</p>
```

Next is a **baby skunk**! Stinkiness is no match for cuteness.

Observation #6: The green text has special characters surrounding it and it gives me more guidance about the content below it. It does not show up on the right side, which probably has something to do with the special characters.

The green text is called a **comment**. Comments describe what different blocks of code do or give guidance on how to work with a chunk of code. They are invisible on your website and don't affect the formatting of your site.

Comments can help anyone - including yourself! - better understand your thinking and the purpose behind the code. Since programmers often collaborate, it's important to comment your code so someone else can add to it or remix it.

Step 3: Add Tags to Your Wireframe (5-10 mins)

Before you start writing any code, let's take a few minutes to learn about the tags you will be using, then label the components of your wireframe with the HTML tags. This will serve as a guide we can refer to when we start building in the next step. You can find a reference for all of the tags we will be discussing at the end of the activity.

Add the Sectioning Tags to Your Homepage

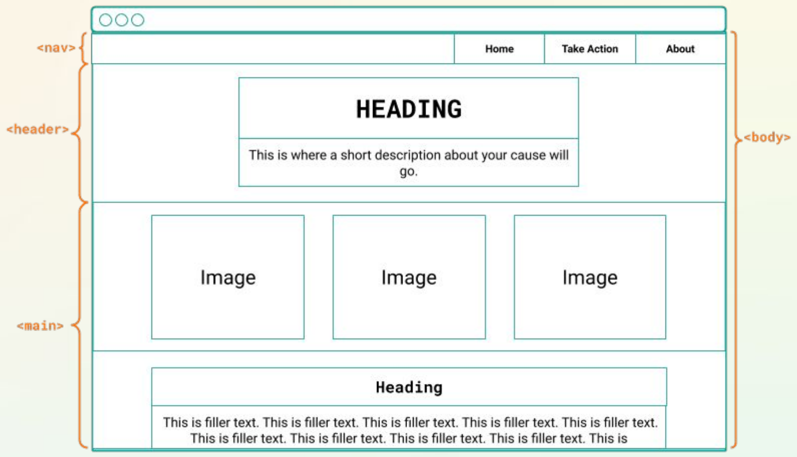
Sectioning tags help us create the larger structure of our website by dividing our content into sections (sort of like a beginning, middle, and end). We can use these tags to group elements together based on where they live on the webpage.

The `<body>` tag holds the content of an HTML document. There can only be one `<body>` tag.

The `<nav>` tag holds the navigation bar in an HTML document.

The `<header>` tag holds the introductory content in an HTML document.

The `<main>` tag holds the dominant content of the `<body>` of an HTML document. There can only be one `<main>` tag on a page. We don't want to put anything in here that repeats on other pages, like the navigation bar.



<nav>
<header>
<main>
<body>
Home
Take Action
About
HEADING
This is where a short description about your cause will go.
Image
Image
Image
Heading
This is filler text. This is filler text. This is filler text. This is filler text. This is filler text. This is filler text. This is filler text. This is filler text. This is filler text. This is

Add Content Tags to Your Homepage

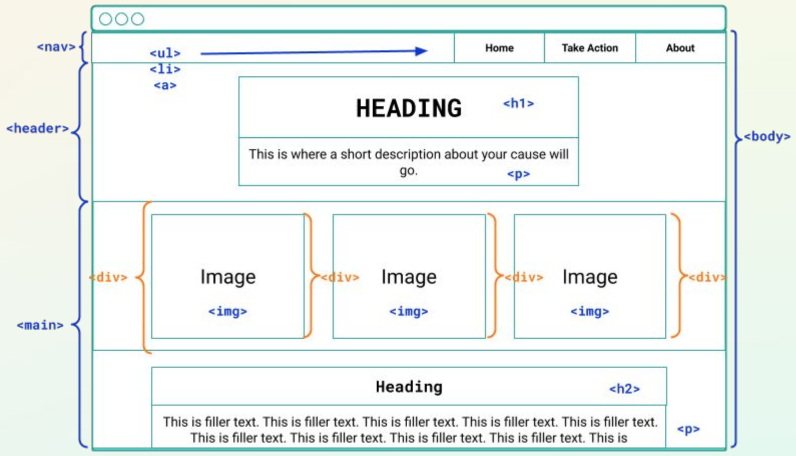
These are the tags we use to organize smaller pieces on our webpage. Read more about them below, then add the tags to your wireframe.

Tag	Use	Syntax	Example
`<h1>` `<h2>` `<h5>`	Use these tags to create headings that distinguish between information sections.	`<h1>Overview</h1>` `<h3>Materials</h3>`	**Overview** Materials
`<ul>` `<ol>` `<li>`	Create an unordered (bulleted) list with `<ul>` and an ordered (numbered) list with `<ol>`. Use `<li>` for each item.	`<ul>` `<li>Item 1</li>` `<li>Item 2</li>` `</ul>`	• Item 1 • Item 2
`<p>`	Use the `<p>` tag for paragraphs of text.	`<p>Puppy kitty ipsum dolor sit good dog stick canary.</p>`	Puppy kitty ipsum dolor sit good dog stick canary.
`<a>`	Use the `<a>` tag for hyperlinks.	`This is a <a href="www.glitch.com"> link </a>!`	This is a link!
`<img>`	Use this tag to include images. Include the alt attribute to add alternative text for accessibility.	`<img src="kitten.png" alt="kitten">`	

Add `<div>` Tags Around the Homepage Images

Oftentimes you'll want to group different HTML elements together so you can adjust the layout of your site or apply similar styling to those elements. We have already talked about some other tags to group elements together like `<header>` and `<main>`. The **`<div>`** tag also lets us group elements together by creating a division or a section in an HTML document. Unlike `<header>` and `<main>` tags, you can use it as many times as you want, you can put it anywhere on the page, and you can put any content inside of it. You can read more about the <div> tag here on the MDN reference and on W3Schools.

```
<div>
    <h1>All about divs</h1>
    <p>You can put any content you
    want grouped together in a
    div.</p>
    <img src="puppy.jpg">
</div
```

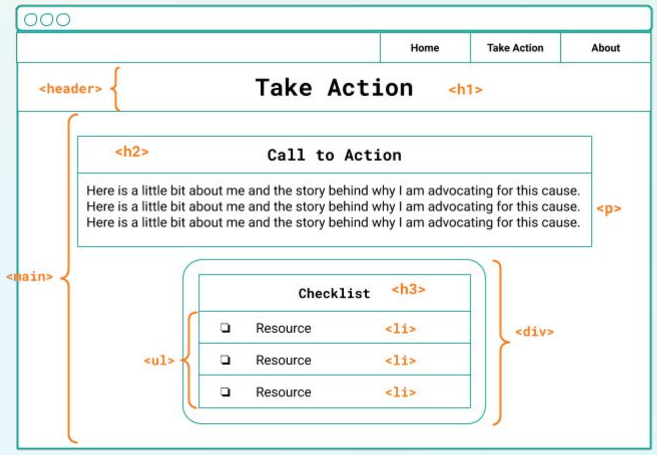


<nav>
<ul>
<li>
<a>
Home
Take Action
About
<header>
HEADING
<h1>
This is where a short description about your cause will go.
<p>
<body>
<div>
Image
<img>
<main>
Heading
<h2>
This is filler text. This is filler text. This is filler text. This is filler text. This is filler text. This is filler text. This is filler text. This is filler text. This is filler text. This is

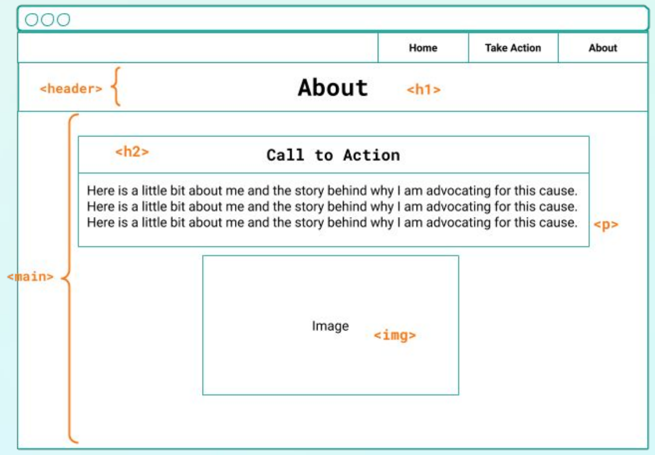
Once we start working in CSS, the true power of `<div>` tags will shine through. For now, we will only use them with our images. Add a `<div>` tag around each image and a `<div>` tag around groups of images that you want to be in a row.

Annotate Your Take Action and About Pages

Don't forget to mark up the **Take Action** and **About** page wireframes as well. Here are some examples:



Home
Take Action
About
<header>
Take Action
<h1>
<h2>
Call to Action
Here is a little bit about me and the story behind why I am advocating for this cause. Here is a little bit about me and the story behind why I am advocating for this cause. Here is a little bit about me and the story behind why I am advocating for this cause.
<p>
<main>
Checklist
<h3>
<ul>
Resource
<li>
Resource
<li>
Resource
<li>
<div>



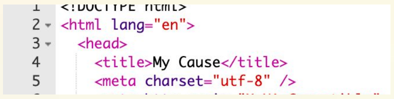
Home
Take Action
About
<header>
About
<h1>
<h2>
Call to Action
Here is a little bit about me and the story behind why I am advocating for this cause. Here is a little bit about me and the story behind why I am advocating for this cause. Here is a little bit about me and the story behind why I am advocating for this cause.
<p>
<main>
Image
<img>

Note: We included a `<div>` tag around the checklist to group those elements together. We will use CSS to style this `<div>` section differently.

Step 4: Build Your Homepage (15 mins)

It’s time to start building your website! You might be thinking, *Wow, finally!* but the more websites you build, the more you will begin to appreciate the planning phases. We promise.

Open the Trinket project you created earlier. For our first task, select your **index.html** file and begin by updating the title on line 4.



```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <title>My Cause</title>
5      <meta charset="utf-8" />
```

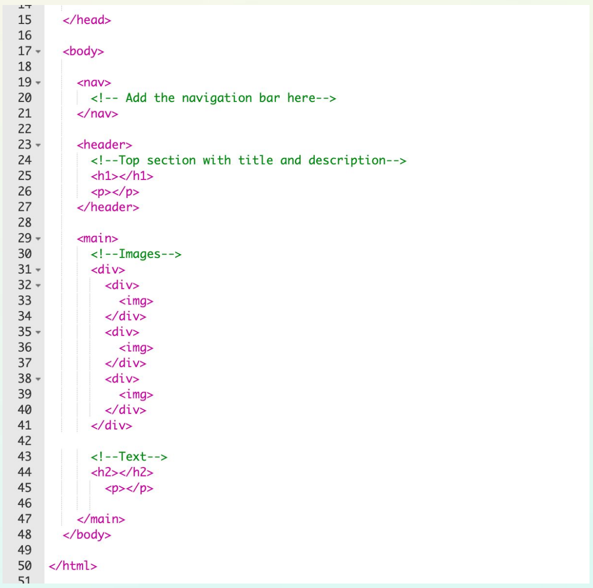
Now, let’s add the rest of our content in the **index.html** file. You will work on your Take Action and About pages later. We will include all the tags from your wireframe first, then fill in the text and images using your mockup. If you prefer, you can add the tags and content at the same time.

Add the Tags

We have already included the sectioning tags (`<body>`, `<nav>`, `<header>`, and `<main>`) and some content tags in the starter code so you can see what the text looks like. Use your wireframe to fill in the content tags in the body.

Don’t add the navigation bar (`<nav>`) just yet - we will cover this in detail in the next step!

➔ **Tip:** If you are unsure how the tag looks, open your fav search engine and type in *How do I use the ____ html tag?*



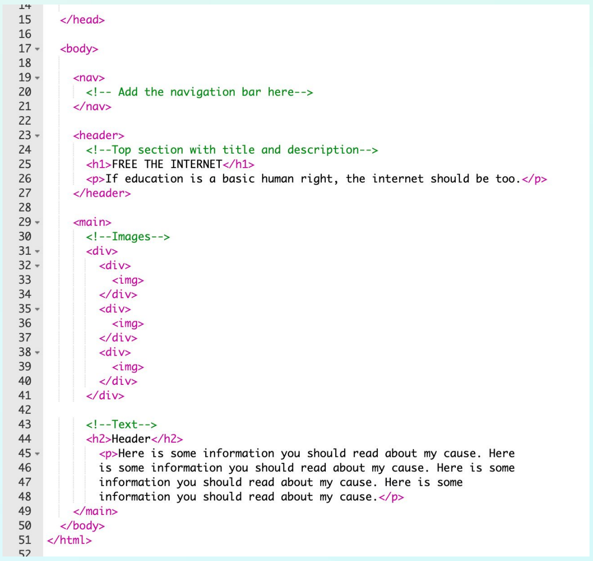
```
  </head>

  <body>

    <nav>
      <!-- Add the navigation bar here-->
    </nav>

    <header>
      <!--Top section with title and description-->
      <h1></h1>
      <p></p>
    </header>

    <main>
      <!--Images-->
      <div>
        <div>
          <img>
        </div>
        <div>
          <img>
        </div>
        <div>
          <img>
        </div>
      </div>

      <!--Text-->
      <h2></h2>
        <p></p>

    </main>
  </body>

</html>
```

Add the Text

Open up your mockup from Part 2. Add in the heading text and paragraph text. Be sure to include any hyperlinks as well.

Test it. Let’s see how it looks:

➔ Save your project.
➔ Open your project in a new tab if it isn’t already. You can do this by clicking **Share** > **Publish** > **Site URL**.
➔ Reload the tab with your full website so it updates with the changes.
➔ Ta daaaa!! You should see your text in the new window. It’s unformatted now, but we’ll fix that soon.



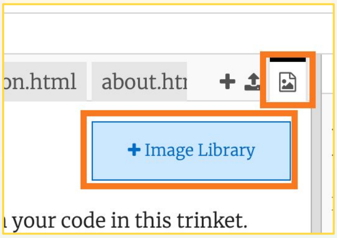
```
  </head>

  <body>

    <nav>
      <!-- Add the navigation bar here-->
    </nav>

    <header>
      <!--Top section with title and description-->
      <h1>FREE THE INTERNET</h1>
      <p>If education is a basic human right, the internet should be too.</p>
    </header>

    <main>
      <!--Images-->
      <div>
        <div>
          <img>
        </div>
        <div>
          <img>
        </div>
        <div>
          <img>
        </div>
      </div>

      <!--Text-->
      <h2>Header</h2>
        <p>Here is some information you should read about my cause. Here
        is some information you should read about my cause. Here is some
        information you should read about my cause. Here is some
        information you should read about my cause.</p>
    </main>
  </body>
</html>
```

Add the Images

Remember that the `<img>` tag links to a source file so the browser knows where to find the image file to display. You can upload them to the project **Image Library** or you can include the URL of an image address on an external site. We prefer the first method because the image is stored with the rest of your page. In the second method, if someone removes the image from their site, it's gone from yours as well!

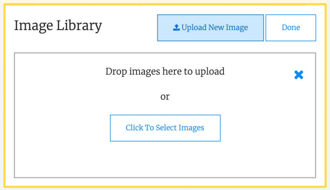
STEP 1

Locate the images you downloaded in Part 3. Click on the **View and Add Image** icon, then click **Image Library**.



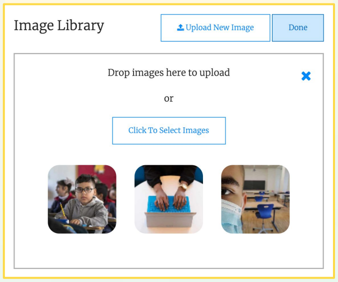
STEP 2

Select **Upload New Image** and add the images you want to include on your website.



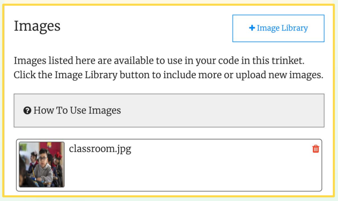
STEP 3

Click **Done** and copy the file name and extension.



STEP 4

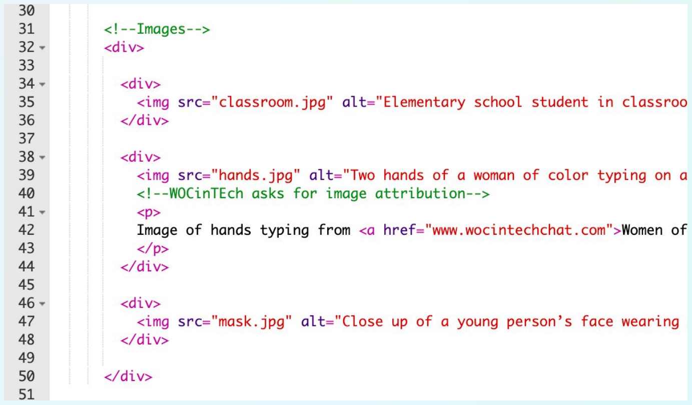
Go back to your **.html** file and paste it into your <img> tag. Don't forget to add your alt text!

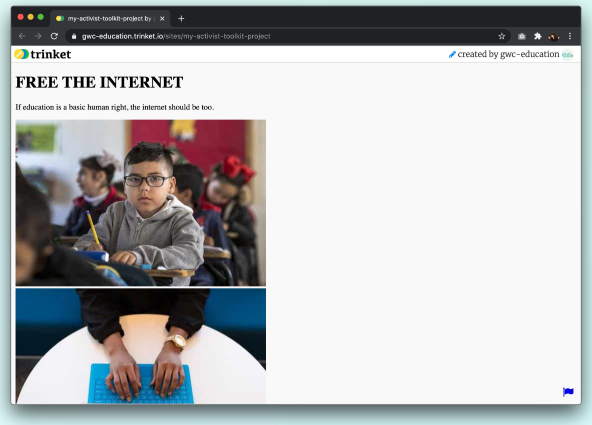


Images listed here are available to use in your code in this trinket. Click the Image Library button to include more or upload new images.

Once you have added in your images, test it to see how it appears:

- Save your project.
- Open your project in a new tab if it isn't already. You can do this by clicking **Share** > **Publish** > **Site URL**.
- Reload the tab with your full website so it updates with the changes.
- Hooray!! You should see your images in the new window.


```

    <!--Images-->
    <div>

      <div>
        <img src="classroom.jpg" alt="Elementary school student in classroo
      </div>

      <div>
        <img src="hands.jpg" alt="Two hands of a woman of color typing on a
        <!--WOCinTEch asks for image attribution-->
        <p>
        Image of hands typing from <a href="www.wocintechchat.com">Women of
        </p>
      </div>

      <div>
        <img src="mask.jpg" alt="Close up of a young person's face wearing
      </div>

    </div>

```

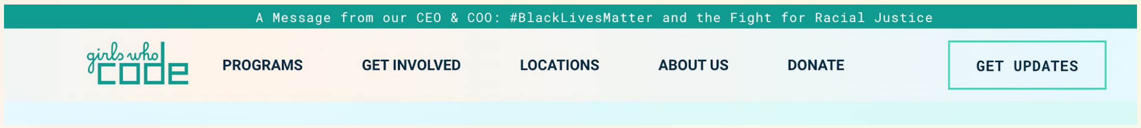

my-activist-toolkit-project by
gwc-education.trinket.io/sites/my-activist-toolkit-project
trinket
created by gwc-education
FREE THE INTERNET

If education is a basic human right, the internet should be too.

Your images might not look the way you expected them to - possibly quite large or quite small and not next to each other. Never fear! We will fix this in the next activity. Just make sure you have your `<div>` tags positioned correctly.

Step 5: Create Your Nav Bar (5-10 mins)

Right now you and people using your site can't move back and forth between different pages of your website. Web developers typically use a navigation or nav bar on websites they build to allow people to easily navigate between different pages. Let's take a look at the nav bar on the GWC website:


A Message from our CEO & COO: #BlackLivesMatter and the Fight for Racial Justice
girls who code
PROGRAMS
GET INVOLVED
LOCATIONS
ABOUT US
DONATE
GET UPDATES

Nav bars are groups of links to the other pages in a website. In the example website above, the nav bar has links called Newsletter, Start Here, Programs, Get Involved, Locations, About Us, Donate, and Get Updates. Without a nav bar, users might get stuck on one web page and be unable to find their desired content.

Let's make the navigation or nav bar structure with HTML. Since a navigation bar is a list of links, you'll need to use the `<ul>`, `<li>`, and `<a>` . We learned a little about these in Step 3, but let's review:

The **`<ul>`** tag is used to create an unordered list, which is a bulleted list (don't worry we will get rid of the bullets when we learn CSS styling). All the pages you want to navigate to will be contained in a single unordered list.

The **`<li>`** tag is used to create a list item. The `<li>` tags should be nested between a pair of opening `<ul>` and closing `</ul>` tags.

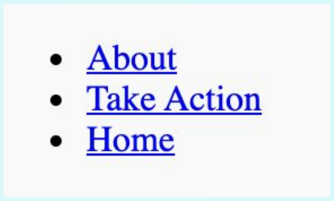
The **`<a>`** tag will be used to create the actual link between the different pages of your website. In your Trinket project all the files are located locally, so the link destination of your home page is just index.html.

```
<nav>
  <ul>
    <li>
      <a href="index.html">Home</a>
    </li>
    <li>
      <a href="about.html">About</a>
    </li>
  </ul>
</nav>
```

Add the Nav Bar

Use the text from your mockup and the `<ul>`, `<li>`, and `<a>` tags to build your nav bar. You already have the `<nav>` tag in the starter code. Once you have added it, test your website to see how it appears:

- Save your project.
- Open your project in a new tab if it isn't already. You can do this by clicking **Share** > **Publish** > **Site URL**.
- Reload the tab with your full website so it updates with the changes.
- Woohoo! You should see a list of your hyperlinked pages!



Step 6: Build the Take Action Page (10 mins)

Now that you have built out your homepage, you can reinforce what you've learned so far by building out your Take Action page. In this step, we will only introduce one new concept - the `<input>` tag.

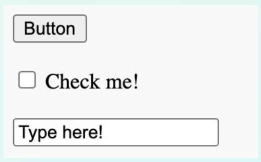
Add the Nav Bar	Add the Tags	Add the Text	Add the Images
Copy and paste the nav bar HTML from the index.html file or practice writing it out.	Use your wireframe to add the tags you need.	Open up your mockup from Part 2. Add in the heading text, paragraph text, unordered list text, and any hyperlinks.	Upload the images you want to include to the Image Library, then copy and paste the file name and extension as the attribute value in the <img> tag.

If you can't remember a full step, go back and look at the process we used to build the homepage. Once you're finished building the Take Action page, test it to see how it appears in the full size window.

Meet the `<input>` Tag

Right now, your checklist doesn't look very much like a checklist - more like a bulleted list of items. Ideally, we would want someone viewing the page to check off an item from the list once they have completed it. There are a few ways we could do this, but the easiest one is to use the **`<input>`** tag.

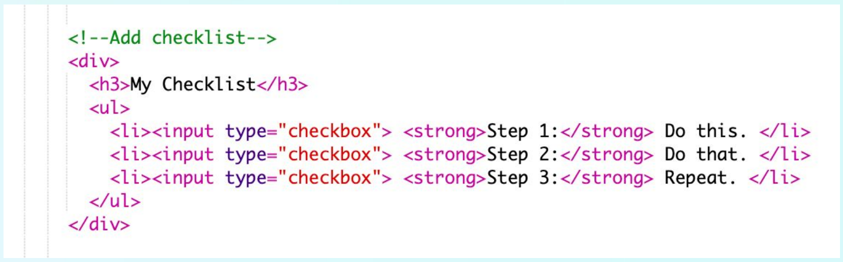
```
<input type="button" value="Button">
<input type="checkbox"> Click me!
<input type="text" value="Type here!">
```



Learn about all the different types of input you can use in this W3Schools reference.

The `<input>` tag specifies an input field where a person can enter data. You can change how to display this field using different type attributes like text or button. For this project, we will use the checkbox type attribute. Add the input tag and checkbox attribute after the each `<li>` element. It is a self-closing tag, so you don't need to worry about closing it.

Check out the HTML for the example project checklist if you need some extra guidance:



```
<!--Add checklist-->
<div>
  <h3>My Checklist</h3>
  <ul>
    <li><input type="checkbox"> <strong>Step 1:</strong> Do this. </li>
    <li><input type="checkbox"> <strong>Step 2:</strong> Do that. </li>
    <li><input type="checkbox"> <strong>Step 3:</strong> Repeat. </li>
  </ul>
</div>
```

Step 7: Build the About Page (5-10 mins)

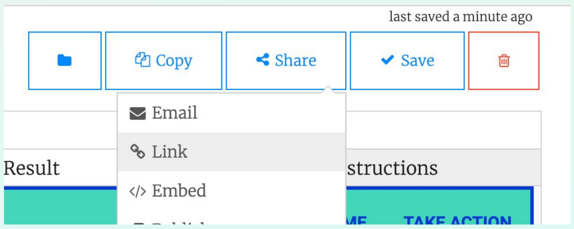
Use what you have learned so far to customize your About page. Follow the process you used to complete your homepage and Take Action page or try mixing up how you build out this file. Instead of adding all the tags then text, try completing one section at a time.

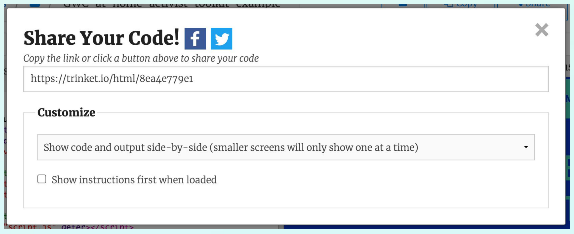
Step 9: Share Your Girls Who Code at Home Project! (5 mins)

We would love to see your ideas! Share your project with us. Don't forget to tag `@girlswhocode` `#codefromhome` and we might even feature you on our account!

Follow these steps to share your project:

- Save your project first.
- Click the **Share** icon
- Choose the **Link** option in the dropdown menu.
- Copy this link and paste it wherever you would like to share it.





Share Your Code!

Copy the link or click a button above to share your code

https://trinket.io/html/8ea4e779e1

Customize

Show code and output side-by-side (smaller screens will only show one at a time)

☐ Show instructions first when loaded

HTML Tag Reference

Activist Toolkit Part 3
Building: Intro to HTML

Below are all the tags we discussed in this activity.

TAG	USE	SYNTAX	EXAMPLE

`<body>`	Holds the content of an HTML document. There can only be one <body> tag.	`<body>`
		`  <nav>`
		`    <ul>`
		`      <li>Home</li>`
		`      <li>About</li>`
		`    </ul>`
		`  </nav>`
		
		`  <header>`
		`    <h1>Hello!</h1>`
		`  </header>`
		
		`  <main>`
		`    <h2>Welcome</h2>`
		`    <p>This is my page about`
		`    cute animals.</p>`
		`  </main>`
		
		`</body>`
			• Home
			• About
			
			Hello!
			
			Welcome
			
			This is my page about cute animals.

`<nav>`	Contains the navigation bar in an HTML document.		
`<header>`	Holds the introductory content in an HTML document.		
`<main>`	Contains the dominant content of the <body> of an HTML document. There can only be one <main> tag on a page. We don't want to put anything in here that repeats on other pages, like the navigation bar.		

`<h1>`
`<h2>`
`<h5>`
	Use these tags to create headings that distinguish between information sections.	`<h1>Overview</h1>`
		
		`<h3>Materials</h3>`
			Overview
			Materials

`<ul>`
`<ol>`
`<li>`
	Create an unordered (bulleted) list with `<ul>` and an ordered (numbered) list with `<ol>`. Use `<li>` for each item.	`<ul>`
		`  <li>Item 1</li>`
		`  <li>Item 2</li>`
		`</ul>`
			• Item 1
			• Item 2

`<p>`	Use the <p> tag for paragraphs of text.	`<p>Puppy kitty ipsum dolor sit good dog stick canary.</p>`	Puppy kitty ipsum dolor sit good dog stick canary.
`<a>`	Use the <a> tag for hyperlinks.	`This is a <a href="www.girlswhocode.com"> link </a>!`	This is a link!
`<img>`	Use this tag to include images. Include the `alt` attribute to add alternative text for accessibility.	`<img src="kitten.png" alt="kitten">`	

`<!-- -->`	This is a comment. It does not show on your site.	`<!--Attribute image-->`
		`<p>Source: WOCinTech</p>`
			Source: WOCinTech

`<input>`	Specifies an input field where a person can enter data.	`<input type="checkbox"> Click me!`	☐ Click me!