



# Girls Who Code At Home

**Website Accessibility**  
Reference Guide

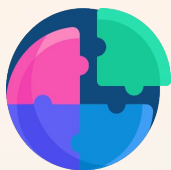
## Website Accessibility - Reference Guide



In this document you will find all of the answers to some of the questions in the activity. Follow along with the activity and when you see this icon, stop and check your ideas here.

### Step 2: Explore a Non-Example

#### Types of Disabilities



There are many different types of disabilities that can be grouped into five different categories: visual, auditory, motor, cognitive, and vestibular. It is important to note many generalizations have been made for different types of disabilities, but individuals may have different experiences that are unique.

Type of Disability	Description
Visual	Visual disabilities may include blindness, low-vision, and even color-blindness. Those who require glasses or some sort of corrective lens also fall under this category.
Auditory	Auditory disabilities affect a person's ability to hear. They may use hearing aids or use a form of sign language to communicate.
Motor	Motor disabilities affect muscle control (i.e movement in the limbs). These are many various types of injuries or conditions that may affect motor control including injury to the limbs, spine, or brain.
Cognitive	Cognitive disabilities may have challenges with memory, attention, or learning differences.
Vestibular	People who have vestibular disabilities are sensitive to fast movement and/or lighting (i.e flickering, flashing, or strobing lights).

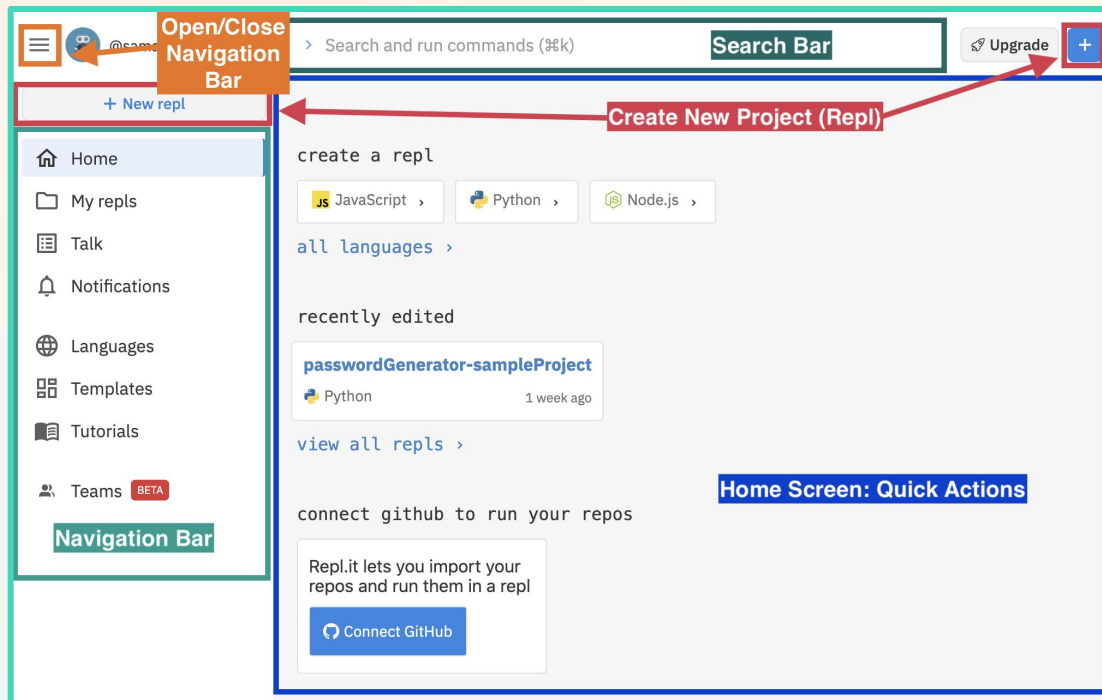
We encourage you to learn more about the different types of disabilities and their unique strengths and challenges. Check out these resources:

- [Australian National University](#)
- [WebAIM: Web accessibility in mind](#)
- [Aruma](#)

## Step 4: Get started with Repl.it

### Explore the Repl.it Platform

Since we already had starter code, we dove right in. But if you want to learn more about using Repl.it, keep reading! Below is your home screen on the Repl.it platform. Let's explore some of the key features available to you started.



- **Navigation Bar:** This column to the left of your screen allows you to access common actions that you might want to engage in. For example you can create a new project (repl), access all of your projects, talk to the Repl.it community, change programming languages, update preferences for your editor under Templates, and learn from some tutorials. In order to open or close this navigation bar you can click the hamburger icon (the three bars icon).
- **Search Bar:** Instead of using the navigation bar, you can also use the search bar to search and run commands.
- **New Repl:** There are two ways to create a new project. The first is through the navigation bar and clicking on the **+New repl** button or clicking the + button on the top right of the window.
- **Main Area:** In the center of your screen will vary depending on your view. When you first log in to Repl.it you will be defaulted to the home screen. Here you will see some quick actions/suggestions for what you might do. As we create a new project, this is where you will do most of your programming.

You can learn more about the Repl.it platform and Repls using this [resource](#).

## Step 4: Get started with Repl.it (cont.)

**Note:** You may notice multiple mentions of GitHub throughout your Repl.it browser. **GitHub** is a popular software development manager. Many programmers use this tool to store all of their projects and code and share amongst a team of developers. We will not go into detail in this tutorial on how to use GitHub. If you want to learn more about this tool check out this [GitHub tutorial](#).

### Create a New Repl.it Project

- ❑ **Create a new repl.** Click the blue **+** button on the top right corner of your screen or the **+New repl** button in your navigation bar.
- ❑ **Select JavaScript under the language option.** For this project we will be working in JavaScript, but remember that for any future projects you can program in over 50 languages on Repl.it.
- ❑ **Name your project.** Give your project a descriptive name like **<yourName>\_acesibleWebsite**. Typically projects should have no spaces and use [camelCase](#) or underscores to separate words.
- ❑ **Click Create Repl.**

## Step 5: I Spy HTML

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

18

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

**Welcome to the Cuteness!**

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

The blue 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.

## Step 5: I Spy HTML (cont.)

**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>
```

**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?

```
17 <header>
18 |   <h1>Welcome to the Cuteness!</h1>
19 </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.

## Step 5: I Spy HTML (cont.)

**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.

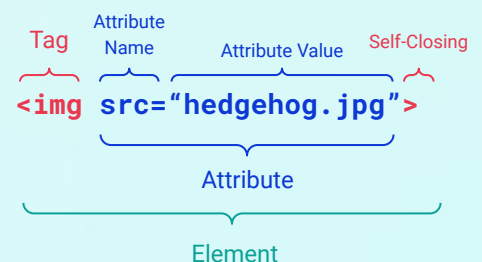
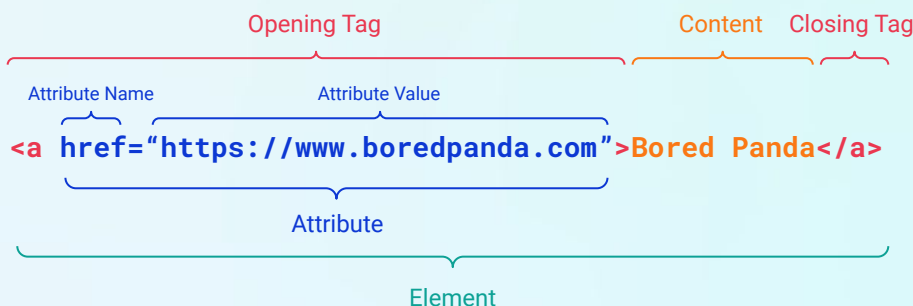
Tags like `<p>` tags are **block elements**. When we use them, they create their own block of space on a webpage by default. They start on a new line and take up the full width available to it. Other block elements include `<p>`, `<h1>`...`<h6>`, `<div>`, `<main>`, `<body>`, `<nav>`, `<ul>`, and `<ol>` 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?

```
  
<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 equals 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.



## Step 5: I Spy HTML (cont.)

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

```
30      <!--Add text and an image of second animal-->
31      <p>Next is a <strong>baby skunk</strong>! Stinkiness is no match for
32      cuteness.</p>
33      
```

**Observation #6:** The gray 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 gray text is called a **code comment**. Code 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 7: Add alt-text to our starter code

Below we only included the code for the three img tags with alt text. Check out our [full sample project](#).

```





```