



Updated: April 14, 2021

How to fix the 5 most common Accessibility errors

by Jay Pope



We recently reanalyzed a sample of up to 100 pages from every college and university in the United States with our [Higher Ed in 4k project](#). Similarly WebAIM analyzed the top million most popular home pages with their [WebAIM Million project](#). Between the projects we have learned lots of interesting information about the state of web accessibility.

My biggest takeaway is that by fixing 5 basic accessibility issues we could fix over 90% of all errors detected. Both projects found the same 5 errors to be the most common; Low color contrast, Missing alternative text, Empty links, Missing form labels and Empty buttons.

While it is important to understand that websites have more accessibility issues than can be automatically detected, these detectable errors are very impactful for many users with disabilities and tend to be related to non-detectable errors. Fixing them would make the internet significantly more accessible. In most cases they are also relatively simple to fix.

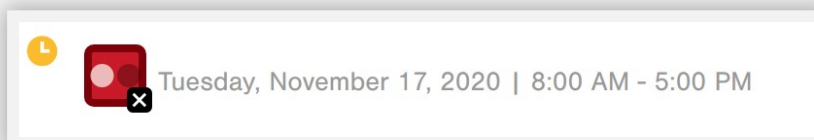
Learning 5 principles and applying them can be an approachable starting point to improve your organization's web accessibility.

I semi-randomly selected an example from a college/university website in the Higher Ed in 4k database to show an actual error for each of the 5 most common errors from a live website and how I would fix it.

Low Color Contrast

In the Higher Ed in 4k project 66% of all errors were low contrast text and in the WebAIM Million these errors were found on 86% of all homepages.

Because not all people see color the same way we should increase color contrast and not rely on color alone to convey meaning. Specifically WCAG AA requires a 4.5:1 contrast ratio for normal sized text. You can use [WAVE](#) or the [WebAIM color contrast tools](#) to test different color combinations.



In this example from a college calendar page we have a really light grey text on a white background. The contrast ratio is only 2.84:1 and can be harder to see for some users or potentially any users in different lighting situations.

This fix is a really easy one, just darken that grey a little bit, you can use WAVE or the Contrast checker to easily find a color with more contrast.

A screenshot of the WebAIM Color Contrast tool interface. It shows two color selection boxes: "Foreground Color" with the hex code #333333 and "Background Color" with the hex code #FFFFFF. Below each box is a "Lightness" slider. The "Contrast Ratio" is displayed as 12.63:1. The "Text Size" is set to "Normal". The "Normal Text" field contains the word "Sample". At the bottom, it indicates "AA: Pass" and "AAA: Pass" in green text.

Don't just try and barely pass but use adequate contrast, when possible aim to pass AAA as well. For a calendar page such as this there really isn't any advantage to using a light

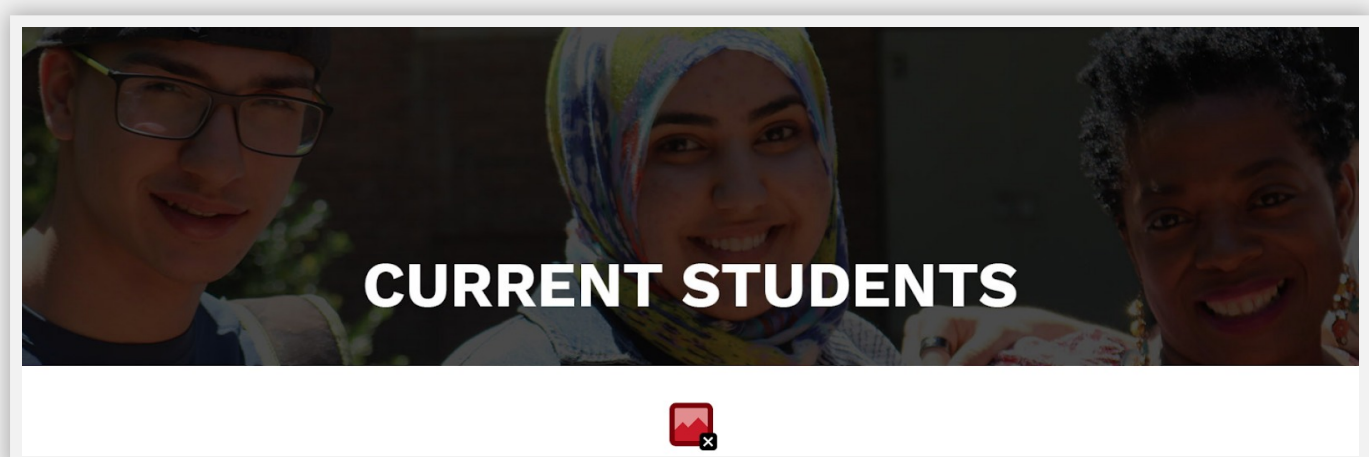
grey and it won't change branding to go darker. Some brand colors can be a bit trickier so consider color contrast when creating designs.

Learn more about Color Contrast by reading the [Contrast and Color Accessibility article by WebAIM](#).

Images missing alternative text

In the Higher Ed in 4k project 8% of all errors were images missing alternative text and in the WebAIM Million these errors were found on 66% of all homepages.

Missing alternative text can be very impactful for screen reader users, sometimes it is as if there is no content there to them without alternative text. In addition if the image without alternative text is a link the function of the link can't be presented to the user.



In this example on a current students page of a University website there are students smiling with the words Current Students burned into the image.

Current html:

```

```

This is problematic because although I can visually see current students it doesn't exist in the content in a way that a screen reader can read it to the user. This example also has non-detectable issues as the image essentially functions as the title and first level heading

<h1> of the page.

Simple solution of adding alternative text:

```

```

Adding alternative text to the image would remove the reported error and now the content added in the alternative text would be accessible to a screen reader user.

The alternative text in this example is a little tricky because this image not only has graphic content (students smiling), it has text content (current students) and it has a function as the page title that is only apparent visually. There are not associated HTML semantics to make this clear to the browser or assistive technologies.

Alternative text that tries to incorporate all 3 of these ideas would be confusing and not what is expected. You could argue that the students smiling are decorative like the example above but you would still not be incorporating the page title function of the image.

Recommended solution:

```
<h1 class="current-student-banner">Current Students</h1>

<!-- and then add the image as a background in CSS -->
```

Now we have semantic html indicating this the page title and we have text content for current students. Other benefits is the text will zoom as the browser zooms (text in on SVG images will get blurry) and the image can be positioned independent of text on responsive layouts. I did assume that the image content of the students smiling is considered decorative but if it was content one possible solution would be to use a

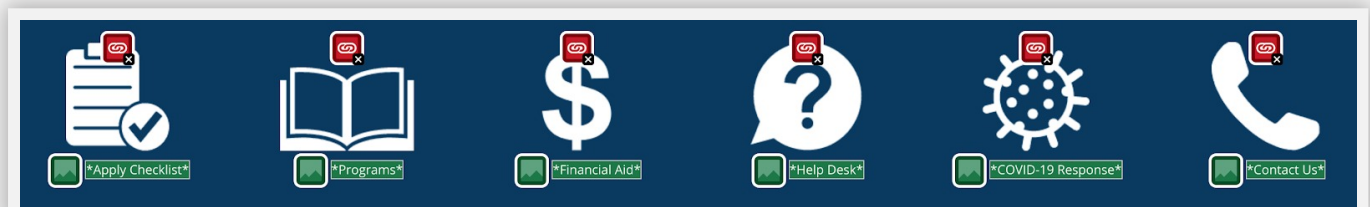
with an alt attribute and position the <h1> on top of it.

Alternative text can be simple and sometimes can take some more thought. For a deeper dive review the [WebAIM Alternative Text article](#).

Empty Link

In the Higher Ed in 4k project 12% of all errors were Empty Link errors and in the WebAIM Million these errors were found on 60% of all homepages.

When a link has no text content, the function or purpose of the link will not be presented to the user. This can introduce confusion for keyboard and screen reader users.



In this example from a College home page we have 6 links next to images. this is part of the core functionality of the page to get information on how to apply or the COVID-19 response for example. Each link is empty though making it really difficult for a screen reader user to know where they go. To make it even more problematic the link url location which is read as a fallback to link text is not human friendly and are 199 characters long.

HTML with issue:

```
<div class="quick-link">
  <a href="/199charactergibberish"></a>
  
</div>
```

I am not sure why they did it this way, but the links don't wrap the image and required additional styling to make the image area clickable with a mouse.

Simplified HTML with issue removed:

```
<a href="/199charactergibberish" class="quick-link">
  
</a>
```

This simplifies the HTML and CSS quite a bit and fixes the issue, now a screen reader will read Apply checklist when focus is set on the link.

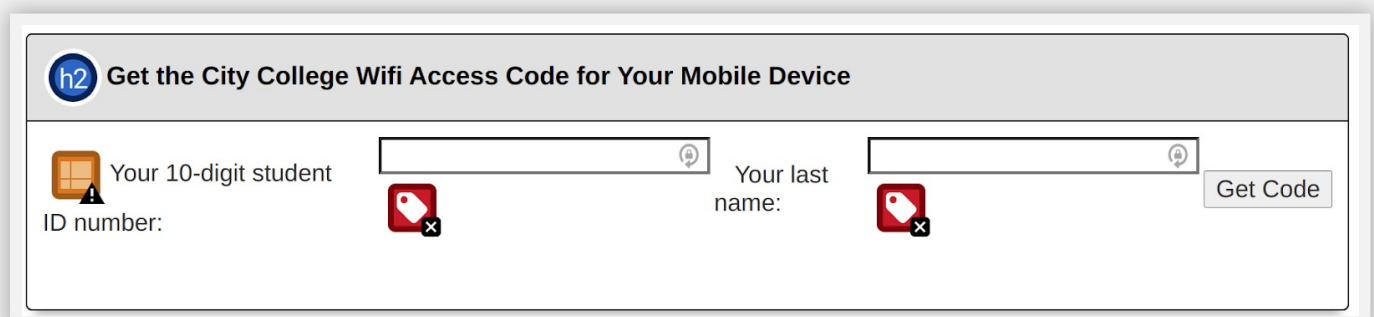
One other consideration in this example is to make sure the icon links function is apparent visually. I would display the text for each icon. From the icons alone it isn't clear that a clipboard is the apply checklist.

Another common culprit of the Empty Link error is font icons. Review this [empty link example on how to fix font icons](#).

Missing form label

In the Higher Ed in 4k project 4% of all errors were images missing alternative text and in the WebAIM Million these errors were found on 54% of all homepages.

Missing form labels can make it hard or sometimes impossible for users using assistive technology to fill out a form. It also loses functionality for mouse users as they can't click on the label to set focus on the input. It is a basic and easy fix.



h2 Get the City College Wifi Access Code for Your Mobile Device

Your 10-digit student ID number: Your last name:

In this example it is a form a on a University library page to get access to the Wifi. Both the student id and first name input fields have no associated labels with them.

HTML with errors:

```
<td>Your 10-digit student ID number:</td>
<td><input name="id" type="text"/></td>

<td>Your last name:</td>
<td><input name="name" type="text"/></td>
```

You will notice there is text next to each input that should be a label and serves as a visual label but it has no semantic meaning and isn't associated with the input next to it.

Updated HTML

```
<label for="student-id">Your 10-digit student ID number:</label>
<input name="id" type="text" id="student-id"/>

<label for="last-name">Your last name:</label>
<input name="name" type="text" id="last-name"/>
```

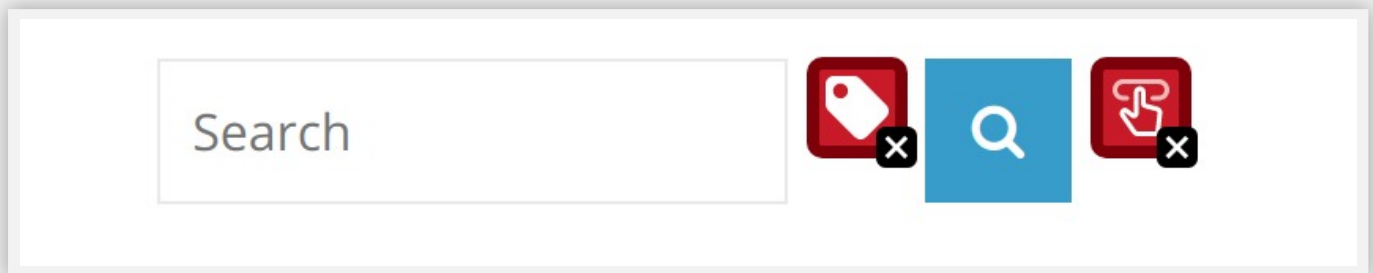
The updated HTML now has a label associated with each input by a for attribute that equals the id of the input. Sometimes people get confused and point it at the input name attribute, this won't work, it needs to be associated with the id attribute. I also converted this to not be a layout table as it was causing issues with responsive layouts on smaller devices.

To learn more about making forms accessible review the [WebAIM Creating Accessible Forms article](#).

Empty Button

In the Higher Ed in 4k project 4% of all errors were images missing alternative text and in the WebAIM Million these errors were found on 29% of all homepages.

Empty buttons are problematic. Lets imagine you have 2 buttons next to each other and one saves the form and the other deletes it. If neither button has any content but only a visual icon then a user who can't see it might not know which does which.



In this example from the header of a University website template there is a search form with a button to submit the search. The button has no text content but only a magnifying icon on the button.

HTML:

```
<input type="text" name="search" placeholder="Search"/>
<button type="submit" name="submit"></button>
```

The button is styled with CSS to add the background image and color. There is no text or content on the button. Furthermore it is next to an input that is missing a label. If the user was using a screen reader/browser combination for example that didn't read placeholders they wouldn't have any idea what the input or button did.

```
<label for="search" class="sr-only">Search</label>
<input type="text" name="search" placeholder="Search"
id="search"/>

<input type="image" src="/searchbutton.jpg" alt="Search">
```

First I added a label, I did hide this one visually as the context of the button icon makes it

clear what it is. You can learn more about [hiding content visually in this WebAIM article](#).

Then I simply converted the button to have a type of image and added alternative text. Nothing changed visually but this is now accessible to users using assistive technology.

Conclusion

By fixing 5 basic accessibility issues we could fix over 90% of all errors detected. We don't need to be accessibility experts to start improving accessibility on our websites. To have an impact today you can start by learning about these 5 errors; Low color contrast, Missing alternative text, Empty links, Missing form labels and Empty buttons.

As you start focusing more on web accessibility consider what an [organizational accessibility strategy](#) would like for your organization.

Email Newsletter

Want to receive emails with accessibility content and tips like this article?

If you subscribe, we will email you web accessibility insights or things we learn a few times a month. You can unsubscribe at any time.

First Name

Email Address

Subscribe

Search Articles



Search

Monthly Accessibility Insights

Sign up to get monthly accessibility content. Unsubscribe at any time.

First Name

Email Address

Subscribe

Free Web Accessibility Scanning

Start improving your web accessibility today with our free plan.

No Credit Card required.

Free Plan



[Sitemap](#)

[Accessibility Statement](#)

[About Us](#)

[WAVE Suite](#)

[Contact Us](#)