Upgrade to Pro
Learn
Catalog

Codecademy Articles > web

Accessibility and HTML

Many visually-impaired users browse the Internet with the user of a screen reader. In this article, you'll learn about various ways to make your content accessible to visually-impaired or blind users.

Accessibility and HTML

Requirements:

- Completed Learn HTML & CSS Unit 2 Lesson 2

1. Introduction
2. Alt text
3. Special tags
4. ARIA

Introduction

When designing and creating web pages, it's important that the web pages are accessible to all audiences. In particular, due to the visual and dynamic nature of the webpages that you'll be making, it's important to make sure that your website will also make sense to visually-impaired or blind users.

Many visually-impaired or blind users access web pages with the help of **screen readers**. Screen readers parse through the HTML of your web page and read the contents out loud, responding to commands to navigate around the page, and take actions such as clicking on a link, typing in an input field, or submitting a form. In this resource, we'll give an overview of a few ways that you can improve the accessibility of your web page and help screen readers better interpret it!

Alt text

One way to make the elements of your page more comprehensible to screen readers is to provide `alt` text for images. Alt, or alternative, text is provided as an attribute to the `<img>` tag to describe the image to the screen reader. For instance, when encountering the below...

```
<!DOCTYPE html>
<html>
  <head>
    <title>My First Web Page!</title>
```

```
  </head>
  <body>
<img src="https://en.wikipedia.org/wiki/File:Brown_bear.jpg" alt="A brown bear"/>
  </body>
</html>
```

...the screen reader will be able to read aloud the description of the image ("A brown bear") to the user, as opposed to simply informing the user that an image exists on the page. Be sure to provide succinct `alt` text descriptions for each image on your web page!

Semantic tags

As we think about styling text on a webpage, you might have noticed some overlap between the following pairs of tags:

`<b>` vs. `<strong>`
`<i>` vs. `<em>`

If you ever try swapping one of the tags in a pair for the other, you'll most likely see that there's little to no change in how the page looks. Both the `<b>` and `<strong>` tags bolden text, and the `<i>` and `<em>` italicize text. However, there's a fundamental difference between each of the tags in a pair when it comes to accessibility!

The `<b>` and `<i>` elements simply denote how the text should *look*: text within these tags should appear as bold or italicized, respectively.

`<strong>` and `<em>`, however, denote how text should be *understood* and, though they result in the same visual appearance, they affect how the screen reader interprets them: text within these tags are read out with a different voice to indicate the emphasis for each. These tags are known as *semantic tags*.

So although `<strong>` and `<em>` result in visually similar text as `<b>` and `<i>`, they provide a way for screen readers to communicate the parts of text that are already highlighted visually to the user.

ARIA

ARIA, also known as WAI-ARIA, stands for `Accessible Rich Internet Applications`. ARIA defines a variety of `markup extensions`, usually HTML5 attributes, that can be added to elements to give screen readers more information about the element and help visually-impaired users better grasp what's happening on the webpage.

For instance, one useful ARIA attribute is the `role` attribute. When added to an element, this will provide the screen reader with extra context about what that element's function is in context of a page. For instance, the following HTML code is easily understood by a human to be a menu:

```
<nav>
 <ul>
   <li>Put navigation here</li>
 </ul>
</nav>
```

To a screen reader, however, it's not so simple. To remedy this, we can add the `role` attribute with value `navigation`:

```
<nav role="navigation">
 <ul>
   <li>Put navigation here</li>
 </ul>
</nav>
```

Now, the screen reader will know that this is a menu and present the options to the user accordingly.

There are many other functions for ARIA attributes, such as telling the user the state of forms and checkboxes, and even calling their attention to an element on the page that was just changed by JavaScript. We won't cover everything related to ARIA in this article, but feel free to check out the spec here.

Ensinando o mundo a programar.

Português (Brazil)