Exercise 2 – Accessibility Audit

# Setup

1. Download the *2-Accessibility-Audit.zip* file from Moodle to your Downloads folder.
2. Open the Finder and go to your Downloads folder.
3. Double-click on *2-Accessibility-Audit.zip* to unzip the file.
4. Double-click the newly unzipped *2-Accessibility-Audit* folder to open it.
5. Double-click the *start-server.command* file.
   1. You should see a terminal window open up saying:

Starting up http-server, serving .

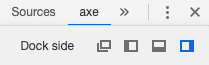
Available on:

http://127.0.0.1:8080

http://192.168.1.6:8080

Hit CTRL-C to stop the server

1. Open *2-Accessibility-Audit.html* in your text editor of choice.
   1. I recommend Visual Studio Code as we will be using it starting next week.
2. [In Google Chrome, install the aXe extension by clicking “Add to Chrome”.](https://chrome.google.com/webstore/detail/axe/lhdoppojpmngadmnindnejefpokejbdd)
3. In a new Chrome tab, navigate to *localhost:8080/2-Accessibility-Audit.html*
4. Right-click anywhere on the page and select *Inspect* to open up the Chrome Dev Tools.
   1. You should see *Elements, Console, …, axe* along the top of the Dev Tools panel.
5. Click on the *axe* tab*.*
   1. To make things a little easier to see, click on the icon of the 3 dots on the top right of the Dev Tools panel, and click on the right-most icon:



1. Click on the button and bask in all of the accessibility flaws!

# Exercise

Your task is to fix all of the accessibility flaws presented by the axe Chrome extension by editing the *2-Accessibility-Audit.html* file. As you click on each error in axe, you will see:

* a description of the issue
* the impact/severity of the issue
* what you need to do to solve the issue

I strongly recommend you click the *Learn More* link located in the issue description to get a more comprehensive explanation and details on how to fix it.

# Example Workflow

1. Pick an issue to fix from the list the axe tool provides.
   1. Click *Learn More* to get a better understanding of the issue.
2. Edit the code in *2-Accessibility-Audit.html* to fix a certain issue.
   1. Feel free to change the CSS as much as you like! The page looks pretty ugly as-is :P
3. Refresh the *localhost:8080:2-Accessibility-Audit.html* page.
4. Re-analyze the page using the axe tool to see if the issue was fixed.
5. Once fixed, add the information to the table below **in your own words (don’t copy-paste from the website)**:

|  |  |  |
| --- | --- | --- |
| **Issue** | **How to Fix** | **Why it Matters** |
| **<li> element most be contained in a <ul> or <ol>** | **Add <ul> or <ol>** | Screen readers notify users when they come to a list |
| **All page contain must be contained by landmark** | **Change <div> to proper landmark** | Screen reader users can navigate to a section based on its HTML element |
| **Most have sufficient color contrast** | **Change color** | Ensure all text elements have sufficient color contrast between the text in the foreground and background color behind it. |
| **Document most have <title> to aid navigation** | **Add <title> in <head>** | Screen reader users use page titles to get an overview of the contents of the page. |
| **Most have a lang attribute** | **Add lang in html**  <html lang="en"> | When configuring a screen reader, users select a default language. |
| **Form elements must have labels** | **Add <label>** | Screen readers users require useful form labels to identify form fields. Adding a label to all form elements eliminates ambiguity and contributes to a more accessible product. |
| **Links must have discernible text** | **Add aria-label** | Ensures that every link has an accessible name. |
| **Page must contain a level-one heading** | **Add <h1>** | Screen reader users can use keyboard shortcuts to navigate directly to the first h1 |
|  |  |  |
|  |  |  |

# Submission

1. Create a zip file with **ONLY**:
   1. This Word document (filled out).
   2. The HTML file with all of your fixes.
2. Name the zip file: *2-Accessibility-Audit-FIRSTNAME-LASTNAME.zip*
3. Submit the zip file on Moodle.