WCAG Compliance Guide for LaTeX Docs

Sergio García-Vergara

April 12, 2023

Abstract

This document details the steps for generating LaTeX documents to be WCAG compliant. The goal is for screenreaders to accurately handle the compiled documents. Namely, this means reading the document in order, and handling the math equations and section headings accurately.

1 Introduction

Web Content Accessibility Guidelines (WCAG) are a set of recommendations for making Web content more accessible. For a document to be considered to be WCAG compliant, the following need

* **Use semantic markup**: Make sure to use semantic markup in your LaTeX document. This means using the appropriate LaTeX commands for headings, lists, tables, and other elements. This will help screen readers and other assistive technologies to understand the structure of the document.
* **Provide alternative text for images**: Use the *includegraphics* command to include images in your LaTeX document, and provide alternative text for each image using the alt parameter. The alternative text should describe the content or function of the image.
* **Use accessible colors**: Choose colors that have sufficient contrast for people with color vision deficiencies. You can use online tools such as the WebAIM Contrast Checker to check the contrast ratio of your color combinations.
* **Use appropriate font sizes**: Use a minimum font size of 12pt for body text, and larger sizes for headings and other important elements. This will ensure that the text is legible for all users.
* **Add metadata**: Add metadata to your document, including title, author, and language information. You can do this using the hyperref package in LaTeX.

This guide describes the steps that authors should follow to generate accessible documents using LaTeX. Namely, this guide will walk you through correctly handling: section and subsection headers, embedded URLs, math equations, figures and images, and tables.

1.1 tex4t: Command-line Tool

Overall, while PDF documents can be made WCAG compliant, HTML is often a more accessible format for online content. By converting your LaTeX document to HTML, you can make your content more accessible to a wider audience.

2 Document-level Instructions

The following sections describe what authors need to do in their *.tex* files such that documents are WCAG compliant. The instructions for generating the appropriate HTML files (once the *.tex* file is complete), are described in Section 3: *Compiling .tex file to HTML*.

2.1 Embedded URLs

This is an example of an embedded URL. This Google link will take you to Google's webpage. You just need to use the `href` package. Make sure to specify the use of the package at the top of your tex doc: **usepackagehref**.

2.2 Images and Figures

Work in progress.

2.3 Equations

The flags used to compile your document to HTML automatically take care of correctly rendering your math equations. You just need to make sure that they're formatted correctly.

This is an example inline math equation: $E = mc^2$. Note the use of the dollar sign before and after the equation.

The following is an in-paragraph math equation:

$$\frac{d^2y}{dx^2} + p(x)\frac{dy}{dx} + q(x)y = f(x) \tag{1}$$

Note that there are no dollar signs in this format. The main difference between these two use cases is that the in-paragraph structure automatically tags it with a number such that it can later be referenced in other parts of the document.

2.4 Tables

Work in progress.

3 Compiling .tex file to HTML

3.1 Ubuntu

Fist you'll need to install the `tex4ht` package by executing the following:

`sudo apt update` `sudo apt install tex4ht`

Then execute the following command on your Linux terminal to compile the html files:

`htlatex mydocument.tex "xhtml,mathml,html5,charset=utf-8" " -cunihtf -utf8"`

Make sure to replace the *myfile.tex* name with the name of your document.

3.2 Widows

Work in progress.

4 Other Methods

As I worked on putting together this guide, I found several tools for automatically tagging PDFs