

LaTeX and HTML for ADA and Accessibility

Ettore Aldrovandi

Florida State University

March 13, 2026

I will review accessibility issues for websites, personal websites, and Canvas. I will then cover practices and workflows to ensure that documents including math-heavy content are fully compliant to accessibility requirements, focusing primarily on \LaTeX , PDFs, and HTML formats.

Plan of the talk

Formats

Sites

LaTeX

LaTeX to HTML

Beyond LaTeX: Markdown + Pandoc

- **PDF** — the dominant format for academic documents
- **HTML** — web pages, Canvas content, online documentation
- **DOCX / PPTX** — Microsoft Office formats
 - Will largely ignore MS Office for the rest of this talk

PDF

- PDF 1.7 (ISO 32000-1)
- PDF 2.0 (ISO 32000-2)
- PDF/UA (Universal Accessibility)

Web

- HTML 5 / XHTML 1.1
- MathML — math on the web, embedded in PDF

Accessibility Requirements

- **Proper tagging** — structural tags that allow screen readers to interpret document organization
- **Color schemes** — sufficient contrast ratios (WCAG AA/AAA)
- **Font choices** — readability, dyslexia-friendly options (e.g., OpenDyslexic, Atkinson Hyperlegible)
- **Alternative text** for images and figures
- **Semantic markup** — meaningful headings, lists, tables

Formats

Sites

LaTeX

LaTeX to HTML

Beyond LaTeX: Markdown + Pandoc

We have to make our sites accessible: change how and what we post, in particular how we deal with PDF and HTML files.

Recommendations

- Link to PDFs *off-site* rather than hosting locally:
 - Link to arXiv.org preprints
 - Link directly to journal articles
 - Use the **FSU Library** to link to papers behind a paywall (proxy links ensure students have access)
- This delegates the accessibility burden to arXiv / the journal

Canvas vs. plain website

- Canvas is *not* just a website — it is a Learning Management System with its own rendering pipeline
- Canvas uses **Ally** to score accessibility of uploaded content
- Ally checks much more than structural tags

Known issues

- PDF files must be accessible before upload
- HTML rendering in Canvas is quirky:
 - Limited (by design?) support for HTML5 tags
 - Cannot simply provide a CSS stylesheet for styling
 - Nevertheless, well-formed HTML files seem to render correctly

Formats

Sites

LaTeX

LaTeX to HTML

Beyond LaTeX: Markdown + Pandoc

- **Source is better.** L^AT_EX provides structured, semantic markup that compilers can translate into properly tagged output.
- Tagging *already existing* PDFs is awkward:
 - Must reconstruct lost structural information
 - Tools (Adobe Acrobat, PAC 3, etc.) help but are imperfect
 - No automated tool handles mathematics reliably
- **Conclusion:** produce accessible PDFs from L^AT_EX source, not by post-processing.

Why luatex?

- luatex is the *engine*; latex is the *format*; the combination is luatex
- luatex is the **future of T_EX**:
 - pdftex is *frozen* — only maintenance, no new features
 - xelatex is *unmaintained*
 - luatex is actively developed and is the tagging engine
- Full **Unicode** support:
 - Type é directly instead of \'e
 - Correct handling of international characters and math symbols
- The new L^AT_EX tagging infrastructure (tagpdf, etc.) targets luatex

Non-negotiable baseline

- A **very** recent T_EX distribution:
 - TeXLive 2026
 - MikTeX on Windows
- Compile with `lualatex-dev`

Clean code

- Use `\usepackage{geometry}` — no explicit page formatting
- Define macros with `\newcommand`
- Use `\[... \]` or `equation*` for displayed math
- Leave sectioning commands untouched

Tagging

- Formulas are tagged automatically by the compiler
- Caption all figures and tables
- Provide alternative text: `\includegraphics[alt={description}]{...}`

Color and fonts

- Consider using color-blind friendly color schemes: `\usepackage{colorblind}`
- Consider using dyslexia-friendly fonts, for alternative versions of documents

Things that *hinder* accessibility or break tagging:

- `\def` instead of `\newcommand`
- `$$...$$` for displayed equations (use `\[...]` instead)
- Using `TeX` primitives instead of `LaTeX` constructs
- `\input` instead of `\include` / `\includeonly`
- Using the `enumerate` package or otherwise overriding standard list environments
- Using the `nopageno` package
- `\thispagestyle{empty}`
- ...

- `\documentstyle` — obsolete \LaTeX 2.09 syntax

- The `\makeatletter ...\makeatother` hack:

```
\makeatletter
% internal \@commands that shouldn't
% be touched by normal users
\makeatother
```

- Googling for \LaTeX commands — or worse, asking **ChatGPT**
 - LLM-generated \LaTeX code is frequently incorrect and often uses obsolete or conflicting commands—**says Claude!**

These lead to **unfixable** PDFs from an accessibility standpoint:

Beamer

- Beamer produces tagged output only with significant effort
- Use `ltx-talk` instead (what you are looking at!)

AMSart / AMSBook

- Produce PDFs that cannot be made fully accessible
- Presumably still okay for HTML output (via `latexml`, `make4ht`, etc.)
- The *packages* `amsmath`, `mathtools`, `amssymb`, `amsthm` are fine to use with any class

```
\DocumentMetadata{
  tagging      = on,
  tagging-setup = {math/setup={mathml-SE,mathml-AF},math/alt/use},
  pdfstandard  = ua-2,
  pdfversion   = 2.0,
  lang         = en,
}
\documentclass{article}
...
...
```

LaTeX to PDF: Skeleton (cont.)

```
...  
\usepackage{hyperref}  
\hypersetup{  
  pdftitle = {Mathematical Accessibility Test},  
  pdfauthor = {Ettore Aldrovandi},  
}  
...  
\begin{document}  
\title{Mathematical Accessibility Test}  
\author{Ettore Aldrovandi}  
\maketitle  
...  
\end{document}
```

Choosing a font with good Unicode coverage is crucial for accessibility. The `unicode-math` package is necessary for good tagging of math symbols.

In the preamble

```
\usepackage{unicode-math}  
\setmainfont{TeX Gyre Termes} % or any other font with good Unicode coverage  
\setmathfont{texgyretermes-math.otf} % math font matching the main font
```

You can leave out the `\setmathfont` command and the compiler will use the default math font, which is usually fine for accessibility purposes. Same for the main font if you are happy with the default (Computer Modern).

Font structure (cont.)

There are dyslexia-friendly fonts available that can be used for better readability. For example, the OpenDyslexic font can be set up as follows:

Preamble with dyslexia-friendly fonts

```
\usepackage{unicode-math}
\setmainfont{OpenDyslexic}[
  Path          = /Path/to/Fonts/,
  Extension     = .otf,
  UprightFont   = OpenDyslexic-Regular,
  BoldFont      = OpenDyslexic-Bold,
  ItalicFont    = OpenDyslexic-Italic,
  BoldItalicFont = OpenDyslexic-BoldItalic,
]
\setmathfont{texgyretermes-math.otf}
```

Including graphics

In the preamble

```
\usepackage[option1,option2,...]{graphicx}
```

Within the document

Use the following alternatives:

```
\includegraphics[alt=<alternative text>,...]{<filename>}
```

```
\includegraphics[actualtext=A,...]{example-image-a}
```

```
\includegraphics[artifact,...]{whatever}
```

The latter should be used for decorative images that do not convey information.

The same technique works for figures created with the TikZ package. They can be made accessible by using the `alt` key in the `\tikz` command or in the `tikzpicture` environment.

In the preamble

```
\usepackage{tikz}
```

Within the document

```
\begin{tikzpicture}[alt={A simple triangle}]  
  \draw (0,0) -- (1,0) -- (0.5,0.866) -- cycle;  
\end{tikzpicture}
```

Compile with latexmk

```
latexmk -lualatex \  
  -lualatex="lualatex-dev" \  
  -synctex=1 <filename>.tex
```

Compile with lualatex directly

```
lualatex-dev -synctex=1 <filename>.tex
```

Formats

Sites

LaTeX

LaTeX to HTML

Beyond LaTeX: Markdown + Pandoc

- ArXiv.org has been converting \LaTeX source to HTML using \LaTeXML (<https://dlmf.nist.gov/LaTeXML/>)
- \LaTeX interpreter implement in Perl. It produces HTML5 output with MathML for math, and it is reasonably good at handling tagging and accessibility features.
- Compile with:

```
latexml --dest=<filename>.html <filename>.tex
```

- Better compilation:

```
latexmlc \  
--pmm1 --cmml --mathtex --unicodemath --index \  
--css=latex-style.css "$input" --dest="$output"
```

This assumes you have a `latex-style.css` file to style the output beyond the \LaTeXML default. Leave it out otherwise.

Formats

Sites

LaTeX

LaTeX to HTML

Beyond LaTeX: Markdown + Pandoc

- Markdown is a simple markup language that can be converted to various formats, including HTML and PDF.
- Pandoc is a powerful tool that can convert Markdown to HTML, PDF, and other formats while preserving accessibility features.

- Compile with:

```
pandoc -s -t html5 -css=<filename>.css <filename>.md -o <filename>.html  
pandoc <filename>.md -o <filename>.pdf
```

- Pandoc supports various extensions for enhanced accessibility, such as adding alt text to images and generating proper headings and lists.

Thank you!

Questions?