

# Creating Accessible Documents with LaTeX

Jim Tyson

19 June 2019

## Contents

<b>1</b>	<b>To non <math>\LaTeX</math> users</b>	<b>1</b>
<b>2</b>	<b>Introduction: what is and isn't accessible</b>	<b>2</b>
<b>3</b>	<b>Guidance for LaTeX users</b>	<b>2</b>
3.1	Structure . . . . .	3
3.2	Formatting . . . . .	3
3.3	Labelling images (Alt text) . . . . .	4
<b>4</b>	<b>Use of colour</b>	<b>5</b>
<b>5</b>	<b>Using pandoc</b>	<b>6</b>
5.1	Creating the web page . . . . .	6
5.2	Shortcomings of Pandoc conversion . . . . .	6
<b>6</b>	<b>Conclusion</b>	<b>6</b>
<b>7</b>	<b>To Do</b>	<b>7</b>
	<b>References</b>	<b>7</b>

## 1 To non $\LaTeX$ users

This document is intended for readers who **already** use  $\LaTeX$ . A number of very good guides already exist for learning  $\LaTeX$ .

## 2 Introduction: what is and isn't accessible

What makes a document accessible is not a matter of what software technology is used to produce it. An accessible document however produced should meet the standards set out in UCL's basic guidance on accessibility pages.

The basics as set out there cover

- structure;
- formatting;
- links;
- labelling images;
- use of colour;
- content;
- transcripts and captions.

Issues regarding video material (**transcripts and captions**) are not of particular relevance to  $\text{\LaTeX}$  users but I will address the other matters below.

## 3 Guidance for $\text{\LaTeX}$ users

For screenreader users, the least good solution is to provide the source  $\text{\LaTeX}$  file. This is not generally acceptable. The burden of understanding content from the raw  $\text{\LaTeX}$  is unacceptably high.

Instead, the source should be prepared in such a way as to produce a document that is acceptably read by a screen reader. In some respects the ideal would be a tagged PDF - since PDF is the usual output generated in educational contexts. However, there is currently (to my knowledge) no way to generate a fully tagged PDF document that includes readable mathematics from the  $\text{\LaTeX}$  source. The most accessible format that can be readily generated from the  $\text{\LaTeX}$  source is HTML. To create this HTML, writers can use the utility **pandoc** ("Pandoc - About Pandoc" n.d.) which ensures that the mathematical content is transformed to MathML to be read by screen reader software.

Mathematics is typeset in the usual way. For example:

$$\frac{\sum (x - \bar{x})^2}{N}$$

I will describe using **pandoc** later in this document after I have outlined steps to be taken in preparing the  $\text{\LaTeX}$  source.

## 3.1 Structure

Most LaTeX writers will use the `\section{}` family of commands to create a structured document. These heading styles will be understood when a PDF is rendered and will create a navigable document for a screen reader user. It is also recommended that a table of contents is included to increase accessibility.

## 3.2 Formatting

### 3.2.1 Line Spacing

Set line spacing for your documents to one and a half lines using the package **setspace**. This makes available the command `\onehalfspace`. All text following this will be set with 1.5 line spacing (unless another of the line spacing commands is issued to switch to another setting).

### 3.2.2 Fonts and typefaces

Where possible, change the font family for the document with the following command in the preamble:

```
\renewcommand{\familydefault}{\sfdefault}
```

This will set a sans serif **Computer Modern** as the default for the document. Use `\textbf{}` for emphasis rather than `\emph{}` which produces italic text by default. If you are confident, you may consider redefining `\emph{}` to produce **bold text**. This is not difficult but to preserve the toggling behaviour of `\emph{}` requires some LaTeX know how. The code below inserted in the document preamble will do it:

```
\makeatletter
\DeclareRobustCommand{\em}{%
  \@nomath\em \if b\expandafter\@car\f@series\@nil
  \normalfont \else \bfseries \fi}
\makeatother
```

(**Formatting - What Is the Canonical Way to Redefine the `\Textbackslashemph` Command** n.d.)

### 3.2.3 Links

**Hyperlinks to web pages** Use the command `\href{url}{linktext}` rather than use bare urls for inserting hyperlinks. So

```
\href{http://www.ucl.ac.uk/isd}{ISD home page.}
```

Which produces ISD home page.

**Internal links and cross references** It will be additionally helpful to make all **cross references** in your document into internal hyperlinks. You can do this simply by import the package **hyperref** in the pre-amble of your document with the `\usepackage` command. Usually, **hyperref** must be the **last** package imported. The **table of contents**, **cross references** to graphics and tables, and citations, will all become document internal links. You can control the colours used for links as follows:

```
\usepackage{hyperref}
\hypersetup{
  colorlinks=true,
  linkcolor=blue,
  filecolor=magenta,
  urlcolor=cyan,
}
```

You can use the general  $\text{\LaTeX}$  `\tableofcontents` command to generate a table of contents which will now be clickable and will be usable for navigation by screen reader users.

## 3.3 Labelling images (Alt text)

If all images (and tables, and graphs and charts) have captions, a document created from your  $\text{\LaTeX}$  document will be reasonably accessible. You might consider the placing of your captions: would it be better for the caption to **precede** the graphical element rather than follow it?

To create something more like true **alt-text**, you can use the package **pdfcomment**. With this package you can create a tooltip for an element as follows:

```
\pdftooltip{\includegraphics{img.png}}{This is the ALT text}
```



Figure 1: This is a chick.

Here is an example:



Note that you cannot use this method with the **figure** environment, and so images cannot have both a **pdftooltip** and a **caption**.

This works reliably with Adobe Reader, but some pdf readers may produce a less good result. You cannot use this method when producing HTML with **pandoc**. Instead you should use the **figure** environment as in figure 1.

```
\begin{figure}
\includegraphics{chick.jpg}
\caption{This is a chick.}
\end{figure}
```

## 4 Use of colour

General guidelines about the use of colour in images apply and should guide the creation of images in  $\text{\LaTeX}$ .

For some users a background colour other than white is helpful. You can set the page background colour using the package **xcolor**. This package makes a variety of colour control commands and of colours available. Some colours are in the ‘base’ **xcolor** package, others are made available by **options** when the package is loaded. Here is an example of loading the package with the **svgnames** options:

```
\usepackage[svgnames]{xcolor}
```

This command

```
\pagecolor{Ivory}
```

in the preamble will set the page background colour to **Ivory** from the colours made available by the **svgnames** option.

## 5 Using pandoc

If you follow the advice above then a PDF document generated in the usual way will be reasonably accessible with one important exception: mathematical content. Unfortunately screen reader software will not cope well with the mathematical content produced by the usual methods (ie running latex or pdflatex commands.)

The best solution I have found is a two step procedure. Generate a **tex** source file and use **pandoc** to convert it to html. **Pandoc** allows an option to convert  $\text{\LaTeX}$  maths code to be converted to MathML which is read very well by modern screen reader software. This means that a document may need to be distributed in two file formats, but the burden is relatively low since both are generated from the same source.

You can download **pandoc** from the pandoc installation page. It is available for Windows, MacOS, linux and Chrome OS.

### 5.1 Creating the web page

Once you have a tex source file you can generate the html equivalent by typing the following code on the command line interface (terminal, cmd):

```
pandoc maths.tex -toc -s -mathml -o maths.html
```

assuming that your source file is called **maths.tex**.

### 5.2 Shortcomings of Pandoc conversion

Note that some things won't survive the conversion to HTML. If you use **pdftooltip** to create alt-text for images, neither the image nor the tool-tip will appear in the HTML page. For this reason, if you are going to create an HTML version of a document for accessibility, you should use the package **graphicx** (as most  $\text{\LaTeX}$  users usually do) and provide captions. Overall, I recommend the use of captions over tool-tip style alt-text for this reason.

## 6 Conclusion

Giving up  $\text{\LaTeX}$  would be burdensome for many writers. Nothing else I have so far provides for such simple and sophisticated typesetting not only of general mathematics, but in my experience of logic and of the notations used in linguistics. I suspect that the same is true of typesetting other STEM subjects such as chemistry and the engineering disciplines in general.

I am still testing documents produced according to the principles in this document. So far the results are good but it is always possible that I am missing some points or that people can offer improvements. I welcome any comments by email to [j.tyson@ucl.ac.uk](mailto:j.tyson@ucl.ac.uk).

## 7 To Do

1. References for pdfcomment.
2. References for setspace.

## References

**Formatting - What Is the Canonical Way to Redefine the `\Textbackslashemph` Command.** n.d. Accessed June 24, 2019. <https://tex.stackexchange.com/questions/6754/what-is-the-canonical-way-to-redefine-the-emph-command>.

"Pandoc - About Pandoc." n.d. Accessed June 26, 2019. <https://pandoc.org/>.