Using Latex to build web content

Latex is an old-school language for document typesetting. It was created by Donald Knuth to typeset his book The Art of Computer Science. You still see LaTex in scientific articles and papers

If you're only familiar with HTML, Latex syntax will look strange. Rather than tags and attributes we have a preamble, package declarations and intructions.

A basic LaTex article, set to print in portrait mode with a body text size of 12 points looks like this:

```
\usepackage{amssymb}
\usepackage{epstopdf}
% Broken into two lines for readability. In production
% the command would go in one line
\DeclareGraphicsRule{.tif}{png}{.png}
{`convert #1 `dirname #1`/`basename #1 .tif`.png}

\title{Brief Article}
\author{The Author}
%\date{}
% Activate to display a given date or no date

\begin{documen
\maketitle
\section{}
\subsection{}
\end{document}
```

documentclass indicates what type of document we want to creaate the parameters (font size in this case) is in square brackets []

\usepackage{} loads modules for use into the document

Commands that start with a backslash, like \geometry{} and

`\DeclareGraphhicsRule{}` are instructions that will generate output of some kind for the document.

Converting Latex to HTML

I like LaTex but I'm still a fan of the web and I want to make sure that whatever I create in LaTex is also available on the web, assuming the publisher allows me to :)

There are two ways to create HTML content from LaTex files. The first one is `tex4ht` or `htlatex` and the second one is `make4ht`, an abstraction on top of tex4ht that simplifies adding options to the different pieces of the configuration.

The rest of the article uses the tex file from [this gist](#) as the source for the commands.

tex4ht

tex4ht converts LaTex sources into one or more HTML documents with a very (and I mean very) basic style sheet that you can customize and expand as needed.

The most basic command will create a single page for all the content along with the corresponding image

```
htlatex article.tex
```

For shorter articlles the single-file approach may be ok (with customized styles) but for larger files or articles with larger sections it may prove harder to read online.

We can break the arcile down into multiple files based on the headings on the document.

The example below will generate mutliple files and it will also generate navigation links within the pages of the document.

The styles, as with the previous document can definitely be enchanced.

```
htlatex article.tex "html,index=2,3,next"
```

For those interested, you can also convert your LaTex to Docbook 5.0. While you can also convert it to TEI, it fails to convert the file successfully and I'm not certain why as the document converts succesfully to Docbook.

```
# Conversion to Docbook
htlatex article.tex "xhtml,docbook" " -cuni"xhtml,docbook"# Conversion to
htlatex article.tex "xhtml,tei" " -cunihtf" "-cdocbk"
```

make4ht

As we've discussed `tex4ht` system supports several output formats with multiple steps and multiple parameters possible for each step and format combination.

    I just want to make sure this is visible as it'll save a lot of time if you know it exists and where you can find its documentation.

    The most basic version of the htlatex command will convert the TeX file into HTML using UTF-8 as the encoding:

```
make4ht -uf html5 filename.tex
```

When you just add new text to your TeX document, without cross-references, or new additions to the table of contents, you can use `draft` mode which will invoke LaTeX only once. It can save quite a lot of the compilation time:

```
make4ht -um draft -f html5 filename.tex
```

As with many things in the TeX universe there are a lot of configuratuion options. I'm deliberately not covering them both to keep the post from ballooning in size and to avoid confusion; I'll assume that you know where to find the documentation if you need it.

Items to research and conclusion

Using LaTeX as the source for documents presents some clear advantages and some interesting challenges. TeX and LaTeX were designed from the start to work as print typesetting languages and the quality of the printed result is clearly better than what we can get from HTML alone.

Particularly with `make4ht` there are many questions left to answer. Sone of the questions that merit additional research:

- Would the output of the tool using the `staticsite` extension be good enough for static sites other than Jekyll?
- Is the output in `Tei` and Docbook good enough to feed to their corresponding processing toolchains? If not what additional changes do we need to make?
- Is it worth learning Lua just to automate one type of task for one tool?

Links and resources

- [tex4ht](#)
- [Tex4ht: HTML Production](#)
- [make4ht](#)
- [tex4ebook](#)