What I learned in TechDoc
Showing the wonderfull world behind Technical Documentation

Dennis Houterman

April 7, 2017

Contents

1   The inner workings of Latex

The first lesson in this course was one that explained how Latex worked. A very introductory glance at what Latex is capable of as well as showing how it is formatted. How you basically have your main text of your document where you write the body of your work, and besides that you have Title's, picture's, headers and footers, table's or anything else, that can be inserted into the document by giving commands within the document. By forcing the user to define all insertable elements in commands, you attain separation of the elements with the text. This approach differs greatly from programs such as Libreoffice Write or Microsoft Word, where the elements are just pushed in the document without the code visible. While for your average user, office products as mentioned above are better because you don't need a lot of technical know-how, but if you want precise control over your document and have great adaptability of your document so it can be re-used easely and transported to other formats if you so desire, Latex is a very interesting language for you.

2   working with Latex Editors

So we have an understanding of what Latex can do, so how do i start?

Lets start with finding ourselves a way to make Latex documents. wonderfully enough, you are able to create document's with Latex in notepad or any other simple text editor where you can edit and save documents. besides that you need a latex compiler. in linux this was dead easy to get with a simple:

```
>sudo apt install latex-full
```

and i could compile my latex files with:

```
>latex [file].tex
```

So im able to make documents! Yay! But this command line approach isn't really fancy, so is there a better way? Of course! Why else would i bring it up?

you can get Editors for Latex easy. Texmaker for example is a Latex editor that is available for Windows, Linux and MacOS. webbrowsers can also do Latex with sites such as www.sharelatex.com and www.overleaf.com . These editors allow you to create documents and check them in the same program. as well as show you errors or problems in your document as well as handle the citations and much more.

3   Gitting Guud

The godfather of documentation would be Git. Git is a versionmanager that can include files in its system and save a snapshot of it. When someone makes an edit to the file, the edit gets saved and yet another time there is made a snapshot of it.

I can hear you say: "Big deal, my dropbox can do that too" but here we come to the divergent part about git. Git is different because the edits that are made to a document are recorded in lines. everytime you press enter in a document, it will get recognized as a new line in Git. This line approach allows multiple people to be working on a document on separate lines.

lets say, if we were to design this document, one person edits the part about "the inner workings of Latex" and the other person edits "working with Latex Editors". when the first person would save its edits and the second person would come after and save it, you would get conflicts in Dropbox, because your documents have edits that the other person didnt have, causing a conflict. In Git, this issue would get resolved without problem. this is because of that line thing i talked about earlier. The lines that the first person edited in the first chapter, didnt overlap with the lines the second person edited in the second chapter. There lies the true power of Git. The possibility to work on each others documents without having constantproblems with needing to merge the differences in files.

"But what if the same line gets edited?" Then you will get a merge conflict. Git will generate a document that will show where the merge conflict has occured completely with the lines that have the differences that caused the merge conflict to occur.

4   More then a thousand words

If you want to display images, it can be done. like this image here:
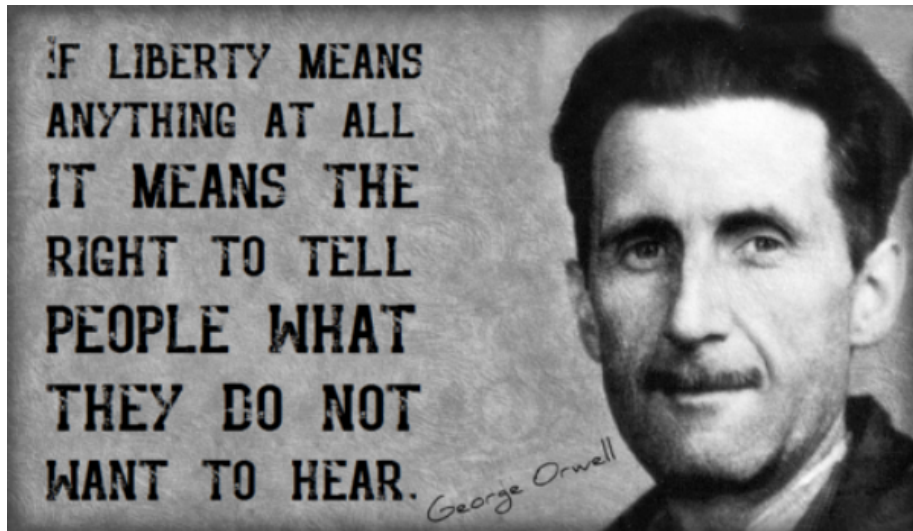
this image is included with the simple command of:

```
\includegraphics *[scale = 0.57]{image}
```

of course you need to include this package before your document starts:

```
\usepackage{fancyhdr}
```

5   Font snazzy, make your work look Classy

The lesson of typography wasn't one that got deep into the nitty gritty of implementing fonts. the implementation of font's isn't that hard in a formal like Latex or XeLatex which is the format that accepts simple imports of font's:

*****I think Carlito is a kickass font that looks elegant as well as playfull and just clear****

```
\usepackage{fontspec}
\setmainfont[Ligatures=TeX]{Carlito}
```

This lesson was more about the intrinsic characteristics of the design of font's. How font's look to the eye's and practical applications of changes in font's to make them more scalable, trustworthy or in general how look and feel change's how the message will get received (for example, making your report in "comic sans" will make your message less trusted).

To get a complete picture of everything that was said about typography, I refer to the slides. the slides have demonstrated excelently what font's can do in a nutshell. as the teacher has said: "typography is a course in and of itself" and to explain everything in a 2 hour span, would be impossible. At least we know that we have to avoid "comic sans".

6   Citation Needed

Besides making a simple assignment for TechDoc, or personal project descriptions, you might want to use Latex for research. but to be able to describe research you need sources to draw you research on. as a soon to be Applied Computer Scientist, I will need to do research and draw my research from sources to see if certain solutions are possible for example. By including research

[?] [?] [?]

7   The wonderfull world of document formats

If you're an eBook lover like me, you probably have encountered .epub and .pdf files. well, besides that .tex is also part of those 3 as it is just a format that holds the elements. where you can add a font to your tex file and some images to make it your own, .epub files have this too. images and fonts can just be put into an .epub file as it is basically a zipped package. A zipped package of all the components that file is made out of like font's used, image's, HTML files witht he body of text the book is made out of and so forth. this has been a big realisation for me, because that means i can just get some text files and transform them into .epub files optimized for my eReader. so everything you make in Latex, can also be exported to .epub if youre willing to spend some time adapting to the fileusage.

8   Conclusions of this course

This course has brought me a lot of valuable information. about the filetype's I have been using but did not comprehend where it was made out of, about how font types can impact how your message is brought and that several fonts can have different messages about the author, how to make very impressive slideshows with reveal.js and most of all, to make technical documentation that can be kept very clean by having full control over every element present in the text as well as the text itself.