

# Neals First DocOnce Publication

Neal Gordon\*

Jun 4, 2015

## Abstract

My goal here is to learn the package `doconce`, which is a markup syntax for creating scientific documents with many different types of outputs, including markdown, html, ipython notebooks, pdfs, and docx files. What's wrong with markdown? The short answer is nothing, it is great and important to know, however, some features that a needed for more advanced features, like figure/equation references, as well as including citations is needed, and haven't any good solution for that. All the up-to-date source code and various document formats can be found on the [github repository](#).

## Contents

<b>1</b>	<b>Introduction</b>	
1.1	SubIntro . . . . .	
<b>2</b>	<b>Headings, Labels, and References</b>	
2.1	my subheading . . . . .	
<b>3</b>	<b>Code</b>	
<b>4</b>	<b>Lists</b>	
<b>5</b>	<b>Comments and Footnotes</b>	
<b>6</b>	<b>Math, Latex</b>	
<b>7</b>	<b>Images</b>	
<b>8</b>	<b>File Structure</b>	
<b>9</b>	<b>Help</b>	
<b>10</b>	<b>Covertng Files</b>	
<b>11</b>	<b>Installation</b>	
<b>12</b>	<b>Sundries</b>	
<b>13</b>	<b>Handy Links</b>	
<b>14</b>	<b>References</b>	

## 1 Introduction

DocOnce is a lightweight markup language that is pure python, tso that it will work on any OS the source markdown files have extension \*.do.txt each document should have the structure with a fig and src directory to keep images and code. The make.sh

file is used to generate various documents with a lot of of custom settings.

### 1.1 SubIntro

Since I have began blogging, I realized the importance of learning html, markdown, etc and now have been trying my best to learn tex and L<sup>A</sup>T<sub>E</sub>X. While doconce is a little buggy, it is still under active development and have found it very useful for learning how all the various markups work together. I still find microsoft products very handy, especially v2013, but when working with the web and version control systems, plain text sources files are great! So, if you haven't heard of doconce yet, check it out!

**1 SubSubIntro.** The main goal here is to generate an example document to help me learn and show you the syntax of doconce. Hopefully it will make sense and the errors are minimal! This is still a work in progress.

## 2 Headings, Labels, and References

For simple documents, chapters are not necessary, so only 7= are necessary to create a header for the section.

```
3      chapter          ===== Heading ===== (9 =)
3      section         ===== Heading ===== (7 =)
3      subsection      ===== Heading ===== (5 =)
3      subsubsection   === Heading === (3 =)
3      paragraph       __Heading__ (2 _)
3      abstract        __Abstract__ Running text...
3      appendix        ===== Appendix: heading ===== (7 =)
3      appendix        ===== Appendix: heading ===== (5 =)
```

Labels are created with any text in curly braces and prefixed with label and then referenced with prefix ref. The ref must be enclosed with paranthesis. References can be made within the document with (1). Figures are referenced using (2), and to reference a section of the document like the code section 3 or the bibliography can be created using the prefix cite. One example of what doconce can be used for is an entire book [1]. Also, equations can be referenced in a similar fashion, such as (1).

Index of keywords can be defined with

```
4      index{name}
```

so that we have indextestindex, and my favorite language index-python

```
# define location label
label{sec:intro}

# reference location label
(ref{sec:intro})

# reference citations
cite{Langtangen}
```

\*Email: [nealagordon@gmail.com](mailto:nealagordon@gmail.com). ifcuriousthenlearn.

## 2.1 my subheading

testing the subheading

**my subsubheading.** finally, the smallest heading we can have

## 3 Code

Code can be identified in the text by enclosing the text in single back quotes, as in

```
'sudo apt-get install numpy'
```

And an inline rendering of that is `sudo apt-get install numpy`. Which is handy when you do not want to start a new line everytime.

```
# a comment in my python code
def f(x):
    return 1 + x
```

Also code can be inserted with this `@@@CODE src/hiwrlld.py`

```
# Here is an example python program
```

```
x = 'hi doconce world'
```

```
def hiworld(x):
    print(x)
```

```
# thats all
```

This way, all the code can be kept in a usable form or in inserted easily if changes

Here is the bash file that is used to generate all the various kinds of output files

```
#!/bin/sh
```

```
# create a variable of the doconce file
dofile='doconce_notes'
```

```
##### cleanup
####remove the pdfs and tex file that will be overwritten
rm $dofile.pdf
```

```
##### Latex
#### Create a plain tex file
doconce format pdflatex $dofile \
    --latex_code_style=vrbl \
    --latex_title_layout=std \
    --tables2csv \
    #--latex_code_style=vrbl
```

```
rm -f *.aux
pdflatex $dofile.tex
bibtex $dofile
pdflatex $dofile.tex
pdflatex $dofile.tex
mv $dofile.pdf $dofile.simple.pdf
```

```
#### Create tex file with fancy formatting and custom latex
doconce format pdflatex $dofile \
    --latex_code_style=pyg \
    --latex_title_layout=std \
    --latex_preamble=customization.tex \
    #--latex_todonotes # includes inline notes
```

```
#### helps position the figures
doconce replace 'begin{figure}' 'begin{figure}' $dofile.tex
```

```
# create the pdf of the tex file
rm -f *.aux
pdflatex -shell-escape -interaction=batchmode $dofile.tex
bibtex $dofile
pdflatex -shell-escape -interaction=batchmode $dofile.tex
pdflatex -shell-escape -interaction=batchmode $dofile.tex
```

```
mv $dofile.pdf $dofile.1col.pdf
```

```
#### creates a two column article
python insert_tex.py $dofile.tex
# create the pdf of the tex file
rm -f *.aux
pdflatex -shell-escape -interaction=batchmode $dofile.tex
bibtex $dofile
pdflatex -shell-escape -interaction=batchmode $dofile.tex
pdflatex -shell-escape -interaction=batchmode $dofile.tex
mv $dofile.pdf $dofile.2col.pdf
```

```
##### Markdown
```

```
#create the github markdown from the doconce_notes.do.txt file
doconce format pandoc $dofile --github_md
```

```
#From markdown to docx
pandoc -s -S $dofile.md -o $dofile.md.docx
```

```
##### ipython notebook
```

```
# create ipython notebook file
doconce format ipynb $dofile
```

```
# create an html file from the ipynb
ipython nbconvert $dofile.ipynb
mv $dofile.html $dofile.ipynb.html
wkhtmltopdf $dofile.ipynb.html $dofile.ipynb.pdf
```

```
##### Tables
doconce format --tables2csv
##### HTML
```

```
# create a styled html file
doconce format html $dofile --html_style=tactile-black
wkhtmltopdf $dofile.html $dofile.html.pdf
```

```
##### cleanup
```

```
doconce clean
```

```
# remove all the files that are generated during doconce make
rm *.toc *.log *.aux *.out *.idx *.bbl *.blg *.gz
```

```
# removes the trash directory
rm -R Trash
```

## 4 Lists

Here are some examples of lists directly from the documentation

Here is a bullet list:

- item1
- item2
  - subitem1 of item2
  - subitem2 of item2, second line of subitem2
- item3

Note that sublists are consistently indented by one or more blanks as shown: bullets must exactly match and continuation lines must start right below the line above.

Here is an enumeration list:

1. item1
2. item2 may appear on multiple lines
  - (a) subitem1 of item2
  - (b) subitem2 of item2
3. item3

And finally a description list:

**keyword1:** followed by some text over multiple lines

**keyword2:** followed by text on the next line

**keyword3:** and its description may fit on one line

## 5 Comments and Footnotes

Creating comments in the text is a handy way to supplement with optional information, one way is to use an inline comment such as **hpl 1:** here I will make some remarks to the text . Another way to add content is to add a footnote <sup>1</sup> is also possible.

## 6 Math, Latex

inline latex can look like this  $x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + a_4}}}$  and

another block

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = 0.$$

and some inline latex,  $\frac{\partial y}{\partial x}$  everybody loves partial differentiation and finally equations, can have a label such as

$$1 = \left( \frac{\sigma_1}{\sigma_{1u}} \right)^2 + \left( \frac{\sigma_2}{\sigma_{2u}} \right)^2 + \left( \frac{\tau_{12}}{\tau_{12u}} \right)^2 - \left( \frac{\sigma_1 \sigma_2}{\sigma_{1u}} \right) \quad (1)$$

## 7 Images

This section explains how to create images. The size can be adjusted with `frac` and the width adjusts the width when exporting to html

for a really tiny image, we can use



Figure 1: My favorite linux desktop with `frac-0.1`.

and for a figure that is a bit bigger, we can increase the fraction size



Figure 2: My favorite linux desktop with `frac-0.9`.

<sup>1</sup>The syntax for footnotes is borrowed from Extended Markdown.

## 8 File Structure

each document should have the structure with a `fig` and `src` directory to keep images and code. The `make.sh` file is used to generate various documents with a lot of of custom settings

```
doconce_notes.do.txt
doconce_notes.md
doconce_notes.pdf
fig
    mint.png
make.sh
test.do.txt
test.pdf
```

## 9 Help

help can be gotten from `doconce format --help`

## 10 Coverting Files

```
doconce format html test # creates an html document
doconce format pandoc test # creates markdown
doconce format latex test # creates latex document
doconce format ipynb mydoc # results in mydoc.ipynb
doconce format mwiki mydoc.do.txt # creates a wiki page
doconce format ipynb test.do # ipython notebook
```

**Pandoc** is a very powerful converter that can be used with `doconce`

```
#From markdown to docx
pandoc -s -S doconce_notes.md -o doconce_notes.md.docx
```

```
#From markdown to PDF:
pandoc doconce_notes.md -o doconce_notes.md.pdf
```

For compiling and document creation, please reference the `make.sh` file in this repository

When running the make file, make sure it is executable and run it with

```
chmod +x make.sh
./make.sh
```

## 11 Installation

I am using a Windows 7 laptop with

[VirtualBox Linux Mint 17.1](#)

Make sure your machine is up to date

```
sudo apt-get update && sudo apt-get upgrade
```

Then, install python. A barebones python install with IMHO, the best python package manager conda, the [miniconda package](#) is easy and [open-source](#)

```
# download
wget https://repo.continuum.io/miniconda/Miniconda-latest-Linux-x86_64.sh
# make sure shell file is executable
chmod +x Miniconda-latest-Linux-x86_64.sh

# Execute
./Miniconda-latest-Linux-x86_64.sh
```

I've gotten an error in the past with Linux Mint that conda command does not exist. This should fix it by creating a link with conda to the bin directory.

```
sudo ln ~/anaconda/bin/conda /usr/local/bin/
```

If miniconda is used, be sure to install the ipython notebook if you plan on using it

```
conda install ipython-notebook
```

Next, make sure all the dependencies are installed. Latex is an integral part of doconce, as well as pandoc for converting markdown files and the like

```
sudo apt-get install texlive-full
sudo apt-get install pandoc
```

Finally, we are ready to install doconce. There is a conda package that makes the installation very easy.

```
conda install --channel johannr doconce
```

Another option is to download a bash script [here](#) and use it to install

```
chmod +x install_doconce.sh
./bash install_doconce.sh
```

## 12 Sundries

DO NOT USE UNDERSCORES IN ANY PART OF YOUR DOCUMENT, ONLY FOR MARKDOWN.

## 13 Handy Links

Examples of a couple of the output files can be found here [doconcenotes1colpdf](#) , [doconcenotes2colpdf](#) , [htmlblack](#), [htmlipynb](#), and, although not great, [doconcedocx](#).

main site <https://github.com/hplgit/doconce>

[tutorial](#)

[tutorial2](#)

[demo](#)

[quickref](#)

[github website](#)

[scientific writing](#)

[more examples](#)

## 14 References

### References

- [1] Hans Petter Langtangen. *A Primer on Scientific Programming With Python*. Springer, 4 edition, 2014.