
DocOnce

Michael Weiß
Simon Schäfer

Jan 9, 2016

Inhaltsverzeichnis

1 DocOnce, eine universelle Markup-Sprache	3
1.1 Was ist DocOnce	3
1.2 Vor- und Nachteile	3
2 Installation	5
2.1 Ubuntu 15.04/15.10	5
2.2 Mac OSX	5
3 Syntax	6
3.1 Zeilenbasierte Auszeichnungen	6
3.2 Inline-Tagging	7
3.3 Block-Auszeichnungen	7
4 Ausgabeformate	9
4.1 Markdown	9
4.2 Rest und Sphinx	10
4.3 LaTeX	10
4.4 HTML	11
4.5 Wiki-Formate	11
5 Parameter für den doconce-Befehl	13
6 Fazit	14
7 Anhang	15
7.1 Setup-Script	15

Kapitel 1

DocOnce, eine universelle Markup-Sprache

1.1 Was ist DocOnce

DocOnce ist eine minimal ausgezeichnete Markup-Sprache, die alle wichtigen Funktionen einer Markup-Sprache bereitstellt und hierbei trotzdem die Anzahl der verwendeten Auszeichnungen möglichst gering hält. Durch die Verwendung von DocOnce ist eine Konvertierung der Ausgangsdatei in viele unterschiedliche Formate (Latex, Markdown, PDF, ...) möglich. Außerdem lässt sich das Ausgabeformat für mehrere Anwendungsbereiche anpassen. Dies funktioniert für analoge Medien, wie z.B. Bücher, Abschlussarbeiten und wissenschaftliche Berichte, aber auch für digitale Veröffentlichungen, wie z.B. in Blogs oder auf Wiki-Seiten.

1.2 Vor- und Nachteile

1.2.1 Vorteile

- viele Ausgabeformate
- geeignet für Autoren die für viele unterschiedliche Medien schreiben
- Viele unterschiedliche Designs für das Erstellen von z.B. html-Dokumenten
- Source-Code kann direkt aus Dateien ausgelesen werden
- kann auch nur als einfacher Text ausgegeben werden z.B. für Email oder Code-Dokumentationen]

1.2.2 Nachteile

- Installation auf Windows nicht möglich

- erstellte Formate (z.B. .tex) benötigen (stellenweise) händische Überprüfung
- erbt probleme der Zielformate wie z.B. automatisches Setzen von Grafiken in LaTeX

Kapitel 2

Installation

2.1 Ubuntu 15.04/15.10

Die Installation unter Ubuntu kann durch ein Bash-Skript (siehe Anhang) durchgeführt werden. Zusätzlich zu den im Original Install-Skript angegebenen Pakete mussten, für die Installation des lxml-Python-Packages, auf Ubuntu 15.04 noch die development-packages `libxml2-dev`, `libxslt1-dev` und `python-dev` installiert werden, da das Compilieren des Package ansonsten fehlschlug.

Nach der Installation kann DocOnce direkt im Terminal mit dem Befehl `doconce` und verschiedenen Parametern(siehe Kapitel 5 “Parameter für den `doconce`-Befehl”) gestartet werden.

2.2 Mac OSX

Unter Mac OS ist es notwendig den Paketmanager Anaconda für Python 2.7 zu installieren. Dieser ist unter <https://www.continuum.io/downloads> zu finden. Außerdem muss für die Umwandlung in das PDF-Format eine TeX-Distribution installiert sein. Sollte diese noch nicht vorhanden sein, kann diese unter <https://tug.org/mactex/> gefunden werden. Sobald diese beiden Pakete installiert sind kann über das Terminal mit dem Befehl

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

das DocOnce Paket installiert werden.

Wenn mit Pandoc Markdown gearbeitet werden soll, muss auch noch dieses Paket gesondert installiert werden, welches unter <http://pandoc.org/installing.html> gefunden werden kann. Das Paket wird benötigt um beispielsweise in das `docx`-Format von Microsoft Office zu konvertieren.

Ebenso wie unter Ubuntu kann DocOnce nun direkt im Terminal mit dem Befehl `doconce` und verschiedenen Parametern(siehe Kapitel 5 “Parameter für den `doconce`-Befehl”) gestartet werden.

Kapitel 3

Syntax

DocOnce verwendet drei grundlegende Arten von Auszeichnungen um beim konvertieren in verschiedene Markup-Ausgabeformate die Formatierung zu übertragen. Diese Auszeichnungsarten sind *zeilenbasierte Auszeichnungen*, das sog. *Inline-Tagging*, Auszeichnungen die im Fließtext vorgenommen werden, und *Block-Auszeichnungen*. Im folgenden sollen die wichtigsten Auszeichnungen behandelt werden.

3.1 Zeilenbasierte Auszeichnungen

Zeilenbasierte Auszeichnungen sind Auszeichnungen, die durch ein bestimmtes Schlüsselwort am Anfang der Zeile eingeleitet werden und dessen Inhalt bis zum Abschluss der Zeile anhalten.

Die wichtigsten zeilenbasierten Auszeichnungen sind: **Simon 1: Liste in Ordnung? oder Beispiele bringen?**

Befehl	Bedeutung
TITLE:	Titel des Dokuments
AUTHOR:	Autor
DATE:	Datum (today=aktuelles Datum)
{copyright,year1-year2 license}	Copyright
TOC:	Inhaltsverzeichnis
===== <i>Kapitel</i> =====	Kapitelüberschrift
===== <i>Abschnitt</i> =====	Abschnitt
===== <i>Unterabschnitt</i> =====	Unterabschnitt
=== <i>Unterunterabschnitt</i> ===	Unterunterabschnitt
-- <i>Paragraph</i> --	Paragraph
FIGURE:[Datei, width= frac=] Unterschrift	Bild einfügen
#	Kommentar
*	Liste
o	nummerierte Liste
- Wort: Erklärung	Beschreibungsliste

Die zeilenbasierte Auszeichnung zum Erstellen einer Tabelle nimmt eine Sonderstellung ein. Sie erstreckt sich über mehrere Zeilen wie in folgendem Beispiel zu sehen ist:

```
|----l-----l-----|
|Beschriftung 1|Beschriftung 2|
|----l-----l-----|
|Eintrag 1    |Eintrag 2    |
|Eintrag 3    |Eintrag 4    |
|Eintrag 5    |Eintrag 6    |
|Eintrag 7    |Eintrag 8    |
|-----|
```

Durch kleine Buchstaben über den Spalten (hier *l*) kann die Orientierung der Spalteninhalte festgelegt werden. Möglich sind *l*(= *left*), *r*(= *right*) und *c*(= *center*).

Das oben gezeigte Beispiel ergibt im fertig gerenderten Dokument folgende Tabelle:

Beschriftung 1	Beschriftung 2
Eintrag 1	Eintrag 2
Eintrag 3	Eintrag 4
Eintrag 5	Eintrag 6
Eintrag 7	Eintrag 8

3.2 Inline-Tagging

Das Inline-Tagging wird für das Hervorheben bzw. die spezielle Formatierung von Wörtern im Fließtext verwendet. Das Inline-Tagging umfasst alles von einfachen Formatierungen wie **Fett** oder *Kursiv* bis hin zum Verlinken von [Websites](#) oder lokalen [Dokumenten](#).

Befehl	Bedeutung
<i>*Text*</i>	<i>Kursiv</i>
Fett	Fett
\textcolor{red}{Text}	farbiger Text
'Code'	Quellcode
''Google'': ''URL''	Weblinks
''mailto'': ''mailto:Email''	mailto-Links
''Dokument'': ''/Pfad/zu/Dokument''	Link auf lokale Dateien
"Zitat"	"Zitate"
----	Horizontale Linie
Zitat----Vorname Nachname	Gedankenstrich(für Zitate)
[Name:Kommentar]	Kommentar im Fließtext
<linebreak>	Zeilenumbruch
\$Gleichung\$	Gleichungen (z.B.: $a^2 + b^2 = c^2$)

3.3 Block-Auszeichnungen

Die dritte Art von Auszeichnungen in DocOnce sind die Block-Auszeichnungen. Die Wichtigsten Vertreter sind der **Formel-Block** und der **Quellcode-Block**.

Formel-Block. Der Formel-Block wird verwendet um Mathematische Formeln und Funktionen übersichtlich dar zu stellen. Er wird durch `!bt` geöffnet und mit `!et` wieder geschlossen. Hier ein Beispiel der Formatierung der **Schrödinger Gleichung** durch einen Formel-Block (Zeilenbrüche bei Code-Beispiel eingefügt um Übersicht zu verbessern!).

```
!bt
\begin{align}
&\frac{\partial^2 \Psi(x,y,z)}{\partial x \partial x} + \frac{\partial^2 \Psi(x,y,z)}{\partial y \partial y} + \frac{\partial^2 \Psi(x,y,z)}{\partial z \partial z} = -c (E - E_{\text{pot}}(x)) \Psi(x) \quad \text{mit} \quad c = \frac{8\pi^2 m}{h^2}
\end{align}
!et
```

Wird zu:

$$\frac{\partial^2 \Psi(x, y, z)}{\partial x \partial x} + \frac{\partial^2 \Psi(x, y, z)}{\partial y \partial y} + \frac{\partial^2 \Psi(x, y, z)}{\partial z \partial z} = -c(E - E_{\text{pot}}(x))\Psi(x) \quad \text{mit} \quad c = \frac{8\pi^2 m}{h^2} \quad (3.1)$$

Zusätzlich werden Gleichungen nach dem Schema (*Kapitelnummer.Gleichungsnummer*) automatisch durchnummeriert.

Quellcode-Block. Wie der Formel-Block wird auch der Quellcode-Block durch zwei Auszeichnungen eingeschlossen. Dies sind am Beginn `!bc` und am Ende `!ec`. Auch hier ein kurzes Beispiel der Formatierung des klassischen "Hello World" Programm.

```
!bc
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World");
    }
}
!ec
```

Wird zu:

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World");
    }
}
```

Um das Code-Highlighting zu verwenden kann man hinter der Start-Auszeichnung `!bc` die im Quellcode verwendete Sprache angeben (im Beispiel `!bc java`). Zusätzlich muss der Parameter `--latex_code_style=pyg`, als Parameter, dem `doconce`-Befehl übergeben werden. DocOnce nutzt für das Code Highlighting das Paket `python-pygments`. Python-pygments unterstützt viele unterschiedliche Sprachen zum Hervorheben von Quellcode(siehe [Pygments-Homepage](#))

Kapitel 4

Ausgabeformate

4.1 Markdown

DocOnce lässt eine Konvertierung in viele Unterarten des Markdown zu und aus diesen lässt sich auch noch in viele andere Formate konvertieren.

Verwendung von Markdown. Die Konvertierung in Markdown erfolgt standardmäßig in das Pandoc-extended Markdown Format

```
doconce format pandoc DATEINAME
```

Die Ausgabe erfolgt in einer `DATEINAME.md` Datei. Die unterstützten Markdown Dialekte sind:

- Pandoc-extended Markdown
- GitHub-flavored Markdown
- MultiMarkdown
- Strict Markdown

Über das Pandoc Markdown lässt sich beispielsweise in `latex`, `html`, `odt` (OpenOffice), `docx` (Microsoft Word) und `rtf` umwandeln.

Ausgabeoptionen für Markdown. Mithilfe von Kommandozeilenparametern lässt sich der Markdown Dialekt spezifizieren:

- `--github_md`: Erzeugt ein GitHub-flavored Markdown File, welches Task-Listen unterstützt (unnummerierte Liste mit `[x]` (Task erledigt) und `[]` (Task noch zu erledigen))
- `--multimarkdown_output`: Erzeugt eine MultiMarkdown Version des Dokumentes
- `--strict_markdown_output`: Erzeugt ein einfaches oder striktes Markdown File ohne viele der Erweiterungen, die Pandoc akzeptiert.

Konvertierung von Markdown in HTML. Die HTML Ausgabe von pandoc benötigt einige Anpassungen um eine vollständige Unterstützung von MathJax \LaTeX Mathematik anzubieten und aus diesem Grund wird empfohlen den `doconce md2html` Befehl zu nutzen:

```
Terminal> doconce format pandoc DATEINAME
Terminal> doconce md2html DATEINAME
```

Die resultierende `DATEINAME.html` Datei kann mit einem Browser geöffnet werden.

Verwendung von Pandoc um von \LaTeX in MS Word zu konvertieren. Pandoc ist nützlich um von \LaTeX Mathematik in beispielsweise HTML oder MS Word umzuwandeln. Hierfür gibt es zwei Wege, welche beide ausprobiert werden sollten, um das beste Ergebnis für ein spezielles Dokument zu erreichen: `doconce format pandoc` gefolgt von `doconce md2latex` (führt pandoc aus), oder `doconce format latex`. Die resultierenden `.tex` Dateien können dann mithilfe von pandoc in das gewünschte Format konvertiert werden. Hier ein Beispiel der zweiten Möglichkeit:

```
Terminal> doconce format latex DATEINAME
Terminal> doconce ptex2tex DATEINAME
Terminal> doconce replace '\Verb!' '\verb!' DATEINAME.tex
Terminal> pandoc -f latex -t docx -o DATEINAME.docx DATEINAME.tex
```

Beachten Sie, dass DocOnce das `Verb` Makro des `fancyvrb` Paketes verwendet während pandoc nur das standard `verb` Konstrukt versteht. Desweiteren wären einige weitere Aufrufe von `doconce replace` bzw. `doconce subst` nötig um Formeln vollständig richtig in ein MS Word Dokument zu konvertieren.

4.2 Rest und Sphinx

Verwendung von Rest und Sphinx.

Ausgabeoptionen für Rest und Sphinx.

Konvertierung von Rest und Sphinx in weitere Formate.

4.3 LaTeX

Verwendung von LaTeX.

Ausgabeoptionen für LaTeX.

Konvertierung von LaTeX in weitere Formate.

4.4 HTML

Um ein DocOnce Dokument in eine Webseite einzubetten bietet sich an dieses in ein HTML-Dokument umzuwandeln.

Verwendung von HTML. Die Konvertierung in das HTML-Format erfolgt über den Befehl

```
doconce format html DATEINAME.do.txt
```

Das erzeugte Dokument mit dem Namen DATEINAME.html kann nun mit jedem Webbrowser zur Ansicht geöffnet werden.

Ausgabeoptionen für HTML. Mithilfe von Kommandozeilenparameter lässt sich das HTML-Dokument noch weiter personalisieren: Über den Befehl `--css=filename` lässt sich ein eigenes CSS Stylesheet zur Verwendung einbinden, wobei über der Befehl `--html_style=name` aus einigen vorgefertigten Styles ausgewählt werden kann.

- solarized: [solarized](#) Style
- blueish: einfacher Style mit blauen Überschriften (Standard)
- blueish2: etwas abgeänderter Variante von *blueish*
- bloodish: wie *blueish*, aber in dunkelrot

Andere Parameter:

- `--html_share=http://...` fügt am Ende des Dokumentes "Share Buttons" für Email, Facebook, Google+, LinkedIn, Twitter und Drucken ein
- `--html_share=http://...,twitter,linkedin` erzeugt "Share Buttons" für Twitter und LinkedIn. Die Seitennamen werden durch Kommata getrennt. Gültige Namen sind: *email*, *facebook*, *google+*, *linkedin*, *twitter* und *print*. Die URL muss auf die Seite weisen, auf der das Dokument publiziert wird.
- `--toc_depth=X`: bestimmt die Tiefe des Inhaltsverzeichnisses im Dokument. Der Standardwert beträgt 3 und erzeugt somit Einträge für Kapitelüberschriften, Abschnitte und Unterabschnitte.
- `--html_body_font=:` bestimmt die Schriftart für den Fließtext. Der Wert ? listet alle verfügbaren Schriftarten auf.
- `--html_heading_font=:` bestimmt die Schriftart für die Überschriften. Der Wert ? listet alle verfügbaren Schriftarten auf.
- `--html_video_autoplay=True,False`: lässt Videos automatisch abspielen, sobald die HTML Seite geladen wird (True, Standard).

4.5 Wiki-Formate

Verwendung von Wiki-Formaten.

Ausgabeoptionen für Wiki-Formate.

Konvertierung von Wiki-Formaten in weitere Formate.

Kapitel 5

Parameter für den doconce-Befehl

10%

Kapitel 6

Fazit

10%

Kapitel 7

Anhang

7.1 Setup-Script

```
#!/bin/bash
# Automatically generated script by deb2sh.py.

# The script is based on packages listed in debpkg_doconce.txt.

set -x # make sure each command is printed in the terminal

function apt_install {
    sudo apt-get -y install $1
    if [ $? -ne 0 ]; then
        echo "could not install $1 - abort"
        exit 1
    fi
}

function pip_install {
    sudo pip install --upgrade "$@"
    if [ $? -ne 0 ]; then
        echo "could not install $p - abort"
        exit 1
    fi
}

sudo apt-get update --fix-missing

# Installation script for doconce and all dependencies

# This script is translated from
# doc/src/manual/debpkg_doconce.txt
# in the doconce source tree, with help of
# vagrantbox/doc/src/vagrant/src-vagrant/deb2sh.py
# (git clone git@github.com:hplgit/vagrantbox.git)

# Python v2.7 must be installed (doconce does not work with v3.x)
pyversion='python -c 'import sys; print sys.version[:3]''
if [ $pyversion != '2.7' ]; then echo "Python v${pyversion} cannot be used with DocOnce"; exit 1; fi

# Install downloaded source code in subdirectory srclib
if [ ! -d srclib ]; then mkdir srclib; fi
```

```

# Version control systems
apt_install mercurial
apt_install git
apt_install subversion

# --- Python-based packages and tools ---
apt_install python-pip
apt_install idle
apt_install python-dev
apt_install python-pdftools
pip_install ipython --upgrade
pip_install tornado --upgrade
pip_install pyzmq --upgrade
pip_install traitlets --upgrade
pip_install pickleshare --upgrade
pip_install jsonschema
# If problems with IPython.nbformat.v4: clone ipython and run setup.py
# to get the latest version

# Preprocessors
pip_install future
pip_install mako --upgrade
pip_install -e git+https://github.com/hplgit/preprocess#egg=preprocess --upgrade

# Publish for handling bibliography
pip_install python-Levenshtein
pip_install lxml
pip_install -e hg+https://bitbucket.org/logg/publish#egg=publish --upgrade

# Sphinx (with additional third/party themes)
pip_install sphinx

pip_install alabaster --upgrade
pip_install sphinx_rtd_theme --upgrade
pip_install -e hg+https://bitbucket.org/ecollins/cloud_sptheme#egg=cloud_sptheme --upgrade
pip_install -e git+https://github.com/ryan-roemer/sphinx-bootstrap-theme#egg=sphinx-bootstrap-theme --upgr
pip_install -e hg+https://bitbucket.org/miiton/sphinxjp.themes.solarized#egg=sphinxjp.themes.solarized --up
pip_install -e git+https://github.com/shkumagai/sphinxjp.themes.impressjs#egg=sphinxjp.themes.impressjs --
pip_install -e git+https://github.com/krisksda/sphinx-sagecell#egg=sphinx-sagecell --upgrade

# Runestone sphinx books
pip_install sphinxcontrib-paverutils
pip_install paver
pip_install cogapp

#pip_install -e git+https://bitbucket.org/sanguineturtle/pygments-ipython-console#egg=pygments-ipython-con
pip_install -e git+https://bitbucket.org/hplbit/pygments-ipython-console#egg=pygments-ipython-console
pip_install -e git+https://github.com/hplgit/pygments-doconce#egg=pygments-doconce

# XHTML
pip_install beautifulsoup4
pip_install html5lib

# ptex2tex is not needed if --latex_code_style= option is used
cd srclib
svn checkout http://ptex2tex.googlecode.com/svn/trunk/ ptex2tex
cd ptex2tex
sudo python setup.py install
cd latex
sh cp2texmf.sh
cd ../../..

# LaTeX
apt_install texinfo
apt_install texlive-full

```



```

#apt_install texlive-extra-utils
#apt_install texlive-latex-extra
#apt_install texlive-latex-recommended
#apt_install texlive-math-extra
#apt_install texlive-font-utils
#apt_install texlive-humanities
apt_install latexdiff
apt_install auctex

# Image manipulation
apt_install imagemagick
apt_install netpbm
apt_install mjpegtools
apt_install pdftk
apt_install giftrans
apt_install gv
apt_install evince
apt_install smpeg-plaympeg
apt_install mplayer
apt_install totem
apt_install libav-tools

# Misc
apt_install ispell
apt_install pandoc
apt_install libreoffice
apt_install unoconv
apt_install libreoffice-dmaths
#epydok is an old-fashioned output format, will any doconce user use it?
#pip install -e svn+https://epydok.svn.sourceforge.net/svnroot/epydok/trunk/epydok#egg=epydok

apt_install curl
apt_install a2ps
apt_install wdiff
apt_install meld
apt_install diffpdf
apt_install kdiff3
apt_install diffuse

# tkdiff.tcl:
#tcl8.5-dev tk8.5-dev blt-dev
#https://sourceforge.net/projects/tkdiff/

# example on installing mdframed.sty manually (it exists in texlive,
# but sometimes needs to be in its newest version)
git clone https://github.com/marcodaniel/mdframed
if [ -d mdframed ]; then cd mdframed; make localinstall; cd ..; fi

```