
DocOnce

Michael Weiß
Simon Schäfer

Jan 11, 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
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	9
4.3 LaTeX	9
4.4 HTML	10
4.5 Wiki-Formate	10
5 Parameter für den <code>doonce</code>-Befehl	11
6 Fazit	13
7 Anhang	14
7.1 Setup-Skript	14
7.2 Skripte für die Ausgabe in Latex und Markdown	16

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 `doonce` und verschiedenen Parametern(siehe Kapitel 5 "Parameter für den `doonce`-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:

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
<code>\textcolor{red}{Text}</code>	farbiger Text
<code>' Code '</code>	Quellcode
<code>"Google": "URL"</code>	Weblinks
<code>"mailto": "mailto:Email"</code>	mailto-Links
<code>"Dokument": "/Pfad/zur/Dokument"</code>	Link auf lokale Dateien
<code>"Zitat"</code>	„Zitate“
<code>--</code>	Horizontale Linie
<code>*Zitat*--Vorname Nachname</code>	Gedankenstrich(für Zitate)
<code>[Name:Kommentar]</code>	Kommentar im Fließtext
<code><linebreak></code>	Zeilenumbruch
<code>\$Gleichung\$</code>	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

Verwendung von Markdown.

Ausgabeoptionen für Markdown.

Konvertierung von Markdown in weitere Formate.

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

Verwendung von HTML.

Ausgabeoptionen für HTML.

Konvertierung von HTML in weitere Formate.

4.5 Wiki-Formate

Verwendung von Wiki-Formaten. Wikis werden in vielen Unternehmen verwendet um verschiedenste Aufgaben zu erledigen. Zu diesen Aufgaben zählen z.B. die Standardisierung verschiedener Prozesse im Betrieb, wie die Einarbeitung von Mitarbeitern oder das Sammeln von externen Berichten (Pressespiegel) über das Unternehmen. Häufig werden Wikis auch als Wissensdatenbank für unterschiedliche Themen (Dokumentation von bestimmten Prozessen in der IT oder Qualitätsmanagement) benutzt. Die wohl bekannteste Anwendung eines Wikis ist die freie Enzyklopädie Wikipedia.

Ausgabeoptionen für Wiki-Formate. DocOnce unterstützt die drei gebräuchlichsten Wiki-Formate Googlecode wiki, MediaWiki und Creole Wiki. Die Auswahl des gewünschten Wiki-Formats wird durch den `format`-Befehl realisiert:

```
doonce format gwiki *Datei* (Googlecode Wiki)
doonce format mwiki *Datei* (MediaWiki)
doonce format cwiki *Datei* (Creole Wiki)
```

Konvertierung von Wiki-Formaten in weitere Formate. Das ausgegebene Wiki-Dokument kann nicht weiter in andere Formate konvertiert werden. Ein Anwendungszweck ist das erstellte DocOnce-Dokument zusätzlich zur Ausgabe als PDF für die Veröffentlichung in einem Wiki zu konvertieren.

Kapitel 5

Parameter für den `doconce`-Befehl

Generell wird das Umwandeln der `.do.txt`-Datei in das gewünschte Ausgabeformat mit dem `doconce`-Befehl eingeleitet. Durch das Anhängen von zusätzlichen Parameter kann die Ausgabe weiter an die Bedürfnisse des Benutzers angepasst werden.

`--help`

Durch das Anhängen des Parameters `-help` werden alle Optionen von DocOnce ausgegeben.

`format X`

Das Ausgabeformat wird dem Parameter `format` angehängt. X kann hierbei `html`, `latex`, `pdflatex`, `rst`, `sphinx`, `plain`, `gwiki`, `mwiki`, `cwiki`, `pandoc`, `epytex` sein.

`--debug`

Erstellt eine Debug-Datei die alle auftretenden Fehler und Zwischenschritte protokolliert.

`--no_abort`

Führt die Konvertierung auch dann fort, wenn `doconce` Fehler in der Syntax registriert.

`--verbose=...`

Gibt den Fortschritt eines Zwischenschritts aus, wenn er länger als die festgelegte Zeitspanne (in Sekunden) dauert.

`--toc_depth=...`

Gibt die Anzahl der Untergliederungen im Inhaltsverzeichnis an.

`--encoding=...`

Gibt die zuverwendete Codierung der Eingabedatei an (z.B. `utf-8` oder `latin1`)

`--no_title`

Entfernt Autor, Titel und Datum im fertigen Dokument

`--device=...`

Gibt an ob das Dokument für die Ausgabe auf einen Bildschirm oder zum Ausdruck optimiert werden soll. Die wichtigsten Optionen sind *screen*, für die Ausgabe auf Bildschirmen (verkleinert Seitenränder), und *paper*, für den Ausdruck (Links werden als Fußnote ausgegeben).

`--urlcheck`

Überprüft die im Dokument verwendeten Links auf Webseiten auf Gültigkeit

Die zur Konvertierung dieses Dokuments, in Latex und Markdown, verwendeten Skripte sind im Anhang beigefügt.

Kapitel 6

Fazit

10%

Kapitel 7

Anhang

7.1 Setup-Skript

```
#!/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/krisda/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
#$ echo "remove the mdframe directory (if successful install of mdframed.sty): rm -rf mdframed"

# DocOnce itself
cd srclib
git clone https://github.com/hplgit/doconce.git
if [ -d doconce ]; then cd doconce; sudo python setup.py install; cd ../..; fi
echo "Everything is successfully installed!"

```

7.2 Skripte für die Ausgabe in Latex und Markdown

Latex beziehungsweise PDF.

```
#!/bin/bash
```



```

if [ $# -eq 0 ]
then
  echo "\e[31mBitte geben sie einen Dateinamen ein!\e[0m"
else
  file=$1
  sudo doconce format pdflatex $file --latex_packages=minted,ngerman,parskip --latex_font=helvetica --la
  echo -e "\e[32mLaTeX File erstellt!\e[0m"
  filename="${file%.*}.*"
  echo $filename
  filetex="$filename.tex"
  echo -e "\e[32mBeginne PDF-Erstellung\e[0m"
  sudo pdflatex -interaction=batchmode -shell-escape $filetex
  echo -e "\e[32mSchritt1: pdfLatex\e[0m"
  sudo pdflatex -interaction=batchmode -shell-escape $filetex
  echo -e "\e[32mSchritt2: pdfLatex\e[0m"
  sudo pdflatex -interaction=batchmode -shell-escape $filetex
  echo -e "\e[32mSchritt3: makeindex\e[0m"
  sudo makeindex $filetex
  echo -e "\e[32mSchritt4: bibtex\e[0m"
  sudo bibtex $filetex
  echo -e "\e[32mSchritt5: pdfLatex\e[0m"
  sudo pdflatex -interaction=batchmode -shell-escape $filetex
  echo -e "\e[32mPDF erstellt!\e[0m"
  rm -f "$filename.aux"
  rm -f "$filename.idx"
  rm -f "$filename.ilg"
  rm -f "$filename.ind"
  rm -f "$filename.out"
  rm -f "$filename.toc"
fi

```

Markdown beziehungsweise Github-MD.

```

#!/bin/bash

if [ $# -eq 0 ]
then
  echo "\e[31mBitte geben sie einen Dateinamen ein!\e[0m"
else
  file=$1
  sudo doconce format pandoc $file --no_abort --github_md
  filename="${file%.*}.*"
  mv $filename.md README.md
  echo -e "\e[32mREADME.md erstellt!\e[0m"
fi

```