A practical guide to DocOnce

Kristian Gregorius Hustad (`krihus@ifi.uio.no`)
Centre for Computing in Science Education, University of Oslo

Jul 4, 2017

DRAFT

Contents

This guide attempts to give an introduction to DocOnce by example. The reader is assumed to be familiar with LaTeX and basic terminal usage.

0.1 Preliminaries

0.1.1 What is DocOnce?

> DocOnce is a modestly tagged (Markdown-like) markup language targeting scientific reports, software documentation, books, blog posts, and slides involving much math and code in the text. From DocOnce source you can generate LaTeX, Sphinx, HTML, Jupyter notebooks, Markdown, and other formats. This means that you from a single source can get the most up-to-date publishing technologies for paper, tablets, and phones.

kgh 1: Elaborate

0.1.2 Installation

DocOnce works best with Linux, but Mac OS X is also supported[1]. (Windows use is discouraged although DocOnce can be installed on Windows via conda.)

The easiest way to install DocOnce is to clone the repo from GitHub and perform an installation via pip. Note that the git repository is quite large (around 600 MB).

Initial installation.

```
git clone https://github.com/hplgit/doconce
cd doconce
sudo pip install --upgrade .
```

Subsequent updates. From the doconce repository, run

```
git pull
sudo pip install --upgrade .
```

0.2 Getting started

0.2.1 What does a minimal document look like?

```
TITLE: A Treatise on Something
AUTHOR: John Doe Email: john@doe.com at Doe Corp. & The Society of Ficticious People
AUTHOR: Jane Doe Email: jane@doe.com at The Society of Ficticious People
DATE: today

It is known that something exists.
```

A document should start with specifying the title, author(s) and date of the document. Each author is specified on a separate line on the following form:

```
AUTHOR: <name> [Email: <email_address>] [at <institution> [& <other institution>]*]
```

Notice.

Words enclosed by `<` and `>` should be replaced by an actual value. Words enclosed in `[` and `]` are optional, and `*` means the prepending expression can be repeated 0 or more times. For instance, it is used above to specify that an author can have more than two institutions.

The date can be set to `today`, in which case Doconce will replace it with the current date, or any other text string.

[1]DocOnce uses a variety of tools for some very specific tasks. Some of these may not be available for Mac, but for most use cases, this is not a problem.

```
DATE: The end of days
```

0.2.2 Headings

Headings are written on a separate line surrounded by 3, 5, 7 or 9 `=` signs on either side. The larger headings have more `=` signs.

```
========= A chapter =========
======= A section =======
===== A subsection =====
=== A subsubsection ===
__ Inline paragraph heading __
```

0.2.3 Code

Code can either be written inline or in separate blocks. For instance, we can write `print "test"` or

```
print "test"
```

A longer example:

```
The expression ‘a/b‘, where ‘a‘ and ‘b‘ are both integers, is evaluated differently in Python 2 and P

!bc pycod
c = 3/2
# Python 2: c is 1
# Python 3: c is 1.5
!ec
```

This renders as: —–

The expression `a/b`, where `a` and `b` are both integers, is evaluated differently in Python 2 and Python 3. While Python 2 uses the common rules of integer division (flooring the result), such that the set of integers are closed under division, Python 3 will return a (decimal) floating point number, if `a % b != 0` (the remainder is non-zero).

```
c = 3/2
# Python 2: c is 1
# Python 3: c is 1.5
```

—–

0.2.4 Formulae

Similar to LaTeX, both inline mathematics and blocks of mathematics are supported. The formula syntax is indentical to LaTeX.

Inline mathematics must be enclosed in `$` signs.

Example:

```
We can choose $n = 2^{16}$
```

renders as

We can choose $n = 2^{16}$

More complicated equations are usually written in a math block. DocOnce supports a selection of the math environments from LaTeX, including `equation` and `align`.

```
!bt
\begin{align}
a = 2
\end{align}
!et
```

$$a = 2 \tag{1}$$

0.3 A complete document

```
TITLE: A Primer on Primes
AUTHOR: Kristian Gregorius Hustad Email: krihus@ifi.uio.no at Centre for Computing in Science Educati
DATE: today

!bwarning
This document is intended to demonstrate DocOnce features.
Consult other texts for an introduction to prime numbers.
!ewarning

======= Theoretical foundations =======
===== Divisibility =====
An integer $b$ is divisible by $a$ if $\frac{b}{a}$ leaves no remainder. In that case, we say that $a
!bt
\begin{equation}
a \mid b \iff b = qa + r \text{ such that } a, b, q \in \mathbb{Z} \text{ and $r=0$}
\end{equation}
!et

In Python, we can obtain the remainder by employing the modulus operator, ‘%‘.
!bc pycod
# b  = q*a + r
# 14 = 4*3 + 2
b = 14
a = 3
b % a # => 2
```

```
!ec

===== Coprimality =====
Two integers $a$ and $b$ are *coprime* if they have no common factors besides 1.

===== Greatest common divisors =====
The greatest common divisor of two integers $a$ and $b$ is the largest integer which satisfies
!bt
\begin{align}
c = \mathrm{gcd}(a, b) \iff c \mid a \; \wedge \; c \mid b
\end{align}
!et

__Definition:__
!bt
\begin{align}
\text{$a$ and $b$ are coprime} \iff \mathrm{gcd}(a, b) = 1
\end{align}
!et

===== Euclid's algorithm =====
!bc pypro
def gcd(a, b):
    """Returns the greatest common divisor of two positive integers"""
    if b == 0:
        return a
    else:
        r = a % b
        return gcd(b, r)
!ec


======= Primality =======
__Definition:__
A prime is a positive integer divisible only by itself and 1.

!bt
\begin{align}
\text{$p$ is prime} \iff \nexists q \, (q \mid p, \; q \in (\mathbb{N}\setminus\{1, p\}))
\end{align}
!et


!bnotice
Any two primes $p$ and $q$ are coprime. Thus we can define the set of primes as the set where any tw

!bt
\begin{align}
%p, q \in P \iff
P = \{ p_1, p_2, \dots, p_k \in \mathbb{N} \; \mid \; \mathrm{gcd}(p_m, p_n) = 1, m \neq n  \}
\end{align}
!et
!enotice

======= Determining primality =======

===== A naive algorithm =====
label{subsec:naive_alg}
We can check whether a number, $n$, is prime by testing divisibility for all primes smaller than $n$.
@@@CODE src/prime.py

===== Sieve of Eratosthenes =====
Although the program in ref{subsec:naive_alg} does work, it is very inefficient. The Sieve of Eratost
```

```
To optimize even further, we shall make use of a few simple observations:
o Any composite number, $c$, has a factor $f \leq \sqrt{c}$
o If we cross off every multiple of every prime, we would cross off all non-squares twice. For insta

@@@CODE src/sieve.py
```

0.4 Best practices

0.4.1 Version control

As with any type of code, DocOnce documents should be kept under version control. In this guide we only consider Git, but the principles should translate to other types of version control systems as well.

First and foremost, only source files, viz. files that cannot be regenerated automatically should be added to Git. All documents should be reproduced by running some sort of makefile. The author prefers GNU Makefiles, but shell scripts can also be used.

When writing long lines, it is recommended to configure the text editor to display them wrapped, while newlines are only added to the document for paragraphs etc. This will make the git history much more legible since adding a single word at the start of a paragraph doesn't cause changes to the entire paragraph.

kgh 2: `.gitignore` for DocOnce