



# Clean Code Documentation

*Release 0+untagged.45.g7732381.dirty*

**Zebedee Nicholls, Felix Kramer, Nicholas Hiron**

Nov 16, 2018



---

## Contents

---

<b>1</b>	<b>Version control</b>	<b>1</b>
1.1	Git . . . . .	1
<b>2</b>	<b>Text editors</b>	<b>3</b>
2.1	SublimeText3 . . . . .	3
<b>3</b>	<b>Reading list</b>	<b>5</b>
3.1	Overview . . . . .	5
<b>4</b>	<b>Scientific computing</b>	<b>7</b>
4.1	Zeb’s list of important tools . . . . .	7
<b>5</b>	<b>Development</b>	<b>9</b>
5.1	Getting setup . . . . .	9
5.2	Building the docs . . . . .	9
<b>6</b>	<b>License</b>	<b>11</b>
<b>7</b>	<b>Index</b>	<b>13</b>



Here we provide tips and tricks on different version control systems.

## 1.1 Git

### 1.1.1 Overview

A bunch of tips and tricks about how to use Git.

1. StackOverflow is your friend, use it!
2. Rebasing [add link to Atlassian explanation] before you merge into master keeps your commit history linear. This makes it much easier to retrace your steps if you need.



Here we provide tips and tricks on different text editors.

## 2.1 SublimeText3

### 2.1.1 Overview

A bunch of tips and tricks about how to use Sublime Text 3.

### 2.1.2 Command line alias

#### Mac

Add alias in `~/.bashrc`

```
alias subl="/Applications/Sublime\ Text.app/Contents/SharedSupport/bin/subl"
```

### 2.1.3 Setting up

Open up Sublime Text

- **install package control**

```
- cmd + shift + p, install package control
```

#### Zeb's recommended packages for climate science

- Fortran
- Git

- GitGutter
- SublimeLinter-pylint
- OmniMarkupPreview

## 2.1.4 Package guide

### MarkdownPreview

Super useful package to let you preview markdown documents as you right. There's heaps to read about this at <https://facelessuser.github.io/MarkdownPreview/usage/>. I'd be super interested if anyone has found a plugin which does live preview well (rather than just preview on demand).

### SublimeLinter-pylint

#### Installation

- follow the [online instructions](#) to install the **Pylint** python package (either python2 or python3) with *pip* [add link to pip guide]
- then do `package control: install package -> SublimeLinter`
- then do `package control: install package -> SublimeLinter-pylint`
- then fix the path settings for this package, SublimeText -> Preferences -> Package Settings (follow [this guide](#))
  - for example

```
{ "paths": {  
  "linux": [],  
  "osx": [  
    "/usr/local/bin/pylint"  
  ],  
  "windows": []  
}}
```

### 3.1 Overview

This document contains our reading list. It is a series of resources we have come across that we have now reviewed and put here.

#### The list

- *Reading list*
  - *Overview*
    - \* *Clean Code by Robert C. Martin*
    - \* *Good Enough Practices in Scientific Computing by Greg Wilson et al.*
    - \* *Refactoring by Martin Fowler*

#### 3.1.1 Clean Code by Robert C. Martin

- **availability**
  - Book depository
  - Free pdf (legality unknown)
- **recommendation**
  - essential (all)
- **price**
  - free (if you can get hold of pdf, otherwise ~\$50AUD)
- **length**

- time: > 10 hrs
- pages: 450
- **ease of use**
  - moderate

### 3.1.2 Good Enough Practices in Scientific Computing by Greg Wilson et al.

- **availability**
  - Free pdf
- **recommendation**
  - high (all)
- **price**
  - free
- **length**
  - time: 30 mins
  - pages: 20
- **ease of use**
  - moderate

#### Other comments

There are more thoughts on this subject in *scientific computing*.

### 3.1.3 Refactoring by Martin Fowler

- **availability**
  - Book depository
  - Free pdf (legality unknown)
- **recommendation**
  - high (Ch. 1, 2, 3, 4, 13, 14, 15)
- **price**
  - free (if you can get hold of pdf, otherwise ~\$65AUD)
- **length**
  - time: > 10 hrs
  - pages: 300
- **ease of use**
  - moderate

## 4.1 Zeb's list of important tools

### 4.1.1 Development tools

This list of development tools is what Zeb relies on to develop scientific software reliably and reproducibly. Links are included with each of these tools to useful starting points.

- Version control: [Git](#)
- Automating repetitive tasks: [Make](#)
- Virtual environments: [Conda virtual environments](#)
  - note the common gotcha that `source activate` has now changed to `conda activate`
  - we use conda instead of pure pip environments because they help us deal with more complicated dependencies: if you want to learn more about pip and pip virtual environments, check out:
    - \* [this introduction](#)
    - \* [this longer piece](#) which explains the details
- Tests: many available frameworks, here's a [link to testing intro that Zeb likes](#)
  - most of the time, a blend of [pytest](#) and the inbuilt Python testing capabilities works
- Continuous integration (CI)
  - [Travis CI](#) is a good choice but there are a number of good providers
- Jupyter Notebooks
  - simply installing `jupyter` (`conda install jupyter`) in your virtual environment is as good a way as any
- [Sphinx](#)

## 4.1.2 Other tools

Other tools also exist which are useful but not necessarily essential and not necessarily related to development. Here we provide a list of these along with useful resources.

- [Regular expressions](#)
  - [regex101.com](#) to help write and check regular expressions, make sure the language is set to Python to make your life easy!

### 5.1 Getting setup

To get setup as a developer, we recommend the following steps (if any of these tools are unfamiliar, please see the resources in *this development tools section*):

1. Install `make` and ensure that you have Python3 installed
2. Make sure the tests pass by running `make test`

### 5.2 Building the docs

The docs are built using `make docs`. This command will build the docs, including all compilation of files from `yaml` files etc.

Welcome to our shared Git repository. Here we store a bunch of notes, code snippets, makefiles etc. from our own experience getting into the world of computers, code and software engineering.



## CHAPTER 6

---

### License

---

This collection is licensed under a [Creative Commons CC0 license](#), unless noted otherwise for specific parts:

The person who associated a work with this deed has dedicated the work to the public domain by waiving all of his or her rights to the work worldwide under copyright law, including all related and neighboring rights, to the extent allowed by law. You can copy, modify, distribute and perform the work, even for commercial purposes, all without asking permission.



## CHAPTER 7

---

### Index

---

- genindex
- modindex