



Sublime Text 3 Heaven

2017

2 min read • 425 words

I decided to revisit my editor configuration the other night, and experimented with every possible editor I could think of / imagine. I heavily configured vim (neovim), PyCharm, Eclipse, Emacs (Spacemacs), VSCode, Atom, Textual, and more. I knew I was going to stay put with my choice of Sublime Text 3 (which I have been using for 5+ years), but it's nice to have validation.

So, I decided to rebuild that configuration from scratch as well. I ended up with a very happy setup that I wanted to share with you. Here's [a screencast](#) of myself writing a little bit of code and pushing it to GitHub with this setup.

Sublime UI Theme: Material

FOLDERS

▼ pipenv

▶ .cache

▶ .venv

▶ docs

▶ pipenv

▶ pipenv.egg-info

▶ test

▶ tests

* .env

◆ .gitignore

Y .travis.yml

HISTORY.txt

© LICENSE

Makefile

* Pipfile

* Pipfile.lock

README.rst

setup.py

t.py

test_pipenv.py

cli.py

24 from __version__ import __version__

25 from . import pep508checker

26 from .environments import PIPENV_COLORBLIND, PIPENV_SHELL_COMPAT,

27

28 try:

29 from HTMLParser import HTMLParser

30 except ImportError:

31 from html.parser import HTMLParser

32

33 # Backport required for earlier versions of Python.

34 if sys.version_info < (3, 3):

35 from backports.shutil_get_terminal_size import get_terminal_size

36 else:

37 from shutil import get_terminal_size

38

39 # -----

40 # | . \<_> _ _ _ | _ _ _> _ _ _

41 # | _ _ / | | | . \ | _ _> | ' | | | |

42 # | _ | | _ | | _ / | _ _ _> | _ | | _ _ /

43 # _ _ _ _ | _ |

44

45 # Enable shell completion.

46 click_completion.init()

47

48 # Disable colors, for the soulless.

49 if PIPENV_COLORBLIND:

50 crayons.disable()

51

52 # Disable warnings for Python 2.6.

53 requests.packages.urllib3.disable_warnings(InsecureRequestWarning)

54

55 project = Project()

56

57

58 def ensure_latest_pip():

59 """Updates pip to the latest version."""

60

61 # Ensure that pip is installed.

62 c = delegator.run('{0} install pip'.format(which_pip()))

63

64 # Check if version is out of date.

65 if 'however' in c.err:

66 # If version is out of date, update.

67 click.echo(crayons.yellow('Pip is out of date... updating'))

68 c = delegator.run('{0} install pip --upgrade'.format(which_pip()))

69 click.echo(crayons.blue(c.out))

70

71

72 def ensure_pipfile(validate=True):

73 """Creates a Pipfile for the project, if it doesn't exist."""

74

75 # Assert Pipfile exists.

76 if not project.pipfile_exists:

77

78 click.echo(crayons.yellow('Creating a Pipfile for this project'))

79

80 # Create the pipfile if it doesn't exist.

81 project.create_pipfile()

82

83 # Validate the Pipfile's contents.

84 if validate:

85 # Ensure that Pipfile is using proper casing.

86 ensure_proper_casing()

87

88

89

On master in pipenv, master*, Comparing against : HEAD, Line 54, Column 1

Sublime Text Extensions:

- **Anaconda** — fantastic Python "IDE" support for Sublime Text. Just works, does everything you'd want it to do, including code completion and PEP8 checking.
- **Color Highlighter** — highlights colors present in code as the value provided (great for css).
- **Emmet** —fantastic HTML shortcut utility.
- **Package Control** — (obviously)
- **SideBarEnhancements** —enhances the sidebar context menu options. Easily create new files and folders, etc.
- **Themr** — easily switch between themes.

Version Control:

- **GitGutter** — display git diff information in the gutter of Sublime Text — extremely useful! Keeps track of added/removed lines.
- **GitSavvy** — very useful tool for committing/pushing with Git right from Sublime!
- **GitStatusBar** — shows git repo status in the bottom bar of Sublime Text.

Syntax Packages:

- **Tomorrow Night Italics Color Scheme** — italics for code comments, for [Operator Mono](#).
- **fish-shell** — syntax highlighting for fish scripts.
- **Jinja2** — syntax highlighting and snippets for jinja2 templates.
- **RestructuredText Improved** —syntax highlighting for RST files.
- **requirements.txt** — syntax highlighting for requirements.txt files.
- **TOML** — syntax highlighting for TOML.
- **VimL** — syntax highlighting for VimL.

Fun Toys:

- **ASCII Decorator** — right click on text, and turn it into ASCII art.
- **Glue** — terminal instance within Sublime.
- **GitAutoCommit** — a nifty little plugin that lets you set certian repos to automatically commit on save (useful for notes, etc).
- **SublimeXiki** — get the power of Xiki (shown in the screencast above, at the end) in Sublime!

User Key Bindings

```
[{ "keys": ["super+2"], "command": "next_bookmark" }, { "keys": ["super+1"], "command":
```

User Settings

```
{"auto_complete": false, "close_windows_when_empty": true, "color_scheme": "Packages/User
```

That's it! Enjoy :)

Generated from kennethreitz.org • 2025