

Sublime Text 2

Environment Setup

This is how I setup Sublime Text. Generally I use Sublime Text for editing Ruby, Rails, HTML, Javascript, Markdown and other text files. Frequently I work on remote/cloud servers where mouse responsiveness is low, so I like to have Vintage Mode (e.g. vi keyboard controls available) ready but not the default. My setup enables Vintage Mode but starts in normal Insert Mode.

If you know of a helpful package or configuration option that I don't have, please let me know.

Vintage Mode and User Preferences

In user preferences add the following...

```
{
  "auto_commit_commit_on_tab": true,
  "font_size": 15.0,
  "ignored_packages":
  [
  ],
  "tab_size": 2,
  "translate_tabs_to_spaces": true,
  "trim_trailing_white_space_on_save": true,
  "highlight_line": true,
  "rulers":
  [
    79
  ],
  "vintage_start_in_command_mode": false
}
```

Change the current tab color

Use "Preferences" > "Browse Packages...", then go into the User folder and create a new file named "Default.sublime-theme". Put the following in the file:

```
[
  {"class": "tab_control", "attributes": ["selected", "file_medium_dark"], "tint_modifier": [255, 255, 255, 80]}
]
```

Package Manager

<https://packagecontrol.io/>

Add Favorite Packages

command-shift-P (ctrl-shift-P) then pick "Package Control: Install Package", then enter the

package name. Install the following packages:

General Tools:

Git

(skip: Cross - it hangs sublime text 2 on the mac when selecting text and your mouse goes outside of the edit window)

Markdown

This article was helpful:

<https://archerimagine.wordpress.com/2014/05/29/complete-guide-to-setup-sublime-text-2-as-a-best-markdown-editor/>

MarkdownEditing

See it's page for special instructions about disabling Sublime's built-in Markdown editor.

1. Preferences -> Settings User
2. Add "Markdown" to the ignored packages list.

Set *.md specific controls:

1. open a markdown file
2. Preferences -> Settings-More -> Syntax Specific - User
3. Enter the following settings (the Title of the file should be: Markdown (standard)-sublime.settings)

```
{
  "spellcheck": true,
  "color_scheme": "Packages/MarkdownEditing/MarkdownEditor-Dark.tmTheme"
}
```

Markdown Preview

usage Ctrl+Shift+P MarkdownPreview and convert to some format
after the first conversion, saving the file will update the converted file

Firefox: LiveReload

install the LiveReload add-on. This will update the markdown preview everytime it is saved.

(tried LiveReload <https://github.com/dz0ny/LiveReload-sublimetext2> but it didn't work 100% of the time. all it saves is clicking refresh in the browser.)

HTML Development:

HTML5

TidyHTML5 (to use, type "TidyHTML" in command palette)

View in Browser

Configuration: modify "View in Browser.sublime-settings" set "selectedBrowser": "chrome" and make sure the command-line for chrome is set correctly (you will have to modify it on windows platforms).

Javascript Development:

Sublime-JSHint (not JSHint)

requires JSHint installed on the system <http://www.jshint.com/install/> (e.g. `npm install -g jshint`)

JSON

JSLint

PrettyJSON `cmd+ctrl+j` or `ctrl+alt+j`

Ruby Development:

Sublime Text 2 Ruby Test

see details at <https://github.com/maltize/sublime-text-2-ruby-tests>

SublimeREPL

to use ruby, you need to install the pry gem

[note: I need to try <https://github.com/JoshCheek/sublime-text-2-seeing-is-believing> sometime.]

Elixir

Elixir package

install from package manager. <https://github.com/rodrigoflores/elixir-sublimetext>

Custom Syntax Highlighting

AAAPackageDev

install from package manager. <https://github.com/SublimeText/AAAPackageDev>

see below for an example script to highlight .log files.

DashDoc (mac only, provides integration with Dash)

install from package manager. <https://github.com/farcaller/DashDoc>

Add Shell Access (Mac)

open Automator and add an new Service

add an action "Run Shell Script"

set the script to: `/Applications/Sublime\ Text\ 2.app/Contents/SharedSupport/bin/subl -n $@`

set "Service receives selected" to "files or folders"

set "Pass input" to "as arguments"

save as "Open-in-Sublime"

this will appear on file or folder right-click menu under the "services" sub-menu!

Add to Terminal (Mac)

`sudo ln -s "/Applications/Sublime Text 2.app/Contents/SharedSupport/bin/subl" /bin/subl`

Keyboard

For a cheat sheet see: <https://gist.github.com/4686573>

Example: Custom Language Highlighting

First you need the AAPackageDev package installed.

To create a definition, use Tools > Packages > Package Control > New Syntax Definition. Save the file in your "Sublime Text 2\Packages\User" folder with the extension ".YAML-tmLanguage".

The following example will highlight log files. It colors lines that start with dates; the word ERROR; Exception Messages; and stack trace lines with certain keywords (Medseek in this example)

```
# [PackageDev] target_format: plist, ext: tmLanguage
---
name: Log File
scopeName: text.log_file
fileTypes: ["log"]
uuid: f237dd9d-3dfd-4b2e-a770-4bb3097e7ca2

patterns:
- match: "[[:digit:]]{4}-[[:digit:]]{2}-[[:digit:]]{2} [[:digit:]]{2}:[[:digit:]]{2}:[[:digit:]]{2},[[:digit:]]{3}.*"
  name: keyword.ssraw
  comment: "Timestamped lines, usually the start of a log message"
- match: "ERROR"
  name: invalid.illegal
  comment: "Error keyword"
- match: "MessageText: (.*)"
  name: invalid.deprecated
  captures:
    "1":
      name: invalid.illegal
      comment: "Exception message"
- match: "at Medseek(.*)"
  name: invalid.deprecated
  comment: "medseek stack trace"
```

The common names of scopes are available at

http://manual.macromates.com/en/language_grammars#example_grammar in section 12.4 Naming Conventions.

Build with ctrl+b or F7. If there are no errors, Sublime will start using the updated syntax definition immediately.