sublime

kevinluo

Contents

目錄

1 basic information

2 website

- main page
- package get
- help
- tutorial

3 install components

1. portable sublime
2. package control install
3. sublimeGit

3.1 package control install

portable sublime 缺省沒有安裝，可 2 個方法安裝

1. tools-package control install

   裝完此菜單消失，preferences 出現->package setting; ->package control

2. 2 種另外的方法，自動和手動

   頁面查看 2 種另外的方法 在頁面查看 2 種另外的方法，自動和手動

3.1.1 如果想要刪除插件，

Ctrl+Shift+P調出命令面板，輸入remove，調出Remove Package選項并回車，選擇要刪除的插件即可，當

3.1.2 用 Package Control 安裝插件的方法

按下Ctrl+Shift+P調出命令面板
輸入install 調出 Install Package 選項并回車，然後在列表中選中要安裝的插件。
不爽的是，有的網絡環境可能會不允許訪問陌生的網絡環境從而設置一道防火墙，而Sublime Text3貌似無

3.2 sublimegit package install

tools-command palette ctl+shift+p
pci package control install
等待載入package information,然後在命令行輸入sublimeGit
安裝完後，在
preference->package Settings-> 此處出現安裝的sublimeGit
同時在
preference->package settings-> package control -> user setting 中可以看到已經增加選項

3.2.1 sumlimeGit usage 用法

https://sublimegit.readthedocs.io/en/latest/

- 【下面的有些問題，看 readthedocs 就行了】

full tutorial, go to

https://sublimegit.readthedocs.io/en/latest/tutorial.html

how to get set up

4 有用的插件

超級文本編輯器 Sublime Text3

4.1 ConvertToUTF8

比上面的那個要方便，直接在菜單欄中可以轉了，專爲中文設計，媽媽再也不通擔心中文亂碼問題了

4.2 markdown,也可以用簡書直接編輯查看

4.2.1 MarkdownEditing

4.2.2 OmniMarkupPreviewer

4.2.2.1 TOC render Preview 支持 右鍵 menu preview markdown in browser, export/copy markdown as html

1. 如果你發現它不支持markdown目錄的預覽生成，那麼不是它不行，是你沒配置。
   當然首先是裝markdwon TOC插件

2. 復制Preferences -> Package Settings -> OmniMarkupPreviewer -> Settings - Default 中的內容到Se

3. 并在 // MarkdownRenderer options區域，即

“renderer_options-MarkdownRenderer”：中添加"toc"，代碼如下
"extensions": ["tables", "strikeout", "fenced_code", "codehilite", "toc"]

4. 然後通過Ctrl+Alt+O快捷鍵生成HTML預覽，或者Ctrl+Alt+X導出。

4.2.2.2 OmniMarkupPreviewer 中支持 LaTeX 公式顯示： 1. 設置。

公式的渲染使用了MathJax庫，所以需要在OmniMarkupPreviewer的設置中，將"mathjax_enabled"設置爲

2. 可能是網速的原因，MathJax 庫下載很慢，所以可以選擇手動安裝。

[下載MathJax](https://github.com/downloads/timonwong/OmniMarkupPreviewer/mathjax.zip)

然後解壓到下面的目錄裏：Sublime Text 2\\003work\002memo\001software\001install\OmniMarkupP
之後在目錄“Sublime Text 2\\003work\002memo\001software\001install\OmniMarkupPreviewer”中
測試，輸入下面內容：
This expression
$\sqrt{3x-1}+(1+x)^2$ is an example of a $\LaTeX$ inline equation.he Lorenz Equations:
$$\begin{aligned}\dot{x} & = \sigma(y-x) \\\dot{y} & = \rho x - y - xz \\\dot{z} & = -\beta z +

在 Sublime Text 3 中使用命令：

Ctrl+Alt+O：在瀏覽器中預覽
Ctrl+Alt+X：輸出爲HTML文件
Ctrl+Alt+C：復制爲HTML文件

4.2.3 Markdown Extended

4.2.4 MarkdownLivePreview [alt+m 在 sublime 啓動并列窗口，實時查看結果]

4.2.5 MarkdownTOC

Sublime Text 3 plugin for generating a Table of Contents (TOC) in a Markdown document.

- Features
- Usage

4.3 reStructuredText Improved

Headings and terms (from definition lists) are available as symbols, so you can use CTRL-R to jump to them.

4.4 Restructured Text (RST) Snippets

裝完後 preferences-package setting 中的名字爲，sumlime-rst-completion

Restructured Text (RST) Snippets

- 用法鏈接
  - 本地 README
  - Git-README
- 快捷鍵
  - magic table
    1. grid table ctrl+t，enter
       1. keep the column width fixed, ctrl+t，r (super+shift+t，r in Mac)

    2. merge simple cells: `ctrl+t, down ctrl+t, up`
  2. simple table `ctrl+t, s`
- Adjust header level: `ctrl+-` | `ctrl+keypad-`
- 補齊: `tab`
- jump between headers: `alt+down` | `alt+up`
- add new footnote: `alt+shift+f`
- go back to the reference with `shift+up`

-usage snippets

shortcut	result
`h1 h2 h3`	Header level 1 Header level 2 Header level 3
`e`	emphasis
`se`	strong emphasis (bold)
`lit`	literal text
`literal`	(inline code)
`list listn listan def code source img fig table link linki fn`	unordered list ordered list auto ordered list te
`cite`	footnote or cite
`quote`	Quotation (`epigraph`)

接上:

shortcut

`attention caution danger error hint important note tip warning`

-編譯 Python 項目文檔

Python 的項目文檔，大都基于 reStructuredText 撰寫，Sphinx 發布，如何在 Sublime 中，通過按 Ctrl + B 直接編譯工程呢？很簡單，點擊 Tools –> Build System –> New Build System，輸入

```
{
    "shell_cmd": "make html"
}
```

保存，打開你工程的 Makefile 文件，然後按 Ctrl + Shift + B 選擇你剛才保存的那個名字，就可以自動編譯成 html 文檔了。

4.5 Anaconda - python completion

Anaconda 强大的補全工具, 還能實時看文檔, 轉到定義, 自動格式化代碼

doc

http://damnwidget.github.io/anaconda/

4.6 HexViewer

hex 查看模式

4.7 latex tools

git latextools 項目

DOC on readthedocs

- 配套
  1. sumatrapdf
     sumatraPdf 網址 gitREP sumatrpdf

4.8 Path Tools

Open the Command Palette (Ctrl/Super + Shift + P) and enter one of the following:

```
Insert File Path
Insert File Directory
Insert File Name
Insert Path Relative to Project
Insert Directory Relative to Project
Copy File Path
Copy File Directory
Copy File Name
```

4.9 timenow

插入日期時間

4.10 SideBarEnhancements

*.html 文件，按 f5 自動啓動 chrome 瀏覽設置

```
[
    { "keys": ["ctrl+shift+c"], "command": "copy_path" },
    //chrome
    { "keys": ["f5"], "command": "side_bar_files_open_with",
        "args": {
            "paths": [],
           "application": "C:\\Program Files\\Google\\Chrome\\Application\\chrome.exe",
            "extensions":".html"//匹配任何文件類型
            ,
        }
    },

]
```

4.11 ChineseOpenConvert

window install:

```
git clone -b st3 https://github.com/rexdf/SublimeChineseConvert.git "%APPDATA%\Sublime Text 3\\
```

4.12 DictionaryAutoComplete

DictionaryAutoComplete

注意：

發生了不取詞的問題。原因是：user setting 文件中 “encoding”: “ISO-8859-1”, 不能爲空。

觸發取詞改成 f1 鍵，輸入時需要小寫

手動安裝 cndict, 因爲 DictionaryAutoComplet 安裝時，不能改成 f1 鍵

到這裏下載

設置

- command 設置自動完成切換，總的和 package 內 Two commands are added in the Command Palette (Ctrl+Shift+P):
  - Dictionary Auto Complete: Toggle : Activate/deactivate this plug-in.
  - Auto Complete: Toggle : Activate/deactivate the sublime auto-completion.
- 手動跳出來，Ctrl + Space

  Just type Ctrl + Space to show auto-completion,
- 自動跳出詞語

  allow auto-complete to always show suggestions by changing your **‘Settings - User’** for example like this: “auto_complete_selector”: “text, comment, string”
- 詞庫

FrequencyDictionaries on github

- dictionary :

A path to alternative dictionary to use in place of the default dictionary used for spell-checking. This allows you for example to use a frequency dictionary that will show in first place the most used words.

in preference-> packagesettin->autodictionarycomplete->user-setting:

```
"languages": {
  "en_US": {
    // this is the encoding for the default ST dictionary
    "encoding": "",
    // you can overwrite here the default dictionary
    // for example by putting
    // "dictionary" : "Packages/User/frequency_en.txt",
    "dictionary" : "Packages/User/kl-dict/large_en.txt"
  },
```

4.13 Chinese-English Bilingual Dictionary

Chinese-English Bilingual Dictionary

- Usage:

Ctrl+Alt+Y: 有道詞典 Youdao

Ctrl+Alt+C: 金山詞典 Jinshan

Select a word you want to translate, use corresponding key-mappings, then depending on the configuration “format”,

- There are three possible parameter for format:
  - “popup”:

a tooltips pop-up will show up, it will not be embeded in but just float on the view. close it by ESC or Crtl+Shift+D

- "phantom":

a block of phantom will show up just below the line: Using Crtl+Shift+D to Erase all Phantoms

- "pannel":

This is the classical option, an Output Pannel will show up from the bottom.

5 其它 package 匯總

5.1 Markdown Numbered Headers

like markdown TOC with additional feature of NUmber Heading

5.2 Insert Nums:

https://packagecontrol.io/packages/Insert%20Nums

inserts (consecutive) numbers across multiple selections or modifies the selections' contents with expressions. Huge configurability.

```
Ctrl+Alt+N
<start>:<step>

The complete syntax is: <start>:<step>~<format>::<expr>@<stopexpr><reverse>
numbers: [<start>][:<step>][~<format>][::<expr>][@<stopexpr>][!]
alpha:   <start>[:<step>][~<format>][@<stopexpr>][!]
```

for the detailed syntax specification, see: format_syntax.txt.

essentially Python's "str.format" syntax

```
format        ::=  [[fill]align][sign][#][0][width][.precision][type]
fill          ::=  <a character other than '}'>
align         ::=  "<" | ">" | "=" | "^"
sign          ::=  "+" | "-" | " "
width         ::=  integer
precision     ::=  integer
type       ::= "b" | "c" | "d" | "e" | "E" | "f" | "F" | "g" | "G" | "n" | "o" | "x" | "X" | "%"
```

Detailed syntax definition:

format_syntax.txt

- **start**
  - *with numbers* (optional): A

    [decimalinteger](http://docs.python.org/2.6/reference/lexical_analysis.html#grammar-token-decimalinteger)

    or

    [floatnumber](http://docs.python.org/2.6/reference/lexical_analysis.html#grammar-token-floatnumber)

    according to Python's syntax specifications with an optional leading sign (- or +). Default: `1`

  - *with alphabet* (required): A sequence of either lower- or uppercase ASCII characters from the alphabet (`a` to `z` and `A` to `Z`).

- **step** (optional)
  - *with numbers*: A

    [decimalinteger](http://docs.python.org/2.6/reference/lexical_analysis.html#grammar-token-decimalinteger)

    or

    [floatnumber](http://docs.python.org/2.6/reference/lexical_analysis.html#grammar-token-floatnumber)

    according to Python's syntax specifications with an optional leading sign (`-` or `+`). Default: `1`
  - *with alphabet*: A

    [decimalinteger](http://docs.python.org/2.6/reference/lexical_analysis.html#grammar-token-decimalinteger)

    with an optional leading sign (`-` or `+`).

- **format** (optional)
  - *with numbers*: A format string in Python's [Format Specific

    Mini-Language](http://docs.python.org/2.6/library/string.html#format-specification-mini-language) (with small and unimportant adjustments for allowed types).
  - *with alphabet*: Similar to *with numbers* but a stripped-down

    version only for strings. This only includes the `[[fill]align][width]` syntax and additionally accepts a `w` character at the end (see above).

- **expr** (optional)
  - *numbers only*: A valid Python expression which modifies the value as you please. If specified, the *format string* is applied afterwards. Here is a list of available variables:

    * `s`: The value of `step` (specified in the format query and defaults to `1`)
    * `n`: The number of selections
    * `i`: Just an integer holding the counter for the iteration; starts at `0` and is increased by `1` in every loop
    * `_`: The current value before the expression (`start + i * step`)
    * `p`: The result of the previously evaluated value (without formatting); `0` for the first value
    * `math`, `random` and `re`: Useful modules that are pre-imported for you

    Note: The return value does not have to be a number type, you can also generate strings, tuples or booleans.

- **stopexpr** (optional)

  A valid Python expression which returns a value that translates to true or false (in a boolean context). Theoretically this can be any value. You can use the same values as in **expr** with addition of the following:

  - `c`: The current evaluated value by the expression (without formatting) or just the same as `_` if there was no expression specified

  This ignores the number of selections which means that you can also have more or less values than selections. Especially useful when generating numbers from a single selection. - If there is more selections than numbers generated when processing the stop expression, all the remaining selections' text will be deleted. - If there is more numbers generated than selections, all further numbers are joining by newlines (`"\n"`) and added to the last selection made. This can be the first selection if there is only one.

- **reverse** (optional)

  Must be ! and results in the regions being filled in reversed order.

5.2.1 Examples:

```
numbers: [<start>][:<step>][~<format>][::<expr>][@<stopexpr>][!]
alpha:   <start>[:<step>][~<format>][@<stopexpr>][!]
```

format= [[fill]align][sign][#][0][width][.precision][type]

1. 傳統法

   ```
   1:1~0>+#04d::_*1@i>=10!
   1:1~0> #04d::_*1@i>=10!
   1:1~k> #04d::_*1@i>=10!

   ~02@p==10 or ~02@_>10 or ~02@i==10

   i|p+3 if i!= 0 else _!

   |re.sub(r' +', ' ', _)

   float加入.

   1:1~0>+#04.2f::_*1@i>=10!
   ```

2. 移位法賦值

   ```
   0~#06x::1<<_@_>10
   ```

3. 字母

   ```
   z:25~w or z:-1~w
   ```

5.3 emmet

html自動補全
ZenCoding
不得不用的一款前端開發方面的插件，Write less，show more.安裝後可直接使用，Tab鍵觸發，Alt+Shi

5.4 VAlign

inspired by alignment, automatically align

5.5 Alignment

代碼對齊，如寫幾個變量，選中這幾行，Ctrl+Alt+A，哇，齊了。

5.6 Ctags

函數跳轉，我的電腦上是 Alt+ 點擊函數名稱，會跳轉到相應的函數

5.7 DocBlockr

注釋插件，生成幽美的注釋。標準的注釋，包括函數名、參數、返回值等，并以多行顯示，省去手動編寫。

5.8 JsFormat

格式化 js 代碼，這個插件很有用，我們有時在網上看到某些效果，想查看是怎麼實現的，但是代碼被壓縮過，很難閱讀，不用怕，用 ST3 打開，按下快捷鍵，即可讓代碼還原，莫非是武林中失傳已久的“還我靚靚拳”。

5.9 Tag

格式化標簽，讓亂七八糟的代碼，瞬間整齊清晰。

5.10 BracketHighlighter

括弧高亮顯示。

5.11 Clipboard History

剪切板歷史，可以保存多個復制信息，按下ctrl+alt+v，可以選擇歷史剪切板。
Goto-CSS-Declaration
跳轉到css文件該class的聲明處，方便修改查看，如圖下所示，注意對應的css文件要同時打開才行。

5.12 SCSS

支持 scss 的語法高亮，裏面附帶了好多 CSS Snippet，無論現用或者改造成，都可節省不少時間。還有很多插件，jquery 語法提示，jsHint 等等。

5.13 Sublime Linter

這個插件幫你找到代碼中的錯誤。它支持很多語言：PHP, Python, Java, CoffeScript, CSS, HTML, JavaScript, Perl, PHP, Python, Ruby, XML 等。Javascript 需要安裝 Node.js 引擎，其他配置詳見項目主頁。强烈推薦安裝。

5.14 Sublime CodeIntel

Sublime CodeIntel 是我最喜歡的插件，它提供了很多 IDE 提供的功能，例如代碼自動補齊，快速跳轉到變量定義，在狀態欄顯示函數快捷信息等。它支持的語言有：PHP, Python, RHTML, JavaScript, Smarty, Mason, Node.js, XBL, Tcl, HTML, HTML5, TemplateToolkit, XUL, Django, Perl, Ruby, Python3.

5.15 Copy Filepath With Line Numbers

5.16 file-downloader

6 tips

6.1 快捷鍵介紹

看這裏，Sublime Text3 使用指南

6.2 列編輯模式

1. 方式一

   Shift+ 鼠標右鍵 or 鼠標中鍵

2. 方式二

   sublime 對列編輯模式 Key binding 設置如下：

   ```
   路徑：Preferences→Key Bindings
     { "keys": ["ctrl+alt+up"], "command": "select_lines", "args": {"forward": false} },
     { "keys": ["ctrl+alt+down"], "command": "select_lines", "args": {"forward": true} },
   但ctrl+alt+up/down 和windows的快捷鍵設置衝突，我們可以自定義上述設置
   路徑：Preferences→Key Bindings    User
   [{ "keys": ["alt+up"], "command": "select_lines", "args": {"forward": false} },
    { "keys": ["alt+down"], "command": "select_lines", "args": {"forward": true} },
   ]
   ```

3. 方式三

   選中需要進行列編輯的多行，然後按下 Ctrl+Shift+L 也可以開啓列編輯模式。

6.3 hex 查看模式

```
HexViewer
Ctrl + Shift + P
安裝HexViewer
Tools > \003work\002memo\001software\001install > Hex Viewer > Toggle Hex View
```

7 FAQ

7.1 字體怎麼調整回來

preferences->font

- 快捷鍵

  larger: ctrl+= smaller:ctrl+shift+ keypad+(注意一定要是小鍵盤上的 +)

- 和 OmnMarkupPreview 中切換標題的快捷鍵的誤用

  增大標題: ctrl+ 減小標題: ctrl+ keypad+

7.2 缺省保存爲 UTF8 文件

```
Preferences 設置-默認
Preferences.sublime-settings文件：
// 默認編碼格式
"default_encoding": "UTF-8",
```

怎麼用正則模式查找替換

(#{1,6}): 表示查找1到6個#的字符,()表示匹配的意思，并放入$1
替換成$1 ： 表示在原先的標題符號後面加上空格

出現服務找不到，preview 不成功如下提示

```
Error: 404 Not Found
Sorry, the requested URL 'http://127.0.0.1:51004/view/28' caused an error:
'buffer_id(28) is not valid (closed or unsupported file format)'
**NOTE:** If you run multiple instances of Sublime Text, you may want to adjust
the `server_port` option in order to get this plugin work again.
```

```
sublime Text > Preferences > Package Settings > OmniMarkupPreviewer > Settings - User
```

粘貼下列的擴展去代替原來的擴展（我用了方法1）

```
{
    "renderer_options-MarkdownRenderer": {
        "extensions": ["tables", "fenced_code", "codehilite"]
    }
}
```

移除了 “Strikethrough” 就好了，但是發現把這個再加回也好了。不知道什麼原因

7.3 爲何始終在當前標簽打開文件？

preferences->setting

// KL+: 解決始終在當前標簽打開文件的問題，可能是安裝了 fileDiff 插件帶來的。

“preview_on_click”: false,

7.4 總是在新窗口中打開文件？

Preferences -> Settings – Default -> 搜索 open_files_in_new_window，將其 true 改爲 false 後，重啓一下 sublime text

7.5 sublime text 的菜單欄隱藏了怎麼再顯示出來？

按住 alt 鍵，就可以暫時顯示菜單欄了，再次點擊 “顯示/隱藏菜單欄” 就能恢復了。

7.6 SublimeText 快速插入多行遞增數字

SublimeText 快速插入多行遞增數字

InsertNums

7.7 sublime ctrl+shift+f 快捷鍵失效的原因

輸入法去掉相應的快捷鍵

7.8 怎麼列編輯操作

Column Selection

- Right Mouse Button + Shift
- OR: Middle Mouse Button
- Add to selection: Ctrl
- Subtract from selection: Alt

8 LaTex 公式案例

latex example:

$$f(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$$

equation.he Lorenz Equations

$$\begin{aligned}
\dot{x} &= \sigma(y-x) \\
\dot{y} &= \rho x - y - xz \\
\dot{z} &= -\beta z + xy
\end{aligned}$$

inline an example of a LaTeX $\sqrt{3x-1}+(1+x)^2$