sublime

kevinluo

Contents

目录

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 缺省没有安装，可 2 个方法安装

1. tools-package control install

   装完此菜单消失，preferences 出现->package setting; ->package control

2. 2 种另外的方法，自动和手动

   页面查看 2 种另外的方法 在页面查看 2 种另外的方法，自动和手动

3.1.1 如果想要删除插件，

Ctrl+Shift+P调出命令面板，输入remove，调出Remove Package选项并回车，选择要删除的插件即可，当

3.1.2 用 Package Control 安装插件的方法

按下Ctrl+Shift+P调出命令面板
输入install 调出 Install Package 选项并回车，然后在列表中选中要安装的插件。
不爽的是，有的网络环境可能会不允许访问陌生的网络环境从而设置一道防火墙，而Sublime Text3貌似无

3.2 sublimegit package install

tools-command palette ctl+shift+p
pci package control install
等待载入package information,然后在命令行输入sublimeGit
安装完后，在
preference->package Settings-> 此处出现安装的sublimeGit
同时在
preference->package settings-> package control -> user setting 中可以看到已经增加选项

3.2.1 sumlimeGit usage 用法

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

- 【下面的有些问题，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

超级文本编辑器 Sublime Text3

4.1 ConvertToUTF8

比上面的那个要方便，直接在菜单栏中可以转了，专为中文设计，妈妈再也不通担心中文乱码问题了

4.2 markdown, 也可以用简书直接编辑查看

4.2.1 MarkdownEditing

4.2.2 OmniMarkupPreviewer

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

1. 如果你发现它不支持markdown目录的预览生成，那么不是它不行，是你没配置。
   当然首先是装markdwon TOC插件

2. 复制Preferences -> Package Settings -> OmniMarkupPreviewer -> Settings - Default 中的内容到Se

3. 并在 // MarkdownRenderer options区域，即

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

4．然后通过Ctrl+Alt+O快捷键生成HTML预览，或者Ctrl+Alt+X导出。

4.2.2.2 OmniMarkupPreviewer 中支持 LaTeX 公式显示： 1. 设置。

公式的渲染使用了MathJax库，所以需要在OmniMarkupPreviewer的设置中，将"mathjax_enabled"设置为

2. 可能是网速的原因，MathJax 库下载很慢，所以可以选择手动安装。

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

然后解压到下面的目录里：Sublime Text 2\\003work\002memo\001software\001install\OmniMarkupP
之后在目录“Sublime Text 2\\003work\002memo\001software\001install\OmniMarkupPreviewer”中
测试，输入下面内容：

```
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：在浏览器中预览
Ctrl+Alt+X：输出为HTML文件
Ctrl+Alt+C：复制为HTML文件

4.2.3 Markdown Extended

4.2.4 MarkdownLivePreview [alt+m 在 sublime 启动并列窗口, 实时查看结果]

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

装完后 preferences-package setting 中的名字为，sumlime-rst-completion

Restructured Text (RST) Snippets

- 用法链接
  - 本地 README
  - Git-README
- 快捷键
  - 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-`
- 补齐: `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`

-编译 Python 项目文档

Python 的项目文档，大都基于 reStructuredText 撰写，Sphinx 发布，如何在 Sublime 中，通过按 Ctrl + B 直接编译工程呢？很简单，点击 Tools –> Build System –> New Build System ，输入

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

保存，打开你工程的 Makefile 文件，然后按 Ctrl + Shift + B 选择你刚才保存的那个名字，就可以自动编译成 html 文档了。

4.5 Anaconda - python completion

Anaconda 强大的补全工具, 还能实时看文档, 转到定义, 自动格式化代码

doc

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

4.6 HexViewer

hex 查看模式

4.7 latex tools

git latextools 项目

DOC on readthedocs

- 配套
  1. sumatrapdf
     sumatraPdf 网址 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

插入日期时间

4.10 SideBarEnhancements

*.html 文件，按 f5 自动启动 chrome 浏览设置

```
[
    { "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"//匹配任何文件类型
            ,
        }
    },

]
```

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：

发生了不取词的问题。原因是：user setting 文件中 “encoding”: “ISO-8859-1”, 不能为空。

触发取词改成 f1 键，输入时需要小写

手动安装 cndict, 因为 DictionaryAutoComplet 安装时，不能改成 f1 键

到这里下载

设置

- command 设置自动完成切换，总的和 package 内 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.
- 手动跳出来，Ctrl + Space

  Just type Ctrl + Space to show auto-completion,
- 自动跳出词语

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

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: 有道词典 Youdao

Ctrl+Alt+C: 金山词典 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 汇总

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. 传统法

   ```
   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. 移位法赋值

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

3. 字母

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

5.3 emmet

html自动补全
ZenCoding
不得不用的一款前端开发方面的插件，Write less，show more.安装后可直接使用，Tab键触发，Alt+Shi

5.4 VAlign

inspired by alignment, automatically align

5.5 Alignment

代码对齐，如写几个变量，选中这几行，Ctrl+Alt+A，哇，齐了。

5.6 Ctags

函数跳转，我的电脑上是 Alt+ 点击函数名称，会跳转到相应的函数

5.7 DocBlockr

注释插件，生成幽美的注释。标准的注释，包括函数名、参数、返回值等，并以多行显示，省去手动编写。

5.8 JsFormat

格式化 js 代码，这个插件很有用，我们有时在网上看到某些效果，想查看是怎么实现的，但是代码被压缩过，很难阅读，不用怕，用 ST3 打开，按下快捷键，即可让代码还原，莫非是武林中失传已久的“还我靓靓拳”。

5.9 Tag

格式化标签，让乱七八糟的代码，瞬间整齐清晰。

5.10 BracketHighlighter

括弧高亮显示。

5.11 Clipboard History

剪切板历史，可以保存多个复制信息，按下ctrl+alt+v，可以选择历史剪切板。
Goto-CSS-Declaration
跳转到css文件该class的声明处，方便修改查看，如图下所示，注意对应的css文件要同时打开才行。

5.12 SCSS

支持 scss 的语法高亮，里面附带了好多 CSS Snippet，无论现用或者改造成，都可节省不少时间。还有很多插件，jquery 语法提示，jsHint 等等。

5.13 Sublime Linter

这个插件帮你找到代码中的错误。它支持很多语言：PHP, Python, Java, CoffeScript, CSS, HTML, JavaScript, Perl, PHP, Python, Ruby, XML 等。Javascript 需要安装 Node.js 引擎，其他配置详见项目主页。强烈推荐安装。

5.14 Sublime CodeIntel

Sublime CodeIntel 是我最喜欢的插件，它提供了很多 IDE 提供的功能，例如代码自动补齐，快速跳转到变量定义，在状态栏显示函数快捷信息等。它支持的语言有：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 快捷键介绍

看这里，Sublime Text3 使用指南

6.2 列编辑模式

1. 方式一

   Shift+ 鼠标右键 or 鼠标中键

2. 方式二

   sublime 对列编辑模式 Key binding 设置如下：

   ```
   路径：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的快捷键设置冲突，我们可以自定义上述设置
   路径：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

   选中需要进行列编辑的多行，然后按下 Ctrl+Shift+L 也可以开启列编辑模式。

6.3 hex 查看模式

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

7 FAQ

7.1 字体怎么调整回来

preferences->font

- 快捷键

  larger: ctrl+= smaller:ctrl+shift+ keypad+(注意一定要是小键盘上的 +)

- 和 OmnMarkupPreview 中切换标题的快捷键的误用

  增大标题: ctrl+ 减小标题: ctrl+ keypad+

7.2 缺省保存为 UTF8 文件

```
Preferences 设置-默认
Preferences.sublime-settings文件：
// 默认编码格式
"default_encoding": "UTF-8",
```

怎么用正则模式查找替换

(#{1,6}): 表示查找1到6个#的字符,()表示匹配的意思，并放入$1
替换成$1 ：表示在原先的标题符号后面加上空格

出现服务找不到，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
粘贴下列的扩展去代替原来的扩展（我用了方法1）
{
    "renderer_options-MarkdownRenderer": {
        "extensions": ["tables", "fenced_code", "codehilite"]
    }
}
```

移除了 “Strikethrough” 就好了，但是发现把这个再加回也好了。不知道什么原因

7.3 为何始终在当前标签打开文件？

preferences->setting

// KL+: 解决始终在当前标签打开文件的问题，可能是安装了 fileDiff 插件带来的。

“preview_on_click”: false,

7.4 总是在新窗口中打开文件?

Preferences -> Settings – Default -> 搜索 open_files_in_new_window，将其 true 改为 false 后，重启一下 sublime text

7.5 sublime text 的菜单栏隐藏了怎么再显示出来？

按住 alt 键，就可以暂时显示菜单栏了，再次点击 “显示/隐藏菜单栏” 就能恢复了。

7.6 SublimeText 快速插入多行递增数字

SublimeText 快速插入多行递增数字

InsertNums

7.7 sublime ctrl+shift+f 快捷键失效的原因

输入法去掉相应的快捷键

7.8 怎么列编辑操作

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 $\sqrt{3x-1}+(1+x)^2$