

目 录

[致谢](#)

[Learning Sublime Text](#)

致谢

当前文档《Sublime Text指南》由 进击的皇虫 使用 书栈(BookStack.CN) 进行构建，生成于 2018-03-27。

书栈(BookStack.CN) 仅提供文档编写、整理、归类等功能，以及对文档内容的生成和导出工具。

文档内容由网友们编写和整理，书栈(BookStack.CN) 难以确认文档内容知识点是否错漏。如果您在阅读文档获取知识的时候，发现文档内容有不恰当的地方，请向我们反馈，让我们共同携手，将知识准确、高效且有效地传递给每一个人。

同时，如果您在日常生活、工作和学习中遇到有价值有营养的知识文档，欢迎分享到 书栈(BookStack.CN)，为知识的传承献上您的一份力量！

如果当前文档生成时间太久，请到 书栈(BookStack.CN) 获取最新的文档，以跟上知识更新换代的步伐。

文档地址：<http://www.bookstack.cn/books/sublime-text-guidance>

书栈官网：<http://www.bookstack.cn>

书栈开源：<https://github.com/TruthHun>

分享，让知识传承更久远！感谢知识的创造者，感谢知识的分享者，也感谢每一位阅读到此处的读者，因为我们都将成为知识的传承者。

Learning Sublime Text

- [Learning Sublime Text \(系统学习ST\)](#)
 - [前言](#)
 - [目录导航](#)

Learning Sublime Text (系统学习ST)

编辑、整理 By Xujiajuan (相关视频整理 <http://www.xitongxue.com/course/2>)

前言

如何编辑本文？[请点击这里](#)

如何fork本文？点击右上角“Fork”按钮。

如何发pull Requests 作出贡献？[[请点击这里](#)]

(<https://github.com/xujiajuan/Learning-Sublime-Text/pulls>)

目录导航

- [1、Why Sublime Text?](#)
- [2、Sublime Text 安装以及简介](#)
- [3、主题](#)
- [4、常用命令](#)
- [5、命令行工具](#)
- [6、自定义配置](#)
- [7、Package Control安装以及管理](#)
- [8、插件安装以及使用](#)
- [9、定制代码片段](#)

1、Why Sublime Text?

0、高效（启动快、快捷键强大）

1、扩展性强（插件丰富）

2、主题丰富

3、支持多屏、多变量操作

4、跨平台

5、不注册亦可免费使用（有米的童鞋可以支持下原作者 USD \$70 ）

6、支持自定义

更多介绍，请观看视频：<http://www.xitongxue.com/course/2/learn#lesson/29>

2、Sublime Text 安装以及简介

Sublime Text 2 下载地址：[\[http://www.sublimetext.com/2\]](http://www.sublimetext.com/2)

Sublime Text 3 下载地址：[\[http://www.sublimetext.com/3\]](http://www.sublimetext.com/3) [推荐]

更多介绍，请参看：<http://www.xitongxue.com/course/2/learn#lesson/27>

3、主题

安装Colorsublime-Plugin来扩展主题

0、打开ST的命令面板 `ctl+shift+p` (Windows/Linux) or `⌘+⇧+p` (OSX)

1、选择Colorsublime: Install Theme

2、用箭头键来浏览各个主题、你可以实时看到切换的主题变化！

[Colorsublime官网](#)

4、常用命令

Win/Ubuntu	Mac	注解
<code>Ctrl+Shift+P:</code>	<code>Cmd+Shift+p</code>	打开命令面板
<code>Ctrl+P:</code>	<code>Cmd+P</code>	搜索项目中的文件
<code>Ctrl+G:</code>	<code>Ctrl+g</code>	跳转到第几行
<code>Ctrl+W:</code>	<code>Cmd+w</code>	关闭当前打开文件
<code>Ctrl+Shift+W:</code>	<code>Cmd+Shift+w</code>	关闭所有打开文件
<code>Ctrl+Shift+V:</code>	<code>Cmd+Shift+v</code>	粘贴并格式化
<code>Ctrl+D:</code>	<code>Cmd+d</code>	选择单词，重复可增加选择下一个相同的单词

Ctrl+L:	Cmd+l	选择行，重复可依次增加选择下一行
Ctrl+Shift+L:	Cmd+Shift+l	选择多行
Ctrl+Shift+Enter:	Cmd+Shift+Enter	在当前行前插入新行
Ctrl+F:	Cmd+f	查找内容
Ctrl+Shift+F:	Cmd+Shift+f	查找并替换
Ctrl+H:	Cmd+H	替换
Ctrl+R:	Cmd+r	前往 method
Ctrl+N:	Cmd+N	新建窗口
Ctrl+K+B:	Cmd+K+B	开关侧栏
Ctrl+/:	Cmd+/**	注释当前行
Alt+Shift+数字:	Cmd+option+num	分屏显示
Alt+数字:	Cmd+num	切换打开第N个文件
Ctrl+Shift+上下键	Cmd+control+上下键	可替换行

5、命令行工具

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

用法

```
1. 运行 subl --help
2. Usage: subl [arguments] [files]      编辑指定的文件edit the given files
3.    or: subl [arguments] [directories]  打开指定的目录
4.    or: subl [arguments] -            编辑stdin
5.
6. Arguments:
7.   --project <project>: 载入指定的project
8.   --command <command>: 运行指定的命令
9.   -n or --new-window: 打开一个新的窗口
10.  -a or --add:         添加文件夹到当前窗口
11.  -w or --wait:        返回前等待文件关闭
12.  -b or --background: 不激活该应用程序
13.  -s or --stay:        文件关闭后保持应用程序激活状态
14.  -h or --help:        显示帮助并退出
15.  -v or --version:     显示版本信息并退出
```

如果从标准输入—wait是隐式的。 使用—stay当文件关闭是不切换到后台控制台(只与是否有等待的文件有关)。

文件名可以通过加:line或者:line:column后缀来指定打开的定位。

6、自定义配置

```

1. {
2.     "color_scheme": "Packages/Colorsublime - Themes/Monokai.tmTheme",
3.     "create_window_at_startup": false,
4.     "font_size": 15,
5.     "highlight_line": true,
6.     "highlight_modified_tabs": true,
7.     "ignored_packages":
8.     [
9.         "Vintage",
10.        "Markdown"
11.    ],
12.    "show_debug": true,
13.    "show_encoding": true,
14.    "show_full_path": true,
15.    "translate_tabs_to_spaces": true
16. }
```

PS: [分享下我的配置](#)

7、Package Control安装以及使用

mac下Opt+` 或者 菜单栏选择View > Show Console

ST3:

```

1. import urllib.request,os,hashlib; h = '7183a2d3e96f11eeadd761d777e62404' +
    'e330c659d4bb41d3bdf022e94cab3cd0'; pf = 'Package Control.sublime-package'; ipp =
    sublime.installed_packages_path(); urllib.request.install_opener( urllib.request.build_opener(
    urllib.request.ProxyHandler()) ); by = urllib.request.urlopen( 'http://packagecontrol.io/' +
    pf.replace(' ', '%20')).read(); dh = hashlib.sha256(by).hexdigest(); print('Error validating
    download (got %s instead of %s), please try manual install' % (dh, h)) if dh != h else
    open(os.path.join( ipp, pf), 'wb' ).write(by)
```

ST2:

```

1. import urllib2,os,hashlib; h = '7183a2d3e96f11eeadd761d777e62404' +
    'e330c659d4bb41d3bdf022e94cab3cd0'; pf = 'Package Control.sublime-package'; ipp =
    sublime.installed_packages_path(); os.makedirs( ipp ) if not os.path.exists(ipp) else None;
    urllib2.install_opener( urllib2.build_opener( urllib2.ProxyHandler()) ); by = urllib2.urlopen(
    'http://packagecontrol.io/' + pf.replace(' ', '%20')).read(); dh =
    hashlib.sha256(by).hexdigest(); open( os.path.join( ipp, pf), 'wb' ).write(by) if dh == h else
    None; print('Error validating download (got %s instead of %s), please try manual install' % (dh,
```

```
h) if dh != h else 'Please restart Sublime Text to finish installation')
```

8、插件安装以及使用

0、shift + cmd + p 打开命令面板

1、输入 “Package Control: Install Package” 命令或者install 关键词找到 “Package Control: Install Package”

2、输入安装插件的简写或全拼, 找到后回车安装

9、定制代码片段

在编写代码的时候, 有遇到反复使用的代码片段, 这时候就需要复制、黏贴, 大大影响效率。我们利用Sublime Text的snippet功能, 就能很好的解决这一问题。

如何创建? Tools > New Snippet

会出现如下:

```
1. <snippet>
2.     <content><![CDATA[
3. Hello, ${1:this} is a ${2:snippet}.
4. ]]></content>
5.     <!-- Optional: Set a tabTrigger to define how to trigger the snippet -->
6.     <!-- <tabTrigger>hello</tabTrigger> -->
7.     <!-- Optional: Set a scope to limit where the snippet will trigger -->
8.     <!-- <scope>source.python</scope> -->
9. </snippet>
```

看不懂? ok我们看下例子

创建我的第一个Snippet:

```
1. <snippet>
2.     <content><![CDATA[ 你需要插入的代码片段${1:name} ]]></content>
3.     <!-- 可选: 快捷键, 利用Tab自动补全代码的功能, 这里你输入superu即可触发 -->
4.     <tabTrigger>superu</tabTrigger>
5.     <!-- 可选: 使用范围, 不填写代表对所有文件有效。 -->
6.     <scope>source.python</scope>
7.     <!-- 可选: 在snippet菜单中的显示说明(支持中文)。如果不定义, 菜单则显示当前文件的文件名。 -->
8.     <description>My first Snippet</description>
9. </snippet>
```

`${1:name}`表示代码插入后，光标所停留的位置，可同时插入多个。其中`:name`为自定义参数（可选）。

`${2}`表示代码插入后，按Tab键，光标会根据顺序跳转到相应位置（以此类推）。