

doobook

☞ ☐ m ☞ ☐ ☐ &

Thank

前言

总结和整理在工作，学习实践中遇到的一些软件安装和问题解决方案。

如果你看了之后，觉得有用，可以帮到你，请随意打赏。

左图：微信支付；右图：支付宝支付



NodeJs

导语

Node.js是一个基于Chrome JavaScript运行时建立的平台，用于方便地搭建响应速度快、易于扩展的网络应用。Node.js 使用事件驱动，非阻塞I/O 模型而得以轻量 and 高效，非常适合在分布式设备上运行的数据密集型的实时应用。

Node是一个Javascript运行环境(runtime)。实际上它是对Google V8引擎进行了封装。V8引擎执行Javascript的速度非常快，性能非常好。Node对一些特殊用例进行了优化，提供了替代的API，使得V8在非浏览器环境下运行得更好。

Node.js安装

Window系统下Nodejs的安装及环境配置

笔者测试环境：windows 7 旗舰版 64位操作系统

安装步骤如下：

Setup 1：下载Nodejs安装包。

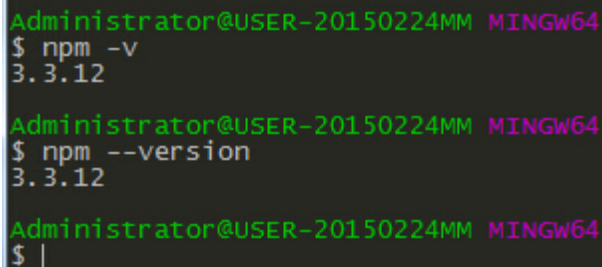
[官网下载](#)

Setup 2：安装Nodejs。

双击下载的Nodejs安装包（如：node-v5.5.0-x64.msi），按提示操作即可。（笔者的Nodejs安装目录为：C:\nodejs\）。安装完毕后，在cmd控制台输入：node -v（小写）或 node --version，控制台将打印出：v5.0.0，出现版本号提示，则表示安装成功。

Setup 3：npm的安装

新版的nodejs已经集成了npm，可以使用cmd命令行输入"npm -v"或"npm --version"来测试是否成功安装。如下图，出现版本号则表示安装成功：



```
Administrator@USER-20150224MM MINGW64
$ npm -v
3.3.12

Administrator@USER-20150224MM MINGW64
$ npm --version
3.3.12

Administrator@USER-20150224MM MINGW64
$ |
```

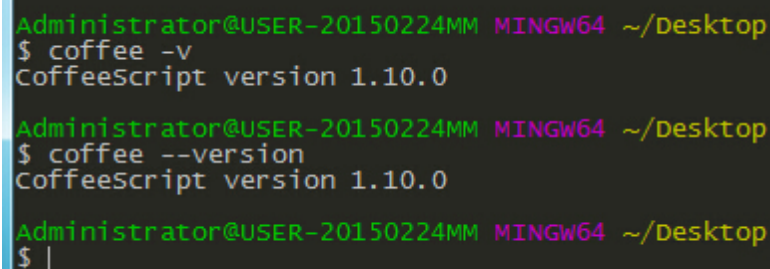
Setup 4 : 安装相关的环境

```
#安装express 3.5.0,避免运行express时提示" express"不是内部或外部命令
npm install -g express@3.5.0
#nodejs模板引擎
npm install -g jade
#mysql
npm install -g mysql
```

Setup 5:安装CoffeeScript

```
npm install coffee-script -g
```

确认安装的命令：coffee -v或coffee --version，出现版本号表示成功安装。如下图，出现版本号则表示安装成功：



```
Administrator@USER-20150224MM MINGW64 ~/Desktop
$ coffee -v
CoffeeScript version 1.10.0

Administrator@USER-20150224MM MINGW64 ~/Desktop
$ coffee --version
CoffeeScript version 1.10.0

Administrator@USER-20150224MM MINGW64 ~/Desktop
$ |
```

Setup 6 : CoffeeScript测试实例

1. 在window D盘（可在任意地方新建）上新建test.js测试文件。写入以下内容：
for i in [1,2,3,4,5,6,7,8]
 console.log "Demo " + i;
接着保存。
2. 按win + r 键，输入cmd，打开命令行窗口，输入命令：
d:
coffee test.js
- 3.效果如下图：

```
C:\Users\Administrator>d:
```

```
D:\>coffee test.js
```

```
Demo 1
```

```
Demo 2
```

```
Demo 3
```

```
Demo 4
```

```
Demo 5
```

```
Demo 6
```

```
Demo 7
```

```
Demo 8
```

```
D:\>_
```

Gitbook

导语

GitBook 是一个基于 Node.js 的命令行工具，可使用 Github/Git 和 Markdown 来制作精美的电子书，GitBook 并非关于 Git 的教程。

GitBook[1] 是一个基于 Node.js 的命令行工具，可使用 Github/Git 和 Markdown 来制作精美的电子书，GitBook 并非关于 Git 的教程。

使用GitBook制作电子书，必备两个文件：README.md和SUMMARY.md。

环境

windows 7 旗舰版 64位操作系统，笔者已安装nodeJS和git。本章所测试的示例，都基于上述环境。

Gitbook的安装

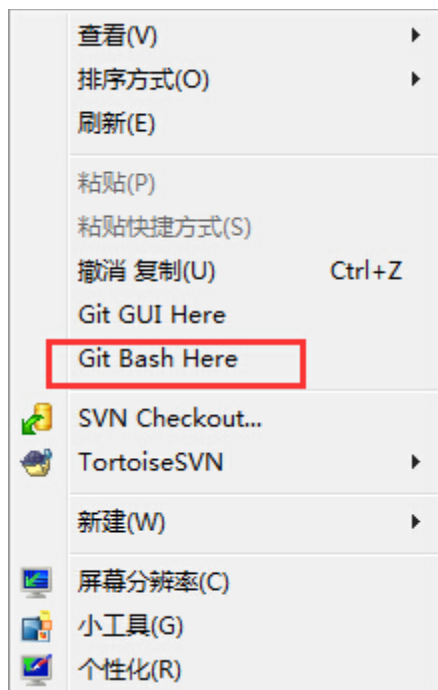
Gitbook的安装

温馨提醒：笔者在安装gitbook时，已经事先安装了nodejs和git。测试环境：windows 7 旗舰版 64位操作系统。

安装步骤如下：

Setup 1：打开git命令窗口，具体操作如下：

点击鼠标右键，选择 Git Bash Here。如图：



Setup 2 : 在git命令窗口中输入如下命令 :

```
npm install -g gitbook
npm install -g gitbook-cli
```

Setup 3 : 检验是否安装成功。

```
gitbook -V 或 gitbook --version // -V中的V是大写
```

若出现版本号提示，则表示安装成功。如下图：

```
Administrator@USER-20150224MM MINGW64 ~/Desktop
$ gitbook -V
1.0.1

Administrator@USER-20150224MM MINGW64 ~/Desktop
$ gitbook --version
1.0.1

Administrator@USER-20150224MM MINGW64 ~/Desktop
$ |
```

Gitbook的配置

配置

title

设置图书的标题。

```
"title" : "DocBook"
```

author

设置图书作者的相关信息。

```
"author" : "Thank"
```

description

设置图书的简单描述。

```
"description" : "总结和整理在工作，学习实践中遇到的一些软件安装和问题。"
```

language

设置Gitbook使用的语言, 版本2.6.4中可选的语言如下：

en, ar, bn, cs, de, en, es, fa, fi, fr, he, it, ja, ko, no, pl, pt, ro, ru, sv, uk, vi, zh-hans, zh-tw

配置使用简体中文

```
"language" : "zh-hans",
```

links

在左侧导航栏添加链接信息。

```
"links" : {  
  "sidebar" : {  
    "DocBook" : "http://bitbybit.top"  
  }  
}
```

styles

自定义页面样式，默认情况下各generator对应的css文件：


```
"styles": {
  "website": "styles/website.css",
  "ebook": "styles/ebook.css",
  "pdf": "styles/pdf.css",
  "mobi": "styles/mobi.css",
  "epub": "styles/epub.css"
}
```

例如使 `<h1>` `<h2>` 标签有下边框，可以在website.css中设置。

```
h1, h2{
  border-bottom: 1px solid #DEDEDE;
}
```

plugins

配置使用的插件。

```
"plugins": [
  "disqus"
]
```

添加新插件之间需要运行gitbook install来安装新的插件。

Gitbook默认带有5个插件：

- highlight //高亮
- search //搜索
- sharing //分享
- font-settings //字体设置
- livereload //重新加载

如果要去掉自带的插件，可以在插件名称前面加减号-。

```
"plugins": [
  "-search"
]
```

pluginsConfig

配置插件的属性。

```
"pluginsConfig": {  
  "fontsettings": {  
    "theme": "sepia",  
    "family": "serif",  
    "size": 1  
  }  
}
```

Gitbook插件

Gitbook相关插件

温馨提示：所有插件的配置都在book.json中进行。在安装插件时，请用npm init 初始化package.json。

若已有package.json，可直接在图书目录下通过git或cmd命令行运行npm install。

附：本章最后将附上笔者实操测试的pageage.json和book.json。

导语

记录Gitbook的一些配置以及插件。

配置：

- title - 标题
- author - 作者信息
- description - 书本描述
- language - 使用的语言
- links - 在侧边栏添加链接
- styles - 自定义样式
- plugins - 插件
- pluginsConfig - 插件配置

插件：

- Disqus - Disqus评论
- Search Pro - 支持中文搜索
- Advanced Emoji - 支持emoji表情
- Github - 添加github图标
- Ace Plugin - 支持ace
- Emphasize - 强调文字(为文字加上底色)

- KaTex - 支持数学公式
- Include Codeblock - 使用代码块显示指定文件的内容
- Splitter - 自由调节侧边栏宽度
- Mermaid - 支持渲染Mermaid图表
- sharing - 分享
- Tbfed-pagefooter - 为页面添加页脚
- Toggle Chapters - 折叠章节
- Sectionx - 将页面分块显示
- Codeblock-filename - 为代码块添加文件名称
- ga - google 统计
- baidu - 百度统计

相关插件实操文档持续更新中。

Disqus

Disqus

添加disqus评论。

插件地址

安装：

```
npm install gitbook-plugin-disqus --save
```

配置：

配置参考

```
//book.json
"plugins": [
  "disqus"
],
"pluginsConfig": {
  "disqus": {
    "shortName": "docbook" // 在Disqus官方网站注册的shortName
  }
}
```

Search Pro

Search Pro

支持中文搜索，需要将默认的search插件去掉，:worried: 在windows下安装该插件时总是出错，同时如果标题中有包含的关键字，标题的样式会有所变化（笔者测试不成功）:sob:。

插件地址

安装：

```
npm install gitbook-plugin-search-pro --save
```

配置：

```
//book.json
"plugins": [
  "-search",
  "search-pro"
],
"pluginsConfig": {
  "search-pro": {
    "cutWordLib": "nodejieba",
    "defineWord": ["中文", "分词"]
  }
}
```

Advanced Emoji

Advanced Emoji

支持emoji表情。

emoji表情列表地址

插件地址

安装：

```
npm install gitbook-plugin-advanced-emoji --save
```

配置：

```
//book.json
"plugins": [
  "advanced-emoji"
]
```

使用示例:

:clap: :muscle: :metal: :eggplant: :blush:

Github

Github

添加github图标。

插件地址

安装：

```
npm install gitbook-plugin-github --save
```

配置：

```
//book.json
"plugins": [
  "github"
],
"pluginsConfig": {
  "github": {
    "url": "https://github.com/xieguo"
  }
}
```

Ace Plugin

Ace Plugin

使gitbook支持ace。

插件地址

安装：

```
npm install gitbook-plugin-ace --save
```

配置：

```
//json.book
"plugins": [
  "ace"
]
```

使用示例:

```
{%ace edit=false, lang='php'%}
```

```
//This is a hello world program for PHP.=>冒泡法排序实例
```

```
function show($arr){
```

```
$count = count($arr);
```

```
    for($i = 0;$i<$count;$i++){
        for($j = 0;$j<$count-$i-1;$j++){ // $i减少循环次数
            if($arr[$j]>$arr[$j+1]){
                $tmp = $arr[$j];
                $arr[$j] = $arr[$j+1];
                $arr[$j+1] = $tmp;
            }
        }
    }

    return $arr;
```

```
}
```

```
{%endace%}
```

Emphasize

Emphasize

为文字加上底色。

插件地址

安装：

```
npm install gitbook-plugin-emphasize --save
```

配置：

```
//book.json
"plugins": [
  "emphasize"
]
```

使用示例：

This text is {% em %}highlighted !{% endem %}

This text is {% em %}highlighted with markdown!{% endem %}

This text is {% em type="green" %}highlighted in green!{% endem %}

This text is {% em type="red" %}highlighted in red!{% endem %}

This text is {% em color="#ff0000" %}highlighted with a custom color!{% endem %}

KaTex

KaTex

为了支持数学公式, 我们可以使用KaTex和MathJax插件, 官网上说Katex速度要快于MathJax。

插件地址

MathJax使用LaTeX语法编写数学公式教程

安装：

```
npm install gitbook-plugin-katex --save
```

配置：

```
//book.json
"plugins": [
  "katex"
]
```

使用示例:

Inline math: $\int_{-\infty}^{\infty} g(x) dx$

Block math:

```
$$
\int_{-\infty}^{\infty} g(x) dx
$$
```

Include Codeblock

Include Codeblock

使用代码块的格式显示所包含文件的内容。且包含的文件必须存在。

插件地址

安装：

```
npm install gitbook-plugin-include-codeblock --save
```

配置：

```
//book.json
"plugins": [
  "include-codeblock"
]
```

使用示例:

`include`

{% em %}注意：Include Codeblock的文件路径是相对的。相对于当前文档所在的目录。{% endem %}

Splitter

本文档使用 [看云](#) 构建

Splitter

让侧边栏的宽度可以自由调节。

[插件地址](#)

安装：

```
npm install gitbook-plugin-splitter --save
```

配置：

```
//book.json
"plugins": [
  "splitter"
]
```

Mermaid

Mermaid

支持渲染Mermaid图表。

[插件地址](#)

安装：

```
npm install gitbook-plugin-mermaid --save
```

配置：

```
//book.json
"plugins": [
  "mermaid"
]
```

使用示例:

{% em %} 注意：需要python的支持，笔者windows 10 尚未安装python测试。 {% endem %}

Sharing

Sharing

分享当前页面，gitbook的默认插件，使用下面方式来禁用。

```
plugins: ["-sharing"]
```

配置:

```
//book.json
"pluginsConfig": {
  "sharing": {
    "weibo": true,
    "facebook": true,
    "twitter": true,
    "google": false,
    "instapaper": false,
    "vk": false,
    "all": [
      "facebook", "google", "twitter",
      "weibo", "instapaper"
    ]
  }
}
```

Tbfed-pagefooter

Tbfed-pagefooter

为页面添加页脚。

插件地址

安装：(安装时存在问题，但不影响，已截图。)

```
npm install gitbook-plugin-tbfed-pagefooter --save
```

配置：

```
//book.json
"plugins": [
  "tbfed-pagefooter"
],
"pluginsConfig": {
  "tbfed-pagefooter": {
    "copyright": "@Copyright &copy bitbybit.top 2016",
    "modify_label": "修订时间：",
    "modify_format": "YYYY-MM-DD HH:mm:ss"
  }
}
```

Toggle Chapters

Toggle Chapters

让左侧的章节目录可以折叠。

[插件地址](#)

安装：安装时存在问题，但不影响，已截图。）

```
npm install gitbook-plugin-toggle-chapters --save
```

配置：

```
//book.json
"plugins": ["toggle-chapters"]
```

Sectionx

Sectionx

将页面分块显示。

[插件地址](#)

安装：

```
npm install gitbook-plugin-sectionx --save
```

配置：

```
//book.json
"plugins": [
  "sectionx"
]
```

使用示例

Insert markdown content here (you should start with h3 if you use heading).

Codeblock-filename

Codeblock-filename

为代码块添加文件名称。

[插件地址](#)

安装：

```
npm install gitbook-plugin-codeblock-filename --save
```

配置：

```
//book.json
plugins: [ "codeblock-filename" ]
```

使用示例:

```
function sum(a, b)
{
  return a + b;
}
```

ga

ga

google 统计。

[插件地址](#)

安装：

```
npm install gitbook-plugin-ga --save
```

配置：

```
//book.json
"plugins": [
  "ga"
],
"pluginsConfig": {
  "ga": {
    "token": "UA-XXXX-Y"
  }
}
```

baidu

baidu

百度统计。

[插件地址](#)

安装：

```
npm install gitbook-plugin-baidu --save
```

配置：

```
//book.json
"plugin": [
  "baidu"
],
"pluginsConfig": {
  "baidu": {
    "token": "YOUR TOKEN"
  }
}
```

Gitbook使用

Gitbook使用

笔者测试环境：windows 7 旗舰版 64位操作系统

前言：Gitbook是一个命令行工具(node.js库)，可使用Gitbook创建漂亮的图书。

要点：图书的基本要素至少包含两个文件: *README.md* 和 *SUMMARY.md*。

必需相关文件

典型的，它应该是你的图书的介绍。它可以自动的被加到最终的summary中。

SUMMARY.md 定义了你的图书的目录结构，它应该包含章节的列表，以及它们的链接。

Gitbook相关命令说明：

图书初始化

//图书初始化，若没有SUMMARY.md和README.md文件，则自动创建。

```
gitbook init [path]
```

```
gitbook serve [path] 或  
gitbook serve [path] -p 8080 //path为图书目录所在路径；8080为自定义的端口
```

生成静态网站

//path : 为图书目录所在路径。

//outputFolder : 静态网站存放的目录路径。

```
gitbook build path --output= outputFolder // =号右边一定要有空格
```

Gitbook serve和Gitbook build 的相关参数：

```
-o, --output <directory> 输出文件，默认为 ./_book。  
-f, --format <name>      书籍的类型，默认为静态站点，可用的格式为: site, page, ebook, json。  
--config <config file>   配置文件，默认为 book.js 或 book.json。
```

输出格式

- 静态站点: 默认格式，创建一个完全交互式的静态网站，可以发布到GitHub等网站。
- eBook: 图书完成后可以使用它创建电子书。创建命令: `gitbook ebook ./mybook`。
你需要安装 `ebook-convert`，输出格式可以是PDF，ePub 或 MOBI。
- 单页网页: 可以生成一个单页的HTML网页。这个格式可以用来转换PDF或者eBook。创建命令:
`gitbook build ./myrepo -f page`。
- JSON: 此格式用来调试或者抽取图书的元数据，创建命令: `gitbook build ./mybook -f json`。

忽略文件和文件夹

GitBook 读取 `.gitignore`，`.bookignore` 和 `.ignore` 得到需要忽略的文件/文件夹的列表。（文件的格式和 `.gitignore` 一样）。

Gitbook 应用实例

步骤如下：

1. 在D盘上新建图书目录：docbook;
2. 在docbook目录下新建README.md文件，输入图书的介绍。
3. 在docbook目录下新建SUMMARY.md文件，输入图书的目录结构(利用markdown)。
4. 打开git中(或cmd)命令窗口，切换至图书目录docbook。
5. 输入命令：gitbook serve。
6. 打开浏览器，请入git(cmd)命令窗口提示的 serving book on <url> 中的url。
7. 根据SUMMARY.md文件的目录结构相对应的文件，编辑图书内容。

相关截图：

summary.md

```
* [基本安装](BaseInstall/README.md)
  - [Node.js安装](BaseInstall/nodejs.md)
  - [Gitbook安装](BaseInstall/gitbook.md)
  - [Gitbook使用](BaseInstall/usegitbook.md)
  - [Sublime3下两款Markdown插件](BaseInstall/sublime_plugin.md)
* [微信]
```

gitbook serve

```
$ gitbook serve
Live reload server started on port: 35729
Press CTRL+C to quit ...

info: loading book configuration...OK
info: load plugin gitbook-plugin-highlight ....OK
info: load plugin gitbook-plugin-search ....OK
info: load plugin gitbook-plugin-sharing ....OK
info: load plugin gitbook-plugin-fontsettings ....OK
info: load plugin gitbook-plugin-livereload ....OK
info: >> 5 plugins loaded
info: start generation with website generator
info: clean website generator
info: OK
info: generation is finished

Starting server ...
Serving book on http://localhost:4000
```

Bookjson

笔者实操测试的book.json

```
{
  "output": null,
  "generator": "ebook",
  "title": null,
  "description": null,
  "extension": null
```



```

"plugins": [
  "-search",
  "-disqus",
  "advanced-emoji",
  "github",
  "ace",
  "emphasize",
  "katex",
  "include-codeblock",
  "splitter",
  "-sharing",
  "-tbfed-pagefooter",
  "toggle-chapters",
  "sectionx",
  "codeblock-filename",
  "baidu"
],
"pluginsConfig": {
  "fontSettings": {
    "theme": "sepia",
    "family": "serif",
    "size": 1
  },
  "disqus": {
    "shortName": "docbook"
  },
  "github": {
    "url": "https://github.com/xieguo"
  },
  "tbfed-pagefooter": {
    "copyright": "@Copyright &copy bitbybit.top 2016",
    "modify_label": "修订时间 :",
    "modify_format": "YYYY-MM-DD HH:mm:ss"
  },
  "baidu": {
    "token": "f095aafc0c7b0201f1c10810832e9a09"
  }
},
"links": {
  "sidebar": {
    "DocBook": ""
  }
},
"pdf": {
  "toc": true,
  "pageNumbers": false,
  "fontSize": 12,
  "paperSize": "a4",
  "margin": {
    "right": 62,
    "left": 62,
    "top": 36,
    "bottom": 36
  }
}

```

```
}  
}
```

Package.json

笔者实操测试的package.json

```
{  
  "name": "docbook",  
  "version": "1.0.0",  
  "description": "Document Book",  
  "main": "README.md",  
  "dependencies": {  
    "ebook-convert": "^1.0.0",  
    "gitbook-plugin-ace": "^0.1.0",  
    "gitbook-plugin-advanced-emoji": "^0.1.5",  
    "gitbook-plugin-baidu": "0.0.2",  
    "gitbook-plugin-codeblock-filename": "0.0.7",  
    "gitbook-plugin-disqus": "^0.1.0",  
    "gitbook-plugin-emphasize": "^1.1.0",  
    "gitbook-plugin-ga": "^1.0.1",  
    "gitbook-plugin-github": "^2.0.0",  
    "gitbook-plugin-include-codeblock": "^1.6.1",  
    "gitbook-plugin-katex": "^1.1.3",  
    "gitbook-plugin-sectionx": "^2.0.1",  
    "gitbook-plugin-splitter": "0.0.6",  
    "gitbook-plugin-tbfed-pagefooter": "0.0.1",  
    "gitbook-plugin-toggle-chapters": "0.0.3"  
  },  
  "devDependencies": {},  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  "author": "Thank",  
  "license": "ISC",  
  "repository": {  
    "type": "git",  
    "url": "git@git.coding.net:xieguo/DocBook.git"  
  }  
}
```

Markdown

导语

Markdown是一种可以使用普通文本编辑器编写的标记语言，通过简单的标记语法，它可以使普通文本内容具有一定的格式。

Markdown具有一系列衍生版本，用于扩展Markdown的功能（如表格、脚注、内嵌HTML等等），这些功能原初的Markdown尚不具备，它们能让Markdown转换成更多的格式，例如

LaTeX, Docbook。Markdown增强版中比较有名的有Markdown Extra、MultiMarkdown、Maruku等。这些衍生版本要么基于工具，如Pandoc；要么基于网站，如GitHub和Wikipedia，在语法上基本兼容，但在一些语法和渲染效果上有改动。

参考：[图灵社区](#)

[简书](#)

[gitbook](#)

Markdown Plugin

Sublime text 3下两款Markdown插件

笔者测试环境：windows 7 旗舰版 64位操作系统。

MarkdownEditing

MarkdownEditing是Markdown写作者必备的插件，它不仅可以高亮显示Markdown语法，还支持很多编程语言的语法高亮显示。

OmniMarkupPreviewer

OmniMarkupPreviewer用来预览markdown编辑的效果，同样支持渲染代码高亮的样式。

注意：应先安装MarkdownEditing再安装OmniMarkupPreviewer。否则OmniMarkupPreviewer的快捷键不起作用。具体原因未明。

MarkdownEditing

MarkdownEditing

笔者电脑上已经安装sublime text 3,且已经安装包管理插件Package Control

安装MarkdownEditing

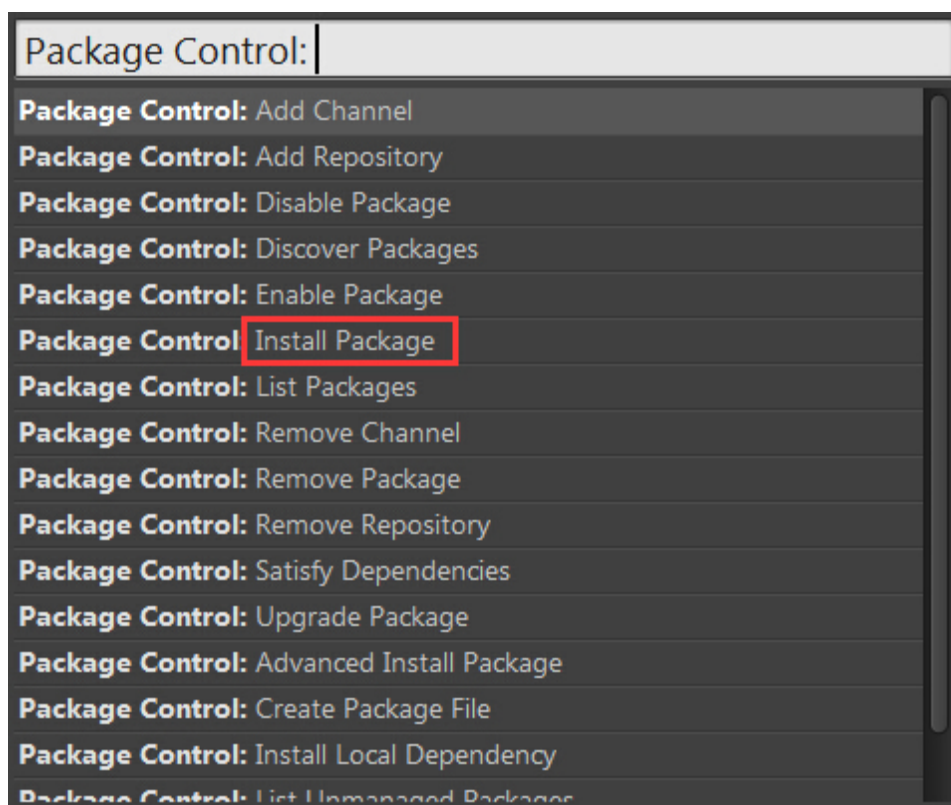
Setup 1 : 打开Sublime text 3 软件。

Setup 2 : 选择下面你喜欢的方式之一打开Package Control命令面板。

1. 使用快捷键 'ctrl + shift + p ' 进入到Sublime 命令面板，接着输入 "package install"，从下拉列表中选择 "install Package"。

2. 从菜单栏中选择Preferences->Package Control。

Setup 3: 接着Setup 2 操作，在命令面板中选择 Install Package 。 如下图：



Setup 4 : 在新弹出的命令面板中输入MarkdownEditing，找到MarkdownEditing并点击。

Setup 5 : 重启Sublime text 3。

MarkdownEditing的使用

新建后缀为.md的任意名称的文件，如README.md。使用Sublime text 3打开，即可使用Markdown语法进行编辑。如下图：

```

... javascript
jQuery.extend( jQuery.easing,
{
  def: 'easeOutQuad',
  easeInOutExpo: function (x, t, b, c, d) {
    if (t==0) return b;
    if (t==d) return b+c;
    if ((t/=d/2) < 1) return c/2 * Math.pow(2, 10 * (t - 1)) + b;
    return c/2 * (-Math.pow(2, -10 * --t) + 2) + b;
  }
});
...

```

MarkdownEditing的快捷键

1. mdi + tab : 插入图片
2. mdl + tab : 插入链接

OmniMarkupPreviewer

OmniMarkupPreviewer

笔者电脑上已经安装sublime text 3,且已经安装包管理插件Package Control

安装OmniMarkupPreviewer

OmniMarkupPreviewer插件的安装与MarkdownEditing的安装方式一样，具体操作请看上一节的MarkdownEditing。这里不作讲述。

OmniMarkupPreviewer的常用快捷键

- Ctrl + Alt + o: 在浏览器中预览
- Ctrl + Alt + x: 导出HTML
- Ctrl + Alt + c: HTML标记拷贝至剪贴板

OmniMarkupPreviewer自定义配置

从菜单栏上选择Preferences->Package Settings->OmniMarkupPreviewer下的相应文件进行配置即可。

Markdown Usage

导语

笔者在这里简单整理总结一下Markdown的一些基本用法。

Enter

回车

规则	效果	示例
回车一次	视为空格	-
回车二次	视为换行	-
行尾加两个空格并回车	视为换行	-

Italics bold

Header

Unordered list

Link

Image

Code

Table

Blockquotes

Horizontal Rule
