layout	title	subtitle	date	author	header-img	catalog	
post	Atom、Markdown与github	proxy，SocksCap64，pac	Tue Mar 21 2017 08:00:00 GMT+0800 (中国标准时间)	elmagnifico	img/git-head-bg.jpg	true	

Markdown

博客的文章一直都是Markdown格式的，而且用多了，基本格式语法什么的都记住了，也很好用。

再加上github一直有对Markdown的拓展，让其内容更加丰富了，所以选择使用Markdown感觉很好。

不像Latex什么的格式那么复杂，而且还有各种模板什么的。

MarkdownPad2

之前一直用 MarkdownPad2 来写博客的文章。

但是 MarkdownPad2 有一个不好的地方，每次都需要安装 awesomium_v1.6.6_sdk_win 才能正常运行。

除了这一点以外，其实格式也有点问题.

比如，我文章的开头的这个部分，其实markdown是不能解析的，因为这个部分是给标题用的。

MarkdownPad2 呢就会自动显示成一团乱七八糟的结构，虽然最后push上去以后没啥问题，但是预览就会显示的很难受。

```
---
layout:     post
title:      "Atom、Markdown与github"
subtitle:   "proxy，SocksCap64，pac"
date:       2017-03-21
author:     "elmagnifico"
header-img: "img/git-head-bg.jpg"
catalog:    true
tags:
    - git
    - blog
    - 代理
---
```

除了这个以外，下面这种包含头文件，他也会解析成一级标题，就算我这里是把他变成了代码格式，预览里显示的时候就变成了一级标题，特别显眼，同样的push以后就没有问题，完全是MarkdownPad2 的解析问题。

```
#include<iostream>
```

sublime text 3

其实这个编辑器也很有名，也是神器级别。

但是，新版的 sublime text 3 实在是太麻烦了。就为了装一个Packages Control，死活装不好，装上了，不显示包安装器。

可以用快捷键强行呼出，但是呼出以后的结果却是什么包都搜不到，也不能安装。

简直不能再难用了，搜解决办法的时候出来了 Atom，看到了一个打字特效，感觉好牛逼的样子。

果断放弃了 sublime text 3，转投 Atom 怀抱

> https://segmentfault.com/q/1010000004189036/a-1020000004189105

Atom

Atom 是 Github 专门为程序员推出的一个跨平台文本编辑器。具有简洁和直观的图形用户界面，并有很多有趣的特点：支持CSS，HTML，JavaScript等网页编程语言。它支持宏，自动完成分屏功能，集成了文件管理器。

其官网：A hackable text editor for the 21st Century

> https://atom.io/

很简洁，二十一世纪的文本编辑器。

某种程度上说和 sublime 非常相似。下载以后直接打开就自动安装好了，完全无法选择安装目录，还好的是，本体不算很大，自动安装就自动安装吧。

关键特效才是最重要的啊。

- 打字特性

- 连击特效

Activate Power Mode

官方地址：

> https://github.com/JoelBesada/activate-power-mode

安装其实很简单，但是国内貌似就是不好用。

一般的安装方法，直接从settings中的packages里搜索或者是install里搜索，讲道理都应该能搜到然后直接安装就行了。

然而并不能，那只好麻烦一点了。

方法一

直接从其github上下载源文件，然后解压放入下面的目录：

```
C:\Users\你的用户名\.atom\packages
```

方法二

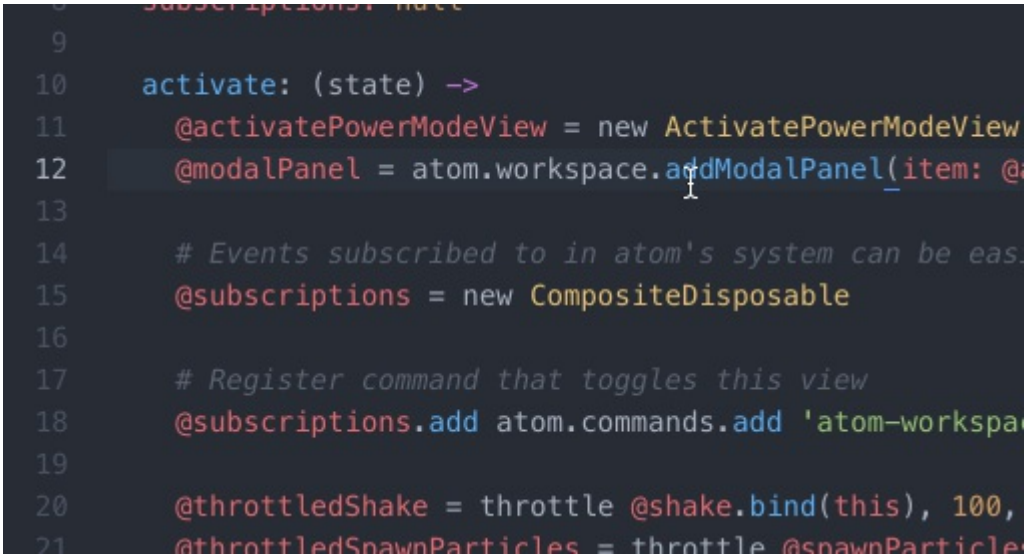
用管理员模式运行PowerShell，不是cmd！！！

cmd无法正常安装，我已经试过了。

```
apm install activate-power-mode
```

大概等个一两分钟，就会有一个done提示，那就安装好了。

默认是开启的，所以立马就有特效了



设置

可以从packages中搜索 Activate Power Mode 然后就能点进去看他的设置了。

除了粒子特效意外，还可以配合震动、声音特效。

声音里有打字机按键的那种声音，也有一个枪声，当然也能自定义，只是自定义的需要考虑到连击的时候声音是否能正常播放。

Activation Threshold

这个可以调节按键多少次以后出现粒子特效，默认是50，其实我感觉1就可以了，直接打字就是特效

Screen Shake

默认是1-3的震动强度，但是我感觉1-3有点眼瞎，震动幅度太强了。

所以我现在尝试的是0.1-1，这样震动幅度不强，但是又能看出来，感觉还是比较好的。

Play Audio

可以设置声音，其实自己用的机械键盘完全没必要听这个声音...

另外一个枪声，实在是有点刺耳，感觉不舒服。

Particles Size

可以调节例子特效的那个小球的最大和最小值

Particles Spawn Count

可以调节每次出现的粒子数量

Particles Colours

粒子特效的颜色，这个很重要。

如果你是深色的编辑背景，那么什么颜色都很明显。

但是如果你是白色的背景，那么其实很多颜色都很淡，不是很明显的。

一共有三种特效，一个是根据你文本颜色来决定粒子颜色。

一个是固定颜色的粒子特性，还有一个是随机颜色的粒子特性

快捷键

首先是activate-power-mode:toggle 他决定右上角的Combo是显示还是隐藏，快捷键是ctrl-alt-o

然后是activate-power-mode:reset-max-combo 他可以把Combo技术清零，没有设置快捷键。

于是我就设置了如下快捷键

```
'atom-workspace, atom-workspace atom-text-editor':
  'F5': 'markdown-preview:toggle'
'atom-workspace':
  'ctrl-alt-p': 'activate-power-mode:reset-max-combo'
```

Atom的快捷键设置很有意思，给你看的快捷键都是系统固定的，你想要自己定义就单独复制出来，然后去keymap里设置一个其他的。

由于Atom的Markdown的预览是基于每个文件独立的，所以每次开一个文件要看预览就得，很复杂的操作一通，感觉很蠢。

我就把 markdown-preview:toggle 设置为了F5，这样每次按一下刷新，就能看到对应的预览界面的，很舒服

汉化包

```
apm install atom-simplified-chinese-menu
```

Atom 编辑器简体中文包：汉化菜单栏、右鍵菜单以及大部分的设置项。

```
apm install simplified-chinese-menu
```

Atom 的简体中文汉化语言包,目前最全的汉化包。包含菜单汉化、右键菜单汉化以及设置汉化

汉化可有可无吧，顶多是汉化了默认的设置和默认的包，自己添加的还是看英文好点

markdown-pdf

有时候需要把内容导出成pdf或者什么jpeg或者png等图片格式，这个时候就需要用markdown转pdf了

> https://github.com/travs/markdown-pdf

今天网络极好，竟然settings中的install可以用了，直接搜索markdown-pdf就能安装了

安好以后设置了一下，其对应的快捷键，一键转换。

```
'.platform-win32 .editor, .platform-linux atom-text-editor':
  'F7': 'markdown-pdf:convert'
```

在markdown-pdf的设置中可以设置对应转换的格式应该是什么。

需要注意的是他需要其他组件支付，一个是 tree-view ，一个是 markdown-preview ,这两个安装Atom就自带了

但是其是这还不够，还有可能出现下面的问题

无法转换问题

```
markdown-pdf: AssertionError: html-pdf: Failed to load PhantomJS module. You have to s
```

遇到这个问题，其是是系统少了几个框架。

node.js

先要装 node.js ，其官网如下：

> http://nodejs.cn/

默认安装即可

phantomjs-prebuilt

cmd 命令行输入：(注意一定不要是管理员模式)

```
npm install phantomjs-prebuilt
```

如果上面安装不动，那么手动下载吧：

> http://phantomjs.org/download.html

把下的安装包扔到下面这个目录里

```
C:\Users\你的用户名\AppData\Local\Temp\phantomjs
```

再执行一次安装命令，很快就能安装完。

```
PhantomJS not found on PATH
Download already available at C:\Users\ELMAGN~1\AppData\Local\Temp\phantomjs\phantomjs
Verified checksum of previously downloaded file
Extracting zip contents
Removing C:\Users\elmagnifico\node_modules\phantomjs-prebuilt\lib\phantom
Copying extracted folder C:\Users\ELMAGN~1\AppData\Local\Temp\phantomjs\phantomjs-2.1.
Writing location.js file
Done. Phantomjs binary available at C:\Users\elmagnifico\node_modules\phantomjs-prebui
C:\Users\elmagnifico
`-- phantomjs-prebuilt@2.1.14
...
  `-- which@1.2.14
    `-- isexe@2.0.0

npm WARN enoent ENOENT: no such file or directory, open 'C:\Users\elmagnifico\package.
npm WARN elmagnifico No description
npm WARN elmagnifico No repository field.
npm WARN elmagnifico No README data
npm WARN elmagnifico No license field.
```

Done 了就行了，剩下的警告无视就好。

这里 elmagnifico是我的用户名，而如果你用管理员模式安装，那么最后安装的路径是不对的，导致Atom实际上还是找不到你安装的phantomjs，我之前就以为管理员安装怎么都对，然后发现安装完根没安差不多。

错误安装如下：

```
Writing location.js file
Done. Phantomjs binary available at C:\Windows\system32\node_modules\phantomjs-prebuil
C:\Windows\system32
`-- phantomjs-prebuilt@2.1.14
  +-- es6-promise@4.0.5
  +-- extract-zip@1.5.0
  ...
  +-- request-progress@2.0.1
  | `-- throttleit@1.0.0
  `-- which@1.2.14
    `-- isexe@2.0.0

npm WARN enoent ENOENT: no such file or directory, open 'C:\Windows\system32\package.j
npm WARN system32 No description
npm WARN system32 No repository field.
npm WARN system32 No README data
npm WARN system32 No license field.
```

当然其实这种安装也不能说绝对错了，只是如果这么安装了需要你配置一下环境变量，才行。非管理员模式就不用配置环境变量。

重开一下Atom，然后再次转换，应该就不会有上面的错误提示了。

image format issue

```
Uncaught Error converting to image format. Check console for more information.

C:\Users\********\.atom\packages\markdown-pdf\lib\markdown-pdf.js:166

The error was thrown from the markdown-pdf package. This issue has already been report
```

其实这个转换器还是有一个问题，我刚好也遇到了，如果遇到图片，而且格式什么的还有问题，那么就自然会发生这个错误。

比如本文就有好几个动态图，根本无法转换，其他没图的就没事，点开View Issue，可以看到也有很多人都有这个问题。

所以，markdown-pdf 就介绍到这里，如果没有发生错误的情况下，可以用他，所见即所得，很好。

当然其实图片也不一定一定会出错，同一个文档有时候生成就出错了，有时候又不出错，非常奇怪，感觉生成不是很稳定。

markdown-themeable-pdf

有了上面的经验，用 markdown-themeable-pdf 就很轻松了。

但是他虽然没有了图片转换的问题，他转换的结果和 markdownpad 2 基本是一致的，这就导致所见不是所得

当然也只是部分格式可能和 markdown-view 里所看到的不一样，大部分还是一样的。

所以其实也不是太完美，他转换完成以后会需要自动打开pdf然后看一下，其实就需要下面的这个插件支持了

除了上面说的问题，他的生成pdf的快捷键修改以后没有反应，这个我也不知道为什么，其他的都可以就他的不行。

```
'atom-workspace, atom-workspace atom-text-editor':
  'ctrl-shift-E': 'markdown-themeable-pdf:export'
```

pdf-view

pdf-view 支持pdf预览，这样转换完成以后就可以直接打开看是否满意了。

遗留问题

```
> https://segmentfault.com/q/1010000004189036/a-1020000004189105
>
> https://github.com/JoelBesada/activate-power-mode
>
> https://ninghao.net/blog/2073
```

每次引用的网页都得这么写，不然就会两个引用占用一行，又或者是一个引用一行，我也不知道是什么原因。

之前用 MarkdownPad2 写了第一个引用以后回车会自动出现 > 而Atom里明显没有这个设置，每次都得自己写，有点蛋疼。

文本编辑器里基本都带有圈选然后鼠标拖动的功能，但是Atom里不行，我写错了位置想拖动到其他

地方就不行。

这几个问题看以后能遇到什么解决办法不。

其实还有一个后来发现的问题，就是如果开着preview，打字输入，有明显的迟钝感，简单说就是渲染的时候很卡，关了preview以后明显就好多了。

Activate Power Mode 的特效反而不是特别卡，就是实时渲染特别卡，当然目前的机器只有集显，但是对于渲染一个文本来说还是足够的，所以只能说他的优化还是不够好，至少MarkdownPad2 是没有出现过这种打字都慢一拍的情况。

两者同时开的情况下，基本肉眼可以看着字蹦出来的慢动作，感觉要死。

Activate Power Mode 如果在高速输入的情况下也是一样的，会卡，会输入延迟，感觉作者可能也不会优化，特效里最卡的地方就是震动，这个震动涉及整页的渲染，所以会特别卡，一直长按回车大概就能看出来了。

所以最好是把震动特效关闭了，这样就能不卡很多

Quote

> https://segmentfault.com/q/1010000004189036/a-1020000004189105
>
> https://github.com/JoelBesada/activate-power-mode
>
> https://ninghao.net/blog/2073