

尝试一下 Pelican

似乎一夜之间所有的极客们都有了 自己的 Github 主页 和 Octopress 博客。就像所有人在他们的博客中指出的，静态博客的确比传统的 WordPress 方式具有更多优势。自从看到这些 我就一直在想着自己搭一个 Octopress。

但是似乎 Octopress 不适合我

一上手就被 Octopress 的搭建步骤 烦到了。RVM 是什么？rbenv 又是什么？看来 Ruby 社区的快节奏发展已经超过了我的想象，他们似乎需要一套发行版管理器来调和不同版本之间的 Ruby 的兼容性问题。虽然同样的兼容性问题在 Python 社区也有 [1]，不过总觉得 Python 至少还没到需要一个发行版管理器的程度 [2]。

真正的问题是我手上还没有一个可以让我随便玩的 Linux 环境（真的想要……）。而无论是 RVM 还是 rbenv 似乎都只支持 Unix/Linux/macOS。身为极客就注定不能用 Windows 么？（或许是

的……)。

剩下的问题就是 Ruby 和 Python 两大阵营的对立问题了。我不熟悉 Markdown，相对来说比较喜欢 ReST。似乎无论哪边都要依赖 Pygments 作为代码着色器，那么其实 Rubyist 也至少需要安装 Python。我倾向于不依赖任何 Ruby 组件，最好没有 C 扩展的纯 Python 实现。

于是我开始在 Github 上找 Python 的静态博客引擎。Flask 的作者 mitsuhiko 写的 rstblog 看起来不错，不过似乎没有多少人在用。Hyde 似乎很完善，不过默认标记语言是 MarkDown，又依赖于几个 Ruby 组建，而且官方网站的设计实在太前卫。最终我看到了 Pelican。

- | | |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [1] | 比如 Python 2.x 与 3.x 之间看似难以跨越的鸿沟，以及 <u>PyPy</u> 、 <u>CPython</u> 、 <u>Stackless</u> 、 <u>Cython</u> 等各个实现之间的微妙差别。 |
| [2] | 是的，我们有 <u>easy_install</u> ，我们有 <u>pip</u> ，不过这些都是包管理器，都是装好特定的 Python 实现之后的事情。Python 实现本身还不需要包管理器来管理。Python 的版本问题基本上也只需要 <u>2to3.py</u> 和 <u>3to2.py</u> 这样的轻量级转换器就可以了，你不需要为了安装多个软件而在硬盘里留下多个不同版本的 Python。如果为了引用的稳定性，你可以用 <u>virtualenv</u> ，不过这又是另一回事了。 |

那么就 Pelican 吧

对我而言，Pelican 相比于 Octopress 有几个好处：

1. 纯 Python 实现。这意味着我可以换用任何 Python 解释器而不必担心兼容性问题。比如我就换成了 PyPy。
2. 多语言支持。因为 Pelican 的作者似乎是个法

国人。不过这个似乎大部分人不需要…… 我是想尽量把一篇博客写成三种语言作为锻炼吧。

3. ReST。这样我就可以用 Leo 的 @auto-rst 直接写 ReST了。简单方便快捷有效。

不过似乎 Pelican 的关注度不如 Octopress 那么高，现在一些部分还有细微的问题：

1. pelican-import 从 WordPress 导入的时候对中文、日文的支持似乎很成问题。
2. 日期格式、时区、字符集、和多语言功能的结合度还不够。 **我在尝试改善它。**
3. 模板还不够丰富。
4. 插件也不够多……

希望这么优秀的工具能够受到更多关注，以上这些问题都是增加关注度之后很快就能解决的问题。

我的设置 settings.py

安装 Pelican 很容易，一句话就够了：

```
1 $ pip install pelican
```

然后把文章写成ReST的格式，放在`pages`文件夹里面。(重新)生成只要：

```
1 $ pelican -s settings.py
```

上传到 Github:

```
1 $ git commit -am "Commit message"
2 $ git push
```

就这么简单。附上我的配置文件：

```
1 # -*- coding: utf-8 -*-
2
3 TIMEZONE = 'Asia/Tokyo'
4
5 DATE_FORMATS = {
6     'en': ('usa', '%a, %d %b %Y'),
7     'zh': ('chs', '%Y-%m-%d, %a'),
8     'jp': ('jpn', '%Y/%m/%d (%a)'),
9 }
10 # windows locale: http://msdn.microsoft.com/en-us/library/ax410z%28VS.71%29.aspx
11 LOCALE = ['usa', 'chs', 'jpn',    # windows
12           'en_US', 'zh_CN', 'ja_JP'] # Unix/Linux
13 DEFAULT_LANG = 'zh'
14
15 SITENAME = 'Farseerfc Blog'
16 AUTHOR = 'Jiachen Yang'
17
18 DISQUS_SITENAME = 'farseerfcgithub'
19 GITHUB_URL = 'https://github.com/farseerfc'
20 SITEURL = 'http://farseerfc.github.com'
21 TAG_FEED = 'feeds/%s.atom.xml'
22
23 SOCIAL = (('twitter', 'http://twitter.com/farseerfc'),
24           ('github', 'https://github.com/farseerfc'),
25           ('facebook', 'http://www.facebook.com/farseerfc'),
26           ('weibo', 'http://weibo.com/farseerfc'),
27           ('renren', 'http://www.renren.com/farseer'),
28           )
29
```

```
30
31 TWITTER_USERNAME = 'farseerfc'
32
33 THEME='notmyidea'
34 CSS_FILE = "wide.css"
35
36 DEFAULT_CATEGORY = 'Others'
37 OUTPUT_PATH = '.'
38 PATH = 'posts'
```