我是如何用 Pelican 搭建博客的 🔗

目录

- □□

2015年2月14日更新：

□□: □□□□□□□□□□□□ [1]

□□□□□□□□□□□□□□□□□□ □□□□ □□□□□□□□□□ □□□□ □□□□□□ □□□□□□□□□ □□ □□□□□□□□□□□ Bootstrap 2 □□□□ □□□□□□□□□□□□□□□□□□□□□□□□□ drop-in □□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□ Bootstrap □□□□□ v3.3 , 和 Pelican □□□□□□ 3.5 □□ □□ □□ Bootstrap 和 Pelican □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□ □□□ □□□□□□□□□□□□ Pelican □□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□

□□□□□□□ ICSR 2015 □□□□ □□□□□□□□□ Pelican □□□□

Bootstrap 3 的更新

- 全新的 mobile-first（移动 优先） 以及 responsive（响应式）：Bootstrap 2的响应式特性是附加的 功能，而现在已经将移动端 的支持直接融入到框架的每一处 hack、@media 以及相关组件 中了，mobile-first（移动 优先） 以及 responsive（响应式）grid system（栅格 系统）是整个框架的核心，这也是 博客们对其褒奖有加的地方。以前也有关注过响应式的技术，但是没有 blog 的需求，所以并没有深入研究，感兴趣的可以看看这一篇 《Issue 响应式》
- 全新的 Navbar（导航 栏） ： Bootstrap 2 中被大家喜爱的导航栏 变得 更加灵活、强大，并且 sticky（固定）在顶部的问题也得到了修复。

  具体可以参看 Bootstrap 3 文档 。

Pelican 3.5 的更新

- Python 2 和 Python 3 都可以使用 ，完全支持 unicode 编码（对于中文写 blog 的同学是个好事），并且支持 pypy 以提升运行效率。
- 可以很容易的通过命令行创建一个 博客 并发布。
- 具有像主题、插件等等丰富的生态系统，向 wordpress 之类的博客系统看齐（好吧……）

- ⬛⬛⬛⬛ ⬛⬛⬛ ⬛⬛⬛ ⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛ pelican ⬛⬛⬛⬛⬛⬛⬛⬛
  hard code
  ⬛ ⬛⬛ ⬛ ⬛⬛ ⬛⬛⬛⬛

  plugin theme

⬛⬛⬛⬛⬛ Pelican ⬛⬛ ⬛

⬛⬛⬛⬛⬛⬛⬛

Pelican ⬛⬛⬛⬛⬛⬛

```
.
├── cache          ⬛⬛⬛⬛⬛ pickle ⬛⬛
├── content        ⬛⬛⬛⬛⬛⬛⬛
│   ├── <categories>    ⬛⬛⬛⬛⬛⬛⬛
│   ├── pages       ⬛ About ⬛⬛⬛⬛⬛⬛⬛
│   └── static      ⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛
├── drafts         ⬛⬛⬛⬛⬛⬛
├── Makefile       ⬛⬛⬛⬛ makefile
├── pelicanconf.py    ⬛⬛⬛⬛⬛⬛⬛ Pelican ⬛⬛
├── publishconf.py    ⬛⬛⬛⬛⬛⬛⬛ Pelican ⬛⬛
├── output       -> ../farseerfc.github.io
├── plugins      -> ../pelican-plugins
└── theme        -> ../pelican-bootstrap3
```

⬛⬛⬛⬛⬛ ⬛⬛⬛⬛ github ⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛ Pelican ⬛⬛⬛⬛⬛⬛ ⬛⬛⬛⬛⬛ ⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛ ⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛ ⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛ 4 ⬛ git repo ⬛⬛⬛⬛⬛⬛⬛

⬛⬛⬛⬛⬛ ⬛⬛⬛⬛⬛⬛⬛ repo ⬛ ⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛ ⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛

⬛⬛⬛⬛⬛ repo ⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛ repo⬛⬛⬛ git submodule ⬛⬛⬛⬛⬛ ⬛⬛⬛⬛⬛⬛ theme ⬛⬛ pelican-bootstrap3 ⬛⬛⬛⬛⬛⬛⬛ pelican ⬛⬛⬛ plugins

□□ pelican-plugins 里面的 plugins 其实只是一个索引，真正的 plugin 在子项目引用里面。我随手改了几个明显的 bug，但是由于这些 plugin 还在用 python 2，懒得 调试 output 里面 farseerfc.github.io 上留下的错误，就放弃了。

　　于是就 自己 撸 了 一个 比较偷懒的办法绕过了。

切换到 Material Design 风格的 Bootstrap 3

　　然后 想起 来之前看到过这么个有意思的 CSS 框架，于是稍微读了读 bootstrap-material-design , DandyDev/pelican-bootstrap3 和 Bootstrap 3 的文档，略微魔改了一下 弄成了现在的样子。 魔改的办法还是通过上述的偷懒方式，在现有的 pelican-bootstrap3 的基础 上面 加载 额外的 Bootstrap3 主题 和 Material 的主题

用 Bootstrap 3 做框架

　　现在这个站点是用 Bootstrap 3 的栅格系统布局的，说是响应式 Web 设计其实也就是大屏幕看起来 是 双栏而已，手机上看就变成单栏的了。

用引用框强调一下

> 用的是 Bootstrap 3 默认的引用框样式

```
1   @screen-xs:    320px;
2   @screen-sm:    598px; /*  768px; */
3   @screen-md:    952px; /*  992px; */
4   @screen-lg:   1350px; /* 1200px; */
5   @screen-xl:   2030px;
6   @container-sm: 582px; /*  750px; */
7   @container-md: 930px; /*  970px; */
8   @container-lg: 1320px; /* 1170px; */
9   @container-xl: 1990px;
```

我们在 Bootstrap 3 的基础上新增了 ░░░░░░░░░░ ░░░░░░░░░░░░ xs 和 sm 断点，░░░░░░░░░░░░ ░░ 。 ░░ ░░░░░░░░░░░░ md ░░░░，Nexus 7 是 960 ░░░░ ░░ ░░░░░░ 1024 ░░░░ lg ░░░░░░░░░░░ 1366 ░░░░░░░░

░░ Bootstrap 3 ░░░░░░░░░░░░░░░░ 1200 ░░░░░░░░░░ ░ ░ 1600、2048 ░░ 2560 ░░░░░░░░░░░░░░░░░░░░░░░ ░░░░░░░░░ ░░░░░░░░░░░░░░░░░░ ░░░░░░░ ░ bootstrap ░░░░░ ^{bigger than bigger} ░░░░ ░ ░ xl ░░░░░░░░░░░ 2048 ░░░░░░░░░░░░░ variables.less ░░ ░░

░░░░░░░░░░░░░░

░░░░░░░░░░░░ Google+ ░░░░░░░░░░░░░░░░░░░░░░░ ░░░ ░░░░░░░░░░░░░░░░░░░░░░░░░░░░░

xs ░░░ ^{fluid} ░░ ░░	sm ░░░░░░░░░░░░░ ░ ░░░ ^{sidebar} ░░░░	md ░░░░░░░░░ ░░░ ░ ░░░

Navbar 标题栏	标题栏	标题栏
标签	标签	标签1 / 导航栏1
导航栏	内容块1 / 内容块2	标签2 / 导航栏2
内容	footer 内容	footer 内容

lg 屏幕尺寸阈值为大尺寸 栅格 布局如下：

xl 屏幕尺寸阈值为超大尺寸 栅格 布局如下：

标题栏		
内容1	内容3	导航栏1
内容2	内容4	导航栏2
footer 内容		

标题栏			
内容1	内容3	内容5	导航栏1
内容2	内容4	内容6	导航栏2
footer 内容			

　　由于我们使用 Bootstrap3 的栅格系统来进行整体的页面布局，要想布出上面所示的各种 布局效果，同样需要书写大量的样式代码，所以我们将这些样式写成了 CSS样式文件，放在 了本章的源码文件中，名称为 waterfall.less，大家可以根据需要自行查看。

欢迎信息栏

　　欢迎信息栏是页面上方一个较为醒目的区域，通常用来显示一些欢迎信息或者系统公告等 内容。我们同样可以使用栅格系统来对其进行布局，这里我们使用了 Bootstrap 的 .jumbotron 组 件，它 可以 帮助我们 快速地实现欢迎信息栏的布局。

□□□□□□□

　　□□□□□□□□□□□□□□□□ @brand-primary: darken(#6B5594, 6.5%); □□□□□□□□□□□□ □□□□□□□□□□ □□□□□ .btn □ white-space: nowrap; □□□□□ □□□□□□ □□□□□□□□□□

2015□1□29□□□

　　□□□□□□□ Bootstrap 3 □□□□□□□ □□□□□ □ □□□□□□□□□□□□□□□□□ javascript □□□ bug …… □□□□□□□□□ □□□□□ variables.less □ @grid-float-breakpoint □ @grid-float-breakpoint-max □□□□0□□□□□□

□ bootstrap-material-design □□□

　　□□□□□□□□□□□□□□□□□□□□□□□ .btn □□□□□□□□ □□□□□□□□□□□□□ □□□□□□□□□□ Win9x □□□□□□□□□□□□□□□ □□□□□□□□□□□□ .btn □□□□□ □□□□□□□□□□□□□□□□□□□□

　　□□□

□□□□□□□□□ pelican-bootstrap3 □

Pelican □□□□□□□□□

　　□□□□□□□□□ Pelican □□□□□□ OUTPUT_SOURCES □□□□□□□□ □□□□□□

```
1  OUTPUT_SOURCES = True
2  OUTPUT_SOURCES_EXTENSION = '.rst'
```

　□□ Makefile □□ pygmentize □□□□□□□□□□□

```
1   find -iname "*.rst" | parallel -I@  pygmentize -f html -o
    @.html @
```

然后在模板中添加一段 jQuery 代码即可：

```
1   <a onclick="$.get('{{SITEURL}}/{{article.slug}}.rst.ht
    ml', function(data){$('#source-code').html(data)});$('#arti
    cle-content').toggle();$('#source-content').toggle();">
```

有兴趣的可 hack 一下此网站的源代码。

我从 pelican-bootstrap3 进行 fork 然后做了自己的修改，目前已经添加了相当多的新特性了，与原有的 pelican 和 bootstrap 2 主题相比要丰富太多了，下面 大概罗列 一下已经实现 和将要实现的新特性和细 节（实时更新）：

1. 利用 bootstrap-material-design 框架进行美化
2. 支持切换主题样式（目前有 Bootstrap 3 原生和质感设计两种）
3. 支持多语言和子站点（基于 i18n-subsite 插件）切换功能
4. 改善了原有主题的排版和样式
5. 添加现代化的评论和分享功能
6. 添加 Atom 订阅源和 breadcrumb 导航
7. 添加源代码和文章切换功能，此功能利用了 Bootstrap 的 modal 控件！
8. 添加置顶按钮 以及bootstrap风格 的翻页和标签样式等
9. 添加代码查看器 （有行号和复制功能）、源码 </> 查看等

附录: 关于 Pelican 和 reStructuredText 的语法

先说说各个博客框架的选择吧。

```
 1   PLUGINS = ["i18n_subsites",
 2           "plantuml",
 3           "youku",
 4           "youtube",
 5           'tipue_search',
 6           'neighbors',
 7           'series',
 8           'bootstrapify',
 9           'twitter_bootstrap_rst_directives',
10           "render_math",
11           'extract_toc',
12           'summary']
```

接下来我们分开看每一个插件负责的不同的功能。

i18n-subsites

这个插件的名字是 internationalization subsite 的 缩写 。

我们知道 Pelican 是一个静态博客生成器 ，而 Pelican 本身并没有多语言的机制，只是定义了 lang 这个变量标记文章的语言 ，同时借助 slug 这个变量标记具有相同内容的不同语言文章。于是链接 到 同一篇文章的 slug 相同而 lang 不同的时候 Pelican 会自动生成 指向其他语言的链接 。但 是 由于没有合适的组织形式，这些多语言的文章 只 能放 在同一个目录下面 ，同时 没有好的切换语言 的方式 。这相比于成熟的CMS的多语言网站建设方式来说非常简陋，不过由于 我是 单枪匹马一个 人维护自己的博客，没有必要引入非常复杂的 机制 以供维护。

于是我想到的多语言机制其实说起来也非常简单：分别针对每个语言单独生成一个网站 ，其中中文在 https://farseerfc.github.io/ 下面，而英文放在 https://farseerfc.github.io/en/ 下 。这样做的缺陷是两种语言的文章是分别独立提供，而 pelican 只能按照如下的方式运作：

1. pelican 扫描整个目录，找到所有文章
2. i18n-subsites 在扫描文章的时候 pelican 增加额外的处理步骤，按照 URL 将 不同语言 的 pelican 文章分别列入不同的列表
3. 生成网站所需要的公用 js 和 css 样式文件，并且拷贝公用的图片等等资源文件。同时将多

□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Pelican □□□□□□□□□□□□□□□□ □□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□ Pelican 3.4 □□□□□□□ API □ □□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□ Pelican □□□□□□□□□□ README.md□□ □

□□ □□□□□□□□
i18n-subsites □□□□□□□□□□□□□

1. □□□
2. □ jinja2 □ i18n □□□□□□ Python □ gettext □□□□□□□□□□□□□□ □□□□□□□
□□ □□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□ *.po □
*.mo □□□□□□□□□□□

□□□□ jinja2 □ i18n □□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□ □ □□□ □□□
SCons □□ □□□□□□□□□□□□□ po □ mo □□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Pelican 3.4 □□□□□□ □□□□□□□
□□□□□□ SITEURL □□□□□□ URL □□□□ RELATIVE_URLS □□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ URL □□□□□
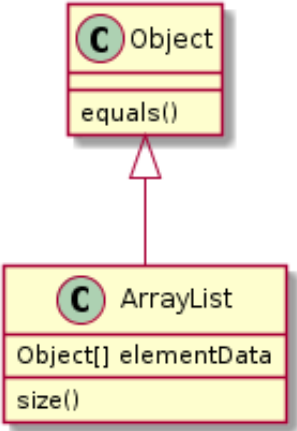
plantuml

□□ PlantUML □□□

PlantUML 是一个Java开发的 开源工具，它可以从文本绘制 UML 图或提供 GUI 编辑。它也可以被整
合进你的 Markdown、 reStructuredText、 AsciiDoc 甚至普通网页文档中。它通过 <u>directive</u> 名叫
为 plantuml 的指令工作。举个例子，在 reStructuredText 里添加 .. uml:: 之后加上你要插入
的代码，就用 plantuml 来渲染 并且把它添加进你的文章里。

下面的代码描述了 UML 图，其中画出了对象之间的继承关系。

```
1    .. uml::
2
3       Object <|-- ArrayList
4
5       Object : equals()
6       ArrayList : Object[] elementData
7       ArrayList : size()
```

我没有使用它很长时间，因为它不能很好的工作在 python2。具体来说它能工作在 python 2
和 3 里，但是它没有被正确的显示出来。我认为这是 pelican 的某些问题在于它的渲染 并给了错误的 URL
到了 i18n-subsites 插件。如果你想了解更多内容，请阅读 <u>这个开放的错误报告</u> 。

添加 Ditaa 的部分

plantuml 仅支持UML图，但是如果你需要画些其他的东西该怎么办呢？ 不用担心 我们还有 <u>ditaa</u> 这个

plantuml と同じように、 reStructuredText 文書内に 画像に変換したいテキストを ditaa の directive に渡す ことで使用できます。

```
 1   .. ditaa::
 2
 3                    +-------------+
 4                    |  ditaa    |-------+
 5                    |  Diagram   |    |
 6                    +-------------+     | PNG out
 7                          ^            |
 8                       | ditaa in     |
 9                       |          v
10      +--------+  +--------+----+   /----------------\
11      |      | --+  Pelican  +--> |             |
12      | Text |  +-------------+   | Beautiful Blog |
13      |Document|  |  !magic!  |  |             |
14      |    {d}|  |        |  |             |
15      +---+----+  +-------------+   \----------------/
16       :                      ^
17       |        Lots of work         |
18         +-----------------------------------+
```
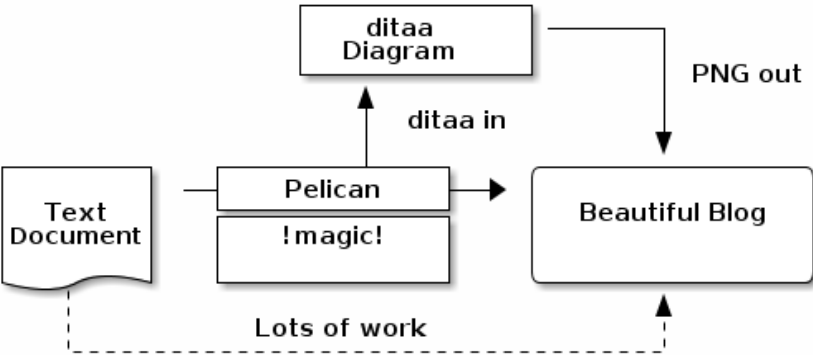
render-math

コンテンツです。

コンテンツ \(A_\text{c} = (\pi/4) d^2\).

数式です

\begin{equation*} \alpha{}_t(i) = P(O_1, O_2, \ldots O_t, q_t = S_i \lambda{}) \end{equation*}

コンテンツです。 reStructuredText 内に LaTeX で書いた数式を書くことができます。

:math: role ロール と .. math:: directive ディレクティブの両方を実装しています。 MathJax を使います。 ー MathJax の CDN を使うので、数式を表示するためにはインターネットへの接続が必要です。 オフラインで表示でき ない 点に注意。

youtube と youku

このプラグインを使うことで、 youtube と youku から、それぞれ youtube 動画と中国の動画サイト youku の動画を貼り付けられます。 ちなみに、 KDE5 Plasma のスクリーンキャスト 動画を貼ってみます。

tipue_search

Tipue search は、静的サイトでも全文検索が可能になる、 ある jQuery のライブラリです。記事の 全文データを含んだ検索用の json ファイルを、 あらかじめ 用意 して おく必要があります。日本語の文章を含むので、 この json ファイルは大変大きくなります。 したがって、ブログの記事が増えれば増えるほど、検索用ファイルのサイズが大きくなります。ちなみに、 この時点での json ファイルは 300KiB 程度。

このプラグインは、 pelican で記事をビルドするたびに、 beautifulsoup4 を使って全文データ を 取り出し、検索用 json を Tipue 用に 出力します。

neighbors と series

このプラグインを使うことで、 neighbors で前後の記事へのリンクを、 シリーズものの記事であれば、 それ に 加えて series で同じシリーズの前後の記事への リンクを、それぞれ記事中に 埋め込めるようになります。 たとえば、この記事の上端と下端を見てもらえれば、リンクがあるとわかるでしょう。

bootstrapify と twitter_bootstrap_rst_directives

就正确地显示出来了 这 要感谢 Bootstrap 的帮助了

bootstrapify 这个扩展用到的技术借助了 beautifulsoup4 去分析生成的页面并且找到其中 的 table , img , embed , iframe , video , object 等标签然后分别 加上需要的 class样式 之后就好看了

twitter_bootstrap_rst_directives 这个扩展提供了一系列

reStructuredText 的 _{role} 指令 和 _{directive} 指令 让 上面提到的 那些样式 可以用在 _{role} :kbd: 可以写 Ctrl+C 这样的键盘按键 然后 :code: 可以写代码然后 :glyph: 可以写图标字体 然后还有 很多 labels 像这样 和 alerts 像这样 和 panels 像这样 还 可以像这样 写 media _{directive}像这样 等

用上面的 panel 还有栅格系统我才能写出这一页 用 lg 或者 xl 的屏幕看应该是并排的 \(\frac{1}{2}\) 和 \(\frac{1}{3}\) 大小的分栏的布局 用小屏幕看的话应该是顺序排列

顺便我还修改了 twitter_bootstrap_rst_directives 添加了一些它还没有但是我需要的功能 比如 上面用到 的 _{role} :ruby: 写出 html5 的 <ruby> 标签这个需要新版本浏览器（firefox）才能 显示 所以用其它浏览器的话可能看 不到 :html: 写 一段 可以 写 html 标签然后像 Markdown 那样嵌入到 reStructuredText 中去当然这样写的代码就和整个文档系统绑定了 所以 应该 谨慎使用

又给 twitter_bootstrap_rst_directives 添加了两个 _{role} 一个 是 用来 :twi: 表示的 twitter 用户名比如我的 @farseerfc 另一个是 :irc: 用来写 freenode 的 channel 比如 #yssyd3 这

又加入了 .. friend:: 指令用来写友情链接 fref 用来在文章中 引用 LQYMGT 的话像

extract_toc 和 summary

下面介绍两个我自己写的小扩展

reStructuredText 本身支持目录树 使用 .. contents:: 指令产生的目录树是由 docutils 渲染产生的 所以如果把整篇文章放在首页的话就会有 多个 extract_toc 这个扩展的功能就是 把其中的第一个目录树抽离出来放到页面的侧栏

□□□□□□□□□□□ panel □□

□□ Pelican □□□□□□□□□□□ ^summary^ □□ □□□□□□□□□□□ Pelican □□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□ □□□□ summary □□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□ summary □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□ □□□□□□□□□□□□

> **2015□1□29□□□**
> □□□ extract_toc □□□□□□□□□□□□□□□□ Bootstrap affix □□□□ □□□□□
> □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Bootstrap 3 □ Affix □□ □□□□□ □
> □□□□□□□□ □

□□

□□□□□□□□□ github □□□ □□□□□ □□□□□□□□□ Makefile □□□□□□□□□□
□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□……□

□□□□□□□□□□□□□□ Kansas Linux Fest fork □□□□□□□□□□□□□□□
□□□□□□□□□ Bootstrap 3 □ bootstrap-material-design □□□□□□□□
□□□□□□□□□□□□□□□□□□ □□□□□□□□□□ commit □□□□

| [1] | □65:17□□□□□□□□□□21:5□□□□□□□□□□ |