
Docs like Code

发布 1

2018 年 08 月 31 日

1	文档代码化	3
1.1	Amazon jekyll doc	3
2	轻量级文档解决方案的优势	5
2.1	优势一	5
3	基于Sphinx的项目示例	7
4	Sphinx快速入门	9
4.1	安装Sphinx	9
4.2	快速新建项目（以Mac系统为例，Window系统类似）	9
4.3	往项目中添加内容	10
4.4	修改主题	10
4.5	安装ReadtheDoc同款主题	11
4.6	实现帮助文档公网可访问	11
4.7	由ReadtheDocs执行发布命令	11
5	使用rinoH输出 PDF	13
5.1	rinoH插件安装	13
5.2	发布单个PDF	13
5.3	发布项目	13
6	使用LaTeX输出PDF	15
6.1	前提条件	15
6.2	基本步骤:	15
6.3	使用XeLaTeX	15
6.4	rST和latex的类比	16
7	reStructuredText简介	19
8	rst 文件直接发布	21
9	rtd theme 配置	23
9.1	主题配置	23
10	Jinja 模板语言	25
11	Markdown+Jekyll+Github Pages	27

11.1	Github Pages 托管静态页面	27
11.2	使用Jekyll Now迅速部署自己的博客	28
11.3	完整版jekyll安装	28
11.4	添加内容并发布新网站	28
12	修改发布主题	29
12.1	主题网站地址:	29
12.2	安装基于gem的主题	29
12.3	商业模板	30
12.4	在 github pages 上添加主题	30
12.5	制作Jekyll主题	30
13		31

本书将介绍一种轻量级的文档解决方案，所用工具均为成熟的开源技术。阅读本书后，读者可以用本书的方法，轻松为公司的产品或服务提供所需的帮助文档。

适合读者：

- 软件文档项目团队
- 小型写作团队

本书用于《技术传播方法》课程的教学，因为服务于技术写作课程的教学，本书将采用当前流行的 *Doc like Code* 的模式进行写作。

所用的工具如下：

1. 内容写作。采用轻量级的标记语言：[reStructuredText](#)
2. 文档发布工具。[Sphinx](#)
3. 协同与版本控制。[github](#)
4. 文档托管。[Read the Docs](#)
5. 写作工具。[MS Visual Studio Code](#)

本课程选课同学将按照技术写作的流程，参与到教程的写作和审校工作中。

文档代码化是指借鉴代码开发的方式开发文档，这样做有很多优势。

1. 代码与文档有机结合。
2. 软件工程师也可以贡献部分文档
3. 低成本。

1.1 Amazon jekyll doc

亚马逊的文档也是基于这一套方式开发，代码的可以在 `github repo` 中找到。

其他几个基于Jekyll开源的文档主题：

- <http://bruth.github.io/jekyll-docs-template/>
- <https://github.com/clojurewerkz/docslate>
- <https://github.com/tomjoht/documentation-theme-jekyll>
- <https://github.com/stitchdata/docs#stitch-documentation>

更多网站，请见：<https://cloudcannon.com/jekyll/2015/06/12/whos-using-jekyll.html>

轻量级文档解决方案的优势

2.1 优势一

1. 可用性高
2. 部署快捷
3. 二次开发容易

基于Sphinx的项目示例

有如下项目使用Sphinx工具进行文档发布和管理

- Flask
- Jinja

date 2018/04/12

author 高志军

Sphinx是一个静态网页发布工具，可将rST和md文件，发布为各类常见的用户帮助如联机帮助，用户手册等。

4.1 安装Sphinx

4.1.1 Mac系统

在Terminal中运行如下代码

```
pip install Sphinx
```

4.1.2 Windows系统

1. 需要首先安装pip，可参考 [pip](#)。
2. 运行如下代码

```
pip install Sphinx
```

4.2 快速新建项目（以Mac系统为例，Window系统类似）

1. 在桌面上创建一个文件夹，并命名为 *sphinx-demo*
2. 在Terminal中浏览至上述文件夹，并运行命令： `sphinx-quickstart`

3. 在对话框式的选择中，Y/N的选项，选Y；如果询问配置，直接复制[]中的内容，如[.rst]，则填写.rst
4. 新建成功后，则会得到如图所示的文件夹结构

```
.
├── build
├── make.bat
├── Makefile
├── source
│   ├── conf.py
│   ├── index.rst
│   ├── _static
│   └── _templates
```

4.3 往项目中添加内容

1. 浏览至 *source* 文件夹，并在其根目录下创建新文件夹demo
2. 在上方 *demo* 文件夹中，新建test.rst文件，并在其中输入如下内容：

```
=====
这是Sphinx的测试
=====
我爱学习Sphinx
```

1. 打开source文件中的 index.rst，将test.rst的文件添加至目录中，具体如下：

```
.. toctree::
   :maxdepth: 1
   :caption: Contents:

   demo/test.rst
```

2. 在Terminal中运行编译命令 sphinx-build -b html source build
3. 编译成功的话，在 *build* 文件夹中则有刚才发布的网站

4.4 修改主题

1. 打开 *source* 文件夹中的conf.py，并找到主题配置行 html_theme = 'alabaster'
2. 从内置主题中挑选需要的主题，如 bizstyle，将其改为 html_theme = 'bizstyle'
3. 重新运行发布命令后，则可得到新主题的样式的帮助文档

注解: Sphinx内置主题的风格可见: <http://www.sphinx-doc.org/en/master/theming.html#using-a-theme>。还可以安装其他主题, 或者按照需要制作自己的主题。

4.5 安装ReadtheDoc同款主题

如果喜欢 readthedocs.org 的主题, 可以按照如下方式安装

```
pip install sphinx_rtd_theme
```

安装之后, 再按照上述步骤, 将 `conf.py` 中的主题行, 修改为 `html_theme = 'sphinx_rtd_theme'`, 再运行 `sphinx-build` 命令重新发布即可。

4.6 实现帮助文档公网可访问

执行 `sphinx-build` 命令后, `sphinx` 会将 `rst` 的内容, 发布为静态网站。只需将 `build` 文件夹中的文件, 托管至 `github`, 即可实现公网访问。

4.7 由ReadtheDocs执行发布命令

每次更新后, 都需执行 `sphinx-build` 命令, 并重新上传至 `Github`, 较为麻烦。这个工作可以由 `ReadTheDocs` 平台自动化完成。

1. 注册 `ReadTheDocs` 账号
2. 将 `Github` 账号关联到 `ReadtheDocs`
3. 将 `source` 文件中的内容, 上传至 `github` 中的某个 `repo` 中
4. 选择 `github` 的相应 `repo`, 自动创建 `webhook`
5. 后续每次源文件内容有变化后, `ReadtheDoc` 均可以自动发布最新的版本

更多内容参见 `ReadtheDocs` 官方文档: https://docs.readthedocs.io/en/latest/getting_started.html

下次课内容

- `reStructuredText`
- 自定义主题
- 制作主题
- 发布为 `PDF` 等其他样式

预习:

- `HTML`, `CSS`
- `Jinja` 模板语言

参考资料

[`Sphinx` 官方教程]: <http://www.sphinx-doc.org/en/master/usage/quickstart.html>

演示本地修改, 自动发布。

使用rinohtype输出 PDF

Sphinx输出为PDF有多种方式，例如依赖LaTeX技术，现将rst转为latex，在转为PDF。现在也有多种其他方案，如Rinoh或rst2pdf等。本节将主要介绍rinohtype插件。。

5.1 rinohtype插件安装

首先通过pip，安装依赖库：

- docutils
- recommonmark ,

如果已经有的话，在Terminal中直接运行 `pip install rinohtype`，即可安装

5.2 发布单个PDF

1. 在Terminal中浏览至桌面上source文件夹，
2. 运行: `rinohtype index.rst` 即可发布为PDF

5.3 发布项目

5.3.1 配置conf.py

需要在conf.py中对“`latex_elements`”进行配置

```
latex_elements = {  
    # The paper size ('letterpaper' or 'a4paper').  
    'papersize': 'letterpaper',
```

(continues on next page)

(续上页)

```
# 'papersize': 'letterpaper',

# The font size ('10pt', '11pt' or '12pt').
#
# 'pointsize': '10pt',

# Additional stuff for the LaTeX preamble.
#
# 'preamble': '',

# Latex figure (float) alignment
#
# 'figure_align': 'htbp',
}
```

配置后, 需要增加

```
# Configure RinoH documents

rinoH_documents = [('index', # top-level file (index.rst)
                   'target', # output (target.pdf)
                   'Sphinx Quickstart', # document title
                   'Zhijun Gao')] # document author
```

配置后, 运行: `sphinx-build -b rinoH source build2`

参考资料:

[1]. <https://github.com/brechtM/rinoHtype>

6.1 前提条件

首先安装MacTeX, 安装步骤: `brew cask install mactex`

6.2 基本步骤:

1. 首先将sphinx 发布为 latex。 “ `sphinx-build -b latex source target`”
2. 运行 `pdflatex target` 将 latex发布为pd使用LaTeX发布PDF

6.3 使用XeLaTeX

XeTeX 也是TEX排版的一种, 支持Unicode而且也支持现代字体技术如, OpenType (OTF), TrueType (TTF), Graphite, and Apple Advanced Typography (AAT)。对应的编译器为 `xetex` 和 `xelatex`。

编译中文是, 在`conf.py`中的做如下设置:

```
latex_elements = {
    'papersize' : 'a4paper',
    'utf8extra' : '',
    'inputenc'  : '',
    'babel'     : r'''\usepackage[english]{babel}''',
    'preamble'  : r'''\usepackage{ctex}''',
}
```

1. `make latex`
2. 使用Texshop 打开进行上一步编译得到 `.tex`文件, 选择 ****XeLaTeX*** 引擎

3. 点击Typeset 即可得到中文版。

因为 readthedocs 上只有pdf_latex引擎，如果需要同时在readthedocs和本地化都能顺利编译中文pdf的话，可以在 conf.py 中添加如下配置：

```
import os

on_rtd = os.environ.get('READTHEDOCS', None) == 'True'
if on_rtd:
    latex_elements = {
        # The paper size ('letterpaper' or 'a4paper').
        #'papersize': 'letterpaper',
        # The font size ('10pt', '11pt' or '12pt').
        #'pointsize': '10pt',
        # Additional stuff for the LaTeX preamble.
        'preamble': r'''
\hypersetup{unicode=true}
\usepackage{CJKutf8}
\DeclareUnicodeCharacter{00A0}{\nobreakspace}
\DeclareUnicodeCharacter{2203}{\ensuremath{\exists}}
\DeclareUnicodeCharacter{2200}{\ensuremath{\forall}}
\DeclareUnicodeCharacter{2286}{\ensuremath{\subseteq}}
\DeclareUnicodeCharacter{2713}{x}
\DeclareUnicodeCharacter{27FA}{\ensuremath{\Leftrightarrow}}
\DeclareUnicodeCharacter{221A}{\ensuremath{\sqrt{}}}
\DeclareUnicodeCharacter{221B}{\ensuremath{\sqrt{3}}}
\DeclareUnicodeCharacter{2295}{\ensuremath{\oplus}}
\DeclareUnicodeCharacter{2297}{\ensuremath{\otimes}}
\begin{CJK}{UTF8}{gbsn}
\AtEndDocument{\end{CJK}}
''',
    }
else:
    latex_elements = {
        'papersize' : 'a4paper',
        'utf8extra' : '',
        'inputenc' : '',
        'babel' : r'''\usepackage[english]{babel}''',
        'preamble' : r'''
\usepackage{ctex}
''',
    }
}
```

6.4 rST和latex的类比

```
=====  
Cartesian closed categories and the price of eggs  
=====
```

:author: Jane Doe
:date: September 1994

My First Chapter
=====

Hello world!

```
\documentclass{article}
\title{Cartesian closed categories and the price of eggs}
\author{Jane Doe}
\date{September 1994}
\begin{document}
\maketitle
\section{My First Chapter}
Hello world!
\end{document}
```


跟其他轻量级标记语言一样，可以做到人和机器都比较好读。

参考资源:

- <http://docutils.sourceforge.net/docs/user/rst/quickstart.html>
- <http://docutils.sourceforge.net/0.6/docs/user/rst/quickref.html>
- <http://docutils.sourceforge.net/docs/ref/rst/directives.html>
- <http://zh-sphinx-doc.readthedocs.io/en/latest/rest.html>

工具:

- 预览工具: <http://rst.ninjs.org>

段落

```
Whitespace, newlines, blank lines, and
all kinds of markup (like *this* or
\this) is preserved by literal blocks.
```

```
The paragraph containing only '::'
will be omitted from the result.
```


CHAPTER 8

rst 文件直接发布

使用 Docutils 工具，可以将rst直接发布为多种格式，如html，latex等。

安装: `pip install Docutils`

rst2html: `rst2html.py test.txt test.html`

参考资料: [1], <http://docutils.sourceforge.net/docs/user/tools.html#introduction>

rtd 主题由Read the Doc团队开发，主题美观大方。本小节将以此主题为例，说明主题如何自定义。

9.1 主题配置

主题的配置文件在 `sphinx_rtd_theme/theme.conf` 文件中，默认配置如下：

```
[theme]
inherit = basic
stylesheet = css/theme.css

[options]
typekit_id = hiwlhhg
analytics_id =
sticky_navigation = False
logo_only =
collapse_navigation = False
display_version = True
navigation_depth = 4
prev_next_buttons_location = bottom
canonical_url =
```

基本选项含义：

- `analytics_id` 字符串。配置 **Google Analytics ID** 可以追踪网站访问情况。
- `display_version` 布尔值。配置是否显示版本号。

导航栏选项：

- `collapse_navigation` 布尔值。启用后，不在导航栏中显示 +。
- `navigation_depth` 整数。最大深度为4层，设置为 -1 表示不限制深度。

更多说明，可见 [官方文档](#)

PDF输出: <http://media.readthedocs.org/pdf/doclikecode/latest/doclikecode.pdf>

Jinja是Python的类库，因为Django框架将其用作模板语言，因为有很大的知名度。

安装: `pip install Jinja2`

快速示例:

```
from jinja2 import Template
template = Template('Hello {{ name }}!')
template.render(name='John Doe')
```

极简模板

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>My Webpage</title>
</head>
<body>
  <ul id="navigation">
    {% for item in navigation %}
      <li><a href="{{ item.href }}">{{ item.caption }}</a></li>
    {% endfor %}
  </ul>

  <h1>My Webpage</h1>
  {{ a_variable }}

  {# a comment #}
</body>
</html>
```

代码含义:

- `{% ... %}` for Statements
- `{{ ... }}` for Expressions to print to the template output

- `{# ... #}` for Comments not included in the template output
- `# ... ##` for Line Statements

参考资料:

[1]. <http://jinja.pocoo.org>

Jekyll帮助页面示例:

- [GitHub Official Teaching Materials](#) , 源代码
- [beegit帮助文档](#) , 源代码
- [circleci帮助文档](#) , 源代码

行业实践:

- ‘[开源项目openstack的文档 https://docs.openstack.org/doc-contrib-guide/index.html](https://docs.openstack.org/doc-contrib-guide/index.html)’_

课程演示任务:

11.1 Github Pages 托管静态页面

1. 下载CATTP平台中的个人单页面压缩包
2. 在Github中新建一个repo
3. 将html页面托管在repo中

漂亮的单页面帮助文档

1. <https://pages.github.com/#user-site>

其他参考教程: [Creating and Hosting a Personal Site on GitHub](#)

11.1.1 具体步骤:

1. 在Github上创建新的repo, 例如命名为 `demo-single-page`
2. 通过Github Desktop同步到本地文件夹 `demo-single-page`
3. 将从CATTP平台上下载的单页面网站, 复制到本地文件夹 `demo-single-page` 中, 并push到远程repo

11.2 使用Jekyll Now迅速部署自己的博客

1. Fork Jekyll-now主题，并将repo改为 yourusername.github.io
2. 在_config.yml中配置自己的个人信息
3. 修改jekyll的主题
 - 更多帮助信息: <https://www.smashingmagazine.com/2016/02/content-modeling-with-jekyll/>
 - Jekyll Documentation Theme: <http://idratherbewriting.com/documentation-theme-jekyll/#>

11.3 完整版jekyll安装

1. 安装jekyll环境: `sudo gem install jekyll bundler`
2. 创建新博客: `jekyll new my-awesome-site`
3. 启动网站。进入网站文件夹后，运行: `bundle exec jekyll serve`
4. 浏览网站。localhost:4000

更多帮助信息: <https://jekyllrb.com>

11.4 添加内容并发布新网站

1. 使用Markdown语法创建任意内容，按照 YYYY-MM-DD-Title 的格式命名
2. 将上方创建的Markdown文件，保存在 my-awesome-site/_posts中
3. 在Terminal中运行 `jekyll build`，jekyll将发布网站，并将静态内容存储在 _site 文件夹中

12.1 主题网站地址:

- <http://jekyllthemes.org>
- <https://github.com/jekyllbootstrap>

12.2 安装基于gem的主题

1. 在 `./Gemfile` 中增加 `gem "jekyll-theme-awesome"`
2. 安装主题。 `bundle install`
3. 在 `_config.yml` 中修改主题为 `theme: jekyll-theme-awesome`
4. 编译网站 `bundle exec jekyll serve`
 - 官方步骤: <https://jekyllrb.com/docs/themes/#installing-a-theme>
 - 主题介绍: <https://jekyllrb.com/docs/themes/#overriding-theme-defaults>

注解: `./Gemfile` 可以添加多个主题, 但是 `_config.yml` 只能以一种主题发布

常用Jekyll帮助主题

-
- 典型左中右三栏结构: [Minimal Mistakes](#)
 - 经典帮助文档: [Jekyll Documentation Them](#)
 - [Jekyll Doc Theme](#)

12.3 商业模板

<https://doks.themejack.com/blue/getting-started/>

12.4 在 **github pages** 上添加主题

官方教程: <https://help.github.com/articles/adding-a-jekyll-theme-to-your-github-pages-site/>

12.5 制作Jekyll主题

CHAPTER 13

索引和表格

- genindex
- modindex
- search