cookiecutter.project

Release {{cookiecutter.release}}

Mar 31, 2018

Contents:

CHAPTER 1

简介

本项目旨在简化 `sphinx` 的使用需求, 将配置模板化, 便于轻松在多个项目完成漂亮的文档编写. 对于中文 pdf 生成, 做了一些改进

本项目环境为 `mac python3`

sphinx 详细的语法参见sphinx-doc

CHAPTER 2

使用项目

本次的安装所有在 python3.5 环境下安装,

什么? python2.x 怎么办?

都什么年代, 你想要被 2 的中文折腾可以自己研究一下

2.1 安装 cookiecutter

```
pip install cookiecutter
```

未防止一些全局版本问题尽量使用 virtualenv

2.2 生成样板工程

切换到测试演示目录, 并且切换到 virtualenv

```
cd tmp
mkdir test_doc
. /data/vpythons/python3/bin/activate
```

用 cookiecutter 命令生成项目模板命令过程中可能需要输入一些基本的数据信息

```
(python3)    test_doc cookiecutter https://github.com/yishenggudou/cookiecutter-sphinx-doc.git
You've downloaded /Users/timgerk/.cookiecutters/cookiecutter-sphinx-doc before. Is it okay to␣
↪delete and re-download it? [yes]: yes
project [example project name]:
project_slug [epo]:
book_title [实例项目文档]:
author [author name]:
version [0.0.1]:
release [0.0.1]:
description [some long description]:
src [docs]:
```

安装环境需要的依赖

```
pip install -r requirements.txt
```

测试 html 部分

```
cd docs
make html
open _build/html/index.html
```

测试生成 pdf

```
cd docs
./topdf.sh
```

CHAPTER 3

使用 markdown 来写作

rst 很强大,But 现在 markdown 更流行, 如果你想在 sphinx 中使用 markdown 来写作也是可以的, sphinx 也提供了组件recommonmark, 具体参见sphinx-markdown 但是 recommonmark 只实现了基础版本的markdown 语法

3.1 简介

参见

1. markdown 语法
2. rst
3. [rst][http://sphinx-doc-zh.readthedocs.io/en/latest/rest.html]

下面演示用法, 你看到的是渲染后的结果了, 具体源码参见示例

3.2 code

```
\`\`\`

code

\`\`\`

\`\`\`ipython
In [1]: import os

In [2]:
\`\`\`
```

结果

```
code
```

```
In [1]: import os

In [2]:
```

3.3 sas

```
hello
```

3.4 math

```
a = \`$ y=\sum_{i=1}^n g(x_i) $\`

a+b

\`\`\`math
  (a + b)^2 = a^2 + 2ab + b^2

\`\`\`
```

结果

$a = y = \sum_{i=1}^{n} g(x_i)$

a+b

$$(a + b)^2 = a^2 + 2ab + b^2$$

3.5 rst

```
\`\`\`eval_rst
.. todo::

    some todo things

\`\`\`


\`\`\`eval_rst
.. note::

    some todo things

\`\`\`



\`\`\`eval_rst
.. warning:: note the space between the directive and the text
\`\`\`

\`\`\`eval_rst
.. seealso:: This is a simple **seealso** note.

\`\`\`
```

结果:

Todo: some todo things

Note: some todo things

Warning: note the space between the directive and the text

See also:

This is a simple **seealso** note.

3.6 table

```
\`\`\`eval_rst
.. table:: Truth table for "not"
   :widths: auto

   =====  =====
     A    not A
   =====  =====
   False  True
   True   False
   =====  =====
\`\`\`
```

结果

Table 3.1: Truth table for “not”

A	not A
False	True
True	False

3.7 脚注

```
\`\`\`eval_rst
Lorem ipsum [#f1]_ dolor sit amet ... [#f2]_

.. rubric:: Footnotes

.. [#f1] Text of the first footnote.
.. [#f2] Text of the second footnote.
\`\`\`
```

结果

Lorem ipsum[1] dolor sit amet …[2]

[1] Text of the first footnote.
[2] Text of the second footnote.

3.8 引证

```
\`\`\`eval_rst

Lorem ipsum [Ref]_ dolor sit amet.

.. [Ref] Book or article reference, URL or whatever.

\`\`\`
```

Lorem ipsum *[Ref]* dolor sit amet.

3.9 注释

```
\`\`\`eval_rst
..
   This whole indented block
   is a comment.

   Still in the comment.
\`\`\`
```

结果:

3.10 image

```
![](https://www.baidu.com/img/bd_logo1.png)
```

CHAPTER 4

HTTPAPI 文档演示

GET /users/(int: *user_id*)**/posts/**

tag The posts tagged with *tag* that the user (*user_id*) wrote.

Example request:

```
GET /users/123/posts/web HTTP/1.1
Host: example.com
Accept: application/json, text/javascript
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/javascript

[
  {
    "post_id": 12345,
    "author_id": 123,
    "tags": ["server", "web"],
    "subject": "I tried Nginx"
  },
  {
    "post_id": 12346,
    "author_id": 123,
    "tags": ["html5", "standards", "web"],
    "subject": "We go to HTML 5"
  }
]
```

Query Parameters

- **sort** – one of `hit`, `created-at`
- **offset** – offset number. default is 0
- **limit** – limit number. default is 30

Request Headers

- Accept – the response content type depends on *`Accept`* header

- Authorization – optional OAuth token to authenticate

Response Headers

- Content-Type – this depends on *Accept* header of request

Status Codes

- 200 OK – no error
- 404 Not Found – there’s no user

CHAPTER 5

用 matplotlib 来绘图

Matplotlib 是一个 Python 的 2D 绘图库，它以各种硬拷贝格式和跨平台的交互式环境生成出版质量级别的图形 [1] 。通过 Matplotlib，开发者可以仅需要几行代码，便可以生成绘图，直方图，功率谱，条形图，错误图，散点图等。

5.1 ipython 代码展示

用法:

```
.. sourcecode:: ipython

 In [69]: lines = plot([1,2,3])

 In [70]: setp(lines)
   alpha: float
   animated: [True | False]
   antialiased or aa: [True | False]
   ...snip
```

结果:

```
In [69]: lines = plot([1,2,3])

In [70]: setp(lines)
  alpha: float
  animated: [True | False]
  antialiased or aa: [True | False]
  ...snip
```

5.2 公式

用法:

```
.. math::

  W^{3\beta}_{\delta_1 \rho_1 \sigma_2} \approx U^{3\beta}_{\delta_1 \rho_1}
```

结果:

$$W^{3\beta}_{\delta_1\rho_1\sigma_2} \approx U^{3\beta}_{\delta_1\rho_1}$$

5.3 绘图

5.3.1 文件绘图

用法:

```
.. plot:: ./pyplots/ellipses.py
   :include-source:
```

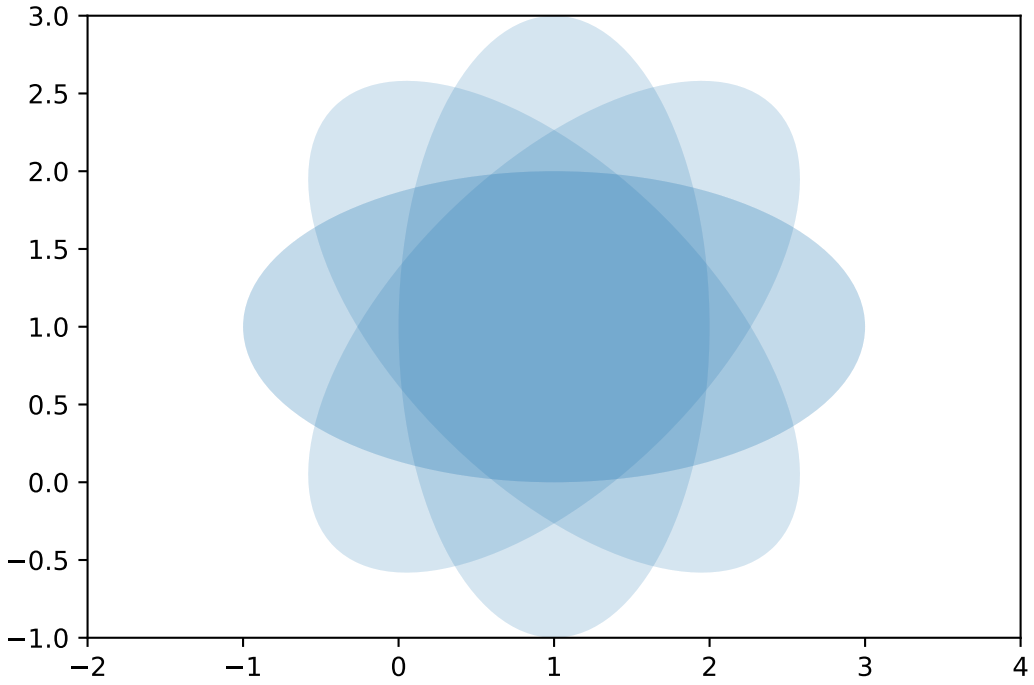
结果:

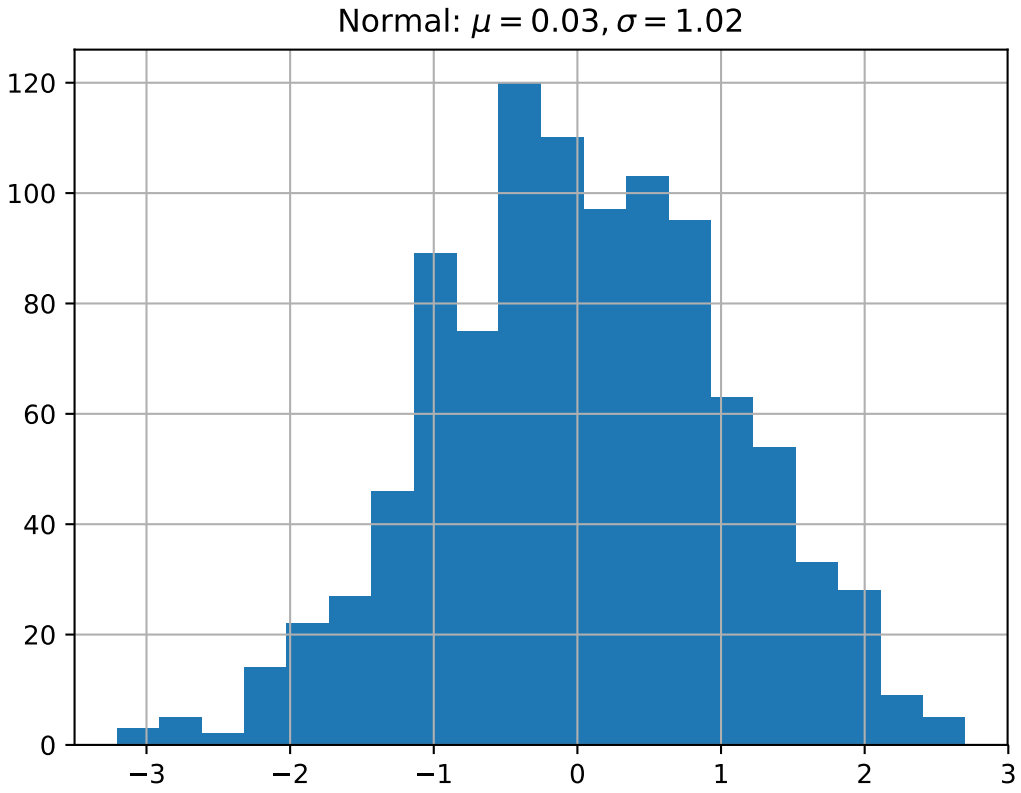
```
# -*- coding: utf-8 -*-

from pylab import *
from matplotlib.patches import Ellipse

delta = 45.0 # degrees

angles = arange(0, 360+delta, delta)
ells = [Ellipse((1, 1), 4, 2, a) for a in angles]

a = subplot(111, aspect='equal')

for e in ells:
    e.set_clip_box(a.bbox)
    e.set_alpha(0.1)
    a.add_artist(e)

xlim(-2, 4)
ylim(-1, 3)

show()
```

用法:

```
.. plot::

   import matplotlib.pyplot as plt
   import numpy as np
   x = np.random.randn(1000)
   plt.hist( x, 20)
   plt.grid()
   plt.title(r'Normal: $\mu=%.2f, \sigma=%.2f$'%(x.mean(), x.std()))
   plt.show()
```

结果


3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−2
−1
0
1
2
3
4


Normal: $\mu = 0.03, \sigma = 1.02$
120
100
80
60
40
20
0
−3
−2
−1
0
1
2
3

CHAPTER 6

使用 plantuml 来画 UML

UML 在各种项目中用来描述项目, 系统, 流程, 便于理解. 我们既然是技术类写作,UML 少不了.

什么是 PlantUML PlantUML 是一个快速创建 UML 图形的组件，PlantUML 支持的图形有：sequence diagram, use case diagram, class diagram, activity diagram, component diagram, state diagram, object diagram, wireframe graphical interface PlantUML 通过简单和直观的语言来定义图形，语法参见 PlantUML Language Reference Guide，它支持很多工具，可以生成 PNG、SVG、LaTeX 和二进制图片

更多内容参见 plantuml 官网 plantuml.

下面说下怎么和 sphinx 一起结合使用. 主要两种方式

sphinx 提供了支持 'plantuml'_ 的插件 sphinx-contrib-plantuml sphinx-contrib-plantuml 提供了一个 'uml'指令

'uml'指令可以有四个配置项

1. 在 rst 中使用文件
2. 在 rst 中使用代码

6.1 使用文件

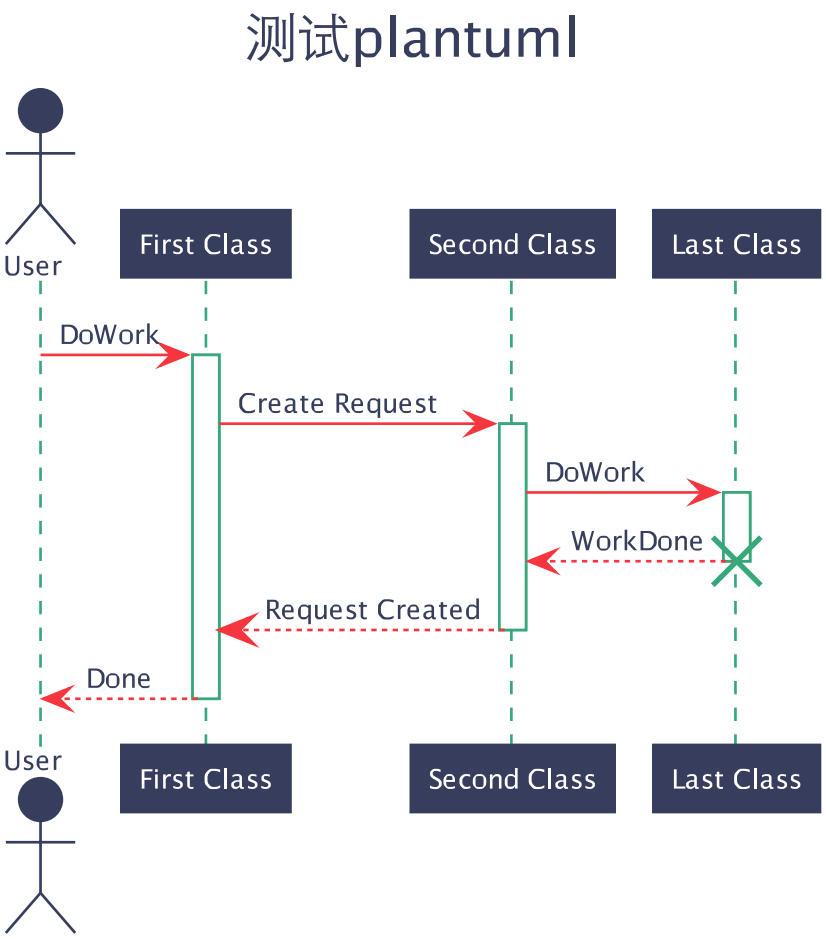
```
.. uml:: /_static/test.puml
```

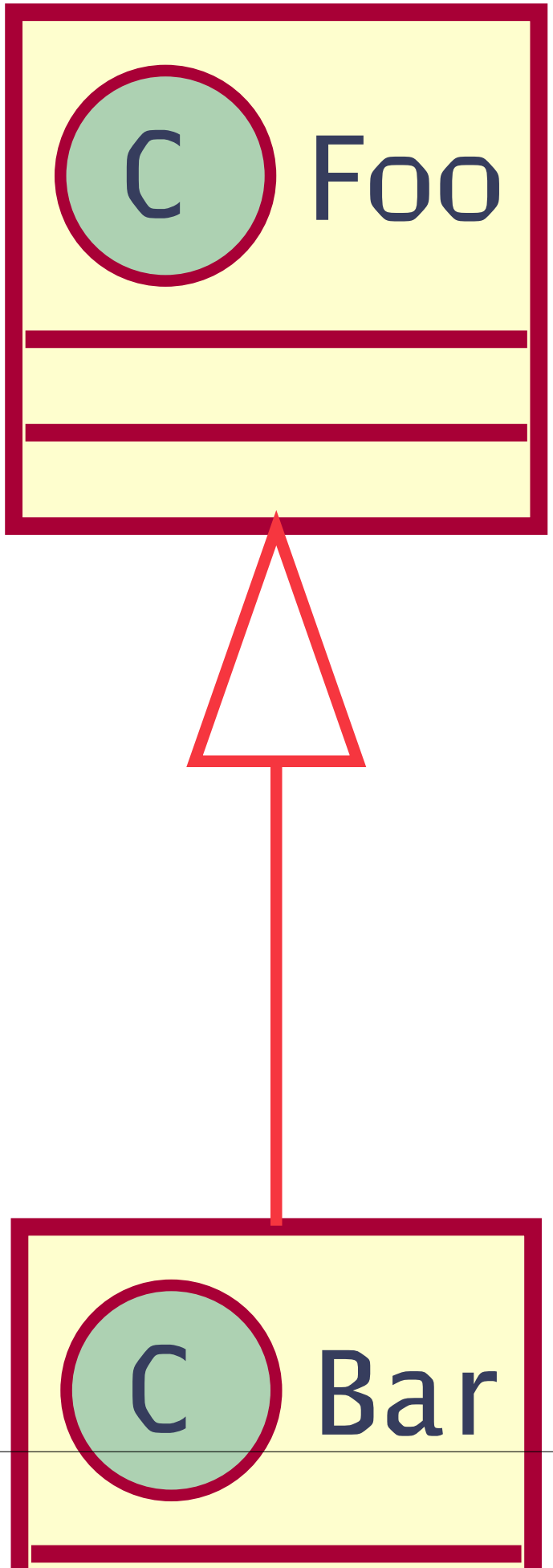
6.2 使用代码

用法

```
.. uml::
   :caption: Caption with **bold** and *italic*
   :width: 100mm

   Foo <|-- Bar
```

结果


测试plantuml
User
First Class
Second Class
Last Class
DoWork
Create Request
DoWork
WorkDone
Request Created
Done
User
First Class
Second Class
Last Class


C
Foo
C
Bar

Note: 使用代码方式主题将不会生效, 主要 sphinx 插件 sphinx-contrib-plantuml 不支持.

CHAPTER 7

最后

follow 微信公众号

CHAPTER 8

Indices and tables

- genindex
- modindex
- search

Bibliography

[Ref] Book or article reference, URL or whatever.

/users

`GET /users/(int:user_id)/posts/(tag)`, 9