Python 系列培训

组织和编写 Python 软件包

报告人：王 靓

2019 年 8 月 25 日

主要内容

- 怎样组织和管理 Python 软件包
  - Pypi 、 Pip
- 如何进行版本控制
  - git 、 github
- 怎样撰写和维护软件包的文档 / 手册
  - reStructuredText 、 Sphinx-doc

Python 打包

目录结构：

```
package/
__init__.py
subpackage1/
    __init__.py
    moduleX.py
    moduleY.py
subpackage2/
    __init__.py
    moduleZ.py
    moduleA.py
```

- __init__.py 文件：将目录变为一个 Python 模块
- __init__.py 主要控制包的导入行为
- 在导入一个包时，实际上是导入了它的 __init__.py 文件

Python 打包

- python 可以将自己写的模块进行打包或发布
  - 方法 1：将类包直接拷贝到 Python 的 lib 目录
    - 不便于管理与维护
    - 存在多个 Python 版本时会造成混乱
  - 方法 2：编写 setup.py 文件对 Python 模块进行打包

```
class MyClass():
    def __init__(self):
        self.data = 10;

    def print_data(self):
        print(self.data)

    def add_data(self):
        self.data = self.data + 5
```

```
from distutils.core import setup

setup(name = 'Test',
      version = '1.0',
      description='A test module',
      author = 'He Boliang',
      author_email = 'hebl@gmail.com';,
      url ='http://hebl.lamost.org';,
      packages = ['naoctest', 'naoctest/subpack1'],
    )
```

其中 distutils 是 python 内建的标准打包库

编译（并把相应的文件放在 build/ 目录中

```
python setup.py build
```

生成 zip/tar.gz 压缩包（并把打包文件放在 dist/ 目录中）

```
python setup.py sdist
```

生成 .exe 安装包

```
python setup.py bdist_wininst
```

生成 rpm 包

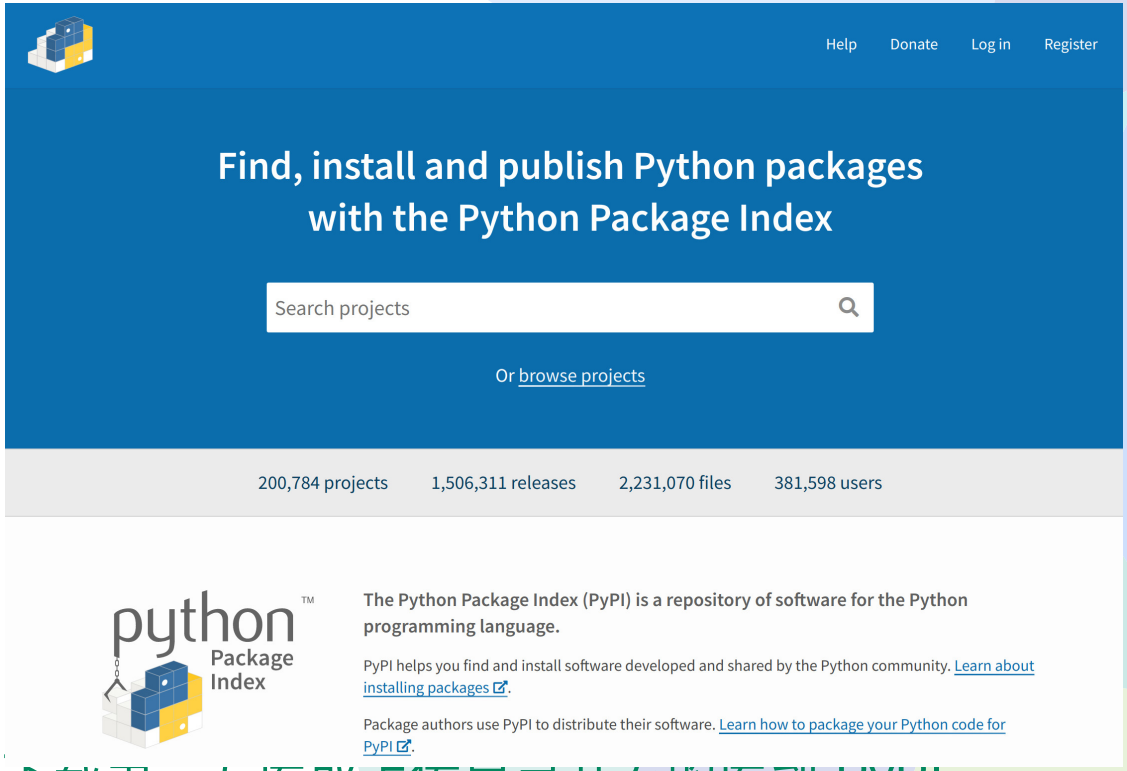
```
python setup.py bdist_rpm
```

用户下载、解压缩后安装：

```
python setup.py install
```

PyPI — Python 软件包仓库

- **PyPI** (Python Package Index) 是 python 官方的第三方软件仓库 https://pypi.org/



- 所有人都可以下载第三方库或上传自己开发的库到 PyPI
- PyPI 推荐使用 pip 包管理器来下载第三方库

Pip 软件包管理工具

pip (python install packages) 是一个通用的 Python 包管理工具

列出所有已经安装的软件包

```
$ pip freeze 或者 $ pip list
```

查找软件包（包括已安装和未安装的）

```
$ pip search astropy
```

安装软件包

```
$ pip install astropy
```

升级已安装的软件包

```
$ pip install --ugrade astropy
```

查看已安装软件包的详细信息

```
$ pip show astropy
```

卸载软件包

```
$ pip uninstall astropy
```

Pip 软件包管理工具

- Pip 和 pip3 的区别
  - 用 pip install XXX 安装的软件包会放在 python2.7/site-packages 目录下；
  - 用 pip3 install XXX 安装的软件包会放在 python3.6/site-packages 目录下；
  - 用 Python3 无法导入 python2.7/site-packages 目录下的软件包。
- Pip 和 Conda 的区别
  - Pip 安装 PyPI 仓库里的 Python 包
  - conda 安装 Anaconda 里的 conda 包，不限于 Python 包，可能包含任何语言编写的软件包

Install and Manage Packages in Python

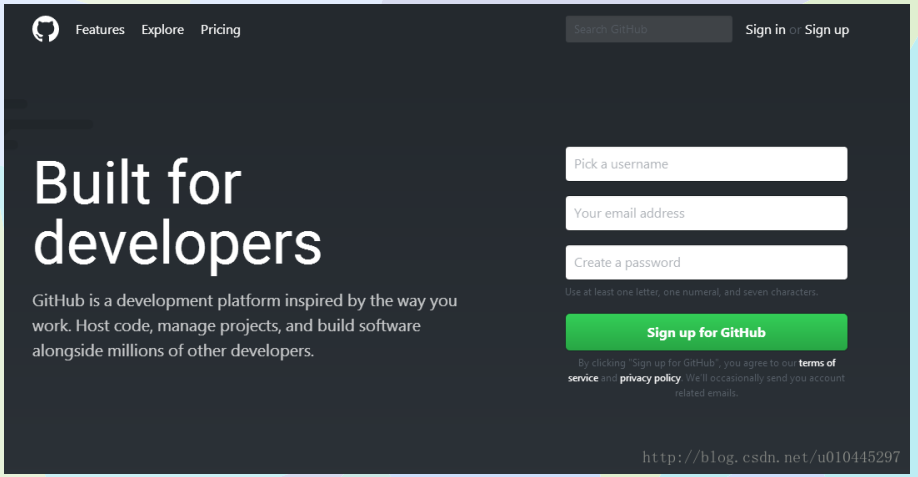
pip	conda
pip search pyserial	conda search pyserial
pip install pyserial	conda install pyserial
pip install pyserial --upgrade	conda update python
pip list	conda list

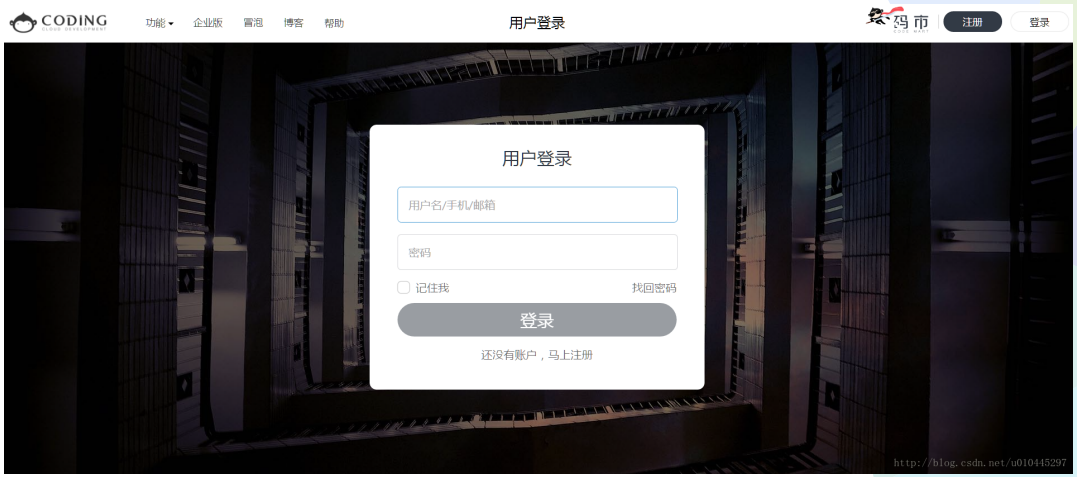
Git 介绍

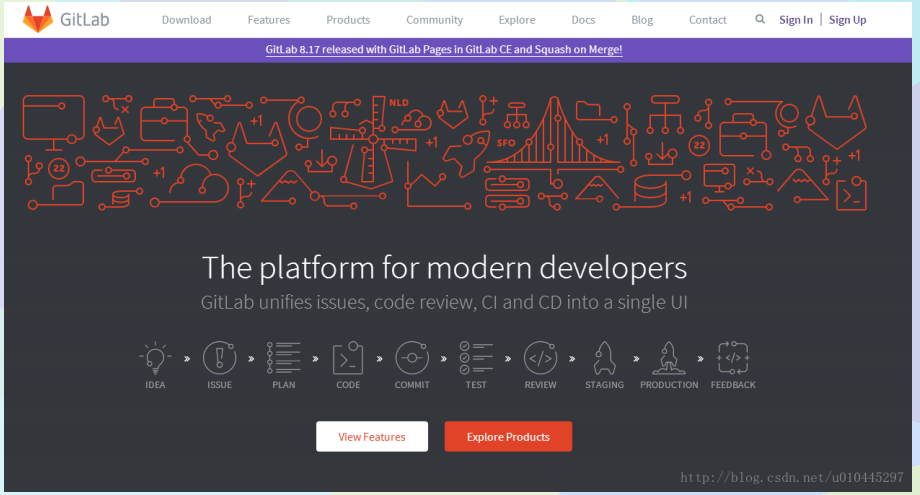
- 什么是 GIT？
  - 不同于 CVS 或 SVN 等**集中式**控制系统， GIT 是一种**分布式**版本控制系统
  - Linus Torvalds 为了管理 Linux 内核而开发的版本控制软件
- 为什么要用 GIT？
  - 每个开发者都有一个完整的版本仓库
  - 可以脱机工作
  - 安全性高

Git 介绍

- 常用的 Git 代码托管平台
  - Github
  - Gitlab
  - Coding.net
  - code.china-vo.org





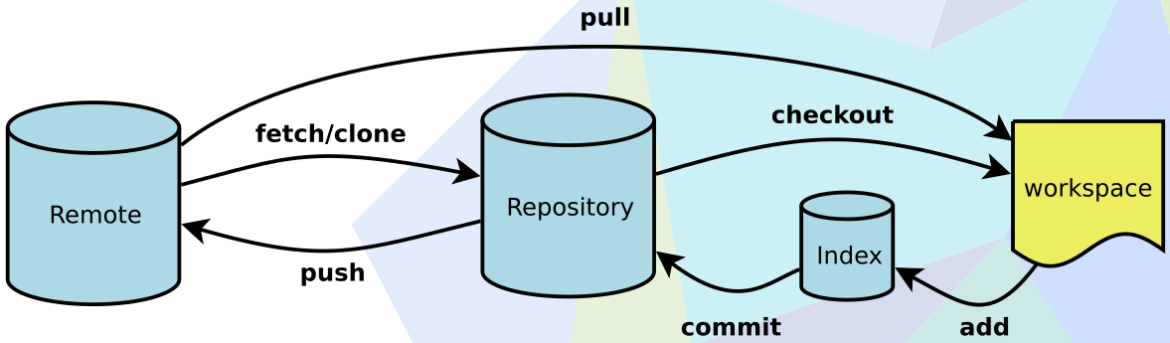


Git 介绍

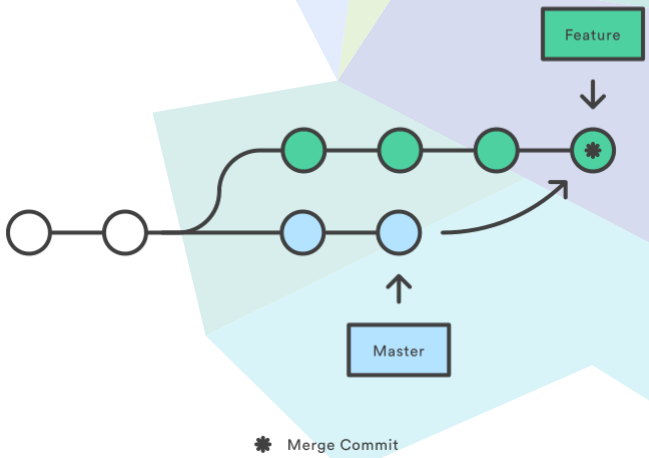
- Github 怎么用

Git 介绍

- Git 怎么用?
  - Git 有两个代码库：服务器上的远程仓库（Remote）；用户在自己主机创建的代码库（Repository）；
  - 用户通过 fetch/clone 命令把远程库同步到本地库，通过 push 命令把本地库的主分支同步到远程库;



  - 代码库有主分支（master），和很多个辅助分支（branch）。用户在辅分支上修改，而后合并到住分支



Git Cheat Sheet

REBELLABS by ZEROTURNAROUND

Create a Repository

From scratch -- Create a new local repository

```
$ git init [project name]
```

Download from an existing repository

```
$ git clone my_url
```

Observe your Repository

List new or modified files not yet committed

```
$ git status
```

Show the changes to files not yet staged

```
$ git diff
```

Show the changes to staged files

```
$ git diff --cached
```

Show all staged and unstaged file changes

```
$ git diff HEAD
```

Show the changes between two commit ids

```
$ git diff commit1 commit2
```

List the change dates and authors for a file

```
$ git blame [file]
```

Show the file changes for a commit id and/or file

```
$ git show [commit]:[file]
```

Show full change history

```
$ git log
```

Show change history for file/directory including diffs

```
$ git log -p [file/directory]
```

Working with Branches

List all local branches

```
$ git branch
```

List all branches, local and remote

```
$ git branch -av
```

Switch to a branch, my_branch, and update working directory

```
$ git checkout my_branch
```

Create a new branch called new_branch

```
$ git branch new_branch
```

Delete the branch called my_branch

```
$ git branch -d my_branch
```

Merge branch_a into branch_b

```
$ git checkout branch_b
$ git merge branch_a
```

Tag the current commit

```
$ git tag my_tag
```

Make a change

Stages the file, ready for commit

```
$ git add [file]
```

Stage all changed files, ready for commit

```
$ git add .
```

Commit all staged files to versioned history

```
$ git commit -m "commit message"
```

Commit all your tracked files to versioned history

```
$git commit -am "commit message"
```

Unstages file, keeping the file changes

```
$ git reset [file]
```

Revert everything to the last commit

```
$ git reset --hard
```

Synchronize

Get the latest changes from origin (no merge)

```
$ git fetch
```

Fetch the latest changes from origin and merge

```
$ git pull
```

Fetch the latest changes from origin and rebase

```
$ git pull --rebase
```

Push local changes to the origin

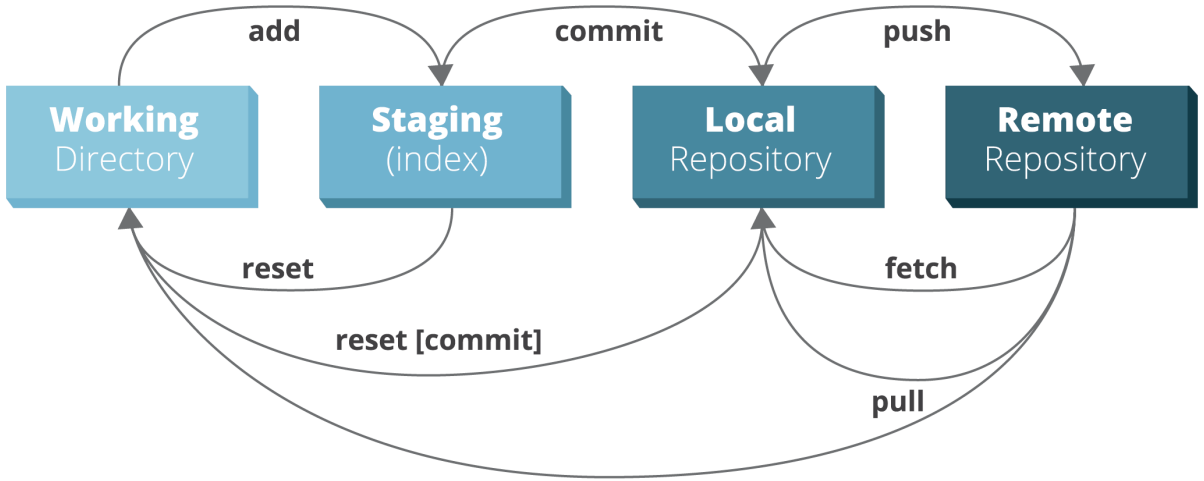
```
$ git push
```

Finally!

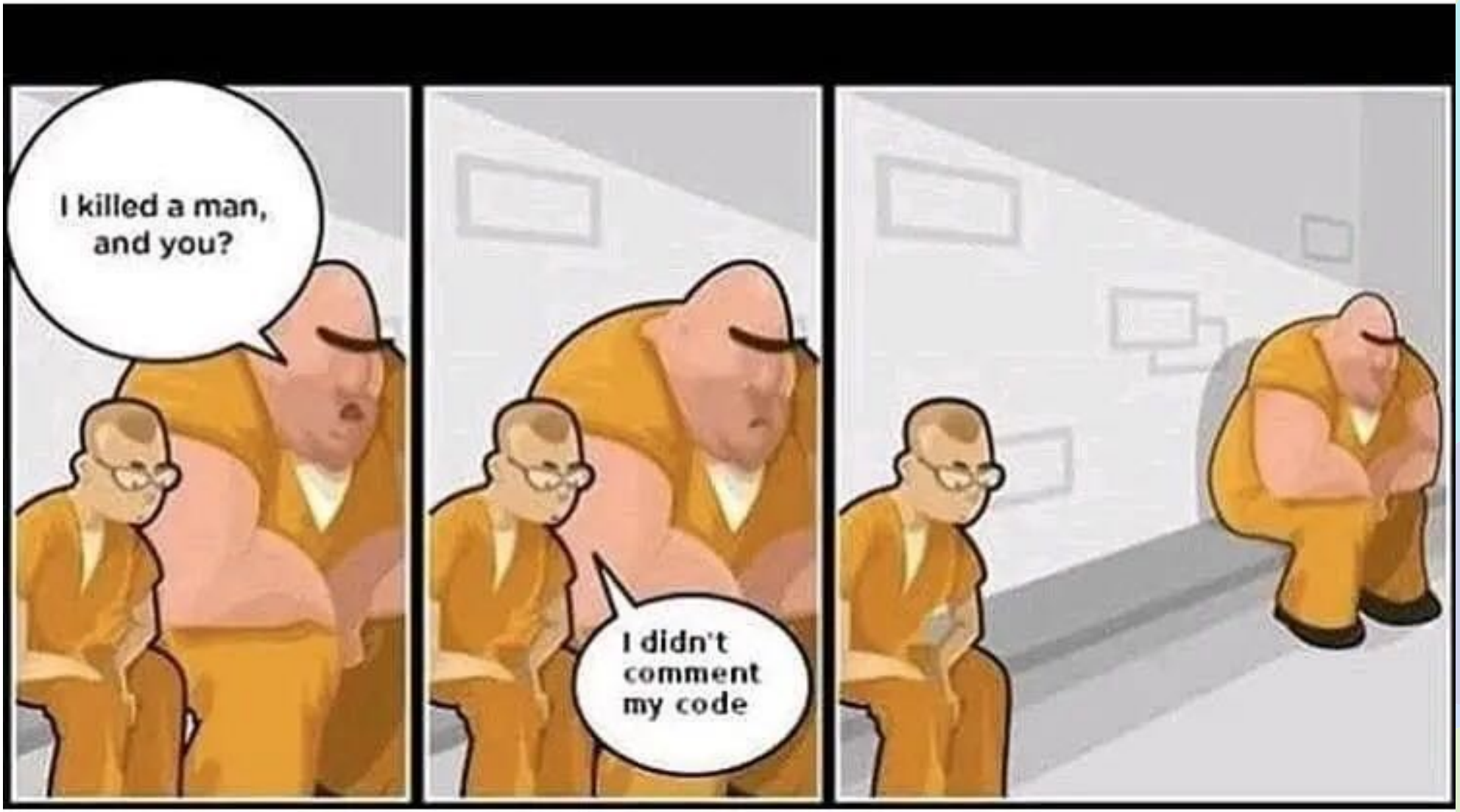
When in doubt, use git help

```
$ git command --help
```

Or visit https://training.github.com/ for official GitHub training.

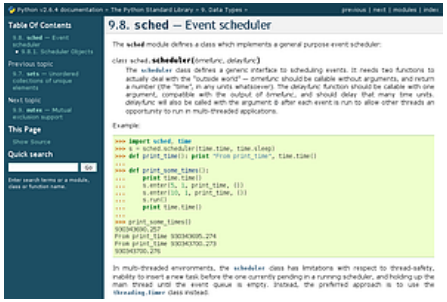


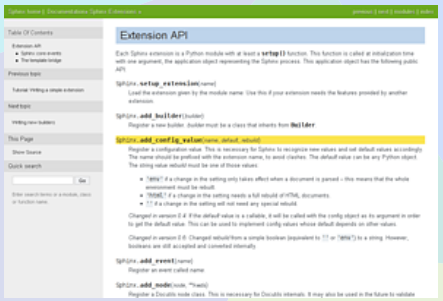
编写文档

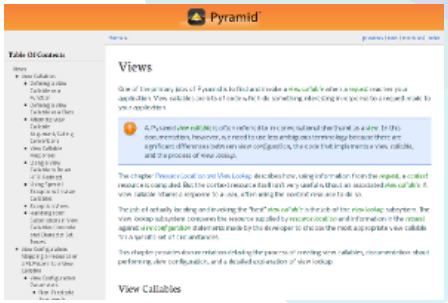


如何编写代码文档

- Sphinx-doc 是 python 的文档编写工具
  - 例如： python、 numpy、 matplotlib、 astropy 等官方帮助文档
- sphinx-doc 使用 reStructuredText 格式（rst），带有简单语法的纯文本
  - 通过编译，可以生成 html,epub,man,pdf 等多种格式
  - 拥有丰富的插件和可自定义的输出效果







Sphinx-doc 简介

安装： pip3 install sphinx

初始化文档： sphinx-quickstart

```
?  make.bat                                  # window 下编译脚本
?  Makefile                                  # Linux 下 Makefile 文件
?
??build              # make 编译后产生的网页目录在 build/html 目录下
??source                                      # 文档源码目录
    ?  conf.py                               # 配置文件
    ?  index.rst                             # 文档源文件入口
    ??_static                          # 编译过程产生的一些图片之类的
    ??_templates                             # 模板
```

练习

- 从 clone 一个真正的项目开始

```
git clone https://github.com/wangleon/NAOCTraining
```

- 做一些修改并推送到主分支

```
git status
git add …
git commit -m “change something”
git push origin master
```

- 编译文档

```
cd doc/
make html
```

- 在 github 上创建自己的项目

```
git init
```