

## MkDocs项目文档生成器(一)

---

# 简介

类别：项目文档生成器，生成静态站点，管理Markdown文档。

官方网址：<http://www.mkdocs.org/>

中文文档：[http://markdown-docs-zh.readthedocs.io/zh\\_CN/latest/](http://markdown-docs-zh.readthedocs.io/zh_CN/latest/) 建议直接看最新的英文官方文档

特点：

- 一个用于创建项目文档的快速、简单、华丽的静态站点生成器，文档源码使用 Markdown 来撰写，用一个 YAML 文件作为配置文档。
- 构建完全的静态 HTML 站点，可以将它托管到 GitHub pages、Amazon S3 等任意地方。
- 默认包含大量美观的主题，可以从 bootstrap, readthedocs 和 12 款 bootswatch 主题中选择。
- 即时预览
- 易于配置
- 交叉索引

# 安装

由于小编当前的公司使用的是Windows开发环境，所以在此处给出MkDocs在Window 10的安装和使用教程，有Macbook的请直接去看官方文档，用Linux的请绕道别处。

1.需要安装Python：面向对象的解释型语言，因为MkDocs是基于Python工作的，所以需要Python环境的支持，Mkdocs支持2.6/2.7/3.3/3.4/3.5(去官网看最新的支持)；

2.需要安装Pip：Python的包管理器，如果安装了高版本的Python，会附带安装；

3.安装MkDocs：即本文主要讲的项目文档生成器；

4.另外，由于MkDocs的官方文档建议使用Chocolatey来安装Python和Pip，所以在此处我们也学习一下这个Windows的包管理器(软件管理器)；

## 我的配置

- chocolatey 0.10.0
- Python 3.5.1
- pip 8.1.2
- mkdocs 0.15.3
- PC Win 10 专业版 64-bit 8G Dell

# Chocolatey

## 简介 - Windows的包管理器

如果使用过Linux的话，一定会被其简洁的软件包管理系统所折服。在Linux世界中，安装一个软件不需要在浏览器中寻找软件的官网，然后将其下载下来，然后双击进行安装。只需要一条简单的命令，就可以完成搜索、安装、更新、卸载等所有操作。例如Debian和Ubuntu中的apt-get命令，Fedora的yum以及新的dnf命令，还有Arch中的pacman。其实Windows下,也有这么一个包管理器，功能虽然不及Linux中那些包管理器强大，但是也让Windows下的软件安装方便了不少。这就是[Chocolatey](https://chocolatey.org/)。

远离插件，远离广告，远离病毒，真正的一行命令安装一个软件，感觉棒棒哒！！

## 官方网址

<https://chocolatey.org/>

## 安装

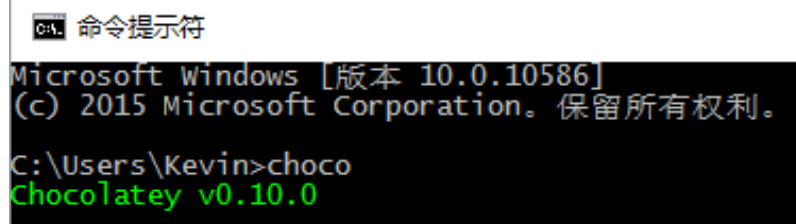
官方教程地址：<https://chocolatey.org/install>

1. C:\Windows\System32，找到cmd.exe，右键选择使用管理员权限打开

2. 将下面的代码粘贴到命令行中，然后回车，等待安装即可。

```
1 @powershell -NoProfile -ExecutionPolicy Bypass -Command "iex ((New-Object
  System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))" &&
  SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"
2 1
```

1. 安装完毕后，输入choco，如果显示版本，那就代表安装成功了。



命令提示符

Microsoft Windows [版本 10.0.10586]  
(c) 2015 Microsoft Corporation。保留所有权利。

C:\Users\Kevin>choco  
Chocolatey v0.10.0

## 1. 常用命令

```
1 1.search: 会搜索出所有包含关键字的软件包，比如关键字为python，就会输入与之相关的各种
  软件包
2     choco search 关键字
3 2.install: 安装软件
4     choco install 软件包名称
5 3.update: 更新软件
6     choco upgrade 软件包名称
7 4.uninstall: 卸载软件
8     choco uninstall 软件包名称
9 1
10 2
11 3
12 4
13 5
14 6
15 7
16 8
```

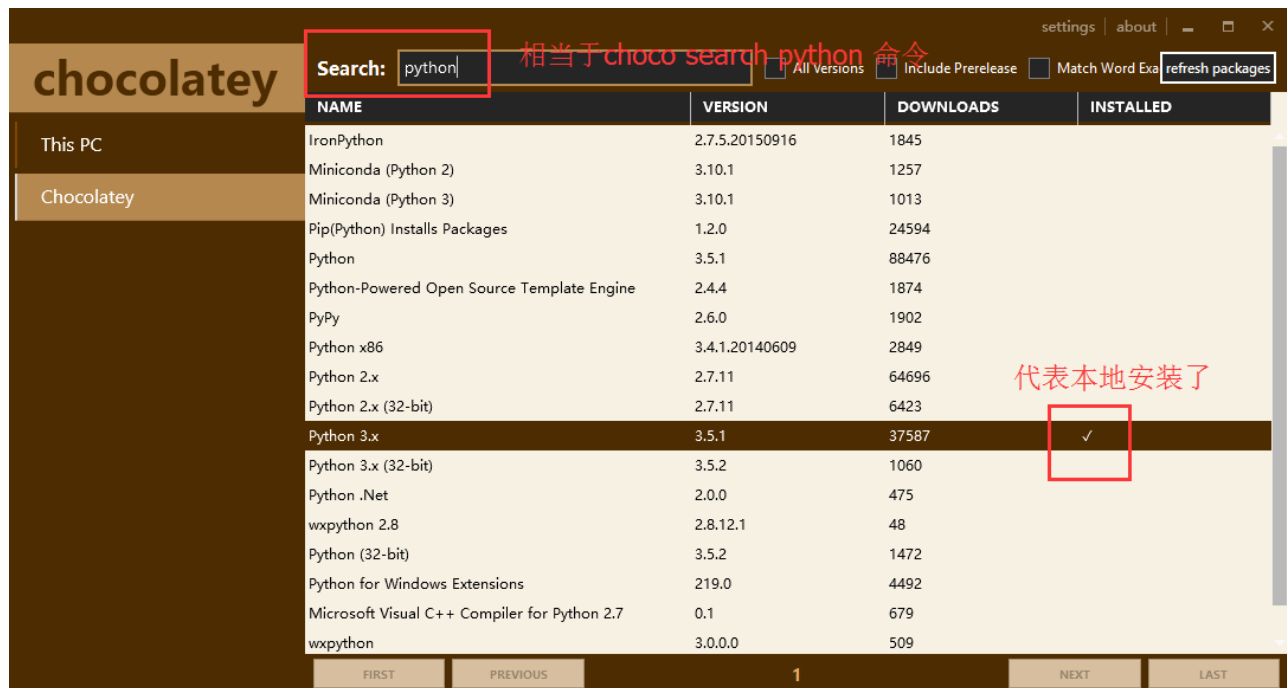
## 1. 如果不想使用命令行，可以安装图形化界面：ChocolateyGUI

```
1 choco install chocolateygui
2 1
```

本地已经安装的软件包：



Chocolatey官网的软件：



## 注意事项

- 如果想要在windows的命令行窗口执行choco，必须使用管理员权限打开cmd.exe才行，否则的话，执行choco命令操作的时候，会经常爆出无法访问某个路径的权限问题。
- 如果没有修改路径，默认安装在C盘，我的安装在了C:\ProgramData\chocolatey。
- 如果想要使用命令行，那么用管理员权限打开cmd.exe，就可以使用choco命令了；
- 如果想要使用GUI，那么直接用Win10搜索框搜索应用即可，或者在C:\Program Files (x86)\ChocolateyGUI找到ChocolateyGui.exe

# Python

## 简介

Mkdocs是使用Python语言开发的，所以要安装Python，由于我们安装了Windows的包管理器Chocolatey，所以我们使用它来安装Python。MkDocs的官方文档给出了它所支持的Python的开发环境版本：2.6，2.7，3.3，3.4，3.5。另外，高版本的Python会自带Pip，所以我们安装稍微高一些的版本，Python2.x系列，需要不低于2.7.9版本，Python3.x系列，需要不低于3.4。

## 安装

如果使用GUI就跟小白一样，所以在此处仅仅给出命令行的安装过程：

1.使用管理员权限打开cmd.exe，然后执行命令

```
1 choco search python
2 1
```

然后就会给出chocolatey官方网站返回的根据关键字python搜索的软件列表，然后我们找到合适的版本即可，小编选择安装的是python3 3.5.1，所以

```
1 choco install python3 3.5.1
2 1
```

然后等待安装即可，如果有问题，选择Y。如果是访问某路径问题，那就代表少侠没有使用管理员权限打开命令行！

## 2.检查是否安装正确

- 输入python -V，会显示python的版本号
- 输入python，会进入python的开发环境，Ctrl+C退出

# Pip

## 简介-Python的包管理器

Python的包管理器，如上文所述，如果安装了高版本的python，会自动安装pip，在命令行中输入pip -V可以查看当前pip版本：

```
C:\WINDOWS\system32>pip -V  
pip 8.1.2 from c:\programdata\chocolatey\lib\python3\tools\lib\site-packages (python 3.5)
```

可以发现，python3会在chocolatey的lib包中，而pip会在python3的tools\lib\site-packages包中。

## 升级

由于自带的pip版本稍低，我们需要升级。如果不升级的话，你直接使用该包管理器下载 MkDocs的话，会提示你版本过低，并且给出你升级的命令，粘贴复制运行即可：

```
1 python -m pip install --upgrade pip  
2 1
```

即执行python命令，使用它的包管理器pip的upgrade命令进行升级pip本身。

升级完毕，然后使用pip -V查看版本，就应该跟上图所示的一样了。

# MkDocs的安装

## 使用Pip安装MkDocs

```
1 pip install mkdocs
2 1
```

该命令经常会爆红色提示，找不到合适的版本，如果python安装的版本正确的话，就有可能是因为网络的问题，毕竟是国外的网站，这个时候需要你多尝试几次，小编第一次安装尝试了三次，不过安装速度倒是特别快，几秒钟完事。

## 检查是否安装正确

1.使用pip list命令查看当前PC使用pip安装的软件包

```
C:\WINDOWS\system32>pip list
click (6.6)
Jinja2 (2.8)
livereload (2.4.1)
Markdown (2.6.6)
MarkupSafe (0.23)
mkdocs (0.15.3)
mkdocs-bootstrap (0.1.1)
mkdocs-bootswatch (0.4.0)
pip (8.1.2)
PyYAML (3.12)
setuptools (18.2)
six (1.10.0)
tornado (4.4.1)
```

2.使用mkdocs -V命令查看当前版本

然后可能也许Maybe你会发现，该命令不是内部命令或者批处理命令的提示，也就是说没有配置路径。然后，小编使用Everything搜索到了mkdocs.exe的路径，是在chocolatey中：

C:\ProgramData\chocolatey\lib\python3\tools\Scripts

也就是说，chocolatey安装的包都在他的lib文件夹中，python3的pip安装的包都在python3中。

然后在上面的路径显示框中输入CMD，然后回车，即可在当前文件夹下打开命令提示符，并且路径是当前文件夹。此时，输入mkdocs -V：

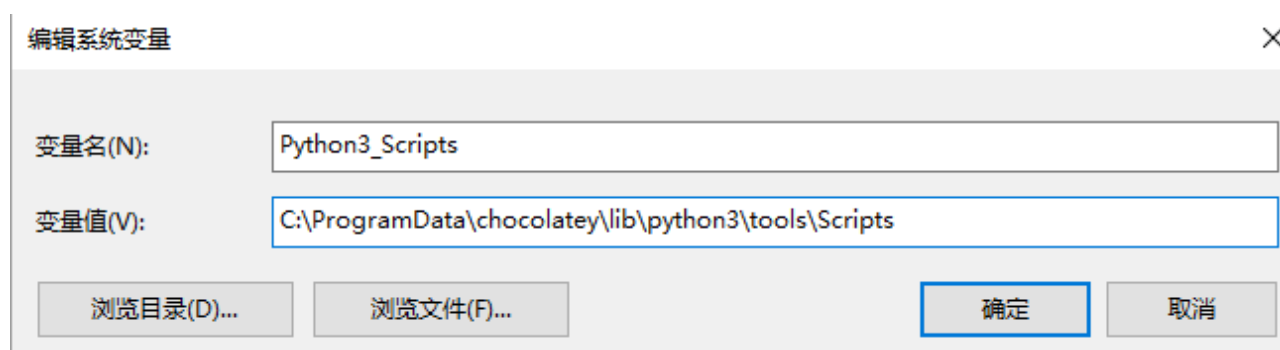
```
C:\ProgramData\chocolatey\lib\python3\tools\Scripts>mkdocs help
Usage: mkdocs [OPTIONS] COMMAND [ARGS]...

Error: No such command "help".

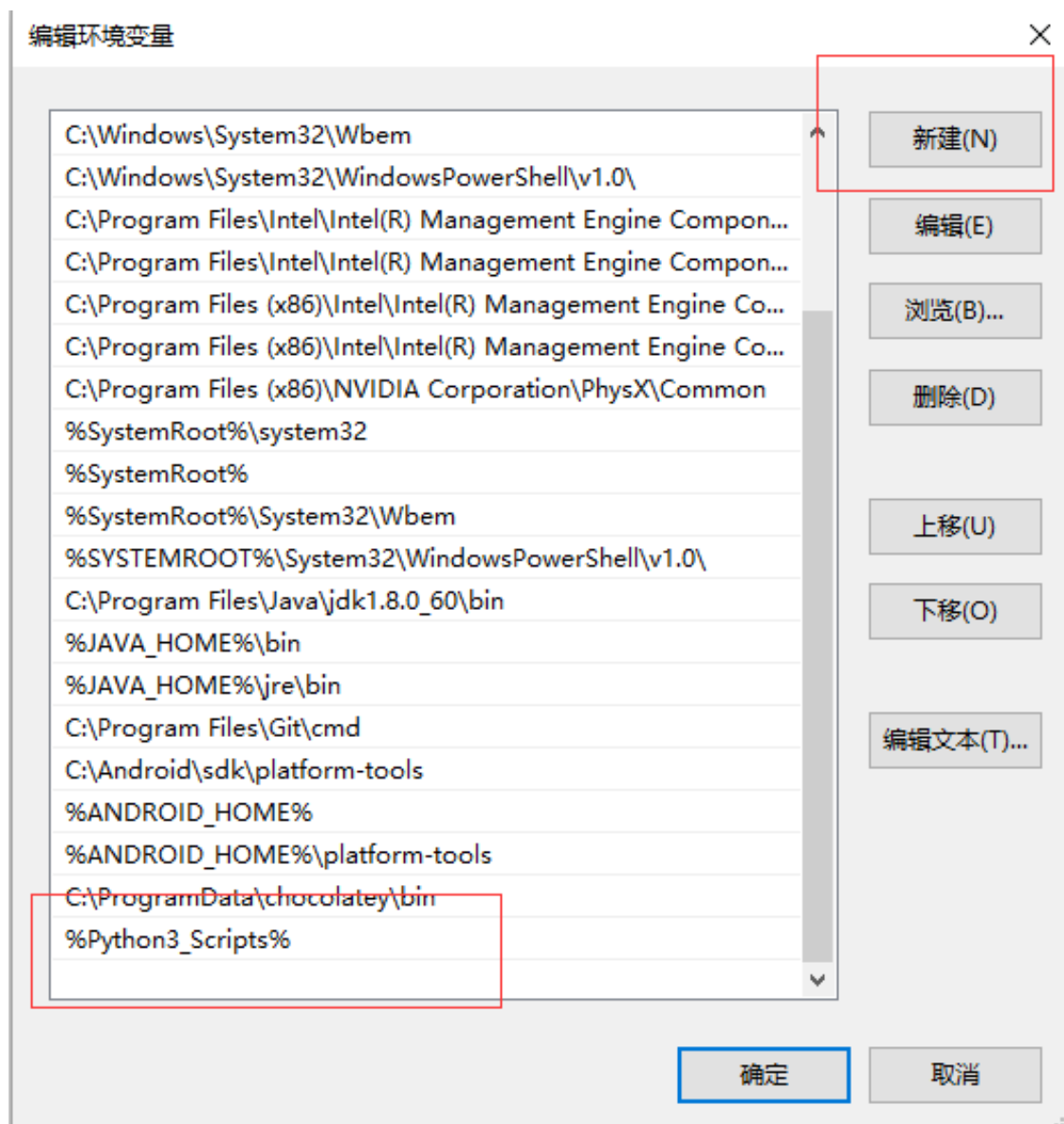
C:\ProgramData\chocolatey\lib\python3\tools\Scripts>mkdocs -V
mkdocs, version 0.15.3
```

添加环境变量：

控制面板->系统和安全->系统->高级系统设置->环境变量，配置系统环境变量，然后加入Path中：







然后重新打开cmd，输入mkdocs -V，如果显示出来了版本，那么恭喜你，终于安装成功了！！！！

## MkDocs的使用

初步试用：根据官方文档的步骤创建和使用MkDocs

官方文档：<http://www.mkdocs.org/#mkdocs>

中文文档：[http://markdown-docs-zh.readthedocs.io/zh\\_CN/latest/](http://markdown-docs-zh.readthedocs.io/zh_CN/latest/)

建议先看中文文档了解过程，然后根据官方文档操作，因为官方文档总是最新的。

# 常用命令

- **build**: 构建MkDocs文档
- **gh-deploy**: 将文档部署到GitHub页面上
- **json**: 将MkDocs文档构建成JSON文件
- **new**: 创建新的MkDocs工程
- **serve**: 运行内建的开发服务器
- **help**: 帮助

## My Test

### 1. 使用mkdocs命令创建工程

```
1 mkdocs new 工程名
2 1
```

```
C:\Users\Kevin>mkdocs new my--project
INFO      - Creating project directory: my--project
INFO      - Writing config file: my--project\mkdocs.yml
INFO      - Writing initial docs: my--project\docs\index.md
```

结果显示创建了工程目录:my-project, 配置文件mkdocs.yml, 初始化文件(主页)index.md。

### 1. 使用serve命令开启服务器

```
1 mkdocs serve
2 1
```

注: 此时开启的服务, 在局域网中是无法访问的, 使用命令 `mkdocs serve --dev-addr:0.0.0.0:9000`, 可以将项目部署到局域网中

```
命令提示符 - mkdocs serve

C:\Users\Kevin>mkdocs new my--project
INFO - Creating project directory: my--project
INFO - Writing config file: my--project\mkdocs.yml
INFO - Writing initial docs: my--project\docs\index.md

C:\Users\Kevin>cd my--project

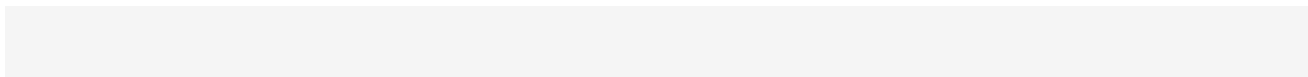
C:\Users\Kevin\my--project>mkdocs serve
INFO - Building documentation...
INFO - Cleaning site directory
[I 160831 11:39:43 server:281] Serving on http://127.0.0.1:8000
[I 160831 11:39:43 handlers:59] Start watching changes
[I 160831 11:39:43 handlers:61] Start detecting changes
[W 160831 11:40:19 web:1971] 404 GET /fonts/fontawesome-webfont.woff2?v=4.5.0 (1
27.0.0.1) 1.00ms
[I 160831 11:40:19 handlers:132] Browser Connected: http://127.0.0.1:8000/
[W 160831 11:40:37 web:1971] 404 GET /fonts/fontawesome-webfont.woff2?v=4.5.0 (1
27.0.0.1) 1.00ms
[I 160831 11:40:37 handlers:132] Browser Connected: http://127.0.0.1:8000/
```

1. 在浏览器中打开 <http://127.0.0.1:8000/> , 你将看到以下页面:



内建服务器支持在配置文件、文档目录或主题发生改变时自动载入并重新生成文档，编辑 `docs/index.md` 文件并保存，刷新浏览器你将看到文档被同步更新。现在可以开始编辑配置文件 `mkdocs.yml` 了，把 `site_name` 改成其他内容并保存文档。

## 1. 添加头部的导航条



```
1 curl 'jaspervdj.be/lorem-markdownum/markdown.txt' > docs/about.md
2 1
```

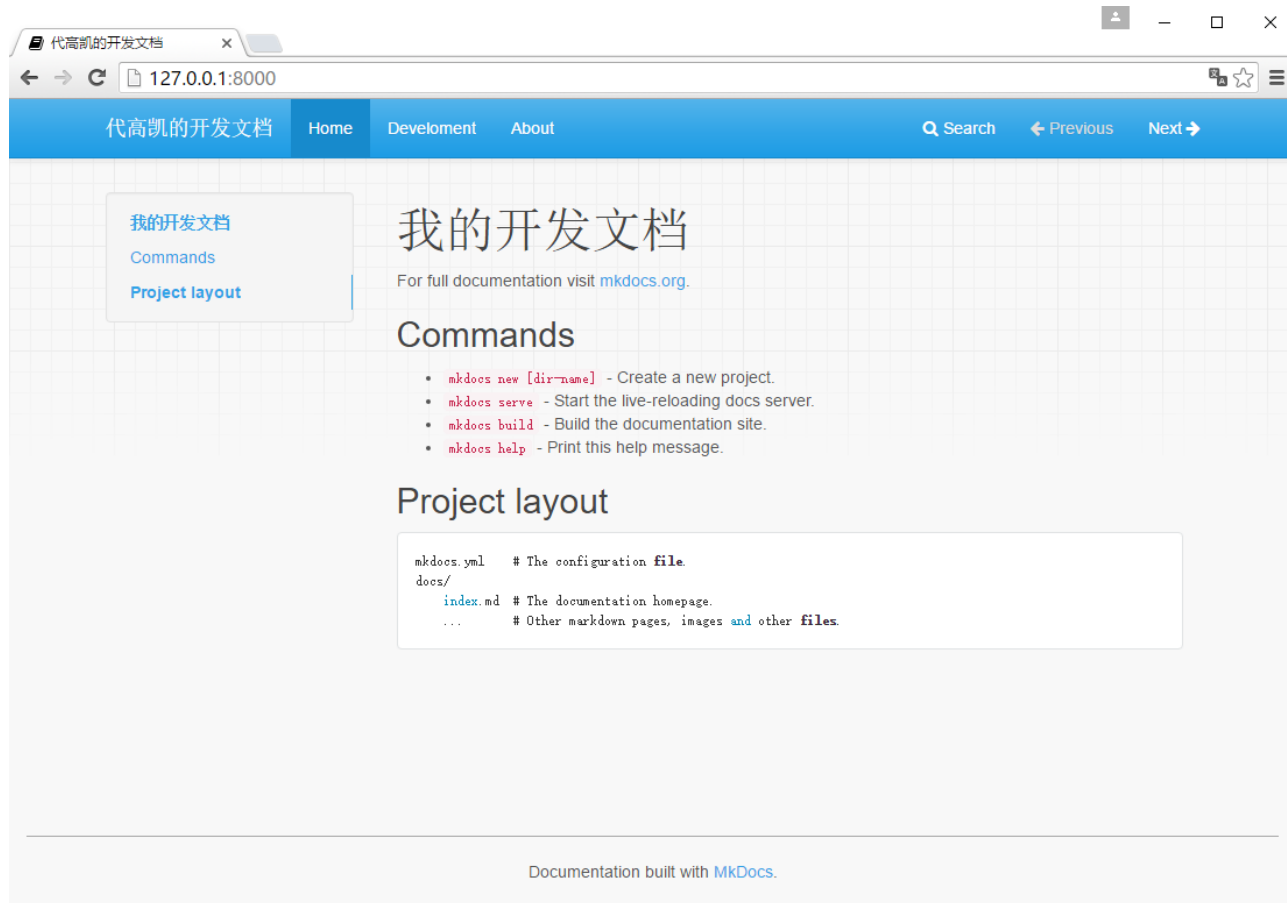
将这条语句粘贴到命令行，然后回车，你会发现，提示不是内部命令，然后小编用Everything搜索了，然并卵。后来发现，执行完这句话以后，在工程的docs/目录下创建了about.md文档。为了测试，我又创建了一个文档：

```
1 curl 'jaspervdj.be/lorem-markdownum/markdown.txt' > docs/development.md
2 1
```

然后，这个时候，需要在配置文件mkdocs.yml中配置导航条：

```
1 site_name: 代高凯的开发文档
2 pages:
3 - Home: index.md
4 - Develoment: development.md
5 - About: about.md
6 1
7 2
8 3
9 4
10 5
```

当Ctrl+s保存后，就会发现网页变成了：

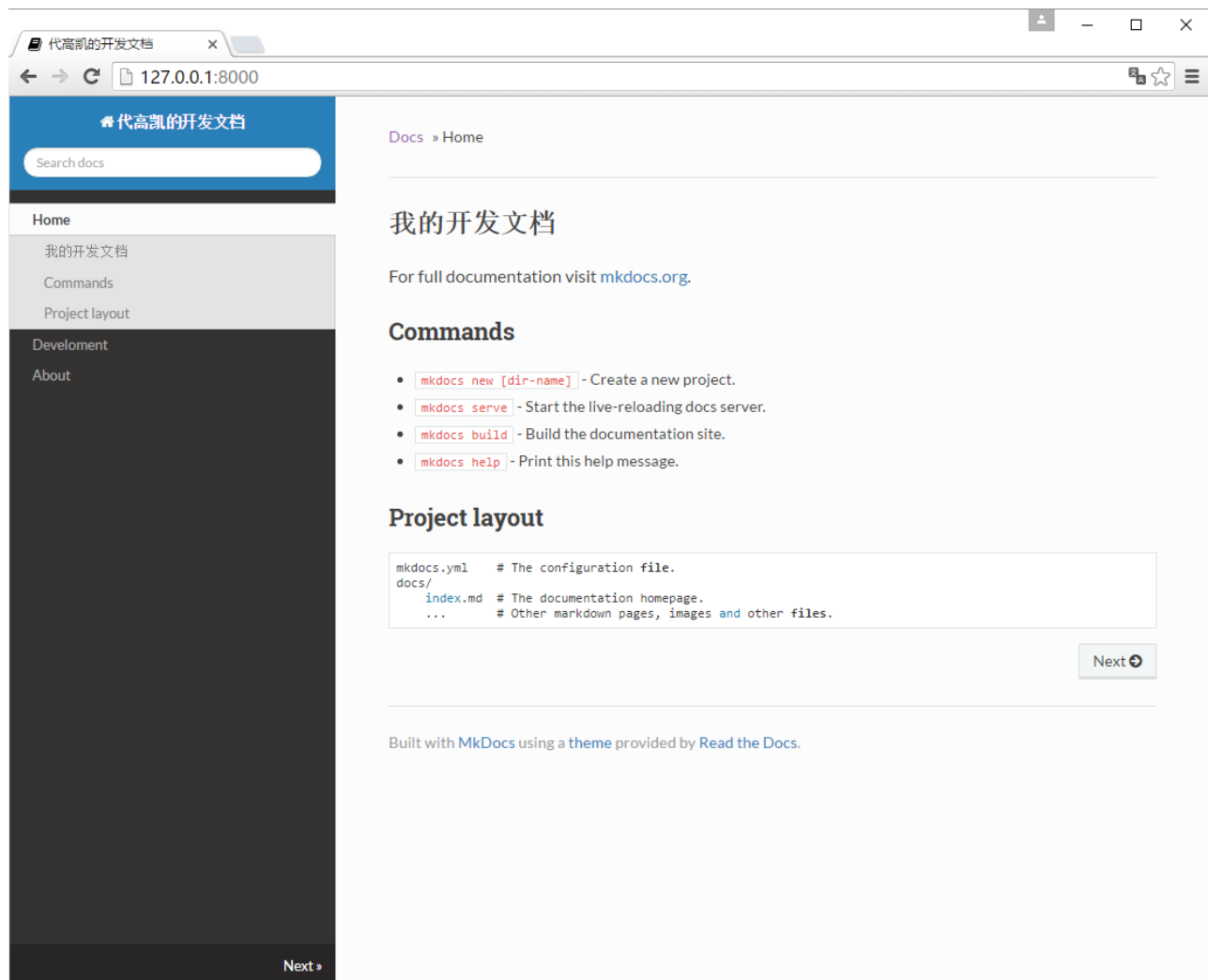


瞬间就变了，十分高效！！！而且，在这个过程中，不用重启内置的服务器，也不用刷新界面。

## 1. 主题

在配置文件中添加主题：

```
1 theme: readthedocs
2 1
```



其实我一直觉得这个主题超级丑，不过官方给出的demo是这个，而且我公司暂时也用的这个，不过还是很丑。

## 1. 站点生成

```
1 mkdocs build
2 1
```

该命令创建了一个 `site` 新目录，可以到项目文件夹中查看，都被保存在了工程名/site目录下。注意源码被分别输出为 `index.html` 和 `about/index.html`。主题中的其他文件也被复制到了 `site` 目录中。

如果你使用 `git` 等版本控制系统，你可能不希望提交构建之后的文档到版本库，在 `.gitignore` 中添加 `site/` 即可忽略该目录。

一段时间后，可能有文件被从源码中移除了，但是相关的文档仍残留在 `site` 目录中。在构建命令中添加 `--clean` 参数即可移除这些文档。

```
1 makedocs build --clean
2 1
```

## 1. 发布

MkDocs 生成的文档只包含静态文件，因此你可以将文档部署到任意地方。[GitHub project pages](#) 和 [Amazon S3](#) 是不错的选择，只需上传 `site` 目录到你发布的位置即可。

- 如果是公司的项目，项目文档不能对外开放，你可以上传到公司的GitLab上。
- 如果是个人的项目，你可以上传到GitHub上。

## 注意事项

1. 如果上传图片，目前我所知道的有两种方式，第一，就是将图片上传到某个地方，比如通过markdown来上传，然后复制过来。。。小编刚开始的时候就是用的这么菜的方法，没办法，为了所见即所得，忍了；
2. 后来小编发现如果创建站点的话，将图片和文档放在同一个文件夹中即可，或者写清楚图片的路径即可。优点就是，比较方便，缺点就是如果直接复制出来.md文档，图片显示不出来。
3. 如果修改了文字，那么保存的时候会瞬间刷新，如果是图片的话，有时候需要重启本地内置的服务器。

来源：<http://blog.csdn.net/kevindgk/article/details/52388542>