

Github Pages + Jekyll 独立博客一小时快速搭建&上线指南

分享你的知识和观点

Posted by Bebop.C on March 26, 2016

只要一小时？！

人生道路上布满了坑，于是有了人生导师。

美丽的地球上布满了坑，于是有了 *Google Earth*。

使用 *Github Pages* 搭建独立博客的过程中布满了坑，所以有了这篇指南。

我在自己查找资料搭建的过程中发现了许多大大小小的坑，大部分是因为软件版本、技术的升级、网络环境的影响、设计方案的变化、操作顺序的不当而引起，于是想整理一篇快速而准确的攻略以供各种同我一样的小白也能轻松获取知识，搭建自己知识分享的平台。

这篇文章能帮助你一小时快速上线（当然要在网络环境良好的情况下，有时候网络环境不好可能需要自备梯子），如果不想自己折腾，直接fork模版，就可以写文章了，如果想把自己的博客打理得更符合自己的口味儿，大家还需要多多学习和实践。

此篇攻略尽力做到清晰明了，但也必有不足之处，遇到问题欢迎大家多多留言交流提问(´·ω·)ﾉ。

版权声明：欢迎分享，转载请注明此文链接。

<http://playingfingers.com/2016/03/26/build-a-blog/>

(<http://playingfingers.com/2016/03/26/build-a-blog/>)

目录

1. Why Github Pages + Jekyll
2. 注册Github
3. 关于域名
4. 安装Git环境
5. 建立个人GitHub Pages
6. 写博客
7. 进阶：安装Jekyll本地编译环境
8. 进阶：添加评论、分享功能
9. 扩展阅读

Why Github Pages + Jekyll

Git & Github

Git是一种分布式版本控制系统，由大名鼎鼎的Linux操作系统创建人Linus Torvalds编写，当时的目的是用来管理Linux内核的源码。

Github是全球知名的使用Git系统的一个免费远程仓库（Repository），倡导开源理念，若要将自己的代码仓库私有则需付费。

GitHub Pages

托管在Github上的项目页面看上去往往是这样：

 CONDUCT.markdown	better file copying	26 days ago
 Gemfile	Lock jemoji to v0.5.1 while we figure out the issue with HTML::Pipeline.	8 days ago
 History.markdown	Update history to reflect merge of #3849 [ci skip]	9 hours ago
 LICENSE	Update copyright notices to 2016 [ci skip]	3 months ago
 README.markdown	Improved content in "License" section in README.markdown	13 days ago
 Rakefile	Rakefile: siteify_file should allow overrides from task [ci skip]	15 days ago
 jekyll.gemspec	Include .rubocop.yml in Gem	2 months ago

README.markdown

Jekyll

gem v3.1.2 build passing coverage 88% code climate 3.3 dependencies out-of-date security no warnings

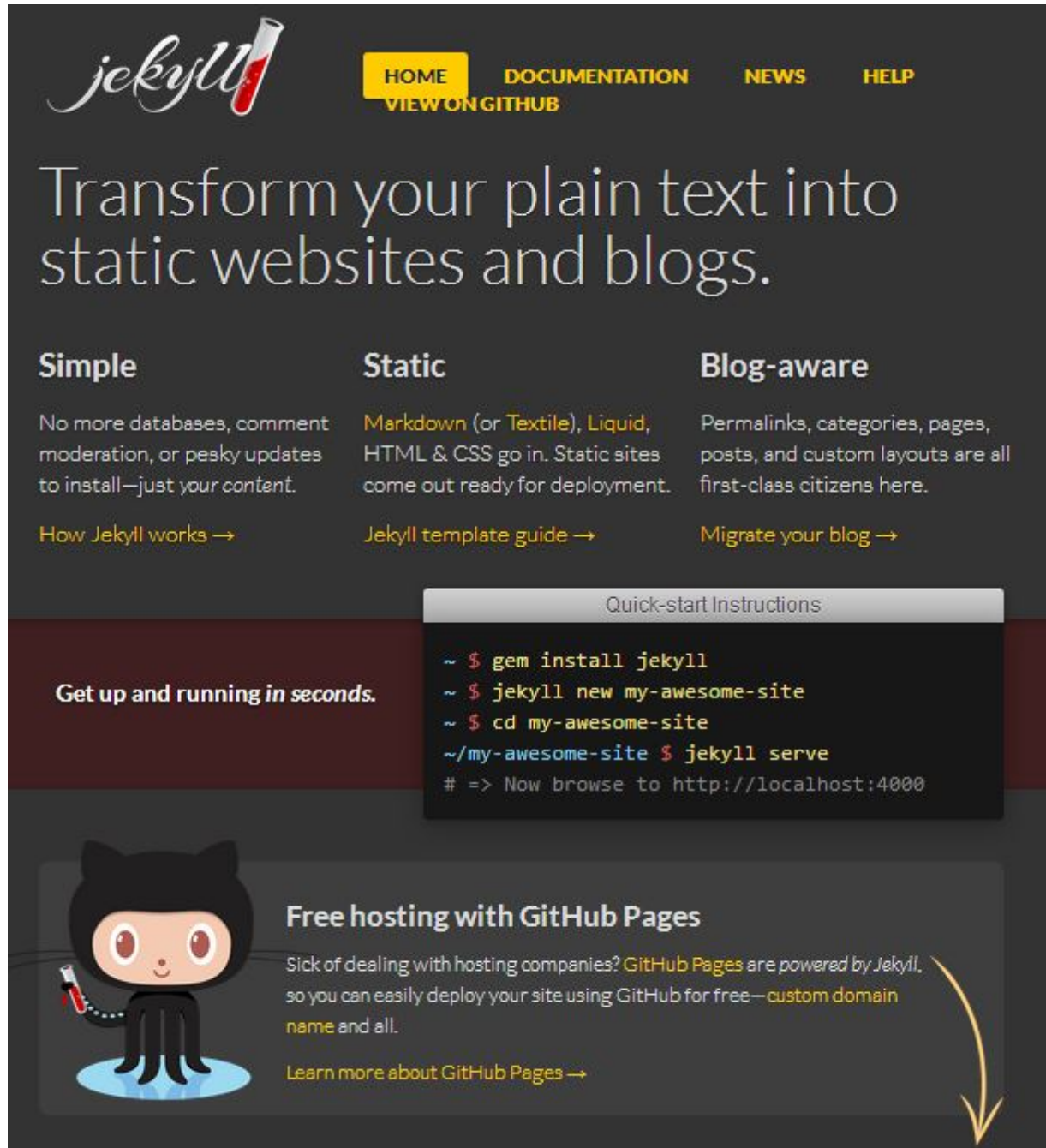
Jekyll is a simple, blog-aware, static site generator perfect for personal, project, or organization sites. Think of it like a file-based CMS, without all the complexity. Jekyll takes your content, renders Markdown and Liquid templates, and spits out a complete, static website ready to be served by Apache, Nginx or another web server. Jekyll is the engine behind [GitHub Pages](#), which you can use to host sites right from your GitHub repositories.

What does the geek say?!

对于非专业人士看到这样的页面有80%只能默默点叉了。

于是GitHub Pages诞生了。它是Github提供给托管项目的开发者一个更个性化展示自己项目的方法，使用GitHub Pages服务可以编写同样是托管在Github上的静态网页。

比如上面Jekyll项目的GitHub Pages长成这样：



Github Pages分为用户、组织、项目三种网站，我们的Blog要用到的是 User Pages site (<https://help.github.com/articles/user-organization-and-project-pages/>)，即用户网站。

Jekyll

下面是来自 Jekyll中文文档站 (<http://jekyll.bootcss.com/>) 的介绍：

将纯文本转化为静态网站和博客。

简单 – 无需数据库、评论功能，不需要不断的更新版本——只用关心你的博客内容。

静态 – 只用 *Markdown* (或*Textile*)、*Liquid*、*HTML* & *CSS* 就可以构建可部署的静态网站。

我们遵循Jekyll规范编写网站源码，上传到Github上，Github会自动进行编译出最终的网站文件，给你一个新世界！

Tip: 如果本地安装Jekyll编译环境，便可实时预览网站，不必每修改一点都要经过上传刷新才能看到结果，大大方便了我们的调试。

Github Pages + Jekyll 方案适合我吗？

Github Pages + Jekyll 方案的优点：

*Github*免费托管源文件，自动编译符合*Jekyll*规范的网站。

引入版本管理，修改网站更加安全方便。

支持 *Markdown* (<http://sspai.com/25137>)，编写具有优美排版的文章。

Github Pages + Jekyll 方案的不足：

需要学习一些基础的*Git*命令。

若要自己全权制作主题的话需要懂一点网页开发。

由于生成的是静态网页，若要使用动态功能，如评论功能（下文解决），则要使用第三方服务。

所以，如果你只是想做一个分享见闻心得的博客，这个方案非常适合你。

注册Github

到 Github的官方网站 (<https://github.com/>) 注册账户，记住自己的用户名，后面会常用到。

Tip: 有时网络状况不好可以喝杯牛奶再来。

关于域名

符合下面条件，可以跳过此部分：

1. 已有域名。
2. 不想使用独立域名，直接使用Github Pages提供的域名访问：

`http://{username}.github.io` //用你的 *Github* 用户名替换网址中的 `{username}`

购买域名

之所以把购买域名放在前面，是因为购买和DNS配置过后可能会有服务延迟，为了让大家尽快看到自己的成果，可以先买域名。

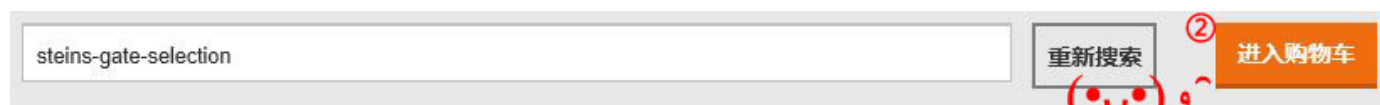
大家可以自由选择域名注册商，这里介绍GoDaddy上的购买流程。

进入GoDaddy中文网站 (<https://sg.godaddy.com/zh/>)（当然英文也可以），在搜索框输入一个霸气的名字，点击“域名搜索”。

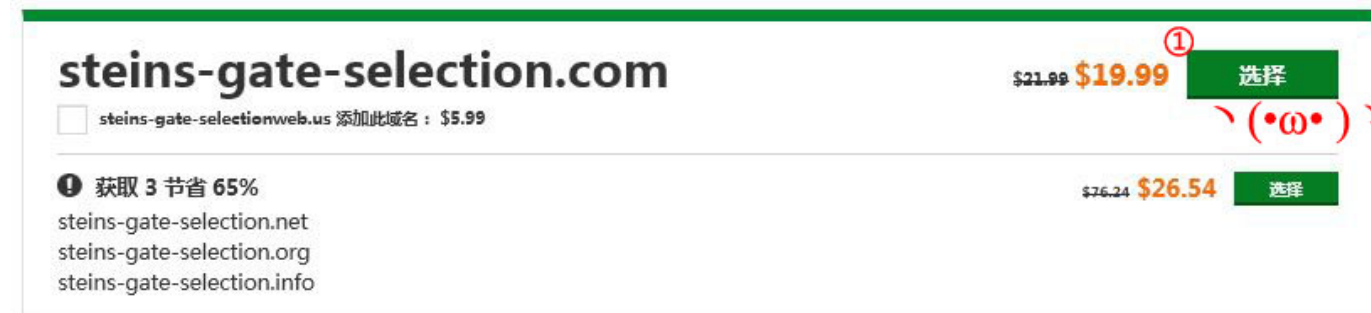


如果幸运的话，还没有人注册过这个域名。

我们只需购买一个 .com 域名即可，点击第一个“选择”，再点击“进入购物车”。



太棒了！您的域名可用。抢在他人下手之前购买此域名。



出现GoDaddy的增值服务，我们不需要的的话，直接点击右上角的“进入购物车”。

下一步您将怎么处理您的域名？

进入购物车

向下滚动进入购物车

第 1 步 | 隐私

保护我的个人信息

起价 \$11.99/域名/年

防止您的个人信息落入他人之手。

不，谢谢

到了购买页面，可以自由选择购买期限，默认两年，不同时间优惠不同。注意这里右侧有个地方可以添加“促销码”，大家可以用搜索引擎找找，有很多可用的优惠，不同的促销码可能会对支付方式和年份等有所限制（无限制时可以使用支付宝、信用卡、银联银行卡等）。

产品	期限	单价	小计
<div><div>WWW</div><div>steins-gate-selection.com</div><div>.COM 域名注册</div><div>4% 折扣</div><div>*外加 ICANN 管理费 \$0.27/年</div></div> <div><div>保护您的个人信息。 ⓘ</div><div><div>公开我的个人信息</div><div>享受域名隐私保护服务，每年只需支付 \$11.99！节省 \$3.00！</div></div></div>	2 年	\$20.72/年* 节省 \$2.00	\$41.44 ⊗ 删除
<div><div>配套的未被注册的 (.net, .co, .info) 域名</div><div>steins-gate-selection.net - \$24.99 \$12.52 + 添加</div><div>steins-gate-selection.co - \$43.99 \$9.61 + 添加</div><div>steins-gate-selection.info - \$25.26 \$4.38 + 添加</div><div>添加所有域名只需要花费 \$94.24 \$26.51</div></div>			

小计 \$41.44

估计税费和费用 \$0.54

促销码

添加

总计 (SGD) \$41.98

去付款 >

总计节省费用 \$2.00

认真核对好购买项目和金额后，点击“去付款”，进行信息注册和结算。

注意！邮箱名字很重要，后期激活和修改密码都必须通过邮箱，请确保无误，可不要让银子白白流走啊。

购买完成后去刚才填写的邮箱收信激活。

设置DNS

重新登录你刚注册的帐号，点击右上角用户中心里的“我的产品”。



此时已经可以看到我们购买的域名了。

为了防止不可控因素导致GoDaddy的域名服务器无法访问，我们可以把域名解析的任务转给国内稳定的域名服务商。

这里推荐使用 DNSPod (<https://www.dnspod.cn/>) 的服务，稳定且免费。

在“我的产品”界面找到刚才注册的域名，点击“管理DNS”。



在“域名服务器”处点击“管理”。

设置	DNS 区域文件	联系人
自动续费 ⓘ	标准版：关闭 已扩展：关闭 管理	
锁定 ⓘ	打开 管理	
域名服务器 ⓘ	F1G1NS1.DNSPOD.NET F1G1NS2.DNSPOD.NET 更新日期：20/3/2016 管理	

按照下图设置好DNSPod的域名服务器。

域名服务器设置

PLAYINGFINGERS.COM

域名服务器将您的域名指向它所在的位置。

设置类型:

- ☐ 标准
 GoDaddy 主机托管、域名转址和域名停放收益。
- ☒ **定制**
 可自定义的域名服务器设置。

域名服务器:

[编辑域名服务器](#)

#	域名服务器	状态
1	F1G1NS1.DNSPOD.NET	已激活 
2	F1G1NS2.DNSPOD.NET	已激活 

[添加域名服务器](#)

设置好后进入 DNSPod (<https://www.dnspod.cn/>)
 注册帐号后登录，进入“域名解析”，点击“添加域名”。

添加域名

分组

更多服务

new

更多操作

☐ 域名

☐ ☆

请输入您要解析的域名

确定

取消

☐ ☆

playingfingers.com

添加好之前购买的域名，点击你的域名进入记录管理。按照下图添加每条记录。

记录管理

域名设置

解析量统计

自定义线路

线路分组

添加记录

暂停

启用

删除

快速查找记录

<input type="checkbox"/>	主机记录	记录类型	线路类型	记录值	权重	MX优先级	TTL	操作
<input type="checkbox"/>	@	A	默认	192.30.252.153	-	-	600	删除 暂停
<input type="checkbox"/>	@	A	默认	192.30.252.154	-	-	600	删除 暂停
<input type="checkbox"/>	@	NS	默认	f1g1ns1.dnspod.n...	-	-	86400	删除 暂停
<input type="checkbox"/>	@	NS	默认	f1g1ns2.dnspod.n...	-	-	86400	删除 暂停
<input type="checkbox"/>	www	CNAME	默认	chaosinmotion.git...	-	-	600	删除 暂停

前两条A型记录值 192.30.252.153 和 192.30.252.154 是GitHub的服务器地址，可以在这里 (<https://help.github.com/articles/troubleshooting-custom-domains/#dns-configuration-errors>)查到。

NS类型记录是默认不可变的。

CNAME类型的记录值填写 {username}.github.io

Tip: 用你的Github用户名替换上面的 {username}

安装Git环境

本文主要介绍windows环境下的安装，其他环境可参考廖大神的教程

(<http://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000/0013739>)

安装git for windows

进入下载页面 (<https://git-for-windows.github.io/>)

安装好以后在开始菜单中找到Git Bash并打开。



在打开的命令行窗口 (Shell) 内执行以下命令，设置你的git用户名和邮箱：

```
$ git config --global user.name "{username}"           // 用你的用户名替换 {username}  
$ git config --global user.email "{name@site.com}"     // 用你的邮箱替换 {name@site.com}
```

SSH配置

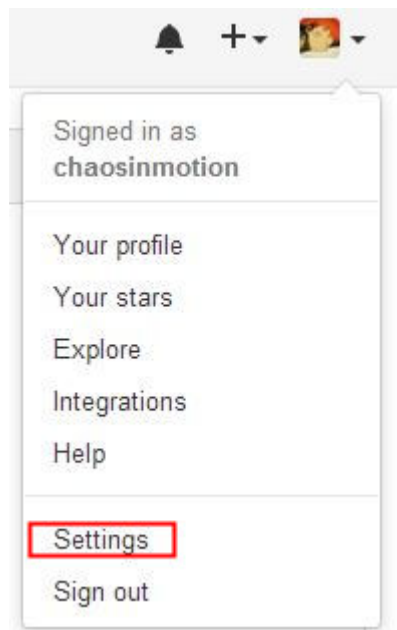
为了和Github的远程仓库进行传输，需要进行SSH加密设置。

在刚才打开的Shell内执行：

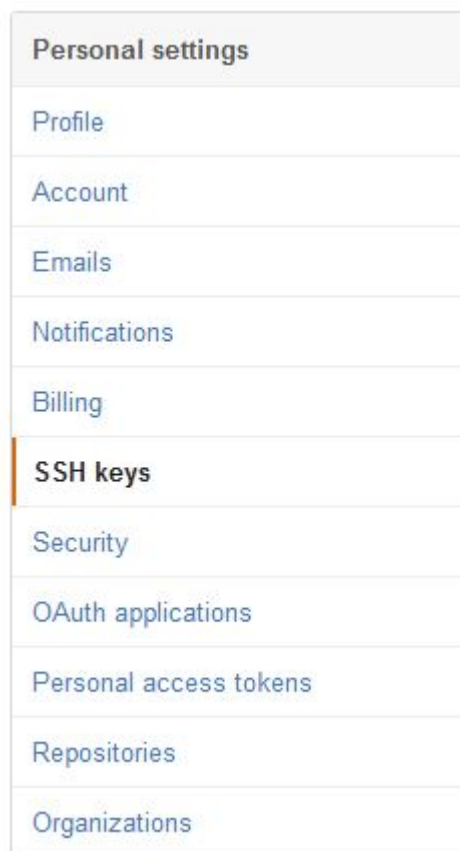
```
$ ssh-keygen -t rsa -C "{name@site.com}"           // 用你的邮箱替换 {name@site.com}
```

可以不输入其他信息，一直敲回车直到命令完成。这时你的用户目录 (win7以上系统默认在 C:\Users\你的计算机用户名) 内会出现名为 .ssh 的文件夹，点进去能看到 id_rsa 和 id_rsa.pub 两个文件，其中 id_rsa 是私钥，不能让怪人拿走， id_rsa.pub 是公钥，无需保密 (原理请自行参看密码学.....我相信你也不会看)。

接下来用你的浏览器登录Github，点击右上角的“Settings”：



用文字处理软件打开刚才的 `id_rsa.pub` 文件，复制全部内容。
点击“SSH Keys”，“Add SSH Key”，将复制的内容粘贴在Key中，点“Add Key”确定。



Title

Key

Begins with 'ssh-rsa', 'ssh-dss', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-

Add SSH key

至此SSH配置完毕。

建立个人 GitHub Pages

建立基于Jekyll的个人Github Pages有两条路线：

- 1.自己学习Jekyll教程和网页设计，设计绝对自我基因的网页。
- 2.Fork（*Git*系统的创建分支，简单来说是把当前仓库复制一份到你的仓库，你可以进行修改，因为你的仓库是原来仓库的新的分支）已有的开源博客仓库，在巨人的肩膀上进行符合自我的创作。

建议和我一样的小白们可以从第二条路线学起。

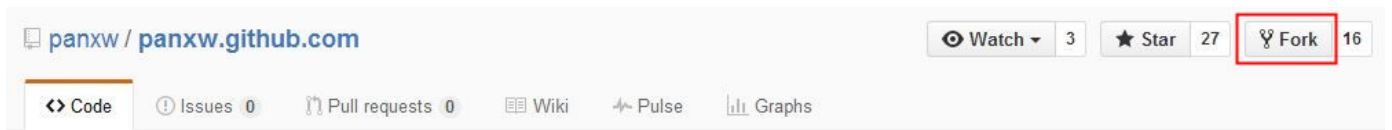
挑选“模版”

Jekyll项目的wiki页面 (<https://github.com/jekyll/jekyll/wiki/sites>)给出了大量优秀的风格各异的网站，这里以 Panxw's blog 为例讲解。

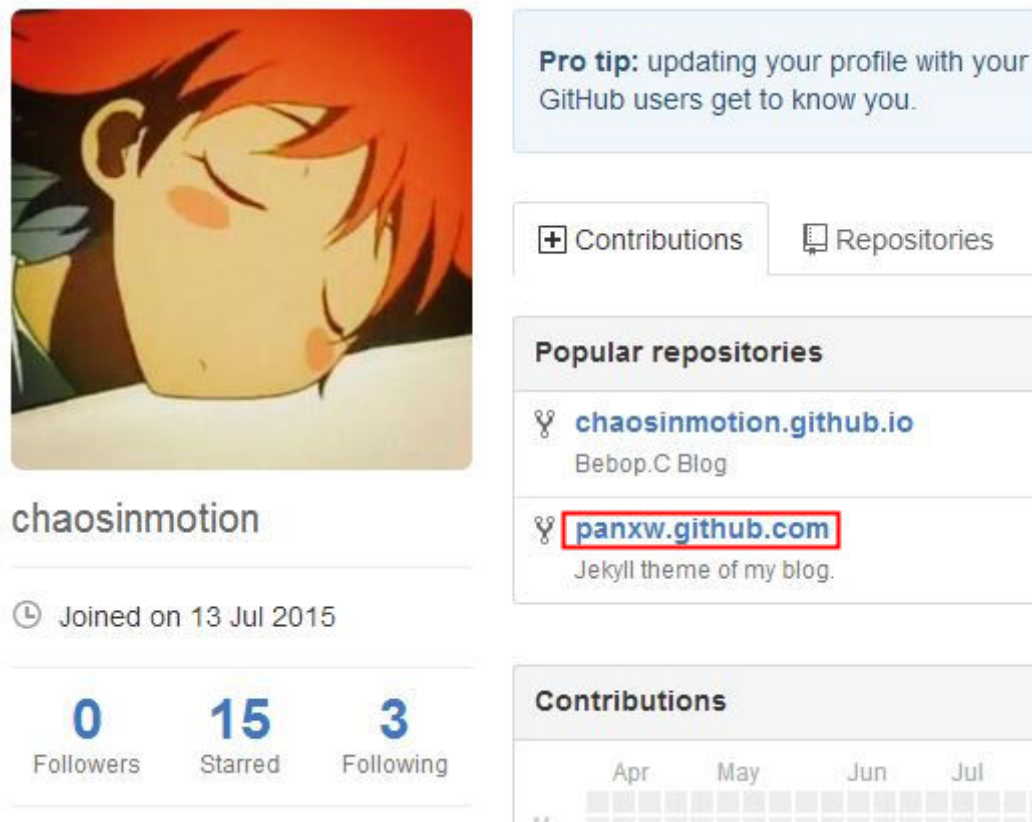
进入wiki页，点击Panxw's blog右边的source链接，进入到作者的模版仓库。

- [Anil Wadghule \(source\)](#)
- [Panxw's blog \(source\)](#) – By Jekyll & GitPages, mobile valid(Chinese, 中文).
- [9leg \(source\)](#) - Java Blog (chinese,中文).
- [Elijah Miller / jqe \(source\)](#)
- [PseudoWeb.Net \(source\)](#)

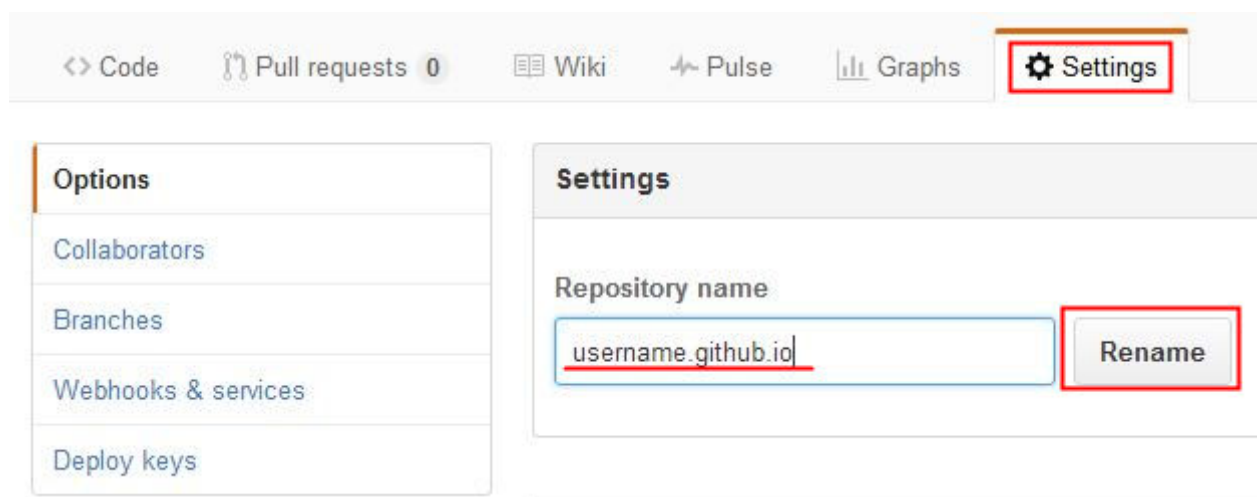
点击右上角的“Fork”。



然后在你的主页里找到你刚才Fork的分支，点进去：



点击“Settings”，将“Repository name”改为 {你的Github用户名}.github.io，点击“Rename”。



此时你会发现已经可以通过 `http://{你的Github用户名}.github.io` 访问你fork下来的网站啦！

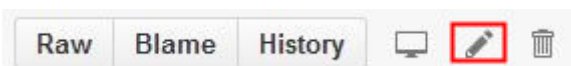
绑定域名

接下来让你能用之前购买的域名来访问你的网站。

在你的代码仓库中找到 CNAME 文件，点击它：

posts	update jekyll theme
.gitignore	add template
404.html	modify theme
CNAME	modify theme
README.md	Update README.md
_config.yml	update jekyll template

点击右方的铅笔图标，把文件内容改成你购买的域名，比如 `www.awesome.com`，这样你就可以用自己的域名访问你的Github Pages啦！



写博客

同步仓库

我们不可能一切编写都像刚才更改CNAME那样在网页上直接进行，那样会十分受限及效率低下，我们需要把托管在Github上的仓库同步到本地计算机来。

再次打开Git Bash，输入以下命令切换到你想放置本地代码仓库的位置：

```
$ cd {本地路径} // 比如: cd e:/workspace
```

clone（克隆）你自己的远程仓库：

```
$ git clone https://github.com/{username}/{username}.github.io.git // 用你的Github
```


结果如下：

```
Admin@Admin MINGW64 ~
$ cd e:/workspace

Admin@Admin MINGW64 /e/workspace
$ git clone https://github.com/chaosinmotion/chaosinmotion.github.io.git
Cloning into 'chaosinmotion.github.io'...
remote: Counting objects: 2004, done.
remote: Compressing objects: 100% (192/192), done.
remote: Total 2004 (delta 102), reused 0 (delta 0), pack-reused 1809
Receiving objects: 100% (2004/2004), 35.56 MiB | 606.00 KiB/s, done.
Resolving deltas: 100% (1270/1270), done.
Checking connectivity... done.

Admin@Admin MINGW64 /e/workspace
$ |
```

这时所有远程仓库里的源码都拷贝到 `e:/workspace/{username}.github.io` 这个文件夹里来了。

Tip: 如果clone失败有可能是网络原因，可以过一个时段再试，或者寻找梯子。

撰写博文

打开本地仓库的 `_posts` 文件夹，你的所有博文都将放在这里，写新博文只需要新建一个标准文件名的文件，在文件中编写文章内容。比如我们fork的模版中 `_posts` 文件夹里有一篇 `2016-03-23-hello-world.markdown`，你的文件命名也要严格遵循 `年-月-日-文章标题.文档格式` 这样的格式，**尤其要注意月份和日期一定是两位数。**

推荐使用Markdown语言写文章，windows下推荐MarkdownPad这个软件编写Markdown文本。

最开始写可以直接模仿别人的博文语法，更多Markdown语法可参考 [认识与入门Markdown \(http://sspai.com/25137\)](http://sspai.com/25137)。

简单Git命令

这里只介绍快速修改上传博客的方法，详细的Git学习可以参考文末给出的扩展阅读。

当你使用Git Bash对你的本地仓库进行操作时，先用 `cd` 命令将你的工作目录设置到你要操作的本地仓库

```
$ cd {你刚才clone下来的项目文件夹路径}
```

每当你对本仓库里的文件进行了修改，只需在Bash中依次执行以下三个命令即可将修改同步到Github，刷新网站页面就能看到修改后的网页：

```
$ git add .  
$ git commit -m "statement" //此处statement填写此次提交修改的内容，作为日后查阅  
$ git push origin master
```

到此你已经可以发表文章到你的个人博客啦！！如有更多需求，可以继续看下面的进阶部分。

进阶：安装Jekyll本地编译环境

每次修改了本地文件，都要至少经过三个命令和服务器的延迟刷新才能看到修改的效果，是不是有点疼？

如果你觉得“疼！”那么这部分就是你的解药。

环境安装

本文主要介绍windows环境下的安装，Mac和Linux环境下可以使用自带的包管理器进行安装，可参考这篇文章 (<http://www.cnblogs.com/daguo/p/4097263.html>)。

1. 安装 Ruby (<http://rubyinstaller.org/downloads/>)

RubyInstallers

Archives»

Not sure what version to download? Please read the right column for recommendations.



Ruby 2.2.4

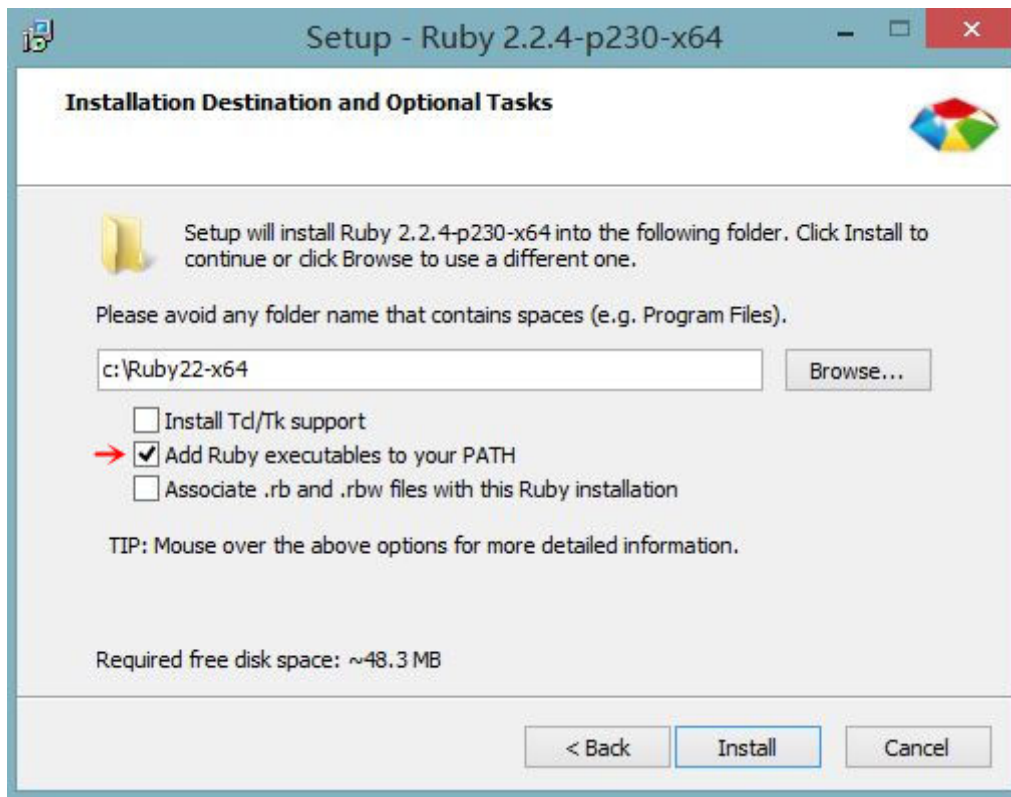
如果你的系统是32位请装这个



Ruby 2.2.4 (x64)

如果你的系统是64位请装这个

注意：这里一定要勾选添加到环境变量PATH！



2. 安装 RubyGems (<https://rubygems.org/pages/download>)

windows下下载ZIP格式较为方便，下好后解压到本地任意路径，下面以 {unzip-path} 代替你解压的路径。打开windows的cmd终端（按win+R快捷键打开“运行”，输入cmd，确定），输入命令：

```
$ cd {unzip-path} //如果你没有解压在C盘，windows的终端切换到其他盘需要写为 cd /d {unzip-path}
$ ruby setup.rb
```

3. 安装Jekyll

在终端里输入

```
$ gem install jekyll
```

4. 安装jekyll-paginate

在终端里输入

```
$ gem install jekyll-paginate
```

如遇到以下错误，说明网络不通：

```
ERROR: While executing gem ... (Gem::RemoteFetcher::FetchError)
Errno::ECONNRESET: An existing connection was forcibly closed by the remote host.
```

这时候有三种解决方法：

1. 等天气好的时候再来
2. 架梯子
3. [推荐] 更换为Ruby China (<https://gems.ruby-china.org/>)源，点进去会有详细的设置方法。（感谢 @mortred 博友告知淘宝源已不可用，建议使用Ruby China的源，上不去的话换个时间就好。）

开启本地实时预览

上一小节的安装都完成以后，在终端中输入命令

```
$ cd {local repository} // {local repository}替换成你的本地仓库的目录
$ jekyll serve
```

- 1.如遇到以下错误，在仓库文件 `_config.yml` 中加入一句 `gems: [jekyll-paginate]` 即可。

```
Deprecation: You appear to have pagination turned on, but you haven't included the `;
```

- 2.如遇到以下错误

```
jekyll 3.1.2 | Error: Permission denied - bind(2) for 127.0.0.1:4000
```

说明有程序在占用这个本地端口，这时候输入命令

```
$ netstat -ano
```

可以看到如下进程与所占用端口的对应情况，找到本地地址为 `127.0.0.1:4000` 的记录，看到该条记录的PID为6668 (当然你的和我的不一样)。

协议	本地地址	外部地址	状态	PID
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING	828
TCP	0.0.0.0:443	0.0.0.0:0	LISTENING	4132
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:902	0.0.0.0:0	LISTENING	3332
TCP	0.0.0.0:912	0.0.0.0:0	LISTENING	3332
TCP	0.0.0.0:1025	0.0.0.0:0	LISTENING	564
TCP	0.0.0.0:1026	0.0.0.0:0	LISTENING	364
TCP	0.0.0.0:1027	0.0.0.0:0	LISTENING	476
TCP	0.0.0.0:1028	0.0.0.0:0	LISTENING	1628
TCP	0.0.0.0:1037	0.0.0.0:0	LISTENING	704
TCP	0.0.0.0:1047	0.0.0.0:0	LISTENING	696
TCP	0.0.0.0:5357	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:47984	0.0.0.0:0	LISTENING	6620
TCP	0.0.0.0:47989	0.0.0.0:0	LISTENING	6620
TCP	127.0.0.1:1029	127.0.0.1:5354	ESTABLISHED	1892
TCP	127.0.0.1:1030	127.0.0.1:5354	ESTABLISHED	1892
TCP	127.0.0.1:4000	0.0.0.0:0	LISTENING	6668
TCP	127.0.0.1:4000	127.0.0.1:34874	CLOSE_WAIT	6668
TCP	127.0.0.1:5354	0.0.0.0:0	LISTENING	1924
TCP	127.0.0.1:5354	127.0.0.1:1029	ESTABLISHED	1924
TCP	127.0.0.1:5354	127.0.0.1:1030	ESTABLISHED	1924
TCP	127.0.0.1:8307	0.0.0.0:0	LISTENING	4132
TCP	127.0.0.1:9990	0.0.0.0:0	LISTENING	2128

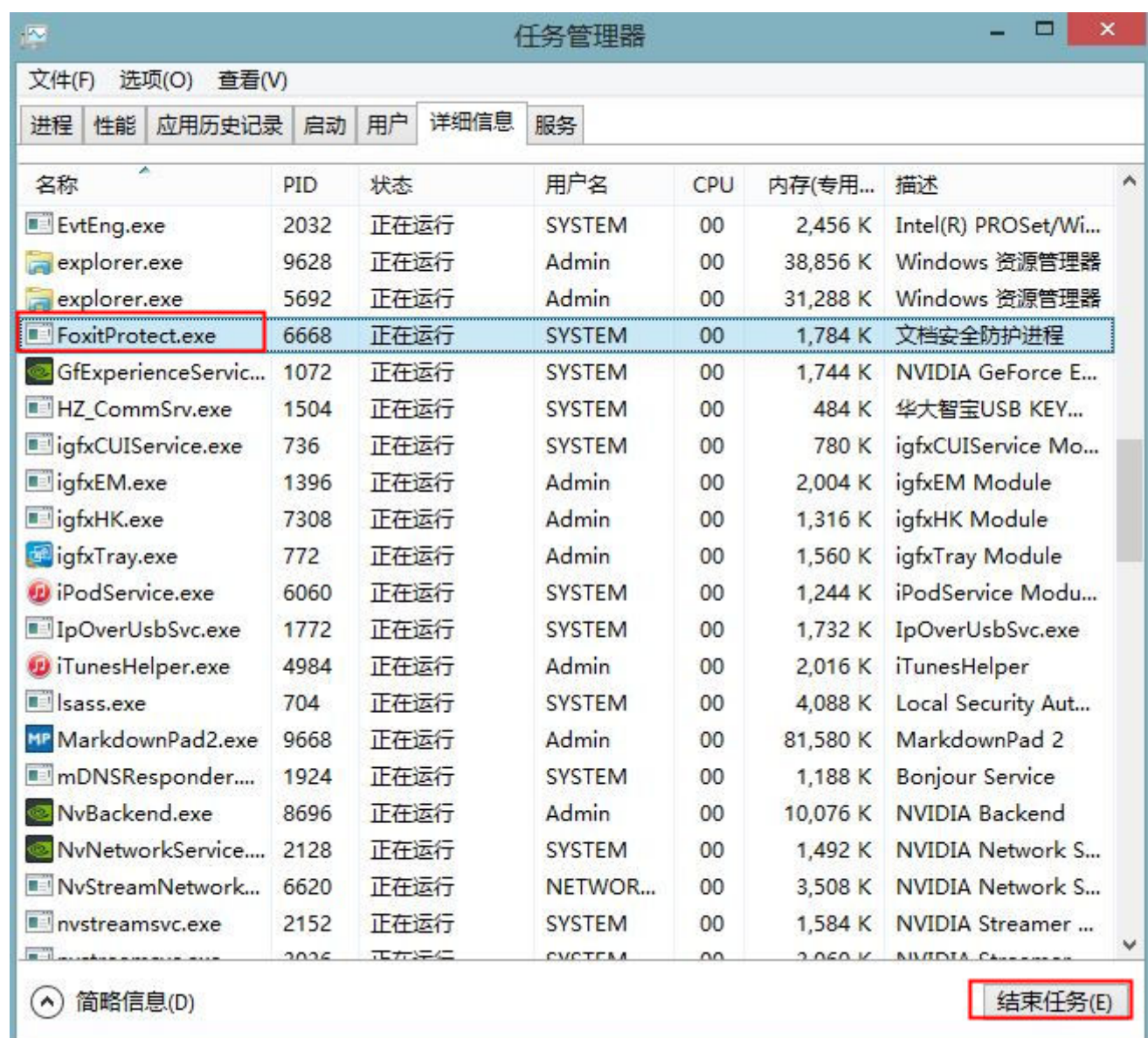
输入命令

```
$ tasklist /svc /FI "PID eq 6668"
```

该进程的名称就会显示出来:

映像名称	PID	服务
FoxitProtect.exe	6668	FxService

打开windows的任务管理器，结束它：



再次运行 `jeekyll serve` 就可以了。

如果一切顺利，通过在浏览器地址栏输入 `http://localhost:4000/` 回车就已经可以看到自己网站的样子啦。

只要 `jeekyll serve` 服务开着，你的本地仓库文件有任何更新，本地网站刷新都能马上看到，欧耶！

编写符合Jekyll规范的网站

如要自己修改fork下来的主题，可以参考以下资料：

使用Github Pages建独立博客 (<http://beiyuu.com/github-pages/>)

搭建一个免费的，无限流量的Blog——github Pages和Jekyll入门

(http://www.ruanyifeng.com/blog/2012/08/blogging_with_jekyll.html)

Jekyll中文文档 (<http://jekyll.bootcss.com/>)

进阶：添加评论、分享功能

常见的第三方评论系统有：Disqus (<https://disqus.com/>)，多说 (<http://dev.duoshuo.com/>)。

简单来说是在html文件中嵌入Javascript代码，注册网站后都有较好的指导，并不困难。

扩展阅读

[1]Github Pages (<https://pages.github.com/>)

[2]Git教程 - 廖雪峰

(<http://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000>)

[3]Jekyll中文文档 (<http://jekyll.bootcss.com/>)

[4]认识与入门Markdown (<http://sspai.com/2513>)

[5]Ruby和Gem (<http://blacktha.com/2015/07/06/tech/Ruby/>)

[6]使用Github Pages建独立博客 (<http://beiyuu.com/github-pages/>)

[7]搭建一个免费的，无限流量的Blog——github Pages和Jekyll入门

(http://www.ruanyifeng.com/blog/2012/08/blogging_with_jekyll.html)

分享到： 微信 微博 豆瓣

PREVIOUS

当我写博客时我写些什么 (/2016/03/23/THINK-TRY-WRITE/)

NEXT

一篇文章搞定JAVA开发环境搭建 (/2016/04/01/JAVA-ECLIPSE/)

19条评论

最新 最早 最热



Bebop.C (<http://www.douban.com/people/ChaosAwake/>)

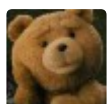
回复 mortred: 建好也可以发地址过来大家去帮忙加加人气

(<http://www.douban.com/people/ChaosAwake/>)6月18日

回复

顶

转发



mortred

回复 Bebop.C: 按照教程搭好了，Ruby China如果连不上，将https更改为http，这样可以

6月9日

回复

顶

转发



Bebop.C (<http://www.douban.com/people/ChaosAwake/>)

回复 mortred: 已更新源, 感谢

(<http://www.douban.com/people/ChaosAwake/>) 6月9日 回复 顶 转发



Bebop.C (<http://www.douban.com/people/ChaosAwake/>)

回复 mortred: 我这里Ruby China的源也一直连不上, 好像也得梯子。可能是间歇性失效, 过段时间再来看看。多谢啦~你的博客搭好了吗

(<http://www.douban.com/people/ChaosAwake/>) 6月9日 回复 顶 转发



mortred

回复 Bebop.C: 4. 安装jekyll-paginate 这一小节中, 替换国内镜像里, 淘宝的镜像源已经不可用了, 建议更新为Ruby China的源 <https://gems.ruby-china.org/>

6月8日 回复 顶 转发



Bebop.C (<http://www.douban.com/people/ChaosAwake/>)

回复 mortred: 谢谢亲的支持! 能对别人有帮助是最好的了!

(<http://www.douban.com/people/ChaosAwake/>) 6月7日 回复 顶 转发



mortred

非常赞的教程, 对新手很友好。感谢作者!

6月5日 回复 顶 转发



Bebop.C (<http://www.douban.com/people/ChaosAwake/>)

回复 qazsweet: 谢谢亲

(<http://www.douban.com/people/ChaosAwake/>) 6月2日 回复 顶 转发



qazsweet

好赞!!

6月2日 回复 顶 转发



Bebop.C (<http://www.douban.com/people/ChaosAwake/>)

回复 廖怡恒: 对于有经验的计算机使用者来说不必要, 但是这篇文章是写给对计算机不那么熟练又想搭建独立博客的朋友呢, 计算机基础好大可以看专业的教程, 谢谢。

(<http://www.douban.com/people/ChaosAwake/>) 6月1日 回复 顶 转发



廖怡恒 (<http://weibo.com/5198100016>)

写了一堆废话。大部分都是些不必要的。

(<http://weibo.com/5198100016>) 6月1日 回复 顶 转发



Bebop.C (<http://www.douban.com/people/ChaosAwake/>)

回复 cosa_79033: 是的, 也是用的开源模版, 你也可以按这个教程做一个

(<http://www.douban.com/people/ChaosAwake/>) 5月27日 回复 顶 转发

cosa_79033 (<http://weibo.com/cosa79033>)



这个是用jekyll做的么 😄

5月26日 回复 顶 转发

(<http://weibo.com/cosa79033>)



Bebop.C (<http://www.douban.com/people/ChaosAwake/>)

回复 啊: 🤔

(<http://www.douban.com/people/ChaosAwake/>) 5月9日 回复 顶 转发



Bebop.C (<http://www.douban.com/people/ChaosAwake/>)

回复 王小杰在路上: 多谢帮忙回复, 具体是少了一步吗, 我感觉是fork的仓库的CNAME写了跳转的域名

(<http://www.douban.com/people/ChaosAwake/>) 5月9日 回复 顶 转发



Bebop.C (<http://www.douban.com/people/ChaosAwake/>)

回复 游客: 应该是fork的仓库的CNAME写了跳转的域名, 把内容删掉看看

(<http://www.douban.com/people/ChaosAwake/>) 5月9日 回复 顶 转发



王小杰在路上 (<http://weibo.com/scalpelcn>)

回复 游客: 的确是少了一步, 需要发布一下就好了

(<http://weibo.com/scalpelcn>) 5月9日 回复 顶 转发



啊



5月7日 回复 顶 转发



游客

"此时你会发现已经可以通过 <http://{你的Github用户名}.github.io> 访问你fork下来的网站啦!", 不行啊, 显示404, 是不是漏了啥步骤?

5月6日 回复 顶 转发

社交帐号登录: 微信 微博 QQ 人人 [更多»](#)



说点什么吧...

发布

playingfingers.com正在使用多说 (<http://duoshuo.com>)

FEATURED TAGS (/tags/)

FRIENDS

Bebop.C Blog (<http://playingfingers.com>) Spike Spiegel Jet Black Faye Valentine Ein



(<https://github.com/chaosinmotion>)

Copyright © Bebop.C Blog 2016

Theme based on Hux (<https://github.com/Huxpro/huxpro.github.io>)