


1 建立GitHub Page站点

1.1 新建一个GitHub仓库

首先，我们假设你已经注册好了GitHub账号。然后登陆，进入GitHub主页，点击左上角的 New，新建一个仓库。




[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)


Your repository "WadeFrank/WadeFrank.github.io" was successfully deleted.


Repositories


New


Find a repository...


 [WadeFrank/vimplus](#)


 [WadeFrank/d2l-zh](#)

 [WadeFrank/handson-ml](#)

 [WadeFrank/ML-NLP](#)

 [WadeFrank/Kaggle](#)

 [WadeFrank/Interview](#)

 [WadeFrank/Java](#)

Show more

Working with a team?


GitHub is built for collaboration. Set up an organization to improve the way your team works together, and get access to more features.


Create an organization

Discover interesting project your personal news feed.

Your news feed helps you keep up with re
and people you [follow](#).

Explore GitHub

 **ProTip!** The feed shows you events from people you [fol](#)

 [Subscribe to your news feed](#)

新建一个仓库的时候需要我们输入仓库名字，默认格式是：用户名.github.io（用户名与前面Owner的名字要一致），输入名字后点击最下面的 Create repository 即可。

Create a new repository

A repository contains all project files, including the repository files that are not stored elsewhere? [Import a repository](#).

Owner *


WadeFrank ▾

Repository name *


WadeFrank.github.io

Great repository names are **WadeFrank.github.io is available**

Description (optional)

☒  **Public**

Anyone on the internet can see this repository. You

☐  **Private**

You choose who can see and commit to this repository

Initialize this repository with:

Skip this step if you're importing an existing repository

☐ **Add a README file**

This is where you can write a long description for your project

☐ **Add .gitignore**

Choose which files not to track from a list of templates. [Learn more](#)

☐ **Choose a license**

A license tells others what they can and can't do with your code

Create repository

然后我们新建好了一个自己的仓库，名字叫：WadeFrank.github.io，但是里面是空的什么都没有。接下来我们需要配置自己的电脑，使得我们能够用自己的电脑对仓库里面的内容进行修改。

1.2 配置SSH Keys

为了能连接上我们的仓库，我们需要将我们本机的SSH密钥加到GitHub账户中去。首先打开Mac终端，输入以下命令，进入到.ssh文件中

```
cd ~/.ssh
```

如果提示 No such file or directory 说明你是第一次使用 git。

接下来创建新的SSH keys，输入以下命令：

```
ssh-keygen -t rsa -C "此处填写你的邮箱地址"
```

然后会依次出现以下信息：

```
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/your_user_directory/.ssh/id_rsa):<回车>
Enter passphrase (empty for no passphrase):<输入加密串><如不想设置密码可直接回车表示为空>
Enter same passphrase again:<再次输入加密串><接着回车确认>
Your identification has been saved in /Users/你的名字/.ssh/id_rsa.
Your public key has been saved in /Users/你的名字/.ssh/id_rsa.pub.
The key fingerprint is:
43:c5:5b:5f:bl:f1:50:43:ad:20:a6:92:6a:1f:9a:3a "你的邮箱地址"
```

最后出现一幅类似长方形的字符画即表示成功。成功后我们将.ssh/id_rsa.pub文件中的密钥拷贝出来：

```
cd ~/.ssh
```

```
vim id_rsa.pub
```

然后进入github主页，点击右上角头像进入settings，选择SSH and GPG keys，再点击New SSH Key。将内容复制进key，title可以为空。最后Add SSH key。

最后可以通过在终端输入如下命令进行测试是否成功：

```
ssh -T git@GitHub.com
```

出现以下字符，则说明添加SSH key成功：

```
Hi WadeFrank! You've successfully authenticated, but GitHub does not provide shell access.
```

1.3 在仓库中添加README.md

我们配置好SSH Keys之后，我们就可以来测试GitHub Pages有没有创建成功了。

首先在终端输入以下命令，在用户目录下新建一个文件夹：

```
cd
mkdir MyBlog
cd MyBlog
```

然后输入以下命令，注意把用户名替换成你的真实用户名：

```
echo "# 用户名.github.io" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/用户名/用户名.github.io.git
git push -u origin main
```

如果没有问题的话，你就会发现仓库中多了一个README.md文件，然后在浏览器中输入：用户名.github.io。如果出现了以下界面（没有报错404），那么就说明GitHub Pages配置成功了。

WadeFrank.github.io

2 安装Hexo

2.1 安装 Git

其实Mac一般都自带git，我们只需要输入git --version来检查一下。

```
git --version
git version 2.28.0
```

2.2 安装 Node.js

第一步：先安装 nvm（node version manager）并确认是否成功，依次输入以下命令：

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh | bash
command -v nvm
```

会出现nvm，如果没出现，则重启终端再次输入第二条命令。

第二步：用 nvm 安装 node

```
nvm install node # "node" is an alias for the latest version
```

2.3 安装Hexo

使用npm一键安装Hexo博客程序：

```
npm install -g hexo-cli
```

初始化并安装部件：

```
hexo init blog # 初始化
cd blog
npm install # 安装组件
```

完成后依次输入下面命令，启动本地服务器进行预览：

```
hexo g # 生成页面
hexo s # 启动预览
```

访问 <http://localhost:4000>，出现 Hexo 默认页面，本地博客安装成功

Hexo 博客文件夹目录结构如下：

```
.
├── _config.yml # 网站的配置信息
├── package.json # 应用程序的信息
├── scaffolds # 模版文件夹
├── source # 存放用户资源，Markdown 文档
│   ├── _drafts
│   └── _posts
├── themes # 主题文件夹
└── public # 网站文件
```

3 部署 Hexo 到 GitHub Pages

本地博客测试成功后，就是上传到 GitHub 进行部署，使其能够在网络上访问。

首先安装 hexo-deployer-git:

```
npm install hexo-deployer-git --save
```

然后修改 `_config.yml` 文件末尾的 Deployment 部分，修改成如下：

```
deploy:
  type: git
  repository: git@github.com:用户名/用户名.github.io.git
  branch: master
```

完成后运行 `hexo d` 将网站上传部署到 GitHub Pages。

完成！这时访问我们的 GitHub 域名 `https://用户名.github.io` 就可以看到 Hexo 网站了。

参考：

https://blog.csdn.net/weixin_41160054/article/details/89531921

<https://www.zhihu.com/search?type=content&q=GitHub%E5%8A%A0hexo>