

# 前言

本文已经收录至<https://gitee.com/dywangk/SQL-study>gitee仓库，欢迎收藏。

你真的了解git、gitee、github吗？本文主要围绕git、gitee、github的使用展开介绍，如何一步步通过实践去打造自己的git工作环境。当然你可以作为写作环境，熟悉一门技术的有效方式就是应用实践。顺带会讲一些gitee和github的pages服务，搭建自己的个人网站。

本文主要环境是在Windows下进行的，所以看到桌面或者在某一目录右键可以直接打开Git Bash。

# 正文

至于我为什么将git工具放在最前面，是因为通过这个工具可以将gitee与github串联起来。可能有人会说，我可以在虚拟机上或者云服务器上搭建git环境测试。但我想说的是，白嫖她不香吗？直呼真香，小伙子有点东西啊！

## 一、git

### 背景

现如今，难以想象有创意的人会在没有备份策略的情况启动一个项目。数据是短暂的，且容易丢失，例如通过一次错误的代码变更或者一次灾难性的磁盘崩溃。所以，在整个工作中持续性地备份和存档是非常明智的。

对于文本和代码项目，备份策略通常包括版本控制，或者叫“对变更进行追踪管理”。每个开发人员都会进行若干变更。这些持续增长的变更，加在一起可以构成一个版本库，用于项目描述，团队沟通和产品管理。版本控制有着举足轻重的作用。

一个可以管理和追踪软件代码或其他类似类容的不同版本工具。通常称为：版本控制系统（VCS），或者源代码管理器（SCM），或者修订控制系统（RCS）。或者其他各种“修订、代码、内容、版本、控制、管理以及系统”等相关叫法。

但其实每二个工具都是出于同样的目的，开发以及维护开发出来的代码，方便管理控制历史记录和修改。

### 诞生

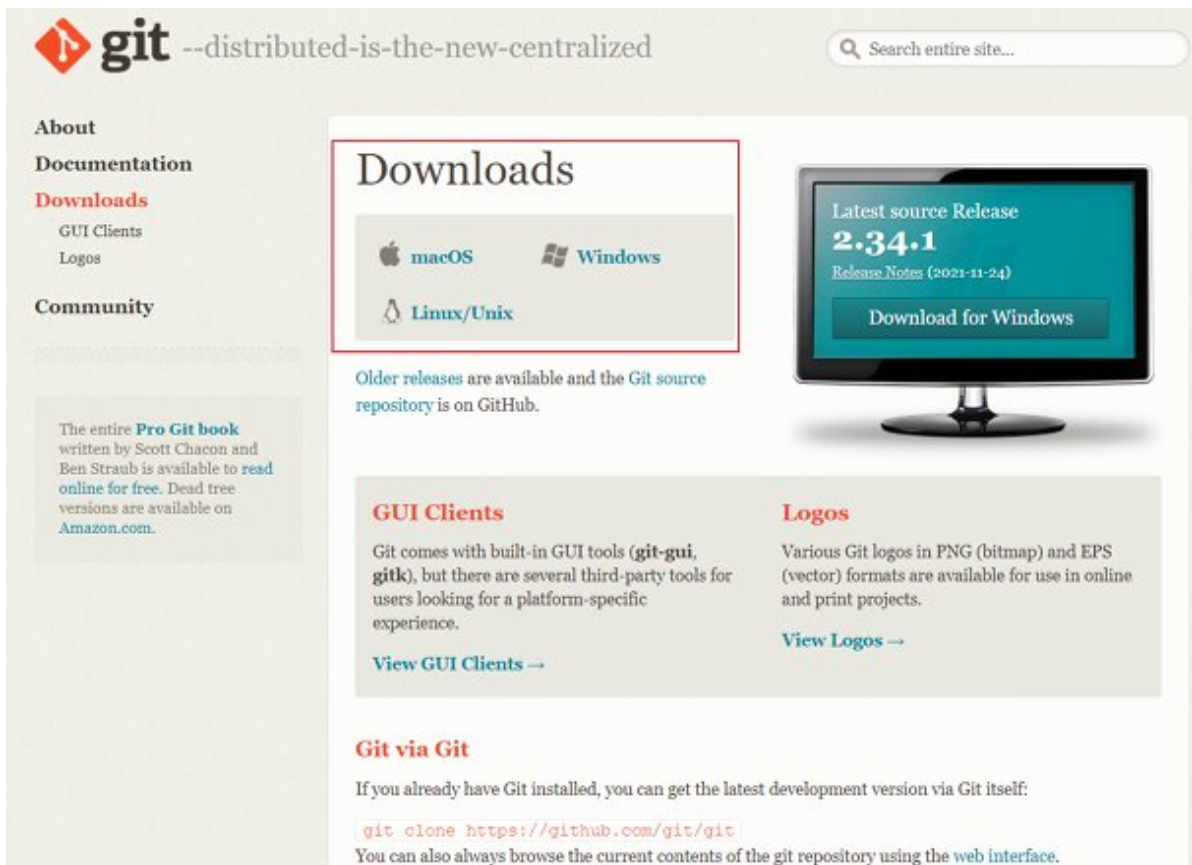
一句话简单概括

BitKeeper的所有方对免费版做了限制，Linux本人开始寻找替代品，在工作中开发出了有助于分布式开发的git版本控制管理工具。

## 1、git客户端工具下载地址

<https://git-scm.com/downloads>

git下载地址：支持Linux/Unix、macOS、Windows



## 2、生成ssh公钥

使用 `git status` 时遇到如下问题：

```
You don't exist Go away!  
you parents must have hated you!  
you sysadmin must hate you!
```

不要慌，此时git在告诉你，我无法确定您的真实姓名，咱交个好朋友吧，请告诉我你的芳名。

通过设置name和email地址可以修复此问题。

### 2.1、配置提交作者

安装好了git工具，此时可以在桌面右键（Git Bash），输入以下命令进行设置用户以及email。

```
$ git config --global user.name "dywangk"  
$ git config --global user.email "dywangk@example.com"
```

### 2.2、生成ssh公钥，Windows下默认在系统盘的当前用的.ssh目录下

```
$ ssh -keygen -t ed25519 -C "test@example.com"
```

### 2.3、gitee提供的方案配置多个key

生成rsa文件时，以指定不同的文件命令作为区分gitee以及github生成的key文件。

```
$ ssh-keygen -t rsa -C 'dywangk@company.com' -f ~/.ssh/gitee_id_rsa
$ ssh-keygen -t rsa -C 'dywangk@163.com' -f ~/.ssh/github_id_rsa
```

生成key后，然后在 ~/.ssh 目录下新建一个config文件，添加如下内容（其中Host和HostName填写git服务器的域名，IdentityFile指定私钥的路径）。看完后我大致明白了，是将gitee和github区分开来，分别读取不同的id\_rsa文件的参数值。

```
# gitee
Host gitee.com
HostName gitee.com
PreferredAuthentications publickey
IdentityFile ~/.ssh/gitee_id_rsa
# github
Host github.com
HostName github.com
PreferredAuthentications publickey
IdentityFile ~/.ssh/github_id_rsa
```

配置好进行验证

```
#验证gitee
$ ssh -T git@gitee.com
#验证github
$ ssh -T git@github.com
```

配置好git之后，利用ssh-keygen命令生成ssh公钥，生成公钥所在的存储目录（以Windows环境为例）。生成完key后，可以通过VSCode、sublime text或者其他的辅助工具打开这个文件，复制到github的ssh公钥并保存。在下面的github介绍过程中，我有提到如何将生成的key保存到github并验证。同理gitee也是一样的操作过程，只是操作界面不一样而已。



## 二、gitee

引用官方的一个介绍

Gitee.com(码云) 是 OSCHINA.NET 推出的代码托管平台,支持 Git 和 SVN,提供免费的私有仓库托管。目前已有超过 600 万的开发者选择 Gitee。

中文模式，使用起来比较方便。没有github那么难上手，毕竟看到纯英文，读起来着实头疼。在gitee的教程上，我没有过多介绍。

提一点，如果不想使用命令提交，gitee也支持拖拉拽形式上传文件，但是每天上传的文件有限制。

## 1、gitee配置ssh公钥教程

gitee官网有很详细的中文说明文档，至于如何配置，我这里给出链接参考。

<https://gitee.com/help/articles/4181#article-header0>

## 2、配置GPG keys工具

当你访问gitee配置的GPG keys的时候，截至目前为止依旧是测试中（beta）。

<https://gpg4win.org/thanks-for-download.html>

## 三、github

### 1、github官网

<https://github.com>

一句话精简概括，一款活跃在互联网中有着庞大的开发团队或者个人并维护者众多的开源项目的开源项目托管平台。

下面是来自于维基百科最新的介绍

GitHub是通过Git进行版本控制的软件源代码托管服务平台，由GitHub公司（曾称Logical Awesome）的开发者Chris Wanstrath、P.J. Hyett和汤姆·普雷斯顿·沃纳使用Ruby on Rails编写而成。

GitHub同时提供付费账户和免费账户。这两种账户都可以创建公开或私有的代码仓库，但付费用户拥有更多功能。根据在2009年的Git用户调查，GitHub是最流行的Git访问站点。除了允许个人和组织创建和访问保管中的代码以外，它也提供了一些方便社会化共同软件开发的功能，即一般人口中的社群功能，包括允许用户追踪其他用户、组织、软件库的动态，对软件代码的改动和bug提出评论等。GitHub也提供了图表功能，用于概观显示开发者们怎样在代码库上工作以及软件的开发活跃程度。

截止到2020年1月，GitHub已经有超过4000万注册用户和1.9亿代码库（包括至少2800万开源代码库），事实上已经成为了世界上最大的代码存放网站和开源社区。

2018年6月4日晚，美国科技公司微软宣布以75亿美元的股票收购GitHub。

就算被微软收购后，依旧没改变github那极不稳定的访问速度。

### 2、github使用配置简介

由于github是纯英文的，所以我会多配置一些截图并辅以说明，望理解。

#### 2.1、github配置ssh公钥

这里不仅仅是支持配置SSH keys，并且还支持配置GPG keys，需要下载支持的工具进行生成GPG公钥然后添加到github。当然，如果你要移除某一个key，点击Delete删除即可。

## SSH keys

[New SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.



**github**  
SHA256 [redacted]  
Never used — Read/write

配合完后，也可以移除

Delete

Check out our guide to [generating SSH keys](#) or [troubleshoot common SSH problems](#).

## GPG keys

配置GPG keys需要下载  
工具生成

[New GPG key](#)

There are no GPG keys associated with your account.

Learn how to [generate a GPG key](#) and add it to your account .

<https://github.com/settings/ssh/new>

## 2.2、账号设置

账号设置，可以修改用户名、以及删除你的账号（谨慎操作）。

Account

Appearance

Accessibility

Account security

Billing & plans

Security log

Security & analysis

Sponsorship log

Emails

Notifications

Scheduled reminders

SSH and GPG keys

Repositories

Packages

Pages

Organizations

Saved replies

Applications

Developer settings

Moderation settings

Change username

修改用户名

Looking to manage account security settings? You can find them in the [Account security](#) tab.

Export account data

Export all repositories and profile metadata for @ [redacted]. Exports will be available for 7 days.

Start export

Successor settings

By clicking "Add Successor" below, I acknowledge that I am the owner of the @onwangk account, and am authorizing GitHub to transfer content within that account to my GitHub Successor, designated below, in the event of my death. I understand that this appointment of a successor does not override legally binding next-of-kin rules or estate laws of any relevant jurisdiction, and does not create a binding will. [Learn more about account successors.](#)

Search by username, full name, or email address

Add Successor

You have not designated a successor.

Delete account

删除账号，谨慎操作，数据无价

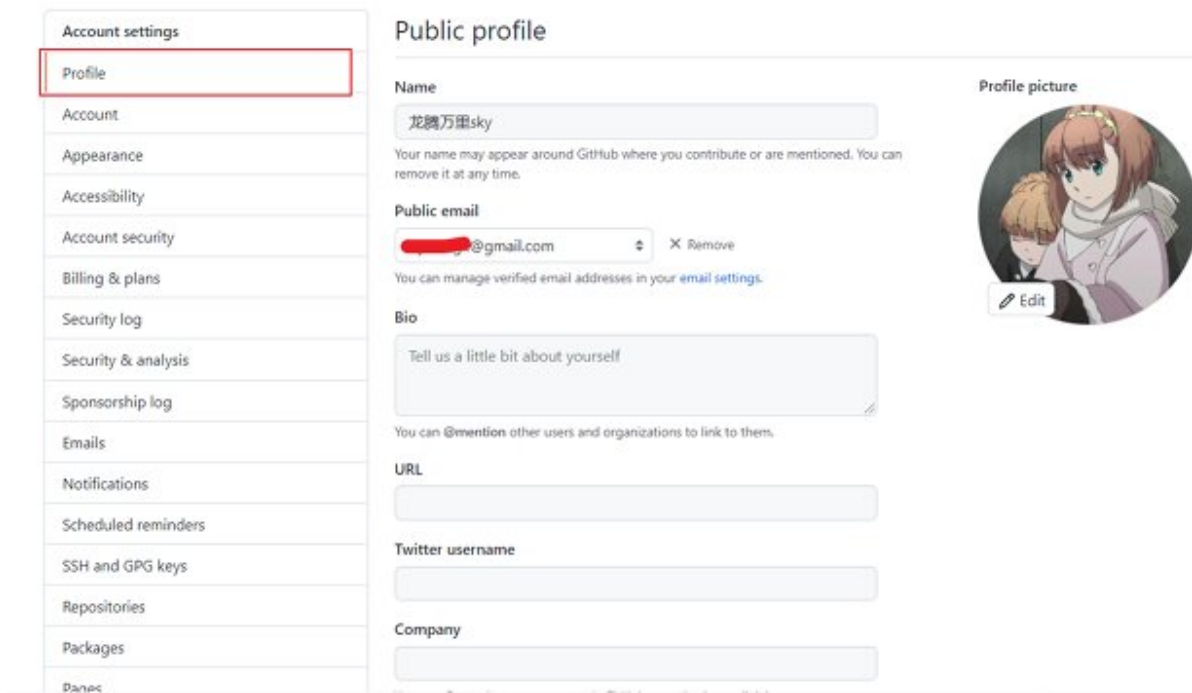
Once you delete your account, there is no going back. Please be certain.

Delete your account

<https://github.com/settings/admin>

## 2.3、个人简介设置

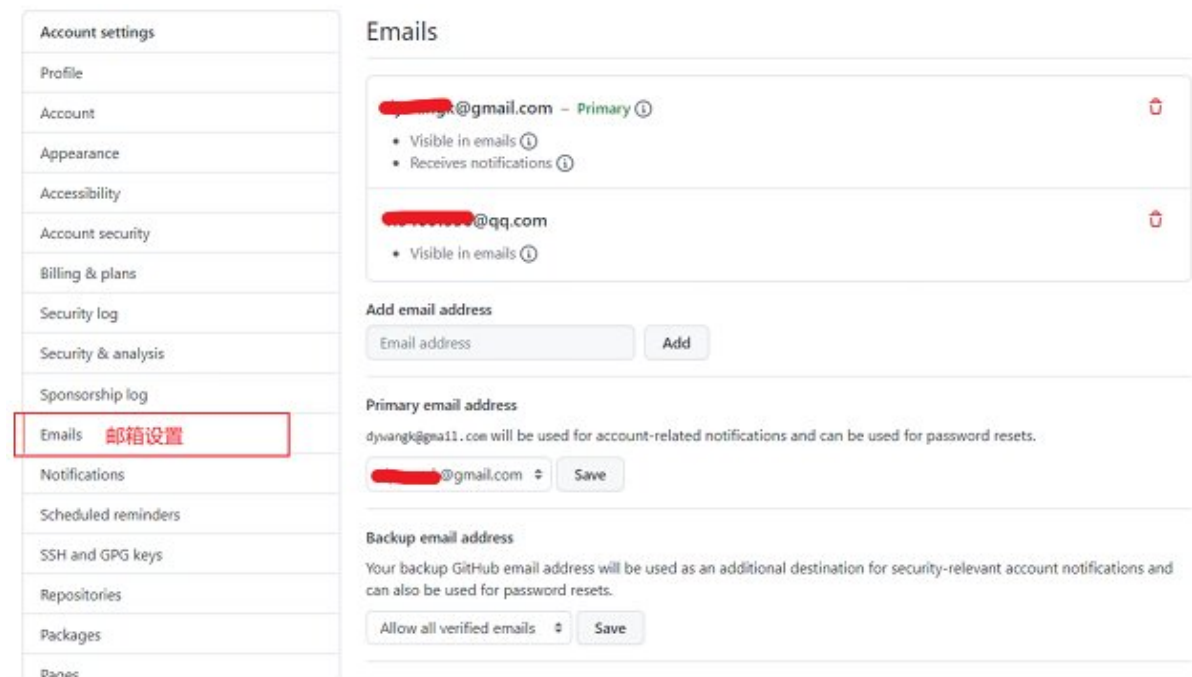
此处是个人简介设置，比如公共邮箱设置、对自己的介绍、URL地址、Twitter用户名以及公司等等介绍。



<https://github.com/settings/profile>

## 2.4、辅助邮箱设置

设置邮箱后，可以使用设置好的邮箱进行登录操作，接收一些比如修改密码时可能需要邮箱验证。



<https://github.com/settings/emails>

## 3、配置验证

Windows下右键打开 `Git Bash`，使用 `ssh -T` 命令测试验证。当前配置了gitee的ssh公钥，验证返回结果成功。未配置github的ssh公钥，测试验证返回的结果是权限（permission denied）拒绝。

使用命令测试验证



```
#验证gitee
ssh -T git@gitee.com
#验证github
ssh -T git@github.com
```

```
admin@wangk MINGW64 ~/Desktop
$ ssh -T git@gitee.com
Hi 龙腾万里sky! You've successfully authenticated, but GITEE.COM does not provide shell access. 配置了gitee的公钥ssh, 返回结果是成功的。

admin@wangk MINGW64 ~/Desktop
$ ssh -T git@github.com
git@github.com: Permission denied (publickey).
github的公钥ssh被我移除了, 测试返回权限拒绝。

admin@wangk MINGW64 ~/Desktop
$ |
```

下面是配置好了github的ssh公钥或者是GPG keys, 进行测试验证, 返回结果是成功。

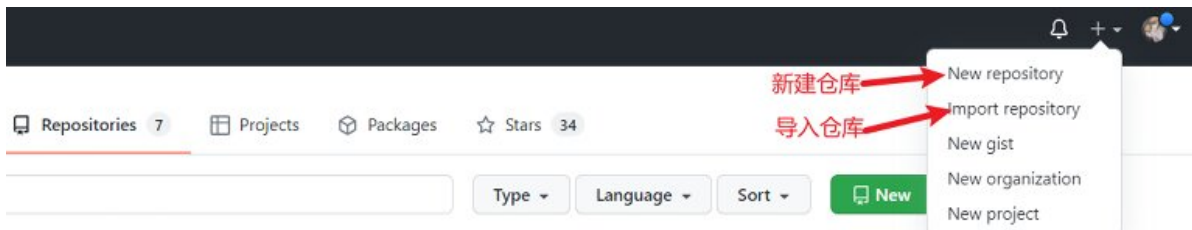
```
admin@wangk MINGW64 ~/Desktop
$ ssh -T git@github.com
git@github.com: Permission denied (publickey).

admin@wangk MINGW64 ~/Desktop
$ ssh -T git@github.com
Hi cnwangk! You've successfully authenticated, but GitHub does not provide shell access. 在github配置好ssh公钥, 测试访问验证返回成功。

admin@wangk MINGW64 ~/Desktop
```

## 4、导入远程仓库到github

导入远程仓库选择**import repositories**, 比如导入gitee的远程仓库。好吧, 我一般是将github仓库同步至gitee作为镜像仓库使用。大家都懂得, 长城网络的墙太殷实了, 就算github被微软收购后这个访问速度依旧感人。



例如, 我同步自己的写作工作空间:

## Import your project to GitHub

Import all the files, including the revision history, from another version control system.

Your old repository's clone URL

填写你的远程仓库地址

Learn more about the types of [supported VCS](#).

Your new repository details

Owner \*

Repository Name \*

Privacy

☒

Public

Anyone on the internet can see this repository. You choose who can commit.

☐

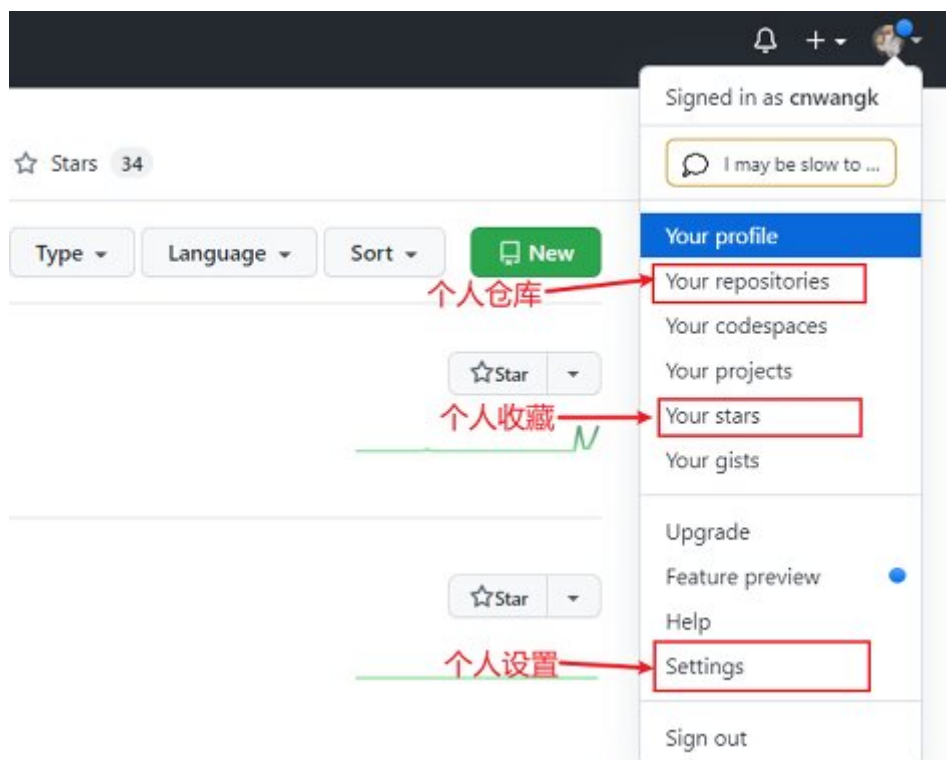
Private

You choose who can see and commit to this repository.

[Cancel](#)

[Begin import](#)

github一些使用介绍，你的个人仓库（your repositories）、个人收藏的仓库（your stars）、个人设置（settings）



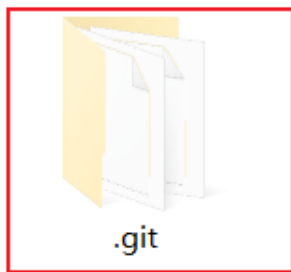
## 四、git基本命令介绍（实用型）

### 1、基本命令

#### 1.1、git init初始化一个仓库

git初始化一个仓库时，会生成.git目录，是一个隐藏文件。linux下可以使用 `ls -a` 查看所有文件，Windows下打开资源管理器的显示隐藏项目才能看到该文件。下面给出Windows下的初始化后的git仓库示例：





隐藏目录文件列表如下，具体作用不做讲解。有兴趣的可以去逛逛官方文档，或者参考git相关的实体书。

- 📁 hooks
- 📁 info
- 📁 logs
- 📁 objects
- 📁 refs
- 📄 COMMIT\_EDITMSG
- 📄 config
- 📄 description
- 📄 HEAD
- 📄 index

Windows下直接切换到test目录，并初始化test这个目录

```
#windows下直接切换到test目录，并初始化这个目录  
$ git init
```

```
admin@wangk MINGW64 /d/work/test  
$ git init  
Initialized empty Git repository in D:/work/test/.git/
```

## 1.2、git add将数据暂存

这里提醒一下，TAB快捷键可以自动补全，使用时输入开头直接按TAB快捷键自动补全。

```
#添加README.md文件，此时加入到暂存区域  
$ git add README.md
```

## 1.3、git commit提交数据

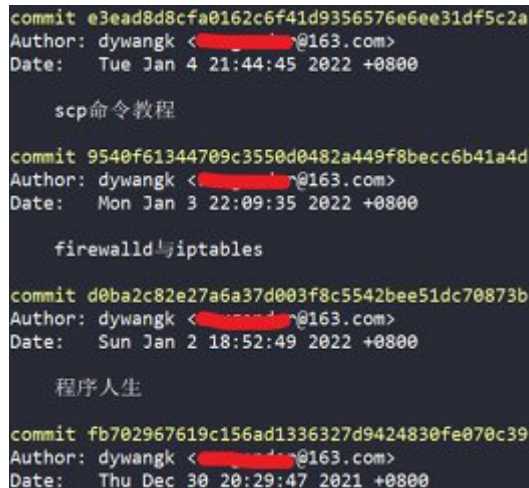
提交创建好的README.md文件，显示如下图所示。这里只是示例，不代表最终推送到仓库。

```
admin@wangk MINGW64 /d/work/test (master)  
$ git commit -m "README.md"  
[master (root-commit) bd156ab] README.md  
1 file changed, 1 insertion(+)  
create mode 100644 README.md
```

```
#提交数据到仓库,提交全部
$ git commit -m "新增README.md文件"
#提交单个指定文件README.md
$ git commit README.md -m "新增README.md文件"
```

## 查看提交

```
$ git log
```



```
commit e3ead8d8cfa0162c6f41d9356576e6ee31df5c2a
Author: dywangk <[redacted]@163.com>
Date: Tue Jan 4 21:44:45 2022 +0800

    scp命令教程

commit 9540f61344709c3550d0482a449f8becc6b41a4d
Author: dywangk <[redacted]@163.com>
Date: Mon Jan 3 22:09:35 2022 +0800

    firewallld与iptables

commit d0ba2c82e27a6a37d003f8c5542bee51dc70873b
Author: dywangk <[redacted]@163.com>
Date: Sun Jan 2 18:52:49 2022 +0800

    程序人生

commit fb702967619c156ad1336327d9424830fe070c39
Author: dywangk <[redacted]@163.com>
Date: Thu Dec 30 20:29:47 2021 +0800
```

## 1.4、git push推送提交的数据到远程仓库

推送之前需要连上自己的远程仓库SQL-study

```
$ git remote add origin git@gitee.com:dywangk/SQL-study.git
```

### 推送提交的数据到远程仓库master

```
#推送提交的数据到远程仓库master
$ git push git@gitee.com:dywangk/SQL-study.git
```

之前看到某呼上有提问，如何记住那么多的linux命令。我当时就在思考，经常使用，自然而然的就记住了。一遍记不住，我操作五遍该有印象了吧。很多命令不用刻意记住，善用系统提供的帮助命令 `--help` 或者linux下的 `man` 辅助你快速上手。**使用时尽量手敲，不要去复制粘贴**，否则效果不大。熟悉后可以使用上下键快速调出历史命令，使用 `history` 命令查看使用的记录。

以上就是我日常书写文章后，进行提交经常使用到的git命令。当你需要优化删除一些不需要的文件，此时可以通过 `git rm` 命令实现操作。当你需要移动某个文件时，可以使用 `git rm` 或者 `git mv` 命令进行移动文件操作。熟悉linux基本命令的同学，更容易上手，很多命令都类似。

## 1.5、克隆仓库到本地

比如，将阿里的开源消息中间件rocketmq-spring克隆到本地，**前提是配置好了ssh公钥**

```
#将阿里的开源消息中间件rocketmq-spring克隆到本地，前提是配置好了ssh公钥
$ git clone git@github.com:apache/rocketmq-spring.git
```

## 2、进阶

### 2.1、以分支为例简单介绍

```
#查看分支
$ git branch
#新增分支
$ git branch bchtest
#删除分支
$ git branch -d bchtest
#检出分支
$ git checkout master
#合并分支命令
$ git merge other_branch
```

关于进阶知识，需要了解更多的git命令使用，比如分支、合并、删除、修改或者移动以及冲突处理。

**在你做删除操作时，切记事先做好备份。**

如果想再进一步，那就推荐系统的看实体书《Git版本控制管理》，我最近也在阅读这本书籍。很多以前一知半解的，现在突然**念头通达**了。深入了解Git的工作原理，辅助你更好地学习与理解。

## 五、Gitee Pages与Github Pages

关于gitee pages与github pages只做简单的介绍。至于新建仓库这些比较简单了，就不一一截图介绍了。要不然显得篇幅很长，有点在灌水的感觉。

### 1、gitee pages服务

<https://gitee.com/dywangk/sky/pages>

gitee的gitee pages，新建仓库，在服务这一栏会有显示。强调的是，开启gitee pages服务需要实名认证。



gitee pages配置页面，这已经是配置好的界面，可以强制使用https安全协议。当你提交数据到仓库时，没有及时生效，就进入此页面进行更新操作，然后测试访问你的git pages仓库即可。

比如访问我之前上传比较的感受一下。曾经的虾米音乐，我只是保留了自己的歌单部分：

<https://dywangk.gitee.io/sky/%E5%BD%B1%E8%A7%86%E7%A5%9E%E6%9B%B2.html>

Pages 服务仅供博客 / 门户 / 开源项目网站 / 开源项目静态效果演示用途, 请勿用于违规内容, 包括但不限于:

- 发布诱导分享/诱导关注/诱导下载/诱导跳转内容
- 发布欺诈/谣言/骚扰信息/广告信息/垃圾信息/特殊识别码、口令类信息
- 发布低俗内容/“宗教性捐献”及相关信息
- 发布侵害他人权利/违法经营及可疑服务类内容
- 发布其它违反国家法律法规的内容

平台保留 Gitee Pages 及 Gitee Pages Pro 使用规则最终解释权, 如发现上述违规行为, 平台将视违规情况严重程度予以 Pages 服务封禁以至账号永久封禁惩罚。

开启 Pages 后会在部署公钥中添加 pages 服务器的公钥

已开启 Gitee Pages 服务, 网站地址: <https://dywangk.gitee.io/sky>

部署分支

master

选择您要部署的分支

部署目录

如: 不填 (整个仓库) 或 src 或 src/jekyll

填写您要部署的分支上的目录

☒ 强制使用 HTTPS

更新

暂停

gitee官网有很详细的入门介绍, 我这里只提一下, 感兴趣的可以测试使用。

<https://gitee.com/help/articles/4136#article-header0>

## 2、github pages服务

github pages的配置页面

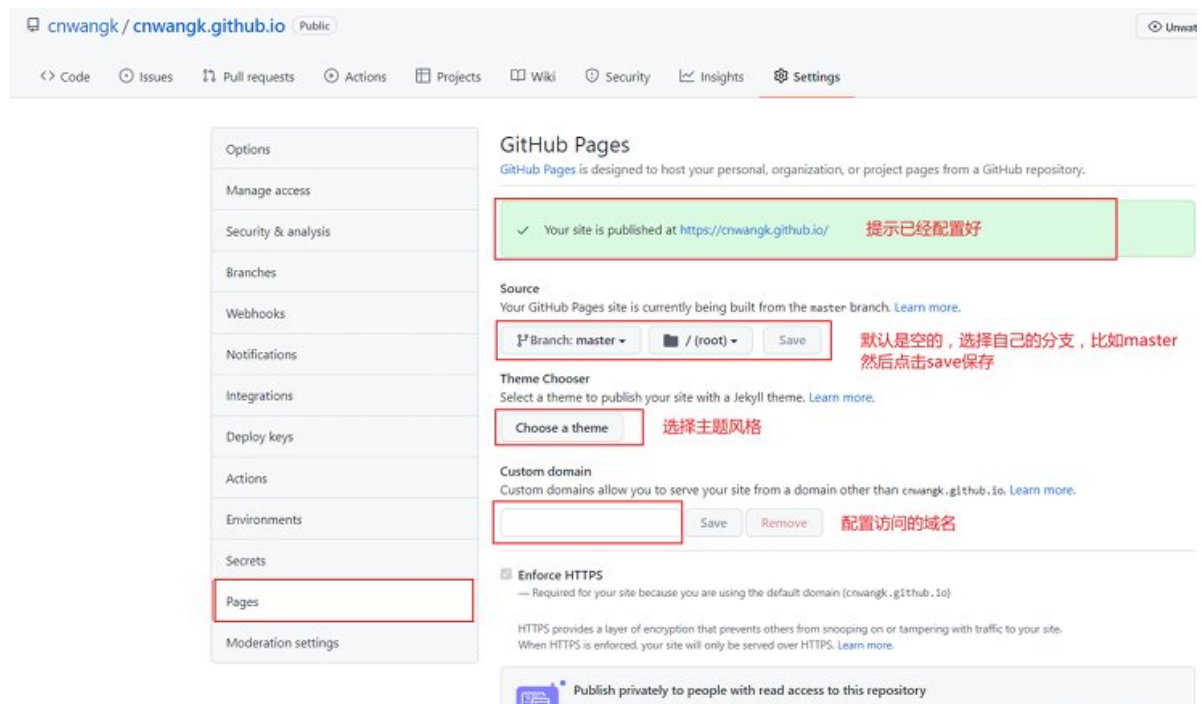
<https://github.com/cnwangk/test/settings/pages>

配置教程, 纯英文的, 可以用Google翻译一下哈

<https://pages.github.com/>

我测试配置了一个仓库

注意: 仓库必须是公开的 ( public )、然后仓库命令可以命令为用户名加github.io



<https://cnwangk.github.io/>

by 龙腾万里sky 转载请标明出处！