layout: post
title: "Git 使用"
date: 2020-05-21
description: "linux使用"

tag: linux使用

最近研究一下git使用，想要用最近三天的时间，来学一下git的基本应用和基本操作。毕竟会使用git与github是程序员必备的技能

系列学习的主要知识，参考廖雪峰老师的git系列教程

Git的安装

使用linux（我主要使用的是ubuntu系统）

在bash中输入git 观察是否安装git，如果显示没有安装

`sudo apt-get install git`

利用上述命令，可以安装成功git，（一般linux系统上自带都有git，如果没有自己安装）

在安装完成之后，利用下边的命令进行配置

```
git config --global user.name "Your Name"
git config --global user.email "email@example.com"
```

Git版本库的创建

版本库建立（repository）

```
mkdir learngit  #建立一个文件夹作为目录使用
cd learngit
pwd  #显示当前目录,
然后通过git init命令将建立的目录变为git的仓库。
```

```
git init #将文件添加到版本库
```

```
touch readme.txt  #新建空白文件
vim readme.txt
```

############在文件中输入

Git is a version control system.

Git is free software.

############

将文件加入仓库中

```
git add readme.txt
```

```
#### 然后利用git commit
git commit -m "wrote a readme file"
#### -m后边的引号中，尽量输入有用的描述，这个操作的意义
```

版本回退

主要的命令

```
git log   ###查看历史记录，输出的主要信息包含  commit id，作者，日期，操作信息，即前文说的-m后的信息

git log --pretty=oneline   ###简洁显示

在git中，HEAD表示当前版本，HEAD^表示上一个版本，HEAD^^表示上上个版本，HEAD~100表示往前100个版本。
git reset --hard HEAD^  ##回到前上上个版本

利用commit id来找到指定的版本
git reset --hard commit_id

记录自己每一次命令的操作git reflog
git reflog


额外的命令cat 直接打印出文本文档的信息
cat readme.txt
```

暂存区与工作区

git中存在暂存区与工作区的概念，上一篇中提到往git的仓库repository中添加文件需要两步，add与commit，利用add命令就是将文件存入暂存区，然后利用commit将文件存入工作区。可以利用git status 查看状态,暂存区是否有文件需要commit

```
git status
```

管理修改

git跟踪并管理的是修改，不是文件，因而在修改后需要将修改利用add加入暂存区，然后提交到工作区

撤销修改

命令git checkout -- readme.txt意思就是，把readme.txt文件在工作区的修改全部撤销，这里有两种情况：

一种是readme.txt自修改后还没有被放到暂存区，现在，撤销修改就回到和版本库一模一样的状态；

一种是readme.txt已经添加到暂存区后，又作了修改，现在，撤销修改就回到添加到暂存区后的状态。

总之，就是让这个文件回到最近一次git commit或git add时的状态。

删除文件

删除文件也是修改操作，一般删除直接在文件目录中删除,然后git status会显示那些文件被删除

```
rm filename
git status
```

此时有两种情况，一种是确实是想要删除，就用git rm删除，然后git commit

```
git rm filename
git commit -m "message"
```

另一种情况就是误删除，可以恢复

```
git checkout -- filename
```

Github使用

使用前提是注册github账号，本地的git仓库与github之间联系是通过github加速的，需要进行几个步骤的设置。

1、创建SSH Key。在用户主目录下，看看有没有.ssh目录，（这是个隐藏文件夹，利用 `ls -ah` 显示 ）

如果有，再看看这个目录下有没有id_rsa和id_rsa.pub这两个文件，如果已经有了，可直接跳到下一步。如果没有，打开Shell（Windows下打开Git Bash），创建SSH Key：

```
ssh-keygen -t rsa -C "youremail@example.com"
```

然后一路回车使用默认值即可

如果一切顺利的话，可以在用户主目录里找到.ssh目录，里面有id_rsa和id_rsa.pub两个文件，这两个就是SSH Key的秘钥对，id_rsa是私钥，不能泄露出去，id_rsa.pub是公钥，可以放心地告诉任何人。

2、登陆GitHub，打开“Account settings”，“SSH Keys”页面：

然后，点“Add SSH Key”，填上任意Title，在Key文本框里粘贴id_rsa.pub文件的内容

为什么GitHub需要SSH Key呢？因为GitHub需要识别出你推送的提交确实是你推送的，而不是别人冒充的，而Git支持SSH协议，所以，GitHub只要知道了你的公钥，就可以确认只有你自己才能推送。

当然，GitHub允许你添加多个Key。假定你有若干电脑，你一会儿在公司提交，一会儿在家里提交，只要把每台电脑的Key都添加到GitHub，就可以在每台电脑上往GitHub推送了。

确保自己拥有远程github仓库后，就可以进行远程仓库学习了

建立远程库

1、在github上建立仓库，点击github上右上角的“+”，然后new repository，在title上写上自己所建立的仓库名字，建立名字为learngit，然后点击创建

GitHub告诉我们，可以从这个仓库克隆出新的仓库，也可以把一个已有的本地仓库与之关联，然后，把本地仓库的内容推送到GitHub仓库。

现在，我们根据GitHub的提示，在本地的learngit仓库下运行命令（命令github页面上有显示）：

```
git remote add origin git@github.com:lxztju/learngit.git
```

添加后，远程库的名字就是origin，这是Git默认的叫法，也可以改成别的,但是origin这个名字一看就知道是远程库

```
git push -u origin master
```

第一次推送到远程时，加入了-u参数，以后往远程库推送文件时，可以直接

```
git push origin master
```

从remote repository clone代码

现在，假设我们从零开发，那么最好的方式是先创建远程库，然后，从远程库clone

1、首先在github上创建一个新的repository，名字随便设例如gitskill

我们勾选Initialize this repository with a README，这样GitHub会自动为我们创建一个README.md文件。创建完毕后，可以看到README.md文件

2、现在远程库已经创建好了，可以从远程clone仓库了

git clone git@github.com:lxztju/gitskill.git
然后进入目录可以看到仓库已经clone，包含一个文件readme.md

Github 版本回退控制

```
git checkout
git reset
git revert
```