Git

2019年7月15日

17:30

参考网站：<https://blog.csdn.net/buknow/article/details/80325986>

1.先在github上创建自己的项目

2.在本地创建一个项目，cd后初始化：git init

3.创建ssh key：ssh-keygen -t rsa -b 4096 -C "your\_email@example.com"

4.在终端复制ssh key，在.ssh/id\_rsa.pub(在本地电脑上，不是你创建的项目的地方）

5.登录github在个人主页点击ssh and GPG keys点击添加

6.检测是否成功绑定：ssh -T git@github.com

7.设置名字和邮箱：git config --global user.name "your name" /git config --global user.email "your\_email@example.com"

8.克隆：先cd到文件下然后git clone 创建成功之后github上的网址

9.查看当前版本库状态：git status

10.将文件加入缓冲区：git add filename/.(所有文件）

11.记录此次提交并上传： git commit -m 'zhushi'/git push origin master（master就是远端分支名，如果想上传其他分支则换成其他分支的名字）

新的branch文件提交，1>切换回master：git checkout master 2>合并branch：git merge branch\_name

3>加入缓冲区：git add . 4>提交：git commit -m "message" 5>上传 git push origin master

提交步骤：1.文件修改2.加入缓冲区3.提交并写修改信息commit4.push到远端

新的branch

12.创建分支：git checkout -b feature

13.在feature上随意修改，然后提交 git commit -m "some message"

14.此时提交代码，先切回主分支：git checkout master 然后git pull

15.删除自己建立的分支：git branch -d feature无法恢复(-D强制删除）

16.查看当前分支的清单：git branch

17.查看已经（或尚未）与当前分支合并的分支：git branch --merged/--no--merged

18.分支切换 git checkout branch\_name

19.分支合并：git checkout master后git merge branch\_name

20.查看每一个分支最后一次提交：git branch -v

高级操作

1.查看版本历史：cd到项目文件夹下，git log

2.回到某一历史版本：git checkout commitId

3.回到最新代码： git checkout master

zhaizhicheng 于 2019/7/23 9:45 修改

git的图形化工具 tortoisegit的学习

branch分支的学习

1.分支的创建： git checkout -b branch\_name(相当于 git branch branch\_name;git checkout branch\_name)

2.分支的切换：git checkout branch\_name;

3.分支的合并：git checkout master;git merge branch\_name;（分为两种，1>master作为branch\_name的祖先，直接将branch\_name作为最新的。

2>master不为branch\_name的祖先，git自行选择哪一次提交作为最优的共同祖先，然后进行合并，如果合并过程中出现冲突/矛盾（master和branch\_name都对某文件的某部分进行了修改）则将由用户决定选择哪一个。

4.合并完成后，删除分支 git branch -d branch\_name

学习网站：<https://git-scm.com/book/zh/v2/Git-%E5%88%86%E6%94%AF-%E5%88%86%E6%94%AF%E7%9A%84%E6%96%B0%E5%BB%BA%E4%B8%8E%E5%90%88%E5%B9%B6>

比较git clone、git pull、git fetch的区别

1.git clone就是最开始的时候本地什么都没有，克隆远端的代码到本地

例：git clone <https://github.com/tensorflow/tensorflow.git> 或 git clone git@github.com:tensorflow/tensorflow.git

2.git fetch是本地有本地仓库，拉取远程分支更新到本地仓库，从远程仓库中获取最新版本

方法一

$ git fetch origin master #从远程的origin仓库的master分支下载代码到本地的origin maste

$ git log -p master.. origin/master #比较本地的仓库和远程参考的区别

$ git merge origin/master #把远程下载下来的代码合并到本地仓库，远程的和本地的合并

# 方法二

$ git fetch origin master:temp #从远程的origin仓库的master分支下载到本地并新建一个分支temp

$ git diff temp #比较master分支和temp分支的不同

$ git merge temp #合并temp分支到master分支

$ git branch -d temp #删除temp

3.git pull它是git fetch与git merge的结合，在远程仓库获取最新版本然后在与本地版本合并

举例：将远程主机origin的master分支拉取过来，与本地的branchtest分支合并。

$ git pull origin master:branchtest

来自 <<https://www.cnblogs.com/lalalagq/p/9968949.html>>

查看分支：git branch（本地）

查看所有分支：git branch -a 红色的为远端分支

查看远端分支：git branch -r

删除本地分支：git branch -d branch\_name

删除远端分支：git push origin --delete branch\_name

当地分支的建立：git branch branch\_name（在哪个分支下创建，则与哪一个分支相同）

远端分支的建立:git checkout -b branch\_name(先在本地建立分支）git push origin new(将其推向远端）

回退到以前的版本

1.Tortoisegit 选择显示日志，在日志中选择版本，右键，重置到此版本

2.git指令回退

1>git reset --hard 目标版本号（7位，通过git log获得）

2>git push -f强制推上去

3.git指令，如果我们想回退到以前的版本，又想保存该目标版本后的版本，记录下这整个版本的变动流程

1>git revert -n 版本号 反做

2>git commit -m "xinxi"提交

3>git push 推送到远端

一般本地分支都对应一个远端分支，这样上传时方便，本地分支中不会存在远端分支不存在的内容。