

学习记录

1. git 命令

git branch name 创建分支

git branch 查看当前目录所有分支

git branch -a 查看远程分支（红色字）

git checkout name 切换分支

git checkout -b name 创建并切换到该分支

git init 创建本地代码库

git remote add <分支名称><仓库url> 创建服务器代码库

git clone url 克隆服务器代码到本地

git clone url directory 克隆服务器代码到指定文件目录

git status 查看缓存区的状态信息

git stash 进行栈内保存

git stash pop 栈内状态恢复

git stash list 栈内状态存储信息列表

git stash apply stash@{1} 读取指定版本号的状态

进阶：

当你多次使用'git stash'命令后，你的栈里将充满了未提交的代码，这时候你会对将哪个版本应用回来有些困惑，'git stash list'命令可以将当前的Git栈信息打印出来，你只需要将找到对应的版本号，例如使用'git stash apply stash@{1}'就可以将你指定版本号为stash@{1}的工作取出来，当你将所有的栈都应用回来的时候，可以使用'git stash clear'来将栈清空

git branch --set-upstream-to=<远程主机名>/<远程分支名> 更改当前分支追踪的服务器分支

git checkout -b <本地分支名>=<远程主机名>/<远程分支名> 更改当前分支追踪的服务器分支简写

git branch <本地分支名>=<远程主机名>/<远程分支名> 更改当前分支追踪的服务器分支简写

git branch -vv 查看全部分支的追踪关系

git branch -d <本地分支> 删除本地分支

git push origin --delete <远程分支> 删除远程分支

git push origin :<远程分支> 删除远程分支（注意：push origin :second 有空格冒号前而后没有）

git fetch <远程主机名> <远程分支名>：<本地分支名> 指定分支与本地分支同步

git fetch <远程主机名> 同步代码库的全部分支
git merge <本地分支> 同步本地代码分支
git merge <远程主机> <远程分支> 同步远程到本地
git reset <版本号> 回退到指定版本
git reset HEAD^ 回退到上次提交
git reset HEAD^^ 会退到上上次提交

2.

1

在Git中，文件状态是一个非常重要的概念，不同的状态对应不同的操作。

Git在未进行commit操作之前，存在三种状态：Untracked files, Changes not staged for commit及Changes to be committed。

Untracked files就是你必须把某些文件放到Git工作目录中，但又不能提交它们，比如保存了数据库密码的配置文件啦，等等。那么看着也烦怎么处理呢？

好在Git考虑到了大家的感受，这个问题解决起来也很简单，在Git工作区的根目录下创建一个特殊的.gitignore文件，然后把要忽略的文件名填进去，Git就会自动忽略这些文件。

(强迫症患者，笑了...)

END

3. git分支合并尝试

本地second 分支向远程master分支合并 失败

本地second分支向远程second分支和并 成功

4.

通常我们提交git的时候都是

```
git add .  
git commit -m "some str"  
git push
```

这三大步，而实际上，你只需要两条命令就够了，除非有新的文件要被添加进去。

```
git commit -am "some str"  
git push
```

5. git rebase 和 git merge 的区别

git merge是将B于A合并有冲突手动解决完毕在合并，而git rebase 在遇到冲突时可选择 处理完冲突继续git rebase --continue 跳过git rebase --skip 取

消合并git rebase -about

已解决

1. 在本地创建新分支 然后新增一个文件但是commit出现下面的效果?

查了查untracked files 是可忽略的文件

```
meicais-MacBook-Pro-2:delete yunshan$ git add .
meicais-MacBook-Pro-2:delete yunshan$ git commit -m "add"
On branch first
Untracked files:
  ../add/add2的副本.txt

nothing added to commit but untracked files present
meicais-MacBook-Pro-2:delete yunshan$
```

看到提示的意思是有文件没有进入commit 状态 但是我已经git add .了, 后来又尝试了一下git add . 然后git status 查看依然是这样 我使用单条git add ../add/add2的副本.txt就没问题了。。

2.

```
meicais-MacBook-Pro-2:delete yunshan$ git checkout second
Switched to branch 'second'
Your branch is based on 'origin/second', but the upstream is gone.
(use "git branch --unset-upstream" to fixup)
meicais-MacBook-Pro-2:delete yunshan$ git branch
  first
  master
* second
meicais-MacBook-Pro-2:delete yunshan$ git fetch origin master:second
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
meicais-MacBook-Pro-2:delete yunshan$ git fetch origin master:second
fatal: Refusing to fetch into current branch refs/heads/second of non-bare repository
meicais-MacBook-Pro-2:delete yunshan$ git fetch origin second:second
fatal: Couldn't find remote ref second
meicais-MacBook-Pro-2:delete yunshan$ git branch -ar
  origin/HEAD -> origin/master
  origin/first
  origin/master
meicais-MacBook-Pro-2:delete yunshan$ git fetch origin first:second
fatal: Refusing to fetch into current branch refs/heads/second of non-bare repository
meicais-MacBook-Pro-2:delete yunshan$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
meicais-MacBook-Pro-2:delete yunshan$ git fetch origin first:second
From https://github.com/weijiyang/Git
! [rejected]        first      -> second  (non-fast-forward)
meicais-MacBook-Pro-2:delete yunshan$
```

模拟场景远程仓库中有新代码需要合并到本地分支上, 出现了问题, 在本地second分支中使用git fetch origin first:second 会出错, 网上经过查找发现是需

要进入master分支进行操作。

3.

```
meicais-MacBook-Pro-2:Git yunshan$ git remote
origin
meicais-MacBook-Pro-2:Git yunshan$ git remote -v
origin https://github.com/weijiyang/Git.git (fetch)
origin https://github.com/weijiyang/Git.git (push)
meicais-MacBook-Pro-2:Git yunshan$ git remote add firstbranch https://github.com/weijiyang/Git.git
meicais-MacBook-Pro-2:Git yunshan$ git remote
firstbranch
origin
meicais-MacBook-Pro-2:Git yunshan$ git remote -v
firstbranch https://github.com/weijiyang/Git.git (fetch)
firstbranch https://github.com/weijiyang/Git.git (push)
origin https://github.com/weijiyang/Git.git (fetch)
origin https://github.com/weijiyang/Git.git (push)
meicais-MacBook-Pro-2:Git yunshan$ git branch
  first
  master
* second
```

理解：remote对远程仓库、服务器、分支进行操作 firstbranch是对远程服务器新建一个仓库 origin则是远程仓库的最初的命名

待解决

1. git add . 失效？ git add name 才生效？