**GIT学习笔记**

1. 安装
2. 注册账号mirshan [mirshan@163.com](mailto:mirshan@163.com) [hanpeng@126.com](mailto:hanpeng@126.com)并且需要通过邮箱审核
3. 终端注册账户：

git config --global user.name "mirshan"

git config --global user.email "mirshan@163.com"

1. 新建一个工程目录，并通过git init升级成为管理仓库

git init

此时工程目录已经成为git仓库了，文件的创建和修改将全部进行跟踪，此时相当于在仓库门口

1. 创建新文件并执行git add 文件名，如git add program.rtf

此时相当于放到了仓库里，还未上架

1. 输入git commit -m “第一次提交，修改内容为…”回车，此时相当于把代码提交到了仓库货架上，并贴上了第一次的标签，以便后面回滚版本
2. git status命令是查看当前是否有未更新代码
3. git diff命令是查看哪里有区别
4. git log查看提交过的版本，并且可以看到版本的编号

git log --pretty=oneline是以简单方式查看

git relog查看使用过的命令

本地版本回退

1. git reset --hard HEAD^ 一个^是前一个版本，两个^是再往前一个版本
2. git reset --hard HEAD^^ 回到上上个版本
3. git reset --hard HEAD~100 回到前100个版本（不实用）
4. git reset --hard 具体版本号 版本号指的是编号，不用全写完，复制开头一部分即可
5. git checkout --文件名，在没有git add 前可以使用此命令撤销掉新代码 ；如果已经提交到了仓库工作区，但没有commit此命令可以回退至add后的状态

连接远程GIT

1. 创建SSH KEY:

黑屏输入：ssh-keygen -t rsa -C “mirshan@163.com”

记录.ssh目录的位置，如/Users/hanpeng/.ssh/，并进入此目录下。

两个文件 id\_rsa私钥 id-rsa.pub公钥。可以通过cat命令查看文件中的key。

每台设备一个不同的key。复制上传至github官网，方法如下：

1. settings
2. SSH and GPG keys
3. New SSH key
4. 输入标题和公钥，保存即可，此后此电脑即可上传版本至GIT HUB了
5. 通过ssh -T git@github.com来测试是否成功，成功后刷新页面，钥匙变绿

到此已经连接上了github

1. 关联远程仓库
2. 复制github远程地址，方法:通过小猫，创建项目后，右侧有clone or download,点击后use SSH,此时能看到地址复制并使用以下命令
3. 进入本地git工程目录：执行以下命令进行关联

git remote add origin [git@github.com:mirshan/mirs.git](mailto:git@github.com:mirshan/mirs.git)

从origin后面是复制来的地址

1. 如果不想关联，执行git remote rm origin即可
2. 下载github仓库，这点很重要，不然无法上传本地代码

git pull origin master --allow-unrelated-histories

推文件方法：

Git push origin master

正常开发是先有远程库

去远程库复制clone SSH地址

执行git clone [git@github.com:mirshan/mirs.git](mailto:git@github.com:mirshan/mirs.git)

此时会把库克隆到本地

这样本地和远程库完全相同，未来开发中可以修改本地后

1、git add 文件名

2、git commit -m “项目修改描述文字”提交； -m前加文件名，只提交某一个文件

3、然后再git push origin master推到远程库里

1. 推代码常用技巧：
2. 建立.gitignore文件，把不需要每次都推的文件，如lib包写到文件中，此文件中记录的文件，在推拉时不参与传输
3. 示例：git add .

git commit . -m “something”

git push origin master

1. Github创建一个新的工程注意：选择initialize…readme

分支管理

1. 创建分支git branch 分支名，如git branch mirs1
2. 查看目前的分支git branch，当前分支有\*
3. 切换分支git checkout 分支名，如git checkout mirs1
4. 创建与切换分支同时进行git checkout -b mirs2

使用方法：

A.Git branch mirs1进入mirs1分支，处理文件后

B.git add .全部提交

C.git commit -m “描述文字”

D.git checkout master切换到master轴

E.git merge mirs1把mirs1的文件合到主轴上

正常情况下，需要在github上建立使用分支管理代码：

1. 在本地建立分支git branch hanpeng
2. 切换到分支上git checkout hanpeng
3. 修改文件
4. 提交代码git add .
5. 上仓库货架git commit -m “描述文字”
6. 上传到github相应分支上git push --set-upstream origin hanpeng (首次需要创建远程分支)

git push origin (除首次外都可以这样提交到hanpeng分支上)

如果切换到一个新的电脑上，使用分支文件的办法：

1. git clone [git@github.com:mirshan/mirs.git](mailto:git@github.com:mirshan/mirs.git) 此时本地下载好了master分支,注意只能下载到master分支
2. 建立hanpeng分支，git branch -b hanpeng
3. 切换分支git checkout hanpeng
4. 拉文件：git pull origin hanpeng
5. 修改文件完成开发后上传文件git add .
6. 上货架git commit -m “描述内容”
7. 提交到自己的分支上git push origin hanpeng
8. 切换到主分支git checkout master
9. 合并项目git merge hanpeng

正常开发有master主分支

Dev用于测试的分支

各开发使用的分支

项目完成后提交到dev分支上，测试通过没问题后，再合到master分支上

**git checkout -b dev origin/dev在本地创建dev分支同时在github上也创建dev子分支**

打标签

1、Git tag V1.0打个版本号标签，如git tag v1.0 git tag v2.0

2、回滚到1.0版本 git checkout v1.0

3、推到远程仓库git push origin v2.0