Git 加入工程的命令行输入

1. 提高pull和push的速度

sudo vim /etc/hosts

在该文件末尾添加

192.168.xx.xx github.com

185.31.17.xx github.global.ssl.fastly.net

这两个IP地址需要查找离你最近的服务器地址。在IPAddress.com输入这两个网址就可以查到。

之后更新DNS缓存（具体命令因系统而异，注销或重启也可以）。然后就会发现速度提高了。

2. 新建一个文件夹，初始化其为git文件夹。

git inti

3. 连接远程仓库。

git remote add origin https://github.com/HanlinWei/ionic_robot.git

4. 同步代码。

git pull origin master

origin

代表我们的本地仓库，即本地文件系统。master代表远程仓库的分支。表示把远程仓库的master分支的最新代码同步到本地。

5. 分支管理

查看分支，当前所在分支有星号标记

git branch

新建分支，新建的分支在push之前不会显示在远程仓库上，github网页端看不到。

git branch 分支名

切换分支，commit会提交到当前所在分支

git checkout 分支名

删除分支（一般别干这种事，留着不碍事）

git branch –d mybranch //如果该分支没有合并到主分支会报错

分支合并（先确认代码无误）

例如，我们之前建立了mybranch进行我们的开发。测试通过后，我们想要把mybranch合并到主分支master里，进行代码的同步。
首先切换到master：git checkout master
合并：git merge mybranch
查看分支之间的区别：git diff master mybranch
此时可能什么都不会输出，表示两者已经同步了。
但是这是假象，只是在本地的两条branch已经同步了，远程仓库里两条branch还是不一样的。可以在Github网页端确认。
此时需要把我们合并后的master push到远程仓库。
git push –u origin master
直接push可能会报错,如下图。

```
weihanlin@ubuntu:~/Documents/ionic_app$ git push -u ori
gin master
To git@github.com:HanlinWei/ionic_robot.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'git@github.com:Hanl
inWei/ionic_robot.git'
hint: Updates were rejected because the remote contains
 work that you do
hint: not have locally. This is usually caused by anoth
er repository pushing
hint: to the same ref. You may want to first integrate
the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push -
-help' for details.
```

这是因为如果两个开发者从中心仓库克隆代码下来，同时作了一些修订，那么只有第一个开发者可以顺利地把数据推送到共享服务器。第二个开发者在提交他的修订之前，必须先下载合并服务器上的数据，解决冲突之后才能推送数据到共享服务器上。
此时应该先pull，再push。
git pull

git push –u origin master

6. 提交代码。假如修改或者新建了某个文件（main.cpp)

git add main.cpp

git commit -m "提交日志，概括一下这一次修改"

commit相当于在本地的git仓库提交了一份版本。git log可以查看目前已有的提交（如果尚未同步，只能看到当前本地的commit记录）。commit后会有一个commit的ID，日后可以使用这个ID来回退到以前的版本。

队友需要你的代码，或者你工作结束要下线了，就把代码push到远程仓库。可以先push到自己

的分支，协商后再合并分支。

git push -u origin 分支名

7. 回退到以前的版本。

回退命令：

$ git reset --hard HEAD^　　回退到上个版本

$ git reset --hard HEAD~3　　回退到前3次提交之前，以此类推，回退到n次提交之前

$ git reset --hard commit_id　　退到/进到 指定commit的sha码