Git常用命令解析

Git push -u origin master

origin指向的就是我本地的代码库托管在Github上的版本

git commit和git push的区别：

git commit是将改动提交本地仓库

git push是将改动提交远端仓库

git diff后加分支名，如: git diff master，就可以显示这个分支下所有提交的改动

git format-patch -n命令：

其中n表示一共要为之前提交的几个节点打patch，会按照提交的先后顺序生成n个.patch文件的

Git push origin branch命令：

将新创建的本地分支push到远端，作为远端分支，否则用git branch命令创建的分支仅在本地能看到，在远端是查看不到的

git branch: 查看本地分支

git branch -r: 查看远端分支

git版本回退命令：

git reset –hard HEAD^

就是可以让版本回退到上一个版本，这样就是原来的代码，但是在回退的过程中，不能同时打开代码的编辑和开发工程

而且一个代码只有回退了，才能打上刚才生成的.patch文件，一个已经打过patch的分支上没有办法再打上相同的patch

git reflog: 显示所有在git上进行的操作和提交的id号

例如：

$ git reflog

1db4a3e HEAD@{0}: reset: moving to HEAD^

07e77f9 HEAD@{1}: am --abort

07e77f9 HEAD@{2}: commit: Add the third char variable

1db4a3e HEAD@{3}: checkout: moving from master to PATCH

1db4a3e HEAD@{4}: commit: Print another char

16aebd5 HEAD@{5}: commit: Char Print Test

65e3eac HEAD@{6}: commit (initial): Struct Test Branch: master

如果说当前head在Print another char的地方，那么这时候如果想前进到Add the third char variable的版本，则需要进行如下命令行的操作：

git reset –hard 07e77f9

再git log一下，就可以看到前进到了最新版本

git show可以显示当前在这个提交版本的情况下文件的具体内容

git apply \*.patch命令：

在执行以后，只是在本地打上了一个patch，这时候需要重新执行git add \*.file和git commit才能在本地库中更改生效，然后再push这个文件才能在远端仓库中生效

Git切换远端仓库的命令：

查看当前连接到的远端仓库

git remote -v

origin https://github.com/BellaZhong/DeepLearningGit.git (fetch)

origin https://github.com/BellaZhong/DeepLearningGit.git (push)

切换远端仓库：

git remote set-url origin <https://github.com/BellaZhong/BellaZhong.github.io>

再次查看，就已经切换到了新的远端仓库

$ git remote -v

origin https://github.com/BellaZhong/BellaZhong.github.io (fetch)

origin https://github.com/BellaZhong/BellaZhong.github.io (push)