一晚上学会最实用的Git!

By wudifeixue credit to S1ngS1ng

简介

一开始，我一直处于对Git的应用和流程似懂非懂的状况，由于星星的热情和我喋喋不休的提问，然后还记录了下来，嘿嘿。就产生了这份Git笔记。也是非常感谢星星半夜教了我好几个小时熬了好久。关于格式大家就凑合看吧，我尽力整理了下，当然我也很欢迎你用这份教程来freeCodeCamp.cn贡献代码哦~ 这里提一下，我用 // 这个符号当注释用，虽然大家都有JS编程经验应该很熟悉，不过再提醒下吧..

关于添加项目

首先fork一份repository.
然后clone 你 fork 的 repo.
举例：git clone https://github.com/wudifeixue/freecodecamp.cn
再git remote add upstream 源项目地址
举例：
git remote add upstream https://github.com/FreeCodeCampChina/freecodecamp.cn
其中 upstream 的名字你可以自己定，我一般喜欢用这个。
假设伙伴Jeremy来了。那就可以去添加他的。
举例：git remote add Jeremy https://github.com/Jeremy/freecodecamp.cn

到了本地

用terminal进入到repo的文件夹后
可以使用：git checkout -b newBranchName //-b可以理解为build
来添加一个新的branch
然后在这个上面写代码，add，commit，然后 git push origin newBranchName

这个过程就相当于，你自己在自己fork的镜像里面操作。如果你正玩儿着，源项目突然更新了。
你只需要 git pull upstream dev //dev is branch name
就可以把 dev branch上的更新获取到本地。

这样流程的好处就是，首先咱们可能都没有源项目的修改权限，不能直接 push 到源项目中。
但通过push 到自己的fork，再发 PR 的方式，是肯定可以工作的。

之前使用的

git remote add upstream 源项目地址
如果想查看项目都有哪些remote
使用：git remote -v

基础branch

git checkout branchName //切换到本地 branchName
git checkout -b newBranchName //新建branch
git pull upstream dev //更新本地到upstream的dev branch

更多branch

对的。一般我习惯的是，branch 加上一个一眼就能认出来的名字。
git checkout -b update-readme //创建update-readme这个branch
git checkout -b fix-runtime-error //创建fix-runtime-error这个branch

想把一个branch的改动发到另外一个branch里面：
本地可以直接 cherry-pick，可以git merge，还可以用刚才说过的方法，reset之后再checkout再commit
首先git log找到commitHash
//commitHash就是类似commit 7ce0d398e9e083fc94bf39e1c5e98eea0973f9c7 的东西
//commitHash只需要7ce0d39这7位就足够
然后，git checkout dev
然后 git cherry-pick commitHash //使用之前的commitHash
这就能复制一个commit到另外一个branch上。

删除branch

比如我们要删除 wudifeixue这个branch。那么，我们是肯定要先checkout到别的branch。不能自己删自己。
git checkout dev
git branch –d wudifeixue //一般删除branch
如果你的wudifeixue branch有没有merge 的 commit，那么会提示你不让删！
你可以 git branch -D wudifeixue //强制删除wudifeixue branch

最普通的commit过程

git add
git commit -m "xxxxxxx" //quick commit
git commit //进入vim commit
进入vim编辑完成后，输入 :wq 来保存commit并且退出vim

查看git状态

git status //查看commit情况和多种信息
git branch //查看branch信息
git log //看commit历史，退出按下键盘上的q
git log 之后，ctrl + D, F 是向下翻页。ctrl + U, B 是向上翻页
D和U翻半个屏幕，F 和 B 翻一个屏幕
Git log的特效！

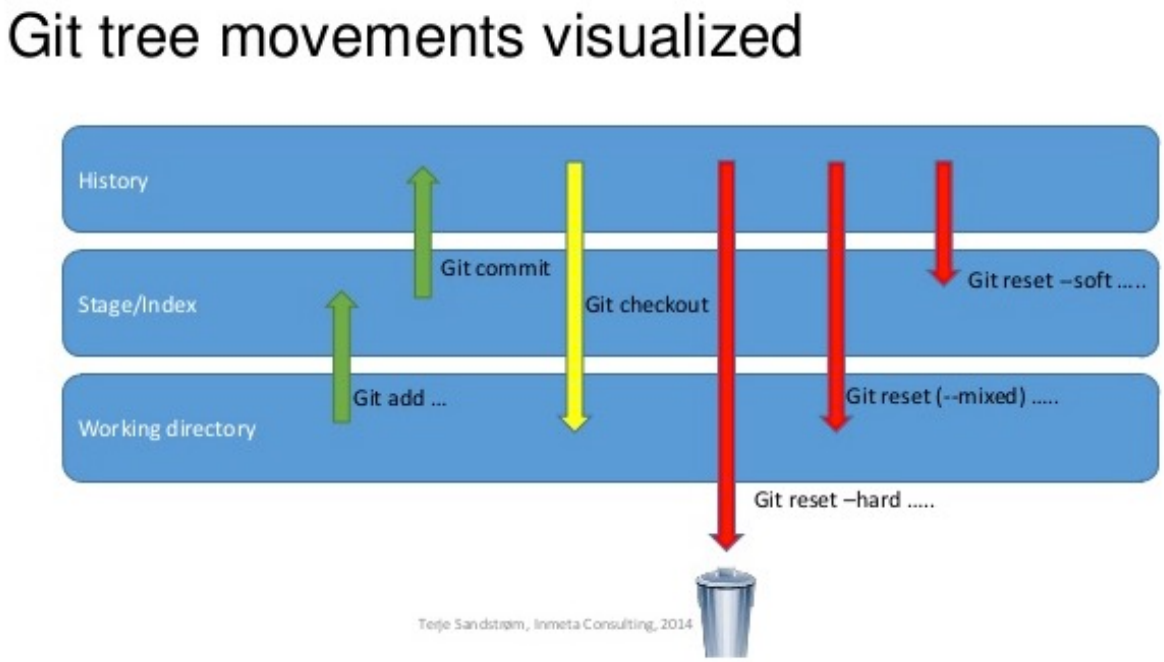
git log --graph --pretty=format:'%C(bold red)%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold green)<%an>%Creset' --abbrev-commit --date=relative
和
git log --graph --pretty='%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset' --abbrev-commit

救命稻草

git reflog //很有用！救了我好几次，这个东西，是你的本地操作历史。
比如你 commit 了，然后 rebase , 然后pull，然后 checkout。
这一串都弄完，你发现你不该commit，但现在情况已经很复杂了😂
git reflog！看到本地历史，然后获得想恢复的版本，例如 HEAD@{7}
这时候就可以恢复到指定版本
git reset --hard HEAD@{7}
来恢复到之前的状态

关于reset

reset 这个命令很麻烦。因为它有三种模式。
git reset --hard
git reset --soft
git reset --mixed
git reset --hard <commitHash> //删掉commit
git reset readme.md //这种默认就是--mixed的reset
hard reset会reset文件，soft reset是reset commit, mixed reset是reset commit之前的add
具体看下图流程：



我需要设置我的Git信息

git config 来设置邮箱一类的
git config --list --global //看我所有的设置

git config --global user.email "xxxx@xxx.com" //设置邮箱
git config --global user.name "xxx" //设置用户名
关于怎么设置隐私邮箱还匹配GitHub看下文
https://help.github.com/articles/keeping-your-email-address-private/

ssh -T git@github.com 来验证ssh
Hi wudifeixue! You've successfully authenticated, but GitHub does not provide shell access.
ssh-add //输入密码
windows 可以用 putty 的 agent

我比较懒想少打东西

使用alias来定义自定义git命令!
git config --global alias.co checkout //把checkout 命令缩短成了co
//git co branchName

之前的超长log特效很棒，但是太长了不想复制!
用alias做个自定义git命令吧!
git config --global alias.nice "log --graph --pretty='%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset' --abbrev-commit"

//注意太长的命令有很多空格要双引号包住
//git nice 就可以看超棒的特效了

//按q退出特效

你要看设置好的alias么?
git config --list --global | grep alias

完成本地修改，提交远程repo

git push origin newBranch //提交本地修改到远程repo
git push origin :newBranch就是删除远程目标的branch //这个一般手动在GitHub上删更好些
//其实可以git push origin xxx:newBranch，xxx是个可选参数