Git的学习

一、基本命令

start a working area (see also: git help tutorial)

clone Clone a repository into a new directory

init Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)

add Add file contents to the index

mv Move or rename a file, a directory, or a symlink

reset Reset current HEAD to the specified state

rm Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)

bisect Use binary search to find the commit that introduced a bug

grep Print lines matching a pattern

log Show commit logs

show Show various types of objects

status Show the working tree status

grow, mark and tweak your common history

branch List, create, or delete branches

checkout Switch branches or restore working tree files

commit Record changes to the repository

diff Show changes between commits, commit and working tree, etc

merge Join two or more development histories together

rebase Reapply commits on top of another base tip

tag Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)

fetch Download objects and refs from another repository

pull Fetch from and integrate with another repository or a local branch

push Update remote refs along with associated objects

1. 正式学习
2. 添加git

进入某个文件夹，git init 添加git。

git add a.txt 表示添加a.txt 这个文件。

git commit –m ‘add a txt’ 提交所有操作。。-m 后面是写信息描述的。一定要写!不然不知道当时做了啥。

git status 查看状态，可以查看当前文件夹有哪些文件需要add

1. 查看Git信息

使用 git log 可以查看所有的版本提交信息，包括版本id和版本修改信息描述

还可以使用 git log –pretty=oneline 这个命令，简单显示版本号和信息

1. 版本回退

突然发现新的修改不用改了！！新功能取消了！！退回到以前的一个版本继续做！！

如果只需要回退上个版本，可以使用 git reset –hard HEAD^ 这个命令。。

HEAD 就是当前文件夹

也可以直接会退到指定版本，git reset –hard 123456 这个命令。。其中123456只不过是id中的前几位。。git会自己找的，即使没写全！当然可以全写.

好了，现在退回去了！！简直美滋滋！！

但是又要再回去。新功能不用删！

好吧，可以使用 git reflog 来查看命令历史。。。

再使用git reset –hard 123457 命令。又回来啦！！

要删除某文件 ，可以使用 git rm a.txt 命令

git ls-files 可以查看当前文件夹下面所有文件名名

ps: 可以使用 git diff a.txt 这个命令查看修改了啥

如果某个文件需要撤销修改，如果还没提交。使用git checkout -- a.txt 命令其中 “--” 一定要有.

git checkout -- 其实是用版本库里的版本替换工作区的版本，无论工作区是修改还是删除，都可以“一键还原”。

1. 远程对接GitHub上的项目

现在GitHub上建好项目，所在电脑上的SSH key 要关联在GitHub上!

在本地，使用

git remote add origin git@github.com:jiajianhao/HardyGit.git

把远程项目git加进来。。

首次传送代码，使用命令git push -u origin master把本地master代码推到远程master上

-u的作用就是把本地master和远程master分支关联起来

再以后，直接使用git push origin master 就好了

ps： origin 其实是指你的代码库repository。。master是他的第一个分支，我们默认的主分支

如果不想要origin，可以换，比如

git remote add github [git@github.com:jiajianhao/HardyGit.git](mailto:git@github.com:jiajianhao/HardyGit.git)

这样，远程代码库的标识就是github，以后执行啥用github就行了

比如： git push github master 推送主分支代码

这样的好处就是不同的git服务商，可以区分开来，而不是笼统的origin

1. 抓取

git clone [git@github.com:jiajianhao/hello-world.git](mailto:git@github.com:jiajianhao/hello-world.git)

clone后面跟的是地址

1. 分支管理

执行创建分支

git checkout -b dev 这个会自动转到分支dev。。-b参数表示创建并切换

git checkout命令加上-b参数表示创建并切换，相当于以下两条命令

git branch dev 和 git checkout dev

使用git branch 查看当前有哪些分支

git checkout master 切回主支

git merge dev 主支合并dev上的修改

git branch –d dev 删除dev分支

git log --graph --pretty=oneline --abbrev-commit 可以查看分支和主分支的合并情况

用git log --graph命令可以看到分支合并图。

如果遇到冲突，手动解决，会有箭头分割线标志出来，各个修改的位置。。这个最好在SourceTree 上看。

1. 分支策略

1>bug分支

在自己分支mybranch开发的时候，发现dev上有bug，需要立即修复！

可以暂停自己的分支开发，把工作区保存起来。。

使用 git stash 保存当前工作区所有的操作。。。

切换回dev。。git checkout dev

在dev分支上创建临时分支issue01分支，git checkout –b issue01

修改好问题，add，commit，再切换回dev

使用 git merge --no-ff -m 'fix issue01' issue01

这样dev上就修复好问题啦

删除issue01分支。。继续回去开发

切回我的分支mybranch，

把暂存的数据拿出来，git stash pop 这样暂存的内容拿出来了，并且暂存的数据也一并删除了。

OK，继续做吧。

2>强行删除分支

git branch –D dev 这里-D表示强制删除

3> 多人协作

多个分支创建分支，都需要得需要推送到远程服务器。比如master、dev等

至于自己的分支，看情况。

推送分支 git push origin master,git push origin de 酱紫

抓取分支 git clone [git@github.com:jiajianhao/hello-world.git](mailto:git@github.com:jiajianhao/hello-world.git)

在本地，只能看到master，还需要创建本地dev

git checkout –b dev origin/dev

这样创建了本地dev，我这边推送的时候，直接就推送上去了，默认本地dev和远程dev关联了。

如果没关联，可以执行命令 git branch –set-upstream dev origin/dev

这样就可以推送dev修改了，比如git push origin dev

小伙伴们的修改也要拉取过来，在dev分支，可以执行git pull命令。。

遇到冲突，手动改再上传。没遇到，那就万事OK

1. 引入tag

发版本的时候的时候最好加上tag。。。

比如1.1版本已经完成，即将发布，使用命令 git tag –a v1.1 –m ‘1.1版本’

这是附注标签命令。。。建议使用附注标签

这样想要找v1.1的时候，只需要git show v1.1就可以很快找到commit id 了

可以推送标签到远程 git push origin v1.1

打错标签也可以删除 git tag –d v1.1

1. 忽略特殊文件

我们需要忽略系统默认生成的一些文件，比如缩略图，编译的中间文件等等

使用.gitignore文件，如果没有，可以自己创建。

1. 本地创建别名

git config --global alias.st status

这样的创建本地电脑上git可以时候用st 代替status事项查看状态，比如 git st就行！

--global 表示全局

差不多了，最后，git官网是<https://git-scm.com>

还有我学的廖雪峰的，大家可以去百度一下廖雪峰的官网。