

Git学习报告

这篇主要根据廖学峰的git教程进行。

什么是git

Git是一个开源的分布式版本控制系统，可以有效、高速的处理从很小到非常大的项目版本管理。

Git的基本操作与相关概念

版本库

什么是版本库呢？版本库又名仓库，英文名**repository**，你可以简单理解成一个目录，这个目录里面的所有文件都可以被Git管理起来，每个文件的修改、删除，Git都能跟踪，以便任何时刻都可以追踪历史，或者在将来某个时刻可以“还原”。

初始化版本库

`git init` 使用`git init`命令将文件夹初始化为一个Git管理目录。

添加文件到仓库

`git add 'filename'` 使用该命令将'filename'添加到暂存区。

将文件提交到仓库

`git commi -m 'sth'`。一般情况下，`commit`的时候需要输入相关的信息，即`-m 'sth'`内容。只有在删除文件后`commit`时，才可以不添加`-m`的内容。

查看当前仓库的状态

`git status`。可以看到修改后为`add`到暂存区的文件，以及`add`到暂存区后未`commit`的文件

查看某个文件改变

`git diff 'filename'`

版本回退

使用`git log`查看提交的信息 在信息中 能够看到每次提交的`commitID` 使用 `git reset -hard 'commitID'` 进行版本的回退 `commitID`只需要前面几位能够唯一识别即可 使用 `git reflog` 查看该仓库的每次提交 课用于返回到未来版本

git的工作区(working directory)和暂存区(stage)

git的工作区可以看成是当前的文件夹。而**暂存区**则是存在于**版本库**中。在版本库中，git add命令将**工作区**修改的文件添加到**暂存区**，而git commit则一次性将**暂存区**的东西加入到**分支**中。使用 git diff 命令是比较**工作区**和**暂存区**的区别。而git diff --cached则是比较**工作区**和**分支(master)**间的区别

git撤销修改

使用git checkout -- filename
可以对工作区的filename文件的修改进行撤销。撤销后最近一次git add 或git commit的状态。即回退到当前暂存区的filename状态 **注意** 在这里--双横杠很重要，如果没有双横杠，则代表切换到另一个分支，为另外一条命令

使用git checkout是将工作区的某个文件撤回到暂存区的状态。但是如果修改后已经add 到了暂存区，此时，需要使用git reset HEAD filename可以将暂存区的文件修改撤销，返回到版本库中该文件的状态

git的删除

使用git rm filename 然后git commit删除版本库中的文件

远程仓库之创建ssh-key

```
ssh-keygen -t rsa -C "youremail@example.com"
```

分支

git 分支管理

在git中 主分支为**master**分支
HEAD指针指向的是当前分支。分支可用于bug调试，新功能的开发的。

创建分支

git checkout -b branchName 创建分支并切换到新分支 等价于 git branch branchName & git checkout branchName 使用git branch 查看当前的分支列表

合并分支

git merge branchName 将分支branchName与当前分支合并

解决提交与远程仓库的冲突

先git pull

把远程仓库下载到本地，下载过程中会自动合并冲突的地方，并在文件中显示。需要手动合并后才能提交。

```
<<<<<< HEAD
test git meger from orgin.
=====
test git meger from copygit
>>>>>> 9b2dc8178f564d798d15cf7063264c4bd6627680
|
```

如果远程仓库中存在本地仓库中不存在的分支，则需要先创建分支并建立与远程分支的链接。

```
git branch --set-upstream dev origin/<branch>
```

删除分支

```
git branch -d branchName 删除branchName分支
```