

git总结

总结自[廖雪峰的Git教程](#)

作者[GitHub](#)

安装git

homebrew 安装git

[homebrew官方链接](#)

安装后的设置

```
$ git config --global user.name "Your Name"
```

```
$ git config --global user.email "email@example.com"
```

使用git进行本地仓库管理

创建版本库（repository）

使用git init 命令将当前目录变成可管理的仓库

```
$ git init
Initialized empty Git repository in /Users/michael/learngit/.git/
```

git add 将文件提交到暂存区

```
$ git add 文件名
```

提交所有文件

```
$ git add .
```

git commit 将暂存区所以文件提交到当前分支

```
$ git commit -m "对于本次提交的注解"
```

git status 查看当前状态

```
$ git status
```

git diff 查看文件区别

```
$ git diff 文件名
```

版本回退

git log 查看日志

利用参数 **--pretty=oneline** 精简信息

```
$ git log --pretty=oneline
```

git reset 回退版本

```
$ git reset --hard HEAD^
```

HEAD为当前版本 上上版本为HEAD^^

返回到指定版本

```
$ git reset --hard 版本号
```

git reflog查看每一次git命令

```
$ git reflog
```

撤销修改

git checkout 撤销修改(本质上 使用版本库版本替代工作区版本)

```
$ git checkout -- 文件名
```

已经提交到了暂存区 用**git reset HEAD <文件名>** 把暂存区修改撤销

```
$ git reset HEAD 文件名
```

删除文件

使用 **git rm** 命令删除后 用**git commit**命令提交

```
$ git rm 文件名  
$ git commit -m "信息"
```

创建合并分支

git branch创建分支

```
$ git branch 分支名
```

git checkout切换分支

```
$ git checkout 分支名
```

一条命令创建切换分支

```
$ git checkout -b 分支名
```

-d 参数删除分支

```
$ git branch -d 分支名
```

git branch命令查看当前分支

```
$ git branch
```

git merge合并分支到master分支

```
$ git merge 分支名
```

git log --graph查看分支合并图

```
$ git log --graph
```

bug 分支

有个bug需要修复 但是你的当前工作还没有提交：

```
$ git status
On branch dev
Your branch is up to date with 'origin/dev'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working
directory)

    modified:   readme.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

1.git stash存储工作现场

```
$ git stash
Saved working directory and index state WIP on dev: 5cf14e5 1
```

现在查看工作区 是干净的:

```
$ git status
On branch dev
Your branch is up to date with 'origin/dev'.

nothing to commit, working tree clean
```

2.创建临时分支:

```
$ git checkout -b issue-001
Switched to a new branch 'issue-001'
```

3.修复bug 并提交:

```
$ git add readme.txt
$ git commit -m "fix bug 001"
[issue-001 25c3a9a] fix bug 001
1 file changed, 1 insertion(+), 1 deletion(-)
```

3.合并 删除bug分支

```
$ git merge --no-ff -m"merge bug fix 001" issue-001
Merge made by the 'recursive' strategy.
 readme.txt | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

4.(1)git stash list 查看保存的工作现场

```
$ git stash list
stash@{0}: WIP on dev: 5cf14e5 1
```

4.(2)git stash apply 恢复现场 git stash drop删除现场

```
$ git stash apply
On branch dev
Your branch is up to date with 'origin/dev'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working
directory)

    modified:   readme.txt

no changes added to commit (use "git add" and/or "git commit -a")
$ git stash drop
Dropped refs/stash@{0} (bcae8d1e5f93636a88d8497370e8ba006c302580)
```

git stash pop 恢复的同时删除stash内容

```
$ git stash pop
```

恢复指定stash:

```
$ git stash stash@{0}
```

5.注意这个bug在当前分支也有 git cherry-pick 提交地址 复制特定提交到当前分支

```
$ git cherry-pick 25c3a9a29ffbb
[dev 86beb0a] fix bug 001
Date: Sat Aug 17 10:01:47 2019 +0800
1 file changed, 1 insertion(+), 1 deletion(-)
```

多人协作

git remote 查看远程库的信息 -v参数显示更详细信息

```
$ git remote
origin
$ git remote -v
origin  git@github.com:wangzitationsky/learngit.git (fetch)
origin  git@github.com:wangzitationsky/learngit.git (push)
wangzitiandeMacBook-Air:learngit myx$
```

本地创建和远程分支对应的分支

```
$ git checkout -b branch-name origin/branch-name
```

建立本地分支和远程分支的关联

```
$ git branch --set-upstream branch-name origin/branch-name
```

远程仓库

添加远程仓库

获取本地SSH密匙 找到.ssh目录 复制其中id_rsa.pub的内容 将此密匙加入GitHub

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

将本地仓库与远程仓库关联

```
$ git remote add origin 仓库地址
```

第一次推送 用git push命令加上-u参数 将当前master分支与远程master分支关联

```
$ git push -u origin master
```

之后每次用 git push 命令将本地的分支推送到GitHub

```
$ git push origin master
```

从远程库克隆

git clone克隆至本地

```
$ git clone 仓库地址
```