

git 常用命令

周威光整理^{*}

2017-7-2

^{*} 简介：恒天云 FTE

1.3 工作区和暂存区

1. 工作区: my_repository 文件夹
2. 版本库: 工作区的隐藏目录 “.git”, 这个不算工作区, 而是 git 的版本库。

Git 的版本库里存了很多东西, 其中最重要的就是称为 stage(或者叫 index) 的暂存区, 还有 Git 为我们自动创建的第一个分支 master, 以及指向 master 的一个指针叫 HEAD。

文件往 Git 版本库里添加的时候, 是分两步执行的:

第一步是用 “git add” 把文件添加进去, 实际上就是把文件修改添加到暂存区;

第二步是用 “git commit” 提交更改, 实际上就是把暂存区的所有内容提交到当前分支。

因为我们创建 Git 版本库时, Git 自动为我们创建了一个 master 分支, 所以, 现在, commit 就是往 master 分支上提交更改。你可以简单理解为, 需要提交的文件修改通通放到暂存区, 然后, 一次性提交暂存区的所有修改。

1.4 管理修改

1. 查看工作区对比版本库中的状态变化

```
1 git status
```

2. 查看指定文件的具体修改

```
1 git diff readme.txt #如果暂存区中存在该文件, 则与暂存区对比, 否则与 master  
分支对比  
2 git diff HEAD — readme.txt # 指定和 master 分支对比
```

1.5 撤销修改

1. 丢弃工作区的修改

```
1 git checkout -- readme.txt #工作区会被还原
```

命令 `git checkout -- readme.txt` 意思就是, 把 `readme.txt` 文件在工作区的修改全部撤销, 这里有两种情况:

一种是 `readme.txt` 自修改后还没有被放到暂存区, 现在, 撤销修改就回到和版本库一模一样的状态;

一种是 `readme.txt` 已经添加到暂存区后, 又作了修改, 现在, 撤销修改就回到添加到暂存区后的状态。

总之, 就是让这个文件回到最近一次 `git commit` 或 `git add` 时的状态。

2. 把暂存区的修改回退到工作区

```
1 git reset HEAD readme.txt #暂存区被清理掉, 不改变工作区的内容
```

`git reset` 命令既可以回退版本, 也可以把暂存区的修改回退到工作区

1.6 删除文件

1. 若不小心删除了文件, 想恢复

```
1 rm test.txt
2 git checkout -- test.txt #用版本库中的恢复
```

2. 若删除了文件, 想同步到版本库

```
1 rm test.txt
2 git rm test.txt
3 git commit -m "remove test.txt"
```

1.7 远程仓库

1. 建立本地库与远程库的关联

```
1 git remote add Life git@github.com:weiguangzhou/Life.git
2 git pull Life master #将远程的 Life 仓库的 master 分支更新到本地, 并与本地 master 合并
3 git push Life master #将本地文件推送到远程库
```

2. 克隆远程库

```
1 git clone git@github.com:weiguangzhou/Life.git
```

1.8 分支管理

1. 创建分支

```
1 git branch #看看当前分支
2 git branch -a #查看所有分支
3 git branch dev #创建dev分支
4 git checkout dev #切换到dev分支
5 git checkout -b dev #创建dev分支，并切换到该分支，相当于上面两个命令
```

2. 合并和删除分支

```
1 git merge dev
2 git branch -d dev
```

3. 解决冲突

```
1 git branch -b feature1 #创建feature1分支，并切换，并修改readme.txt内容，保存提交
2 git checkout master #切回主分支，同样修改readme.txt内容，保存提交。
3 git merge feature1 #合并新建分支时，会冲突
4 git status #查看冲突文件，并手动修改
5 git add readme.txt
6 git commit -m "conflict merge" #提交合并
7 git log --graph --pretty=oneline --abbrev-commit #查看分支合并情况
8 git log --graph #同样可以查看分支合并情况
```

参考文献

- [1] 廖雪峰之 Git 教程