

### 1、git add和git stage的区别是什么

git add：作用是将您需要提交的文件添加到暂存区，需要提交的文件包含文件内容修改（modified）以及新文件（new），但不包括被删除但文件

git stage: 与git add 区别不大，都是将需要提交的文件添加到暂存区

### 2、git rm --cached 和git rm -f的区别是什么

git rm --cached: 默认情况下，git rm file 会将文件从缓存区和你的硬盘中删除。可以使用 git rm --cached。

git rm -f：要删除之前过并且已经放到暂存区的文件，则必须使用强制删除选项，即git rm -f。这是一种安全特性，用于防治误删还没有添加到数据，这样数据不能被git恢复

### 3、git和svn的区别是什么

git和svn的区别:

1.git是分布式的，svn不是

2.git把内容按元数据方式存储，而SVN是按文件

3.git可以有无限个版本库,svn只能有一个指定中央版本库

4.git没有一个全局版本号，而svn有

5.git每个工作成员可以任意在自己的本地版本库开启无限个分支;在svn，分支是一个完整的目录。且这个目录拥有完整的实际文件

6. 提交（Commit）在SVN，当你提交你的完成品时，它将直接记录到中央版本库。当你发现你的完成品存在严重问题时，你已经无法阻止事情的发生。如果网路中断，你根本没办法提交！而Git的提交完全属于本地版本库的活动。而你只需“推”（git push）到主要版本库即可。Git的“推”其实是在执行“同步”（Sync）。

4、筛选出 2018.10.1 到 2018.10.20之间的日志,并且输出为地理图,并且没有做过合并

```
p1901 -- -bash -- 80x24
Date: Mon Mar 25 19:48:27 2019 +0800

xxx

* commit c0c32d7a12d4d085156d5159c8881571eb890e58
Author: admin <admin@wangxuanlin.local>
Date: Mon Mar 25 11:00:15 2019 +0800

aaaaaa
:~
wangxuanlin:p1901 admin$ git log --graph --after='2019.3.25' --since='2019.3.25'
* commit b3173c3bbd24d0e09aa4123d46a1b64d175ce874 (HEAD)
Author: admin <admin@wangxuanlin.local>
Date: Mon Mar 25 19:48:27 2019 +0800

xxx
[wangxuanlin:p1901 admin$ git log --graph --after='2019.3.25' --since='2019.3.25'
--no-merges
* commit b3173c3bbd24d0e09aa4123d46a1b64d175ce874 (HEAD)
Author: admin <admin@wangxuanlin.local>
Date: Mon Mar 25 19:48:27 2019 +0800

xxx
wangxuanlin:p1901 admin$
```

## 5、git init和git clone的区别

git init: 将一个目录初始化为 git 仓库。方法是在目录中执行git init, 执行之后在该项目目录中有个 .git 的子目录, 这就是git仓库, 所有相关的此项目的快照数据都将存放在这里

git clone: 复制一个 git 仓库。方法: 执行 git clone [url], [url] 为你想要复制的项目

## 6.每次提交都忽略.idea文件夹里面的东西怎么办

```
p1901 — -bash — 80x24
* commit b3173c3bbd24d0e09aa4123d46a1b64d175ce874 (HEAD)
  Author: admin <admin@wangxuanlin.local>
  Date:   Mon Mar 25 19:48:27 2019 +0800

    xxx
[wangxuanlin:p1901 admin$ ls
ZhihuSpider      text          text.txt      txt          txt1
[wangxuanlin:p1901 admin$ echo '*.ider' > .giti
[wangxuanlin:p1901 admin$ cat .giti
*.ider
[wangxuanlin:p1901 admin$ git add.
git: 'add.' 不是一个 git 命令。参见 'git --help'。

最相似的命令是
  add
[wangxuanlin:p1901 admin$ git add .
[wangxuanlin:p1901 admin$ git status
头指针分离于 b3173c3
要提交的变更:
  (使用 "git reset HEAD <文件>..." 以取消暂存)

    新文件:   .giti

[wangxuanlin:p1901 admin$
```

7、如果编辑一个文件之后并且加入了暂存区,但是你后悔了,想把文件恢复到没有修改之前的样子,怎么办

```
p1901 — -bash — 80x24

最相似的命令是
  add
[wangxuanlin:p1901 admin$ git add .
[wangxuanlin:p1901 admin$ git status
头指针分离于 b3173c3
要提交的变更:
  (使用 "git reset HEAD <文件>..." 以取消暂存)

    新文件:   .giti

[wangxuanlin:p1901 admin$ ls
ZhihuSpider      text          text.txt      txt          txt1
[wangxuanlin:p1901 admin$ vim file
[wangxuanlin:p1901 admin$ git add file
[wangxuanlin:p1901 admin$ git status
头指针分离于 b3173c3
要提交的变更:
  (使用 "git reset HEAD <文件>..." 以取消暂存)

    新文件:   .giti
    新文件:   file

[wangxuanlin:p1901 admin$
```

8、如何检出标签?

```
wangxuanlin:p1901 admin$ vim file
wangxuanlin:p1901 admin$ git add file
wangxuanlin:p1901 admin$ git status
头指针分离于 b3173c3
要提交的变更:
(使用 "git reset HEAD <文件>..." 以取消暂存)

    新文件:   .giti
    新文件:   file

wangxuanlin:p1901 admin$ git log
commit b3173c3bbd24d0e09aa4123d46a1b64d175ce874 (HEAD)
Author: admin <admin@wangxuanlin.local>
Date:   Mon Mar 25 19:48:27 2019 +0800

    xxx

commit c0c32d7a12d4d085156d5159c8881571eb890e58
Author: admin <admin@wangxuanlin.local>
Date:   Mon Mar 25 11:00:15 2019 +0800

    aaaaaa
    :#
wangxuanlin:p1901 admin$
```

## 9、git fetch 和 git pull的区别

- 1、git fetch 相当于是从远程获取最新版本到本地，不会自动merge;这将更新git remote 中所有的远程仓库所包含分支的最新commit-id, 将其记录到.git/fetch\_head文件中
2. git pull: 相当于是从远程获取最新版本并merge到本地

## 10、如何添加远程仓

```
file          text.txt          txt1
wangxuanlin:p1901 admin$ mkdir git-test
wangxuanlin:p1901 admin$ cd git-test/
wangxuanlin:git-test admin$ echo '作页' > file
wangxuanlin:git-test admin$ git init
已初始化空的 Git 仓库于 /Users/admin/project/p1901/git-test/.git/
wangxuanlin:git-test admin$ git remote add git-testgit@github.com:wangxuanlin/
nux111.git
用法: git remote add [<选项>] <名称> <地址>

    -f, --fetch          抓取远程的分支
    --tags              抓取时导入所有的标签和关联对象
                        或不抓取任何标签 (--no-tags)
    -t, --track <分支> 跟踪的分支
    -m, --master <分支> 主线分支
    --mirror[=(push|fetch)]
                        把远程设置为用以推送或抓取的镜像

wangxuanlin:git-test admin$ git remote add git-test git@github.com:wangxuanlin/
linux111.git
wangxuanlin:git-test admin$ git remote -v
git-test          git@github.com:wangxuanlin/linux111.git (fetch)
git-test          git@github.com:wangxuanlin/linux111.git (push)
wangxuanlin:git-test admin$
```