

## 1、git add和git stage的区别是什么

git add:是将需要提交的文件添加到暂存区，是之前那个要收费的用的

git stage: 和git add差不多，是之后git用的，不收费那个

## 2、git rm --cached 和git rm -f的区别是什么

git rm --cached:默认情况下，git rm 会将文件从缓存区和硬盘中删除，但是可以用git rm--cached在工作目录中留着该文件。

git rm -f: 要删除之前修改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，用于防止误删还没有添加到的数据，这样的数据不能被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，提交在svn中，当你提交你的完成品的时候，它将纸巾记录到中央版本库。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。如果网络中断，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。而你只需要git push到主要版本库即可。git的git push其实是在执行‘同步’。

## 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。

```
* commit 0de0d0b4b65614b67831e6e8b130eb02e0d7551a (HEAD -> master, git_hub/master)
| Author: rimi <rimi@pengdeiMac.local>
| Date: Mon Mar 25 14:58:53 2019 +0800
|
| 克隆
```

## 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，执行之后再该项目目录中有个.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存放在这里

git clone：复制一个git仓库。方法：执行git clone xxxx，xxxx为要复制的项目地址

## 6、每次提交都忽略.idea文件夹里面的东西怎么办

```
pengdeiMac:p1901 rimi$ ls
ZhihuSpider    b.py          file.idea
ai.py          c.py          test.idea
pengdeiMac:p1901 rimi$ git status
位于分支 master
要提交的变更：
  （使用 "git reset HEAD <文件>..." 以取消暂存）

    新文件：   .gitignore

pengdeiMac:p1901 rimi$ git add file.idea
下列路径根据您的一个 .gitignore 文件而被忽略：
file.idea
使用 -f 参数如果您确实要添加它们。
pengdeiMac:p1901 rimi$
```

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

```

pengdeiMac:p1901 rimi$ ls
ZhihuSpider      b.py              file.idea
ai.py             c.py              test.idea
pengdeiMac:p1901 rimi$ vim file
pengdeiMac:p1901 rimi$ git add file
pengdeiMac:p1901 rimi$ git status
位于分支 master
要提交的变更:
  (使用 "git reset HEAD <文件>..." 以取消暂存)

    新文件:   .gitignore
    新文件:   file

pengdeiMac:p1901 rimi$ git reset HEAD file
pengdeiMac:p1901 rimi$ git status
位于分支 master
要提交的变更:
  (使用 "git reset HEAD <文件>..." 以取消暂存)

    新文件:   .gitignore

未跟踪的文件:

```

## 8、如何检出标签

```

pengdeiMac:p1901 rimi$ git log
commit 0de0d0b4b65614b67831e6e8b130eb02e0d7551a (HEAD -> master, git_hub/master)
Author: rimi <rimi@pengdeiMac.local>
Date:   Mon Mar 25 14:58:53 2019 +0800

    克隆

commit 9c73e1d5eaba0527f8088a3221006102e3a21f6e
Author: rimi <rimi@pengdeiMac.local>
Date:   Mon Mar 25 10:30:21 2019 +0800

    three

commit b2b682c4ecfee5213509c367837ef213c1e68503
Author: rimi <rimi@pengdeiMac.local>
Date:   Mon Mar 25 10:29:40 2019 +0800

    第二次修改

commit 406414fed9b5a681d73e99f17d15be9f1a4c7a55
Author: rimi <rimi@pengdeiMac.local>
Date:   Mon Mar 25 10:22:15 2019 +0800

```

## 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、如何添加远程仓库

git remote add git\_hub xxxxxx地址