

1. 撤销未提交的修改

```
...  
  
# 修改文件，之后撤销修改到上次提交状态  
git checkout <file>  
  
# 或者，文件已经进入暂存区，要恢复  
git reset HEAD <file>  
  
git checkout <file>  
  
...
```

2. 修改文件之后，先git add 进入暂存区，然后再git commit

3. git reset 不会真的删除记录，使用git reflog 之后可以看到完整的log，包括reset 操作。另，git reflog 只能看到本地的提交历史？

4. fetch

Git中的fetch和pull,Git中从远程的分支获取最新的版本到本地有这样2个命令：

4.1. git fetch：相当于是从远程获取最新版本到本地，不会自动merge

```
...  
  
git fetch origin master  
git log -p master..origin/master  
git merge origin/master  
...
```

以上命令的含义：

首先从远程的origin的master主分支下载最新的版本到origin/master分支上

然后比较本地的master分支和origin/master分支的差别

最后进行合并

上述过程其实可以用以下更清晰的方式进行：

```
...  
  
git fetch origin master:tmp  
git diff tmp  
git merge tmp  
...
```

从远程获取最新的版本到本地的test分支上

之后再进行比较合并

4.2. git pull：相当于是从远程获取最新版本并merge到本地

```

```
git pull origin master
```

```

上述命令其实相当于git fetch 和 git merge

在实际使用中，git fetch更安全一些

因为在merge前，我们可以查看更新情况，然后再决定是否合并

5. 撤销提交

本地：

```
`git commit --amend`
```

远程：

```

删除远程的master git push origin :master

本地修改

将本地内容作为master 重新push

```

6. github 添加ssh

```

```
cd ~/.ssh
```

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

copy ~/.ssh/id\_rsa.pub 到github 的ssh keys

```

7. 设置账户、邮箱

单独：

```

```
git config user.name anothergitaccount
```

```
git config user.email anothergitaccount@example.com
```

```

全局：

取消

```

```
git config --global --unset user.name
```

```
git config --global --unset user.email
```

```

设置：

```

```
git config --global user.name name
```

```
git config --global user.email email
```

```

8. 本地新建repo 再传给github

```

新建repo

```
git remote add origin 仓库地址。
```

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

```