* 恢复丢失的版本

丢失版本最常见的问题就是 比如使用了 git reset –hard HEAD^ 之类的操作，结果发现丢弃的版本还想恢复回来，但是已经没有任何分支能够reference到这个commit了。幸运的是，git 对各个分支的head还有一份log记录叫做reflog，你可以在.git/logs/refs/heads/ 目录下看到它们。 通过 git reflog 可以显示变更历史。使用类似 master@{1} master@{“2 days ago”}之类的格式，就能索引到你想要的commit。例如对应于git reset –hard HEAD^ 使用 git reset –hard HEAD@{1}即可恢复到reset之前的commit上。

* 出现 \* (no branch)的处理

如果当前正工作在 （no branch）上：  
$ git checkout -b working，就会把（no branch）上的东西checkout到working分支  
如果不小心从（no branch）branch切换到其他分支了，用git log不能查找到（no branch）的信息，不要担心，先用git reflog查到（no branch）的commit，然后：  
$ git checkout -b backup *commit*

* Git push,merge,pull,fetch,rebase各自在什么情况下使用

****Git push****：用本地的refs更新远程的refs。比如从远程的git repository中clone了一个副本，在此基础上做了一些修改，然后用git push将你的修改更新到远程，前提是要有远程的写权限。

****Git merge：****合并不同的分支，详细看[git merge 合并分支](http://blog.microsuncn.com/?p=2000" \o "Permanent link to git merge 合并分支)

****Git pull：****从另一个repository中fetch并且merge到一个本地分支，比如从远程的git repository中clone了一个副本，过了一段时间，远程的版本可能已经更新，用git pull将远程的更新fetch并且merge到当前分支。

****Git fetch：****详细看，[git fetch 从另一个repository中下载objects and refs](http://blog.microsuncn.com/?p=2093" \o "Permanent link to git fetch 从另一个repository中下载objects and refs)

****Git rebase：****详细看，[git-rebase 跟上游分支同步](http://blog.microsuncn.com/?p=1989" \o "Permanent link to git-rebase 跟上游分支同步)