

## git format-patch

git format-patch HEAD^ <==最近的1次commit的patch

git format-patch HEAD^^ <==最近的2次commit的patch

git format-patch HEAD^^^ <==最近的3次commit的patch

git format-patch HEAD^^^^ <==最近的4次commit的patch

git format-patch commit号 <== 最近的 所有commit的patch

1. 两个commit间的修改 ( 包含两个commit )

git format-patch <r1>..<r2>

2. 单个commit

git format-patch -1 <r1>

3. 从某commit以来的修改 ( 不包含该commit )

git format-patch <r1>

## git am

git am \*.patch

git am -abort

git am --resolved

## git log

git log -n //显示最近n条提交

git log -p //查看版本之间的代码差异

git log 18f822e...0bb3dfb //(18f822e,0bb3dfb]的历史记录

git log 18f822e... //18f822e到HEAD所有的记录

^ : 作用相当于回溯一个版本

~N : 指回溯N个版本

查看远程分支与主分支间的差异提交

git log github/master ^master

## git archive

### 导出仓库到zip包

git archive -o "./output.zip" master

git add -u 只添加版本库中的文件变化到暂存区

回退到之前版本创建新分支

git checkout -b new\_branch\_name commit\_id