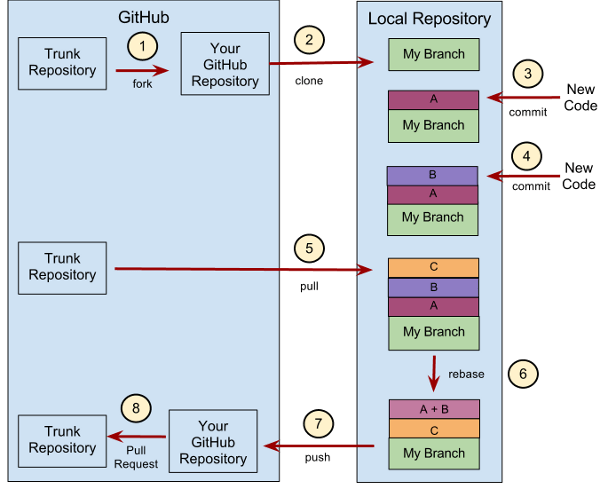
# git



## 新建分支

# 获取主干最新代码

$ git checkout master

$ git pull

# 新建一个开发分支myfeature

$ git checkout -b myfeature

## 提交分支

# 可以用git add . 代替

$ git add --all

$ git status

$ git commit --verbose

### more

git commit -m '提交信息'

# -a 参数就是可以把还没有执行add命令的修改一起提交

git commit -a -m ‘提交信息’

# 1.可以修改上一次的提交信息

# 2.可以将最近的修改追加到上一次的提交上

git commit --amend

## 与主干同步

# 分支的开发过程中，要经常与主干保持同步。

$ git fetch origin

$ git rebase origin/master

## 合并commit

$ git rebase -i origin/master

git rebase命令的i参数表示互动（interactive），这时git会打开一个互动界面，进行下一步操作。

-i部分说明：

pick：正常选中

reword：选中，并且修改提交信息；

edit：选中，rebase时会暂停，允许你修改这个commit（参考这里）

squash：选中，会将当前commit与上一个commit合并

fixup：与squash相同，但不会保存当前commit的提交信息

exec：执行其他shell命令

### more

另外一种合并commit的简便方法，就是先撤销过去5个commit，然后再建一个新的。

$ git reset HEAD~5

$ git add .

$ git commit -am "Here's the bug fix that closes #28"

$ git push --force

squash和fixup命令，还可以当作命令行参数使用，自动合并commit。

$ git commit --fixup

$ git rebase -i --autosquash

## 推送到远程仓库

$ git push --force origin myfeature

git push命令要加上force参数，因为rebase以后，分支历史改变了，跟远程分支不一定兼容，有可能要强行推送（参见这里）。

提交到远程仓库以后，就可以发出 Pull Request 到master分支，然后请求别人进行代码review，确认可以合并到master。

# git命令

