## Git安装-windows

<https://pan.baidu.com/s/1kU5OCOB#list/path=%2Fpub%2Fgit>

## 初始化配置信息

$ git config --global user.name "[name]"

$ git config --global user.email "[email address]"

$ git config --global color.ui auto

## Git SSH Key 生成

ssh-keygen -t rsa -C "dengjili@outlook.com"

id\_rsa和id\_rsa.pub

秘钥：id\_rsa

添加到本地：ssh-add id\_rsa

公钥：id\_rsa.pub

添加到github

## 创建版本库

进入目标文件夹

git init

生成了.git隐藏文件夹

添加文件

git add [file]

将文件添加到仓库

git commit -m "[descriptive message]

修改提交到仓库的文件

执行git status（罗列出所有新增与修改，但是还未提交的文件）

查看差异文件（工作区与暂存区的比较）

git diff

git diff –staged（暂存区与仓库的比较）

指明两个具体的分支比较

git diff [first-branch]...[second-branch]

git diff HEAD -- readme.txt （比较仓库与当前工作区文件）

查看版本的列表

git log

## 版本回退

head代表当前版本，master代表仓库

commit 回退

git reset [commit]

版本回退，回退到上一个

git reset --hard HEAD^

ps: 用HEAD表示当前版本，也就是最新的提交3628164...882e1e0（注意我的提交ID和你的肯定不一样），上一个版本就是HEAD^，上上一个版本就是HEAD^^，当然往上100个版本写100个^比较容易数不过来，所以写成HEAD~100。

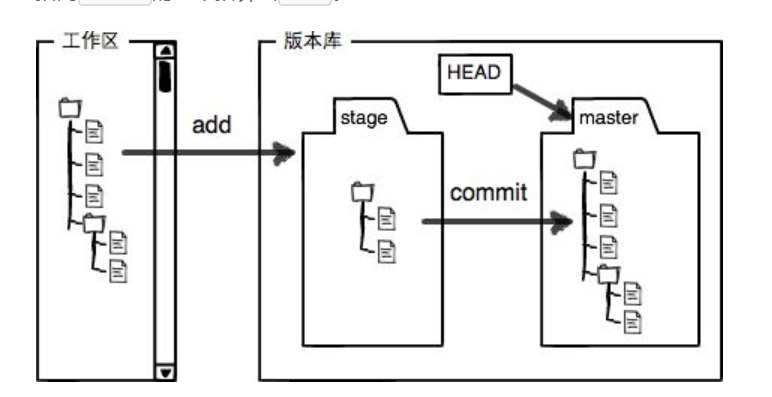
也可以使用前面那个字符，代表具体的版本

查看之前的命令

git reflog

ps：第一列黄颜色代表着版本号

## 工作区-暂存区-仓库



## 撤销修改

丢弃工作区的修改

git checkout -- readme.txt

暂存区的修改撤销掉（unstage），（对暂存区的撤销）

git reset HEAD file

## 删除文件

git rm [file]

## 添加远程库（先有本地库）

echo "# localgit" >> README.md

git init

git add README.md

git commit -m "first commit"

git remote add origin git@github.com:dengjili/localgit.git

git push -u origin master

ps:红色为创造已经存在的仓库

## 从远程库克隆（先有远程库）

git clone git@github.com:dengjili/clonegit.git

## 创建与合并分支

查看分支：git branch

创建分支：git branch <name>

切换分支：git checkout <name>

创建+切换分支：git checkout -b <name>

合并某分支到当前分支：git merge <name>

删除分支：git branch -d <name>

删除远程分支：git push origin -d dev

## 解决冲突树形路径（倒图）

git log --graph --pretty=oneline --abbrev-commit

关闭fast forward模式

git merge --no-ff -m "merge with no-ff" dev

## bug分支

正在dev上进行的工作还没有提交

$ git stash

$ git stash list

$ git stash pop

$ git stash drop

## 多人合作

远程仓库的默认名称是origin

要查看远程库的信息

git remote

创建远程origin的dev分支到本地

git checkout -b dev origin/dev

从远程仓库同步到本地仓库

git pull

设置同步原与目（master/dev:代表本地的dev）

git branch --set-upstream dev origin/dev

## 标签使用

命令git tag <name>用于新建一个标签，默认为HEAD，也可以指定一个commit id；

git tag -a <tagname> -m "blablabla..."可以指定标签信息；

git tag -s <tagname> -m "blablabla..."可以用PGP签名标签；

命令git tag可以查看所有标签

命令git push origin <tagname>可以推送一个本地标签；

命令git push origin --tags可以推送全部未推送过的本地标签；

命令git tag -d <tagname>可以删除一个本地标签；

命令git push origin :refs/tags/<tagname>可以删除一个远程标签。