Git的快速使用：

# Git使用的软件

Git

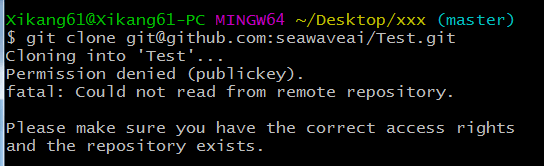
TortoiseGit

Github

# Git的传输方式

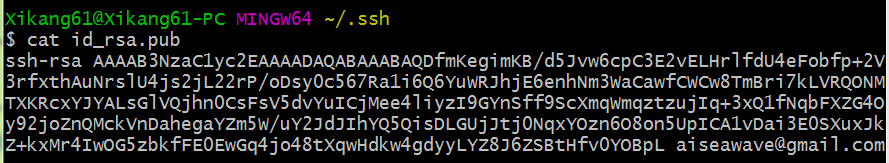
https方式: 每次提交代码都要输入账号和密码，不方便。

ssh 方式: rsa进行加密，在git服务器对应项目上配置公钥，配置一次就ok。

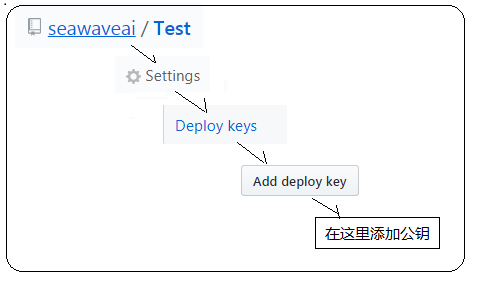


# Git生成SSH秘钥

1. ssh-keygen –t rsa -C “xxx@163.com” (三次回车即可生成ssh key)
2. cat ~/.ssh/id\_rsa.pub (查看公钥)



1. 添加公钥



1. ssh -T [git@github.com](mailto:git@github.com) (测试配置是否成功)
2. 有关ssh更详细问题可以查看

<https://help.github.com/articles/checking-for-existing-ssh-keys/>

# git命令的使用

首先，你需要配置名字和邮箱

git config --global user.name "you\_name" (不输入名字、邮箱就可以查看是否配置成功)

git config --global user.email "xxx@163.com"

git init (初始化本地仓库)

git clone url (克隆远程仓库)

git add xxx (添加单个文件或文件夹)

git add . (添加全部文件)

git commit -m "first commit" (一定要跟-m添加描述信息，否则非法)

git pull origin master (origin部分指定的是远程仓库的git地址，master部分指定的是同步到哪个分支上)

git push origin master

xxx.gitignore 忽略文件

git status (查看自上次提交后文件的修改情况)

git diff

git diff xxx (查看具体文件前后的不同)

git checkout xxx (撤销修改—没执行add命令的情况)

git reset HEAD xxx (撤销修改—执行了add命令的情况)

git log (查看日志---日志包括：id，提交人，提交日期，提交描述)

git log -1 (看一条) git log -2看两条

git log id -1 (也可以加id)

git log -1 -p (加-p查看这条记录具体修改了什么内容)

git branch -a (查看版本库中有那些分支, 前面有\*标识的代表当前分支)

git branch version1.0 (创建分支version1.0)

git checkout version1.0 (切换当前分支到version1.0)

git checkout master

git merge version1.0 (此两条指令,实现version1.0分支合并到master分支)

git branch -D version1.0 (删除version1.0分支---删除的时候,version1.0不能是当前版本)

git fetch origin master (执行后,代码会存在origin/master分子上)

git diff origin/master (查看远程版本到底修改了哪些东西)

git merge origin/master (将origin/master上的修改合并到主分支上)

==== git pull origin master (fetch与pull之间的区别)