git的下载地址：<https://git-scm.com/download/win>

# 本地仓库

1. 配置用户信息

git config --global user.name “ycy”

git config --global user.email [“haohaizi\_yang@163.com”](mailto:\“tyler@163.com\”)

1. 安装完成后在搜索栏输入 git bash，打开命令行后输入上面两个配置命令，通过以下命令查看配置是否成功，如果user.name和user.email都已经在里面了，则配置成功

git config --list

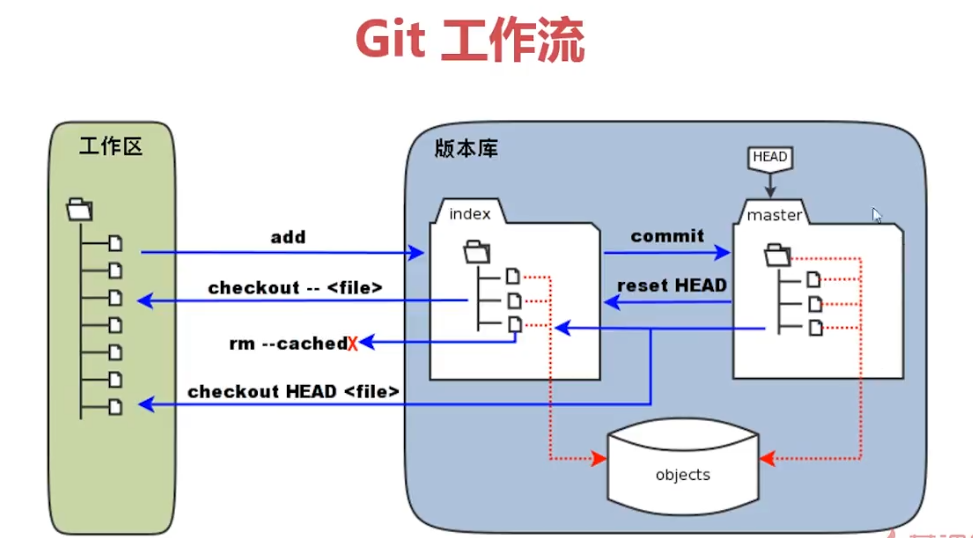
1. 安装客户端工具sourcetree
2. 初始化版本库：git init
3. 添加文件到版本库：git add和git commit
4. 查看仓库状态：git status
5. 演示添加一个文件到本地仓库中：
6. 图形界面模式：
7. 文件->新建->创建新仓库(实际使用的是git init命令)，目标目录选择一个空目录即可，输入名字，点击创建
8. 在左侧新创建的仓库上右键选择在资源管理器里打开，在里面添加文件后就可以在中间**未暂存文件**中看到，选中文件，右键选择添加(实际使用的git add命令)，会将文件从未暂存区域放入**已暂存区域**
9. 点击上方的进行提交，可以在中间下部添加提交注释，点击提交按钮(实际使用的git commit命令)，第一次提交时会默认创建一个master分支，会显示提交时的注释信息，以及本次提交提交了什么文件
10. 命令行模式：
11. 进入代码目录demo（目前为一个空目录即可）使用git init命令初始化版本库，初始化成功后会在本地生成一个隐藏的.git目录
12. 创建test.txt文件，并输入内容“git repo2”(相当于在未暂存区域)通过使用git add test.txt命令将test.txt放入已暂存文件区域
13. 使用git commit -m “repo2 commit” 将代码提交到本地仓库，-m 是用来添加提交说明的。
14. 使用git status来查看状态，会显示当前仓库在哪个分支上，是否都进行了提交等等。
15. 具体操作案例

**通过图形界：**

1. 新建一个仓库，并将demo3当做本地目录，在里面有gui\_demo.txt，文件内容为：需求，通过添加、提交等操作提交到本地仓库。
2. 修改文件后，如果没有测试，则将文件添加到暂存区。如果想回滚到上一个版本，则在暂存区右键文件，选择丢弃，则会回到上一个版本
3. 再次修改进行添加 提交到本地仓库，如果此时想回滚到上一个版本，则在分支上选择需要回滚的版本，右键选择重置当前分支到此次提交；然后将未暂存区域的相应文件进行丢弃操作
4. 如果需要删除文件，则在本地删除文件后，在未暂存区域进行添加，在暂存区域进行提交，则删除完成。

**通过命令行：**

1. 在本地目录创建一个文件bash\_demo.txt输入内容 bash\_demo，使用git status可以查看到有一个文件没有添加到暂存区
2. 使用git add bash\_demo.txt将文件添加到暂存区
3. 使用git commit -m “bash first commit”将文件提交到本地仓库中
4. 本地目录中修改了bash\_demo.txt，使用git add bash\_demo.txt提交到暂存区
5. 使用git reset HEAD bash\_demo.txt将暂存区的文件回滚到工作区
6. 使用git checkout -- bash\_demo.txt将工作区内容进行清理（去掉之前改动的内容，恢复成最初的模样）
7. 修改bash\_demo.txt，使用git add bash\_demo.txt和git commit -m “second commit”
8. 需要回滚到第一次提交时使用命令：git log拿到第一次提交的commit号
9. 使用命令：git reset --hard 第一次提交的commit号，将工作区和仓库都回滚到第一次提交
10. 使用git rm bash\_demo.txt清除本地仓库
11. 使用git commit -m “delete bash demo”将工作区和仓库清理干净



# 远程仓库

1. 创建SSH key(本地仓库和远程仓库进行连接，是通过ssh进行加密)，通过以下命令进行key的创建：ssh-keygen -t rsa -C [“haohaizi\_yang@163.com”](mailto:\“haohaizi_yang@163.com\”)

邮箱必须为GitHub上注册的邮箱

1. 将上述命令在git bash命令上进行执行，根据提示找到key的生成文件id\_rsa.pub，进行内容拷贝
2. 登陆网页github，在用户头像那选择settings，选择左侧SSH and GPG keys，点击右上方的New SSHkey，填写title，然后将步骤①中拷贝的key粘贴到key中点击Add SSH key即可。
3. 测试本地仓库和github的连通性，通过在bash命令行执行：ssh -T [git@github.com，第一次进行ssh连接的时候会有warning，后面再连接就不会有了](mailto:git@github.com，第一次进行ssh连接的时候会有warning，后面再连接就不会有了)
4. 在github上创建远程仓库，点击，选择new repository

**使用命令行方式：**

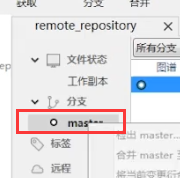
1. 在本地目录中新建一个readme.txt使用git init命令初始化git仓库
2. 使用git add 和 git commit命令将本地文件添加到本地仓库
3. 将本地仓库与远程仓库关联起来，在本地目录中执行bash命令：git remote add origin [git@github.com:haohaizi\_yang@163.com/demo4.git](mailto:git@github.com:haohaizi_yang@163.com/demo4.git)

(蓝色部分为远程仓库的地址)

1. 将本地仓库中的文件推向远程仓库，使用命令：git push -u origin master

(tips:-u 参数是第一次使用git push时用来关联本地的master分支和推送到的origin主机)

**使用sourcetree方式进行**

1. 使用source创建本地仓库，新建一个test.txt，添加到暂存区后，提价到本地仓库中
2. 点击中间的选择创建拉取请求，点击设置，在仓库设置中选择远程仓库，点击添加，将github创建仓库时生成的拷贝到URL/路径中，远端名称默认为origin，用户名是提交的用户名ycy（跟第一步的配置一致），点击确定，如果在master下面出现了远程、origin则表示关联成功
3. 提交文件到远程仓库：右击中间的master选择推送到origin，点击推送即可。

(tips:可以在提交注释的下面勾选推送到origin，这样提交到本地仓库的同时也会提交到远程仓库)

# 克隆仓库

**使用命令行方式**

1. 在github上创建一个远程仓库gitdemo，克隆地址为：<https://github.com/Bypassooo/gitdemo.git>
2. 打开bash命令行，进入到工作目录（必须是空的，不是能有.git的），使用命令

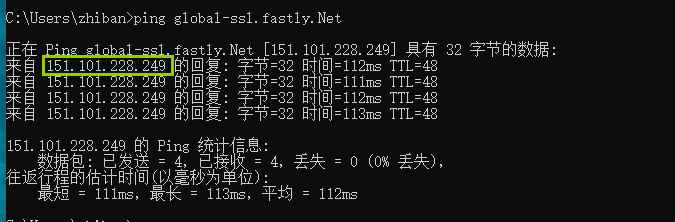
git clone <https://github.com/Bypassooo/gitdemo.git> 克隆成功

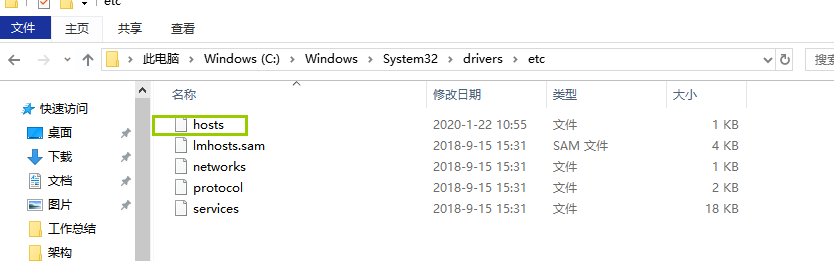
**使用sourcetree的方式**

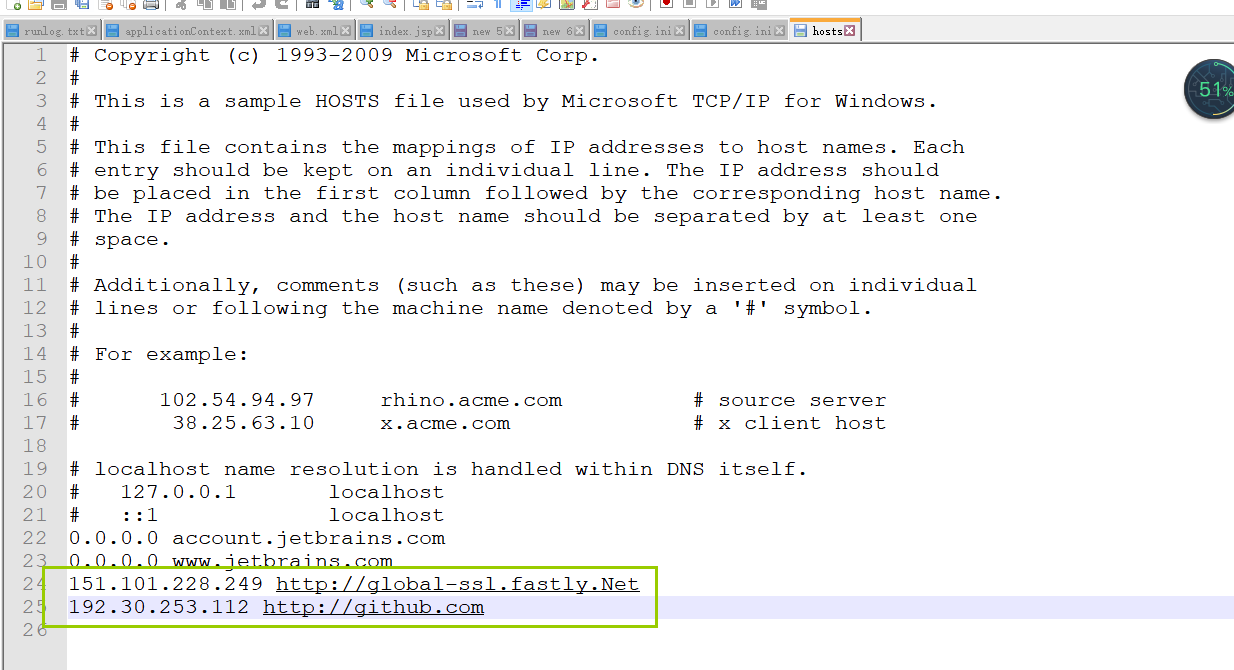
1. 在github上建立仓库
2. 在sourcetree上选择新建->克隆仓库，然后把克隆地址复制到URL中，目标路径选择本地工作目录，点击克隆即可。

git push过慢问题：

ping github.com的跳转地址和下面的跳转地址添加到host文件中







使用以下命令刷新DNS

