

## 1.准备工作

- 下载Git

Windows环境下: <https://git-scm.com/download/win>

- 安装

next到底

- 配置用户名以及邮箱

```
git config --global user.name "Colin"
```

```
git config --global user.email "1036977279@qq.com"
```

- 下载并安装可视化客户端

<https://www.sourcetreeapp.com/>

用bitbucket账号密码登录

## 2.常用命令

命令行模式中:

**pwd** 显示当前在哪个目录下面

**ll** 展示下面所有的文件

**cd ..** 返回到上一级目录, 注意cd和..之间的空格

**mkdir 文件名** 创建文件

**cd 文件名** 进入文件中

**git init** 初始化版本库 (会产生.git隐藏文件)

**ls -a** 展示所有文件 (包括隐藏文件)

**echo " git repo " >>text.txt** 将 git repo追加到text.txt文件中 (如果text.txt文件不存在的话, 就直接创建了这个文件)

**cat text.txt** 展示文件的内容

**git add text.txt** 将文件添加到暂存区

**git commit -m "first commit"** 提交到本地仓库, 并进行备注

**git status** 查看仓库状态

## 3.工作流

在sourcetree中进行工作区到暂存区的提交, 暂存区到本地仓库的提交, 暂存区到本地仓库的回滚, 最新提交分支的删除。然后用命令行模式演练上述过程。

- 暂存区到工作区的回滚----直接丢弃暂存区的文件, 可以回到未修改之前的状态也就是 git add后, 还未git commit到本地仓库。在sourceTree上直接丢弃掉即可; 在git上可以通过

```
git restore --staged bash_demo.txt
```

```
git restore bash_demo.txt
```

或者

```
git reset HEAD bash_demo.txt
```

 将暂存区的修改回滚到工作区

```
git checkout -- bash_demo.txt
```

 将工作区清理干净

- 仓库到暂存区的回滚----选择最近的不需要做回滚的文件，重置到此次文件，并丢弃掉不需要的提交。在sourceTree上先选择需要回滚到的提交处，然后选择重置当前分支到此次的提交，将不需要的部分直接丢弃，最后将丢弃这个操作提交，才算成功回滚；在git上的操作：

```
git log
```

 用git log 找到需要回滚到的提交的 提交号

```
git reset --hard b3fddf541cb6b59a83292b220c998d7991bffe08
```

 重置到序号为b3fddf541cb6b59a83292b220c998d7991bffe08处的提交

```
rm 'bash_demo.txt'
```

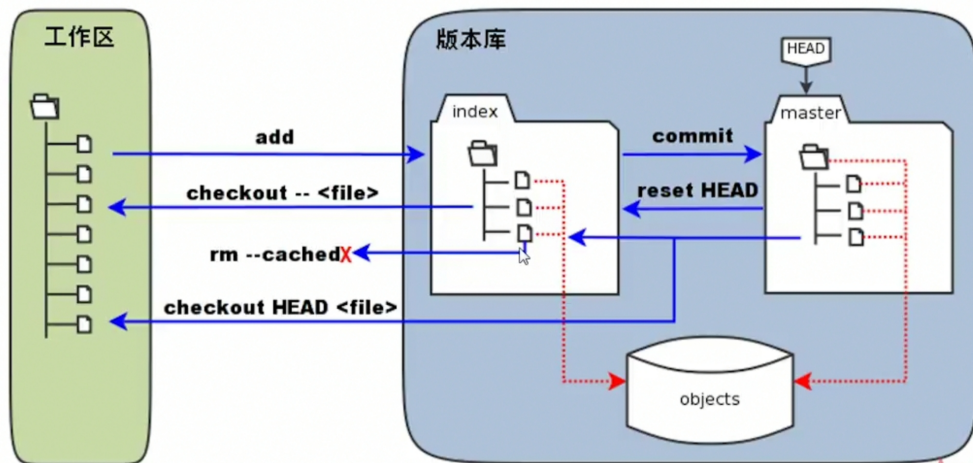
 彻底清理干净

```
git commit -m 'delete demo'
```

 提交此次更改并添加备注。

所有的操作如图所示：

## Git 工作流



慕课网

### 4.远程仓库

- 注册github账号
- 创建SSH key

`ssh-keygen -t rsa -C "youremail@example.com"` 在命令行中用这个命令创建SSH密钥，进入id\_rsa.pub文件中复制所选内容（`cat id_rsa.pub`），然后检查是否与远程仓库连通了（`ssh -T git@github.com`）。

- 添加远程仓库

#### 4.1在bash上的操作

`git remote add origin git@github.com:ColinWangzw/muke_demo.git` 本地仓库与远程仓库关联起来

`git pull origin master --allow-unrelated-histories`

`git push -u origin master` 推到远程仓库

先在github上创建一个repository,设置为public,创建好后code模块有提示如何操作将本地仓库和远程仓库关联起来, 并进行push操作。

`echo "# muke_demo_sourcetree" >> README.md`

`git init`

`git add README.md`

`git commit -m "first commit"`

`git remote add origin`

`https://github.com/ColinWangzw/muke_demo_sourcetree.git`

`git push -u origin master`

此处关联两个仓库时, 用的是

`git remote set-url origin https://github.com/ColinWangzw/muke\_demo.git`

来源于: <https://www.jianshu.com/p/3380ec281729>

然后进行push 操作: `git push -u origin master`

修改重新提交后只需要: `git push`即可。

#### 4.2在SourceTree上的操作

首先新建repository, 设为public。然后再sourceTree上创建本地仓库, 右键master,点击创建拉取请求, 添加远程仓库, 每次更改文件后, 可以直接推到远程仓库。

#### 5.克隆仓库

找到一个没有git本地仓库的路径, 用

`git clone git@github.com:ColinWangZW/muke_demo.git`

更改相关内容后, 直接 `git push`推送到远程仓库

在sourcetree上直接可以克隆, 注意选好仓库的地址即可

#### 6.标签管理

`git tag` 查看已有标签

`git tag name` 创建标签

`git tag -a name -m "commit"` 指定提交信息

`git tag -d name` 删除标签

`git push origin name` 推到远程仓库

在sourcetree上直接进行操作即可，注意先提交更改到本地仓库，然后添加标签

## 7.分支管理

`git branch name` 创建名为name的分支

`git branch` 查看所有分支

`git checkout name` 切换到名为name的分支

`git merge name` 合并分支，保留的分支为当前目录分支，将name分支信息合并到当下

`git branch -d name` 删除分支

在sourcetree上直接用对应的图标来实现。