

## 版本控制系统:VCS,Version control System

1.集中式版本控制系统:svn,subversion(向服务器提交,不能向自身提交)

2.分布式版本控制系统:Git(向服务器提交,也能向自身提交)

svn服务器:taoCode,SVNchina

Git服务器:gitee,github

## Git使用:

- 1 Git安装:
- 2 下载地址:<https://git-scm.com/download/win>

- 1 初始版本化
- 2 `git init`
- 3 注:当前文件夹,有文件要进行版本控制
- 4 (`git:(master)`)

- 1 查看当前版本库的状态
- 2 `git status`

- 1 指定某个文件进行版本控制
- 2 `git add 文件`

- 1 红色:未进行版本控制
- 2 绿色:新增文件
- 3 蓝色:修改文件

- 1 git配置
- 2 `git config --global user.name "姓名"`
- 3 `git config --global user.email "邮箱"`

```
4 git config --global user.email "1971169846@qq.com"
```

```
1 删除git信息配置
2 git config --global --unset user.email
```

```
1 查看git信息配置
2 git config --global --list
3
4 //注:
5 //--system:系统
6 //--global:当前用户
7 //--local:当前仓库
```

```
1 提交修改
2 git commit -m "注释"
```

```
1 和上一个版本进行比较,查看修改文件信息
2 git diff 文件(比如:git diff src/Main.java)
3 注:按q退出查看
4
5 export LANG="zh_CN.UTF-8"
6 export LC_ALL="zh_CN.UTF-8"
```

```
1 查看日志
2 git log
```

```
1 HEAD:当前的指向,最后一次提交
2
3 退回到HEAD的前一个指向
4 git reset --hard HEAD^
5 git reset --hard HEAD~100
```

```
1 查看所有日志
```

## 2 git reflog

- 1 工作区,暂存区,版本库,远程仓库
- 2 工作区:当前编辑的工程
- 3 暂存区:当执行了`git add`后,文件会存入暂存区
- 4 版本库:当执行了`git commit`后,会把暂存区的文件提交到版本库
- 5 工作区,暂存区,版本库都在自己电脑上
- 6 暂存区,版本库,都在`.git`
- 7 远程仓库:`git`服务器
- 8
- 9 撤销修改
- 10 1.工作区:直接修改,或命令:`git checkout -- 文件`
- 11 2.暂存区:直接修改,再次`add`,或命令:`git reset HEAD 文件`
- 12 3.版本库:版本回退
- 13 版本回退
- 14 `git reset --hard 版本号`
- 15 `git reset --hard 8bf68`
- 16 怎样回来:`reflog`

- 1 删除文件:
- 2 `git rm 文件`

## 远程仓库

- 1 添加远程仓库
- 2 `git remote add 仓库名 仓库地址`
- 3 比如:`git remote add github https://github.com/mr-zhl/study.git`
- 4
- 5 把本地仓库推送到远程仓库
- 6 `git push github master`
- 7 `git push gitee master`
- 8
- 9 第一次如果初始失败
- 10 解决方案
- 11 远程仓库下载东西
- 12 `git pull`
- 13 `git pull github master`
- 14 `git pull gitee master`

```
15 git pull github master --allow-unrelated-histories
16 git pull gitee master --allow-unrelated-histories
17
18 查看远程仓库(看本地路径)
19 git remote -v
20
21 产生远程仓库
22 git remote remove 仓库名
23
24 添加第二个远程地址时使用以下命令:
25 git remote set-url --add 仓库名 仓库地址
```

## 工作仓库

### 1.创建远程仓库

### 2.从远程仓库下载代码

### 3.修改代码并提交

## 分支管理

```
1 查看分支
2 git branch
3
4 创建分支
5 git branch 分支名
6
7 切换分支
8 git checkout 分支名
9 创建并切换分支
10 git checkout -b 分支名
```

## 冲突

### 1,手动修改冲突代码

### 2.add.commit

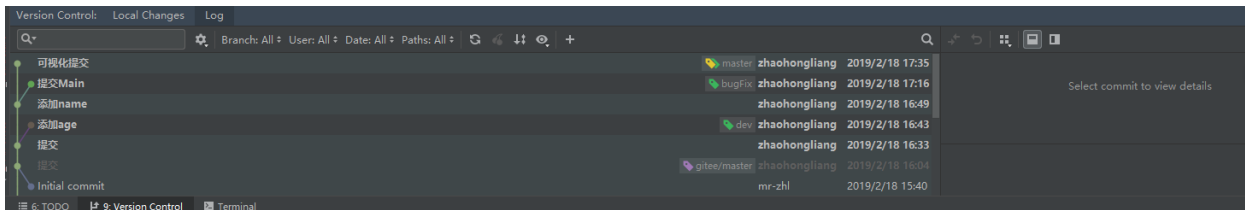
### 3.再合并分支

```
1 把Dev 合并到master中(方法)
2 1.切换到master
3 2.执行git merge dev
4
5 删除分支
6 git branch -d 分支名
7
8 git log --graph --all
```

\*

```
1 //保持当前工作区内容
2 //git stash
3
4 //查看保持内容
5 //git stash list
6
7 //恢复工作区内容
8 //git stash apply
9
10 //删除备份
11 //git stash drop
12
13 //恢复工作区内容,并删除备份
14 //git stash pop
```

### 可视化操作

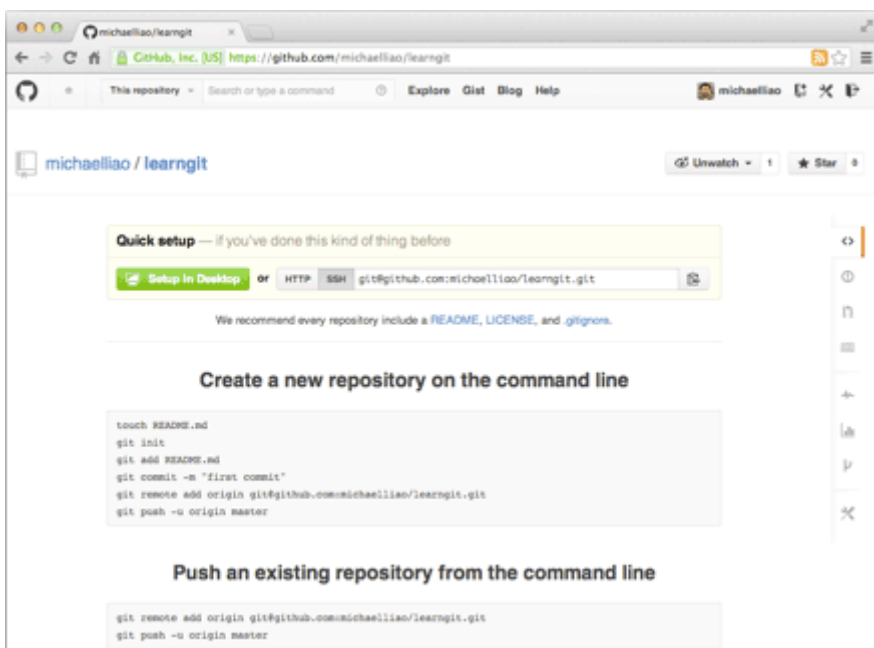
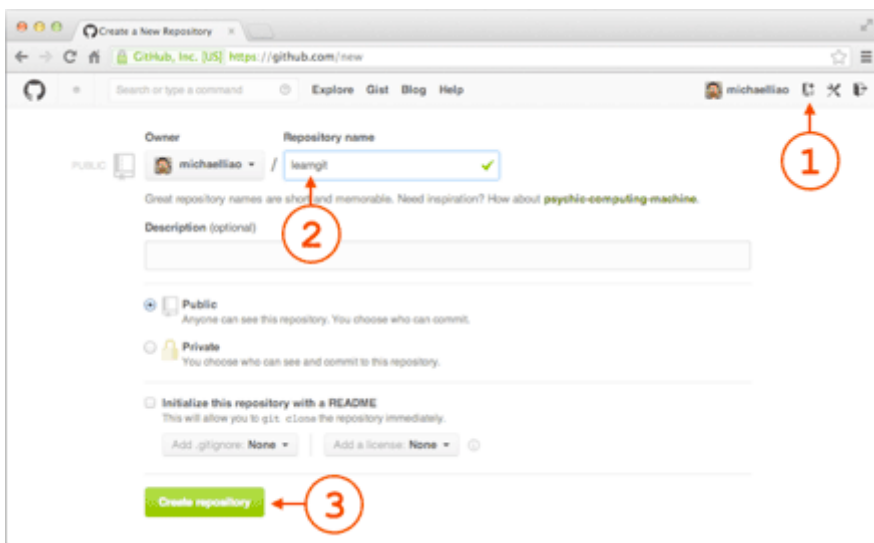


## 练习:如何上传笔记到远程仓库

本地仓库:新建文件夹,存放笔记,01 02 03 04 一个分支名为github

远程仓库:github上新建仓库,登陆GitHub, 然后, 在右上角找到“Create a new repo”按钮, 创建一个新的仓库

下图:



本地仓库如何推送到远程仓库:

```
$ git remote add origin  
git@github.com:michaelliao/learngit.git  
$ git push -u origin master
```

## SSH公钥

### SSH警告

当你第一次使用Git的 `clone` 或者 `push` 命令连接GitHub时，会得到一个警告：

```
The authenticity of host 'github.com (xx.xx.xx.xx)' can't be established.  
RSA key fingerprint is xx.xx.xx.xx.xx.  
Are you sure you want to continue connecting (yes/no)?
```

这是因为Git使用SSH连接，而SSH连接在第一次验证GitHub服务器的Key时，需要你确认GitHub的Key的指纹信息是否真的来自GitHub的服务器，输入 `yes` 回车即可。

Git会输出一个警告，告诉你已经把GitHub的Key添加到本机的一个信任列表里了：

```
Warning: Permanently added 'github.com' (RSA) to the list of known hosts.
```

这个警告只会出现一次，后面的操作就不会有任何警告了。

如果你实在担心有人冒充GitHub服务器，输入 `yes` 前可以对照GitHub的RSA Key的指纹信息是否与SSH连接给出的一致。



```
MINGW64/c:/Users/lanou/Desktop/新建文件夹
lanou@SC-201811011627 MINGW64 ~/Desktop/新建文件夹 (master)
$ git init
Reinitialized existing Git repository in C:/Users/lanou/Desktop/新建文件夹/.git/

lanou@SC-201811011627 MINGW64 ~/Desktop/新建文件夹 (master)
$ git remote -v
bash: git: command not found

lanou@SC-201811011627 MINGW64 ~/Desktop/新建文件夹 (master)
$ git remote -v
origin  git@github.com:mr-zhl/study_Notes.git (fetch)
origin  git@github.com:mr-zhl/study_Notes.git (push)

lanou@SC-201811011627 MINGW64 ~/Desktop/新建文件夹 (master)
$ git add 学习笔记一.txt

lanou@SC-201811011627 MINGW64 ~/Desktop/新建文件夹 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   "\345\255\246\344\271\240\347\254\224\350\256\260\344\270\200.txt"

lanou@SC-201811011627 MINGW64 ~/Desktop/新建文件夹 (master)
$ git commit -m"第一次提交"
[master (root-commit) 3db7e59] 第一次提交
 1 file changed, 30 insertions(+)
 create mode 100644 "\345\255\246\344\271\240\347\254\224\350\256\260\344\270\200.txt"

lanou@SC-201811011627 MINGW64 ~/Desktop/新建文件夹 (master)
```

## 小结

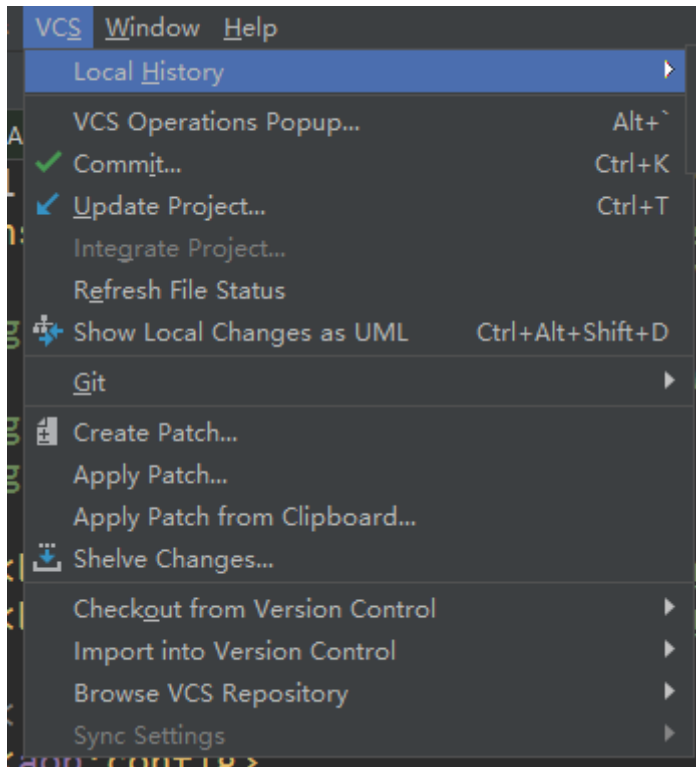
要关联一个远程库，使用命令 `git remote add origin git@server-name:path/repo-name.git`；

关联后，使用命令 `git push -u origin master` 第一次推送master分支的所有内容；

此后，每次本地提交后，只要有必要，就可以使用命令 `git push origin master` 推送最新修改；

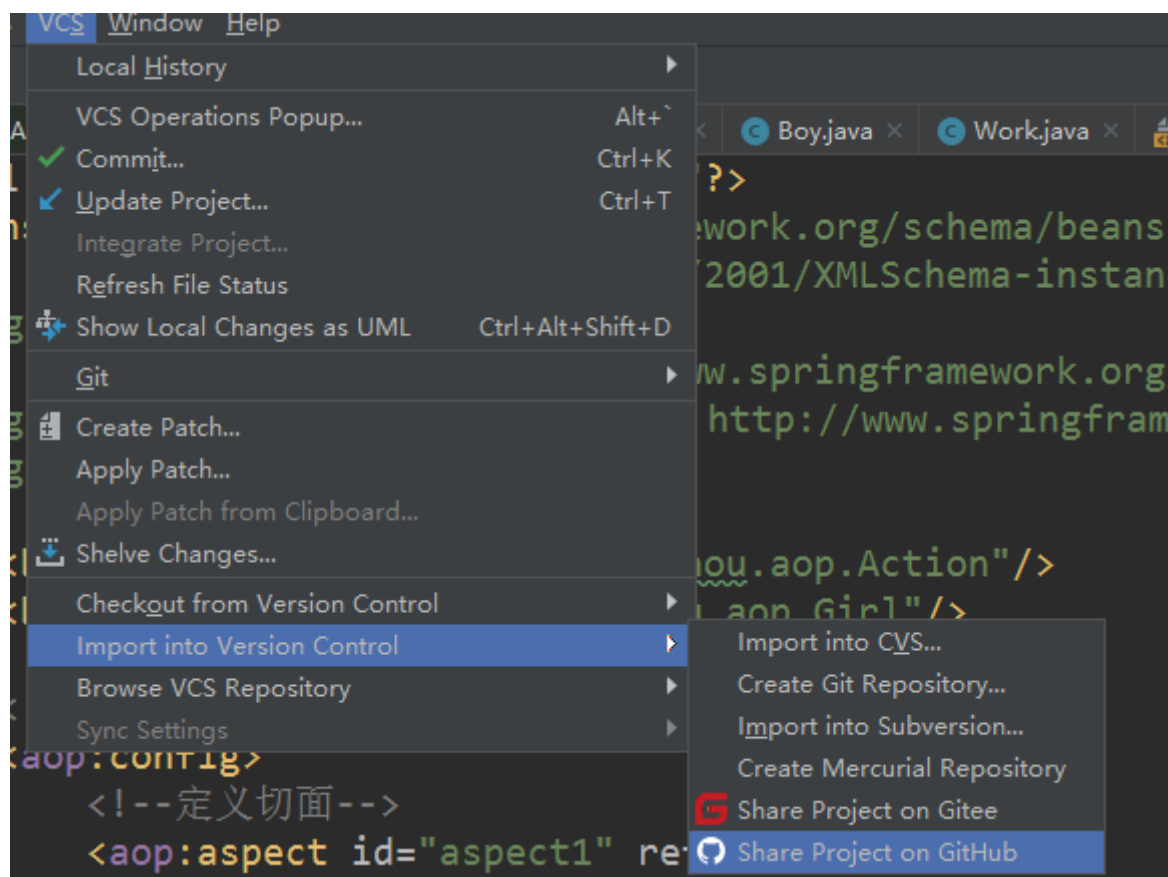
分布式版本系统的最大好处之一是在本地工作完全不需要考虑远程库的存在，也就是有没有联网都可以正常工作，而SVN在没有联网的时候是拒绝干活的！当有网络的时候，再把本地提交推送一下就完成了同步，真是太方便了！

## 练习2:idea中如何推送到GitHub



## init初始化:





提交:

