1.检测用户名是否可用(get /checkuname)，将用户名以get方式传递；

获取get传递的数据

判断是否为空

如果不存在 {code:200,msg:'用户名未被注册'} 否则{code:201,msg:'用户名已注册'}

http://127.0.0.1:8080/v1/users/checkuname?uname=tao

2.检测邮箱是否可用(get /checkemail)，将邮箱以get方式传递

http://127.0.0.1:8080/v1/users/checkemail?email=tao@126.com

3.检测手机号码是否可用(get /checkphone)，将手机以get方式传递

http://127.0.0.1:8080/v1/users/checkphone?phone=18312345678

4.删除用户(delete /编号)

成功{code:200,msg:'删除成功'} 失败{code:201,msg:'删除失败'}

5.用户列表(get /)，以get方式传递当前的页码pno，每页的数据量count

最后响应{code:200,msg:'成功', data: ....}

http://127.0.0.1:8080/v1/users?pno=1&count=4

1.VCS系统

版本控制系统，用于项目的存储、共享、合并、历史回退、代码追踪文件历史等功能

常用的VCS软件

CVS 2000以前

SVN 2010以前

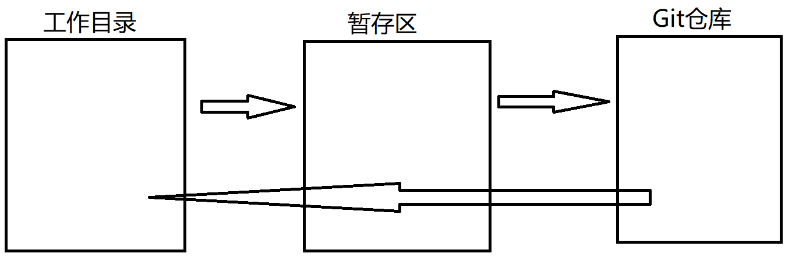
Git 2010至今

2.Git中常用的概念

工作目录：是一个目录，保存项目中所有的文件

暂存区：是内存中的一块区域，用于临时存储项目中的文件

Git仓库：是一个特殊的目录，保存项目中所有的文件以及每次修改的记录



3.Git常用的管理命令

(1)第一次使用git前，告诉git系统你是谁

git config --global user.name "自定义用户名"

git config --global user.email "用户邮箱"

(2)创建Git仓库，使用Git系统来管理当前的项目

git init

会在当前的目录下生成.git目录，保存所有项目文件的历史

(3)查看当前Git系统的状态

git status

(4)将指定的文件添加到暂存区

git add 文件名称

git add . 将所有变化的文件添加到暂存区

(5)将暂存区的所有文件提交到Git仓库

git commit -m "提交内容说明"

(6)查看Git仓库中所有的提交日志

git log

git reflog 查看所有的提交日志、回退记录等..

(7)忽略文件

有些文件或者目录不需要提交到仓库，可以忽略

使用任意的编辑器创建文件.gitignore，把要忽略的文件名称写入到这个文件中

(8)历史回退

git reset --hard 提交的id

回退到第2版，第1版，最后回退到第4版

练习：创建目录tedu作为工作目录，打开Git窗口，初始化一个Git仓库

练习：创建index.html，随便写内容，提交到仓库

练习：创建目录img，添加两种图片，提交到仓库

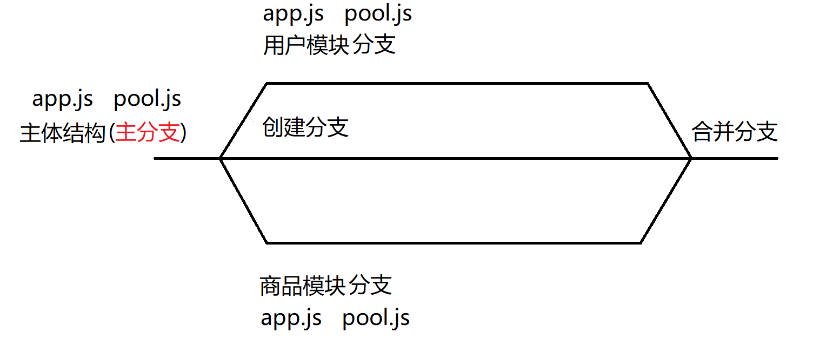
练习：添加忽略文件.gitignore，忽略node\_modules目录，提交到仓库

练习：创建目录css，添加1.css和2.css，随便写内容，提交到仓库

练习：不断回退到第1版，最后回退到最新版。

4.分支

在不影响主线开发的前提下所分离出来的文件，Git系统下默认只有一个主分支(master),最终分支要并入主分支



(1)创建分支

git branch 分支名称

拷贝当前分支的版本作为一个新的分支

(2)查看所有分支

git branch

(3)切换分支

git checkout 分支名称

练习：在主分支下创建商品分支product，切换到商品分支，添加文件product.html，随便写内容，提交到仓库

练习：在主分支下创建购物车分支shopping，切换到分支，添加文件shopping.html，随便写内容，提交到仓库

(4)合并分支

git merge 分支名称

(5)删除分支

git branch -d 分支名称 删除已经合并的分支

git branch -D 分支名称 强制删除分支，不管是否合并

5.远程仓库

码云：国内

GitHub：世界上最大的

先创建远程仓库

将本地的仓库推送到远程仓库

git push 仓库地址 分支名称

|  |
| --- |
| 输入错误的用户名或者密码  控制面板 -> 凭据管理器 -> Windows凭据  删除对应的账号密码 |

git clone 仓库地址 克隆一个远程仓库到本地(没有仓库)

git pull 仓库地址 分支名称 从远程仓库拉取一个分支到本地(已有仓库)