**Git版本管理工具（** 拉代码和提交代码的工具 **）**

拉代码：刚开始工作，项目代码需要从某个服务器(远程仓库)下载到你的电脑上。

提交代码：把你修改好的代码上传到这个服务器(远程仓库)上，更新项目代码，以便别人下载。

上班的至少3种服务器环境：

1：本地环境 （你自己的电脑）在本地环境运行项目。（你自己的电脑跑）

2：测试环境 （请求的地址服务器）

3：生产环境(线上环境)（互联网上的服务器）

常见的代码管理工具

1. Git ：（分布式）版本管理工具。
2. SVN ：（集中式）

仓库：

1. 本地仓库

你自己的电脑就叫 本地仓库。

Git可以从测试环境（远程仓库）下载代码到你的电脑（本地仓库）。(pull)

1. 远程仓库

代码放在某个服务器上，这个服务器就叫 远程仓库

Git可以把你的代码（本地仓库）上传到测试环境（远程仓库）。(push)

**本地仓库 => 你自己的电脑**

**远程仓库 => 老师的电脑**

**Svn => 拉老師每天上課的代碼 （更新）=> 从远程仓库拉代码到本地仓库**

**Svn => 每天交作業 (提交) => 从本地仓库提交代码到远程仓库**

注意：

Git的远程仓库，不同公司的地址是不一样。（上班一定会有人给你仓库地址）

如何用Git创建本地仓库？

可用的远程免费仓库：github，码云。

相关概念：

工作区 => 你操作的文件夹就是工作区

暂存区 => 暂存区你看不见（内存）add操作会把修改存入暂存区

本地仓库 => commit 就是把暂存区的修改提交到本地仓库。（.git文件夹内）

远程仓库（代码托管中心）push 把本地仓库的文件上传到远程仓库

**1：如何拉取代码**

A：第一次拉代码 , git clone 远程仓库地址

B：更新代码，git pull

**2：如何上传代码**

第一步：git add 文件或文件夹

（缓存）（提交到暂存区）（git add . 就是add所有修改）

第二步：git commit 文件或文件夹 -m 备注信息

（提交到本地仓库）（commit时不填任何文件名，就是全部提交）

git commit -m 备注 -> 提交全部修改的文件.

第三步：git pull

（提交前需要更新一次代码，防止覆盖或冲突）

第四步：git push

（真正的上传，把本地仓库内的代码提交到远程仓库）

**3: ssh公钥免密设置 （github）**

**第一步：ssh-keygen –t rsa –C 你的github注册邮箱**

第二步：在C:\Users\Administrator\.ssh里面有一个id\_rsa.pub.打开它复制里面的所有内容。

第三步：打开github网站，登录，右侧settings => SSH and GPG key => 新建SSH => tittle随便填，key就粘贴。

id\_rsa.pub.文件内复制的内容，最后点生成。

第四步：设置仓库地址别名：git remote add 别名 仓库地址 (git remote –v 查看别名的列表)。

第五步：手动push到指定别名：git push 指定别名 指定的分支。

**4: 分支的使用. (项目的副本) (为了项目安全或者减少冲突)**

为了保证安全性，开发一般都是在分支进行，而不是在主分支上进行。

master或者main就是默认的主分支名。在master分支下的所有操作都是在修改master分支上的项目文件.

1：新建分支

git branch 新分支名

2：切换分支

git checkout 分支名 （切换分支后，之后的操作都是在操作当前分支）

3：合并

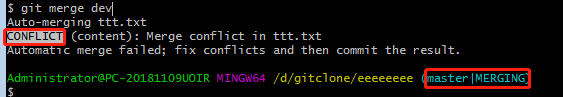
git merge 某个分支名 (合并之后,需要再次add和commit等操作)

**5: 冲突**

**有时候需要查看当前的仓库状态：git status**

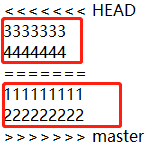
合并的过程中有可能有冲突，需要跟组员商量到底怎么解决冲突。

手动解决冲突后，需要再add，再commit才能合并完成。



冲突后,文件自动变成类似下面的样子.你需要手动修改,保留你要保留的内容.

然后再add,再commit,再push



**4．面试题**

你们项目是如何分分支的? (按什么划分分支的)

1: 按功能分分支. (例如,注册登录用户中心,可以单独作为一个分支开发)。

2: 按版本划分. (多人分配好页面，只要开发时遵守好就可以不冲突)。

3: 按人头分. (不太靠谱)

你们项目是如何处理冲突的

出现了冲突，就跟相关人员了解原因，商量解决办法，然后手动修复冲突后再次提交。

查看当前状态：git status

可视化工具：git小乌龟，sourcetree，gitlab（一个类似于github远程服务器的客户端）

工具插件类的东西都可以问 => 我以前没用过.