1. Git的基本概念和使用

Git是什么？

Git：是一个开源的分布式版本控制系统，可以有效、高速的处理从很小到非常大的项目版本管理

GitHab: 全球最大的面向开源及私有软件项目的托管平台，免费注册并且可以免费托管开源代码

GitLab：与GitHab类似，也是属于第三方基于Git开发的产品，不同的是：GitLab可以部署在自己的服务器上，代码数据都是掌握在自己的手中，适合公司内部团队开发。

1. Git和SVN的区别

SVN版本集中管理，所有的代码都在一台服务器上；分支和主线是不一样的路径；

Git分布式管理，去中心化，服务器和每个开发人员都拥有一个本地的代码管理仓库；分支和主线路径是一样的。

注意：本地仓库，远程仓库

总结：Git比SVN更强大，大部分公司都在用

1. 公司都是如何使用Git的？
2. 安装好git工具；

Git ---- 命令行模式操作

TortoiseGit ----- 图形化界面操作

1. 公司会有一个服务器（多个仓库，一般项目<-->仓库），针对不同的人账号有不同的权限

首先，你需要有一个Git的账号，公司的所有文件或者代码都放到Git仓库中国（仓库地址），需要有账号才能克隆相应的分支下来，得到所有的文件或者代码

然后，你也可以把本地推动到远程仓库中（账号有权限）

1. 分支（Branch）和主线（Master）

主线（master）：主树干，一般放稳定的代码（最终的文档）

分支（branch）：树杈，每人（组）维护一个分支，等稳定后合并到主线

使用分支意味着你可以把你的工作从开发主线上分离开来，以免影响开发主线。比如：每个人都有代码/文档提交到一个公共的服务器，避免相互影响，自己提交到自己的分支里，由专门的人来合并（merge）到主干分支。

1. Git的常用操作
2. Git克隆：URL+本地目录，获取远端的仓库地址，并且有对应的账号权限，克隆代码/文档到本地
3. Git拉取：远端仓库里的文件更新了，下拉到本地：Git拉取--->选择分支--->确认；点击拉取到的差异，可以查看本次的变动

* 拉取（pull）：可以自动进行分支的合并，用的比较多
* 获取（fetch）：不可以自动合并，用的比较少

1. Git推送

本地增加了文件/代码，需要推动到远端仓库

* 添加（add）：添加到本地缓存区
* 提交（commit）：提交到本地库，需要添加日志，（此时远端GitHub是没有文件的）
* 推送（push）：推送到远端服务器。（远端仓库有看本文件）
* Git的删除：先删除本地的文档

1. 提交：删除的本地库里的文件（远端文件还在）
2. 推送：删除远端仓库的文件（远端文件不存在了）

* Git的撤回：误操作后，再提交之前，可以revert

1. 兼容性测试结束
2. 软件+硬件平台兼容性：系统（Windows macOS，Android，iOS）
3. 软件+软件 之间的兼容性：依赖第三方的软件：web应用浏览器兼容性
4. 不同软件版本之间的兼容：APP升级测试--老数据
5. 浏览器兼容性测试结束

产生浏览器兼容性问题的原因：

因为不同浏览器使用内核及所支持的HTML（标准通用标记语言下的一个应用）等网页语言标准不同;以及用户客户端的环境不同（如分辨率不同）造成的显示效果不能达到理想效果。最常见的问题就是网页元素位置混乱，错位，重叠

预见所有客户可能遇到浏览器。-兼容性-web应用

内核：决定了浏览器如何显示网页的内容以及页面的格式信息

1. 常见的浏览器都有哪些？你们常用的有哪些？
2. 选择浏览器做兼容性测试的原则：
3. 一般兼容性测试是怎么来做的？