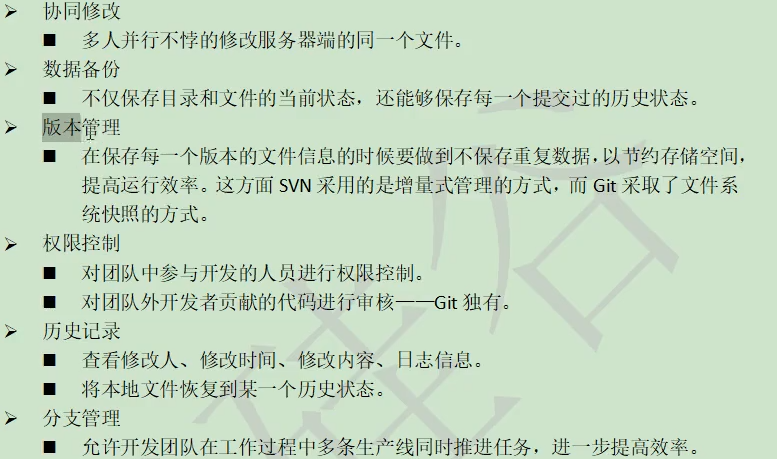
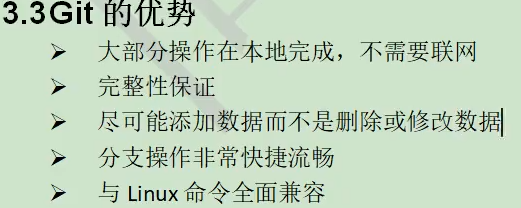
### Git&GitHub

1, 版本控制工具需要具有的功能



2, Git的优势



3, 安装时是否修改path环境变量



4, 初始化本地库: git init

5, windows 中以 .开头的资源都是隐藏的资源

6, 设置签名



7, 查看文档, (猫一眼) : cat

8, 当前系统用户: ~/

9, git签名的系统级别保存路径: cat ~/.gitconfig

10, 查看工作区暂存区状态: git status

11, 添加到暂存区(cached): git add fileName

12, 从暂存区(cached)移除某个文件: git rm --cached fileName

13, 查看历史记录: git log, 简洁化显示: git log --pretty=oneline, git reflog , 过去的历史记录和现在的都显示, git log --oneline 只显示过去的历史记录

14, 使用less 翻页命令

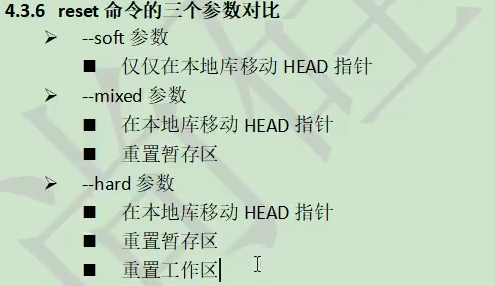
1. : 空格 向下翻页
2. : b 向上翻页
3. : q 退出

15, 基于索引值的版本前进和后退: git reset --hard 哈希值

16, 使用^符号: git reset --hard HEAD^^^^ : 一个＾　表示后退一步，n个表示n步

17, 使用~符号: git reset --hard HEAD~5 : 后面加上要后退的步数

18, reset参数命令区别



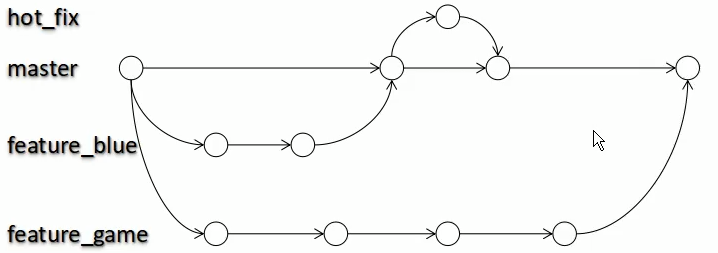
19, 查看帮助文档 :git help reset

20, 文件比较:　git diff [文件名] 将工作区中的文件跟暂存区中的文件进行比较

跟历史记录中某一个历史版本比较: git diff HEAD^ [文件名]

不指定步数和文件, 就会比较所有历史文件: git diff HEAD

21, 分支, 使用多条线路推进项目



22, 查看所有分支: git branch -v

23, 创建一个分支: git branch hot\_fix

24, 切换分支: git checkout hot\_fix

25, 合并到主分支时, 必须在master分支上进行合并:

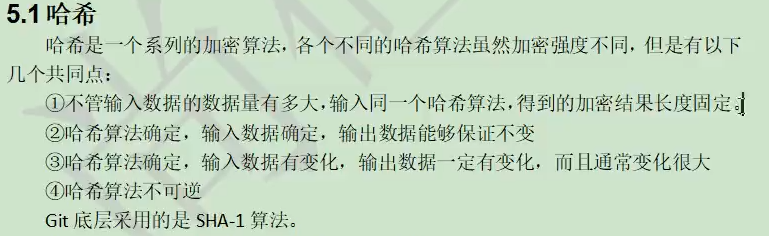
先: git checkout master

然后: git merge hot\_fix

26, 标记为已解决冲突: git add [文件名]

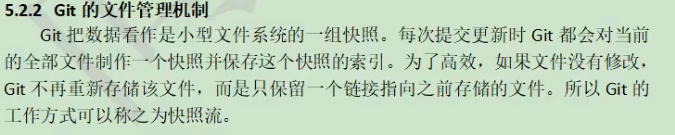
27, 如果解决标记为已解决, 则还需要提交, 注意不要加文件名: git commit -m “resolve confilice”

28, Git 基本原理, 哈希算法



29, SVN只是保存修改的那一部分, 对服务器的空间比较友好

Git:



30, 查看地址别名: git remote -v

31, 给地址起别名: git remote add [别名] [http/ssh地址]

32, 推送: git push [别名或地址] [分支名称]

33, 克隆: git clone [克隆地址], 会自动初始化仓库, 会自动创建origin别名

34, 从远程库中抓取(对和pull对远程库来说都是读操作, 不用登陆):

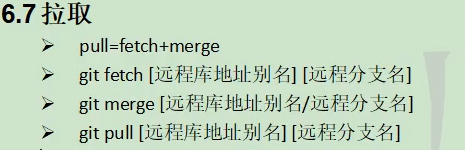
git fetch [origin /地址别名] [master/分支名称]

35, 切换到从远程库中抓取的仓库: git checkout origin/master , 参数跟上面的解释一样

36, 把远程的分支合并到本地: git merge origin/master

37, pull 相当于: fetch + merge

38, 拉取: git pull origin master



39, 团队协作: 在网页中Fork 一份到自己的仓库中, 然后在网页中 pull request

40, 生成SSH密钥, 参数-C 是大大写的C: ssh-keygen -t rsa -C [2388948195@qq.com](mailto:2388948195@qq.com)

41, Git 忽略文件网址: <https://github.com/github/gitignore>

42, 配置Git的全局忽略文件:

全局文件路径: C:/Users/Administrator/

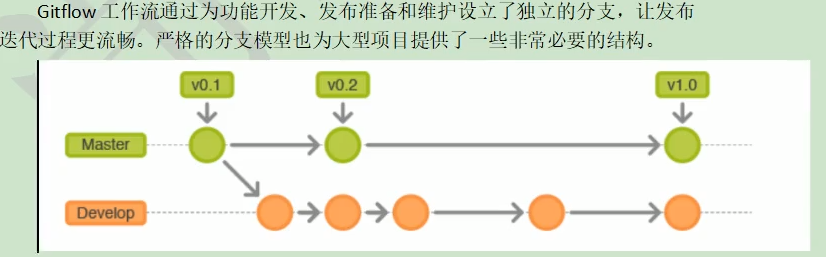
忽略文件路径: C:/Users/Administrator/java.gitignore

|  |
| --- |
| [user]  name = tom\_global  email = tom\_global@review.com  [core]  excludesfile = C:/Users/Administrator/java.gitignore |

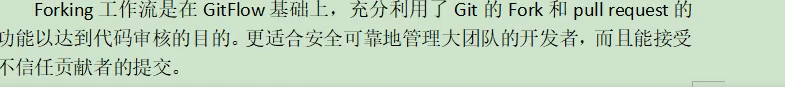
43, 查看暂存区文件: git ls-files

44, Git工作流

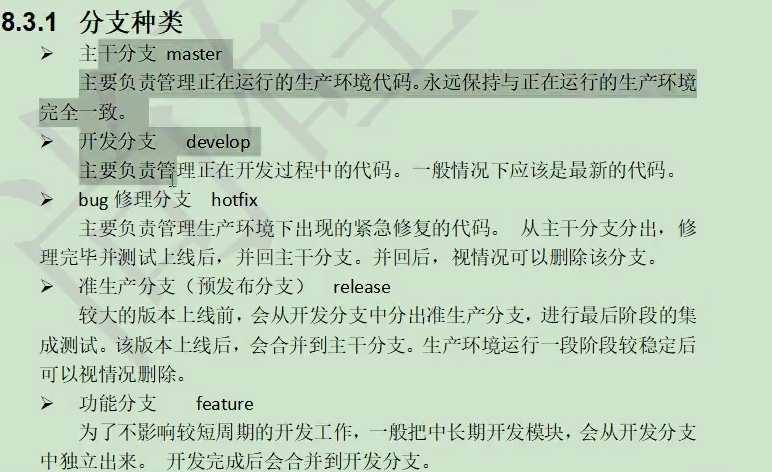
1. 集中式工作流, 类似SVN
2. Gitflow



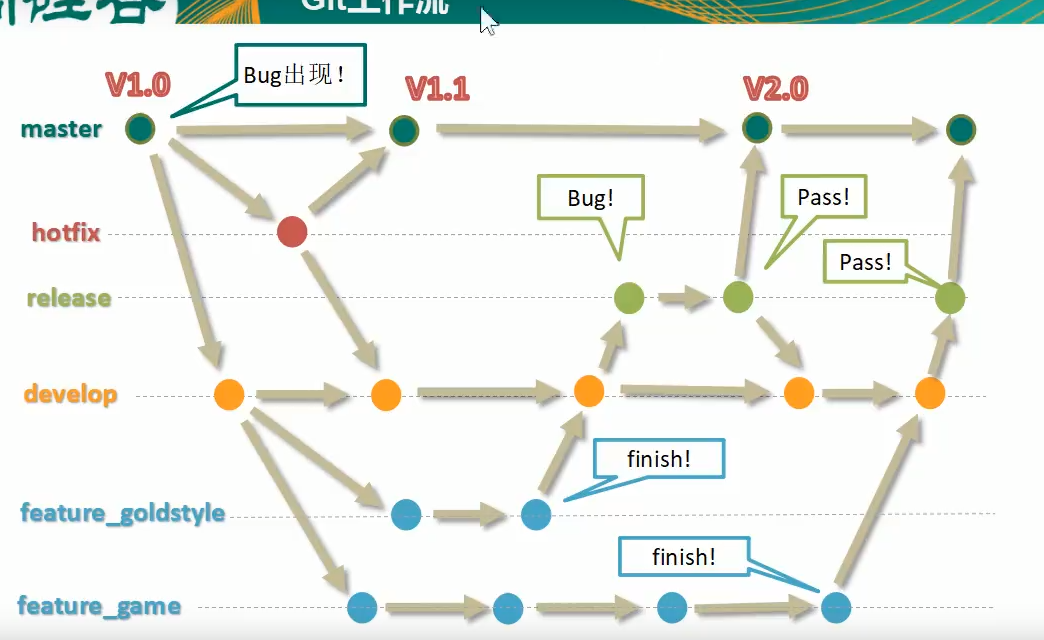
1. Forking



45, 分支种类



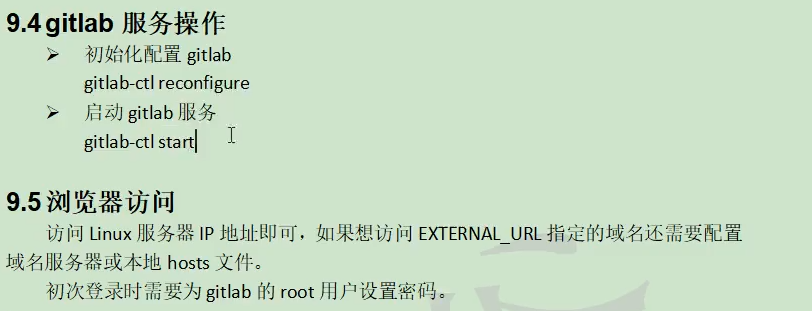
常见分支设计



46, 安装GitLab

安装后需要, 执行9.4的操作

停止gitlab服务: gitlab-ctl stop



如果直接访问linux主机ip不能访问, 需要关闭防火墙

47, 记录从官网的安装

centOS8 官网安装Gitlab文档地址: <https://about.gitlab.com/install/#centos-8>

|  |
| --- |
| RPM下载: <https://packages.gitlab.com/gitlab/>  https://packages.gitlab.com/gitlab/gitlab-ce/packages/el/8/gitlab-ce-13.7.1-ce.0.el8.x86\_64.rpm |
| 安装视频地址: <https://www.bilibili.com/video/BV1pW411A7a5?p=58>  文件实在太大: 808mb |
| 下方的安装过程网络不好不安装  sudo dnf install -y curl policycoreutils openssh-server  sudo systemctl enable sshd  sudo systemctl start sshd  # Check if opening the firewall is needed with: sudo systemctl status firewalld  sudo firewall-cmd --permanent --add-service=http  sudo firewall-cmd --permanent --add-service=https  sudo systemctl reload firewalld  sudo dnf install postfix  sudo systemctl enable postfix  sudo systemctl start postfix  curl https://packages.gitlab.com/install/repositories/gitlab/gitlab-ee/script.rpm.sh | sudo bash  sudo EXTERNAL\_URL="https://gitlab.example.com" dnf install -y gitlab-ee |