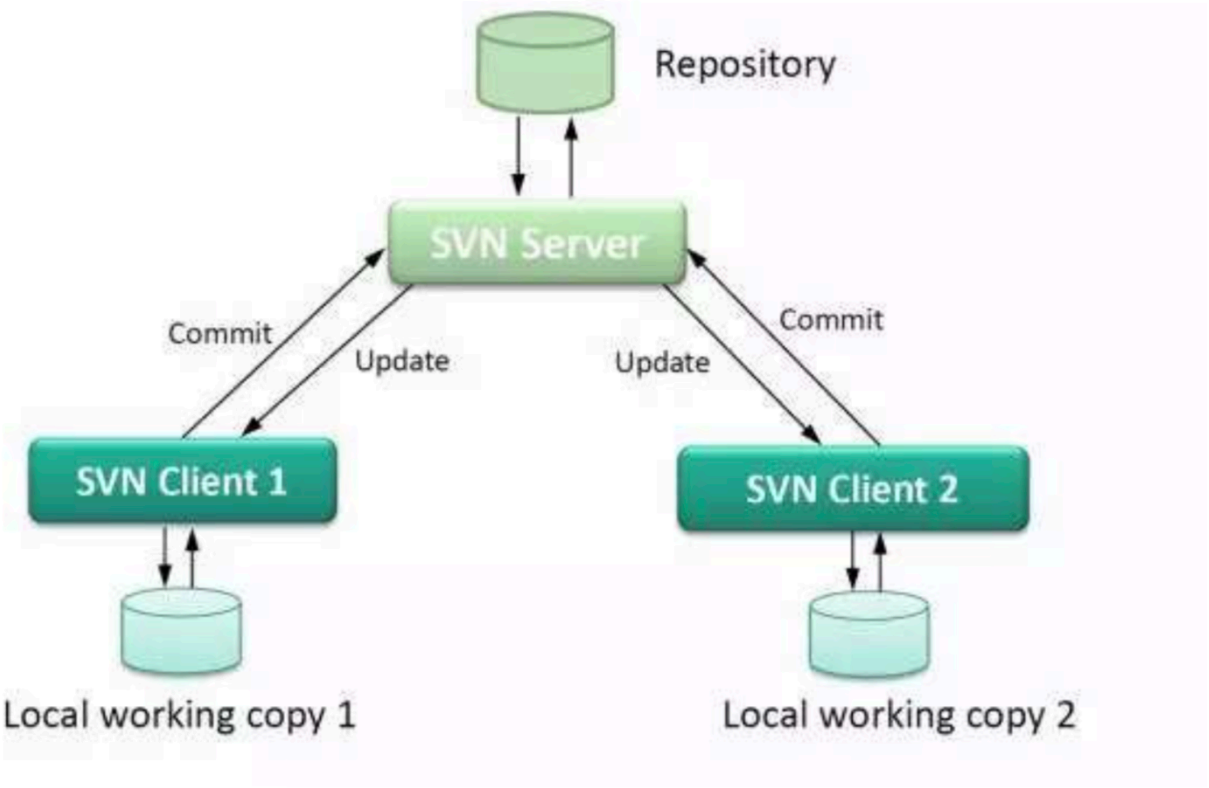
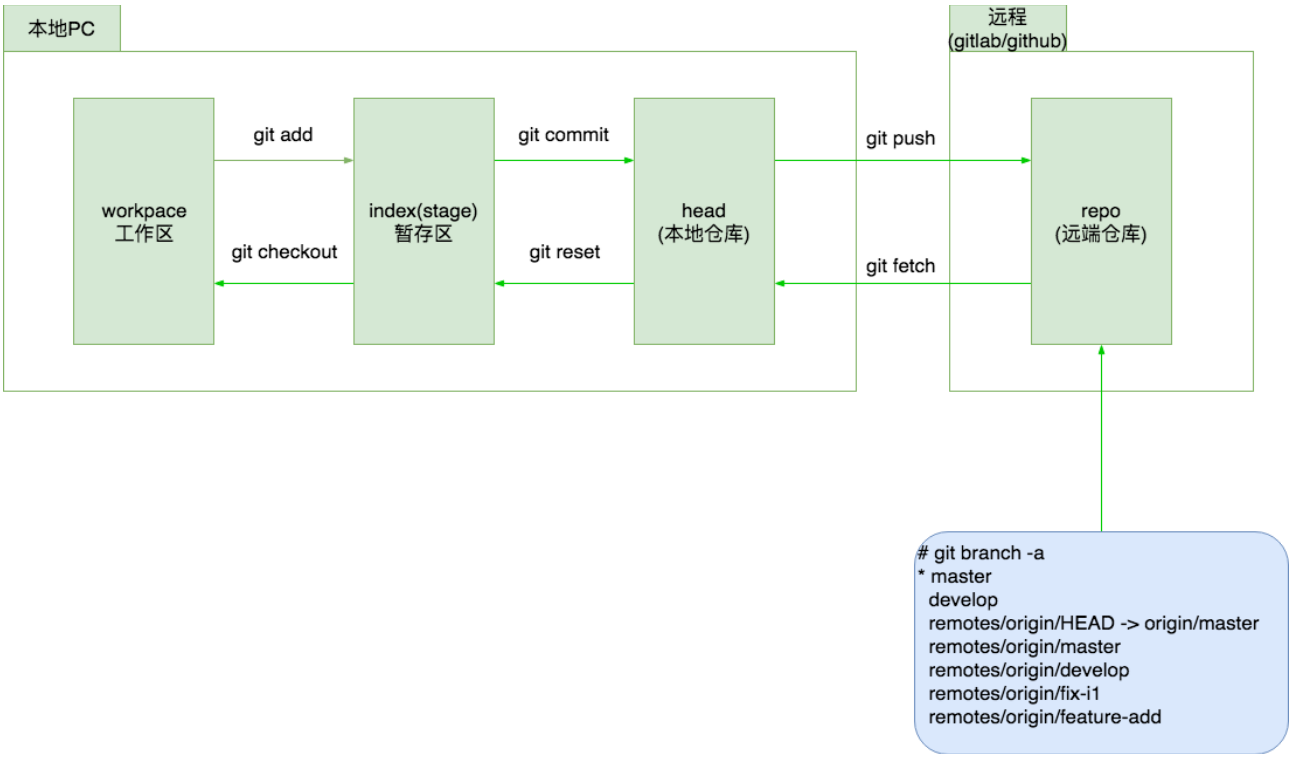
Git操作手册

一. 分布式仓库

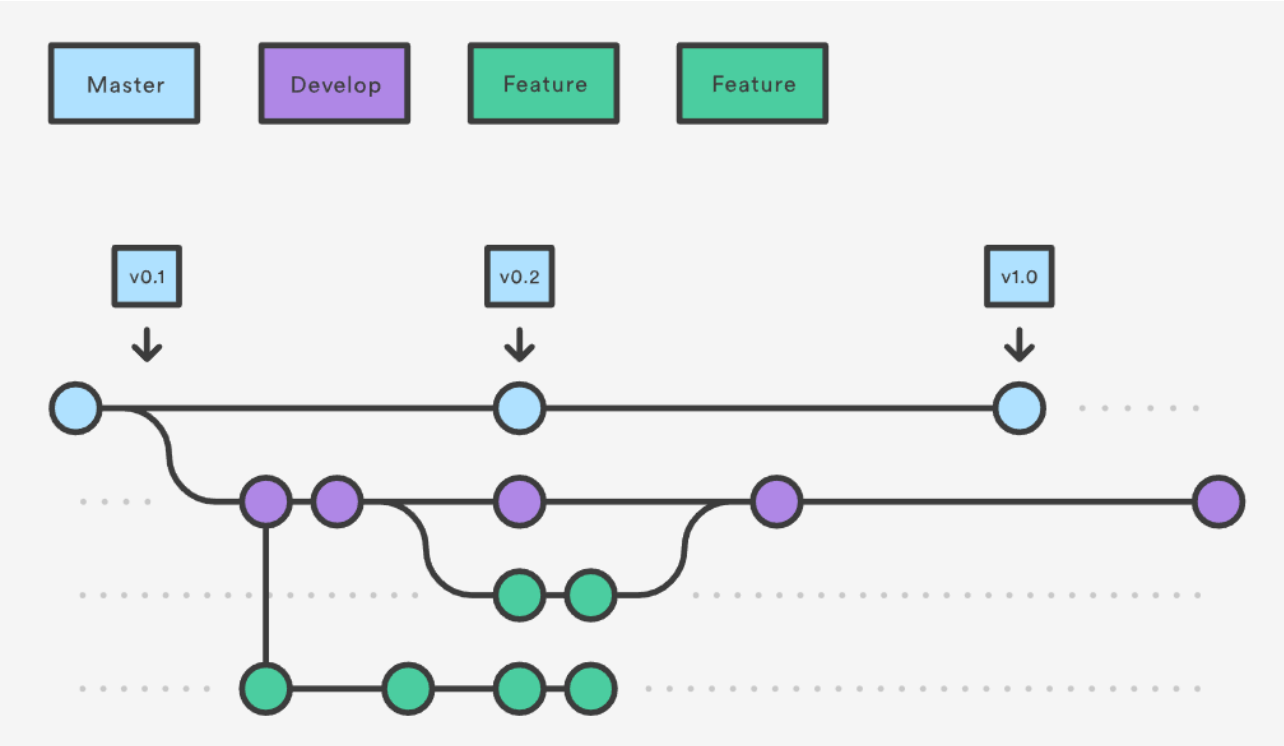
SVN:



Git:



二. 分支

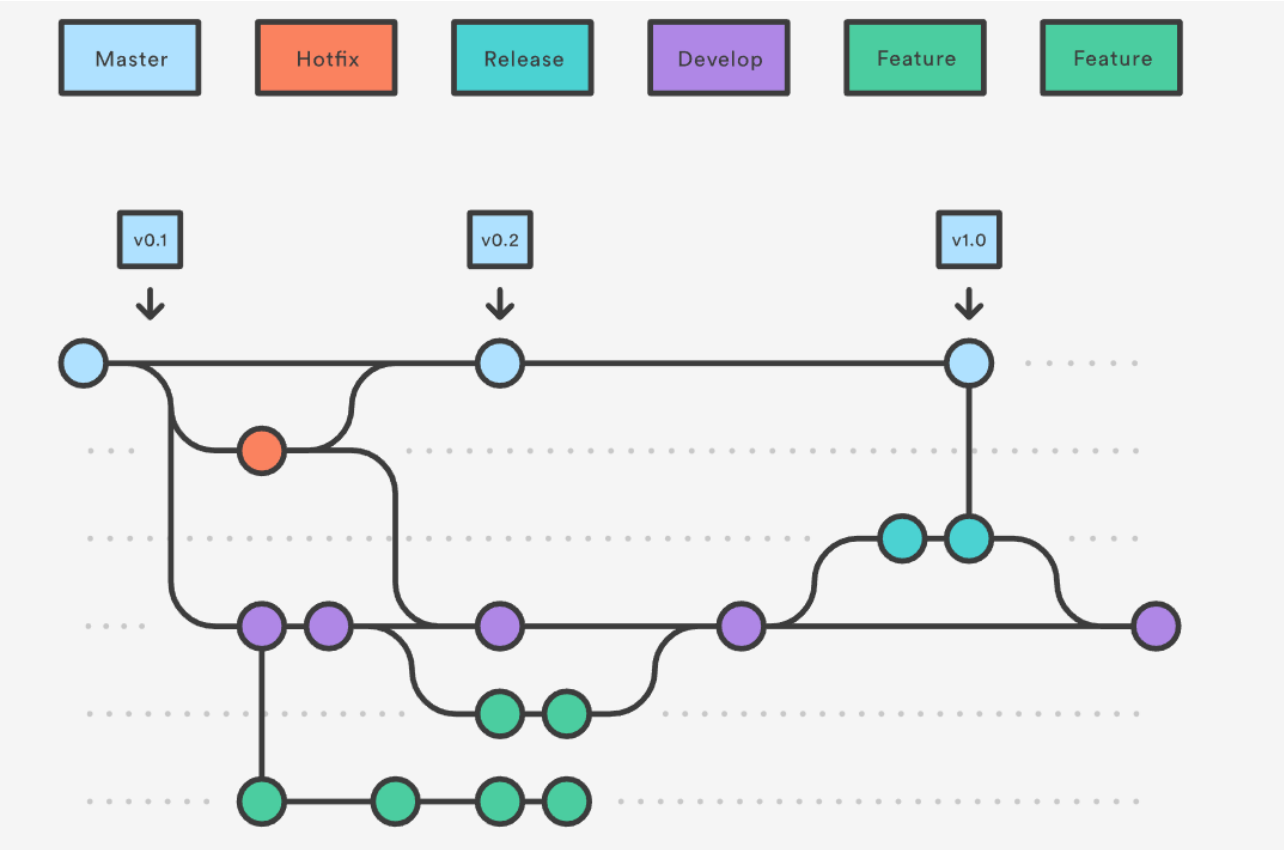


- 新建分支: git checkout -b develop
- 切换分支: git checkout (-b) develop
- 追踪远程分支: git checkout -t origin/develop
- 合并分支: git merge develop (—no-ff)
- 删除本地分支: git branch -d develop
- 查看分支: git branch -a
- 推送至远程分支: git push origin develop:develop
- 删除远程分支: git push origin :develop
- 拉取远程分支: git fetch origin develop
- 拉取远程分支并合并到本地分支: git pull origin develop:develop

相关资料:

- https://git-scm.com/book/zh/v1/Git-%E5%88%86%E6%94%AF-%E8%BF%9C%E7%A8%8B%E5%88%86%E6%94%AF
- https://learngitbranching.js.org/?NODEMO

三. 工作流(git flow)



1. **新建功能特性开发任务**
   - git checkout develop — 切换至develop分支
   - git pull origin develop — 更新develop分支最新代码
   - git checkout -b <特性分支名> — 新建功能特性分支
   - 代码开发
   - git pull origin develop — 更新其他开发者提交的代码(注意: 如无特殊情况,建议每天最少更新一次,否则长期过期的特性分支在合并时很难解决冲突)
   - git push origin <特性分支名> — 推送功能特性分支到远程仓库(只有在多人开发相同功能时使用)
   - git pull origin <特性分支名> — 更新功能特性分支(只有在多人开发相同功能时使用)

2. **完成功能特性开发**
   - git checkout develop — 切换至develop分支
   - git pull origin develop — 更新develop分支代码
   - git merge <特性分支名> — 合并功能特性分支到develop分支
   - git push origin develop
   - git log --pretty=format:"%Cred%h%Creset -%C(yellow)%d%Creset %cn | %s %Cgreen(%cr)%Creset" — 获取当前提交commitId

3. 发布正式环境

- git checkout master — 切换至master分支
- git pull origin master — 更新master分支代码
- git merge develop — 合并develop分支到master分支
- git push origin master— 推送master分支和tag到远程仓库

4. 修复线上bug(hotfix)

- git checkout master — 切换至master分支
- git pull origin master — 更新master分支代码
- git checkout <tag> — 切换至线上版本的tag
- git checkout -b hotfix-<bug代号> — 创建对应hotfix分支
- 修复bug
- 走上述流程(合并到master及develop分支)

相关资料:

- https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow
- https://blog.csdn.net/zsm180/article/details/75291260

四. 注意事项

1. 无特殊情况,尽可能的频繁执行git pull origin develop更新其他同学开发的代码,尽早处理冲突,特别是长线分支或重构工作.长期过期的分支最终合并代码时会导致过多冲突难以解决
2. 使用git push/pull命令时尽量把所有参数带上(如: git push origin develop),使用默认形式(如: git push)稍不注意会容易导致非预期情况需要回退.
3. 使用git merge合并分支或git pull拉取远端分支并合并时需要时刻注意当前head是否是你所希望被合并的
4. 合并分支或切换分支前尽量先使用清空暂存区
5. 谨慎使用git reset —hard命令,该提交之后的所有提交将会丢失.可用git rever代替