

git工具的学习报告

笔记本： My Notebook

创建时间： 2022/8/5 23:22

更新时间： 2022/8/6 0:20

作者： 赵凌珂

git介绍：

a、集中式版本控制工具

集中式版本控制工具，版本库是集中存放在中央服务器的，每个人工作的时候都需要从中央服务器下载代码，必须联网才能工作。个人修改后然后提交到中央版本库。

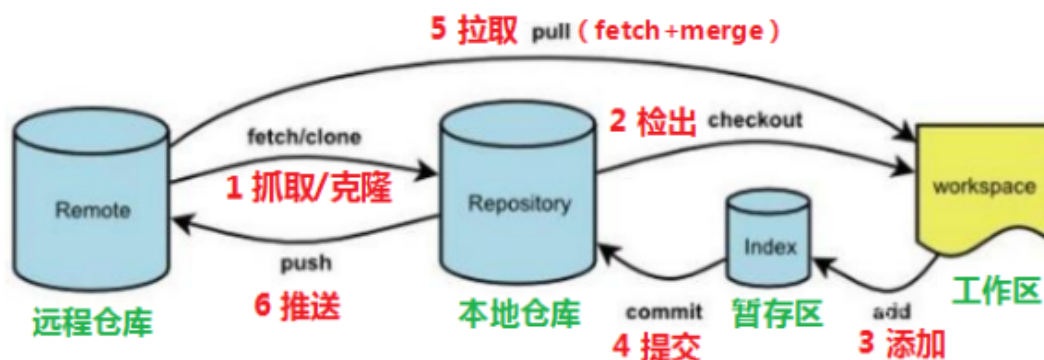
举例：SVN和CVS

b、分布式版本控制工具

分布式版本控制系统没有“中央服务器”，每个人的电脑上都是一个完整的版本库，这样工作的时候，就无需要联网了。多人协作只需要各自的修改推送给对方，就能互相看到对方的修改了。

举例：Git

工作流程：



命令如下：

1. clone (克隆)：从远程仓库中克隆代码到本地仓库
2. checkout (检出)：从本地仓库中检出一个仓库分支然后进行修订
3. add (添加)：在提交前先将代码提交到暂存区
4. commit (提交)：提交到本地仓库。本地仓库中保存修改的各个历史版本
5. fetch (抓取)：从远程库，抓取到本地仓库，不进行任何的合并动作，一般操作比较少。
6. pull (拉取)：从远程库拉到本地库，自动进行合并(merge)，然后放到到工作区，相当于 `fetch+merge`
7. push (推送)：修改完成后，需要和团队成员共享代码时，将代码推送到远程仓库

#用于输出git提交日志

```
alias git-log='git log --pretty=oneline --all --graph --abbrev-commit'
```

#用于输出当前目录所有文件及基本信息

```
alias ll='ls -al'
```

git的使用：

1.一开始要进行配置：

```
git config --global user.name "itcast"
```

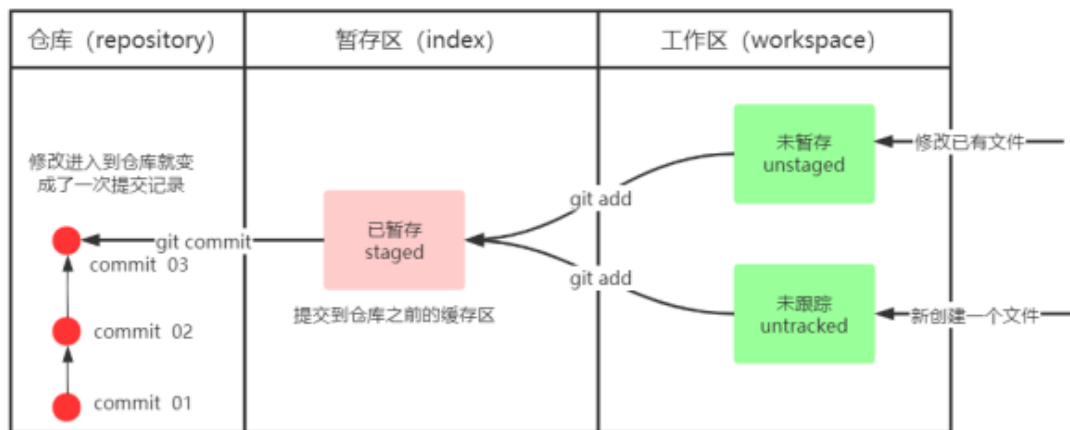
```
git config --global user.email "hello@itcast.cn"
```

2.获得本地仓库：

```
git init
```

可以看到.git文件

3.基础操作指令：



1. git add (工作区 --> 暂存区)

git add . 是传入全部文件

2. git commit (暂存区 --> 本地仓库)

git commit -m '注释内容'

3.git status 查看修改状态

4.git log 查看提交日志

git-log (git log --pretty=oneline --all --graph --abbrev-commit)

命令形式：git log [option]

options

--all 显示所有分支

--pretty=oneline 将提交信息显示为一行

--abbrev-commit 使得输出的commitId更简短

--graph 以图的形式显示

5.版本切换：git reset --hard commitID

git reflog查看已经删除的记录

git的分支：

查看本地分支：git branch

创建本地分支：git branch 分支名

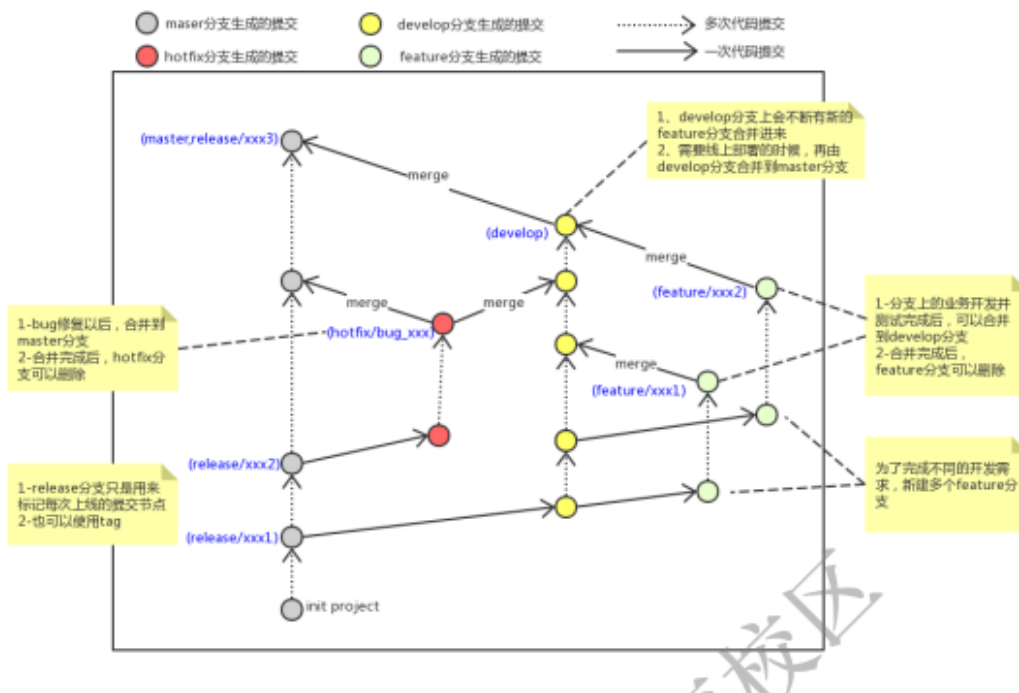
切换分支(checkout)：git checkout 分支名

我们还可以直接切换到一个不存在的分支（创建并切换）：git checkout -b 分支名

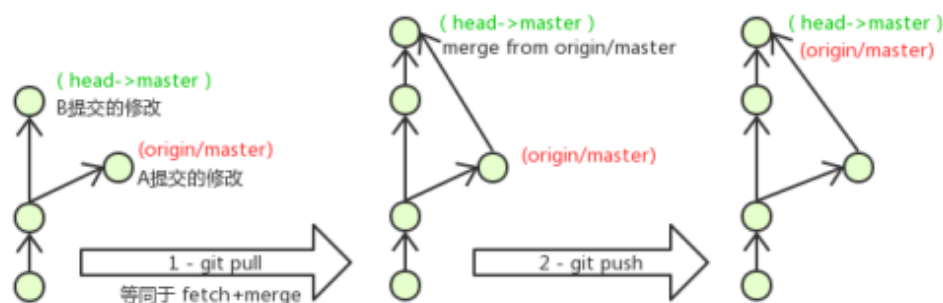
合并分支(merge) 一个分支上的提交可以合并到另一个分支：git merge 分支名称

删除分支git branch -d b1

删除分支时，需要做各种检查 git branch -D b1 不做任何检查，强制删除



冲突：



远程仓库：

添加：`git remote add <远端名称> <仓库路径>`

查看：`git remote`

推送：`git push [-f] [--set-upstream] [远端名称 [本地分支名][:远端分支名]]`

如果远程分支名和本地分支名称相同，则可以只写本地分支 `git push origin master`

抓取 命令：`git fetch [remote name] [branch name]`

抓取指令就是将仓库里的更新都抓取到本地，不会进行合并如果不指定远端名称和分支名，则抓取所有分支。

拉取 命令：`git pull [remote name] [branch name]`

拉取指令就是将远端仓库的修改拉到本地并自动进行合并，等同于 `fetch+merge` 如果不指定远端名称和分支名，则抓取所有并更新当前分支。