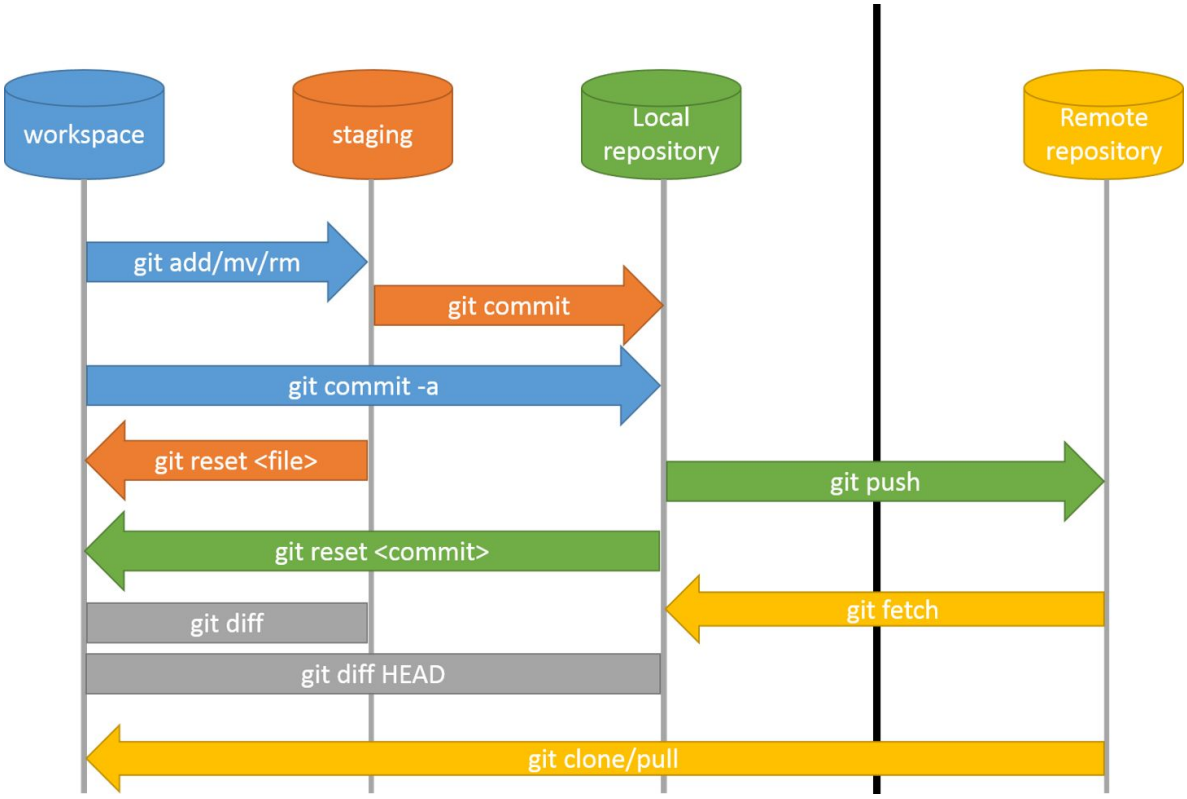
git 从零解读

git 是一个先进的分布式版本控制系统
仓库（版本库）：分为本地仓库和远程仓库，简单的理解就是2个文件夹（包含一个 .git(文件夹)），
.git目录（文件夹）：是Git来跟踪管理版本库的，git只能跟踪文本文件的改动
本地仓库与远程仓库：其实就是2个文件夹，分别放着一些东西



workspace：工作区

staging：暂存区

local repository：本地仓库

remote repository：远程仓库

初始化一个本地仓库

- git init　　//创建一个空的Git仓库或重新初始化一个现有仓库

```
git init
```

- git clone //克隆远程仓库代码

```
git clone <版本库的网址> <本地目录名>
```

git clone 克隆远程指定分支

```
git clone -b <指定分支名> <远程仓库地址>
```

git add 将工作区的修改提交到暂存区

```
git add　　//提交指定修改文件到暂存区
git add .　　　　　　//提交所有修改到暂存区
```

git commit 将暂存区的文件提交到本地仓库

> git commit -m "修改说明" //将暂存区文件提交到本地仓库

本地分支管理 git branch /git checkout

- **git branch 一般用于分支的操作，比如创建分支，查看分支**

> git branch //查看本地分子,在当前分支前 有一个*标记
>
> git branch -a // 查看本地及远程所有分支
>
> git branch newbranch //创建新的名为:newbranch 的分支,但是还是在当前分支
>
> git branch -d newbranch //删除分支:newbranch 但是在分支中有一些未merge的提交,会删除失败,可以用 git branch -D newbranch 进行强制删除
>
> git branch -vv //查看本地对应的远程分支
>
> git branch -m oldBranchName newBranchName //给分支重命名

- git checkout //操作分支,操作文件

> 操作文件
>
> git checkout filename //放弃单个文件的修改
>
> git checkout . //放弃当前文件下的修改

> 操作分支
>
> git checkout master //将分支切换到主分支
>
> git checkout -b Dev //将分支切换到Dev,如果Dev不存在则创建一个Dev分支再切换到Dev

git status

> 用于显示工作目录和暂存区的状态。使用此命令能看到那些修改被暂存到了, 哪些没有, 哪些文件没有被Git tracked到。`git status`不显示已经`commit`到项目历史中去的信息,看项目历史的信息要使用`git log`

git pull

git clone

git fetch

git merge