**git 特点**

1. 直接记录快照 而非差异比较

2. 近乎所有操作都是本地执行

3. 时刻保持数据完整性

4. 多数操作仅添加数据

5. 文件的三种状态

已修改（modified）

已暂存（staged）

已提交（committed）

Git远程操作核心

本地分支

远程分支

远程分支

什么时候会创建远程跟踪分支

当用户将本地分支 push到远程仓库时会创建;

push时本地分支与远程跟踪分支之间是独立的

**注意点：**

只有当本地分支 与 远程跟踪分支之间 建立了一对一关系之后

才可以直接使用git push命令将提交推到远程仓库

才可以直接使用git pull命令将远程仓库的修改同步到本地

**高层命令(远程操作)**

1. git remote -v : 查看当前仓库拥有的别名

2. git remote add 别名 远程仓库的地址 : 给当前仓库新增别名

3. git remote rename 老名 新名 : 对别名进行重命名

4. git remote rm name : 删除别名

8&9前提:本地分支 要和 远程分支 有一一对应的关系

5. git branch -vv : 查看当前仓库本地分支与远程分支的对应关系

6. git clone url : 克隆仓库

7. 当克隆项目时;克隆完成后的项目它的master 与 远程的master分支自动建立了一一对应的关系

所有其他情况都不会建立一一对应的关系;如何处理除master分支外其他分支

a. git checkout 分支名 (本地已经存在了对应的远程跟踪分支 别名/分支名 : git pull / git fetch 别名)

b. git checkout --track 远程跟踪分支 (本地已经存在了对应的远程跟踪分支 别名/分支名 : git pull / git fetch 别名)

a和b做了三件事: 新建分支 创建联系 拉取数据

c. git branch 分支名

git checkout 分支名

git branch -u 远程跟踪分支

git merge 远程跟踪分支

8.git push : 将对应本地分支的内容提交到远程

9.git pull : 将对应远程分支的内容拉取到本地

当git pull无法使用时;可以使用git fetch 别名 + git merge 远程跟踪分支