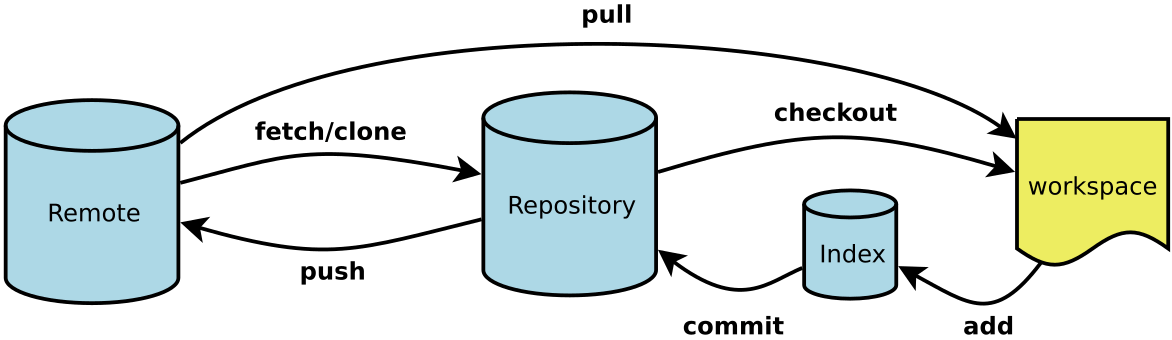
# 工作区、暂存区、版本库、远程仓库

## 一、概念

## 1、四个工作区域

Git本地有四个工作区域：工作目录（Working Directory）、暂存区(Stage/Index)、资源库(Repository或Git Directory)、git仓库(Remote Directory)。文件在这四个区域之间的转换关系如下：



**Workspace**： 本地工作区，就是你平时存放项目代码的地方，电脑上能看到的目录即工作目录，比如：/e/git\_repo/

**Index / Stage**： 本地暂存区，用于临时存放你的改动，事实上它只是一个文件，保存即将提交到文件列表信息，暂存区就是版本库中的一个区域。

**Repository**： 本地仓库区（或版本库），就是安全存放数据的位置，这里面有你提交到所有版本的数据。其中HEAD指向最新放入仓库的版本；工作区有一个隐藏目录.git，这个不算工作区，而是git的版本库。git的版本库里存了很多东西，其中最重要的就是称为stage（或者叫index）的暂存区，还有git为我们自动创建的第一个分支master，以及指向master的一个指针叫HEAD。

**Remote**： 远程仓库，托管代码的服务器，可以简单的认为是你项目组中的一台电脑用于远程数据交换

## 2、工作流程

git的工作流程一般是这样的：

１、在工作目录中添加、修改文件；

２、将需要进行版本管理的文件放入暂存区域；

３、将暂存区域的文件提交到git仓库。

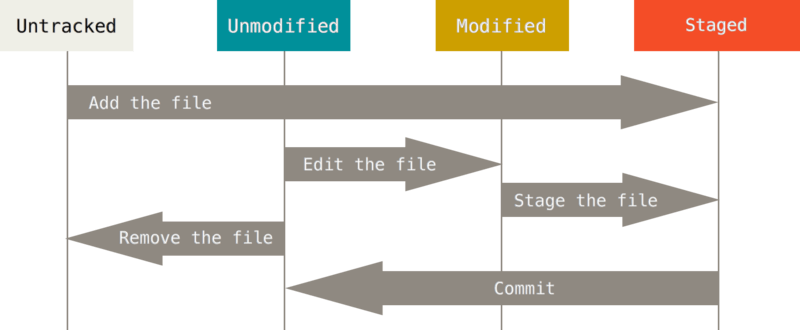
因此，git管理的文件有三种状态：已修改（modified）,已暂存（staged）,已提交(committed)

## 3、文件的四种状态

版本控制就是对文件的版本控制，要对文件进行修改、提交等操作，首先要知道文件当前在什么状态，不然可能会提交了现在还不想提交的文件，或者要提交的文件没提交上。

GIT不关心文件两个版本之间的具体差别，而是关心文件的整体是否有改变，若文件被改变，在添加提交时就生成文件新版本的快照，而判断文件整体是否改变的方法就是用

SHA-1算法计算文件的校验和。



**Untracked:**   未跟踪, 此文件在文件夹中, 但并没有加入到git库, 不参与版本控制. 通过git add 状态变为Staged.

**Unmodify:**   文件已经入库, 未修改, 即版本库中的文件快照内容与文件夹中完全一致. 这种类型的文件有两种去处, 如果它被修改, 而变为Modified.

                   如果使用git rm移出版本库, 则成为Untracked文件

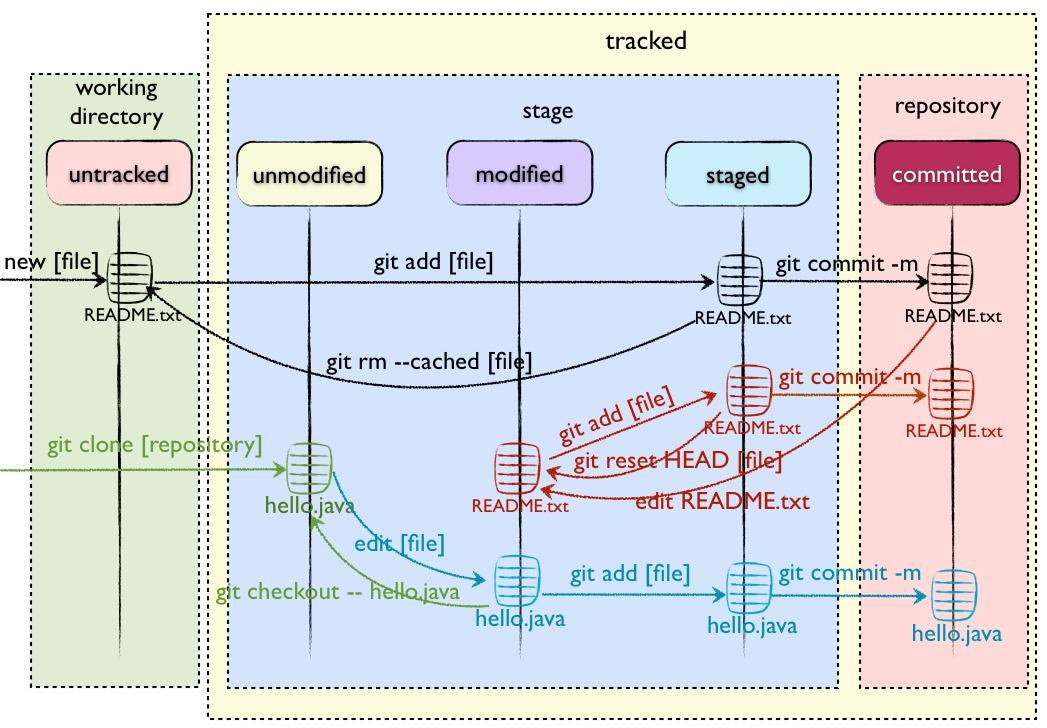
**Modified:** 文件已修改, 仅仅是修改, 并没有进行其他的操作. 这个文件也有两个去处, 通过git add可进入暂存staged状态, 使用git checkout 则丢弃修改过,

                返回到unmodify状态, 这个git checkout即从库中取出文件, 覆盖当前修改

**Staged:** 暂存状态. 执行git commit则将修改同步到库中, 这时库中的文件和本地文件又变为一致, 文件为Unmodify状态. 执行git reset HEAD filename取消暂存,

               文件状态为Modified

 下面的图很好的解释了这四种状态的转变：



新建文件--->Untracked

使用add命令将新建的文件加入到暂存区--->Staged

使用commit命令将暂存区的文件提交到本地仓库--->Unmodified

如果对Unmodified状态的文件进行修改---> modified

如果对Unmodified状态的文件进行remove操作--->Untracked

## 4 、分支(Fork)

分支（branch）有什么用呢？假设你准备开发一个新功能，但是需要两周才能完成，第一周你写了50%的代码，如果立刻提交，由于代码还没写完，不完整的代码库会导致别人不能干活了。如果等代码全部写完再一次提交，又存在丢失每天进度的巨大风险。  
现在有了分支，就不用怕了。你创建了一个属于你自己的分支，别人看不到，还继续在原来的分支上正常工作，而你在自己的分支上干活，想提交就提交，直到开发完毕后，再一次性合并到原来的分支上，这样，既安全，又不影响别人工作。

## 4、Git工作状态

Commit只会记录在本地计算机上做的更改，不会标记远程存储库的更改

NOTE: 一个branch（分支）只有在被合并后才可以被删除。

### 1、同步

提交和同步执行三件事。首先，它将提交。其次，它将执行拉取(从远程存储库中获取更新的信息)。最后，它将合并和远程仓库的最新代码。

### 2、提取(Fetch)

获取最新版代码，从远程获取最新版本文件到本地，不自动合并/merge，最新版本在“分支”选项中的“remotes/origin”文件夹下可以查看，可以选择将其合并到master分支上。使用“提取/fetch”更安全，在merge之前可以看清楚更新情况再决定是否合并。

### 3、拉取（Pull）

获取最新版代码，从远程仓库拉取最新版本文件到本地，自动合并/merge。

### 4、推送

将完成提交并将其推送到远程存储库。这意味着您所做的任何更改也将保存到远程存储库中。