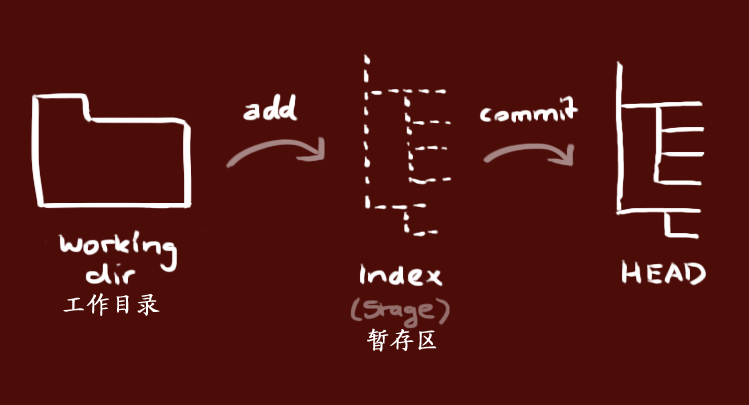
Git学习笔记

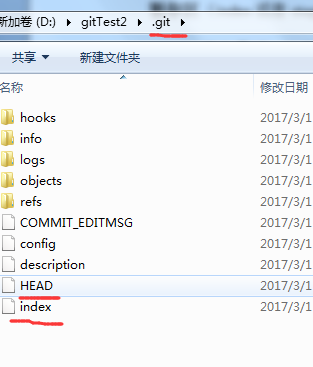
一.Git工作流

1.在自建的**Git本地仓库中**，有三个区域：**本地目录**、**暂存区**、**HEAD**。



**工作目录**，就是受Git控制的文件夹，所有被跟踪记录的文件都包括在其中；  
  
**暂存区**（index或者stage），类似一个缓存区域，**临时保存你做的改动**；  
  
**HEAD**，指向最近一次提交后的结果。

工作目录：

暂存区（index）和HEAD：

**2.项目中的文件**来说，分为**未跟踪**（untracked）和**已跟踪**（tracked）两种状态。

1）**未跟踪**的文件的改动不受到Git的版本控制。

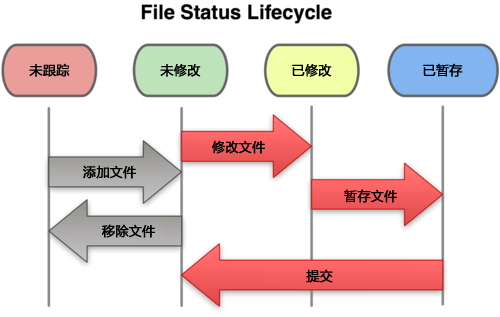
2）**已跟踪**的文件（被纳入版本控制的文件）又分为：

（1）未修改（unmodified）、

（2）已修改（modified）、

（3）已暂存（staged）

三种状态。



3.工作流分析

1）当在工作目录中**新加入一个文件**时，它**处于未跟踪状态**，这表示其没有纳入Git的版本控制。

通过 **git add** 命令可以将其**加入跟踪，**并同时**放入暂存区**。

也就是说，暂存区中有这个文件。此时即是**已暂存**状态，同时也是**已跟踪**和**未修改**状态  
 2）一个**已经被跟踪**的文件，如果没有做过新的修改，就是**未修改**状态。一旦对其做了改动，就变成了已修改状态。

通过 **git add** 命令可以**将已修改**的文件**放入暂存区**。

3）初次克隆某个仓库时，工作目录中所有文件都是**已跟踪**且**未修改**的状态。

4）**git commit** 命令会将**暂存区**中的文件**提交至**HEAD**所指向的**分支。（HEAD 文件中的内容其实只是包含了一个索引信息，并且，这个索引将总是指向你的项目中的当前开发分支。）

\* 一个叫 HEAD 的文件，我们现在来查看一下它的内容：

|  |  |
| --- | --- |
| 1 | $cat.git/HEAD |

现在 HEAD 的内容应该是这样：

|  |  |
| --- | --- |
| 1 | ref:refs/heads/master |

当被commit之后，被提交的暂存区的文件将回到**未修改**状态（依然还在暂存区中）。

5）另外，也可以通过 git rm 将**已经跟踪的文件**从Git中移除。

6）git status 命令是用来查看仓库中文件的状态。

4.crossingit看到第六点，未完待续。。。