```
已修改 (modified) 修改了文件,但尚未提交到数据库
                                                 文件处于三种状态
                                                                   已暂存(staged) 对已修改文件的当前版本做出了标识并将其加入下一次要提交的快照中
                                                                     工作目录 — 暂存修改 — 暂存区
                             Chapter1 入门
                                                 Git项目三个主要区域
                                                                     暂存区 ____ 提交 ___ Git目录 (仓库)
                                                                     Git目录 (仓库) — 检出项目 — 工作目录
                                                                                     /etc/gitconfig文件: 针对系统中的所有用户及其仓库的值
                                                 配置变量 (每一级会覆盖上一级的设置)
                                                                                     ~/.gitconfig或~/.config/git/config文件: 针对个人用户
                                                                                     当前仓库的Git目录中的config文件: 针对单个仓库
                                                                                                 git init
                                                                     对现有目录中初始化的Git仓库
                                                                                                git add [filename]
                                                                                                  git commit -m 'initial project version'
                                                   2.1 获取Git仓库
                                                                                    git clone [url] (克隆至当前目录)
                                                                    克隆现有仓库
                                                                                    git clone [url] dirname (克隆至dirname目录)
                                                                                                                                            已修改
                                                                           Untracked Unmodified Modified Staged
                                                                                                     已跟踪 (tracked) : 上一次快照中包含的文件
                                                                                Edit the file Stage the file
                                                                            Retove the file
                                                                                                                                            已暂存
                                                                                 文件状态
                                                                                                      未跟踪 (untracked) : 不在上一次快照中,也不在暂存区的文件
                                                                                                                                                        工作目录中出去已跟踪文件之外的所有文件
                                                                           查看当前文件状态 — git status
                                                                                         git add [filename]
                                                                           跟踪新文件
                                                                                         git add [dirname] —— 递归添加该目录下的所有文件
                                                                           暂存已修改的文件 —— 已跟踪的文件被更改 —— 使用git add 添加到暂存区(只有暂存区的文件会被提交到本地数据库)
                                                                                                                                  未被跟踪的文件??
                                                                                                                                  已暂存的新文件 A
                                                                           显示更简洁的状态信息
                                                                                                  git status -s或git status --short
                                                                                                                                  已修改的文件 M
                                                                                                                                  注释:文件列表的标记是被分为两列的,左列表明了文件是否已暂存,而右列表明了文件是否已修改
                                                                                                                                            空行或者以#开始的行会被忽略
                                                                                                                                                                                 doc/*.txt
                                                                                                                                                                                 忽略doc目录的.pdf文件
                                                                                                                                             支持标准的glob模式(简化版正则表达式)
                                                                                                                                                                                 doc/**/*.pdf
                                                                                                                                                                                 忽略doc目录下的所有.pdf文件
                                                                           忽略文件 — 创建.gitignore文件 — 写入.gitignore文件的匹配模式的规则
                                                                                                                                                                                /TODO
                                                                                                                                             以斜杠(/)开头的模式可用于禁止递归匹配
                                                   2.2 在Git仓库中记录变更
                                                                                                                                                                                只忽略当前目录的TODO文件,而不忽略子目录下的TODO
                                                                                                                                                                        build/
                                                                                                                                             以斜杠(/)结尾的模式表示目录
                                                                                                                                                                        忽略build目录下的所有文件
                                                                                                                                             以感叹号(!)开始的模式表示取反 ____ ! lib.a|忽略.a类型的文件
                                                                                                     git diff ____ 查看尚未添加到暂存区的变更
                                                                           查看已暂存和未暂存的变更
                                                                                                     git diff --staged
                                                                                                                         查看暂存的变更与上一次提交的内容的比较
                                                                                                     git diff --cached
                                                                           跳过暂存区 ____ git commit -a
                                                                                        git commit(打开文本编辑器添加提交信息)
                                                                                        git commit -m "提交信息"
                                                                            提交变更
                                                                                        git log
按时间顺序列出仓库的所有提交
                                                                                                            git rm [filename] — 从工作目录移除
                                                                                                            git rm -f [filename] — 从暂存区移除
                                                                           移除文件(从已追踪文件列表中移除)
                                                                                                            git rm --cached [filename] — 从暂存区移除,但保留在工作目录中
                                                                                                            git rm log/\*.log —— 移除log目录中所有扩展名为.log的文件
                                                                                                            git rm \*~ ____ 移除以~结尾的文件
                                                                           移动文件(重命名)
                                                                                               git mv [file_from] [file_to]
                                                                    git log -p
显示每次提交引入的差异
                                                                    git log -p -times
只输出最近times次提交
                            Chapter2 Git基础
                                                                    git log --stat
查看每个提交的统计信息
                                                                     git log --pretty=format
以format形式输出日志
                                                  2.3 查看提交历史
                                                                     git log --pretty=format命令的一些有用的选项
                                                                       git log的常用选项
                                                                   《精通Git》学习笔记
                                                                     限制提交历史的输出范围
                                                                  git commit --amend
                                                                  修正第一个提交的结果
                                                   2.4 撤销操作
                                                                                     git reset Head [filename]
                                                                  撤销已暂存的文件
                                                                                      把文件移除缓存区
                                                                  撤销对文件的修改
                                                                                     git checkout -- [filename]
                                                                                               —— 列出每个远程仓库的简短名称
                                                                                      git remote
                                                                       显示远程仓库
                                                                                      git remote -v —— 列出每个远程仓库对应的URL
                                                                                   git remote [shortname] [url]
                                                                       添加远程仓库
                                                                       从远程仓库获取和拉取数据 — git fetch [remote-name]
                                                   2.5 远程仓库的应用
                                                                       将数据推送到远程仓库 —— git push [remote-name] [branch-name]
                                                                       检查远程仓库 ____ git remote show [remote-name]
                                                                                             git remote rename [origin-name] [new-name]
                                                                       删除和重命名远程仓库
                                                                                             git remote rm [remote-name]
                                                              列举标签
                                                                          git tag
                                                                                        git tag -a v1.4 -m "my version 1.4"
                                                                          注释标签
                                                                                        创建标签v1.4
                                                                                        -m指定了标记信息
                                                              创建标签
                                                                          轻量标签 —— 不需要-a,-s,-m等选项
                                                   2.6 标记
                                                              补加标签
                                                                           git push origin [tagname]
                                                              共享标签
                                                                          git push origin --tags
把服务器上没有的标记都推送过去
                                                               检出标签
                                                   2.7 Git别名
                                                                               发起提交时,Git存储的是提交对象,其中包含了指向暂存区快照的指针。
                                                                                               作者姓名和邮箱地址
                                                                              提交对象包括:
                                                                                               已输入的提交信息
                                                                                               指向其父提交的指针
                                                                                                                                              1、暂存操作会为每个文件计算校验和
                                                                                                                                              2、把文件的当前版本保存到Git仓库中(blob对象)
                                                                                                                                              3、把校验和添加到暂存区
                                                                                         假设有一个包含三个文件的目录,而你将他们加入到暂存区并提交
                                                                                                                                                                                                               一次提交及其树对象
                                                                                                                                              4、执行git commit进行提交时,Git先为每个子目录计算校验和
                                                                                                                                              5、再把树对象保存到Git仓库中
                                                                              形象化
                                                                                                                                              6、Git随后创建提交对象,其中包括元数据以及指向项目根目录的树对象的指针
                                                                                        多次提交
                                                            3.1 分支机制简述
                                                                                                      多次提交及其父提交
                                                                                            git branch testing
                                                                                            创建一个指向当前提交的新指针
                                                                                                                        指向同一系列提交的两个分支
                                                                              创建新分支
                                                                                                                                         是一个指向当前所在的本地分支的指针
                                                                                            Git如何确定当前在哪一分支上?
                                                                                                                         HEAD 特殊指针
                                                                                                                                          git branch 命令只会创建新分支,不会切换到新的分支上去
                                                                                          git checkout [branch] — 切换Head指针到该分支
                                                                                                   $ git checkout testing
                                                                              切换分支
                                                                                                          HEAD
master
                                                                                          示例
                                                                                                  这条命令做了两件事,一般使HEAD 指回 seaster 分支,二是将工作目录恢复或 seaster 分支所指的的技能内容。 战战是尽,你现在做抢救的话,项目将给于一个校目的废本,本质上来讲,这就是影响 teeting 分友所使的核改,以便于向另一个方向进行开发。
                                                                                                  $ vim test.rb
$ git commit -a -m 'made other changes'
                                                                                                   现在,这个项目的提交历史已经产生了分叉(参观项目分叉历史)。因为刚才你创建了一个新分支,并切除过
近面行了一些工作,随后又切除回master 分支电行了另外一些工作,上述两次或引针的指示例分支,你可以
在不同分支周不免集团回唤和工作,并在时机或使用它引力并起来,指标组返工作,你需要的命令只有
                                                                                                   96/9 3462 9848 5762
                                 chapter3 Git分支机制
                                                                                                  你可以简单地使用git log命中查看分叉的走。进行git log --oneline --decorate --graph
,它会够比你的提叉可变。各个分支的指向以及使用的分支分叉模定。
                                                                                                  8 git log --smeline --fecorate --graph --all

* c2b9e (HEAD, master) made other changes

| * 67ab2 (testing) made a change
                                                                                                  * f38ob add feature #32 - ability to add new formats to the

* 34ac2 fixed bug #132B - stock overflow under certain conditions

* 9Eco9 initial commit of my project
                                                                                                  告于 Gt 野沙支工商上仅居包含作塩汁酸技能能 (长度力 40 的 544-1 電野符号) 町文井、所以日野信建和管理
都料理高效。创建一个新分金效量性一个文件中写入41 个字符 (44 个字符和 1 个実行行) 。 記述范ລ報報の
(422)
                                                                                                  git checkout -b [new-branch] —— 创建并切换到新分支
                                                            3.2 基本的分支与合并操作 — git branch -d [branch] — 删除该分支
                                                                                     git merge [branch] — 将branch合并到Head分支
                                                                                        获得当前所有分支的简短列表
(*代表当前所在分支即Head指向的分支)
                                                                           git branch —
                                                                           git branch -v — 获得每个分支的最新提交
                                                            3.3 分支管理
                                                                           git branch --merged ____ 查看有哪些分支并入了当前分支
                                                                           git branch --no-merged ____ 查看尚未合并的工作的所有分支
                                                                                   长期分支
                                                            3.4 与分支有关的工作流
                                                                                   主题分支 — 短期的,用于实现某一特定功能极其相关工作的分支
                                                                           推送 ___ git push [remote] [branch]
                                                                           跟踪分支 —— 基于远程分支创建的本地分支会自动成为跟踪分支
                                                            3.5 远程分支
                                                                                    git fetch ____ 拉取本地没有的远程所有最新更改数据,但完全不会更改你的工作目录
                                                                           拉取 -
                                                                                   git pull — 等同于执行git fetch之后紧跟着执行了git merge
                                                                           删除远程分支 ___ git push [remote] --delete [branch]
                                                                        工作原理: 首先找到两个要整合的分支(你当前所在的分支和要整合到的分支)的共同祖先,然后取得当前所在分支的每次提交引入的更改(diff),并把这些更改保存为临时文件,
                                                                        这之后将当前分支重置为要整合到的分支,最后在该分支上依次引入之前保存的每个更改
                                                            3.6 变基
                                                                                             将当前分支相对于共同祖先的更改引入到branch分支上,并将当前分支置为branch分支
                                                                        git rebase [branch]
```

已提交 (committed) 数据已被安全存储到本地数据库中