Git版本管理软件使用

1. 软件安装

请参考 <https://git-scm.com/book/zh/v2/%E8%B5%B7%E6%AD%A5-%E5%AE%89%E8%A3%85-Git>

1. git基础

## 2.1快照的概念

全球网络存储工业协会SNIA（StorageNetworking Industry Association）对快照（Snapshot）的定义是：关于指定数据集合的一个完全可用拷贝，该拷贝包括相应数据在某个时间点（拷贝开始的时间点）的映像。快照可以是其所表示的数据的一个副本，也可以是数据的一个复制品。

## 2.2git初始化

$ gitinit

该命令将创建一个名为 .git 的子目录，这个子目录含有你初始化的 Git 仓库中所有的必须文件，是使用git软件的基础。

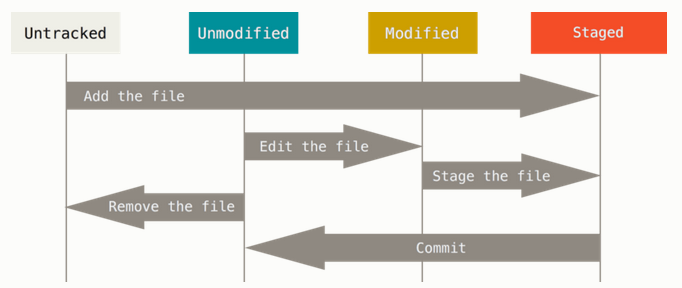
如果有远程服务器，也可以使用克隆命令来初始化。$ git clone url

## 2.3 git版本管理下各文件的状态特性

### 2.3.1共4总状态：

* Untracked:未跟踪的文件意味着Git在之前的快照（提交）中没有这些文件；Git不会自动将之纳入跟踪范围。
* Unmodifid:没有改变的文件
* Modified:改变的文件
* Staged:暂存的文件

## 2.4 文件状态的变化综述



### 2.4.1 查看当前文件相对于之前的状态

要查看哪些文件处于什么状态（相对于之前的），可以用 git status 命令。

### 2.4.2 文件状态从Untracked到Staged

使用命令git add开始跟踪一个文件。

### 2.4.3 文件状态从Staged到Unmodified

运行提交命令 git commit。使用该命令一般会加上一个–m的参数来提示已经改变的内容的说明。

### 2.4.4 文件状态从Modified 到Staged

同样使用git add 命令把文件的状态变为Staged

### 2.4.5 文件状态从Unmodified 到 Untracked

要从 Git 中移除某个文件，就必须要从已跟踪文件清单中移除（确切地说，是从暂存区域移除），然后提交。 可以用 gitrm 命令完成此项工作，并连带从工作目录中删除指定的文件，这样以后就不会出现在未跟踪文件清单中了。

## 2.5查看提交历史

回顾下提交历史。完成这个任务最简单而又有效的工具是git log命令。

## 2.6远程仓库的使用

### 2.6.1 查看远程仓库

如果想查看你已经配置的远程仓库服务器，可以运行 git remote 命令。

### 2.6.2 添加远程仓库

运行 git remote add <shortname><url> 添加一个新的远程 Git 仓库

### 2.6.3 从远程仓库拉取或抓取

git fetch [remote-name]

### 2.6.4 推送的远程仓库

git push [remote-name] [branch-name]

### 2.6.5 远程仓库的名字修改

git remote rename 去修改一个远程仓库的简写名

### 2.6.6 删掉远程仓库

如果因为一些原因想要移除一个远程仓库 - 你已经从服务器上搬走了或不再想使用某一个特定的镜像了，又或者某一个贡献者不再贡献了 - 可以使用 git remote rm

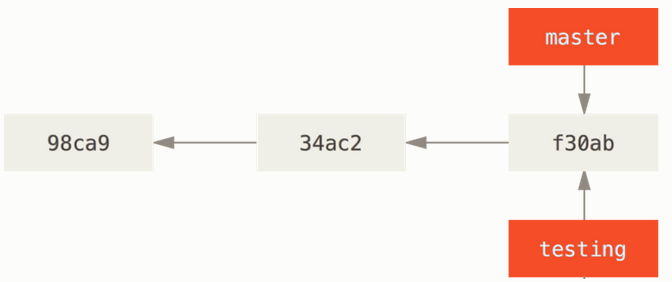
## 2.7 回到之前的版本

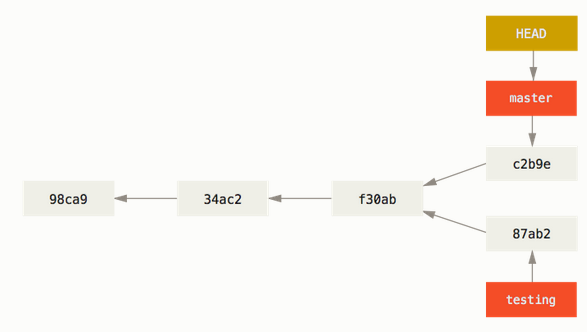
使用git log 来查看已经提交的版本

使用git reset –hard “string”

1. Git分支介绍

有人把 Git 的分支模型称为它的`‘必杀技特性’'，也正因为这一特性，使得 Git 从众多版本控制系统中脱颖而出。Git 鼓励在工作流程中频繁地使用分支与合并，哪怕一天之内进行许多次。 理解和精通这一特性，你便会意识到 Git 是如此的强大而又独特，并且从此真正改变你的开发方式。分支可以独立进行运行。





## 3.1 创建分支

git branch xx

## 3.2 切换分支

git checkout xx

## 3.3 查看分支

git branch

## 3.4 合并分支

Git checkout master

Git merge xx

1. Github

Github其实就是一个远程仓库。

## 4.1 一个例子

echo “# project\_python”>>README.md

gitinit

git add README.md

commit –m “first commit”

git remote add origin <https://github.com>.....

Push –v origin master

**git config core.autocrlf false**

git config --global http.postBuffer 524288000