Git 菜鸟教程：

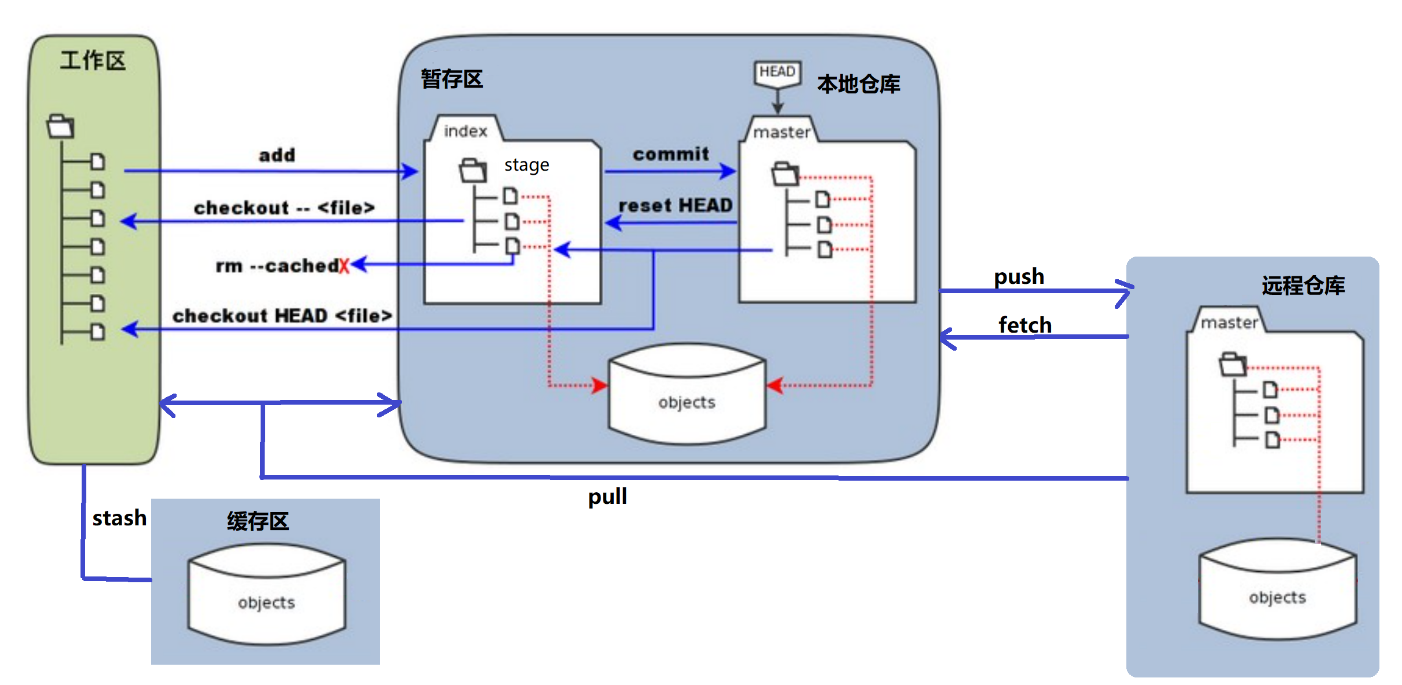
<https://www.runoob.com/git/git-tutorial.html>

Git 本机帮助命令：

git help git

请回答下面题目

1. 在git体系中，一共有几个地方可以存放代码？如何去查看在这里地方的代码情况？



Diagram

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **名称** | **英文名称** | **作用** | **查看工具** |
| 1 | 工作区 | Workspace | 用户的工作区域 | file explorer / windows os |
| 2 | 暂存区 | Stage / Index | 存放文件索引 | git command |
| 3 | 本地仓库 | Repository | 存放文件内容（本地） | git command |
| 4 | 远程仓库 | Remote | 存放文件内容（远程） | browser |
| 5 | 缓存区 | Stash | 临时缓存文件 | git command |

1. git init和git clone分别做什么用？

git init和git clone是两个初始化的命令，一般来说，用户只需要执行一次。

git init，是指本机上先有代码，然后基于本机代码创建一个本地仓库repository。但是并没有关联到远程仓库（因为不知道远程仓库地址），我们可以通过git remote来指定远程仓库地址。

例如：git remote add origin [git@192.168.31.130:/srv/sample.git](mailto:git@192.168.31.130:/srv/sample.git)

url – remote link

也可以不用此命令，直接通过browser在git server上创建一个空仓库，再push代码。

git clone，是指其它人已经在git server上创建好了代码仓库repository，开发人员需要同步到本机上，大部分人是执行此命令。

1. 解释branch和tag的区别

在git 的数据结构中，branch和tag是一样的，但在使用时有点不同

branch是分支的意思，相当于一棵树的树枝，是一个可读可写的对象，可以基于branch继续提交代码，进行新一轮开发。

tag是标签的意思，相当于一棵树的树叶，只是一个可读的对象，用来进行某种有意义的标识（因为git commit是一个guid，虽然是全局唯一的，但是不具有标识性）。如果在基于tag进行新的开发，需要基于tag先创建一个branch，然后在新branch上进行开发。

1. 在git体系中，对每一次提交commit，git是如何管理的？

每一次git commit都是唯一的，git是使用40bit hash来进行管理。

例子：aab92d7e540138b2eb45e6252b5a05a1e0bd5063

而不是git learning tool上的c1, c2 (比较直观，好理解)

1. git log 和 git status的区别是什么？

git log 查看提交记录，可以在后面加选项，用不同的模式进行显示，比如

git log --oneline 或者 git log --pretty=oneline 以精简模式显示

git log --graph 以图形模式显示

git log --stat 显示文件更改列表

git status查看本机上文件的修改状态或提交状态。

1. 在git操作中，最常用的命令是git pull和git push，作用是什么？

先了解清楚问题7，再来回答问题6

git pull和git push是最常用的git 命令，这也是二个相反的命令。

git pull是将代码从远程仓库拉到本机工作区，注意这里是指本机工作区。

事实上，git pull = git fetch + git merge，所以本地仓库也会同步。

git push则是将本地仓库中的代码同步到远程仓库，所以它的前提条件是你本机上的修改内容必须先提交到本地仓库，才能同步到远程仓库，也就是，你必须先git commit，然后git push。

1. git fetch和git merge的作用又是什么？

git fetch 命令用于从远程仓库获取代码到本地仓库，注意只是到本地仓库，并不到本地工作区。

如果你想要看看其他人都在做些什么，往git server上提交了什么时，就要使用fetch命令。

git merge 用来做分支合并，将其他分支中的代码合并到当前分支中，需要注意的是，合并的目的地是当前分支（可以用git branch来查看当前分支）。

既然是二个不同分支上的代码进行合并，有可能这二个分支上的代码是不同的，如果不同的代码没有冲突时，merge会自动进行合并，并产生一个新的commit，如果代码是有冲突的（比如同一个文件的同一个函数），merge就会给出提示，让用户进行手工merge。

1. 在git体系中，会经常用到一个单词HEAD，它的作用是什么？

在git 中，HEAD就是一个可以进行移动的指向指针，指向当前所在的本地仓库分支（或者说本地仓库分支的最后一个提交点），也就是可以将 HEAD 想象为当前分支的别名

一般情况下，是指针最后一个commit的位置，也就是说HEAD将随git commit的提交而发生改变，反过来，git commit提交的位置也是基于HEAD的。

可以通过git checkout 改变HEAD所指向的位置

在repository 目录下，有一个隐藏的子目录.git，你能找到HEAD文件，它的内容就是HEAD指针当前所指向的位置。

有时候利用HEAD，可以简化git commit 参数，比如HEAD, HEAD^1, HEAD~3，等。

1. git reset 和 git revert，以及git rebase的区别是什么？

先来理解字面意思，reset是重置，回退、revert是恢复，反做

git reset 命令用于回退版本，可以指定退回某一次提交的版本。（结合第一题中的图来理解）

git revert是用一次新的commit来rollback之前的commit，（这里可以对比一下git merge）

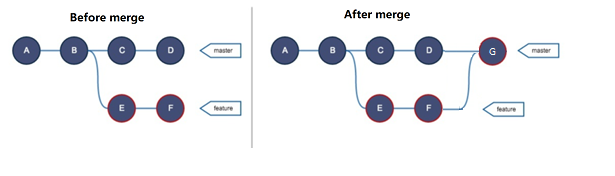
git reset是直接删除直接的commit（没有commit历史记录了）

至于rebase，跟git reset和git revert根本不是一类的命令，无非只是以re-开头而已

git rebase同类的命令应该是git merge，或者git cherry-pick，它们都属于重新组织commits

Bubble chart

Description automatically generated



另外，不带参数的git pull = git fetch + git merge，但是如果带有—rebase的参数，就不是这样的了，git pull –rebase = git fetch + git rebase

1. 在git learning 中有一个命令叫git fakeTeamwork，真正git command是没有它的，只是为了方便学习而虚构的，那么对应功能的git command是什么？

git fakeTeamwork = (git add) + git commit + git push

1. 演示题

假设某个项目有二个工程师同时开发，分别为工程师a和工程师b

1，a 在本机提交三次代码修改，分别为a1, a2, a3

2, 在a开发的同时，b也提交了三次代码修改b1, b2, b3，并先于a提交到远程服务器上

3，由于a的功能依赖于b，所以a必须要与b进行代码集成，但是a发现b提交的代码b2是一段测试代码，没有用，只需要b1和b3

4，a把代码和b1, b3集成后，发现功能不够完善，需要修改代码

5，于是a基于集成后的代码，又提交了二次a4, a5

6，经测试后，发现b遗露了某个功能，于是让b继续开发

7，b基于集成后的代码，分别提交了b4, b5

8，a基于最后修改后的代码测试，没有问题，任务完成完

请基于以上需求，用git learning完成相应的演示

Git 菜鸟测验

<https://www.runoob.com/git/git-quiz.html>

Diagram

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence