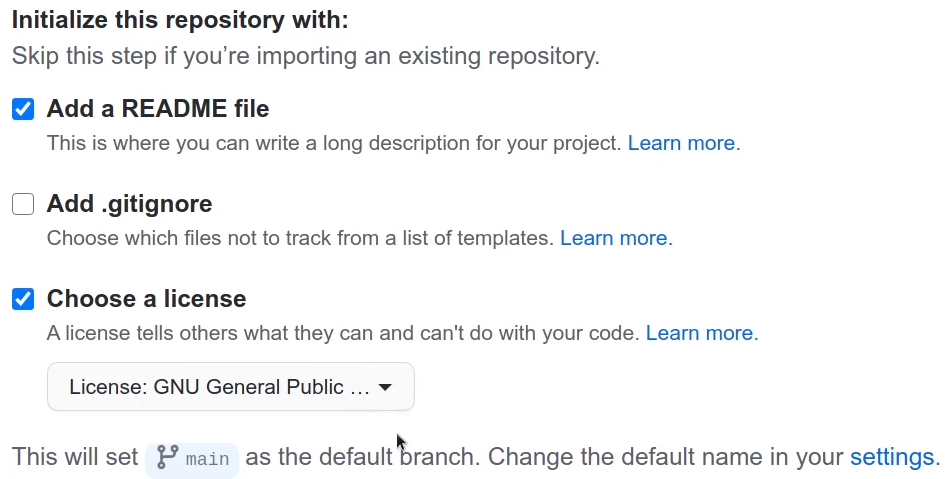
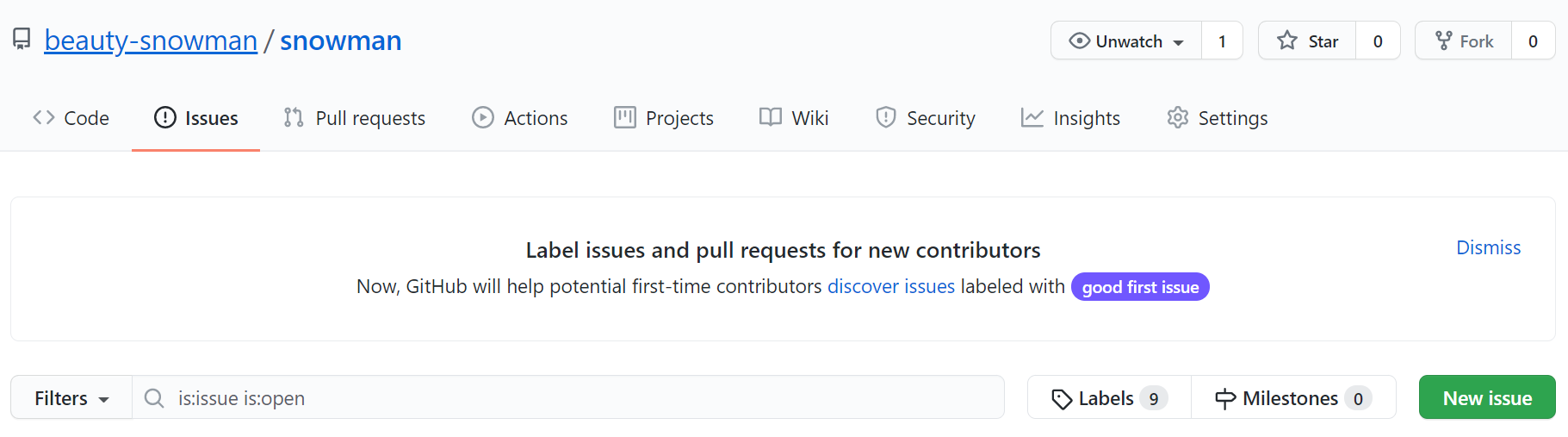
### Github



**readme file**是一个markdown类型文件(实质上还是个文本)，用来写一些项目介绍相关的东西

**.gitignore**好像是上传时屏蔽掉一些文件，不是把整个项目都上传到github中

**license**是一些许可



https://www.zhihu.com/question/21682976

**Issues：**是时时bug跟进用的，或者向作者提出问题(像是贴吧发帖评论)

**Pull requests：**first fork author repository，fork后这个项目就会克隆到自己的github中，并独立存在，fix some bugs后，then pull requests to author，让作者看到你的pull requests，如果作者认为正确，他会merge这个requests(如果他反悔，他就会点revert)，至此pull requests结束



可以将pr分配给自己

**Star：**收藏

**fork：**分叉，复制克隆项目到自己的github上，该项目fork到自己的github上后是独立存在的

**watch：**关注

**Git**

通过git管理github托管项目代码

安装选项：<https://www.cnblogs.com/hdlan/p/14395189.html>

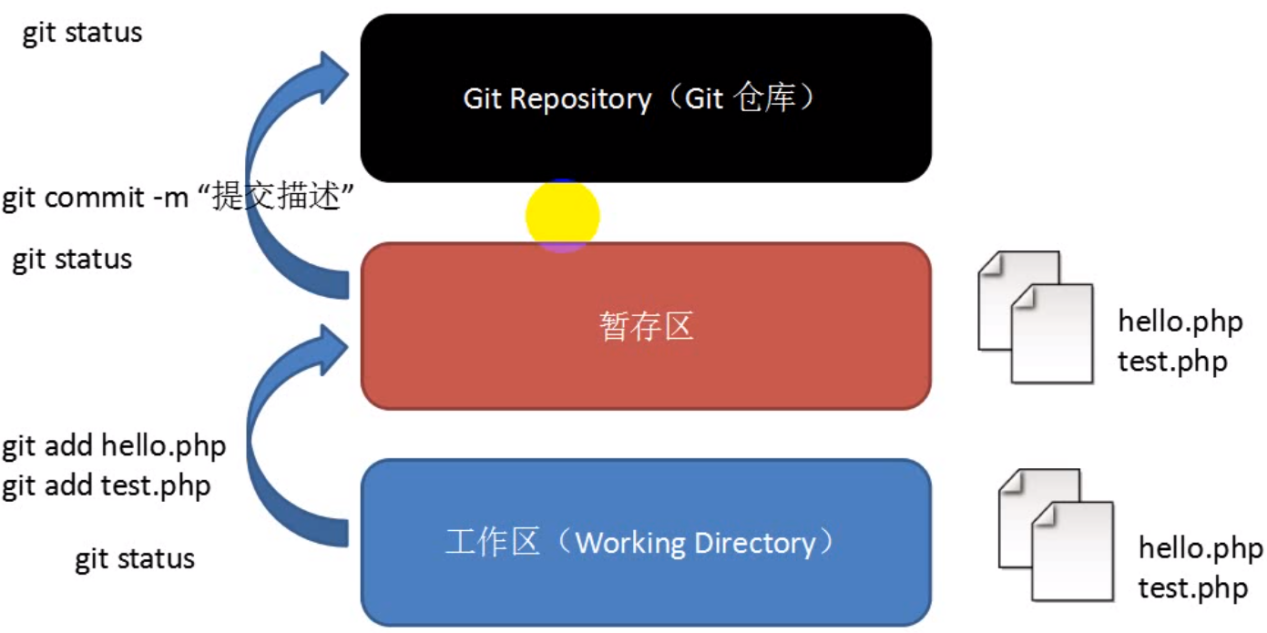
**git本地仓库**



working directory：工作区域，编辑文件等

暂存区：暂存文件，提交到git中

git repository：最后提交到仓库中的文件，他人可见(这个是个本地的仓库，本地仓库的文件别人是看不到的，必须Push后别人才能看到)



linux下操作指令：

git status：查看文件状态

git add 文件名：把working directory文件提交到暂存区

git commit -m "提交描述，仅仅是个描述用的"

ls：当前文件

pwd：当前目录

clear：清空

git config --global user.name 'beauty-snowman'：设置用户名

git config --global user.email 'jxp\_snowman@163.com'：设置邮箱

mkdir 名：创建文件夹

cd 名：进入到这个文件夹下

git config --list：查看设置

git init：会产生一个.git的文件夹，用来存储仓库所有信息

touch 文件名：创建一个文件



初始化本地仓库(和工作区域一个意思?)，然后添加到暂存区，最后提交到git仓库

修改git仓库的文件：

ls：会显示当前的所有文件？

vi 文件名：修改这个文件的内容，结束后按esc，再按:wq退出

cat 文件名：查看这个文件内容

无论创建修改都要add和commit

删除仓库的文件：

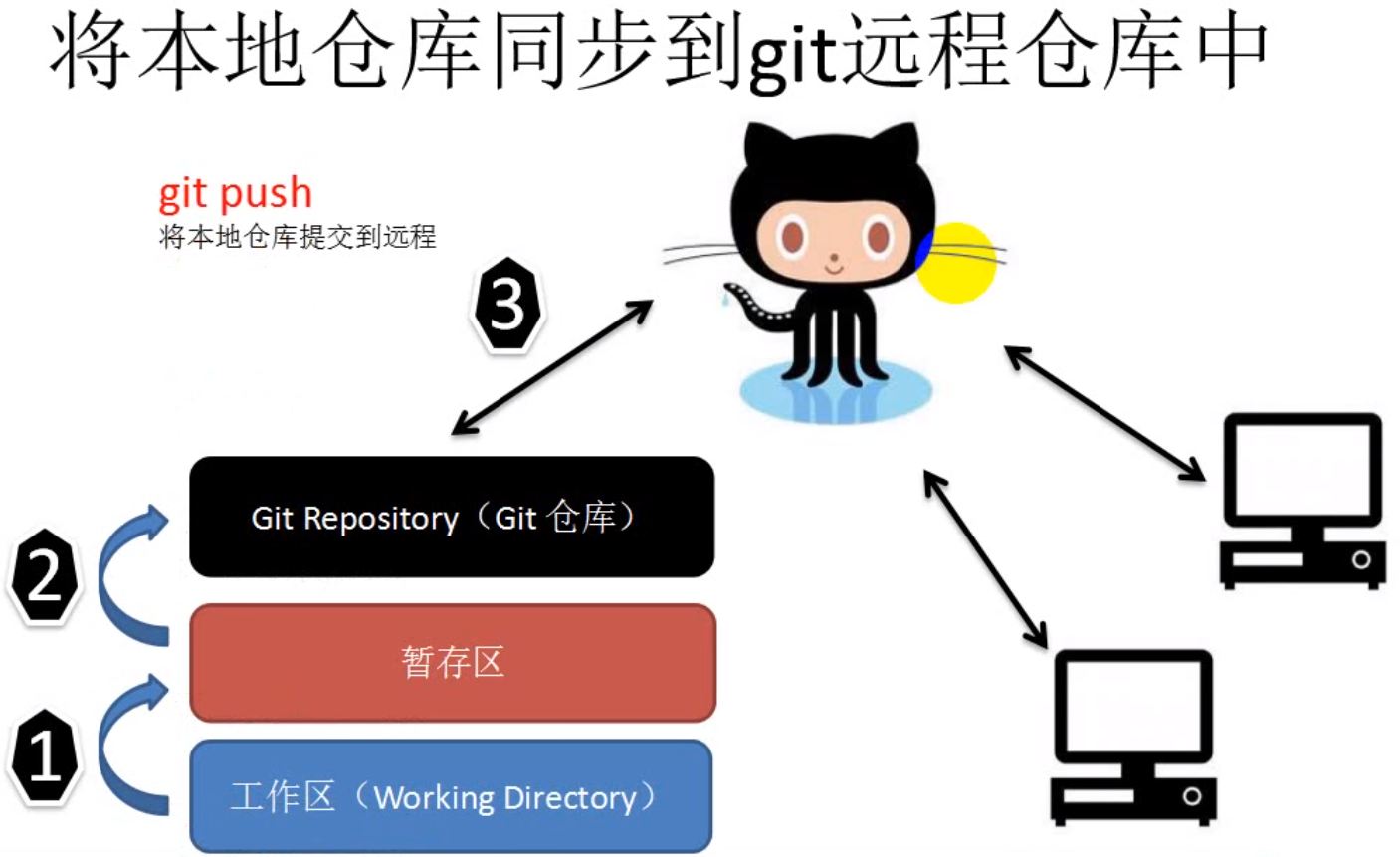
rm -rf 文件名：删除本地文件

git rm test.py：删除git仓库中的文件

git commit -m '提交描述'

**git远程仓库**

作用：备份，事先代码共享集中化管理



git克隆：

将远程仓库(github对应的项目)复制到本地

git clone 地址。执行结束后，github的项目出克隆到本地。

git push：修改后，从本地github仓库提交到远程github仓库