1 Github (github.com)

1-1 功能

托管项目代码

1-2 基本概念

》Repository（仓库）

用于存放项目代码，一个仓库对应一个项目，一个用户可以创建多个仓库存放多个项目。

》Star（收藏）：

收藏其他用户的项目（仓库），方便下次浏览查看。

》Fork（复制）：

fork按钮位于一个仓库的中，用户可以通过fork的功能克隆其他用户的仓库，即生成一个内容一致的独立的仓库，同时在仓库上会标注仓库来源（forked from…）。

》Pull Request（发起请求）

若用户B使用fork复制了用户A的一个仓库，并对该项目做了改进，此时B可以通过该仓库的pull request功能想项目所有人A发送一个请求，提出自己的改进意见；A收到B的请求后，可以查看B请求的具体内容，若采纳B的意见可以选择将B的修改合并到自己的项目中。

》Watch（关注）

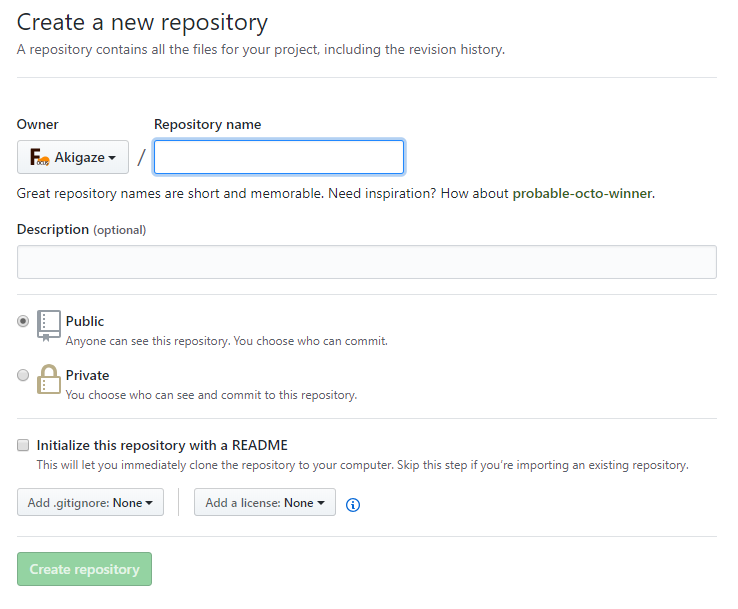
关注项目，接收项目的更新推送。

》Issue（事务）

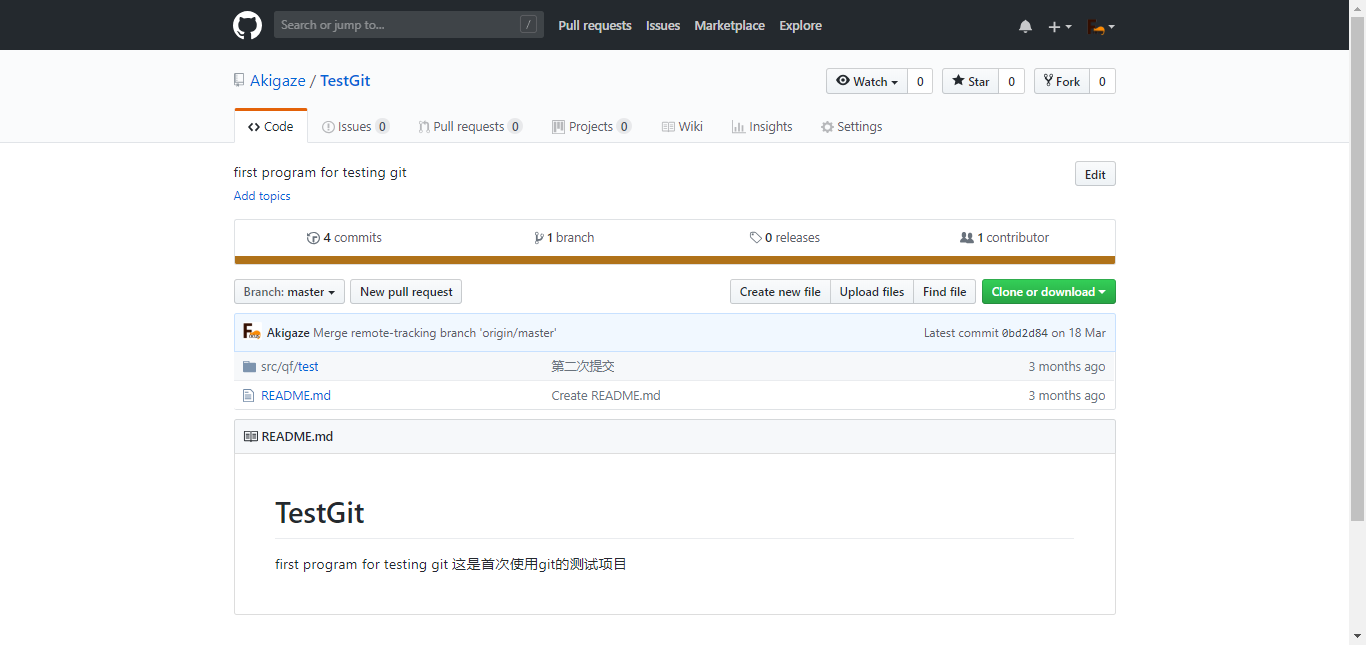
若用户B在使用用户A的开源项目的过程中，发现存在缺陷或bug，则B可以在此项目上对A发送一个issue，提出自己发现的问题和一些解决方案，待A接收并解决了这些问题后，就可以将该issue关闭。

1-3仓库管理

》创建仓库（new repository）

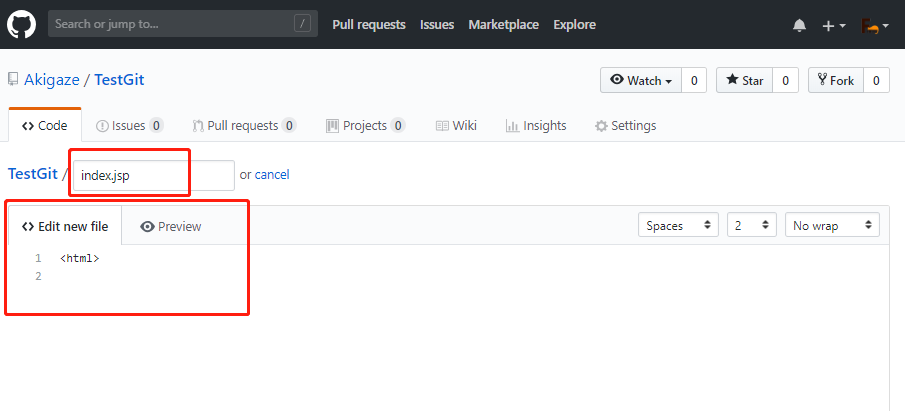


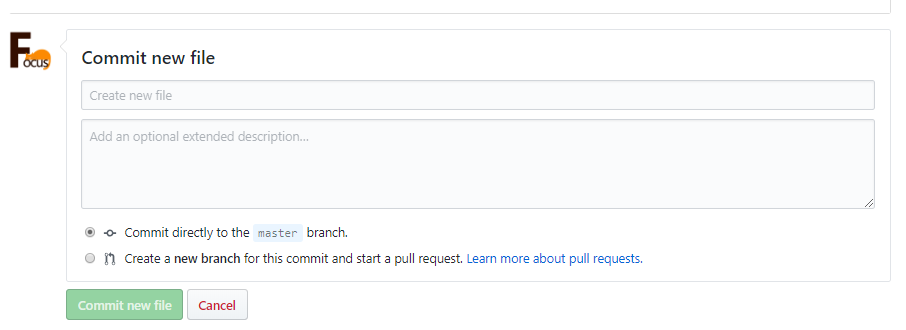
》仓库主页



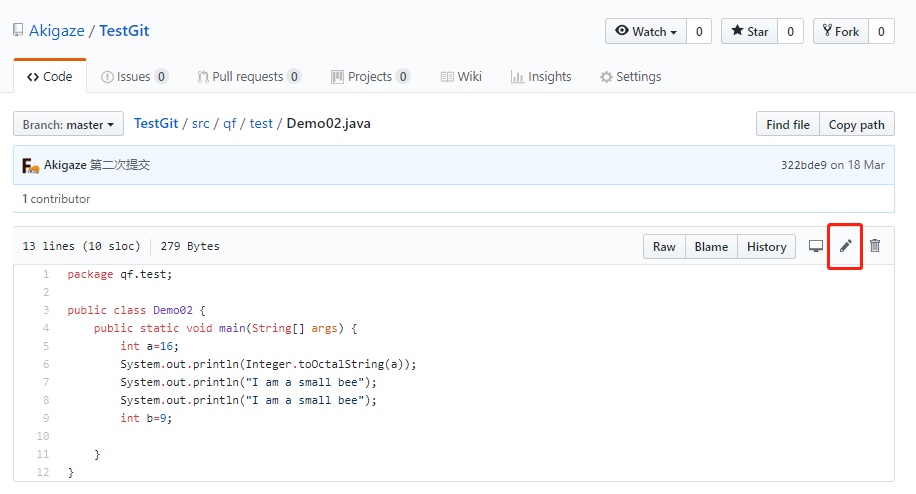
》创建文件（Create new file）

编写新文件并添加说明。



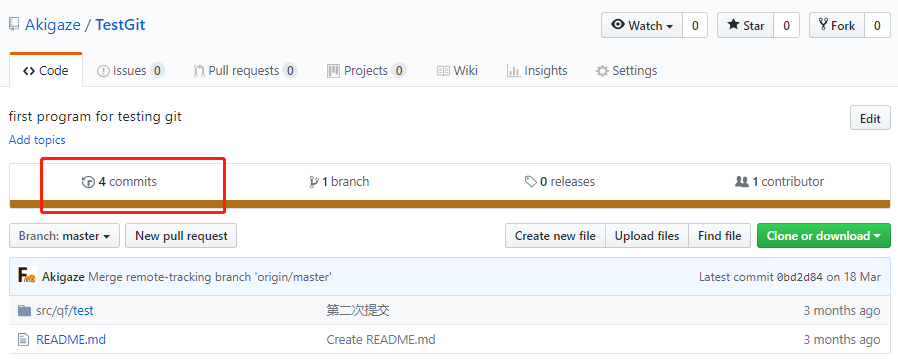


》编辑、删除文件

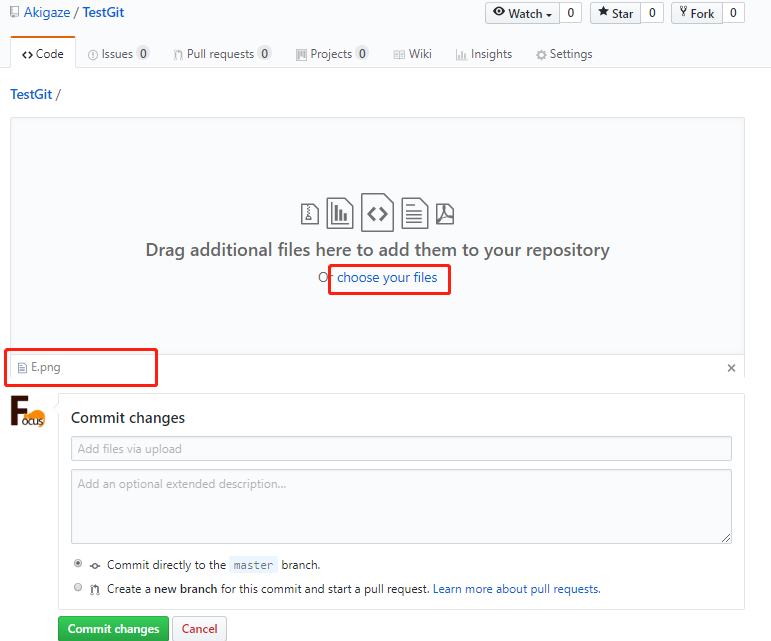


》查看项目修改记录

每次进行新增、修改、删除文件等操作都需要进行提交，此时仓库主页的commits选项卡会显示提交的次数，点击可以查看每次具体的修改。

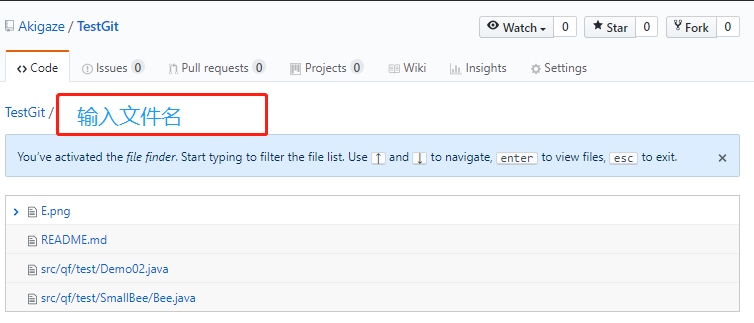


》上传文件（upload files）



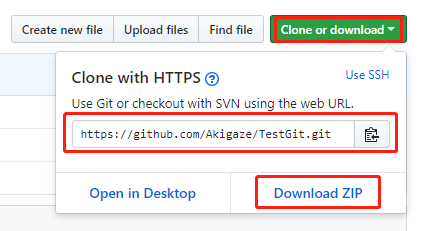
》搜索文件（find file）

在仓库首页使用快捷键“t”可打开此功能。



》下载项目（Clone or download）

URL的链接是用于Git或SVN管理项目时使用。

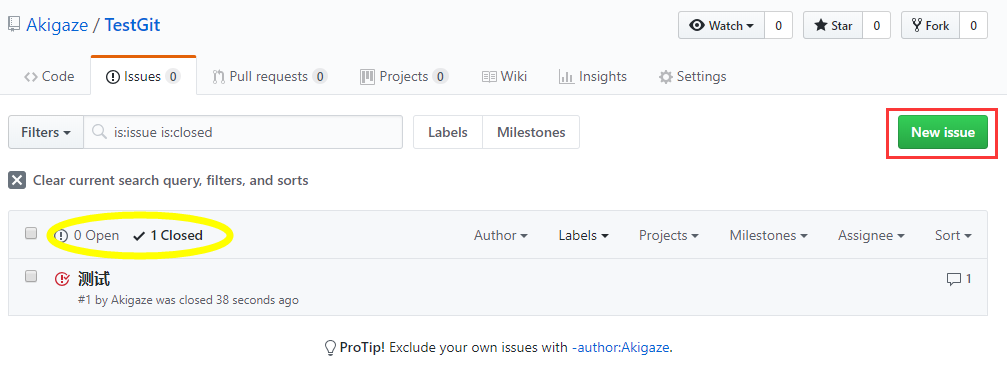


1-4 Guthub Issues

作用：用户之间对一个项目进行提问交流。

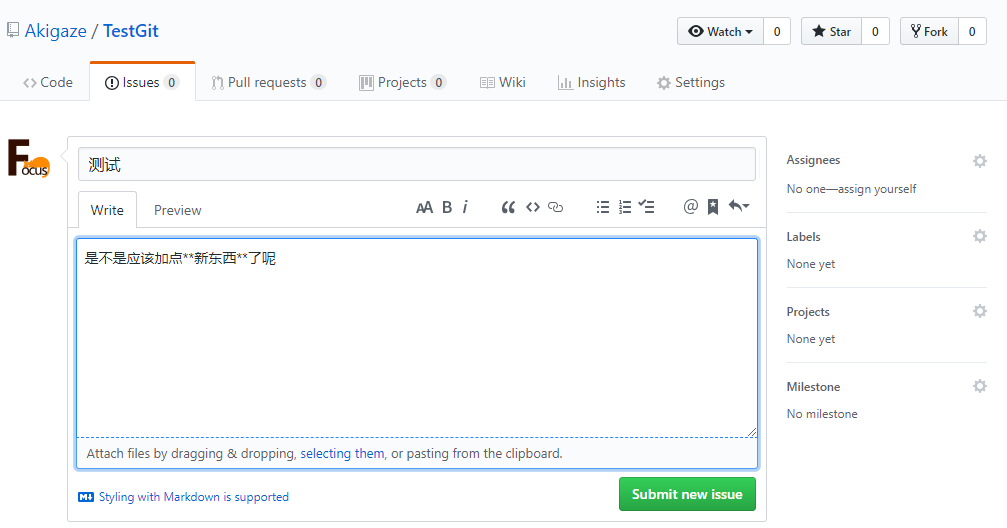
》查看Issue

在仓库主页的Issues选项卡可以查看一个项目的所有Issue或对其新增Issue，同时。



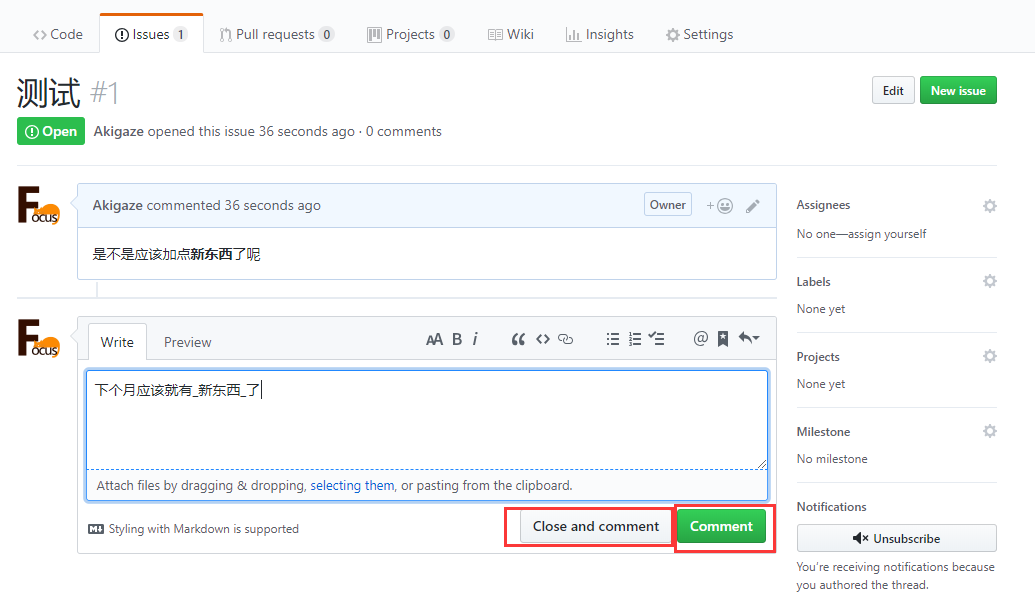
》提出Issue

对一个项目提出一个Issue，可以对自己的仓库提出Issue。



》关闭Issue

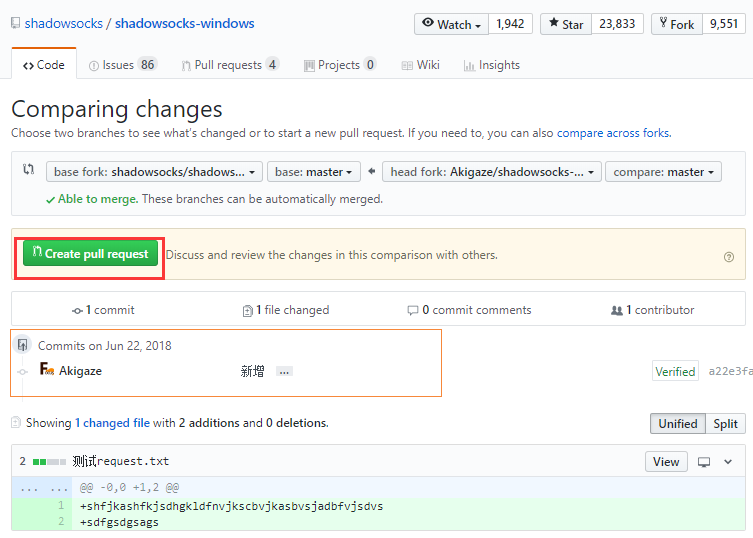
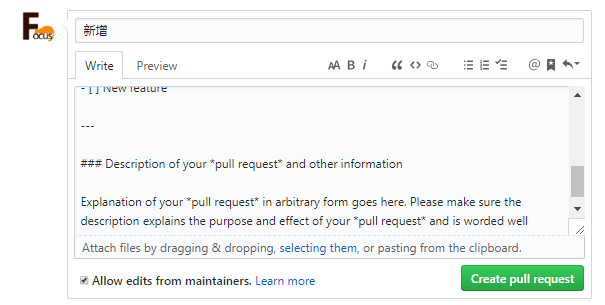
用户收到一个Issue后，可以回复该Issue，回复后将Issue关闭（Close）。



1-5 发起请求

对一个项目进行编辑后，可以使用Pull Request对项目的原始所有人发起合并修改的请求，对于他人的项目，只有fork拷贝到自己的仓库才能编辑并发起Request。

点击Create pull request按钮后可以编写对本次request的说明。

2 Git (git-scm.com)

Git是一个开源的版本控制软件。

目的：通过Git在本地管理Github托管项目代码

2-1 Git的工作区域

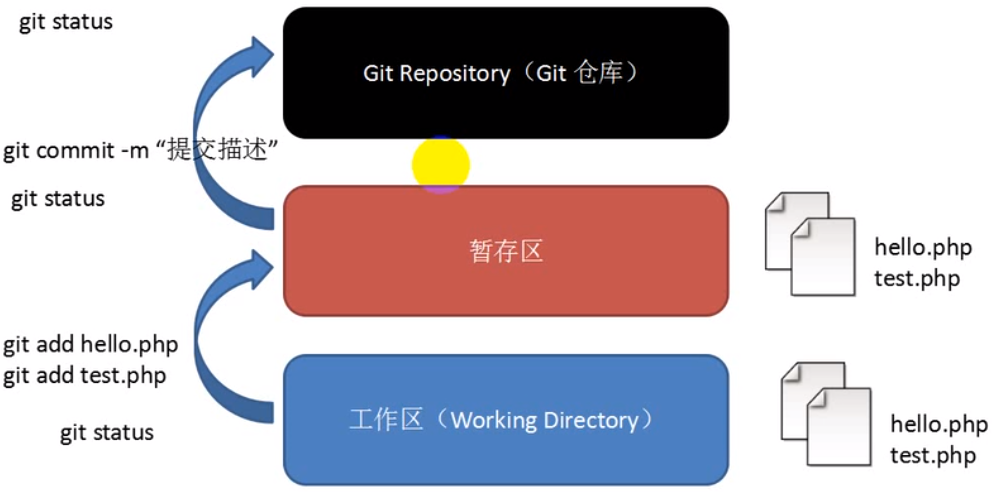


2-2 向Git仓库添加文件

**git status：**查看所有项目文件在Git中的状态。

**git add：**将项目文件从工作区添加到暂存区，git add 文件名.扩展名。

**git commit：**将暂存区的文件提交到Git仓库，并添加提交说明，git commit –m ‘描述信息’。



2-3 Git初始化及仓库创建

》关联Github账号

关联用户名：git config --global user.name ‘Github用户名’

关联用户邮箱：git config --global user.email ‘Github用户注册使用的邮箱’

查看Git配置：git config --list

》创建初始化仓库

git仓库需要一个工作空间，相当于一个工作文件夹，所以Git仓库需要在一个空文件夹中创建。

**创建文件夹：**使用mkdir命令或在Windows窗体创建文件夹。

**进入文件夹：**使用cd命令或是直接在仓库文件夹中右键选择“Git Bash Here”打开Git命令窗口。

**初始化仓库：**git init 在当前目录下创建并初始化一个git仓库，同时会生成一个.git结尾的隐藏数据文件。

》新增修改仓库文件

每次对仓库中的文件进行新增、修改等操作之后，都要先将文件添加到暂存区（git add），再提交到Git仓库。

》删除仓库文件

**删除工作区中的文件：**使用Windows窗体删除或rm –rf命令删除文件。

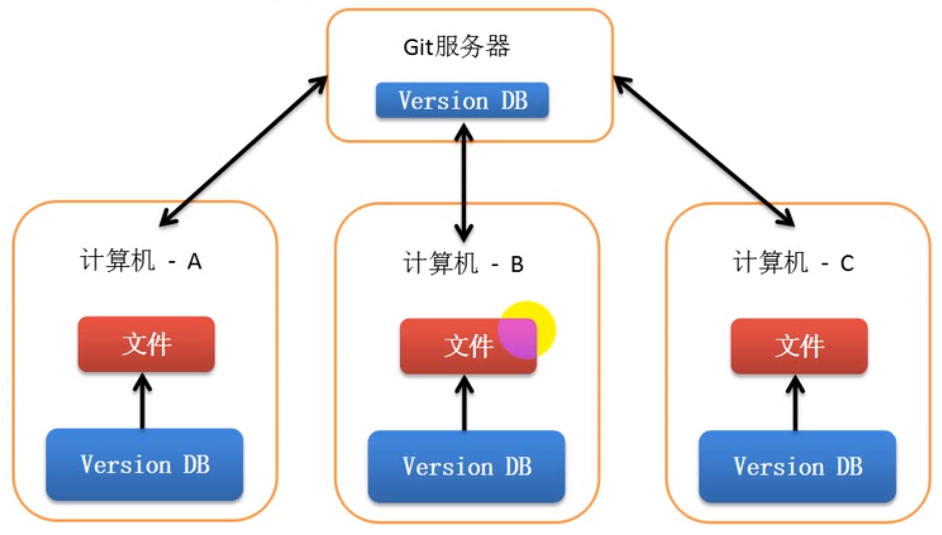
**删除Git中的文件：**git rm 文件名.扩展名

**提交到Git仓库**

2-4 Git远程仓库

这里所说的远程仓库应该就是指Github。

作用：实现项目备份、代码共享和集中化管理，将本地Git仓库中的项目代码同步到Github仓库中。



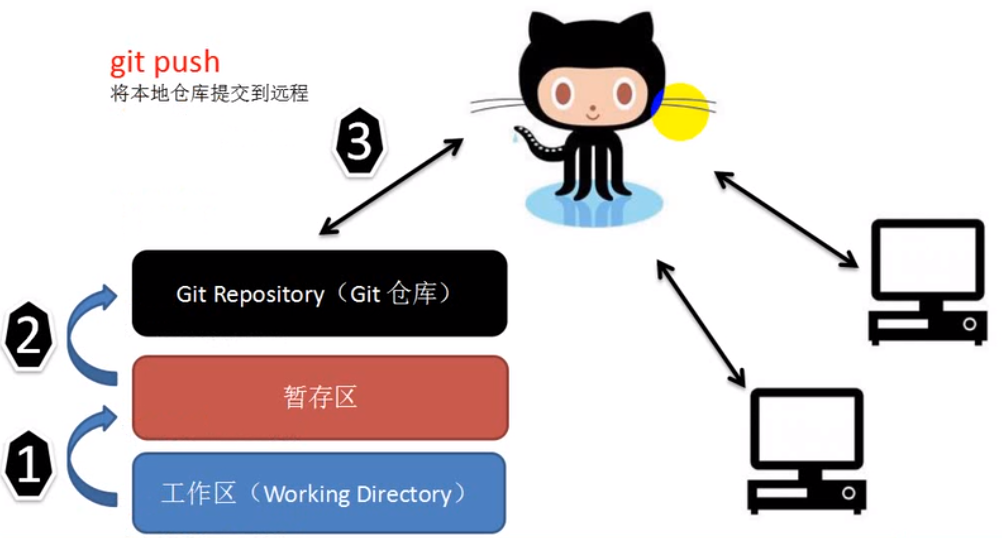
》从Github仓库克隆项目

git clone：git clone 仓库URL 指定Github仓库的地址，将指定仓库克隆（下载）到本地当前目录，在本地形成一个新的仓库。

Github仓库URL从Github仓库主页的“Clone or download”按钮可以查看得到。

》代码同步到远程仓库

git push：将当前工作目录的仓库同步到远程仓库。



》无法同步问题



2-5 Linux命令

ls：查看当前文件

pwd：查看当前工作目录

clear：清楚命令行代码

mkdir：mkdir 文件夹名称 在当前目录下创建文件夹

cd：cd 子文件夹名称 进入子文件夹的目录中，

touch：创建文件，touch 文件名.扩展名

vi：vi 文件名.扩展名 打开编辑文件，若文件不存在，则新建一个

rm –rf：rm –rf 文件名.扩展名 删除文件

3 Github搭建个人网站

3-1 Github Pages

搭建个人Github的网站，网址：https://用户名.github.io

原理：利用Github仓库作为个人网站的载体，实际访问的都是仓库中写好的HTML文件。

》创建站点

实际上就是创建一个仓库，在仓库中新建HTML文件。

* 仓库名称必须为 用户名.github.io。
* 仓库中必须有一个index.html文件，作为访问的首页。
* 仓库中只能有HTML文件。

3-2 Project Pages

项目网站，即为每个项目添加一个单独的站点，网址：https://用户名.github.io/仓库名称

在仓库主页的settings选项卡，在Github Pages设置中生成或选择仓库网站的主题。

