## 一、目的

借助 github 托管项目代码----放到仓库中（Repository）

## 二、基本概念

1仓库（Repository）

仓库用来存放代码，每个项目对应一个仓库，多个开源项目则有多个仓库

2关注(Watch)

看到开源项目感觉不错，点击Watch，那么以后该项目有任何更新，自己都会第一时间收到关于这个项目的通知提醒

情景：张三关注了李四的项目，李四添加项目文件，张三的 github 主页就会有该动态提示

3收藏（Star）

收藏项目，方便下次查看

情景：张三无意访问到李四的开源项目感觉不错并进行收藏

4复制克隆项目（Fork）

Fork的项目是独立存在的

1.张三/test仓库

2.李四/test仓库

forked from 张三/test仓库

情景：张三 fork 了李四的项目，相当于复制了李四的项目，所以自己也单独有了一个一样名称的仓库（注：该仓库声明来自李四，但是是独立存在的）

5发起请求（Pull Request）

1.李四fork了张三的test仓库

2.李四修改了forkd test仓库,并可以发送请求（Pull Request）---将自己修改部分, 提醒张三可以进行更改，等待李四查看

3.李四感觉还不错，则可以进行合并，到自己的仓库中

情景：张三修改了 fork 李四的项目中的文件，希望更新到原来的仓库，这时候他要新建一个 pull requset； 李四收到动态消息之后可以选择：merge pull resquest 将项目进行同步更改/合并

6事务卡片（Issue）

你的一个开源项目，别人发现有Bug、或者哪里做的不好，就可以向你提出一个Issue(问题)，你就可以根据Issue进行修复Bug以及问题

情景：张三发现李四开源项目有问题或者Bug，则提交了一个 Issue,告知李四; 李四隔天登陆在 github 主页看到通知并和张三交流，最后关闭 Issue

7主页

Github主页

账号创建成功或者点击网站导航栏github图标都可以进入的github主页

该页左侧主要显示用户动态以及关注用户仓库的动态，右侧显示所有的git库

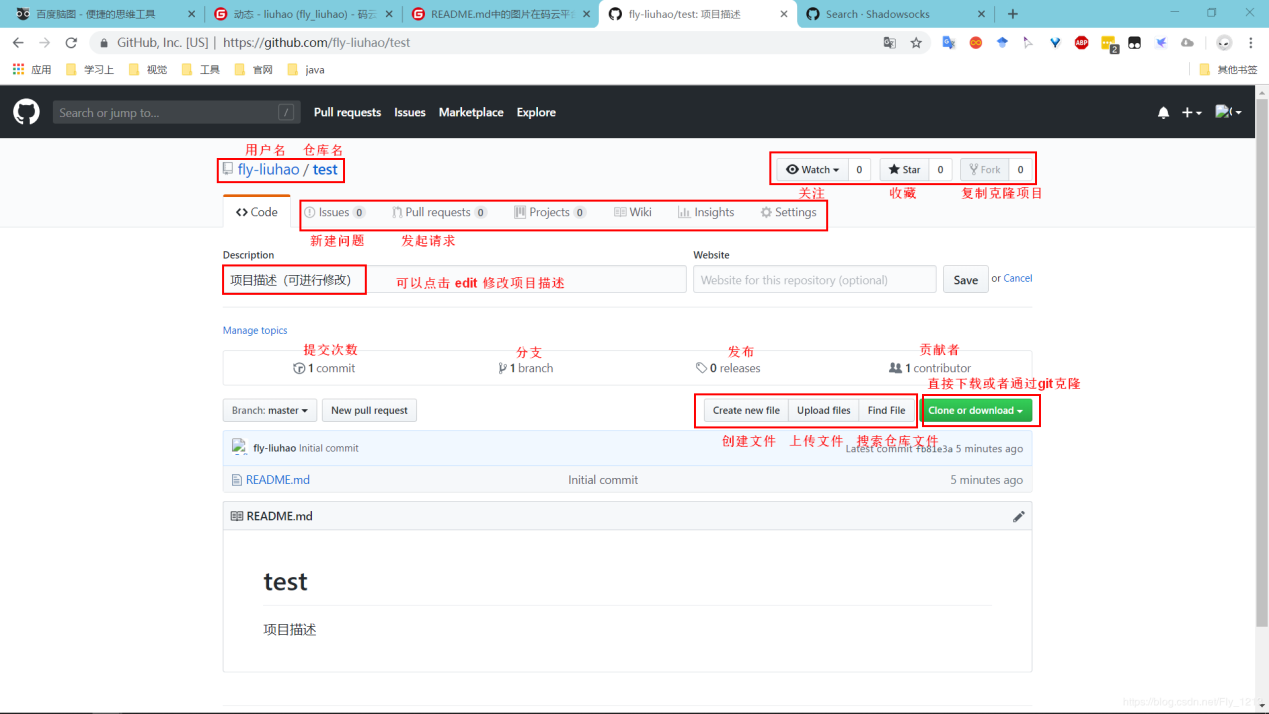
仓库主页

仓库主页主要显示项目的信息，eg:项目代码、版本、收藏/关注/fork情况

个人主页

个人信息：头像、个人简介、关注我的人、我关注的人、我关注的git库,我的开源项目、我贡献的开源形目等信息

图示如下



## 三、注册 github 账号

官网地址：github.com

因为 github 在国外服务器所以访问较慢或者无法访问，需要翻墙（Shadowsocks）自己搭建vnc服务器

私有仓库只能自己或者指定的朋友才有权操作（私有仓库是收费的）

新注册的用户必须验证邮箱才能创建git仓库（qq邮箱验证时需要设置白名单，具体操作：进入邮箱，设置，反垃圾，设置域名白名单，添加github.com）

**四、创建仓库/新项目**

一个仓库对应一个开源项目

通过 git 管理 git 库

README文件：项目描述文件

创建流程：输入仓库/项目名称—输入项目描述—选择public公开项目—选择生成README说明文件

**五、仓库管理**

创建文件—create new file—…---Commit new file

编辑文件—点击文件名—点击笔图案（Edit this file）—…---Commit changes

删除文件—点击文件名—点击垃圾桶图案（Delete this file）—…---Commit changes

上传文件—点击上传文件按钮 Upload files----拖拽/选择 需上传的文件

搜索仓库文件—点击查找文件按钮Find file 或者 按快捷键"T"—输入关键字

项目下载/检出—点击按钮Clone or download

开源项目贡献流程

Pull request—先将别人的文件fork一下，再进行pull，发送添加文件请求。

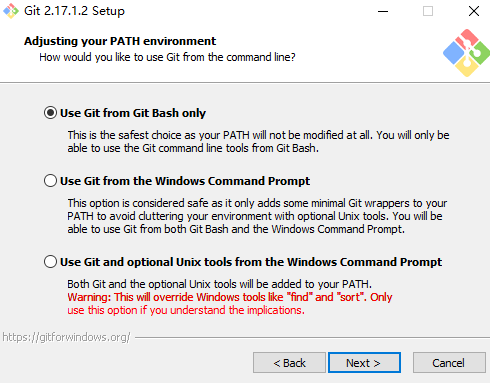
## 六、Git

### 1. Git的安装

目的：使用git管理github托管项目代码

官方下载网址：<https://git-scm.com/download/win>

安装：注意这里选择第一个；其余傻瓜式安装即可



检验是否安装成功：右击鼠标显示Git GUI Here和Git Bash Here

### 2. Git基本工作流程：

1. Git Repository（Git仓库）最终确定的文件保存到仓库，成为一个新的版本，并对他人可见

2. 暂存区 暂存已经修改的文件最后统一提交到Git仓库中

3. 工作区（Working Directory）添加、编辑、修改文件等动作

### 3. Git的使用

**1. Git基础设置：**

在自己电脑上，（新建一个文件夹，进入文件夹）右键打开git终端（linux系统指令）

1）设置用户名 git config --global user.name 'Seasea77'

2）设置用户名邮箱 git config --global user.email ‘15221167190@163.com’

3）查看设置 git config –list

**2.初始化一个新的Git仓库：**

1）创建文件夹（就是仓库，mkdir 名称）

2）在文件夹内初始化Git（创建Git 仓库）

命令行进入当前目录（cd 名称），使用 git init命令，成功会显示.git文件。

用于存储仓库所有信息。

3）此时创建的仓库是本地仓库

4）向仓库中添加文件：touch a.py 或者 直接创建

**3.git增删改仓库信息**

**（1）增**

工作区转入暂存区：

git status（用于查看文件在git中的状态）

git add a.py

暂存区转入Git 仓库：

git status

git commit –m ‘提交描述’

确定文件是否已在Git仓库中：

git status

**（2）删**

文件修改

使用vi a.py进入目标文件

按i进行文件编辑

编辑完成按：加上wq回车

工作区转入暂存区：

git status

git add a.py

暂存区转入Git 仓库：

git status

git commit –m ‘提交描述’

确定文件是否已在Git仓库中：

git status

**（3）改**

工作区文件删除

rm a.py

从git中删除文件

git rm a.py

提交操作

git commit –m ‘提交描述’

确定文件是否已在Git仓库中：

git status

### 4. 远程仓库

使用目的：备份、实现代码共享集中化管理

如何将本地仓库同步到远程仓库中：

1. 将远程仓库（github对应的项目）复制到本地：

git clone 仓库地址

注意：仓库地址在clone or download按钮下取得

2. 进行文件增删改查，并添加到Git仓库中

3. 将本地仓库同步到远程仓库中

使用命令：git push

如果git push出现The requested URL returned error：403 Forbidden while accessing问题如何解决：



## 七、 Github Pages 搭建网站

**1. 个人站点**

访问: https://用户名.github.io

搭建步骤：

（1）创建个人站点：—>创建仓库（注：仓库名必须是 【用户名.github.io】）

（2）在仓库下新建index.html 的文件即可

注：

（1）github pages 仅支持静态网页

（2）仓库中只能含有index.html文件

**2. Project Pages 项目站点**

访问：https://用户名.github.io/仓库名

搭建步骤：

（1）进入项目主页，点击settings

（2）在setting页面，点击 【Choose a theme】

（3）选择一个主题 【Select theme】，即可

注：

（1）这种类型的仓库通过修改项目的Setting建立，一个项目只能建立一个；

（2）仓库新建了一个gh-pages分支用于构建和发布页面；

（3）多用于展示项目效果

## 八、网友评论：

---------------------

建议那些感觉老师讲的太浅的（慢的）自己买一本 《Pro Git》应该从入门到精通都可以用到了。  
初学者可以看 git - 简明指南  
学了一点的可以去 try.github.com网站上看一下也可以学到很多东西

---------------------

Git官方文档：https://git-scm.com/book/zh/v2

---------------------

看完该视频建议看：https://www.bilibili.com/video/av8261658/?spm\_id\_from=333.788.b\_636f6d6d656e74.144

---------------------

---------------------

原文：参考文章链接：https://blog.csdn.net/Fly\_1213/article/details/89294702