



主讲教师：传智.花荣

传智播客 PHP学院 <http://php.itcast.cn> 发布

## 1. 了解Git和Github

### 1.1 什么是Git

Git是一个免费、开源的版本控制软件

## 1.2 什么是版本控制系统

版本控制是一种记录一个或若干个文件内容变化，以便将来查阅特定版本修订情况得系统。

➤ 系统具体功能  
记录文件的所有历史变化  
随时可恢复到任何一个历史状态  
多人协作开发或修改  
错误恢复

## 1.3 什么是Github

Github是全球最大的社交编程及代码托管网站（<https://github.com/>）。  
Github可以托管各种git库，并提供一个web界面（用户名.github.io/仓库名）

## 1.4 Github和Git是什么关系

Git是版本控制软件  
Github是项目代码托管的平台，借助git来管理项目代码

## 1.5 为什么学习github

学习优秀的开源项目  
关注行业前辈了解最新的行业动态

如：

PHP 鸟哥，前新浪总架构师 <https://github.com/laruence>

国内最流行的PHP开发框架（thinkphp）：<https://github.com/top-think/thinkphp>

全球最流行的PHP框架（laravel）：<https://github.com/laravel/laravel>

PHP编码规范（FIG-PHP）：<https://www.gitbook.com/book/jifei/php-fig-standards/details>

# 2.使用Github

## 2.1 目的

借助github托管项目代码

## 2.2 基本概念

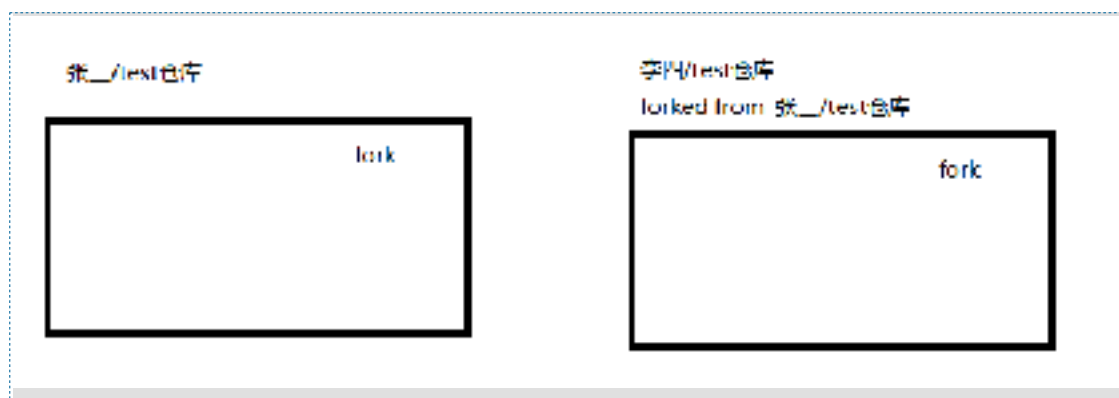
### 仓库（Repository）

仓库用来存放项目代码，每个项目对应一个仓库，多个开源项目则有多个仓库

### 收藏（Star）

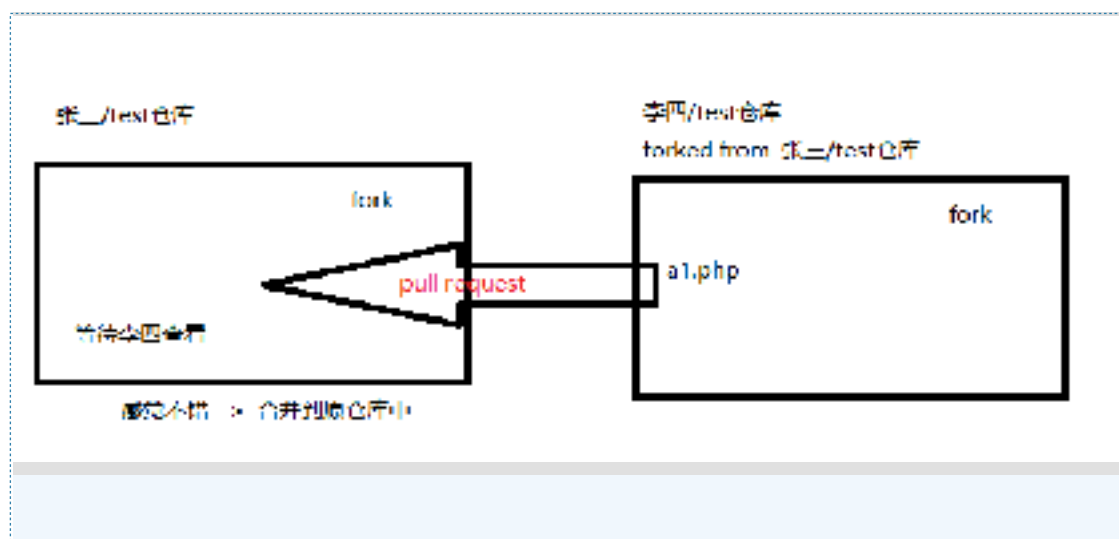
收藏项目，方便下次查看

### 复制克隆项目（Fork）



脚下留心：该fork的项目时独立存在的

### 发起请求（Pull Request）



## 关注（**Watch**）

关注项目，当项目更新可以接收到通知

## 事务卡片（**Issue**）

发现代码BUG，但是目前没有成型代码，需要讨论时用；

## Github主页

账号创建成功或点击网址导航栏github图标都可进入github主页：该页左侧主要显示用户动态以及关注用户或关注仓库的动态；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，如：项目代码，版本，收藏/关注/fork情况等

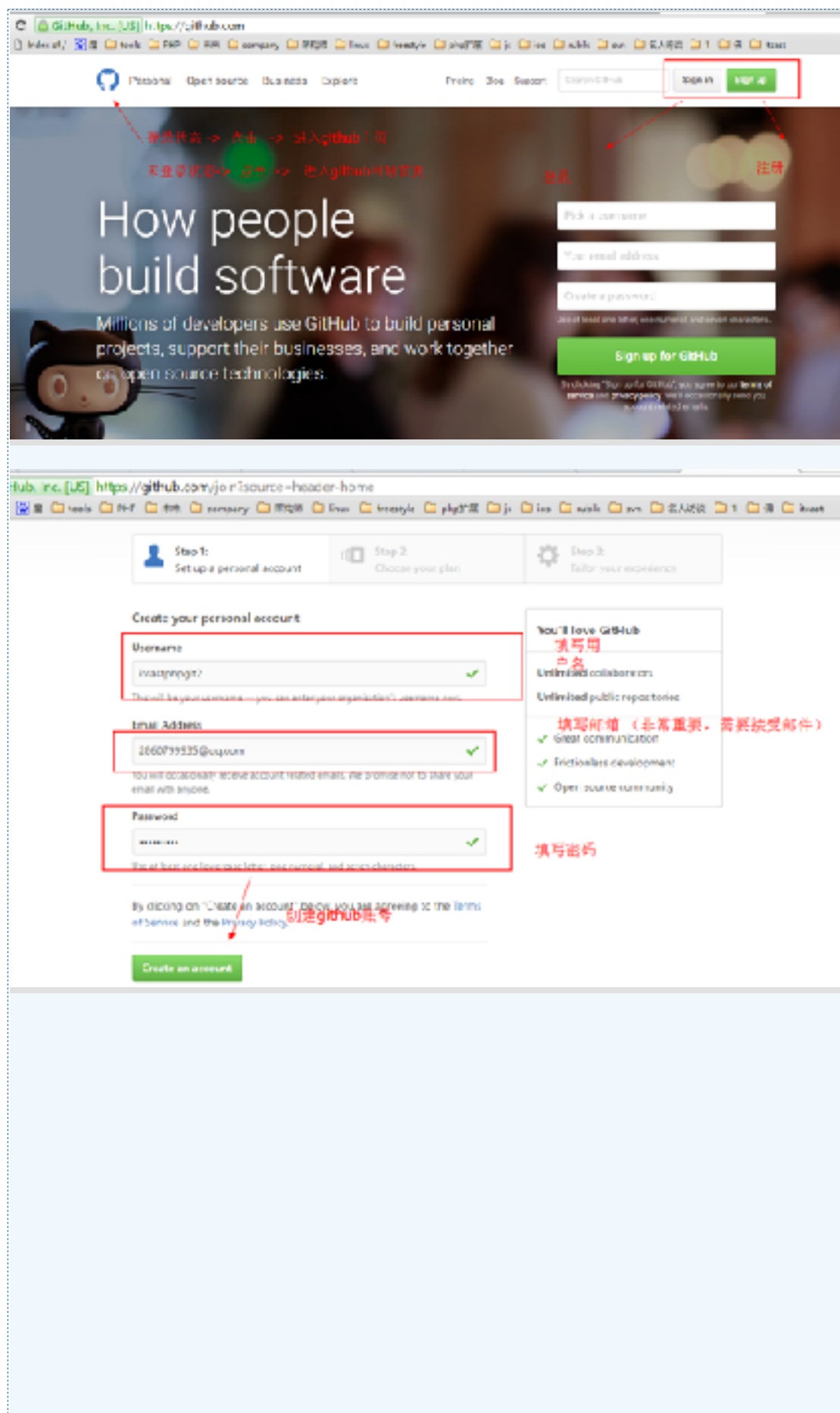
## 个人主页

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，我贡献的开源项目等信息

## 2.3 注册github账号

### 官方网址

[github.com](https://github.com)



The image shows two screenshots of the GitHub account creation process. The first screenshot is the 'Choose your plan' step, and the second is the 'Tailor your experience' step. Both screenshots have handwritten Chinese annotations.

**First Screenshot: Choose your plan**

URL: <https://github.com/join/plan>

Steps: Step 1: Completed (Set up a personal account), Step 2: Choose your plan, Step 3: Tailor your experience

Choose your personal plan:

- ☐ Unlimited public repositories for free.
- ☐ Unlimited private repositories for \$5/month.

Both plans include:

- ✓ Collaborative code review
- ✓ Issue tracking
- ✓ Open source community
- ✓ Unlimited public repositories
- ✓ Join any organization

Alternatively, you can cancel or upgrade at any time.

☐ Help me set up an organization next  
Organizations are essential for business accounts and are best suited for businesses who need to manage permissions for many employees.  
[Learn more about organizations](#)

Continue

**Handwritten Annotations (Red):**

- 选择你的计划 (Choose your plan)
- 免费无限使用公开的仓库（任何人都可以访问和fork该仓库项目！） (Free unlimited use of public repositories (anyone can access and fork the repository project!))
- 私有的，收费需要美元 (Private, paid in dollars)
- 默认继续下一步 (Continue to the next step by default)

**Second Screenshot: Tailor your experience**

URL: <https://github.com/join/customize>

Steps: Step 1: Completed (Set up a personal account), Step 2: Choose your plan, Step 3: Tailor your experience

How would you describe your level of programming experience?

- ☐ Very experienced
- ☐ Somewhat experienced
- ☐ Briefly new to programming

What do you plan to use GitHub for? (check all that apply)

- ☐ Project Management
- ☐ Research
- ☐ Design
- ☐ Development
- ☐ School projects
- ☐ Other (please specify)

Which is closest to how you would describe yourself?

- ☐ I'm a professional
- ☐ I'm a hobbyist
- ☐ I'm a student
- ☐ Other (please specify)

What are you interested in?

e.g. tutorials, android, ruby, web-development, machine-learning, open-source

Submit skip this step

**Handwritten Annotations (Red):**

- 跳这一步 (Skip this step)



## 脚下留心

1. 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）
2. 私有仓库只能自己或者指定的朋友才有权限操作（私有仓库是收费的）
3. 新注册的用户**必须验证邮箱**后才可以创建git仓库（）

## 2.4 创建仓库/创建新项目

### 说明

一个git库（仓库）对应一个开源项目  
通过git管理git库

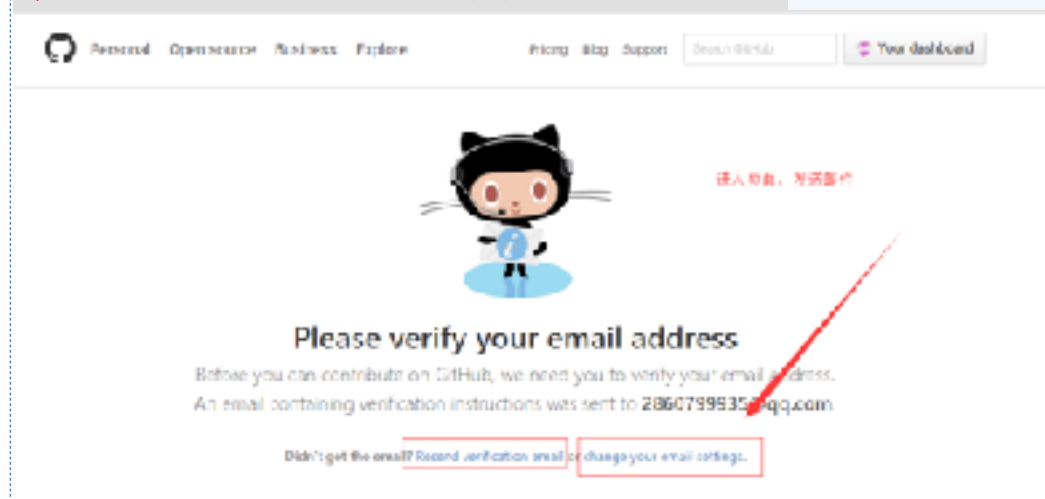
### 创建仓库

1) 点击【Start a project】创建一个仓库

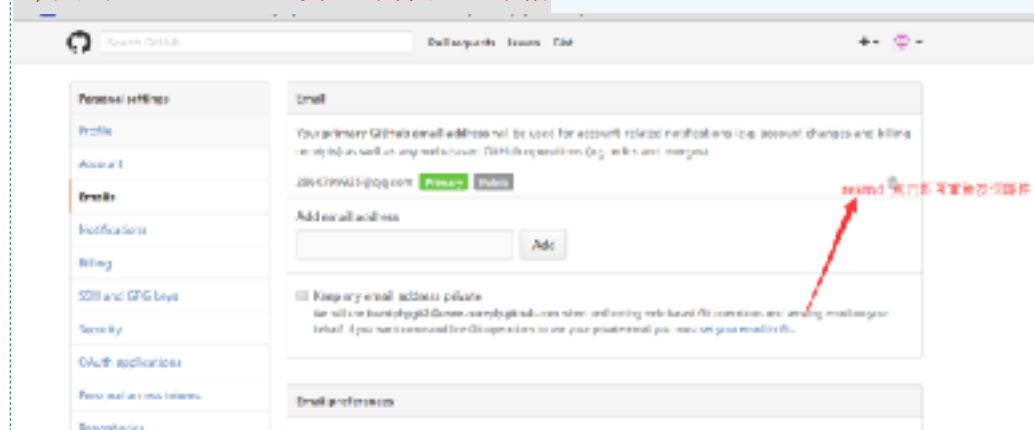


2) 问题：点击【Start a project】创建一个仓库，后出现该页面

2) 原因：未验证邮箱，点击下图框框中的链接进行验证



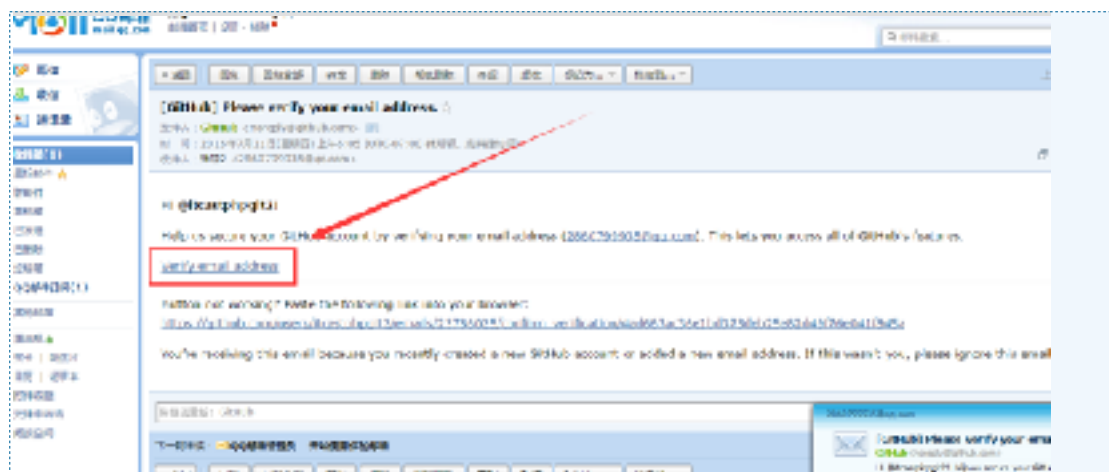
3) 点击【resend】发送邮件验证邮箱



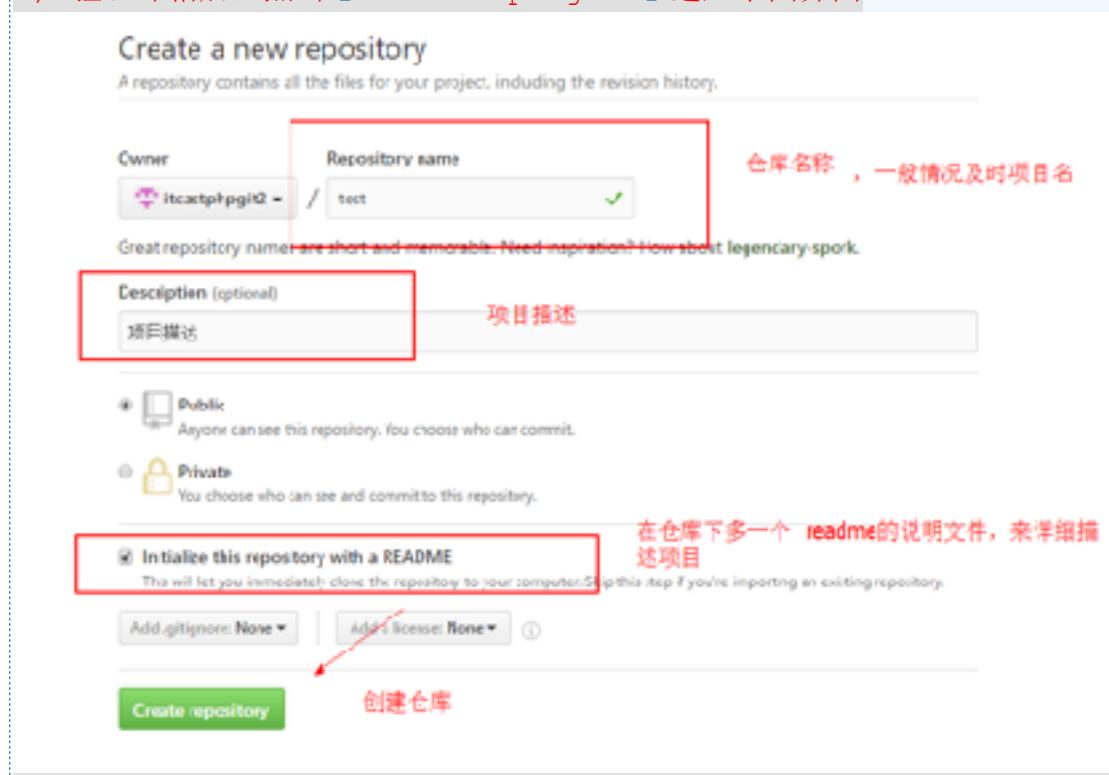
4) 点击【verify email address】验证邮箱

说明：验证成功后会自动跳转github主页，重新点击【Start a project】即可创建仓库

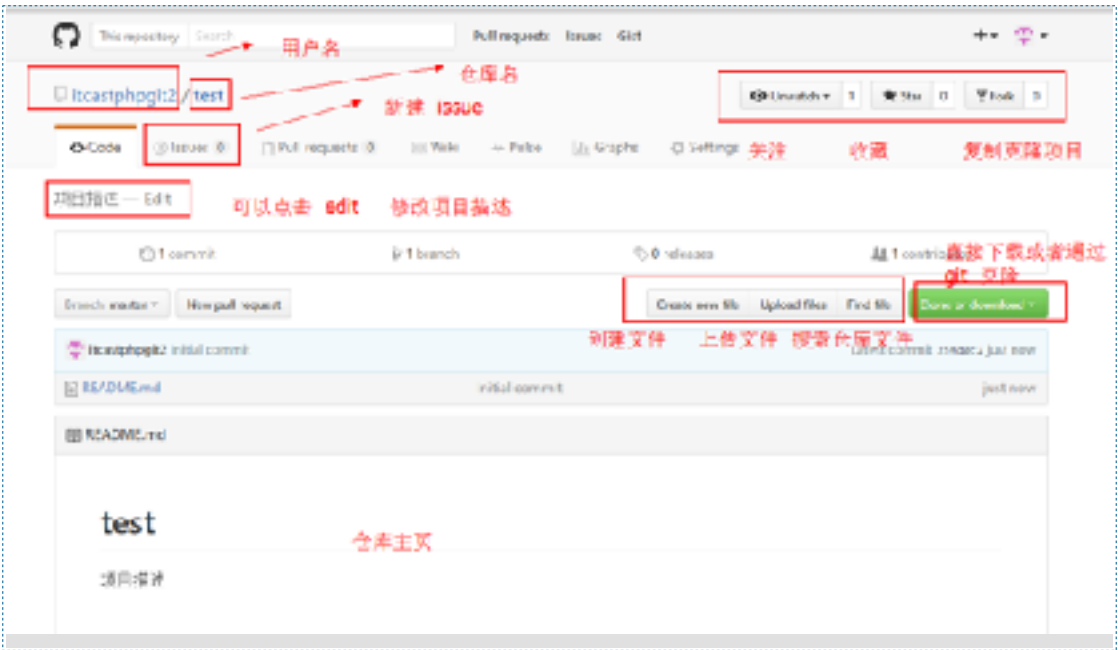




5) 验证邮箱后, 点击【Start a project】进入下图界面



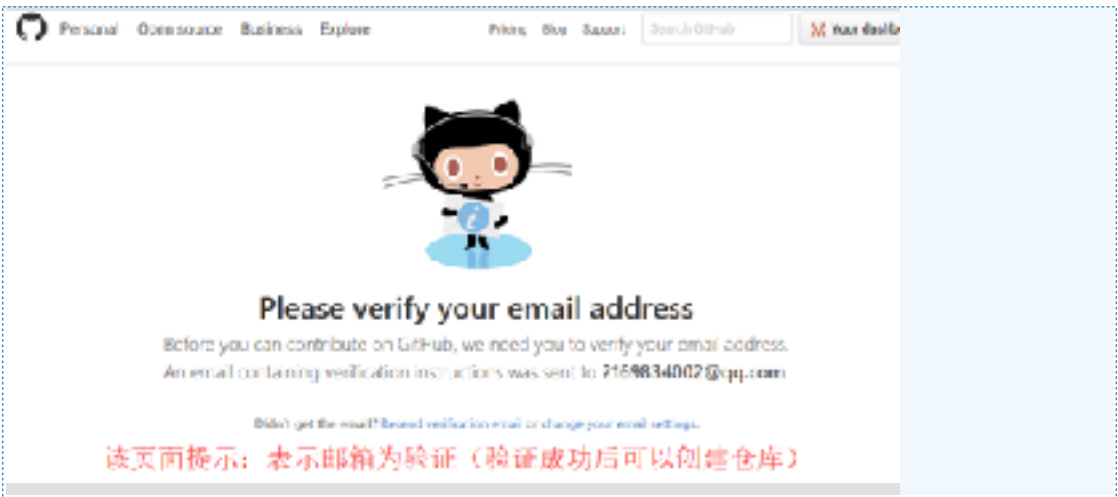
仓库主页说明



脚下留心

qq邮箱需要设置白名单才可以收到邮件

} 未验证邮箱提示页面



## } 设置QQ邮箱白名单

1. 打开QQ邮箱、点击【设置】
2. 点击【反垃圾】
3. 点击【设置域名白名单】
4. 在新页面的input框中输入【github.com】添加即可



## 2.5 仓库管理

### 新建文件

仓库主页，点击【create new file】创建仓库文件



点击创建文件

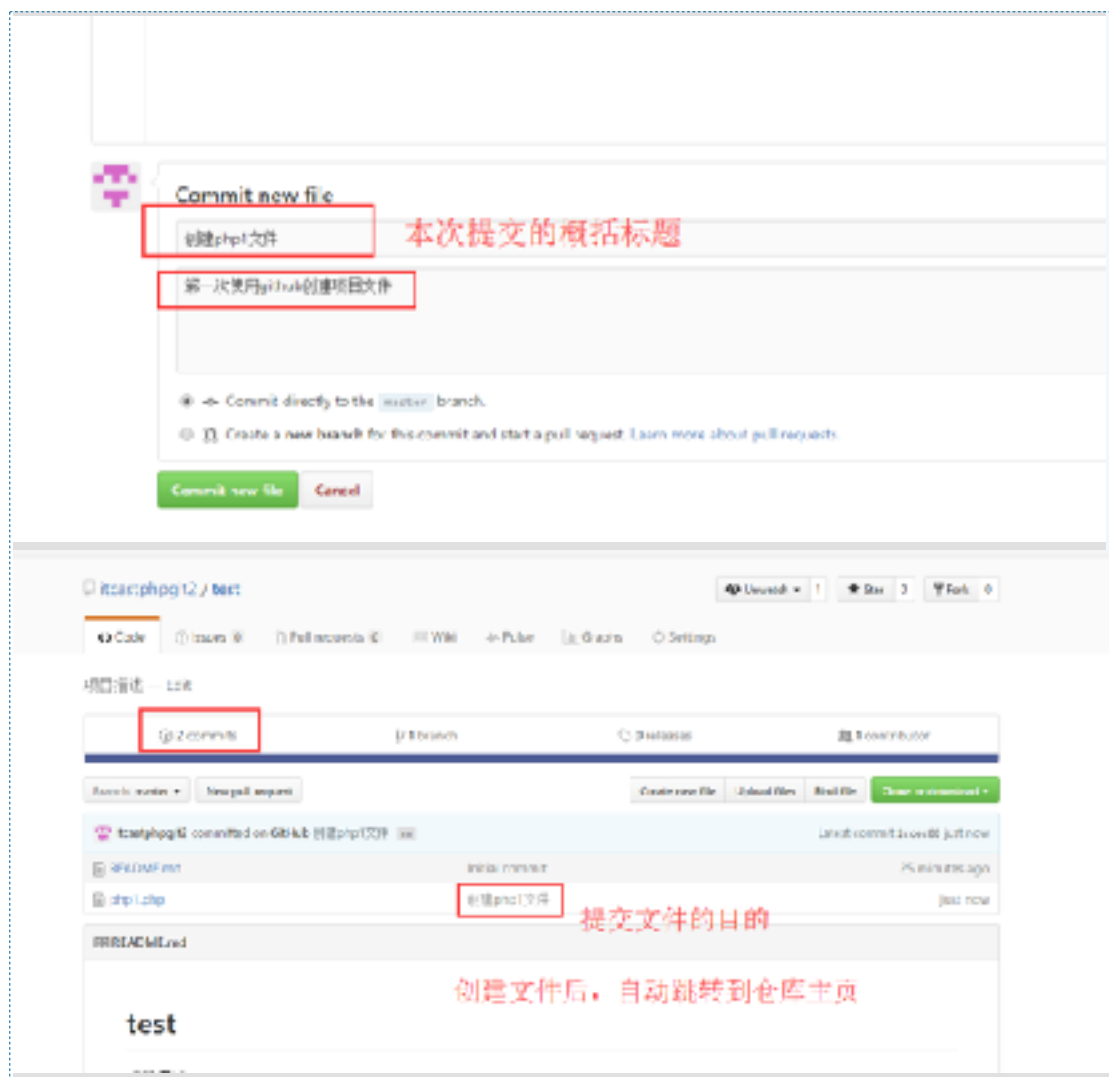
test

test / get.php

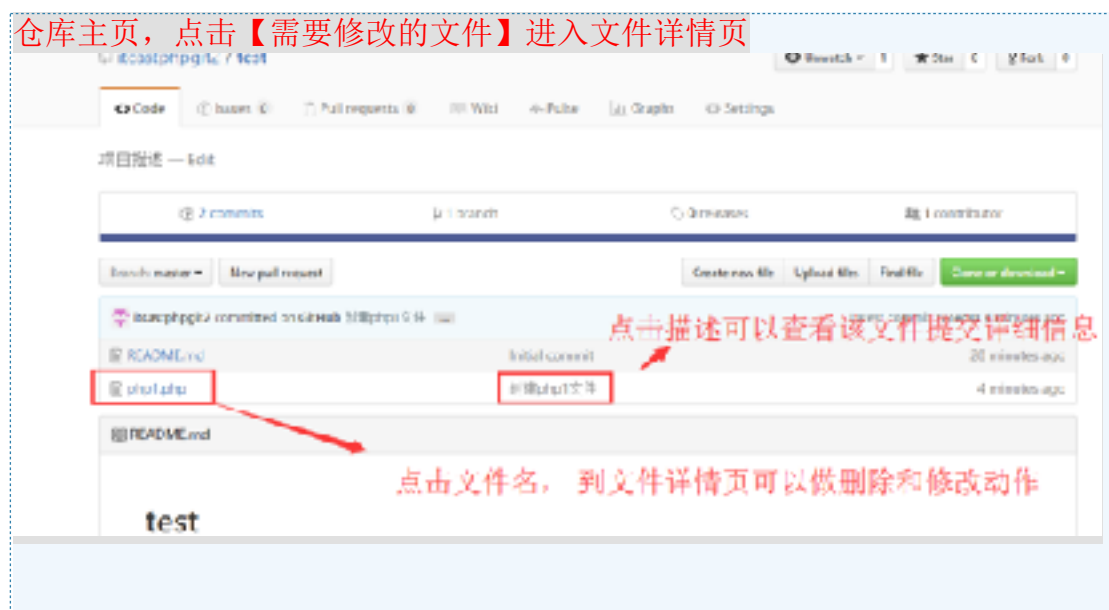
填写文件名

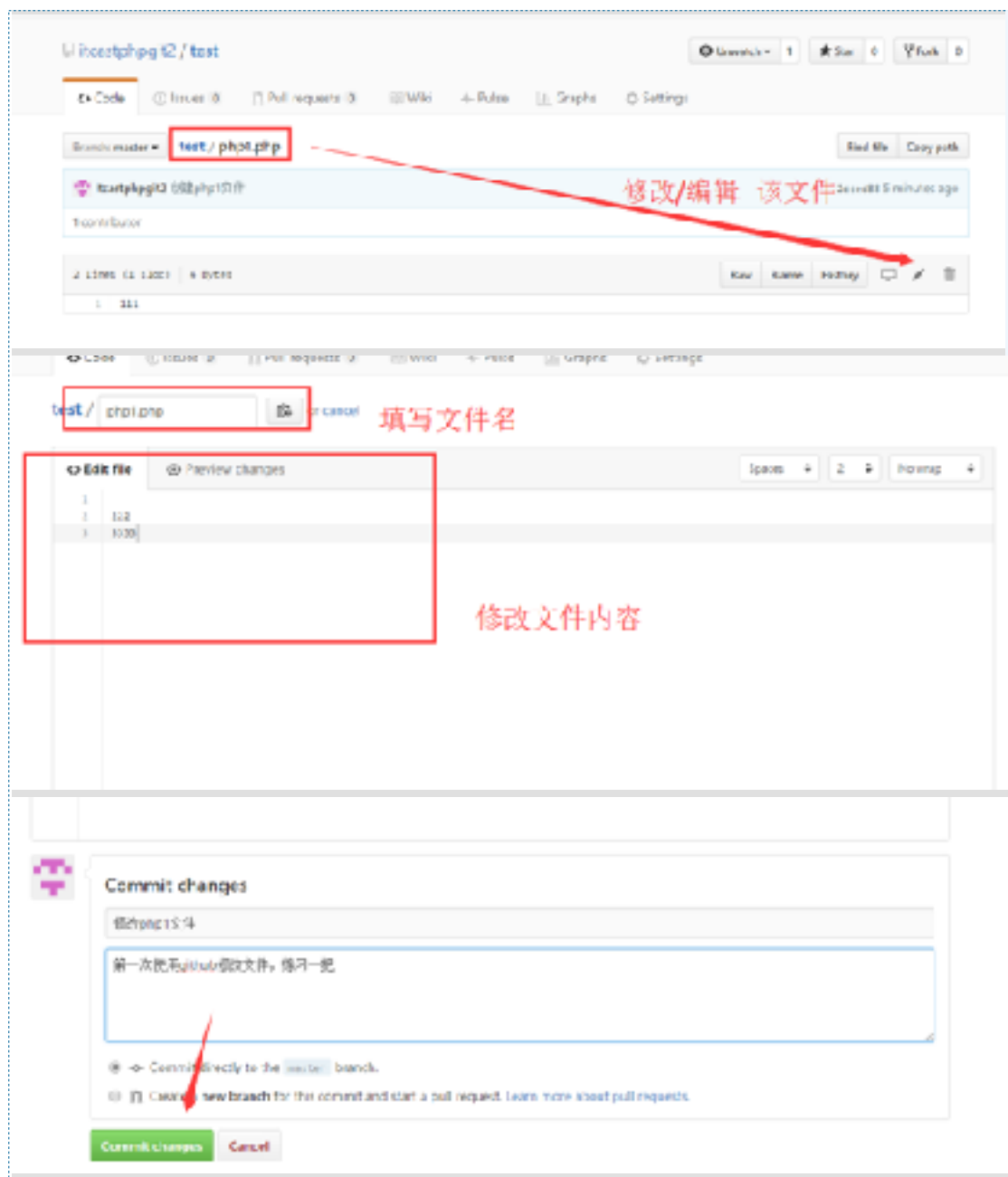
填写文件内容

下面表单将填写每次提交的目的，原因：为了方便其他开发者知道本次添加或修改的原因



## 编辑文件





## 删除文件

项目描述 — Edit

3 commits 1 branch 0 releases

Branch: master New pull request Create new file Upload files

itcastphpgit2 committed or GitHub 添加php1文件 Latest commit

README.md Initial commit

**php1.php** 修改php1文件

README.md

test

点击文件名，进入文件详情页

test / php1.php

Find file Copy path

itcastphpgit2 添加php1文件 1 contributor 1 commit

4 lines 42 sim 16 Bins

File Blob History Edit

test / php1.php

3 lines 30 Bins

Commit changes

删除php1文件

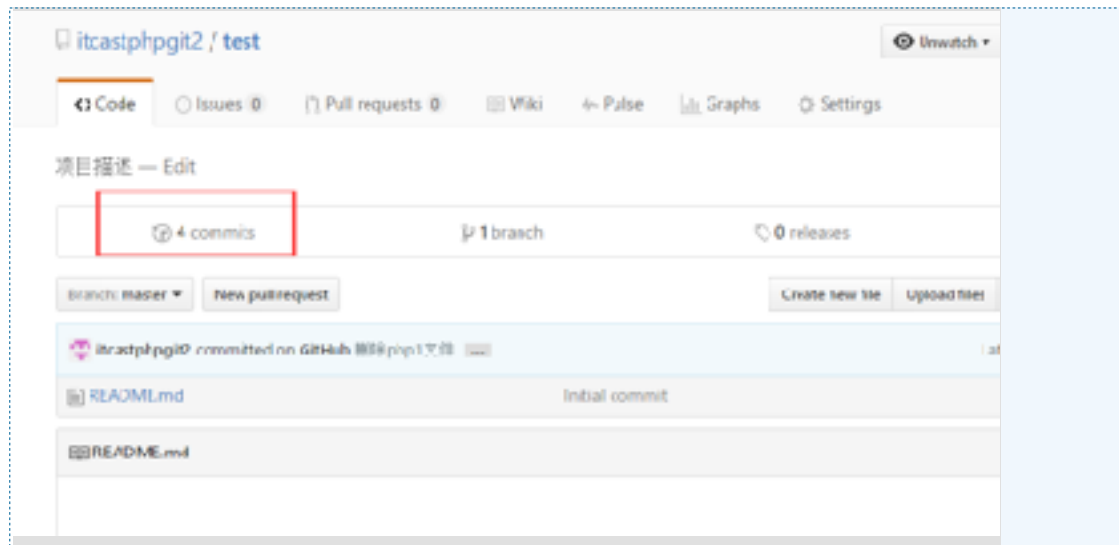
再一次使用github删除文件，练习一遍

Commit directly to the master branch. Create a new branch for this commit and start a pull request. Learn more about pull requests

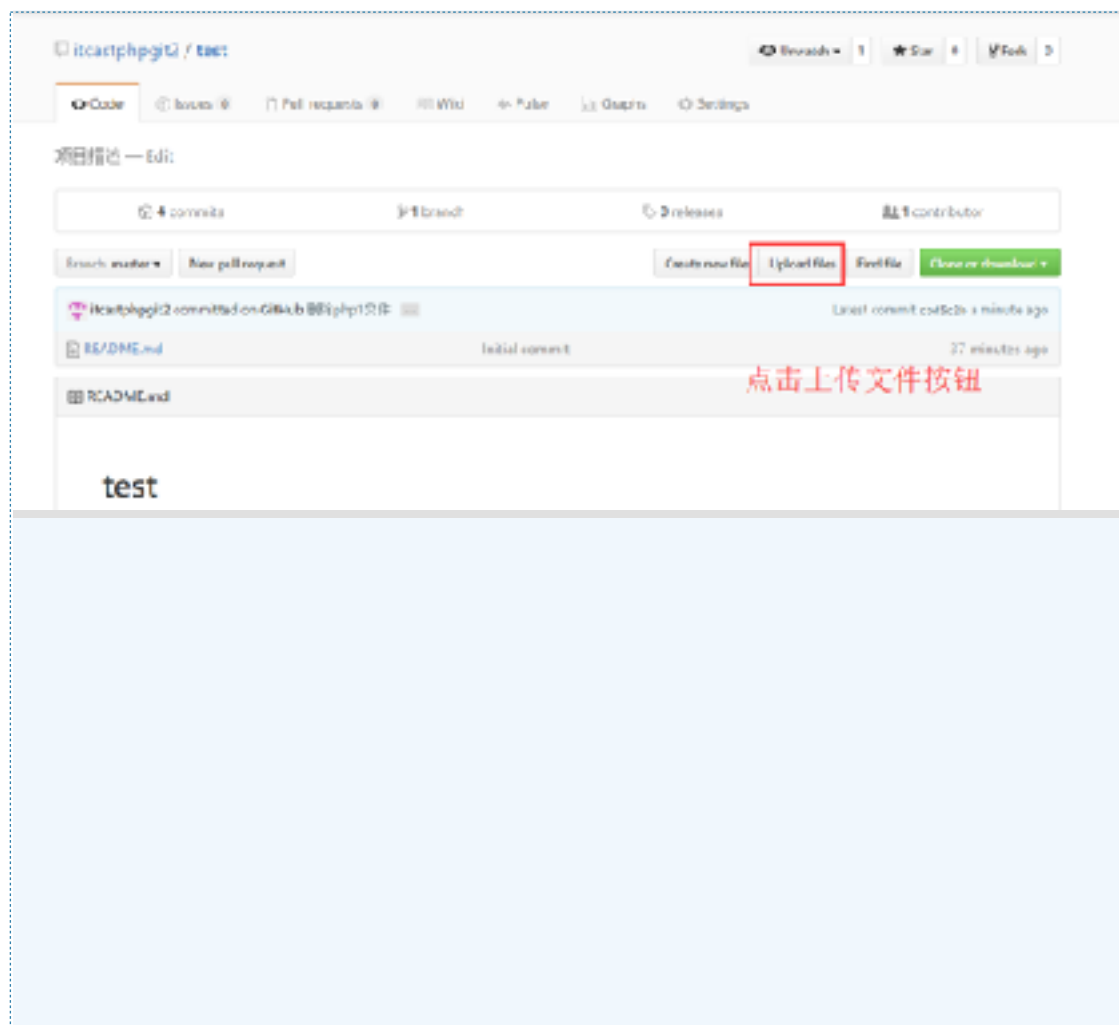
Commit changes Cancel

} 思考被删除文件如何查看信息

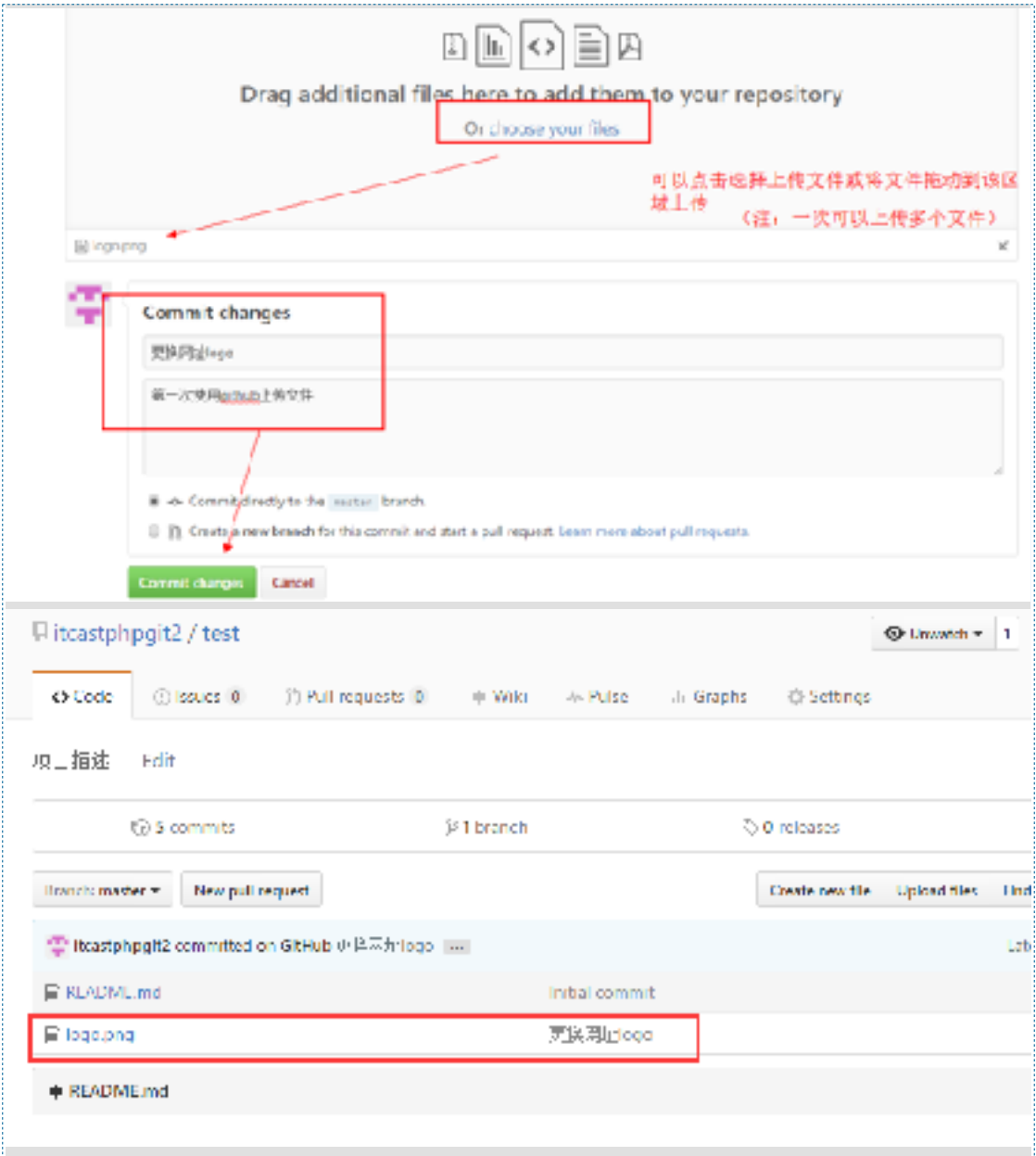
答案：点击commits按钮查看



## 上传文件

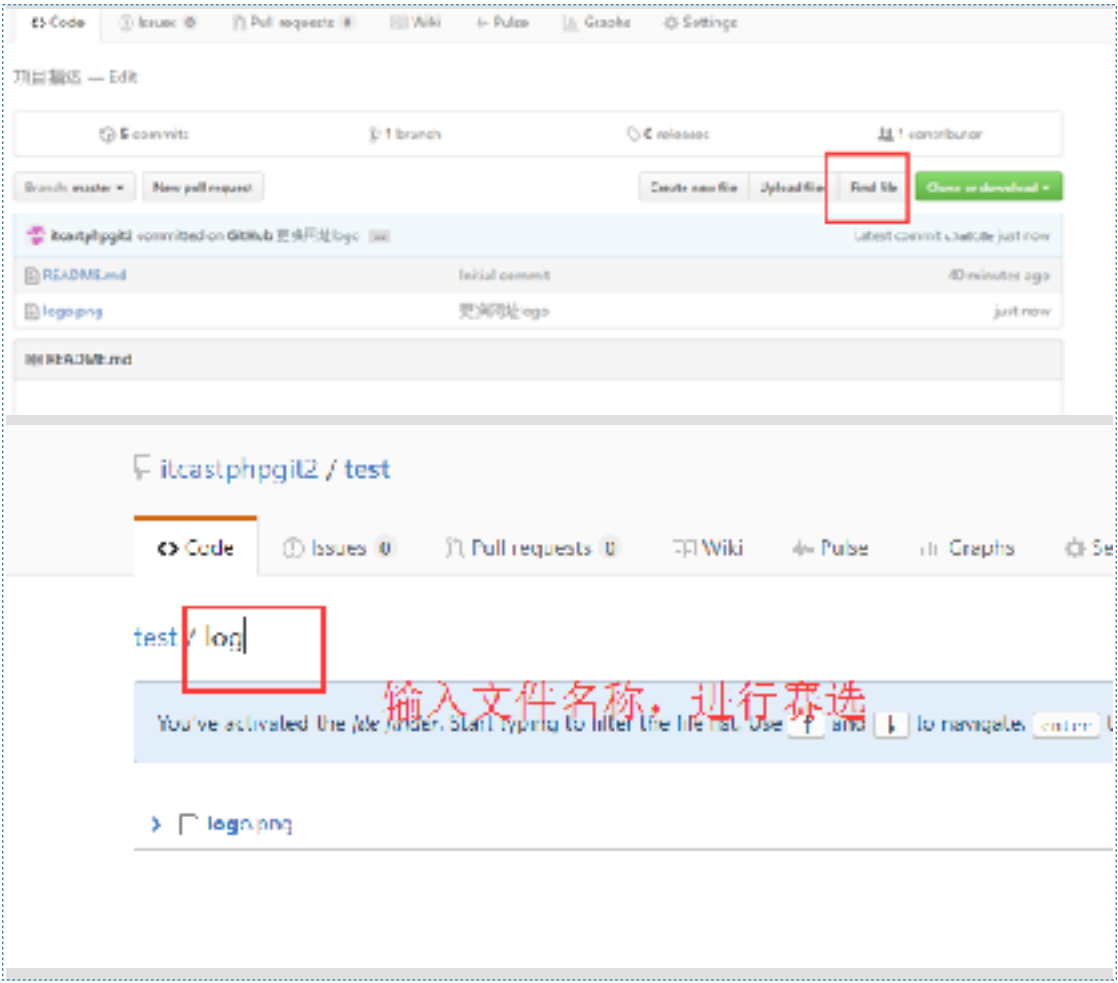




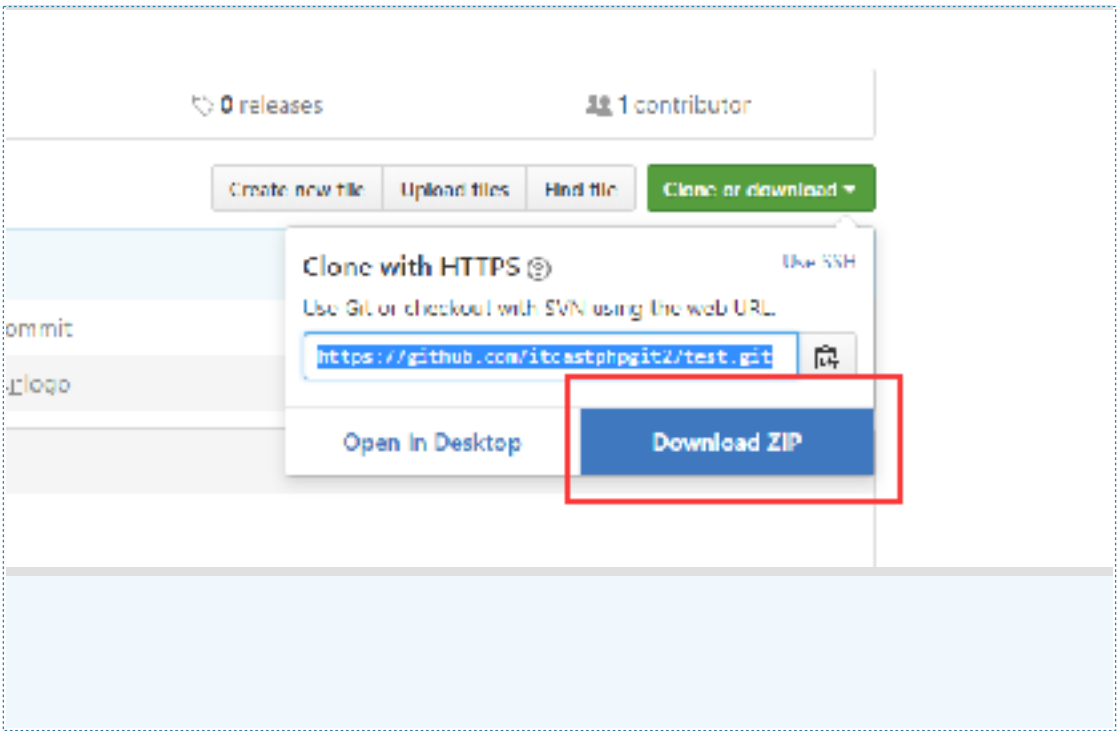


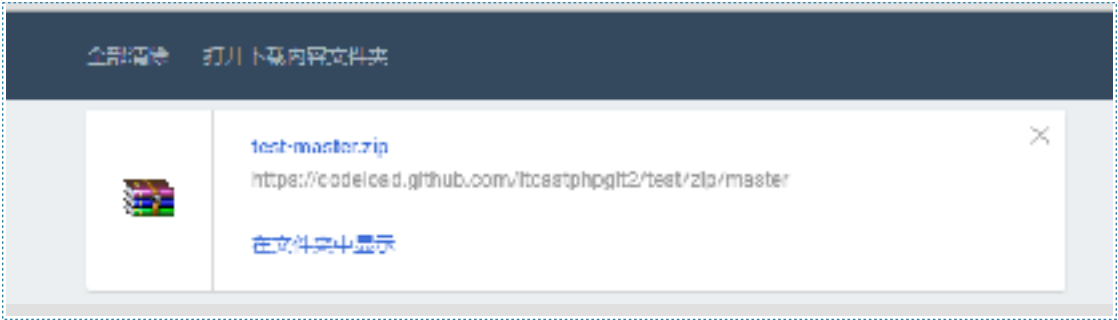
## 搜索仓库文件

多学一招：快捷键（t）



下载/检出项目

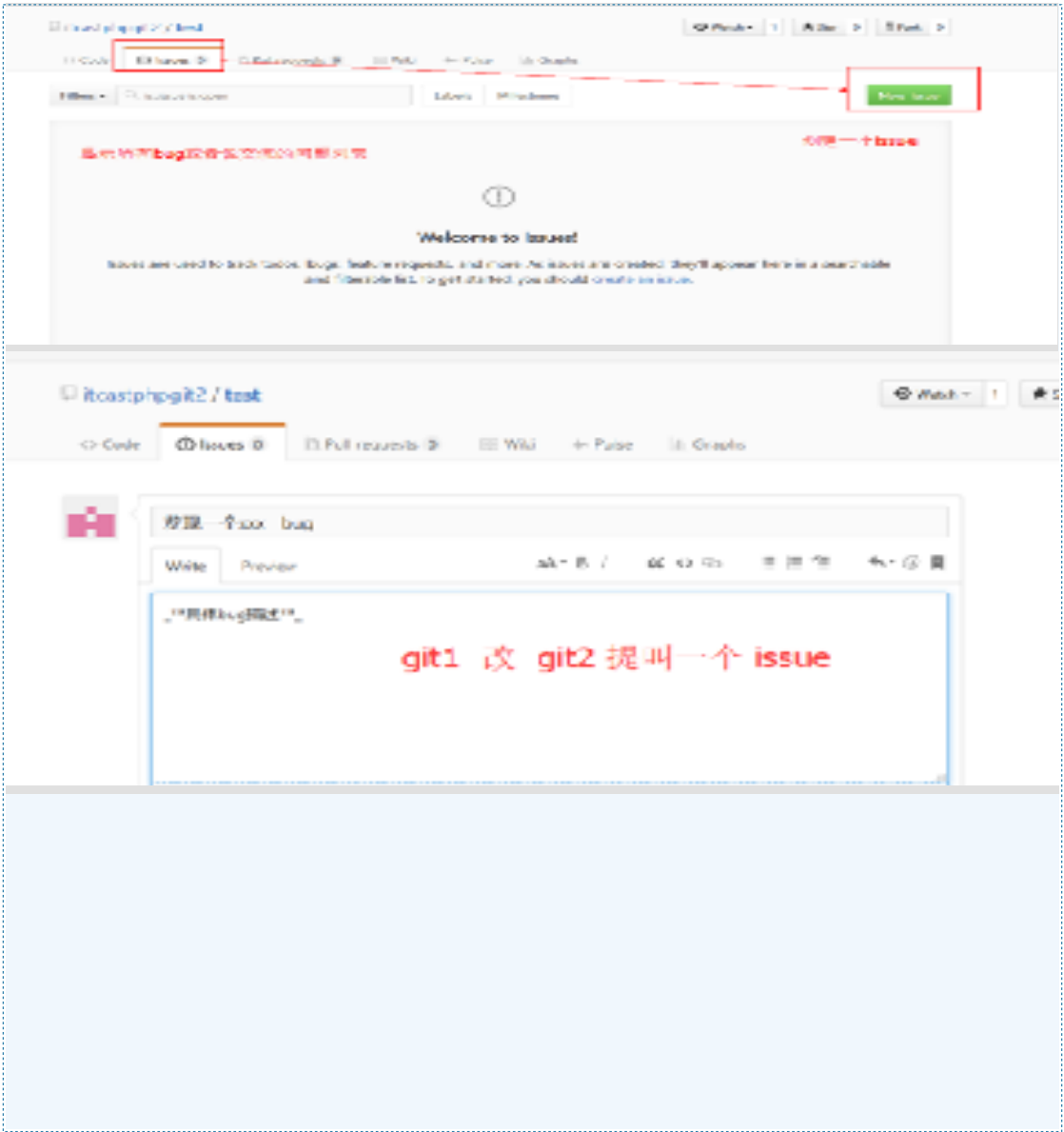


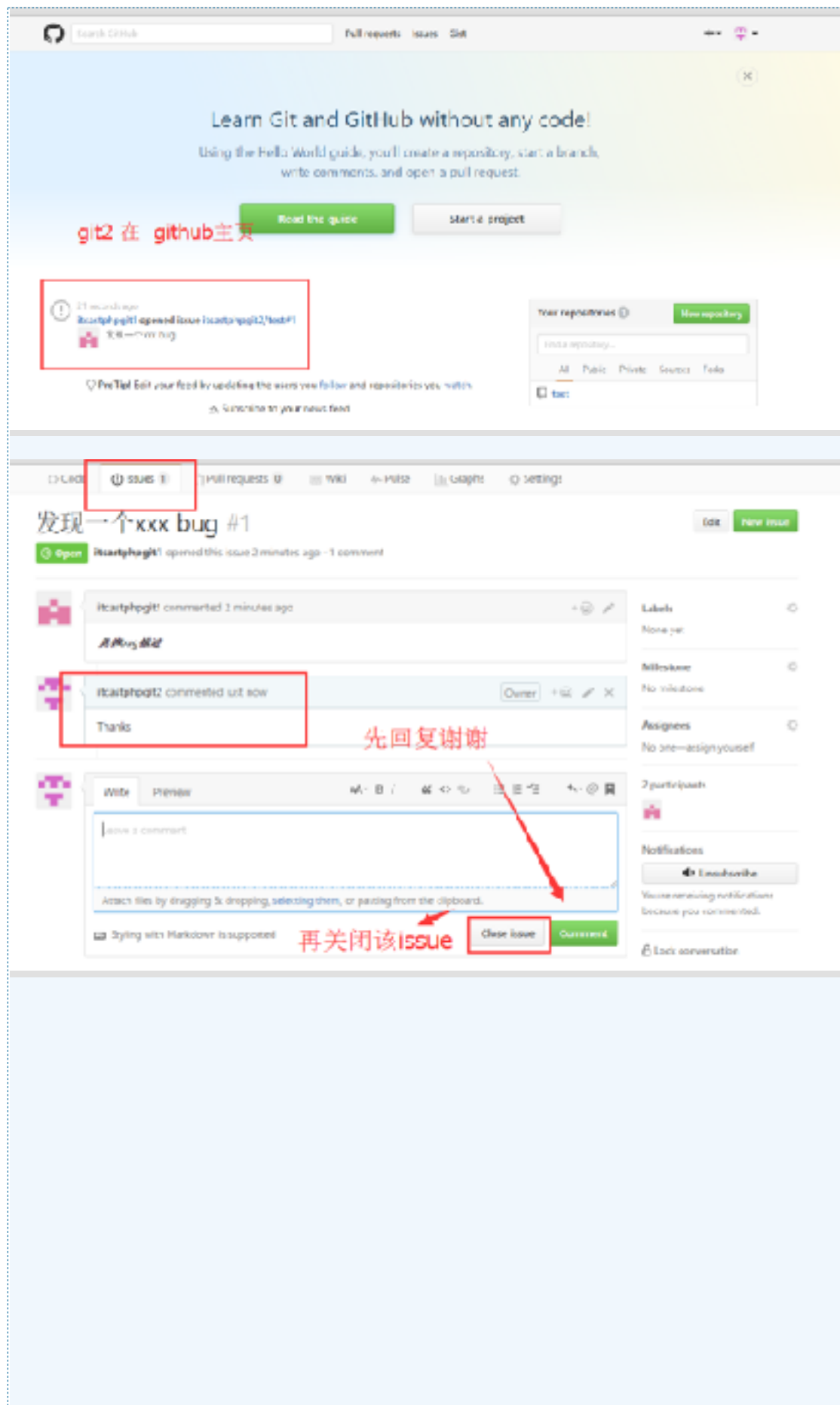


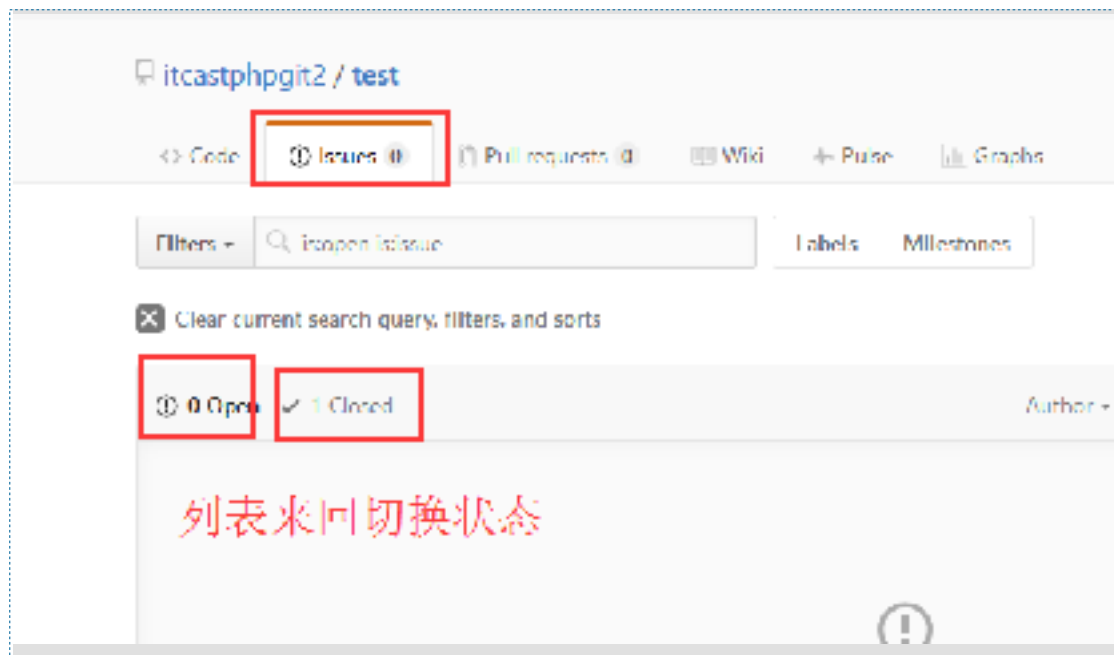
## 2.6 Github Issues

**作用：**发现代码BUG，但是目前没有成型代码，需要讨论时用；或者使用开源项目出现问题时使用

**情景：**张三发现李四开源git库，则发提交了一个issue；李四隔天登录在github主页看到通知并和张三交流，最后关闭issue

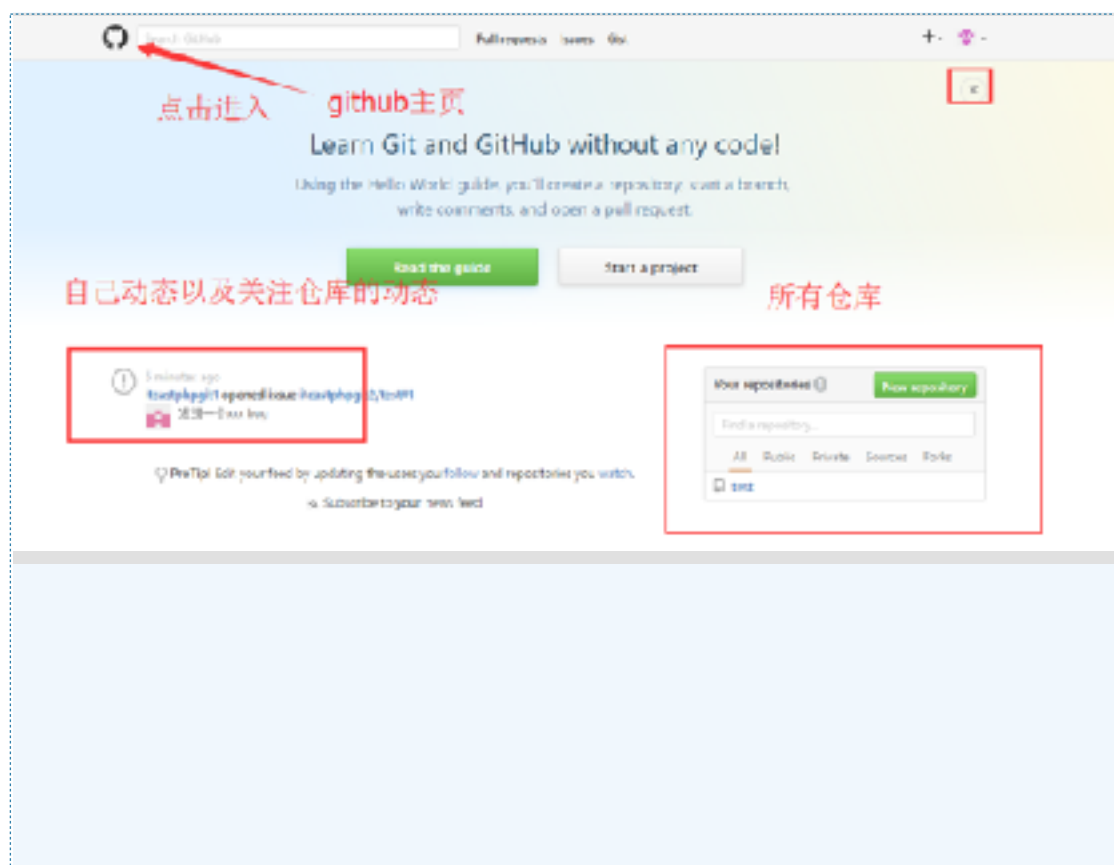




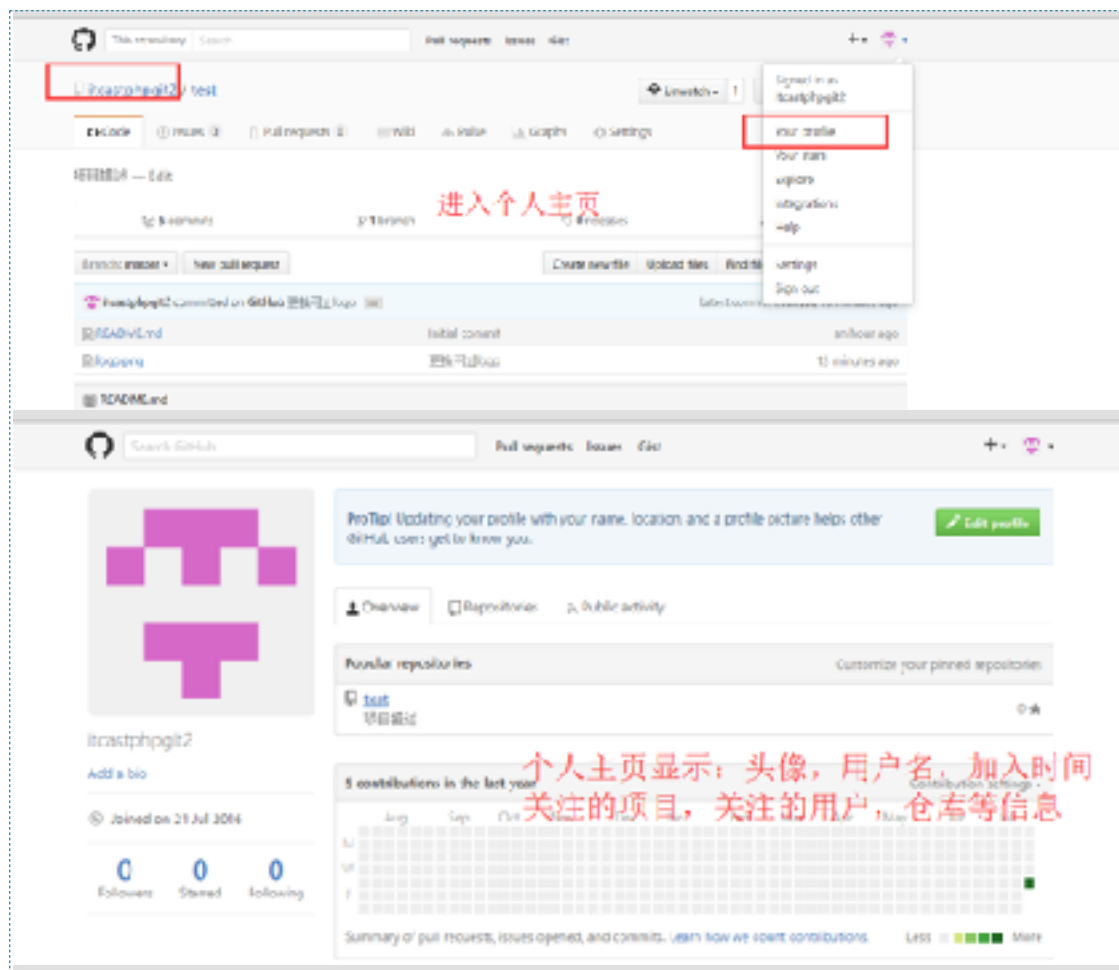


## 2.6 基本概念（实战操作）

### Github主页



## 个人主页

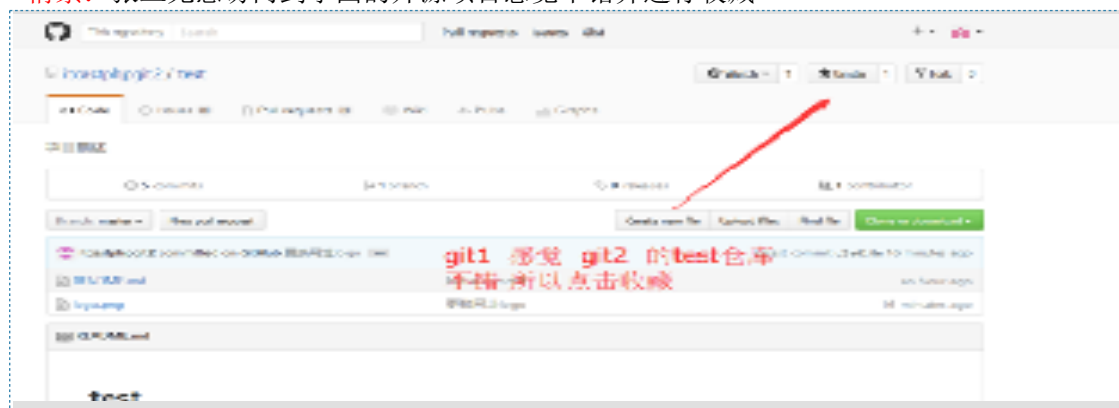


## 收藏（Star）

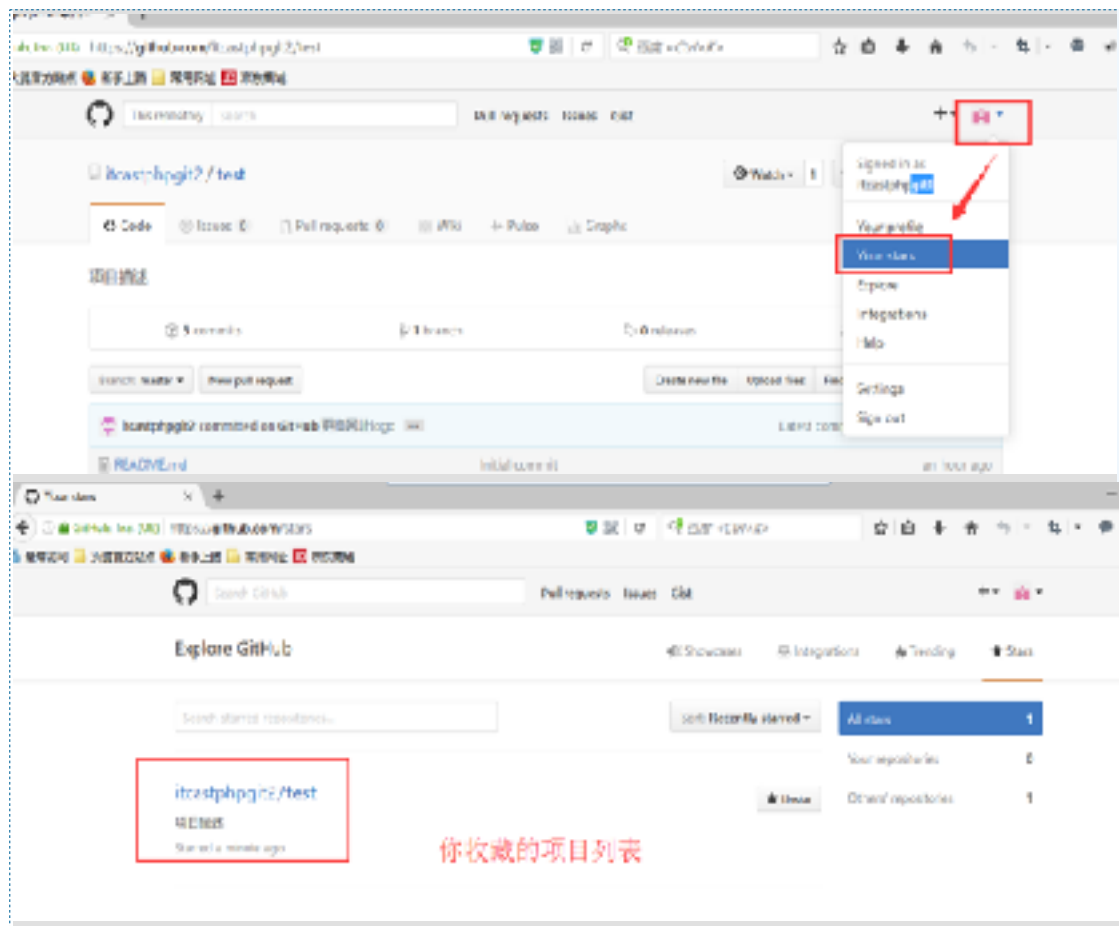
### 【如何收藏】

操作：打开对应项目主页，点击右上角 star 按钮即可收藏

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

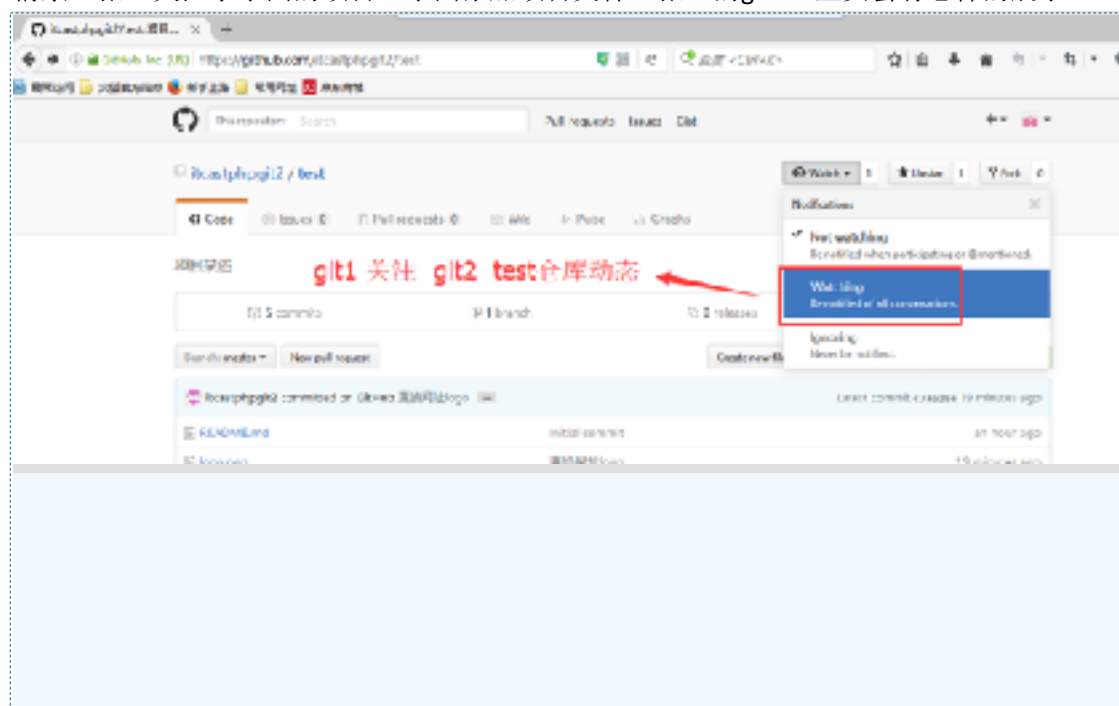


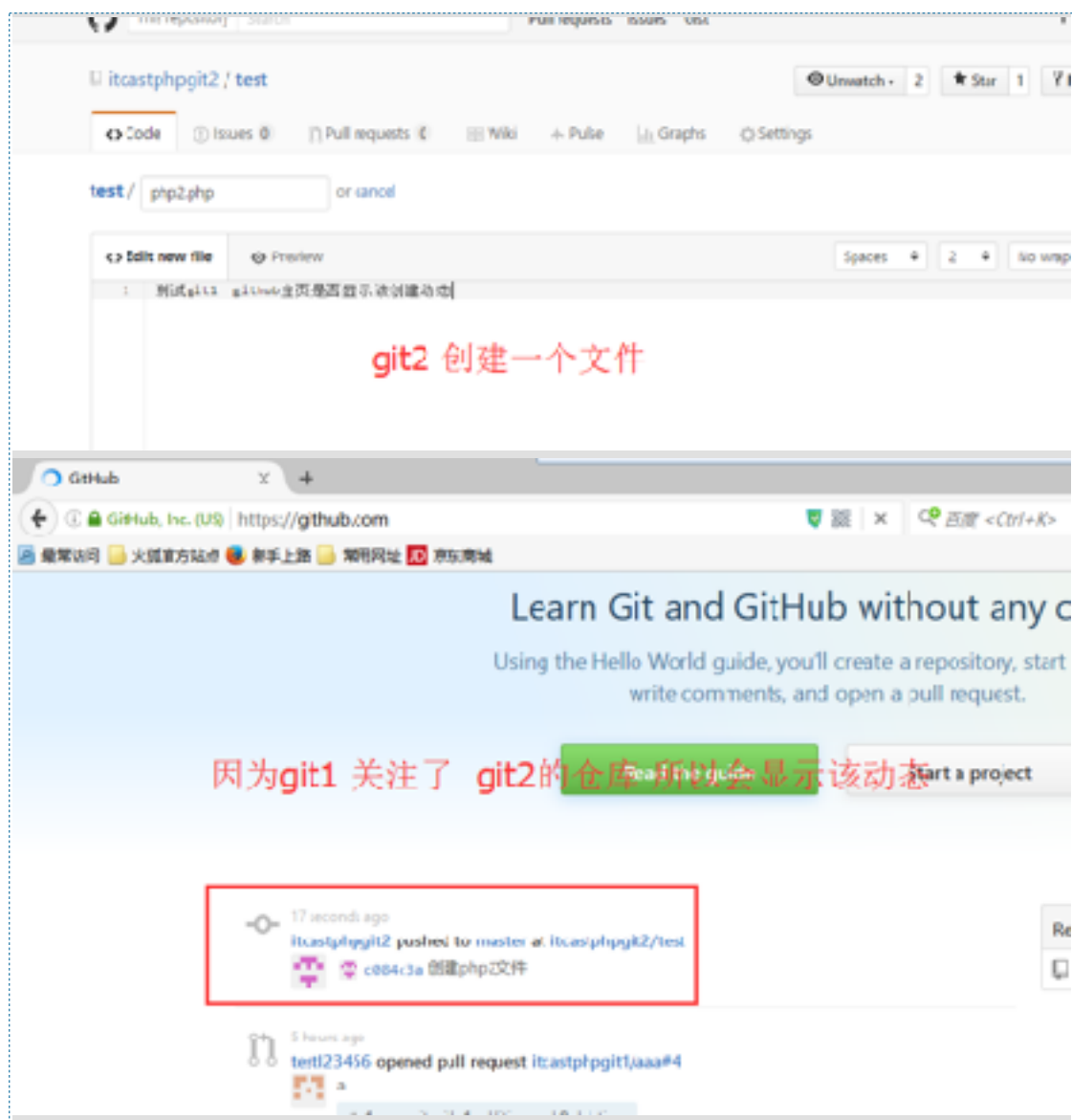
### 【如何查看自己得收藏】



## 关注（Watch）

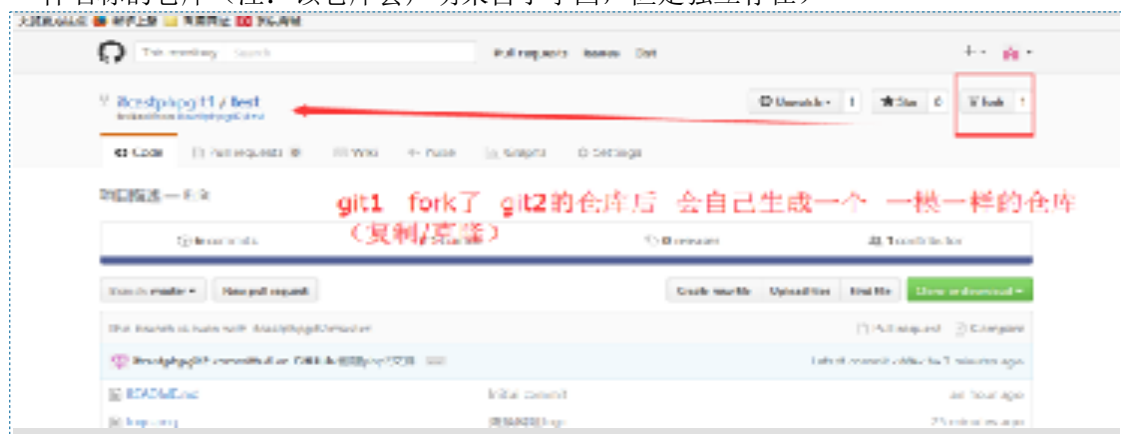
情景：张三关注了李四的项目，李四添加项目文件，张三的github主页会有怎样的展示？





## 复制克隆项目（Fork）

情景：张三fork了李四的项目，相当于张三复制了李四的项目，所以自己也单独有了一个一样名称的仓库（注：该仓库会声明来自于李四，但是独立存在）





验证：fork后的仓库是否单独存在

The top screenshot shows the GitHub interface for the repository `itcastohpgit1 / test`, which is a fork of `itcastphpgit2/test`. The repository has 7 commits and 1 branch. The file list shows `README.md`, `login.php`, `php2.php`, and `test3.php`. A red arrow points to `test3.php`.

The bottom screenshot shows the GitHub interface for the repository `itcastphpgit2 / test`. The repository has 6 commits and 1 branch. The file list shows `README.md`, `login.php`, and `php2.php`. A red box highlights the repository name.

没有test3.php文件，则验证fork后仓库是独立存在的

## 发起请求（Pull Request）

**情景：**张三修改了fork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 request

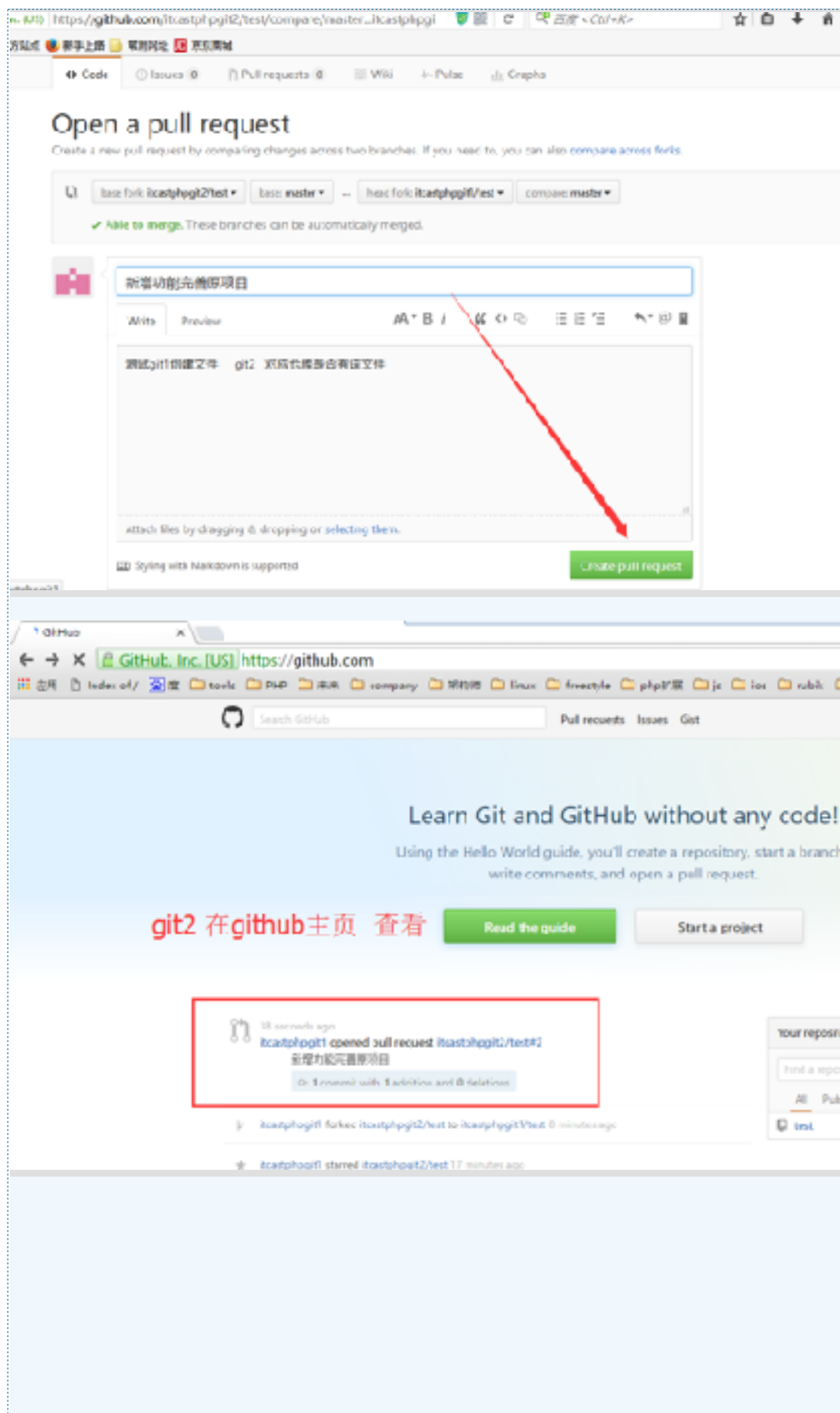
### 1) Git1 发起一个更新请求

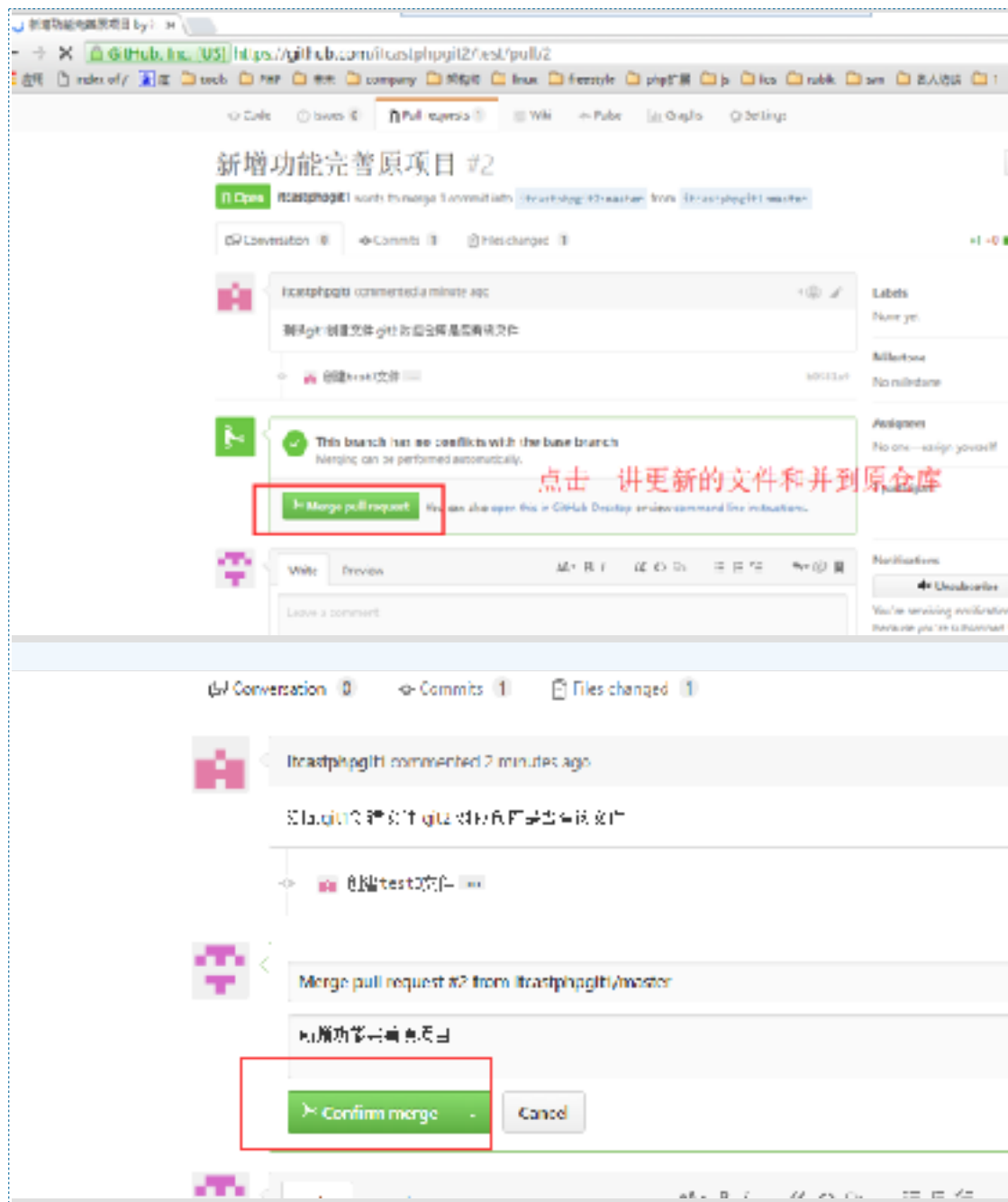
The screenshot illustrates the GitHub interface for creating a Pull Request. The top section shows the repository 'itcastphpgit1 / test' with a red box around 'Pull requests' and a red arrow pointing to 'New pull request'. The bottom section shows the 'Comparing changes' page with a red box around the 'Create pull request' button and a red arrow pointing to it. The bottom section also shows a list of changes with a red box around the details of the first change.

**发起请求**

**创建一个 pull request**

**详细修改信息**





## 2.7 开源项目贡献流程

### 1.新建Issue

提交使用问题或者建议或者想法

### 2.Pull Request

步骤:

- 1.fork项目
- 2.修改自己仓库的项目代码
- 3.新建 pull request
- 4.等待作者操作审核

## 3.Git安装和使用

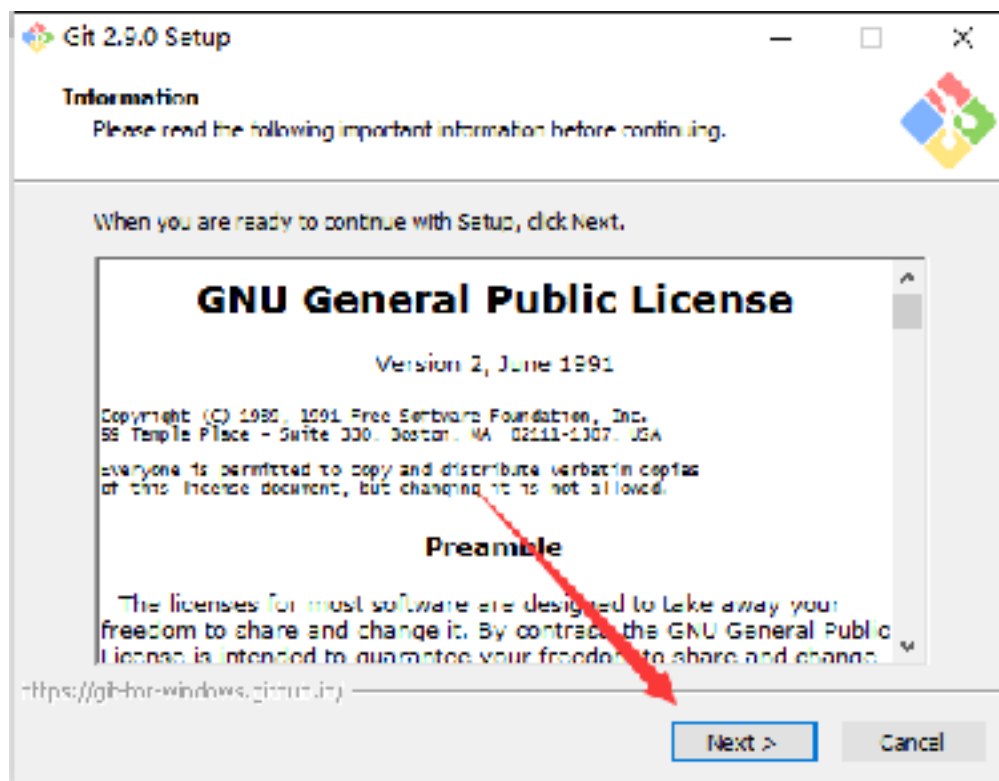
### 目的

通过git管理github托管项目代码

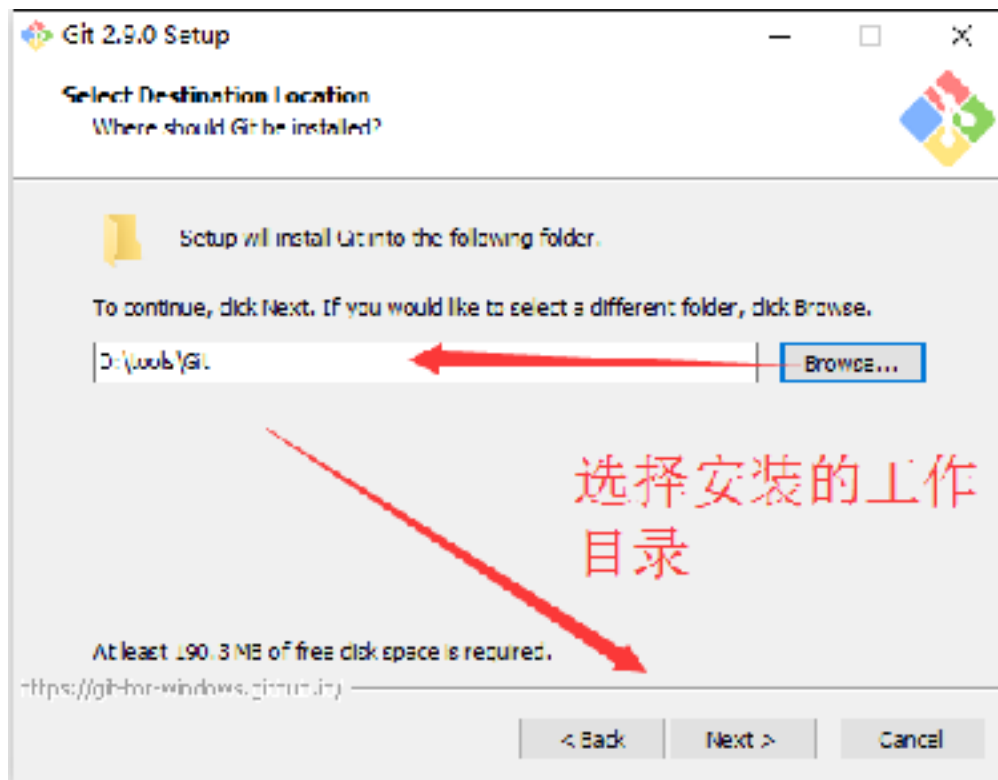
## 下载安装

1.Git官网下载: <https://www.git-scm.com/download/win>

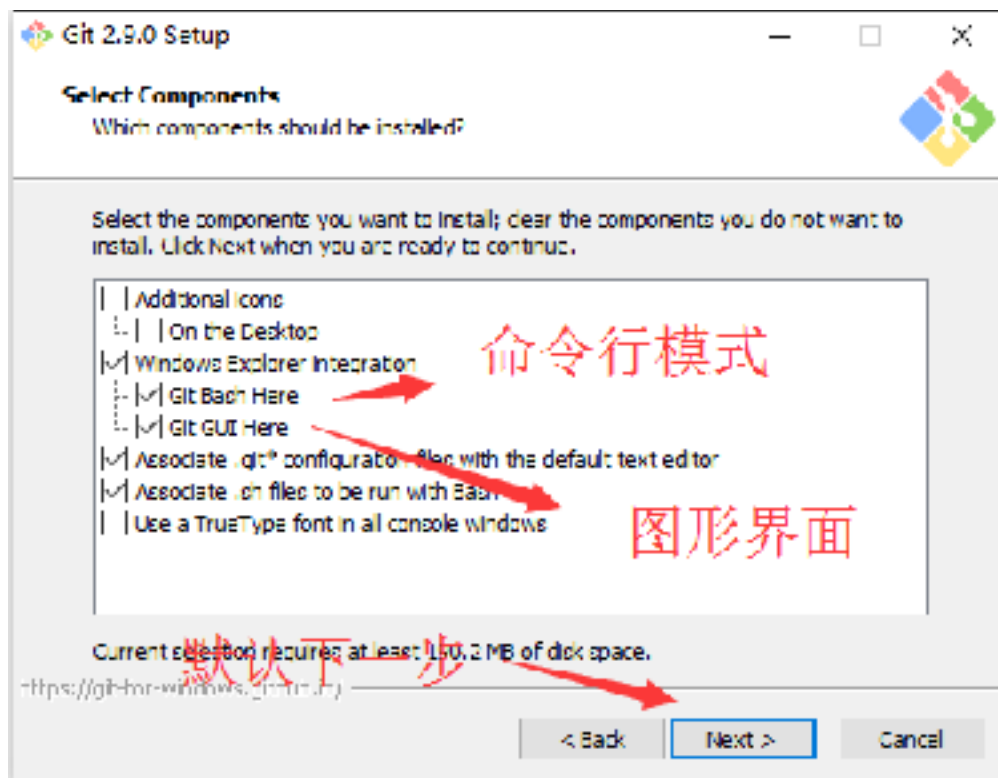
2) 双击安装



### 3) 选择安装的工作目录

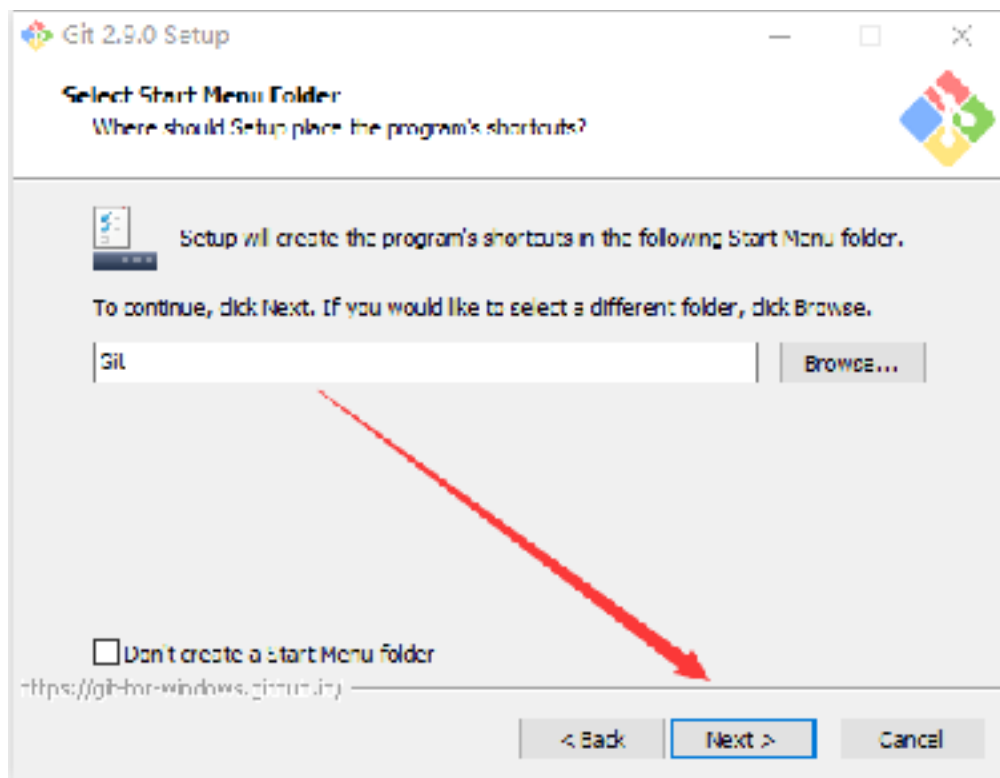


## 4.选择组件

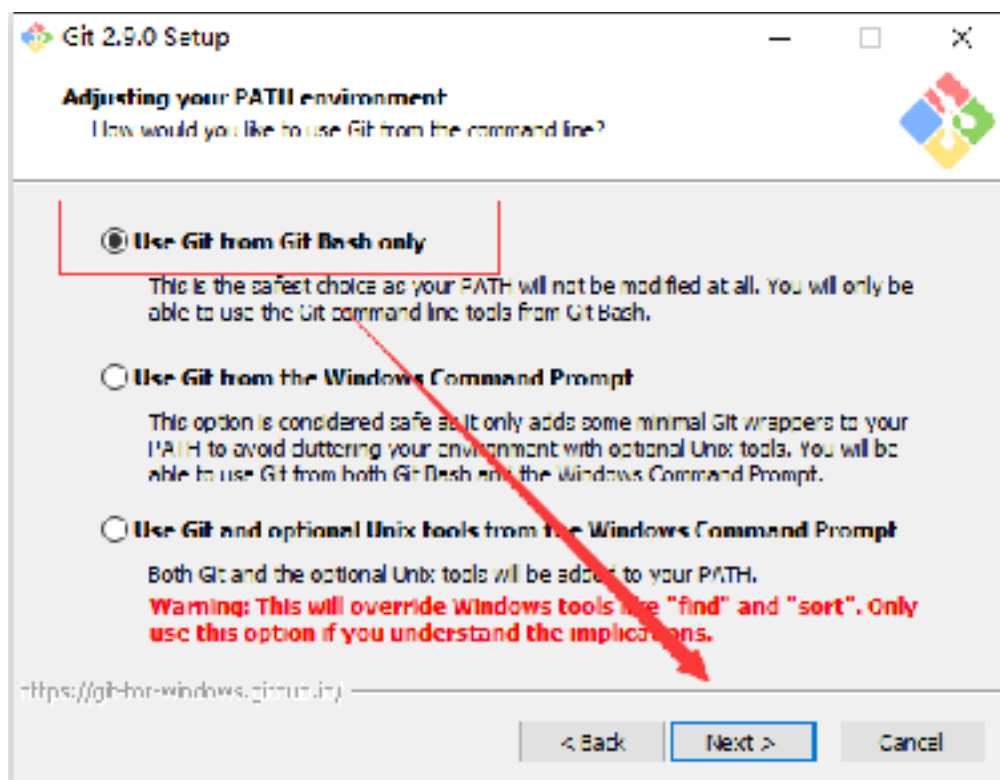




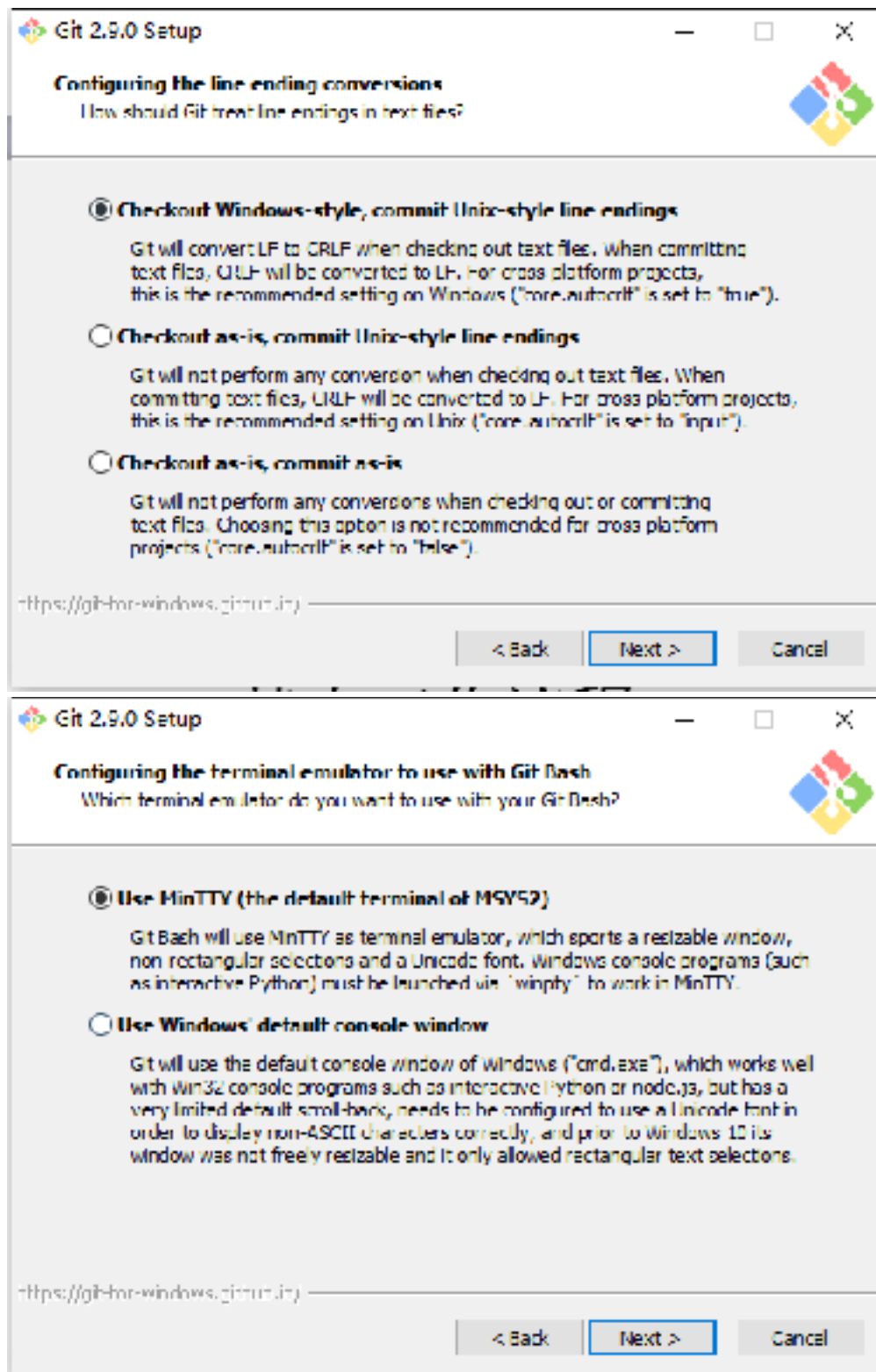
## 5.开始菜单目录名设置

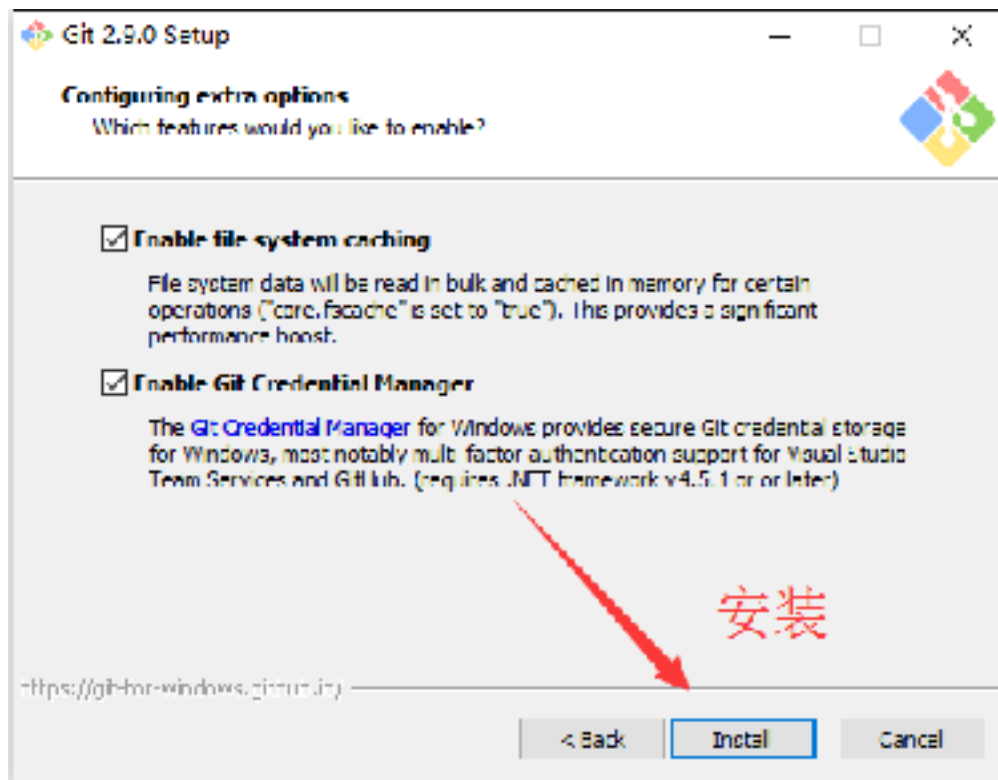


## 6.选择使用命令行环境

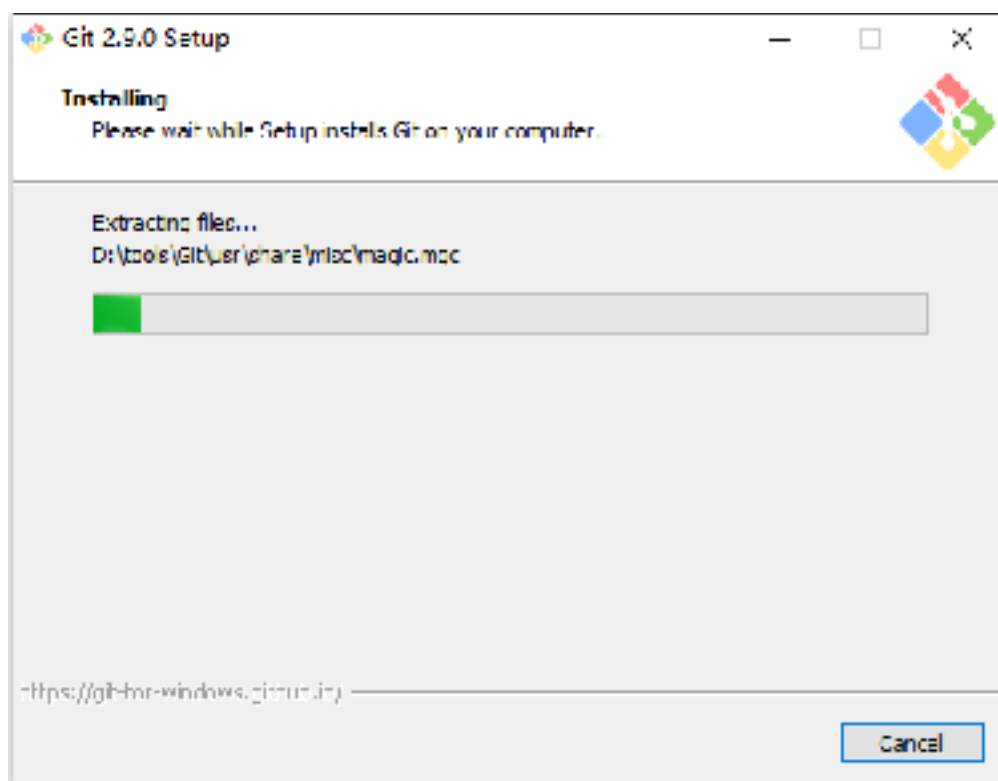


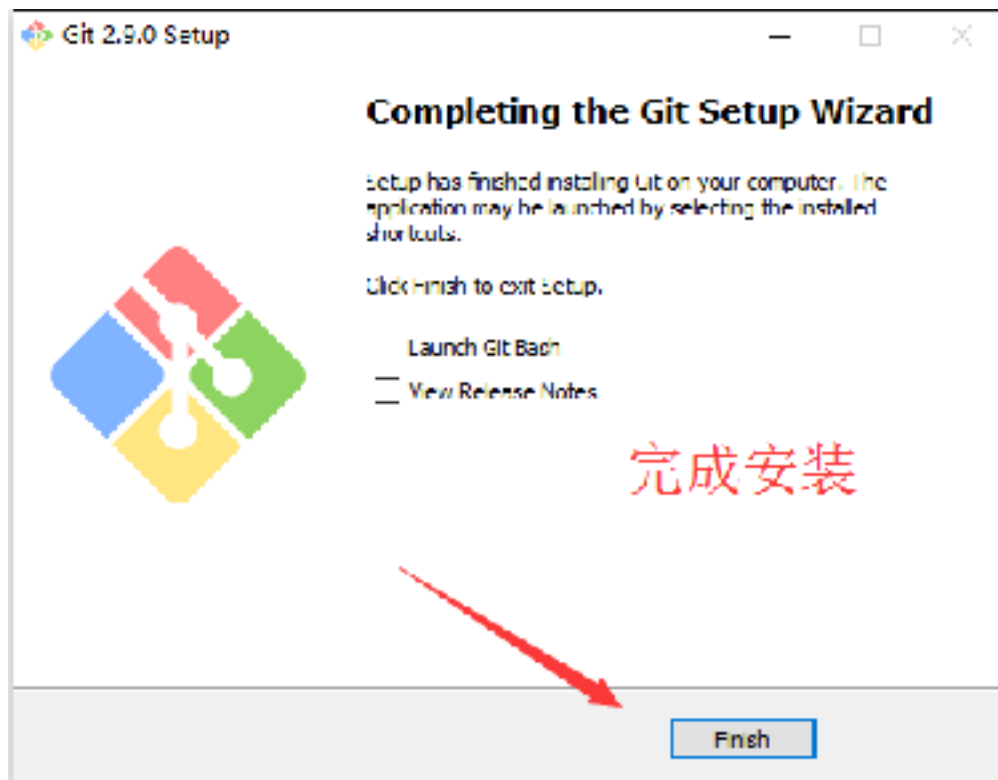
## 7.以下三步默认，直接点击next



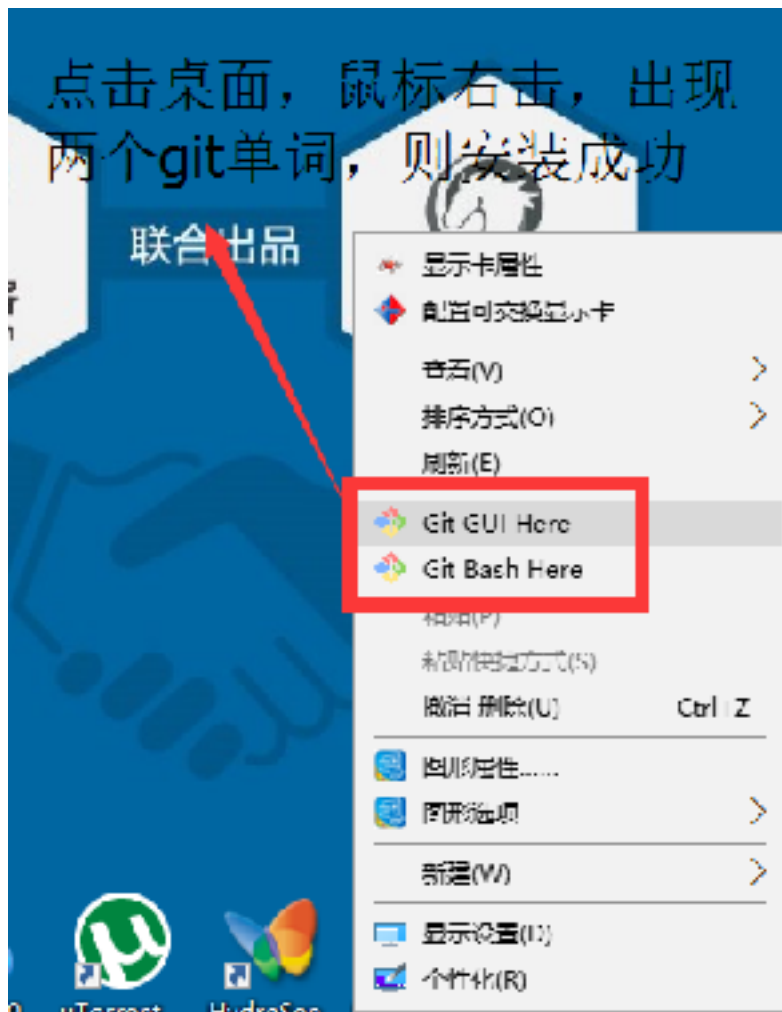


## 8) 等待安装



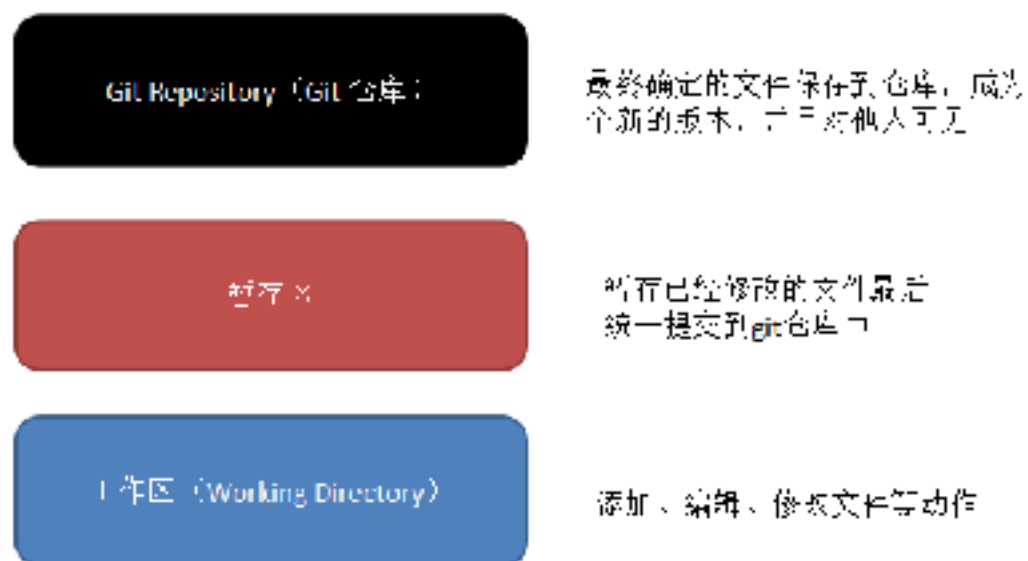


## 9. 检验是否安装成功

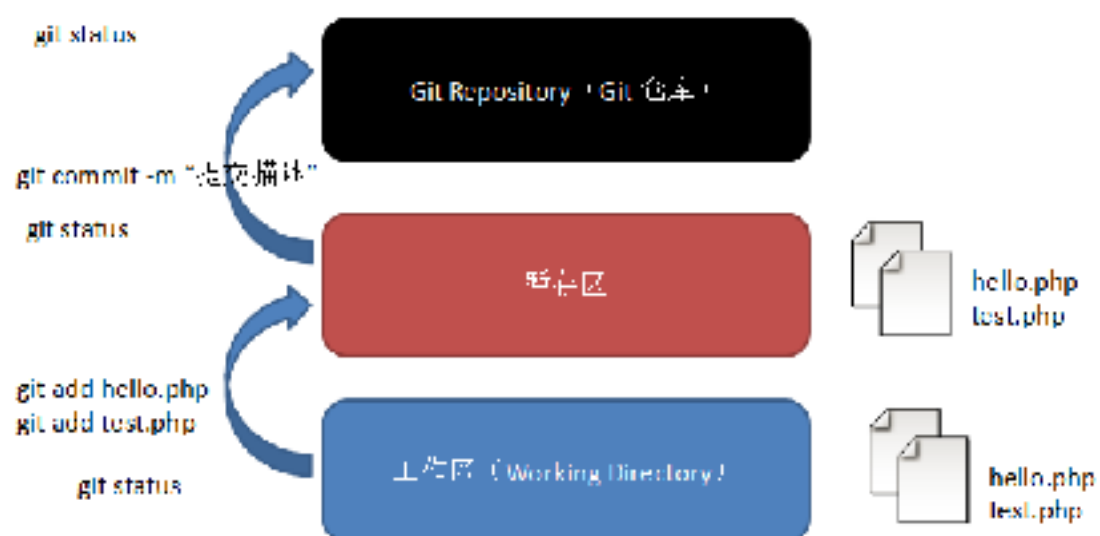


## 4.Git基本工作流程

### Git工作区域



### 向仓库中添加文件流程



## 5.Git初始化及仓库创建和操作

### 基本信息设置

#### 1. 设置用户名

```
git config --global user.name 'itcastphpgit1'
```

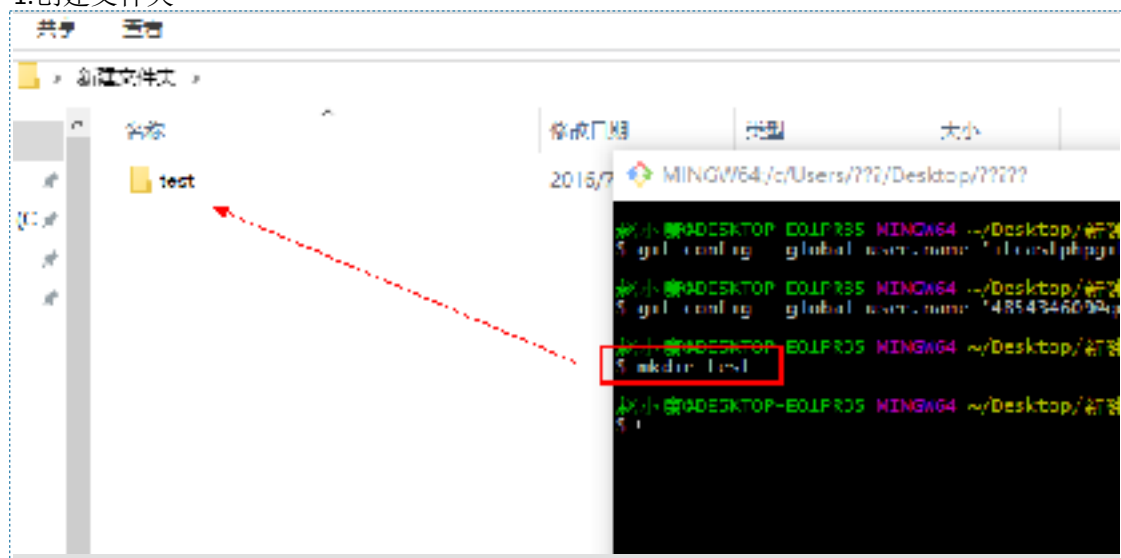
#### 2. 设置用户名邮箱

```
git config --global user.email '485434609@qq.com'
```

脚下留心：该设置在github仓库主页显示谁提交了该文件

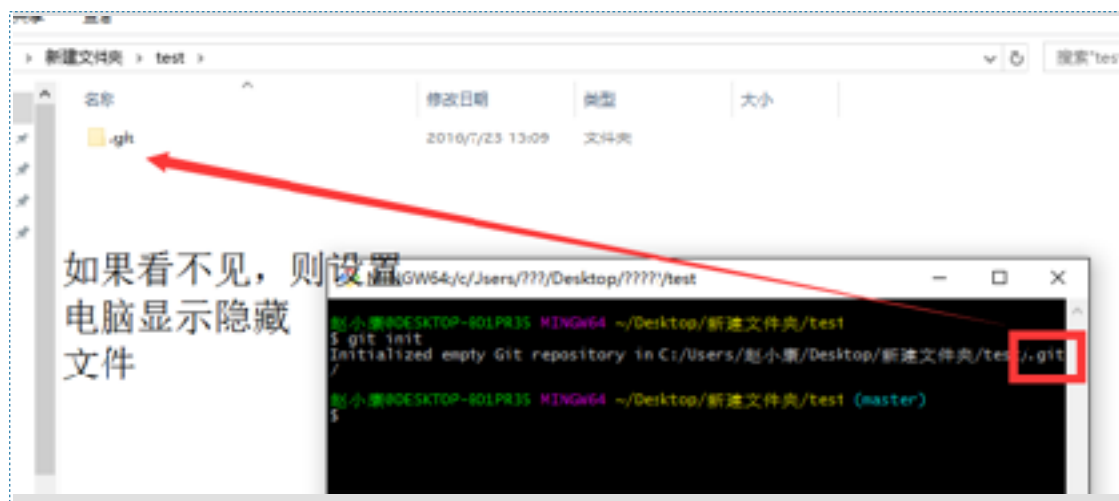
### 初始化一个新的Git仓库

#### 1.创建文件夹

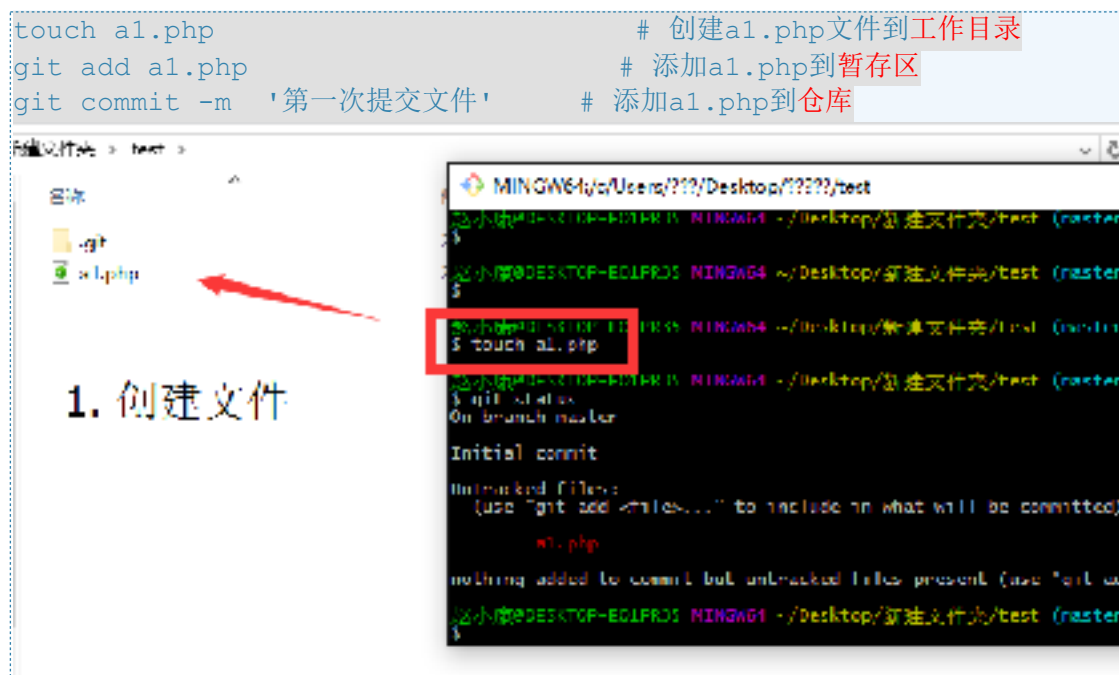


#### 2.在文件内初始化git（创建git仓库）

```
cd test  
git init
```



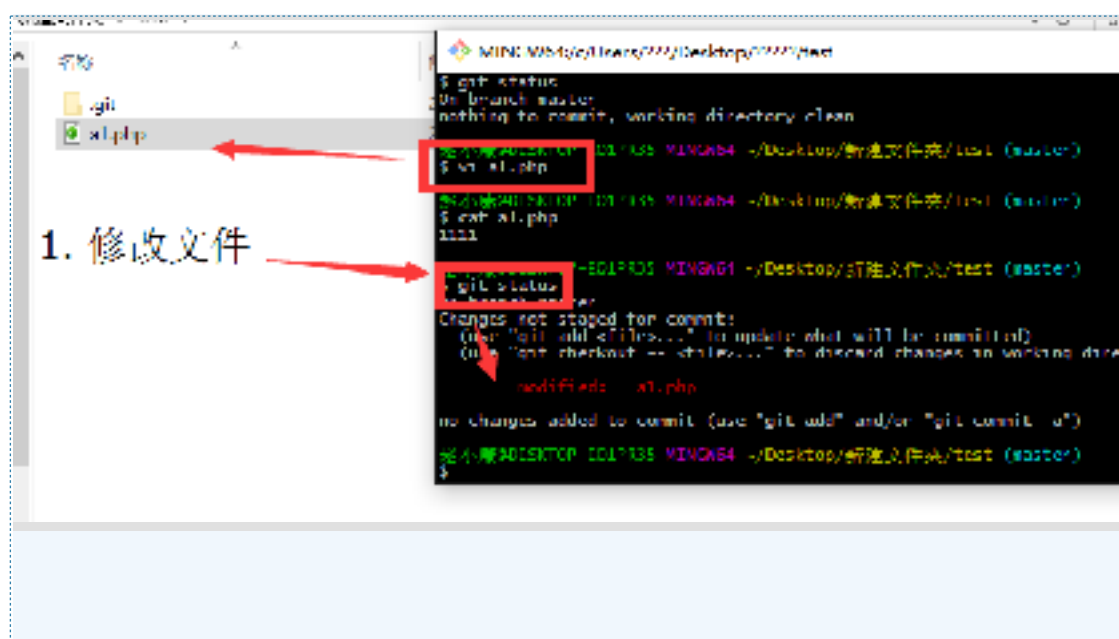
## 向仓库添加文件

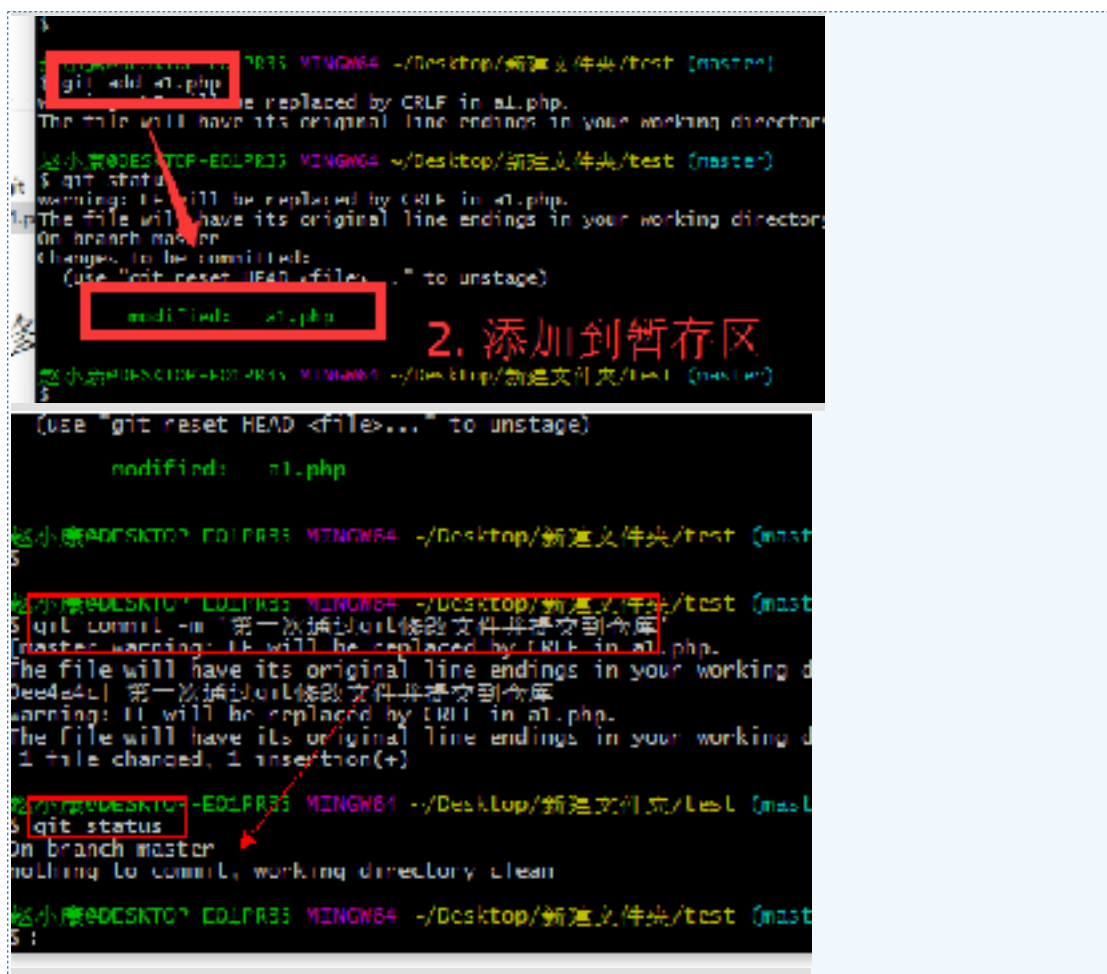




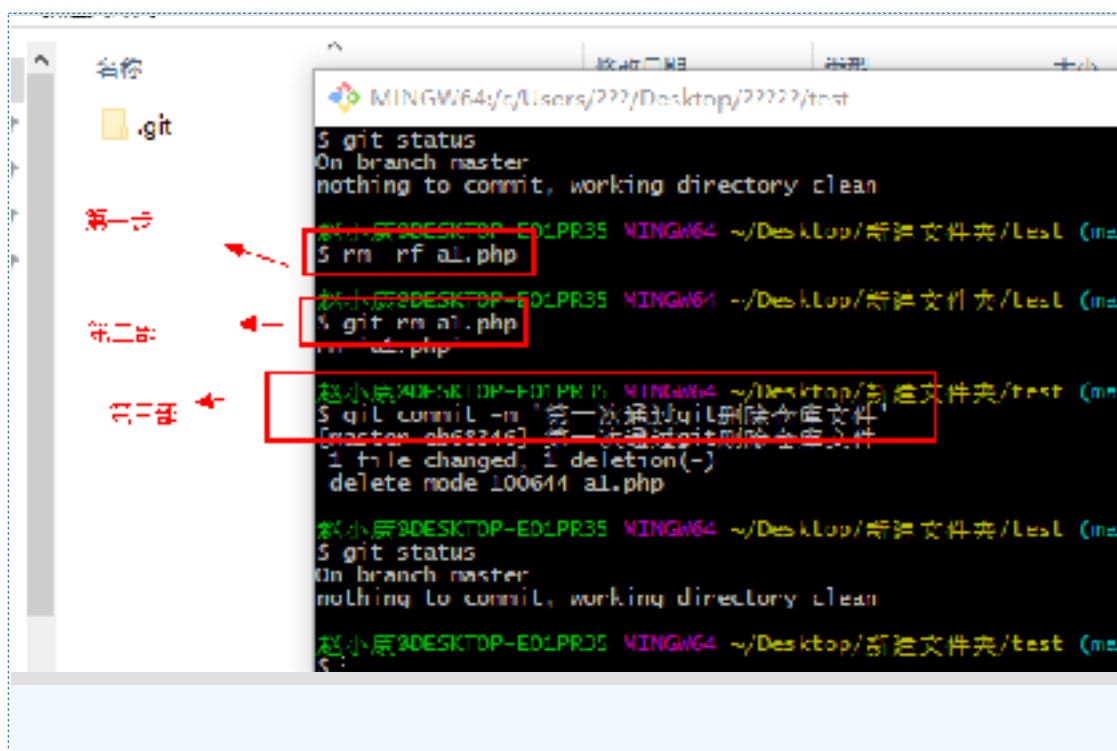


## 修改仓库文件





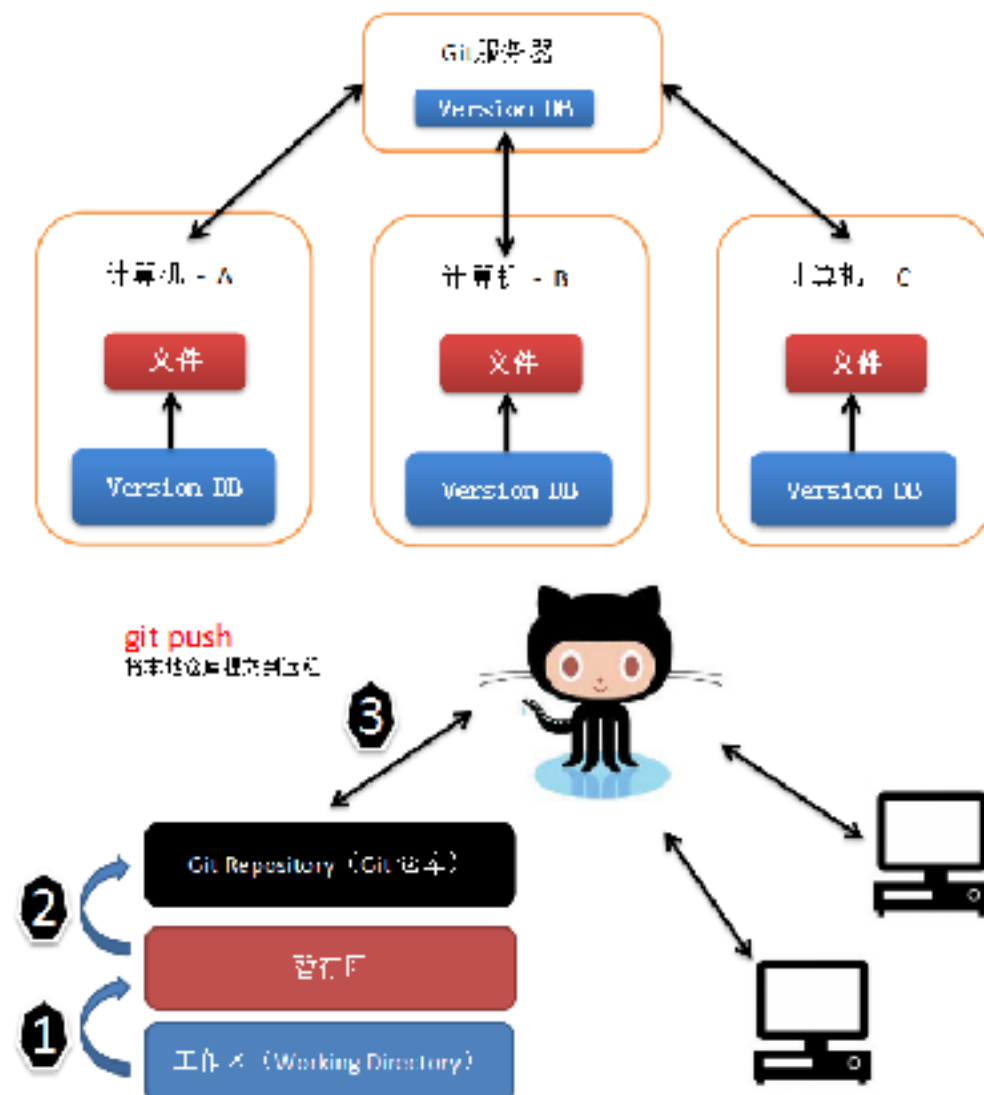
## 删除仓库文件



## 6. Git管理远程仓库

### 使用远程仓库的目的

作用：备份，实现代码共享集中化管理

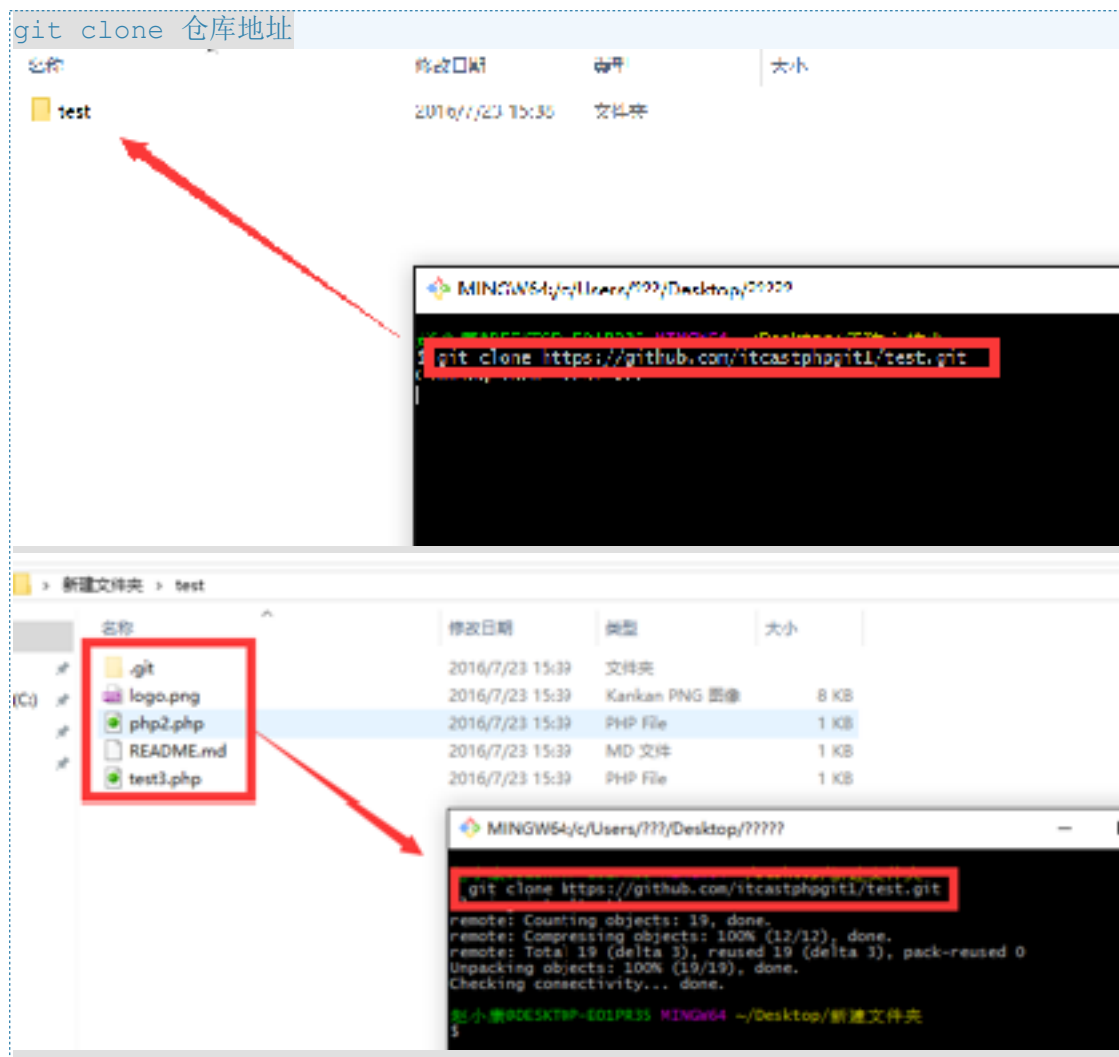


### Git克隆操作

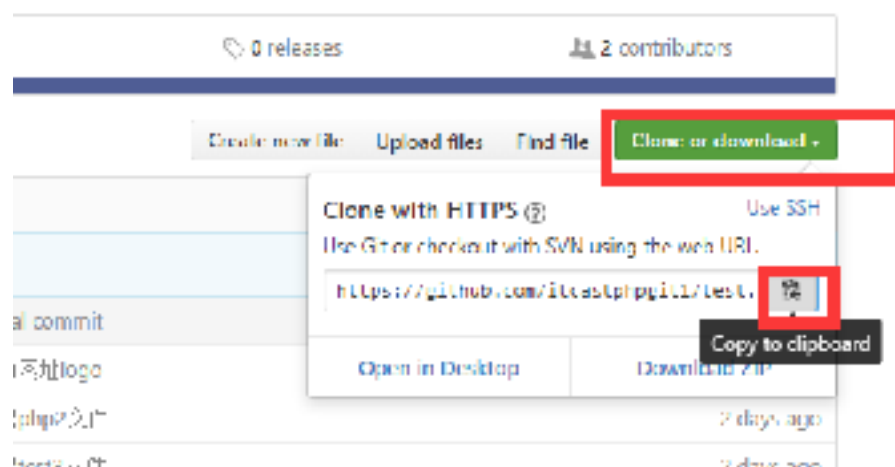
#### 目的

将远程仓库（github对应的项目）复制到本地

## 代码

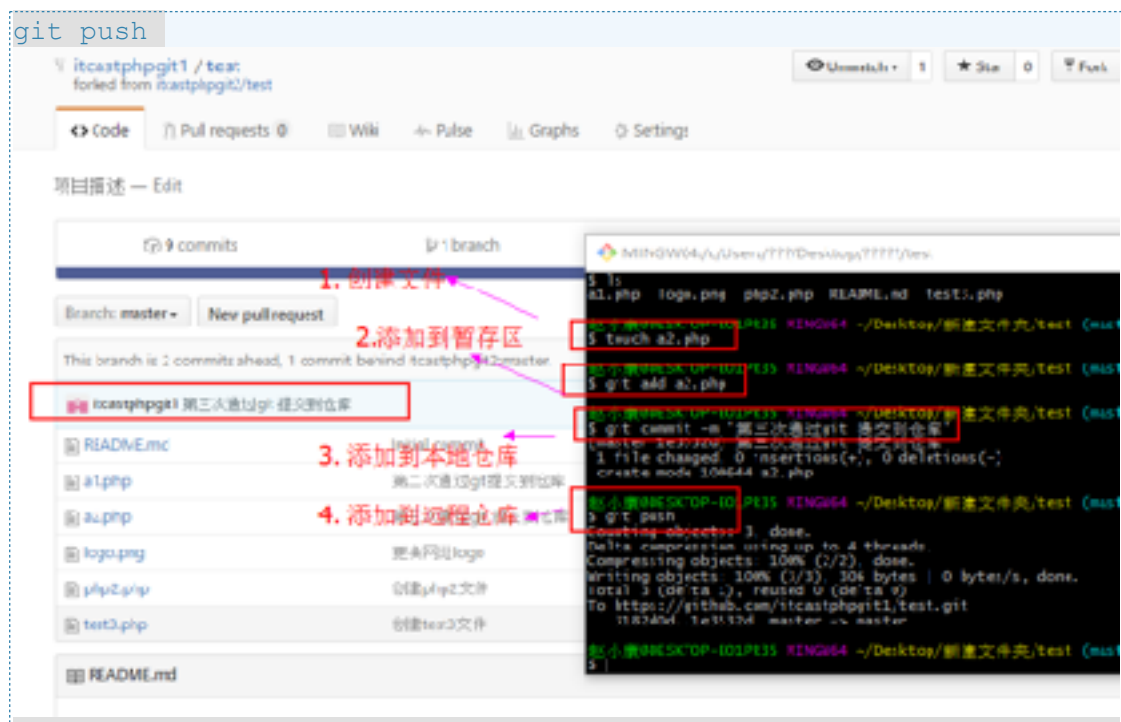


多学一招：仓库地址由来



## 将本地仓库同步到git远程仓库中

git push



1. 创建文件

2. 添加到暂存区

3. 添加到本地仓库

4. 添加到远程仓库

```
$ ls
a1.php  logo.png  php2.php  README.md  test1.php
$ touch a2.php
$ git add a2.php
$ git commit -m "第三次通过git 同步到仓库"
$ git push
```

思考：为什么无法同步

或没有权限

The requested URL returned error: 403 Forbidden while accessing

答案：私有项目，没有权限，输入用户名密码，或者远程地址采用这种类型：

```
vi .git/config
```

# 将

```
[remote "origin"]
  url = https://github.com/用户名/仓库名.git
```

修改为：

```
[remote "origin"]
  url = https://用户名:密码@github.com/用户名/仓库名.git
```

## 6.Github Pages 搭建网站

### 个人站点

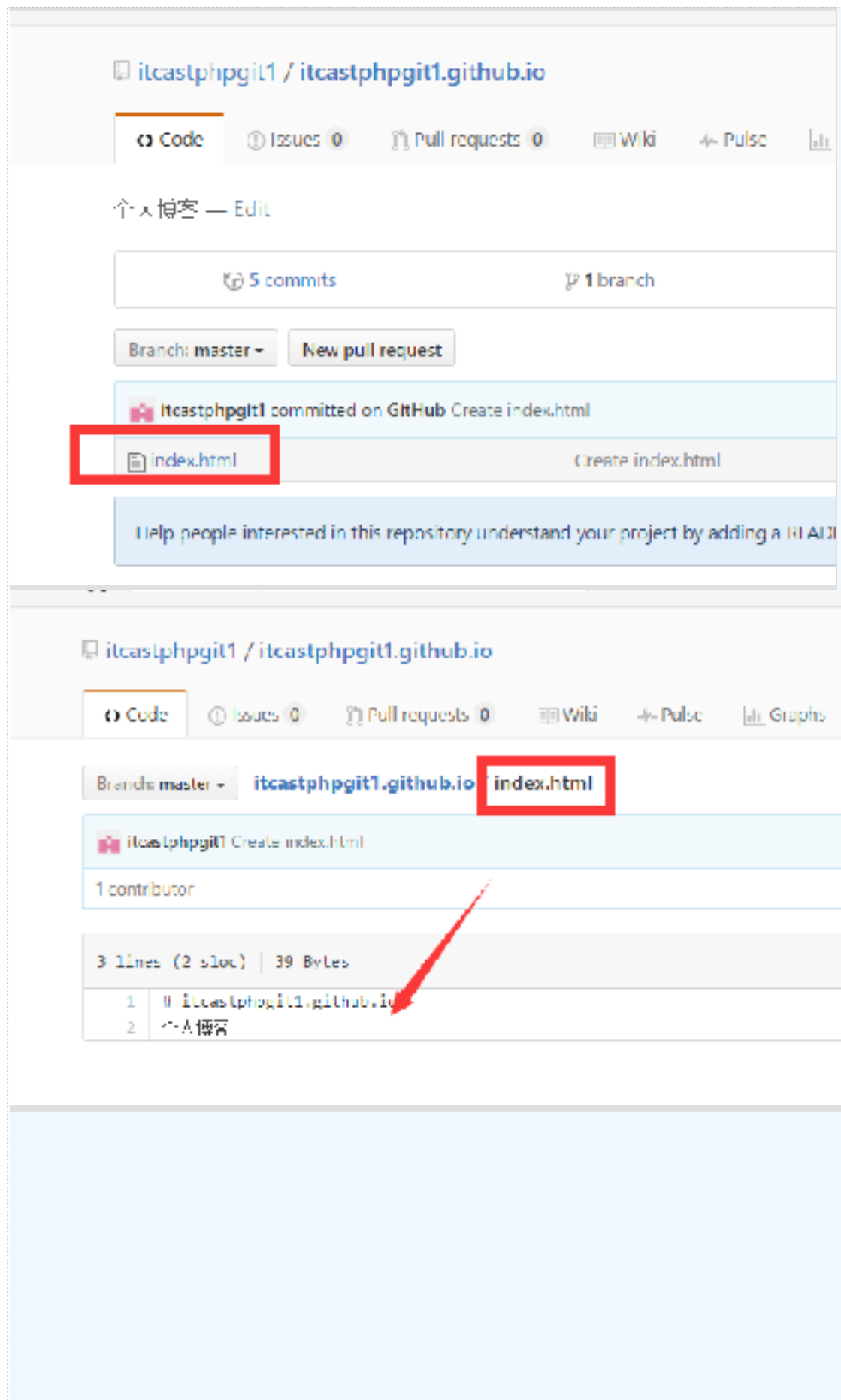
### 访问

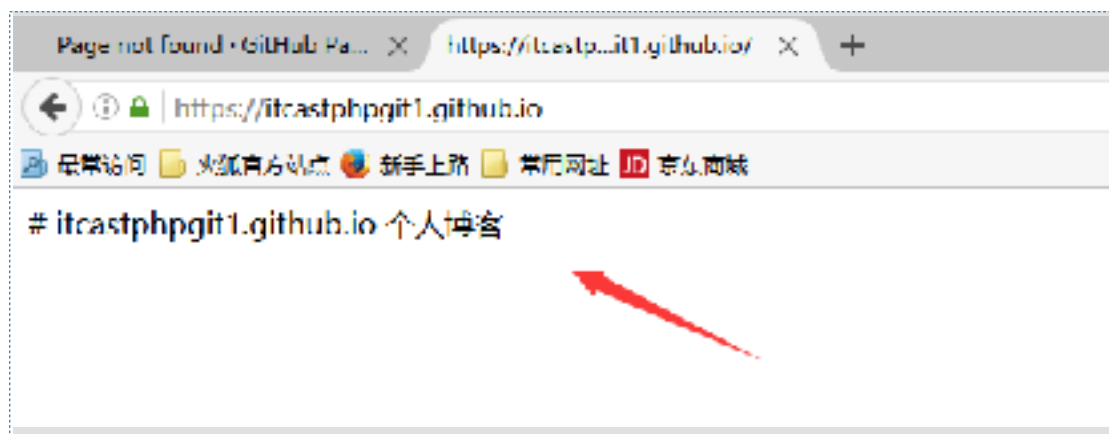
<https://用户名.github.io>

### 搭建步骤

- 1.创建个人站点 -> 新建仓库（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

The screenshot shows the GitHub 'Create a new repository' page. The 'Repository name' field is highlighted with a red box and contains 'itcastphpqiti.github.io' with a green checkmark. The 'Description (optional)' field is also highlighted with a red box and contains '个人网站'. The 'Public' radio button is selected. The 'Initialize this repository with a README' checkbox is checked. The 'Create repository' button is at the bottom.





## 脚下留心

- 1.github pages 仅支持静态网页
- 2.仓库里面是.html文件
- 3、个人主页也可以设置主题

## Project Pages 项目站点

### 访问

<https://用户名.github.io/仓库名>

### 原理

gh-pages 用于构建和发布

### 搭建步骤

- 1.进入项目主页，点击settings
- 2.在settings页面，点击【Launch automatic page generator】来自动生成主题页面
- 3.新建站点基础信息设置
- 4.选择主题
- 5.生成网页