

## 基本概念：

**Repository** 仓库，用于存放项目代码，每个项目对一个仓库

**Fork** 复制克隆项目，该 **fork** 的项目是独立存在的

**Pull request** ：发起合并请求，基于 **fork**

**Watch**：关注项目，可以接收到项目更新提醒

**Issue**：事务卡片，发现代码 **bug**，但目前没有成型代码，需要讨论时可用

注意：私有仓库只能自己或者指定的朋友才有权限操作（收费）

## 对文件的操作：

点击描述可以查看文件提交的详细信息；

增加文件：选择 **create new file** 按钮新建文件；**upload** 按钮上传文件；

编辑文件：在代码仓库中，点击文件名，进入文件详情页，进行编辑；

删除文件：在代码仓库中，点击文件名，进入文件详情页，进行删除；

下载检出文件：在代码仓库中点击 **clone or download** 按钮

注意：删除的文件详细信息可以在 **Commits** 中查看，**Commits** 可以查看每次修改的相关信息；编辑文件也算一次提交

## 对 issue 的操作：

解决 **issue** 后进行关闭

**实战操作**：这里不做解释，实践出现问题建议返回第五集



## 如何为开源项目做出贡献：

### 1. 新建 issue

提交使用问题或者建议或者想法

### 2. Pull request

步骤：

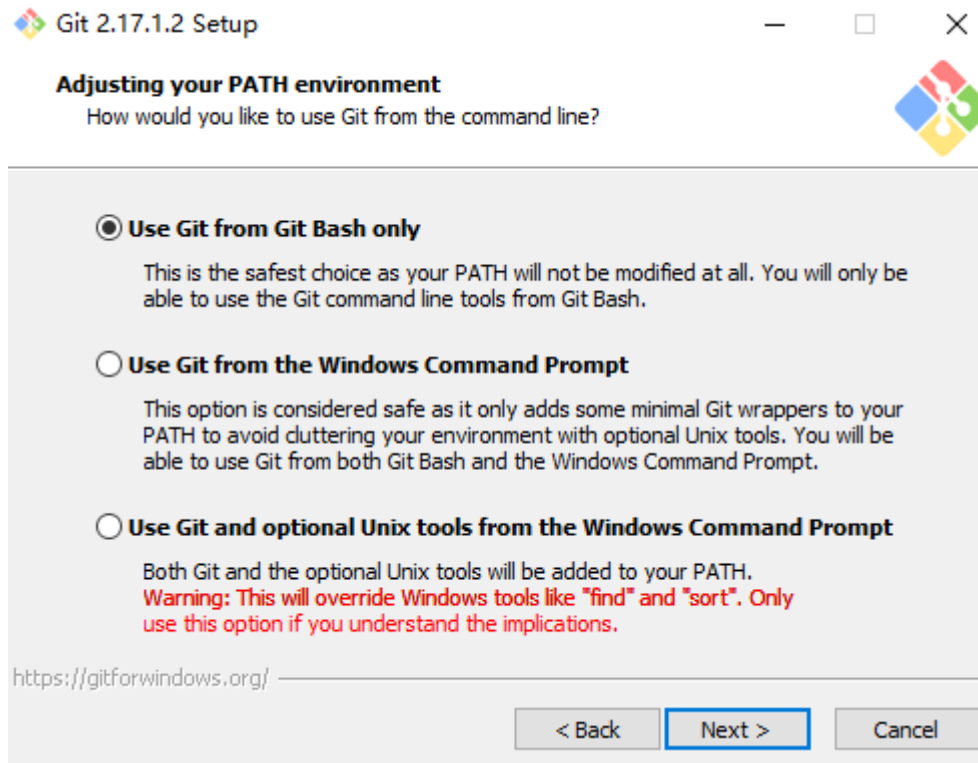
1. **fork** 项目
2. 修改自己仓库的项目代码
3. 新建 **pull request**
4. 等待作者操作（合并）

### Git 的安装和使用:

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

### Git 基本工作流程:

Git 工作区域:

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

一些需要记住的命令:

git status: 确定文件当前所处 Git 工作区域; 这里假设在工作区有文件 HelloWorld.cpp

工作区转入暂存区:

git status

git add HelloWorld.cpp;

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

删除工作区文件:

`git rm -f 文件名`

例如: `git rm -f a.txt`

修改文件:

`vi 文件名`

例如 `vi a.txt`

进入 vim 修改文件 (退出使用: `wq`)

如果对 vim 操作理解有困难的还是建议看一下视频, 视频讲了增删改查的相应操作, 看一下 vim 操作指南, 对 vim 操作有全面的认识

Git 基础设置:

1. 设置用户名

`git config --global user.name '这里填写自己的用户名'`

2. 设置用户名邮箱

`git config --global user.email '这里填写自己的用户名邮箱'`

3. 查看设置

注意: 该设置在 GitHub 仓库主页显示谁提交了该文件, 注意这里的 - 数目为 2!

初始化一个新的 Git 仓库:

1. 创建文件夹

2. 在文件夹内初始化 Git (创建 Git 仓库)

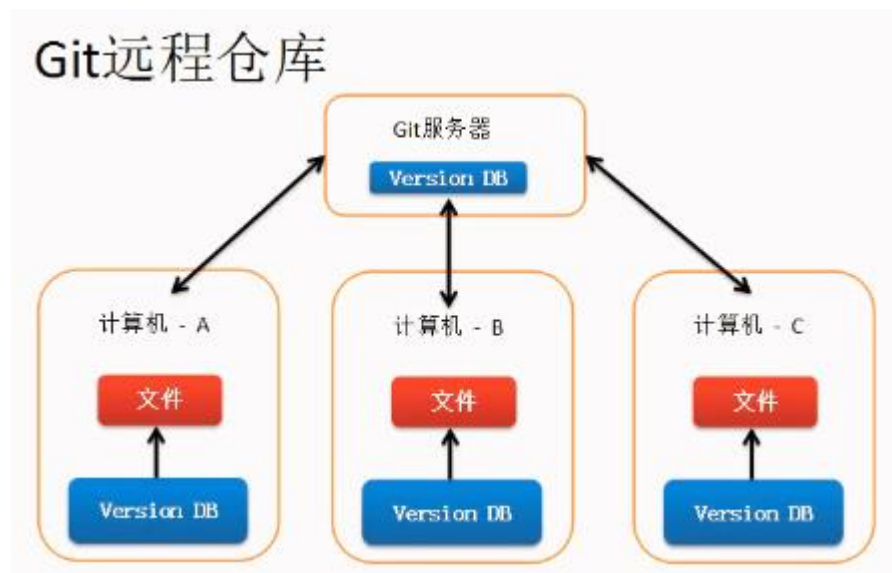
命令行进入当前目录, 使用 `git init` 命令, 成功会显示 .git 文件

3. 向仓库中添加文件

Git 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

以下本人进行本地仓库同步的完整代码（Git 为文件夹名；-Git 为仓库名）

```
123@DESKTOP-MHQCGKI MINGW64 ~/Desktop/Git/-Git (master)
```

```
$ git status
```

```
On branch master
```

```
Your branch is up to date with 'origin/master'.
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
GitLearning.docx
```

```
nothing added to commit but untracked files present (use "git add" to track)
```

```
123@DESKTOP-MHQCGKI MINGW64 ~/Desktop/Git/-Git (master)
```

```
$ git add GitLearning.docx
```

```
123@DESKTOP-MHQCGKI MINGW64 ~/Desktop/Git/-Git (master)
```

```
$ git status
```

```
On branch master
```

```
Your branch is up to date with 'origin/master'.
```

```
Changes to be committed:
```

```
(use "git reset HEAD <file>..." to unstage)
```

```
new file:   GitLearning.docx
```

```
123@DESKTOP-MHQCGKI MINGW64 ~/Desktop/Git/-Git (master)
```

```
$ git commit -m 'vision 1.0'
```

```
[master cf5a4e1] vision 1.0
```

```
1 file changed, 0 insertions(+), 0 deletions(-)
```

```
create mode 100644 GitLearning.docx
```

```
123@DESKTOP-MHQCGKI MINGW64 ~/Desktop/Git/-Git (master)
```

```
$ git status
```

```
On branch master
```

```
Your branch is ahead of 'origin/master' by 1 commit.
```

```
(use "git push" to publish your local commits)
```

nothing to commit, working tree clean

123@DESKTOP-MHQCGKI MINGW64 ~/Desktop/Git/-Git (master)

\$ git push

Logon failed, use ctrl+c to cancel basic credential prompt.

Username for 'https://github.com': yezhaodan

Counting objects: 3, done.

Delta compression using up to 4 threads.

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 133.91 KiB | 11.16 MiB/s, done.

Total 3 (delta 0), reused 0 (delta 0)

To https://github.com/yezhaodan/-Git.git

aa5e68c..cf5a4e1 master -> master

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



## Github Page 搭建网站

个人网站

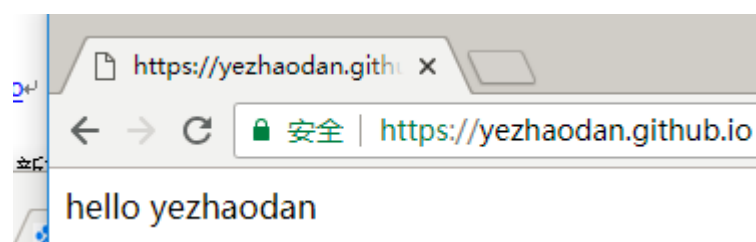
访问:

<https://用户名.github.io>

搭建步骤:

1. 创建个人站点 -> 新建仓库 (注意: 仓库名必须是 用户名.github.io)
2. 在仓库下新建 index.html 的文件

测试:



注意: github pages 只支持静态网页  
仓库里面只能是.html 文件

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 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