如何在Github上协作

安装和配置Git

1. 安装git，从git官网下载后直接点击可以安装：git官网地址
2. 在命令行运行下面的命令配置你的用户名和电子邮箱:

```
git config --global user.name "your-username"
git config --global user.email "youremail@gmail.com"
```

3. 使用下面的命令配置git的安全证书（请在你的git应用的子目录找到安全证书文件的位置，然后填写在命令中）

```
git config --global http.sslCAInfo "C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt"
```

4. 如果不小心安全证书设置错误可以使用下面的命令重新配置：

```
git config --global --unset-all http.sslCAInfo
git config --global http.sslCAInfo "C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt"
```

5. 使用下面的命令可以查看编辑git配置文件，打开编辑器手动输入编辑配置信息，到此git的安装和配置完成。

```
git config --global --edit
```

注册和登陆Github账号

组长和每个小组成员都要在github.com注册自己的账号，然后使用github的账号和密码在Windows上登陆，方法如下：

1. 打开"控制面板"->"凭据管理器"->"Windows凭据"。
2. 选择命令"添加普通凭据"。
3. 按照下面的格式输入,其中用户名和密码请使用你的github用户名和密码：
    1. Internet地址或网络地址: git:https://github.com
    2. 用户名: your-username
    3. 密码： your-password
4. 最后在vscode中选择账号，输入github账号进行登陆。

创建项目仓库

1. 组长负责创建仓库，打开你的Github主页：https://github.com/teamleader,点击右上角的"+"，选择"New Repository".
2. 输入你的仓库名字，例如"learn-git", 点击最下面的按钮"Create repository".
3. 组长打开创建的仓库主页，点击"Settings", 在左侧菜单选择"Collaborators",然后点击按钮"Add people", 输入你的组员的Github账号的电子邮箱，发出邀请.
4. 每个组员打开自己的Github主页，点击右上角的通知图标，然后接受组长的邀请.

开发流程

1.项目组长将项目的Git仓库链接分享给其他组员，组长和组员clone到本地，例如：`git clone https://github.com/<teamleader>/learn-git.git`。

2.每个组员包括组长在进行开发前都需要将本地仓库更新，使用的命令`git pull origin master`.

3.每个组员都需要创建各自的`dev`分支并在这个分支上进行开发，使用的命令例如：`git branch -b dev1`,

4.组员进行开发后，例如添加编写了`ship.py`文件，使用下面的命令保存到本地仓库的dev1分支：

```
git add .
git commit -m "add a ship class"
```

5. 使用下面的命令将本地仓库的更新推送到github上的远程仓库的dev1：

```
git push origin dev1
```

6. 组员打开github上仓库的pull request页面，提出`pull request`。
7. 项目组长或者项目小组成员对提交的代码要进行`code review`，通过了`code review`以后可以选择将这次pull request提交的代码合并到master分支，具体细节请查看下面视频和文档:
    - Github Pull Request视频教程：B站链接
    - Github的文档：如何创建pull request

8.将`dev`分支合并到`master`分支, 在合并分支经常会遇到冲突（merge conflicts，冲突在当两个分支对同一个文件的同一个地方的修改不同时发生），处理冲突的工作流程是:

```
- 项目团队成员需要进行沟通，了解冲突的来源
- 讨论可行的解决冲突的方案，例如：
  - 保留当前分支的修改
  - 保留要合并的分支的修改
  - 最理想的方案是重写两个分支的代码并提交，在没有冲突的情况下再合并
  - 合并后再推送（push）到远程仓库
```

处理冲突的视频教程：B站链接