

本次作业的要求

1. 完成git的下载安装配置，并在本地创建一个名字为 helloworld.txt 的文件（大家需要有一个认识：git本身其实只能管理文本文件，word之类的二进制文件并不适用，所以比较适合管理代码项目）

（补充）由于上述原因，我们在github里用于说明的文件通常使用markdown格式而不是纯文本的txt格式，因为它可以渲染成各种样式，也可以方便的导出为pdf，我这篇说明就是markdown所写，Typora是一个win、mac平台通用的轻量级编辑器。大家有兴趣可以尝试这种文本的编写，非常的方便哦！

1. 文件里面写一些自己的感想（关于程序设计基础这门课程的，关于老师的，关于我的，什么都行），有话则长无话则短，能写一些自己学习git的心得更好，吐槽它哪里不好用，或者你为什么讨厌它。我不需要你给我写什么命令是什么意思。有兴趣可以去这里学习 <https://www.runoob.com/git/git-tutorial.html>，或者搜索廖雪峰的git教程。我希望大家不要去网上找了贴过来。我花时间给大家用心写东西，希望大家也能用心回馈我）
2. 在10月27号之前完成提交。

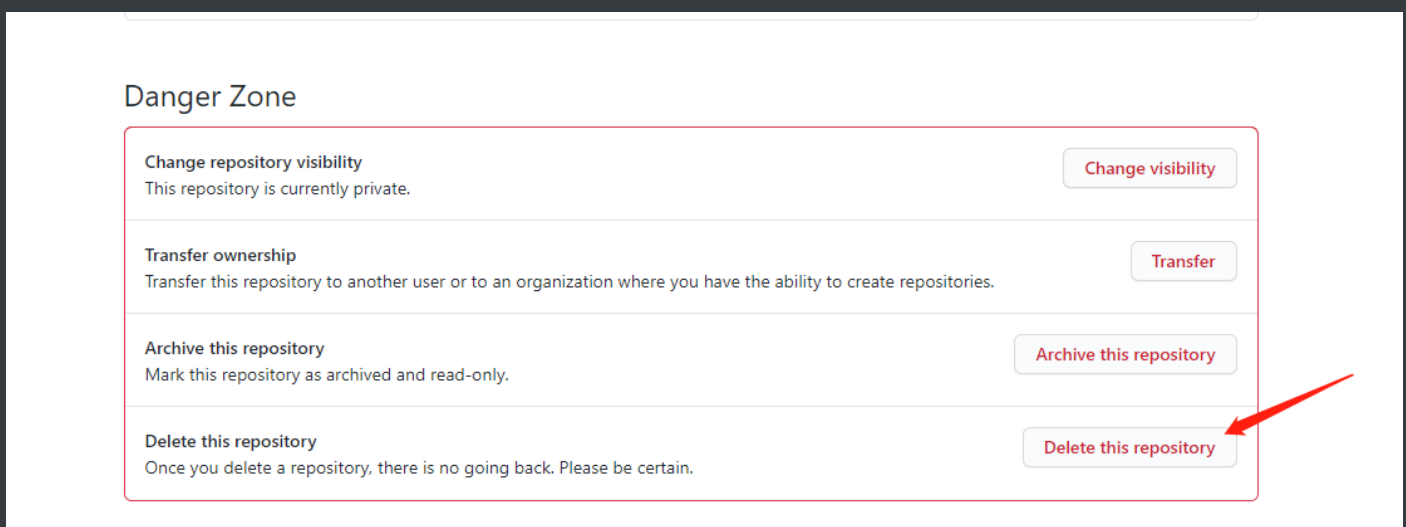
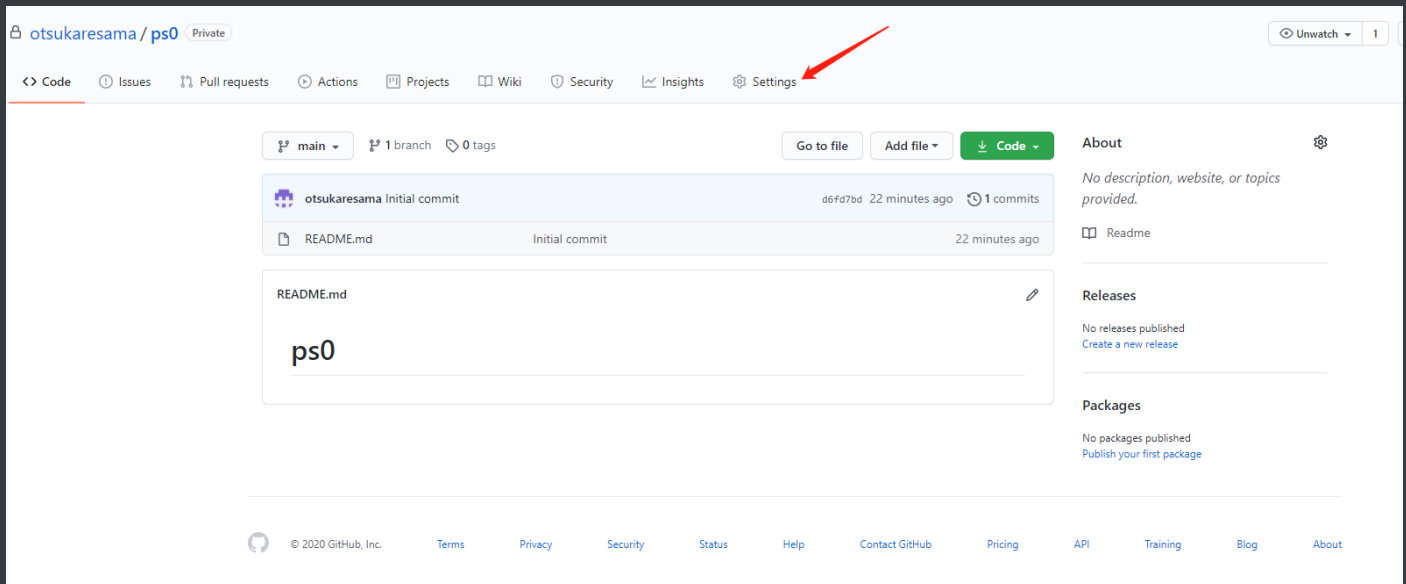
通过github进行作业提交

一、说明


本次课程**最终只会有两个仓库**（repository），一个就是ps0，用来存放大家平时的10次作业。另外是一个名字叫project，用来存放期末需要提交的小项目。

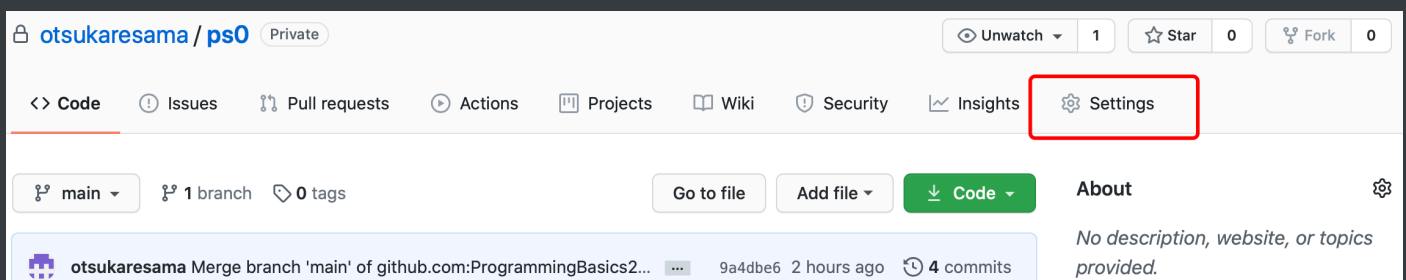
也就是说大家这学期一共只需要创建两个repositorys，不麻烦吧！

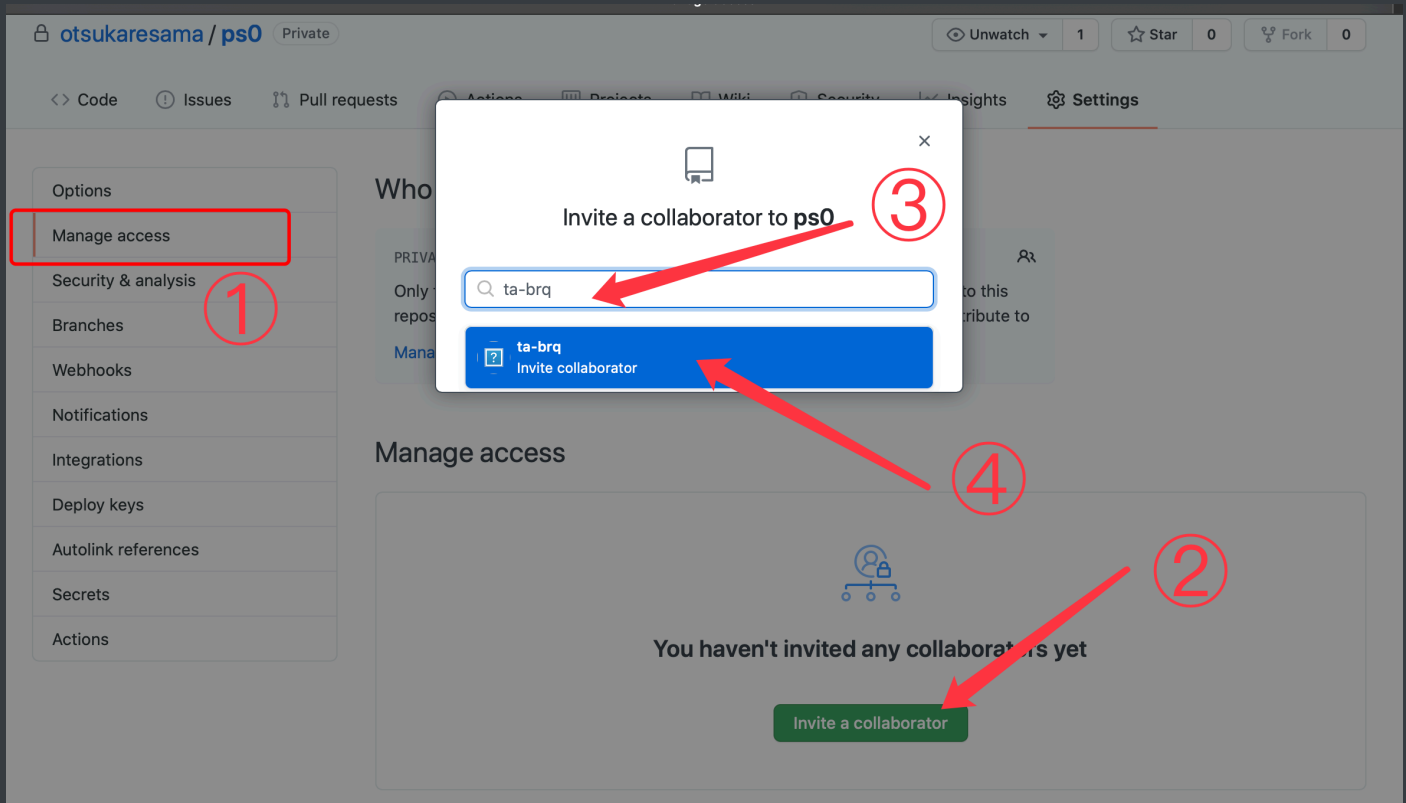
为了避免大家前期因为不规范操作，和我的一些错误指导造成的各种各样的问题。我希望大家在**github网站上，把自己的创建的ps0这个项目删掉**。操作步骤如下：



二、如何创建一个仓库（以ps0为例）

1. 在<https://github.com/repositories/new>页面，或者是点击右上角的  创建一个名为ps0的仓库，选 **private**，**下面的复选框都不要勾选**。每次新建完，一定要保证自己这个repo里面是空的！
2. 将这个项目分享给我，操作步骤如下的截图：





3. 将这个自己新建的repository下载到自己的电脑上，在自己的gitbash或者是mac的终端里，运行如下的命令：**注意不要直接复制下面的链接。把 mygithubname改成自己的用户名！**

```
git clone https://github.com/mygithubname/ps0.git
```

4. 切换目录到ps0文件夹内，运行如下的命令：

```
cd Desktop\ps0
```

5. 将自己本地的repository和我们课程主页的repository关联起来。其中course是我们给课程的repository取的一个别名。运行如下命令：

```
git remote add course git@github.com:ProgrammingBasics2020/ps0.git
```

6. 从课程repository上把作业要求拉取到本地自己的repository中：（course表示我要拉取的是我上面取别名的课程repository对应的地址，main表示的是分支名）

```
git pull course main
```

7.至此我们就完成了前期的准备工作，作业的要求就已经同步到我们自己的仓库里了。接下来我们就可以在自己的机器上，新建作业文件啦！

三、提交自己的作业

1. 在ps0文件夹内新建一个文件夹，名字叫 hw1 ，代表我们的第一次作业存放的目录。在hw1文件夹内，新建一个名为 helloworld 的文本文件。

2.将这次创建的文件添加到git系统管理的范围里。 运行如下命令：

```
git add helloworld.txt
```

3. 将文件提交。意思就是让git系统形成一个还原点，方便以后进行恢复。 运行如下命令：

```
git commit -m "双引号里填写自己的说明，相当于是备注"
```

4. 最后我们就可以把我们所做的更改提交到github网站上我们自己的私人仓库里了。

```
git push --tags origin main
```

这条命令里面的origin和之前提到的course是一个意思，都是一个别名。只不过这个别名代表的是github网站上托管的我们自己的repository的地址，而不是我们课程公共的repository的地址。

5. 这些步骤完成后，所有的本地变化我们就能够在github自己repository页面看到了，同时我也能看到。

6. 最后补充一个命令，当你不知道当前该运行什么git命令时，你可以在黑框框中键入：

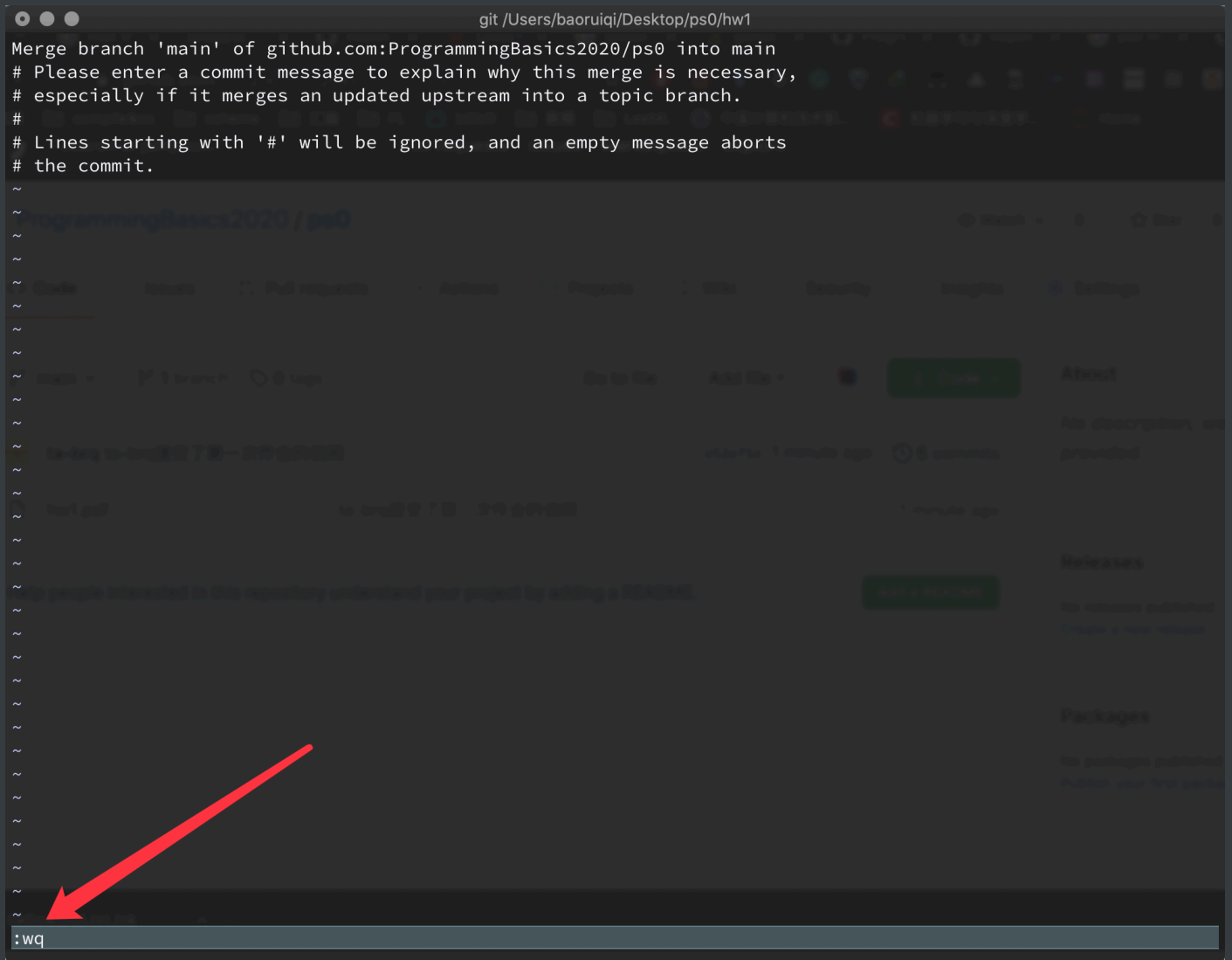
```
git status 输出的信息会给你一定的提示
```

四、后续的作业提交

1. 每次我发布作业后会通知大家，运行这个命令将作业要求拉取到自己本机的仓库中：

```
git pull course main
```

2. 运行上面的命令，如果成功后，gitbash会进入VIM编辑器的窗口，提示我们git为我们合并了一个分支。（大家暂时不需要理解是什么意思）按照下图所示，在英文输入法的状态下，输入 :wq （即冒号加英文的wq两个字符），重新回到gitbash界面说明成功完成了分支的合并。这里不需要鼠标操作，直接键盘输入冒号和wq即可。



2. 根据每一次作业要求的文件名如第一次叫hw1.pdf，在自己电脑上的ps0这个文件夹里新建一个文件夹名字叫做hw1。第二次作业名叫hw2.pdf，那么新建的文件夹对应为hw2，以此类推。

3. 进入hw1。在里面按照作业的要求，新建对应的文件并完成作业。

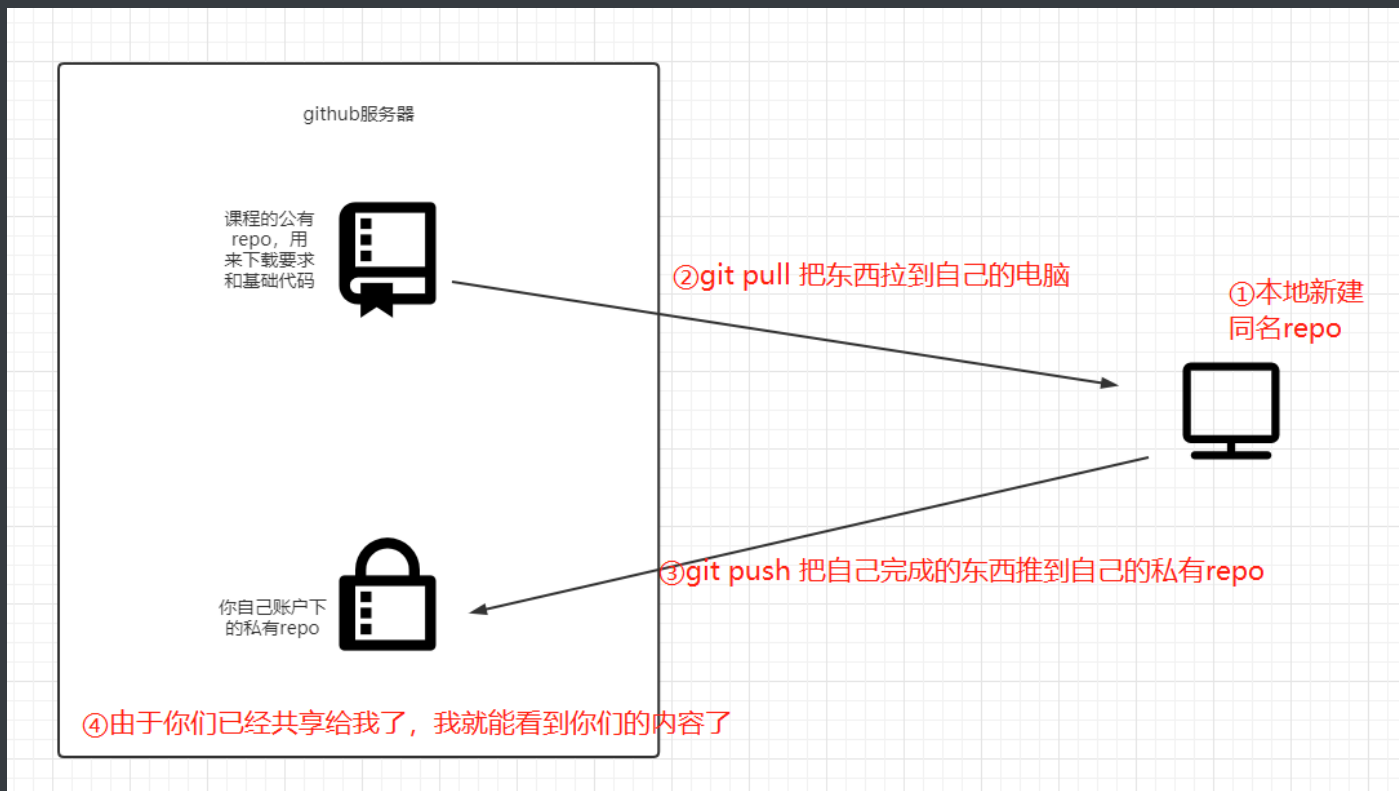
4. 将你所做的更改提交到github网站上自己的私人仓库里。

```
git push --tags origin main
```

5. 这些步骤完成后，所有的本地变化我们就能够在github自己repository页面看到了，同时我也能看到。

附

下图是我画的整个过程的示意图。



我和顾老师其实都忽略了很重要的东西，教育和教学需要因材施教、循序渐进。一下子给大家灌输太多的东西，会让人觉得无所适从。所以觉得不想学这一套工具的可以以自己的方式把每次作业交给我。我尽量保证大家最后的成绩，不会因为使用工具的不同，而有太大差异。

但是另一个角度我想说明的是，顾老师的这种教学理念是完全正确的。比如你们看我上面这张图。通过一段时间的反复操作之后，你们会自然而然的对什么是client，什么是server有比较深刻的认识，这对你们后面学习网络或者是应用开发等，都有很大的帮助。世界上不存在绝对的没有用的事情，你下的功夫，花的时间总会在以后的某个时刻让你感觉到它的价值！