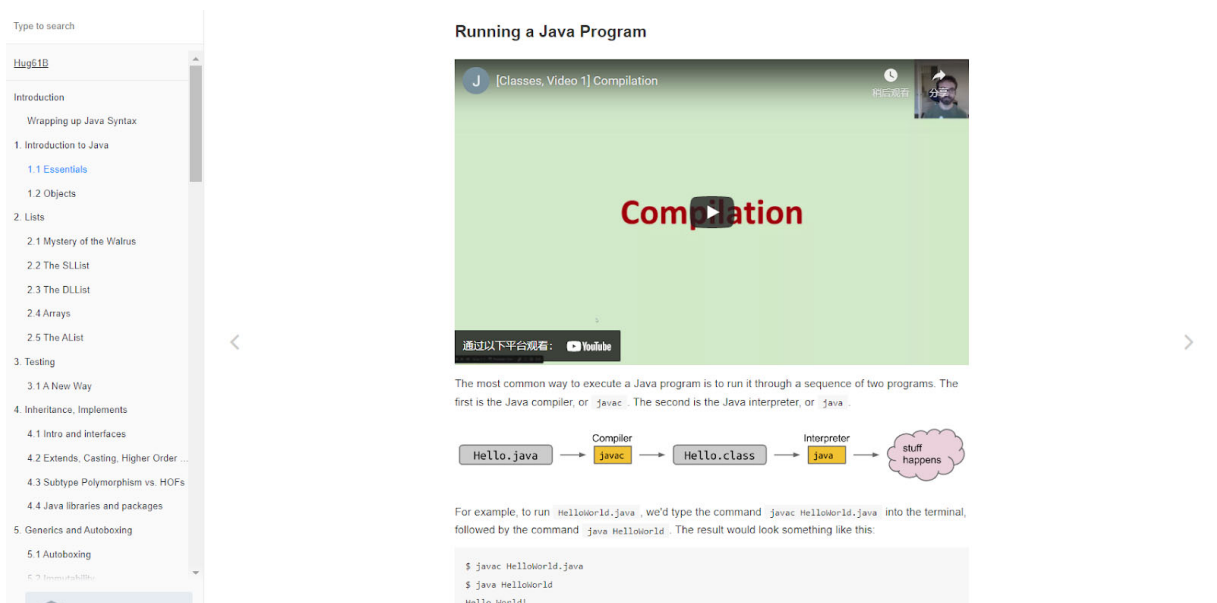


# Fx 从0开始的 CS61b生活 V1.0

## 1.大体介绍

课程主页:[CS 61B Spring 2018](#)

一章里面嵌套一些Lecture视[CS 61B Spring 2018](#)频，相当于一段知识一节视频，对学习节奏很有帮助gradescope.com



The screenshot shows the CS61B website interface. On the left is a sidebar with a search bar and a table of contents. The main content area is titled 'Running a Java Program' and features a video player with a 'Compilation' title card. Below the video player is a diagram illustrating the Java compilation process: 'Hello.java' is compiled by 'javac' into 'Hello.class', which is then interpreted by 'java' to produce 'stuff happens'. The text explains that the most common way to execute a Java program is through these two steps: compilation and interpretation. It also provides an example of running 'HelloWorld.java' in a terminal.

```
$ javac HelloWorld.java
$ java HelloWorld
Hello World!
```

## Lecture

包括Video,Slide和Guide，Video可以跟着上面的Reading里面看，这里的Video实际上是一章的全部Video，没有像Reading一样分节，每节课后可以查看Guide巩固理解

## Discussion

discussion是一些课后题目，对知识点的加深练习，并附有Solution

## Assignments

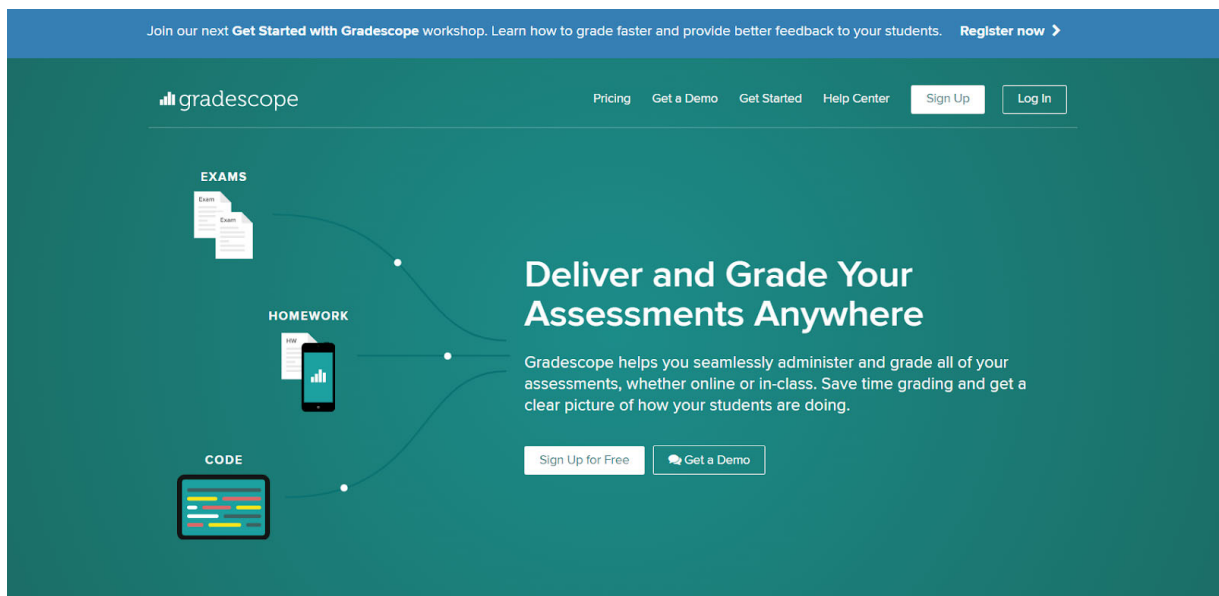
这部分主要是Lab与Project, Homework组成，Spring 2018有14个lab，10个homework以及3个project，每个作业均有说明文档，来指导你完成

# 2.搭建环境

cs61b sp18是采用线上评分的方式，也就是autograder，具体是在gradescope该网站上评测

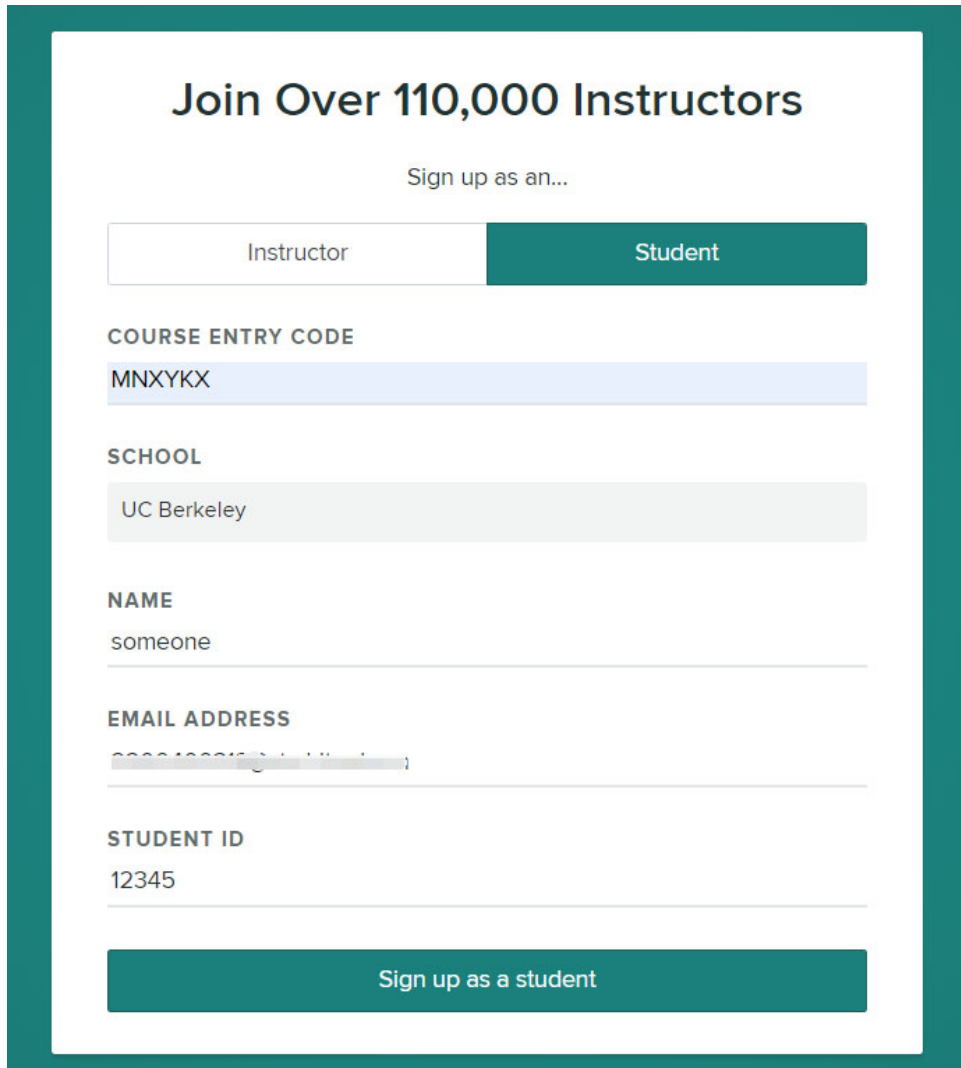
## 1.gradescope

网站地址:



注册步骤:

选择Sign Up, Sign up as an **Student**



The screenshot shows a sign-up form titled "Join Over 110,000 Instructors". Below the title is a "Sign up as an..." dropdown menu with two options: "Instructor" and "Student". The "Student" option is selected and highlighted in teal. Below this are several input fields: "COURSE ENTRY CODE" with the value "MNXYKX", "SCHOOL" with the value "UC Berkeley", "NAME" with the value "someone", "EMAIL ADDRESS" with a masked email address, and "STUDENT ID" with the value "12345". At the bottom is a teal button labeled "Sign up as a student".

Join Over 110,000 Instructors

Sign up as an...

Instructor Student

COURSE ENTRY CODE

MNXYKX

SCHOOL

UC Berkeley

NAME

someone

EMAIL ADDRESS

someone@example.com

STUDENT ID

12345

Sign up as a student

## 课程代码

Spring 2018 : MNXYKX

Spring 2021 : MB7ZPY

区别说明:2018的评测最全, 2021主要是为了写著名项目Gitlet

## SCHOOL

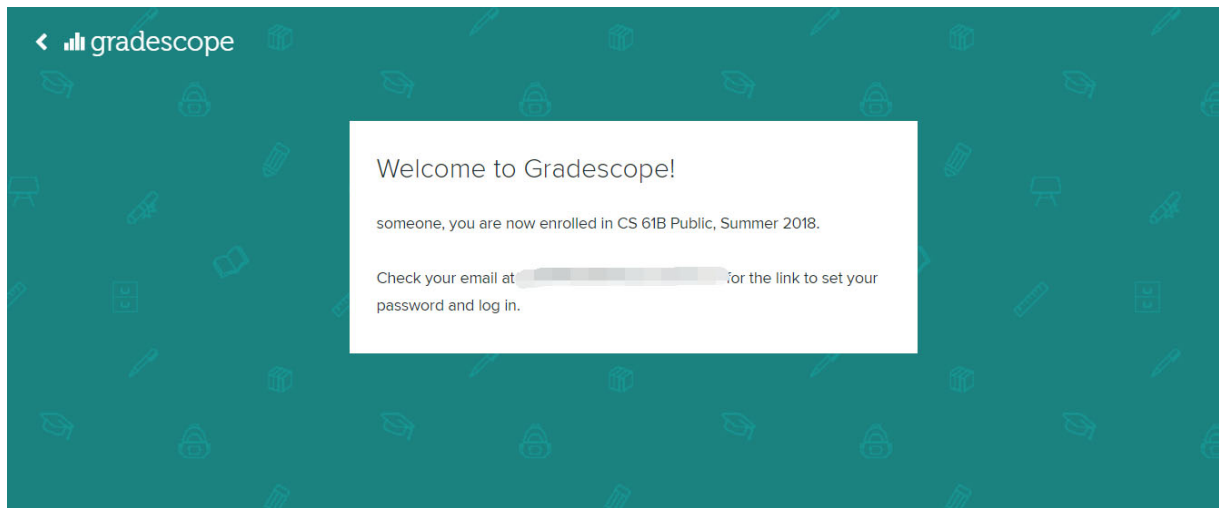
学校请选择**UC Berkeley**, 一定要**拼全**, 否则无法进入

## EMAIL ADDRESS

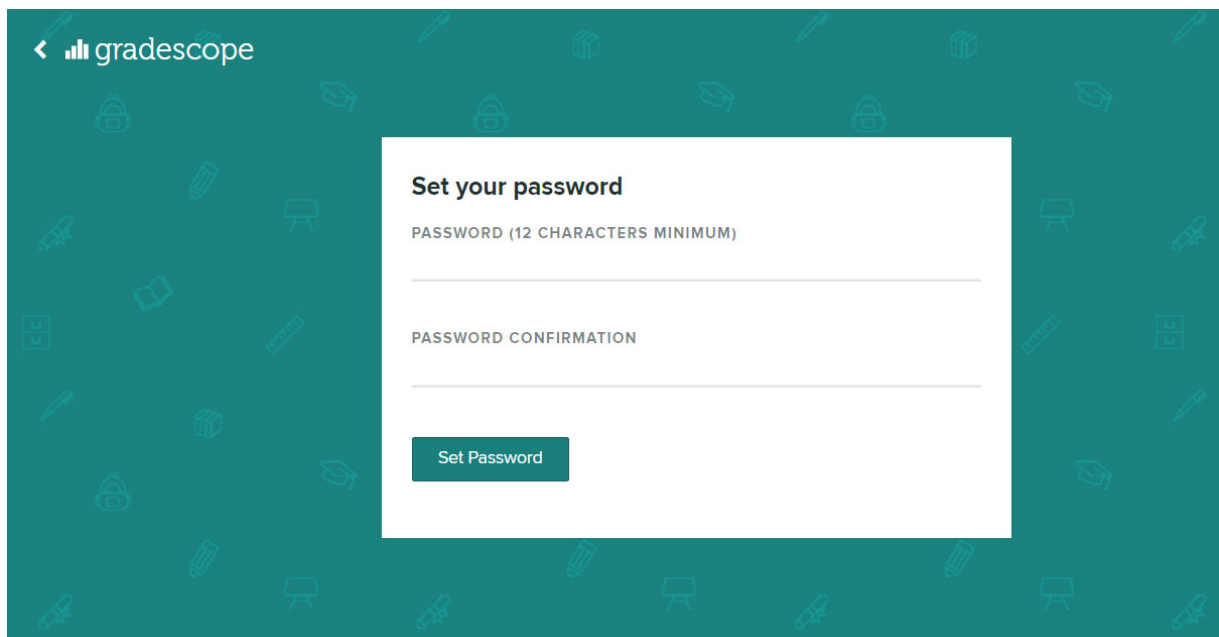
填入一个你自己的邮箱，不一定是Berkeley的邮箱，任何邮箱包括QQ邮箱也可

关于NAME和STUDENT ID随便填即可

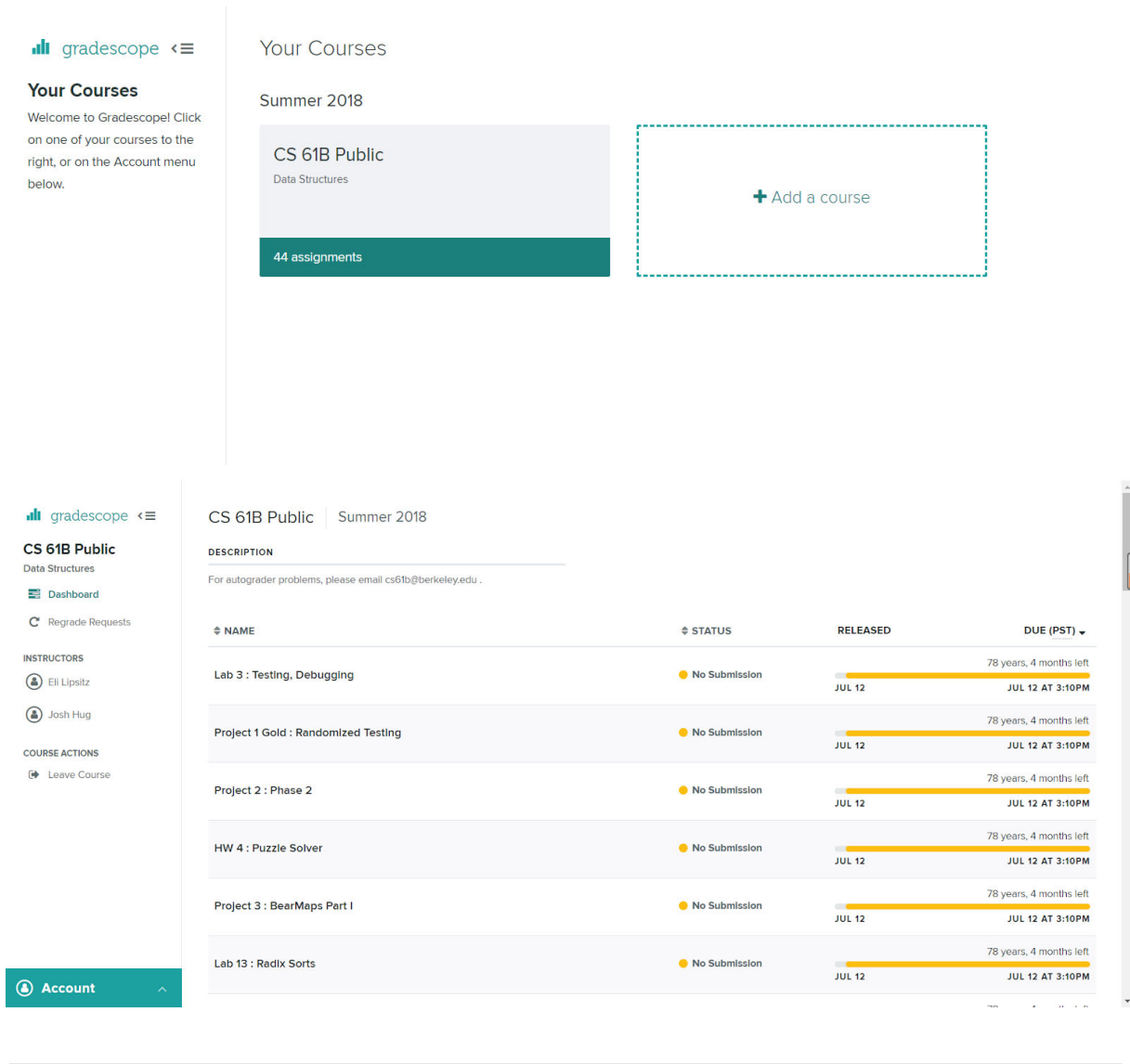
出现以下界面说明注册成功



根据提示进入自己的邮箱，设置初始密码



成功之后进入Gradescope



2.Github

网址:<https://github.com/>

由于在gradescope上进行评测是借助于你的远程仓库上的代码库，因此实际操作过程中需要将你的本地代码上传到Github并提交评测首先去注册一个Github的账号，然后新建一

## 个远程仓库，该步骤比较简单

### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \*

Fallenpetal ▾

Repository name \*

CS61B Tutorial ✓

Great repository names are Your new repository will be created as CS61B-Tutorial. out congenial-giggle?

Description (optional)

☒ Public

Anyone on the internet can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

☐ Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

☐ Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

## 新建成功后出现

Quick setup — if you've done this kind of thing before

Set up in Desktop

 or 

HTTPS

SSH

https://github.com/Fallenpetal/CS61B-Tutorial.git

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# CS61B-Tutorial" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M master
git remote add origin https://github.com/Fallenpetal/CS61B-Tutorial.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/Fallenpetal/CS61B-Tutorial.git
git branch -M master
git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

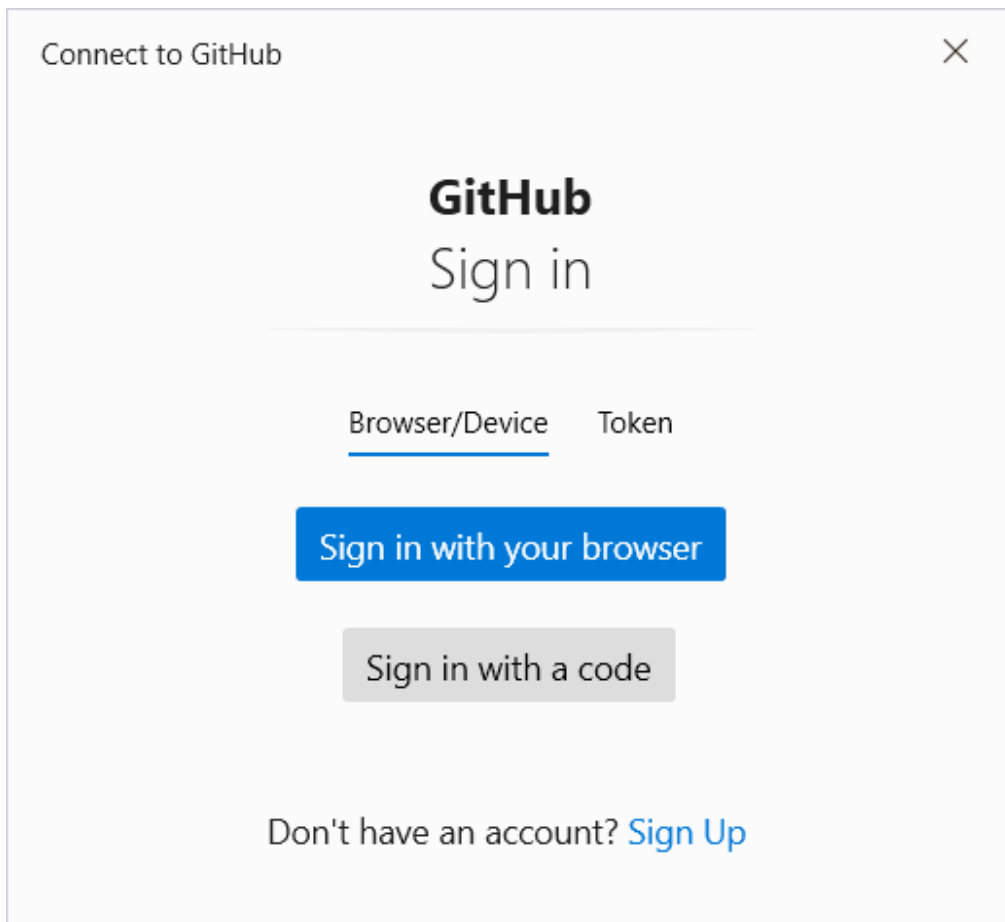
Import code

记住自己的链接，上方的SSH

例如 `git@github.com:Fallenpetal/CS61B-Tutorial.git`

## 配置SSH Key

原本使用https链接进行上传是对新手很友好的，在上传时只需要验证一下账户密码即可，类似这样



但是github在2021年8月13日取消了账户密码的验证，详见

<https://github.blog/2020-12-15-token-authentication-requirements-for-git-operations/>

配置token更加麻烦，因此推荐配置SSH密钥较为简单，一个关于配置SSH密钥的教程

[Github 生成SSH秘钥（详细教程） - yucreator - 博客园](#)

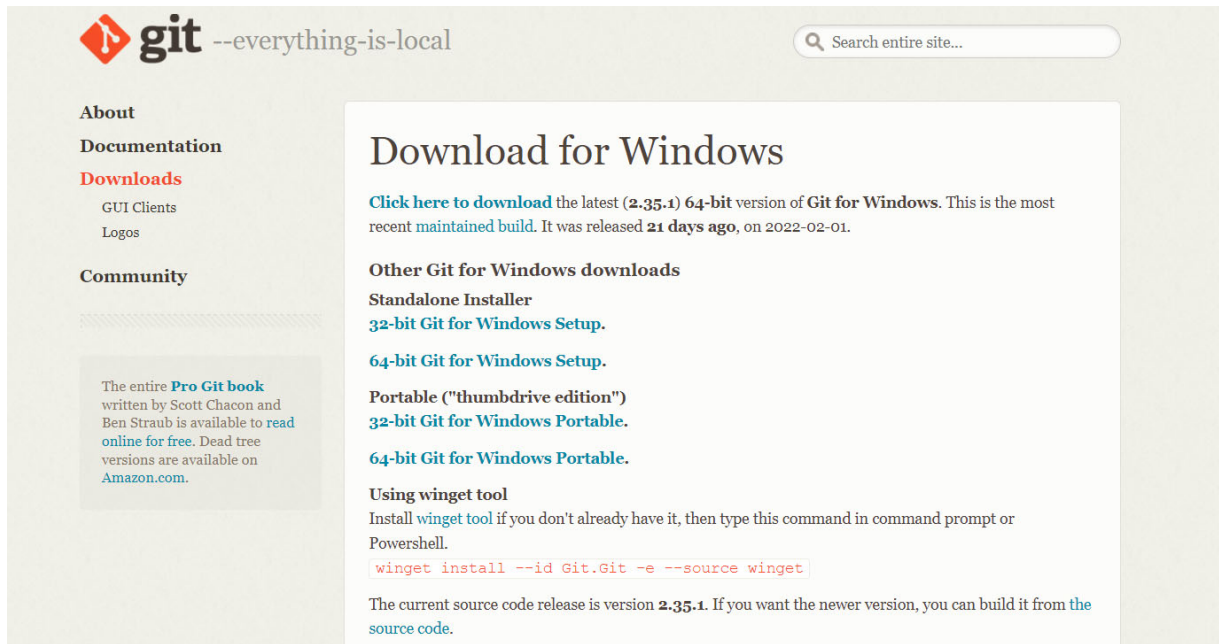
由于该教程写的非常全面，我就不需要再写一遍了

## 3. 安装Git

Github只是你的远程托管，相当于网盘，而将本地代码上传到网盘则使用Git

## 在Windows上安装Git

官网:<https://git-scm.com/download/win>



选择合适的版本安装，各种选项目前均可默认

安装完成后，在开始菜单里找到“Git”->“Git Bash”，蹦出一个类似命令行窗口的东西，就说明Git安装成功！

安装完成后，需要设置使用者的邮箱和用户名，在Git Bash中输入：

```
git config --global user.name "Your Name"
git config --global user.email "email@example.com"
```

MacOS与Linux的安装请自行google

## 4.建立本地仓库

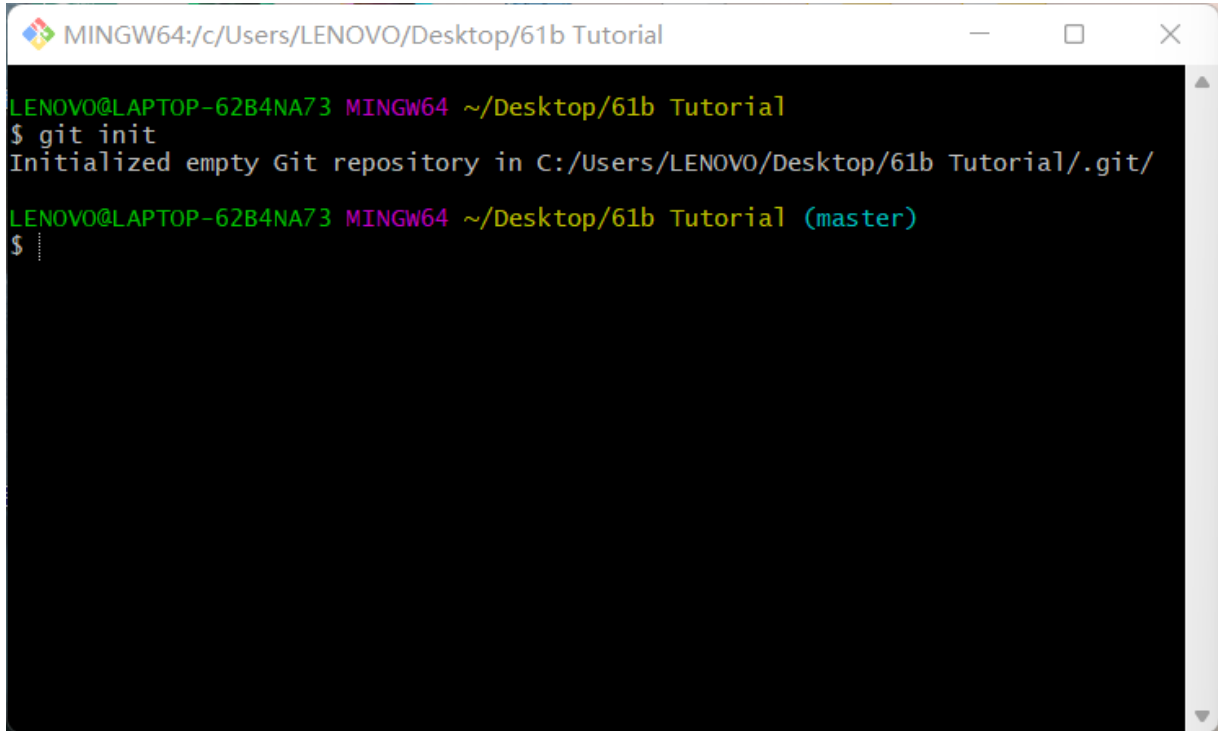
在电脑上新建一个文件夹，这个文件夹相当于你以后写代码的地方，也就是以后的61b代码都存储在这里，之后打开文件夹，右键选择**Git Bash Here**，弹出以下窗口



执行命令

```
git init
```

进行本地仓库初始化d



```
MINGW64:/c/Users/LENOVO/Desktop/61b Tutorial
LENOVO@LAPTOP-62B4NA73 MINGW64 ~/Desktop/61b Tutorial
$ git init
Initialized empty Git repository in C:/Users/LENOVO/Desktop/61b Tutorial/.git/
LENOVO@LAPTOP-62B4NA73 MINGW64 ~/Desktop/61b Tutorial (master)
$
```

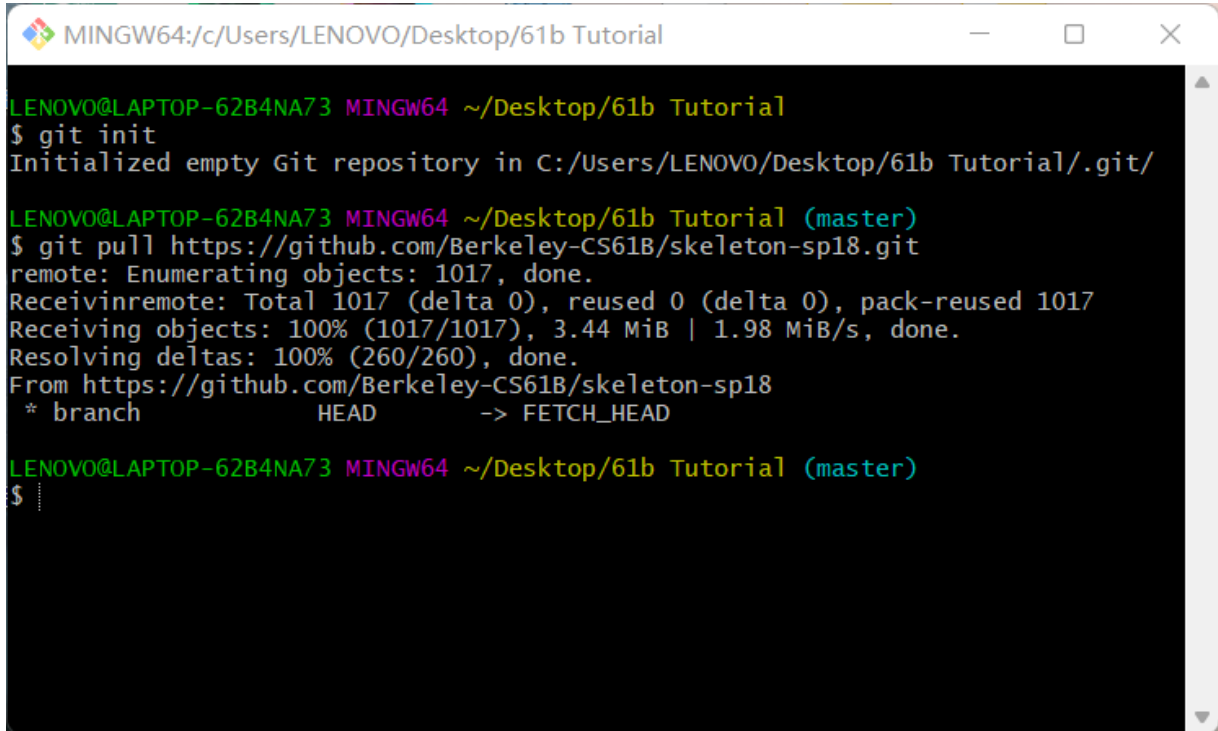
## 5.获取Skeleton代码

由于61b每个项目都有一定的框架代码，我们需要这些代码的辅佐。

在刚才的Git Bash里面，执行命令：

```
-t git
```

等待片刻后，出现



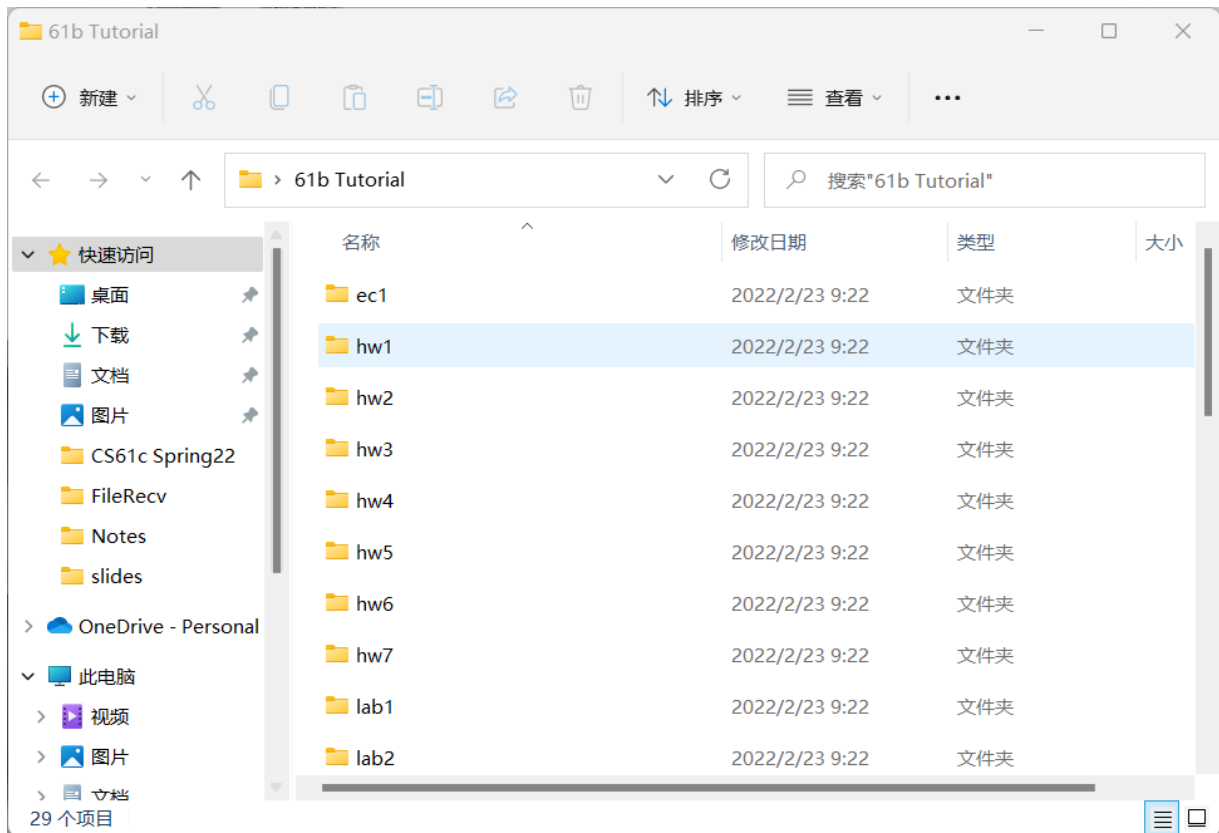
```
MINGW64:/c/Users/LENOVO/Desktop/61b Tutorial

LENOVO@LAPTOP-62B4NA73 MINGW64 ~/Desktop/61b Tutorial
$ git init
Initialized empty Git repository in C:/Users/LENOVO/Desktop/61b Tutorial/.git/

LENOVO@LAPTOP-62B4NA73 MINGW64 ~/Desktop/61b Tutorial (master)
$ git pull https://github.com/Berkeley-CS61B/skeleton-sp18.git
remote: Enumerating objects: 1017, done.
Receivinremote: Total 1017 (delta 0), reused 0 (delta 0), pack-reused 1017
Receiving objects: 100% (1017/1017), 3.44 MiB | 1.98 MiB/s, done.
Resolving deltas: 100% (260/260), done.
From https://github.com/Berkeley-CS61B/skeleton-sp18
 * branch                HEAD      -> FETCH_HEAD

LENOVO@LAPTOP-62B4NA73 MINGW64 ~/Desktop/61b Tutorial (master)
$
```

则成功，此时点开刚才我们新建的本地文件夹，发现里面多了很多文件，这些就是 skeleton 代码



## 3. 进行代码作业

之后的课程lab,project,homework均在框架代码的基础上进行编写，现在模拟一下第一次写lab并提交到gradescope上评测的过程

### Spring 2018 lab1 闰年判断

地址:<https://sp18.datastructur.es/materials/lab/lab1/lab1#f-leap-year>

lab内容描述:

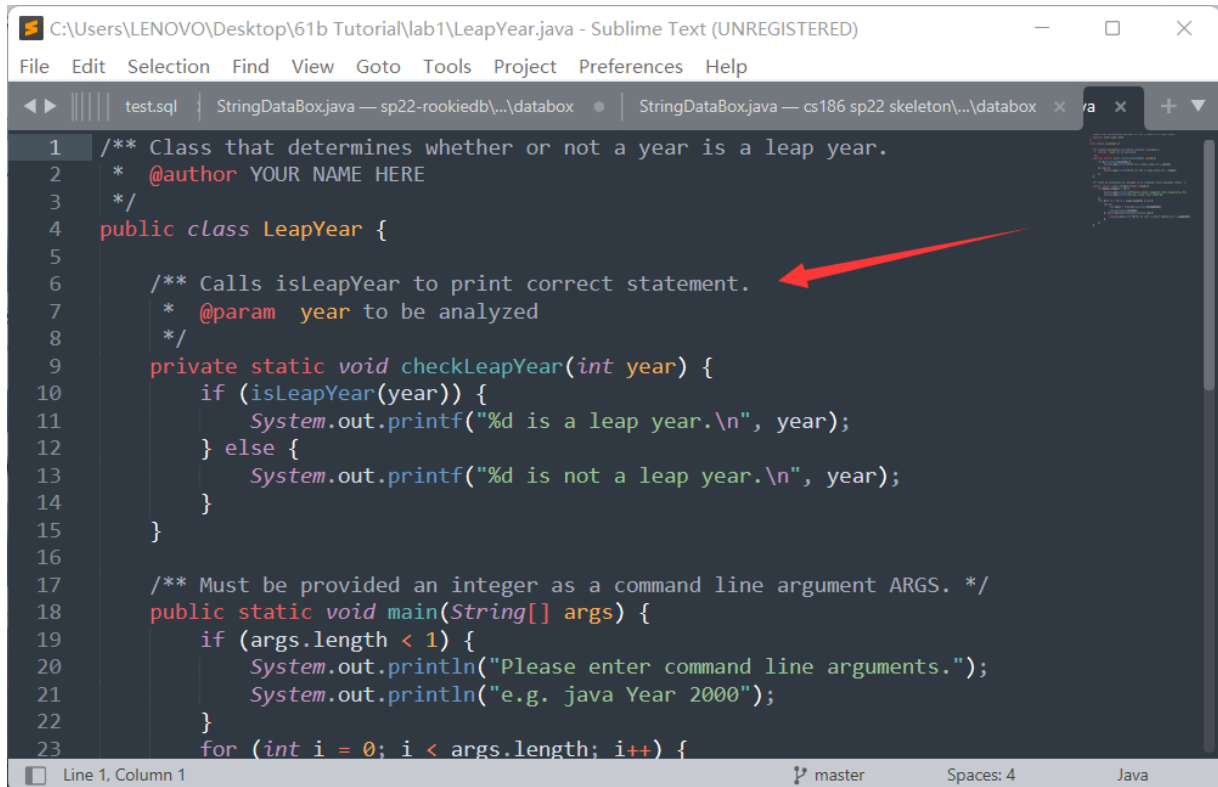
闰年是:

- 可被 400 整除或
- 能被 4 整除，不能被 100 整除。

例如, 2000 年和 2004 年是闰年。1900、2003 和 2100 不是闰年。

你的目标:编写一个函数方法判断一个年份是否是闰年

进入本地的61b文件夹, 可以看到有一个lab1文件夹, 点击进入, 打开里面的 LeapYear.java文件



```
1  /** Class that determines whether or not a year is a leap year.
2  *  @author YOUR NAME HERE
3  */
4  public class LeapYear {
5
6      /** Calls isLeapYear to print correct statement.
7       *  @param year to be analyzed
8       */
9      private static void checkLeapYear(int year) {
10         if (isLeapYear(year)) {
11             System.out.printf("%d is a leap year.\n", year);
12         } else {
13             System.out.printf("%d is not a leap year.\n", year);
14         }
15     }
16
17     /** Must be provided an integer as a command line argument ARGS. */
18     public static void main(String[] args) {
19         if (args.length < 1) {
20             System.out.println("Please enter command line arguments.");
21             System.out.println("e.g. java Year 2000");
22         }
23         for (int i = 0; i < args.length; i++) {
```

在箭头处使用Java语法编写一个函数判断是否是闰年,答案仅供参考:



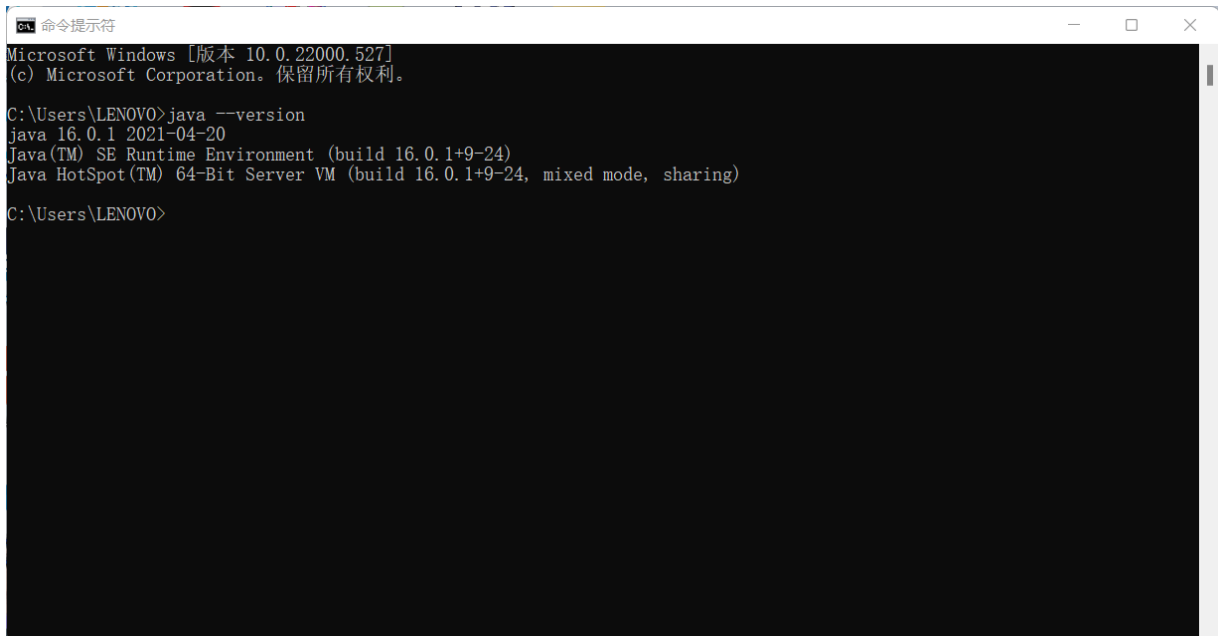
## 运行Java程序

首先需要安装Java

官网:[Java Downloads | Oracle](https://www.oracle.com/in/java/technologies/javase-downloads.html)

默认下载安装即可, 安装完毕后打开cmd, 运行

```
java --version
```



```
命令提示符
Microsoft Windows [版本 10.0.22000.527]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\LENOVO>java --version
java 16.0.1 2021-04-20
Java(TM) SE Runtime Environment (build 16.0.1+9-24)
Java HotSpot(TM) 64-Bit Server VM (build 16.0.1+9-24, mixed mode, sharing)

C:\Users\LENOVO>
```

则说明安装成功，回到刚才**LeapYear.java**处的文件夹内，右键打开GitBash，执行

```
javac 文件名.java
```

进行编译，此处则是

```
javac LeapYear.java
```

编译完成后会生成.class文件，暂时先不管，继续执行

```
java 文件名 arguments
```

在文件名后隔一个空格输入一个参数，此处则是

```
java LeapYear 2000
```

表示判断2000是不是闰年

```
LENOVO@LAPTOP-62B4NA73 MINGW64 ~/Desktop/61b Tutorial/lab1 (master)
$ javac LeapYear.java

LENOVO@LAPTOP-62B4NA73 MINGW64 ~/Desktop/61b Tutorial/lab1 (master)
$ java LeapYear
Please enter command line arguments.
e.g. java Year 2000

LENOVO@LAPTOP-62B4NA73 MINGW64 ~/Desktop/61b Tutorial/lab1 (master)
$ java LeapYear 2000
2000 is a leap year.

LENOVO@LAPTOP-62B4NA73 MINGW64 ~/Desktop/61b Tutorial/lab1 (master)
$
```

完成该lab

最开始的几周是使用 `javac` , `java` 的 命令行进行编译, 之后会使用 `IntelliJ IDEA`, 网站也会教大家配置

---

## 4. 上传代码至Github

---

在完成自己的代码作业后, 需要提交到gradescope上进行评测, 而gradescope是读取你的github上的代码仓库, 因此此处需要先将本地代码上传至github:

### git1. 绑定你的远程仓库

步骤2中, 你已经创建了一个远程仓库, 并记下仓库的SSH链接, for example:

```
git@github.com:Fallenpetal/CS61B-Tutorial.git
```

接下来需要将本地仓库与远程仓库进行关联:

回到刚才的文件夹里, 打开Git Bash, 执行:

```
git remote add origin 你的cd仓库链接
```

此处示例则是:

```
git remote add origin git@github.com:Fallenpetal/CS61B-Tutorial.git
```

绑定完成

### 2. 查看当前仓库状态与上传

执行命令:

```
git status
```

可以查看当前仓库状态

```
LENOVO@LAPTOP-62B4NA73 MINGW64 ~/Desktop/61b Tutorial/lab1 (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   LeapYear.java

no changes added to commit (use "git add" and/or "git commit -a")
LENOVO@LAPTOP-62B4NA73 MINGW64 ~/Desktop/61b Tutorial/lab1 (master)
$
```

可以发现**标红**处正是刚才我们修改的代码，接下来将代码上传至github:

执行:

```
git add LeapYear.java
```

**Optional:**如果需要添加的文件很多，也可以使用

```
git add --all
```

一次性添加所有代码，然后执行:

```
git commit -m"填写一些你想说明的信息"
```

示例:

```
LENOVO@LAPTOP-62B4NA73 MINGW64 ~/Desktop/61b Tutorial/lab1 (master)
$ git add --all

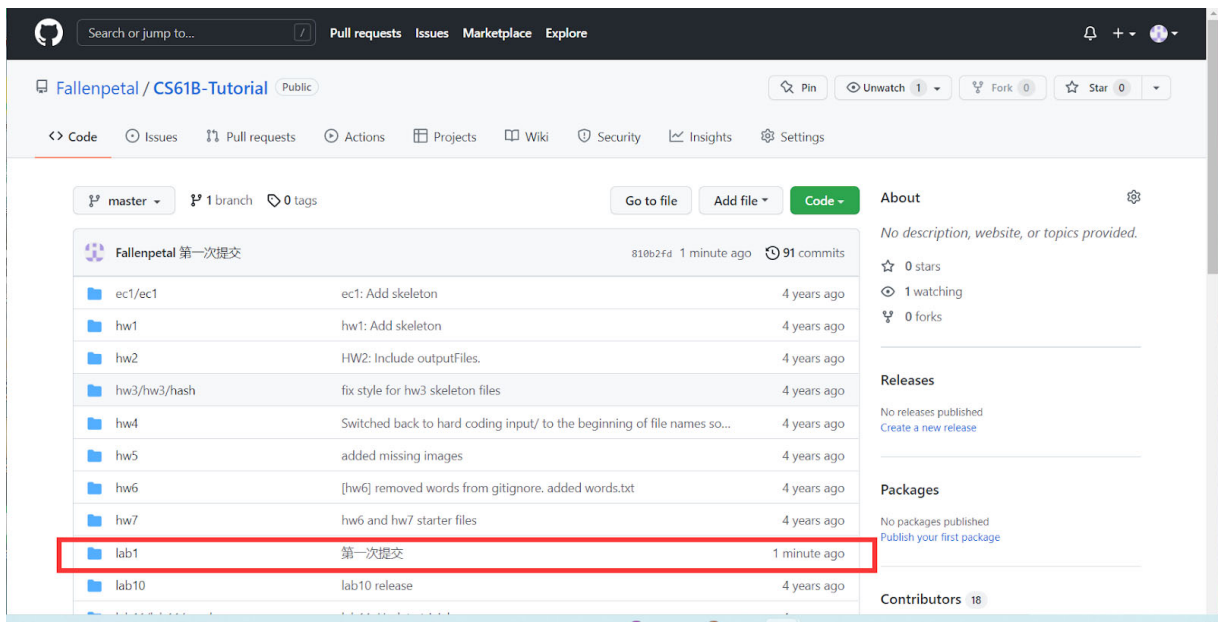
LENOVO@LAPTOP-62B4NA73 MINGW64 ~/Desktop/61b Tutorial/lab1 (master)
$ git commit -m"第一次提交"
[master 3a87206] 第一次提交
1 file changed, 4 insertions(+), 2 deletions(-)
```

最后执行

```
git push origin master
```

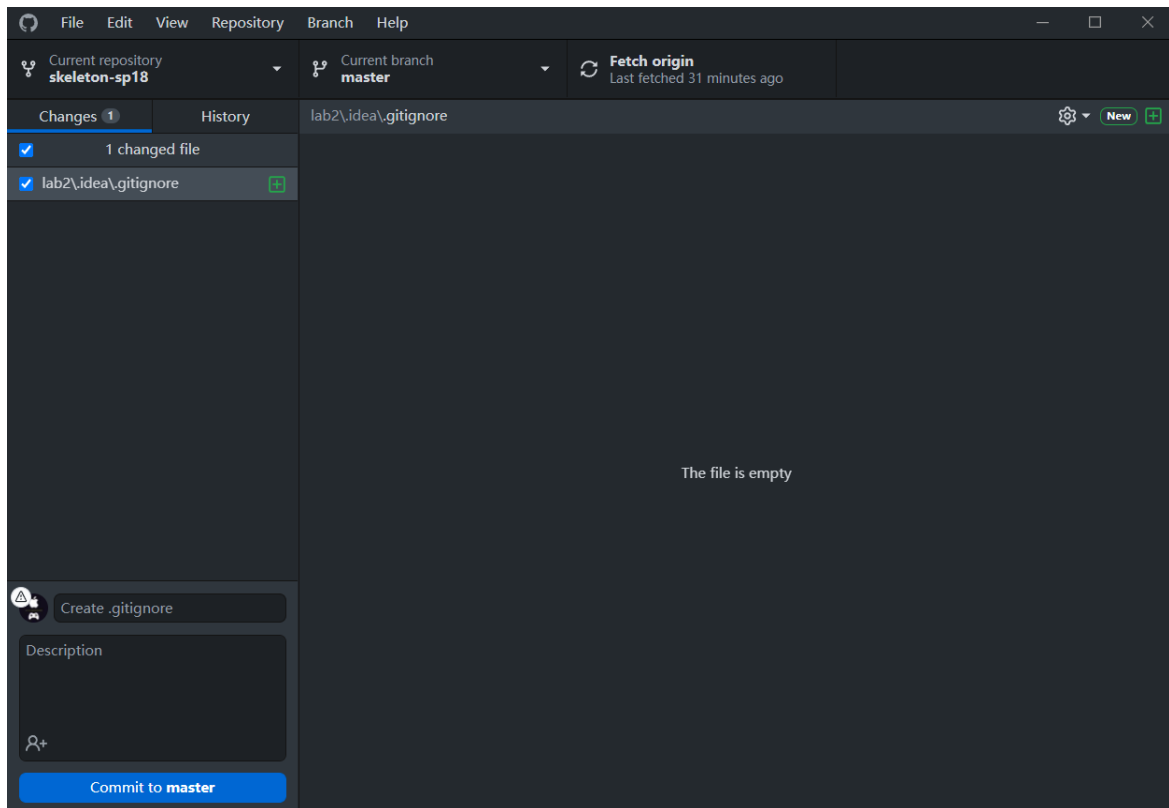
静待片刻后，出现

此时再去查看Github的远程仓库，发现本地代码已经全部上传至Github





## (optional) 推荐使用GUI软件 github desktop



先添加你的仓库进来：file→add local ... →选择你的仓库文件夹

之后每次修改可以用过图中的蓝色√来勾选，选中相当于add了它

然后左下角输入commit的信息，点击“commit to master”

点击图中Fetch origin那个位置的按钮，即可实现push（我这里是要先fetch，你们用的时候多点几次那个按钮即可）

实测比各种ide绑定的git以及命令行操作出问题概率要小很多，很多人的命令程序可能都是不过代理的，所以也可能遇到网络问题

---

## w5.Gradescope评测

由于spring 2018的ab1已过期，作为测试，我们可以选择spring 2021的版本进行Lab1的测试（只是lab1选择Spring2021进行测试，其余的lab以I2018为准）

Spring 2021 : s

选择Lab1,提交方式选择Github

CS 61B (Public) | Spring 2021

DESCRIPTION

In 61A, the correctness of a program was our primary goal. In CS 61B, we're concerned also with engineering. An engineer, it is said, is someone who can do for a dime what any fool can do for a dollar. Much of 61B will be concerned with the tradeoffs in time and memory for a variety of methods for structuring data. We'll also be concerned with the engineering knowledge and skills needed to build and maintain moderately large programs.

NAME	STATUS	RELEASED	DUE (PST)
Lab 1: Welcome to Java	No Submission	SEP 01	1 year, 10 months left DEC 31 AT 11:59PM LATE DUE DATE: DEC 31 AT 11:59PM
Lab 2: Debugging	No Submission	SEP 01	1 year, 10 months left DEC 31 AT 11:59PM LATE DUE DATE: DEC 31 AT 11:59PM
Lab 3: Randomized Testing and Timing	No Submission	SEP 01	1 year, 10 months left DEC 31 AT 11:59PM LATE DUE DATE: DEC 31 AT 11:59PM
Lab 4: Debugging	No Submission	SEP 01	1 year, 10 months left DEC 31 AT 11:59PM LATE DUE DATE: DEC 31 AT 11:59PM

选择自己的仓库和分支（注意要选对，也就是说你的代码在哪个分支就选哪个，有可能是main，只是名字而已）

Submit Programming Assignment

Upload all files for your submission

SUBMISSION METHOD

☐ Upload ☒ GitHub ☐ Bitbucket

CONNECT YOUR ACCOUNT

REPOSITORY

Select a repository...

BRANCH

Select a branch...

最后等待评测结果即可

最后说明一下，本案例只是教大家如何编写Java  
Spring 2021 的lab1不是判断闰年，望周

程序，编译运行并提交，实际上

**Autograder Results** Results Code

Advice from your friendly neighborhood Academic Intern:  
'Deeply understanding every project and HW is essential.'

**File Checking (0.0/0.0)**

\* Found required files for Lab.

**Compilation (0.0/0.0)**

Compiling tests for Lab...  
success.

**a001) HelloNumbers (16.0/16.0)**

Your HelloNumbers output: 0 1 3 6 10 15 21 28 36 45  
Expected output: 0 1 3 6 10 15 21 28 36 45

**b001) Collatz (16.0/16.0)**

Your Collatz output: 5 16 8 4 2 1  
Expected output: 5 16 8 4 2 1

**STUDENT**  
liuxing

**AUTOGRADER SCORE**  
**32.0 / 32.0**

**PASSED TESTS**  
File Checking (0.0/0.0)  
Compilation (0.0/0.0)  
a001) HelloNumbers (16.0/16.0)  
b001) Collatz (16.0/16.0)

知，按照以上步骤提交会出现下图(因为不是同一个lab哈哈哈)，右上角即为打分

**Autograder Results** Results Code

Advice from your friendly neighborhood Academic Intern:  
'Start projects early.'

**File Checking (0.0/0.001)**

\* Missing required files for Lab:  
- HelloNumbers.java  
- Collatz.java

**Assessment for Lab (0.0/32.0)**

Unable to run assessment for Lab: missing required files

**STUDENT**  
someone

**AUTOGRADER SCORE**  
**0.0 / 32.0**

**FAILED TESTS**  
File Checking (0.0/0.001)  
Assessment for Lab (0.0/32.0)

## 提交作业的流程

以后的作业过程均为以上步骤，在写完代码后，提交即可，具体git 命令：

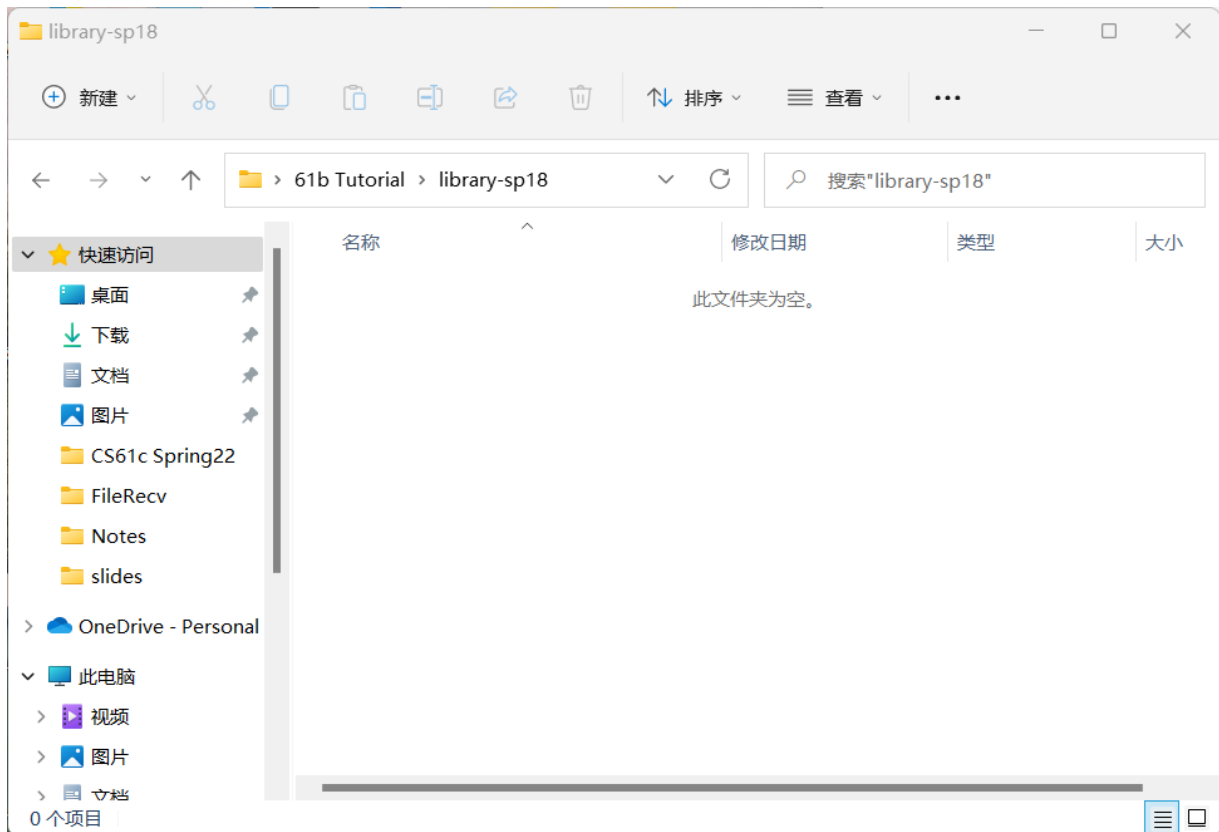
```
git status      查看当前仓库状态，标红文件即为修改的文件
git add --all
git commit -m "说明"
git push origin master
```

这些命令均在上文出现，具体使用效果参见上文提交LeapYear.java的过程

## 6.关于额外的说明

### 下载library-sp18

在后面进行Project 0的作业过程中，需要用到**library-sp18**文件夹，进入我们的本地仓库打开发现文件夹为空



在群里下载解压即可

聊天 公告 相册 文件 作业 设置

共99个文件 (已使用1.18GB/10GB)

文件	更新时间	过期时间	大小	上传者	下载次数
大话数据结构 by 程杰 (z-lib.org).pdf	2021-11-13	永久	44.6MB	.	21次
library-sp18.zip	2021-11-11	永久	3.23MB	一直站...	11次
javajlib.rar	2021-11-03	永久	2.71MB	新加坡...	12次
main.pdf	2021-11-02	永久	2.06MB	新加坡...	49次
ucb.png	2021-10-28	永久	642KB	61b	23次
SICP Python 描述 中文版 by it-ebook...	2021-10-28	永久	2.66MB	61b	33次
截屏2021-10-09 21.20.01.png	2021-10-09	永久	58.5KB	kicki	15次
14261834_如何高效阅读 BREAK-THR...	2021-10-03	永久	36.9MB	修梦	53次

以后如有坑点待补充ing.....

## 7.关于idea的使用：Lab2-setup

这

种大型ide都有严格的项目结构，参考官网lab2的pre-lab，一个坑点，2018版本的idea比较旧，现在找不到import那个按钮，请打开idea后：file→new→import from existing source，若直接用open打开会识别不了项目结构