

System Setup & Grader Intro.

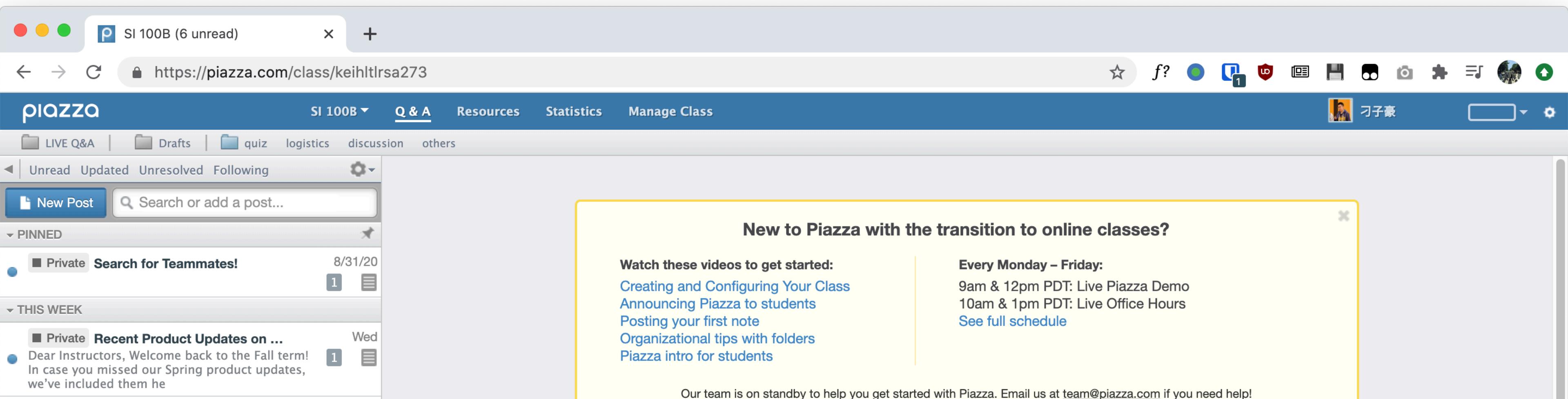
SI100B Fall 2020 Tutorial Week 1

Zihao Diao

Administrivia

Tools

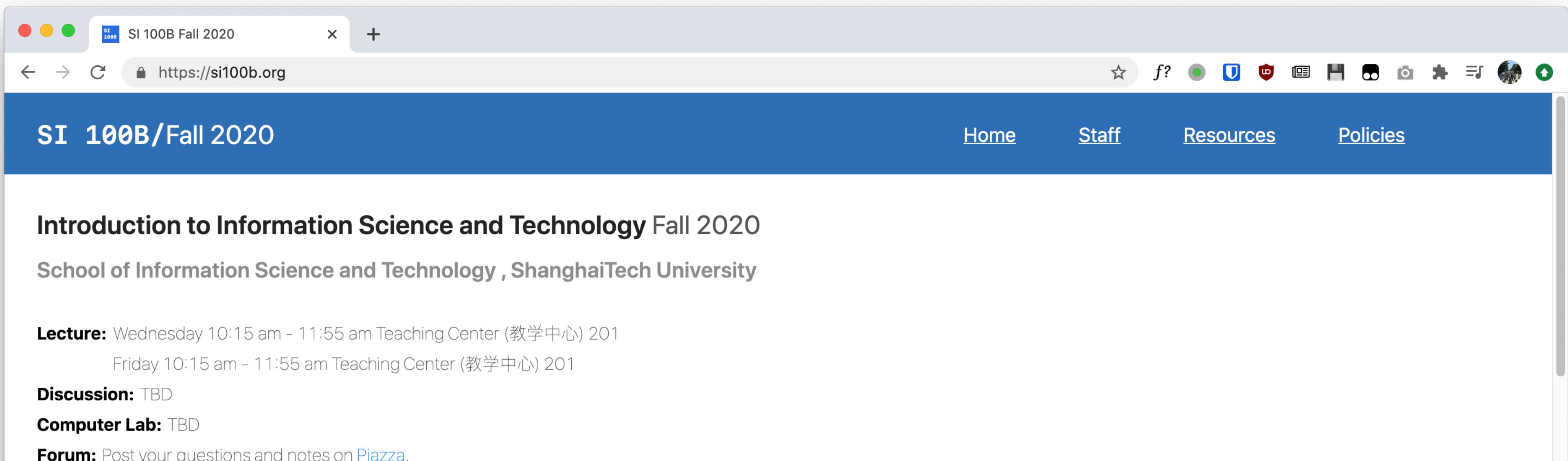
- **Piazza:**
 - Post your questions and notes on Piazza;
 - Register at <http://piazza.com/shanghaitech.edu.cn/fall2020/si100b>;
 - Check at least every 12 hrs for updates (new lecture slides, new assignments etc.);
 - Change your username to your Chinese name (e.g. 王大锤 or 丁<space>一)



Administrivia

Tools

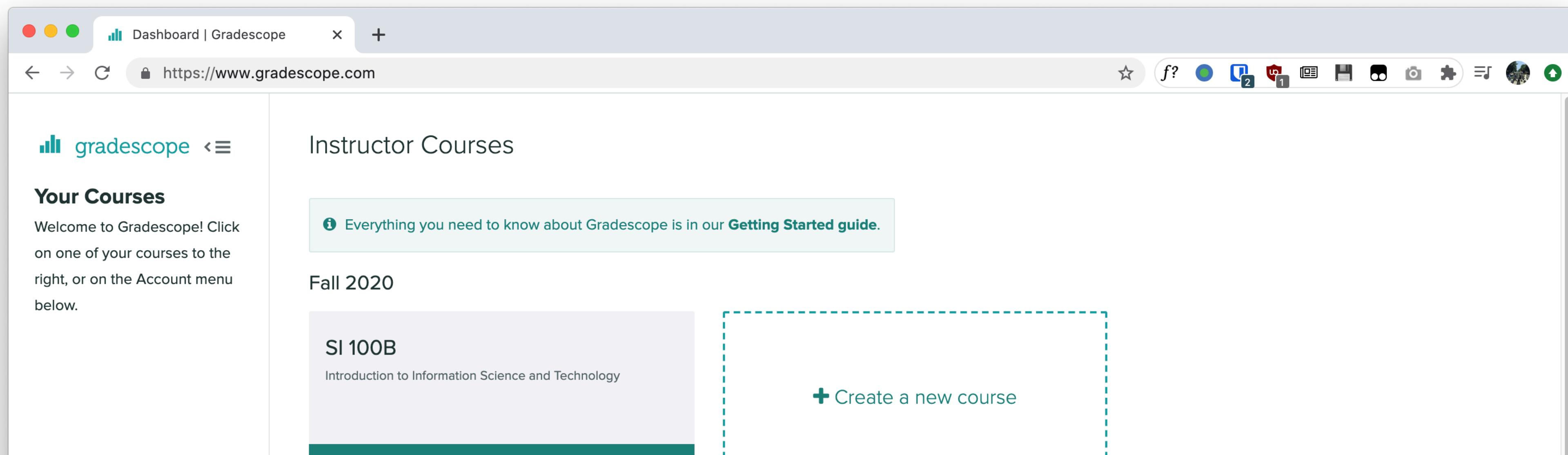
- **SI100B.org:**
 - The course homepage: general information;
 - Course syllabus, reading material, OH, policy etc.;
 - Check it for reference;



Administrivia

Tools

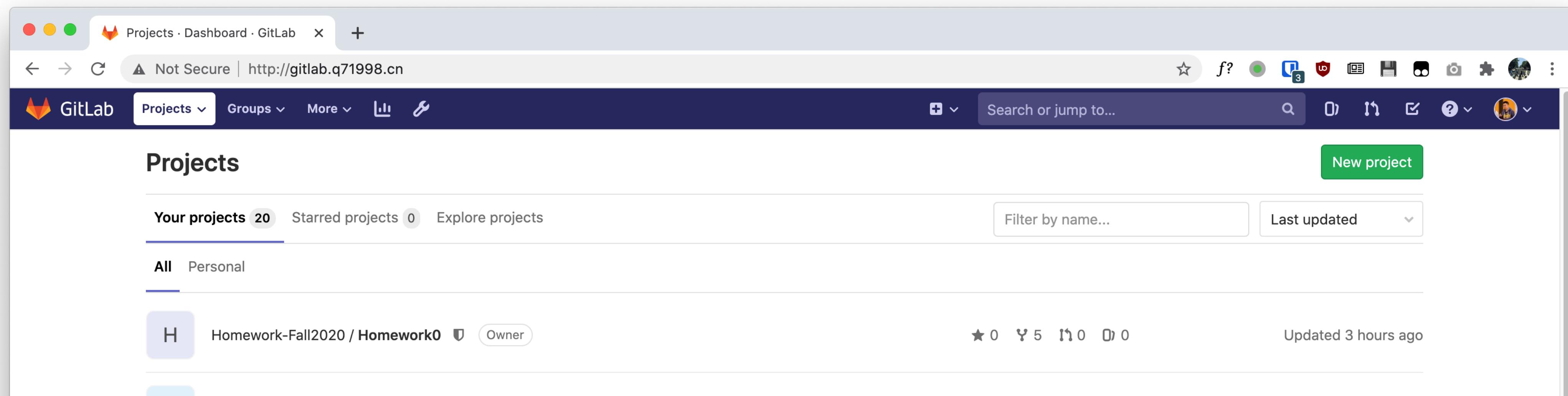
- **Gradescope:**
 - Submit your writing assignment here;
 - Register at <https://www.gradescope.com> with entry code 9D7J7V;
 - Fill in your Chinese name and student ID in your settings;



Administrivia

Tools

- **Auto-grader** <http://gitlab.q71998.cn>:
 - Submit your programming assignment here;
 - Password email has been sent to your ShanghaiTech email. Change your password ASAP;
 - User manual at <https://si100b.org/content/gitlab-manual.pdf>
 - Usage subject to the course code of conduct: <https://si100b.org/resource-policy/#policies>



Administrivia

Policy

- **Zero-tolerance towards plagiarism**
 - Fact: we have sophisticated technologies and tools to detect plagiarism and we will use them;
 - You may not:
 - Read or possess solution code written by other people;
 - Submit code derived from code written by other people;
 - Allow other people to read or possess solution code by yourself actively or passively;
 - When in doubt, ask before acting;
 - Full code of conduct: <https://si100b.org/resource-policy/#policies>

Administrivia

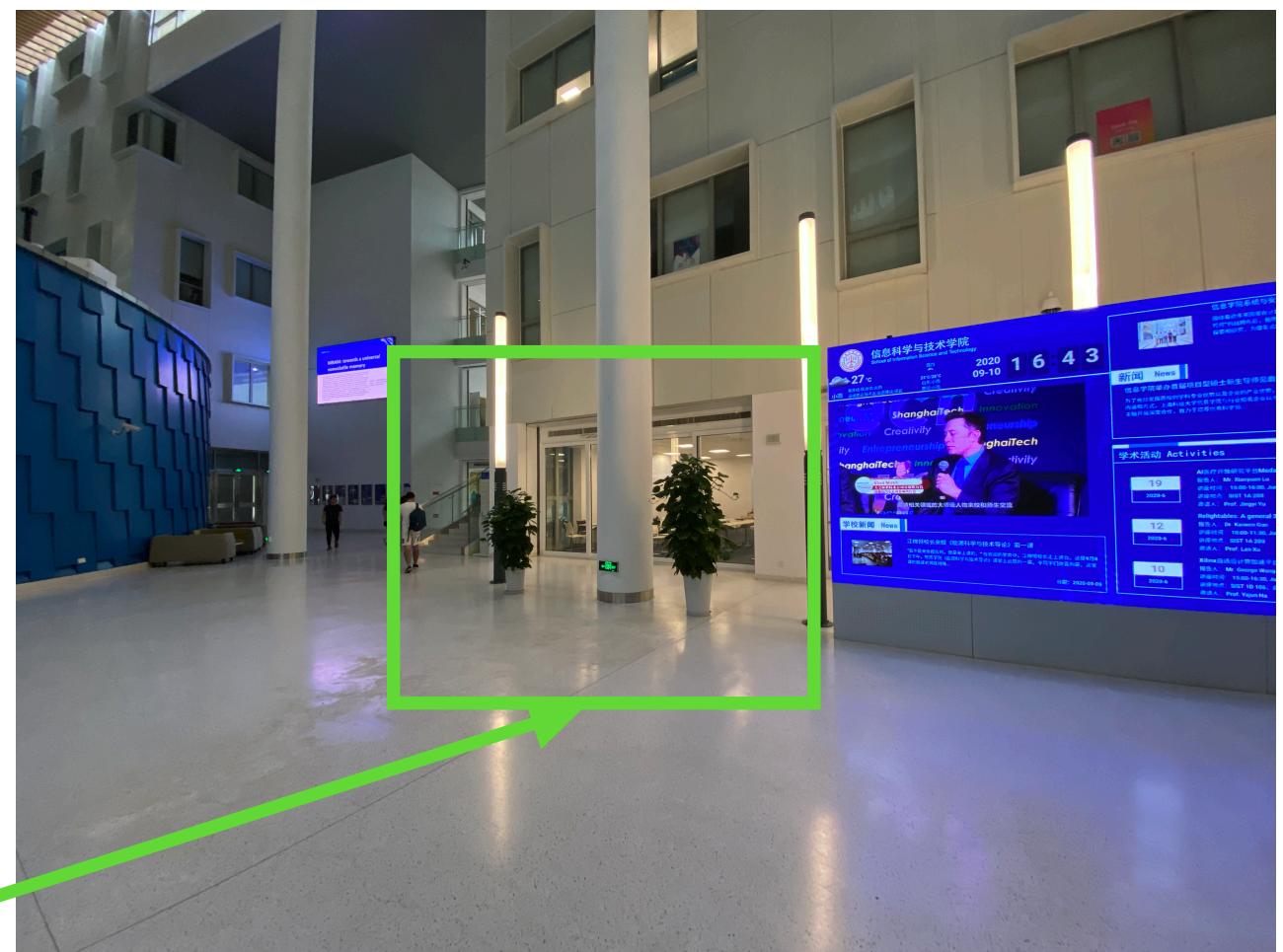
Assignments

- **Writing Assignment:** Logistics Assignment
 - Due: Sept. 16 23:59 (next Wed.) on Gradescope;
 - Concerning the **code of conduct** (<https://si100b.org/resource-policy/#policies>) of SI100B;
 - 1% of your final score;
- **Programming Assignment 0:** System Setup
 - Due: Sept. 16 23:59 (next Wed.) on auto-grader;
 - Dummy assignment help you to set up the Python environment on your computer;

Administrivia

Discussion & OH

- **Office Hour:**
 - Time for asking questions to TAs face to face - high efficiency;
 - Time table: <https://si100b.org/>, starting this Fri.;
 - Location: SIST 1B-101
- **Discussion:**
 - TAs lecture supplementary materials or introductions to PAs - just like this one;
 - Three more discussions for Python part;
 - Vote for time slot on Piazza.



System Setup

Python Interpreter

- **Installation**
 - Linux (Ubuntu Linux e.t.c.):
 - Your system should ship with Python 3;
 - Check your version: SI100B is using Python version 3.7:
 - Inconsistent version may cause problem;
 - Install with your package managers:
 - For Ubuntu Linux: `apt update && apt install python3`

System Setup

Python Interpreter

- Installation
 - macOS:
 - Your system should ship with Python 3;
 - macOS Catalina comes with Python version 3.8:
 - Or you can install it with homebrew:
 - Install homebrew from <https://brew.sh/>
 - Install Python 3.7 with `brew install python@3.7`

System Setup

Python Interpreter

- Installation
 - Windows:
 - Option 0 (recommended): Install a Linux distro as a dual-boot to your Windows
 - Ubuntu Linux is recommended for green hand: <https://www.ubuntu.com/>

System Setup

Why Linux (UNIX)?

- For CS student:
 - Linux (or other UNIX) is an unavoidable topic during your whole career;
 - CA and OS course require you to use Linux;
 - Many software you will use in your research only have Linux version;
 - Your tools (like Python interpreter) is designed for UNIX — use Linux for better UX and performance;
 - Start early means you will get familiar with Linux earlier — huge advantage for yourself;
- For EE student:
 - Linux (or other UNIX) is unavoidable if you want to have a better taste in CS areas (like some of you will do)

System Setup

Python Interpreter

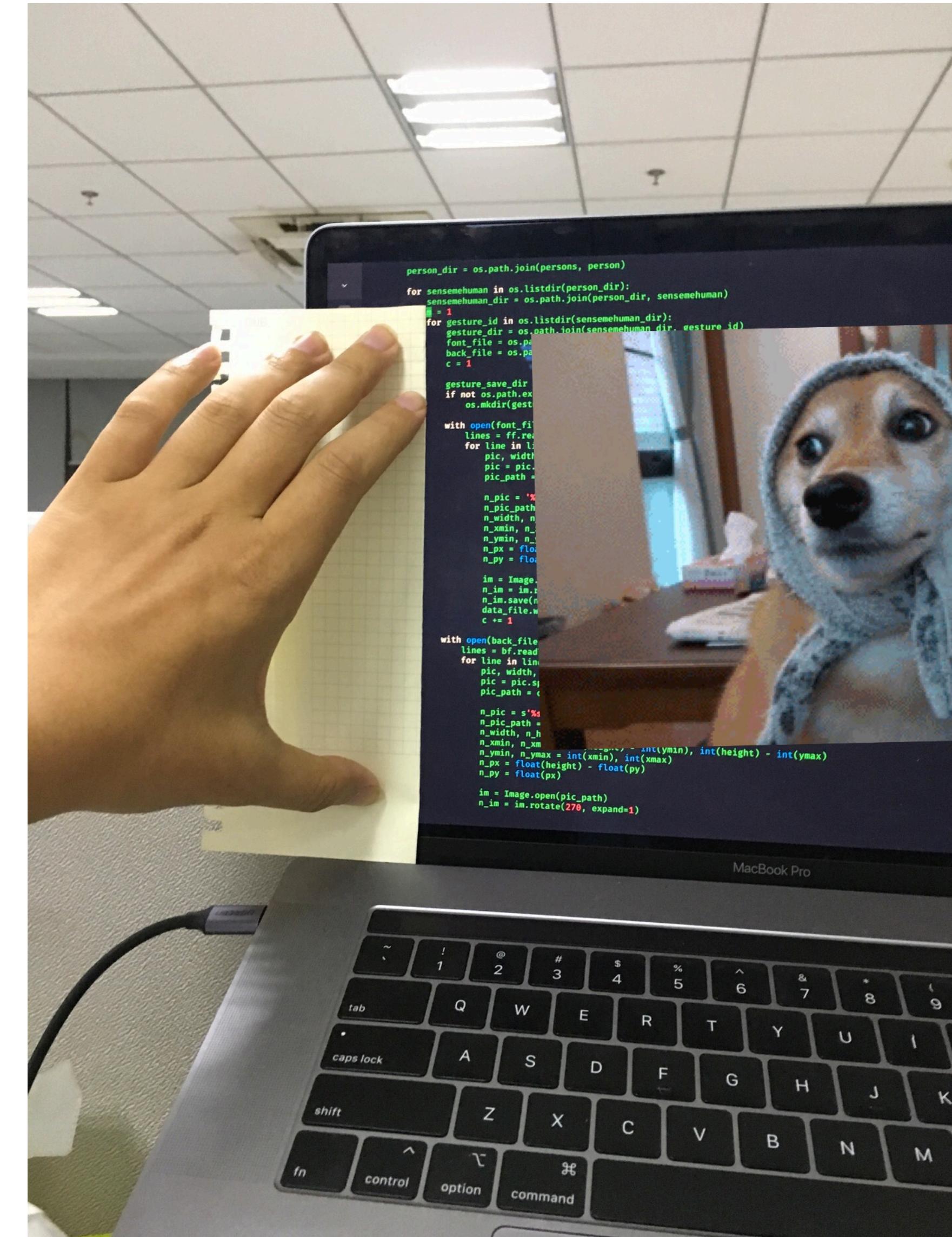
- **Installation**
 - Windows:
 - Option 1: Use Windows Subsystem of Linux (WSL)
 - Follow instructions in <https://docs.microsoft.com/en-us/windows/wsl/install-win10>;
 - Then use WSL the same way as a Linux
 - Option 2: Install Python on your Windows machine:
 - Download the setup from <https://www.python.org/downloads/windows/> and install it.

System Setup

Editors & IDEs

- **Editor or IDE?**

- Fact: you can use the most basic text editors to write your Python program (e.g., 记事本 on Windows);
- Modern editors have more functions like syntax highlight than the most basic text editors and remain somewhat lightweight;
- IDE — Integrated Development Environment — include almost everything you need for a huge project like debugger, build system integration and so on. Thus very heavy;

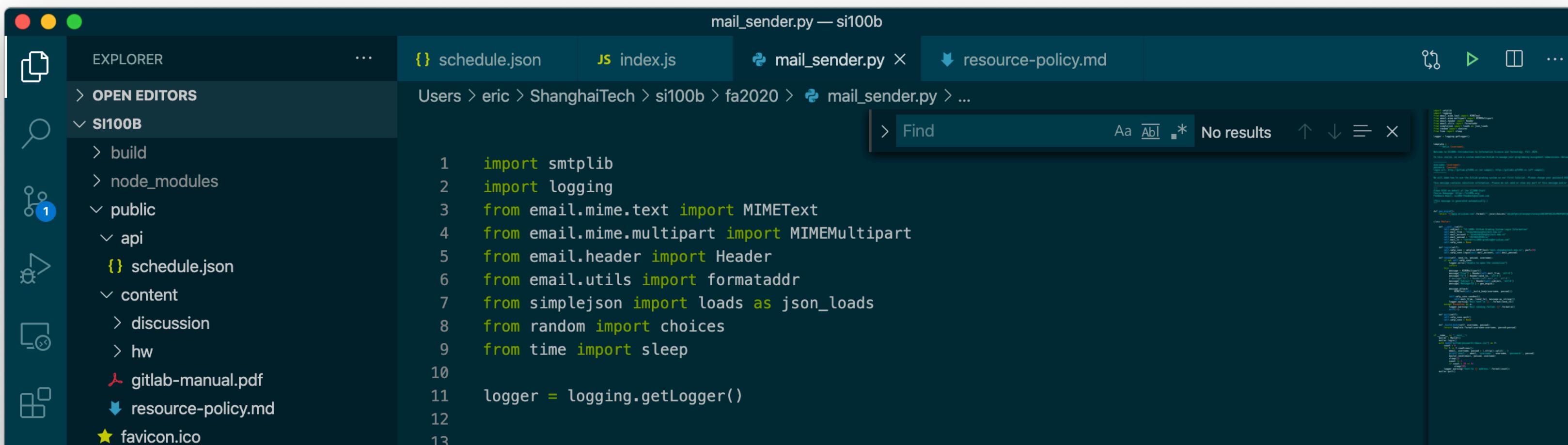


System Setup

Editors & IDEs

- **If you want to use an editor...**

- You can use VS Code:
 - Developed by Microsoft, free and (mostly) open source;
 - Have some functions that historically only exists in IDEs while remain somewhat lightweight;
 - Download from <https://code.visualstudio.com/> for your platform;
 - Or you can use Sublime, or use vim if you are power user.



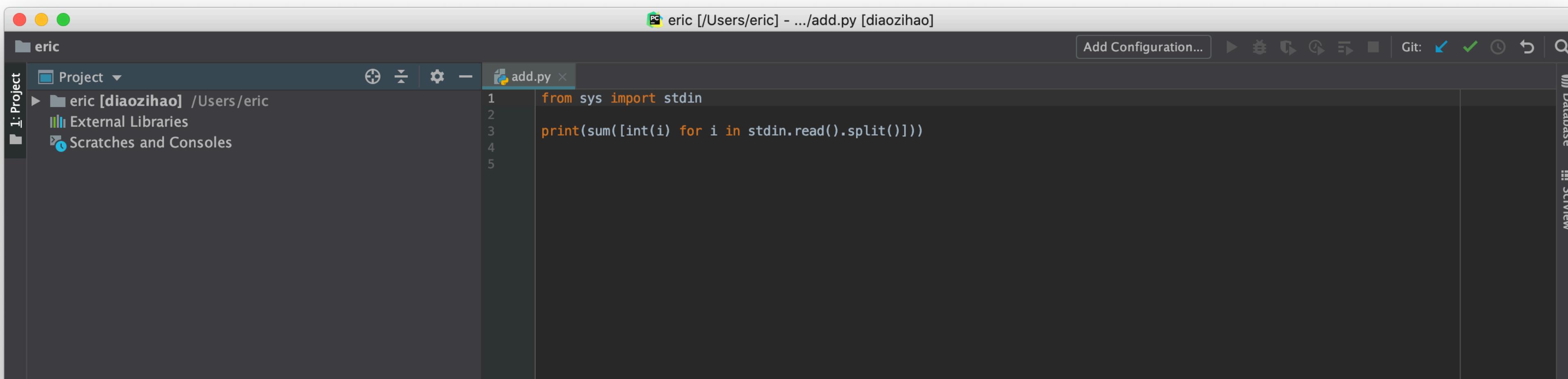
The screenshot shows the Visual Studio Code (VS Code) interface. The title bar indicates the file is "mail_sender.py — si100b". The left sidebar shows the "EXPLORER" view with a tree structure of files and folders, including "SI100B", "build", "node_modules", "public", "api", "content", "discussion", "hw", "gitlab-manual.pdf", "resource-policy.md", and "favicon.ico". The main editor area displays Python code for "mail_sender.py":

```
1 import smtplib
2 import logging
3 from email.mime.text import MIMEText
4 from email.mime.multipart import MIMEMultipart
5 from email.header import Header
6 from email.utils import formataddr
7 from simplejson import loads as json_loads
8 from random import choices
9 from time import sleep
10
11 logger = logging.getLogger()
```

System Setup

Editors & IDEs

- **If you want to use an IDE...**
 - You can use PyCharm:
 - Developed by JetBrains from Czech, commercial software;
 - Full function Python IDE;
 - Free Community version with limited functions;
 - Could apply for free Professional version with your ShanghaiTech email;
 - Get from <https://www.jetbrains.com/pycharm/>



The screenshot shows the PyCharm IDE interface. The title bar indicates the project is 'eric' and the current file is 'add.py'. The code editor displays the following Python script:

```
from sys import stdin
print(sum([int(i) for i in stdin.read().split()]))
```

The left sidebar shows the project structure with a single folder 'eric [diaozihao] /Users/eric'. The right sidebar includes tabs for 'Database' and 'SciView'.

System Setup

Editors & IDEs

- If you want to use an IDE...
 - Or you can use Spyder:
 - Open source software;
 - Download from <https://www.spyder-ide.org/>

The screenshot shows the Spyder Python IDE interface. The top menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, and Help. The toolbar below has icons for file operations like Open, Save, and Run. The main window features several panes: Project explorer, Editor, Outline, and Variable explorer.

Project explorer: Shows a tree view of the project structure. The root folder is 'Spyder' containing 'Data', 'spyder', and 'tests'. Under 'tests', there are files: '_init_.py', 'cli_options.py', 'mac_stylesheet.qss', 'mainwindow.py', 'restart.py', and 'start.py'.

Editor: Displays a Python script named 'interpolation.py' with the following code:

```
6
7 import pylab
8 from numpy import cos, linspace, pi, sin, random
9 from scipy.interpolate import splprep, splev
10
11 # % Generate data for analysis
12
13 # Make ascending spiral in 3-space
14 t = linspace(0, 1.75 * 2 * pi, 100)
15
16 x = sin(t)
17 y = cos(t)
18 z = t
19
20 # Add noise
21 x += random.normal(scale=0.1, size=x.shape)
22 y += random.normal(scale=0.1, size=y.shape)
```

Variable explorer: A table showing the variables and their properties:

Name	Type	Size	Value
array_int8	int8	(2, 3)	Min: -7 Max: 6
array_uint32	uint32	(2, 2, 3)	Min: 1 Max: 7
bars	container.BarContainer	20	BarContainer object of matplotlib.conta...
df	DataFrame	(3, 2)	Column names: bools, ints
filename	str	1	C:\ProgramData\Anaconda3\lib\site-pac...
list_test	list	2	[Dataframe, Numpy array]
nrows	int	1	344
r	float64	1	7.611082589334796
radii	float64	(20,)	Min: 0.4983036638535687 Max: 9.856848974942551
region	tuple	2	(slice, slice)

Basics of UNIX Shells

cd	change directory
mkdir	make directory
mvdir	move directory
rmdir	remove directory
ls	dirs and files in current dir
cp	copy file/dir
mv	move file/dir
rm	delete a file/dir
cat	display a file in terminal
diff	compare differences between files

Basics of UNIX Shells

Demo

Linux Misc.

Use Mirrors

- Many Linux package manager (i.e., *App Store* for Linux) are using website abroad — slow access;
- Solution: Setup local server that periodically download package from those server and let you download from the local server (crude CDN servers);
- Choices:
 - GeekPie Mirrors (on-campus): <https://mirrors.geekpie.club/>
 - Tsinghua TUNA: <https://mirrors.tuna.tsinghua.edu.cn/>
 - USTC Mirrors: <https://mirrors.ustc.edu.cn/>
 - User guide available.

System Setup

Git

- **What is git?**
 - Version control tool;
 - It tracks every lines of code you write and allow you to go back to a history version;
 - Developed by Linus Torvalds - the man behind Linux - in 10 days;
 - **Not** a homework submitting tool:
 - But we will use it for homework submission
 - **Not** a specific service or website:
 - GitHub and GitLab are 2 popular websites that provide git hosting service.

System Setup

Git

- **Installation**
 - **Linux:**
 - Your distro may ship with git, if so use it without effort;
 - Otherwise install it with your package manager.
 - E.g., For Ubuntu Linux: `apt install git`.
 - **macOS:**
 - Open your Terminal.app (终端.app), run `xcode-select --install`.
 - **Windows:**
 - Download the setup from <https://git-scm.com/downloads> or use WSL (last slide)

Grader Intro.

Basic Info.

- The grader is build upon git;
 - Using git is the only way of submitting your solution code.
- Full manual on the course homepage: <https://si100b.org/content/gitlab-manual.pdf>;
- Environment: CPython interpreter version 3.7.6 on a amd64 machine running Ubuntu 16.04 LTS.

Grader Intro.

Regulation

- Usage of the grader is subject to the course code of conduct: <https://si100b.org/resource-policy/#policies>;
- Under NO circumstance could you “borrow” your GitLab account out or “lend” other students’ GitLab account;
- No abuse of the grader.
- 30 chances of grading (i.e. `git tag`) by default, unlimited times of syncing up your local repository with the remote one (i.e. `git push`);
- You have at most 10 chances of grading request in every 24 hour. If you reach the limit, you will not be able to make a grading request.

Grader Intro.

Start

- **Step 1:** Clone the repo from the server;
 - Copy the URL from the grader's website;
 - Clone with `git clone <URL you copied>`;
- **Step 2:** Write and debug your code
 - The specification is in `README.md`

Grader Intro.

Submit

- **Step 1:** Add your modified files to the stage area:
 - Run `git add <files to add>;`
- **Step 2:** Make a local commit:
 - Run `git commit;`
 - Fill in your commit message in the opened text editor;
- **Step 3:** Tag your current commit:
 - Run `git tag <your new tag>;`
- **Step 3:** Push your commit to the grader:
 - Run `git push;`
 - The grader will grade your submission upon a pushed tag to it;

Grader Intro.

Demo

Gradescope Intro.

Demo

Questions?