

# DS2001 - CS Practicum

Spring 2023

## Practicum 0 - Welcome and Set Up

Practicum is DUE at the end of your scheduled practicum.

Work on the programming assignment below. You may not finish every problem every week, but we expect your best effort.

We will let you know when there are 15 minutes left in class – at that point, it is time to wrap up your code and move on to the self-reflection portion of the assignment! Read the self-reflection questions below, write your answers, and save them in a PDF.

Submit both your code and your self-reflection PDF on Gradescope (<https://www.gradescope.com/courses/482506>). Entry Code: Y7K77K

### Grading Policy

You will receive full credit if you submit your self-reflection (with thoughtful answers) and any code by the deadline. All questions in the self-reflection must be answered for full credit. **Note:** Practicum 0 is worth zero points (i.e., it is not graded).

### Feedback

We will provide feedback on your code if requested in your self-reflection. Otherwise, we will not look at your code. Our feedback will mirror the expectations of your DS200 homeworks, and we will use the same criteria for grading your projects in DS2001. So, make sure you ask for feedback when helpful, and read our notes!

## Today's Goals

By the end of practicum today you will:

- Know your classmates
- Have Anaconda installed
- Write a Python program with Spyder
- Submit an assignment on Gradescope

## Warm Up

### 1. Get to know your classmates

- a. As you read earlier, you are strongly encouraged to work on your practicum assignments in pairs. To make that easier, we will spend the first part of today getting to know each other! Below are instructions for a series of ice breakers. We wrote them out for your reference, but your instructor will take you through them step by step as a class.
- b. Ice breaker #1 (~10 mins)
  - i. Form a group of 3
  - ii. Introduce yourself to your groupmates (name, pronouns, anything else you want to share)
  - iii. Find a common attribute between the three of you (Do you all like the same color? Are you all from the same state? Are you all distance running superfans?!)
  - iv. Pick one spokesperson for your group
  - v. After 5 - 10 minutes we will come together as a class, and each group's spokesperson will introduce themselves, their groupmates, and share your common attribute
- c. Ice breaker #2 (~10 mins)
  - i. Form a **new group of 3**
  - ii. Introduce yourself to your groupmates (name, pronouns, anything else you want to share)
  - iii. As a group, discuss what you are excited to learn in this class and what you are nervous about for this class (spoiler: everyone is nervous about something, it's completely normal :) )
  - iv. Pick one spokesperson for your group
  - v. After 5 - 10 minutes we will come together as a class, and each group's spokesperson will introduce themselves, their groupmates, and share what your group is excited to learn and nervous about

## Programming Assignment

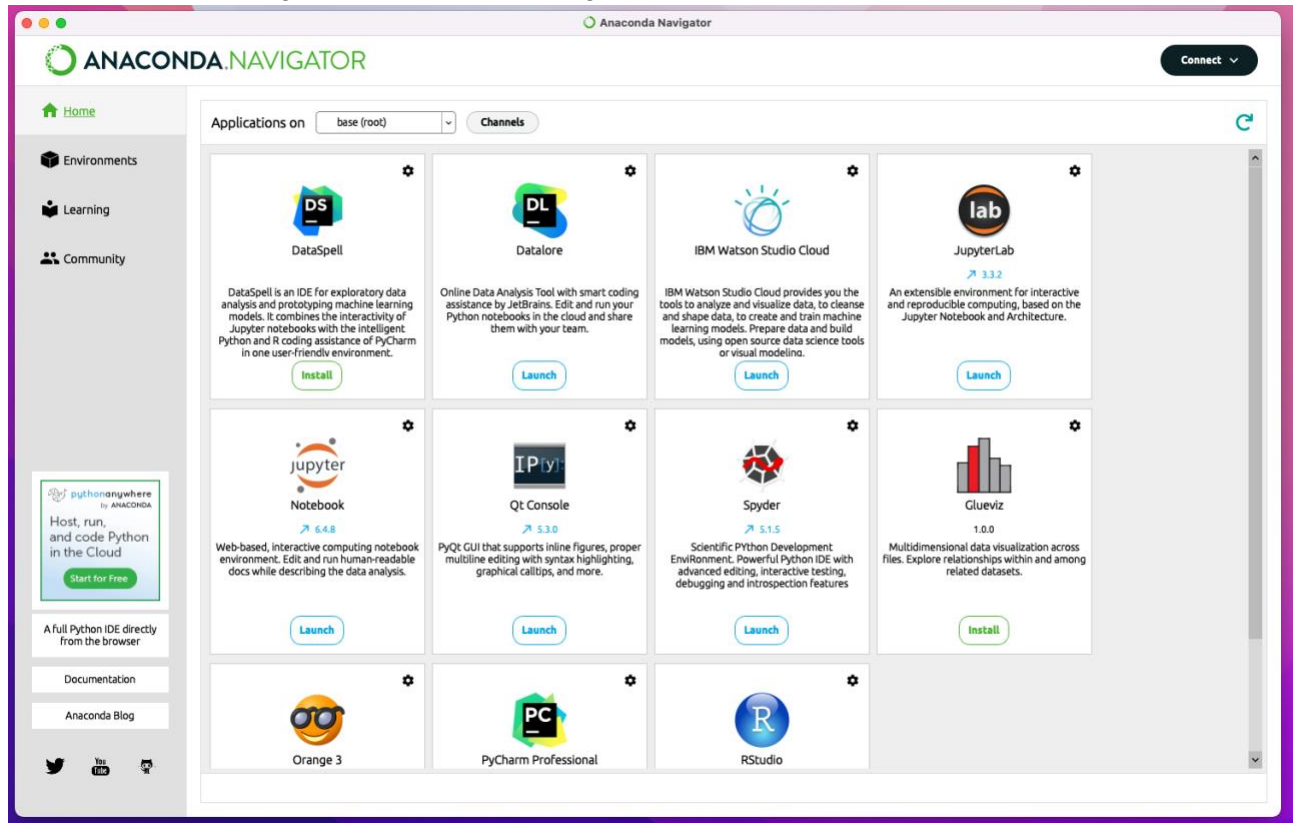
### 2. Install Anaconda

- a. Anaconda is a distribution of the Python (and R) programming language that will make your life simpler as you learn Python.
- b. Navigate to <https://www.anaconda.com/> and follow the link to download Anaconda.
- c. Once your Anaconda .pkg is downloaded, open it up and your computer's installer should launch automatically and walk you through the install process.
- d. If you run into any trouble ask Ab, or a TA for help (seriously, we want to help!)

### 3. Explore Anaconda

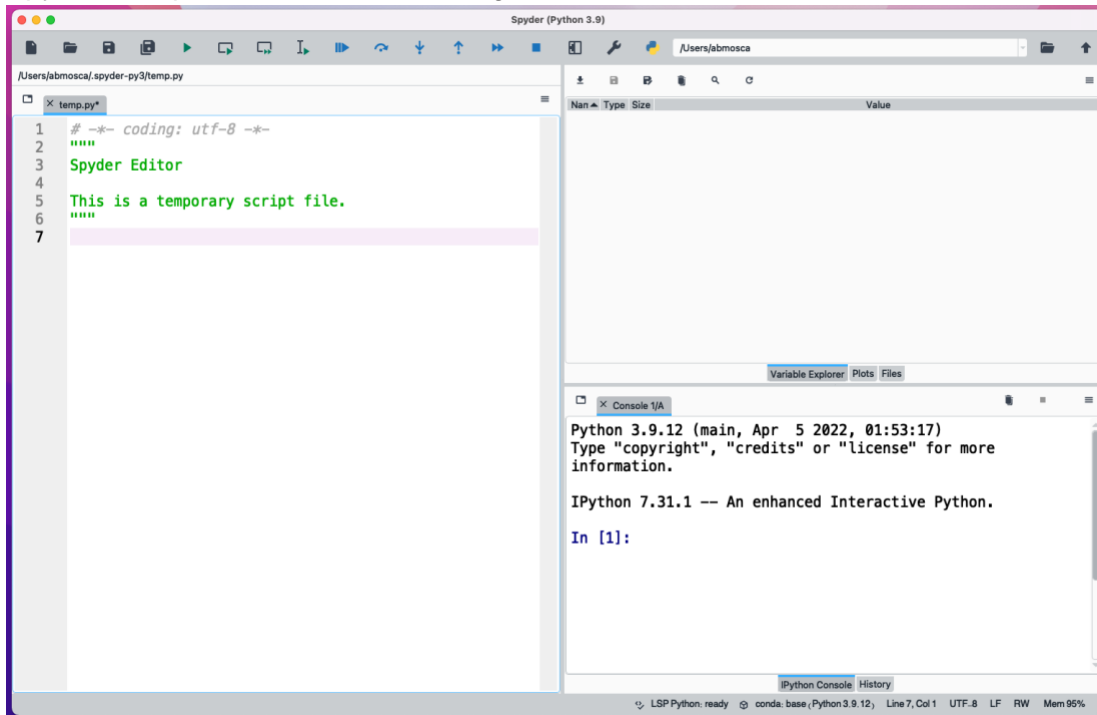
- a. Find "Anaconda Navigator" among your installed apps and open it.

- i. If you're like Ab, you will immediately get a message that “Anaconda Navigator needs to be updated.” Follow the prompts to update before moving on.
- b. Your Anaconda Navigator will look something like this:



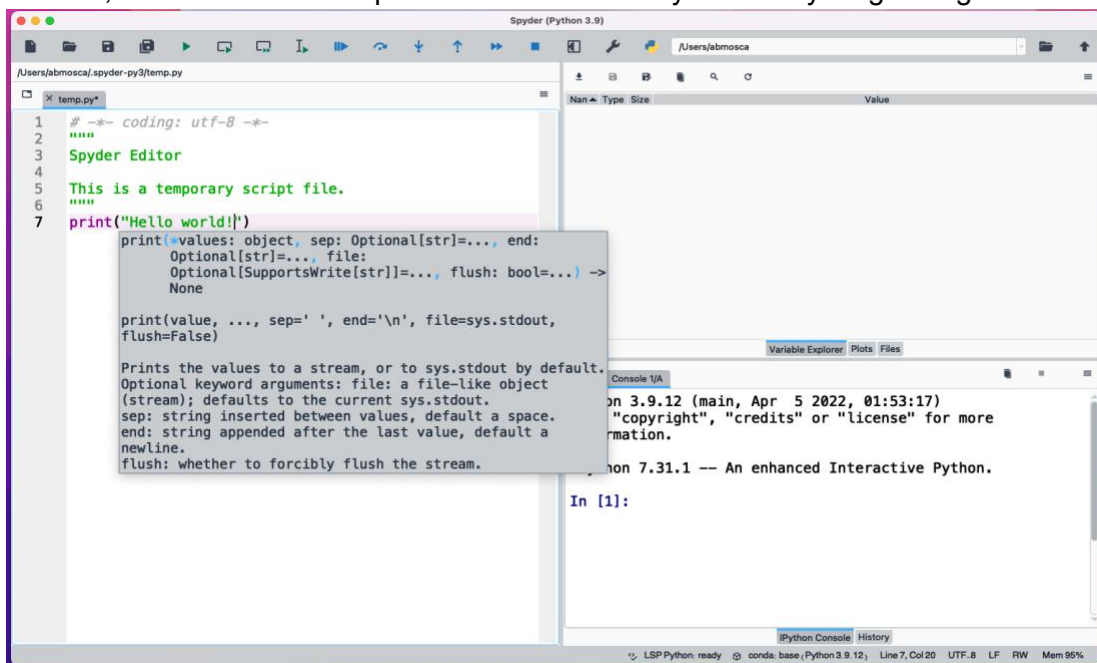
- c. In this class, we will use Spyder, which is an Integrated Development Environment (IDE) made for Python. What is an IDE? It's a tool that allows you to code, run code, track variables, and a whole bunch more all in one place.
- d. Launch Spyder by clicking the “launch” button (second row, third to the right in the screenshot above).
  - i. Again, you may immediately get a message that Spyder needs to be updated, too. Go ahead and update as instructed before moving on. (Spyder's instructions can be confusing and annoying – we can help you figure them out!)

- e. Spyder will open, and looks something like this:

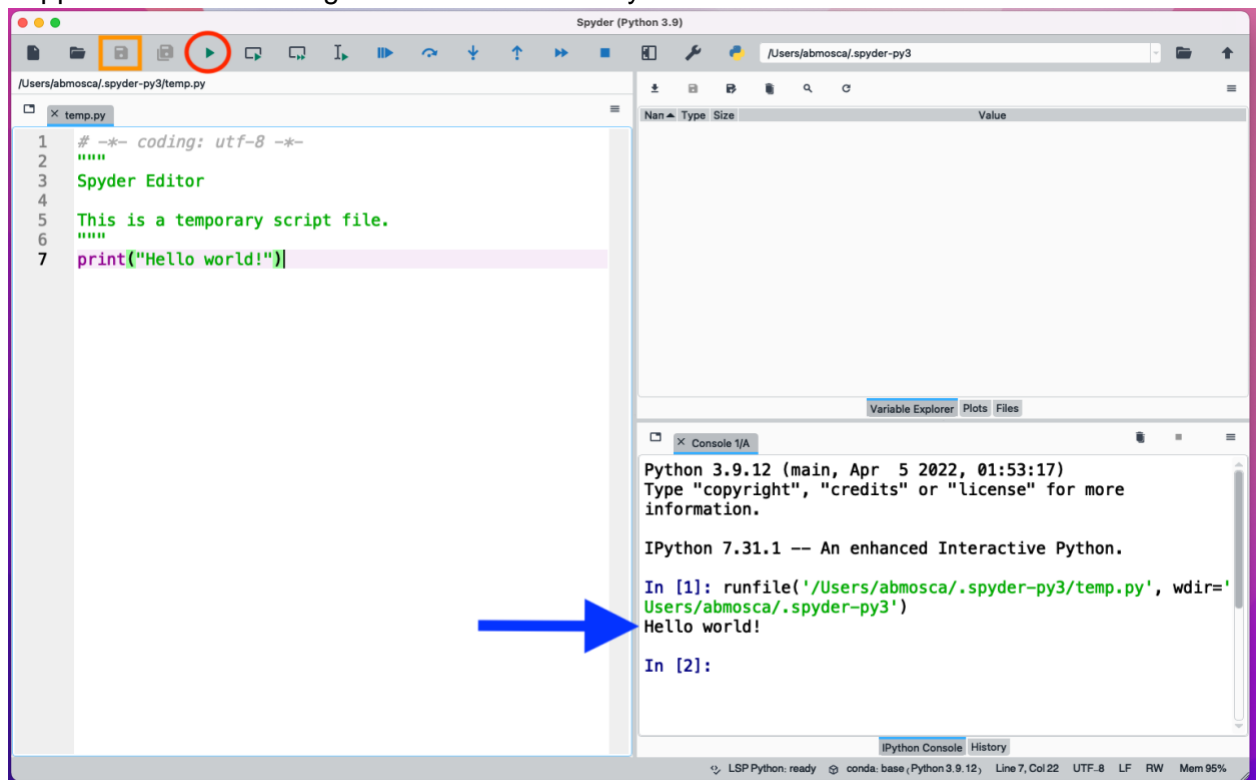


### 3. Write a Python Program

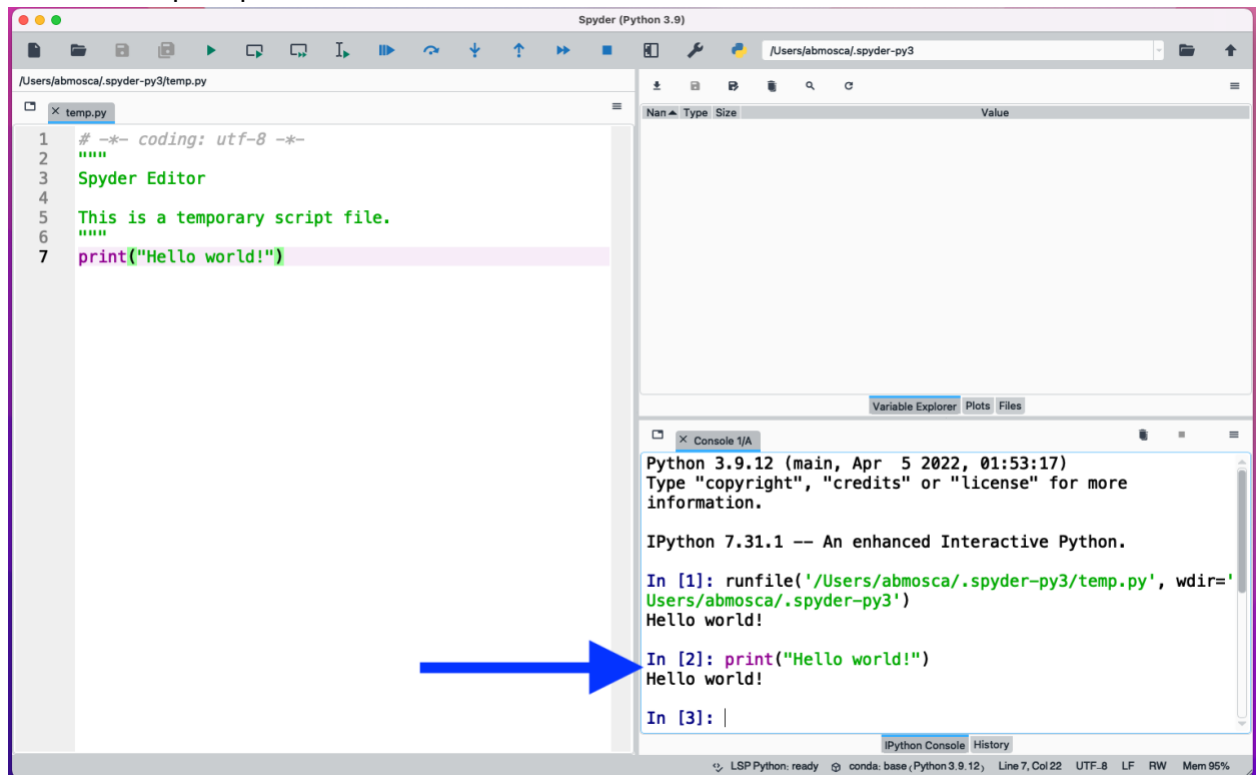
- a. On the left-hand side of the window in Spyder is a file editor. In the example below, a file named temp.py is open in the editor. This is the area where you can write and edit Python code to create programs. Add the following code to your file: `print("Hello world!")`. As you type, a box will pop up with some helpful hints and info. That's part of the IDE, it is meant to be helpful and doesn't mean you did anything wrong.



- b. To save your file, go to File > Save As. Similar to saving a Word document, you can rename the file if you want, and choose where you want it saved. Notice that the default folder is *not* the most intuitive place to save the file if you want to find it again later. We recommend choosing Desktop or Documents for now. Now that you have done “Save As” once, another option for saving is to click the white floppy disk in the top menu bar (squared in orange below).
- c. To run the code in `temp.py`, click the green triangle in the top menu bar (circled in red below). When you click the green arrow, a modal window will open asking you to select some settings. You can stick with the defaults for now and hit “Run”. Notice what happens in the bottom right of the window after you hit Run!



- d. Cool! The code `print("Hello world!")` prints the phrase Hello world! in the bottom right window! As you might notice from the picture, this window is called the console. We can also run commands directly in the console. Try typing `print("Hello world!")` at the next "In" prompt in the console then hit enter.



- e. Neat! That also prints the phrase Hello world!! You might be wondering what the difference between typing code in `temp.py` and hitting Run, and typing directly into the console is. Well, code you type in `temp.py` gets saved when you save `temp.py`. *Code you type directly in the console does not get saved.* This means that in order to have reproducible code (and something to turn in at the end of practicum!) you should get in the habit of creating `.py` files in the file editor and running them to test your code.
- f. There is of course a third window in Spyder that we have not talked about yet. That window will help you keep track of variables (among other things) and we're going to save talking about it in depth for a future lesson.
- g. You're off to a great start! Go ahead and move on to part 4 of this assignment.

#### 4. Submit an Assignment on Gradescope

- a. For this final portion of practicum, we'll practice submitting on Gradescope. Follow the instructions below for writing your self-reflection and submitting your assignment. If you run into any trouble, please ask for help!
- b. **Note: Today's practicum is not graded.** (You'll notice it is worth 0 points on Gradescope).

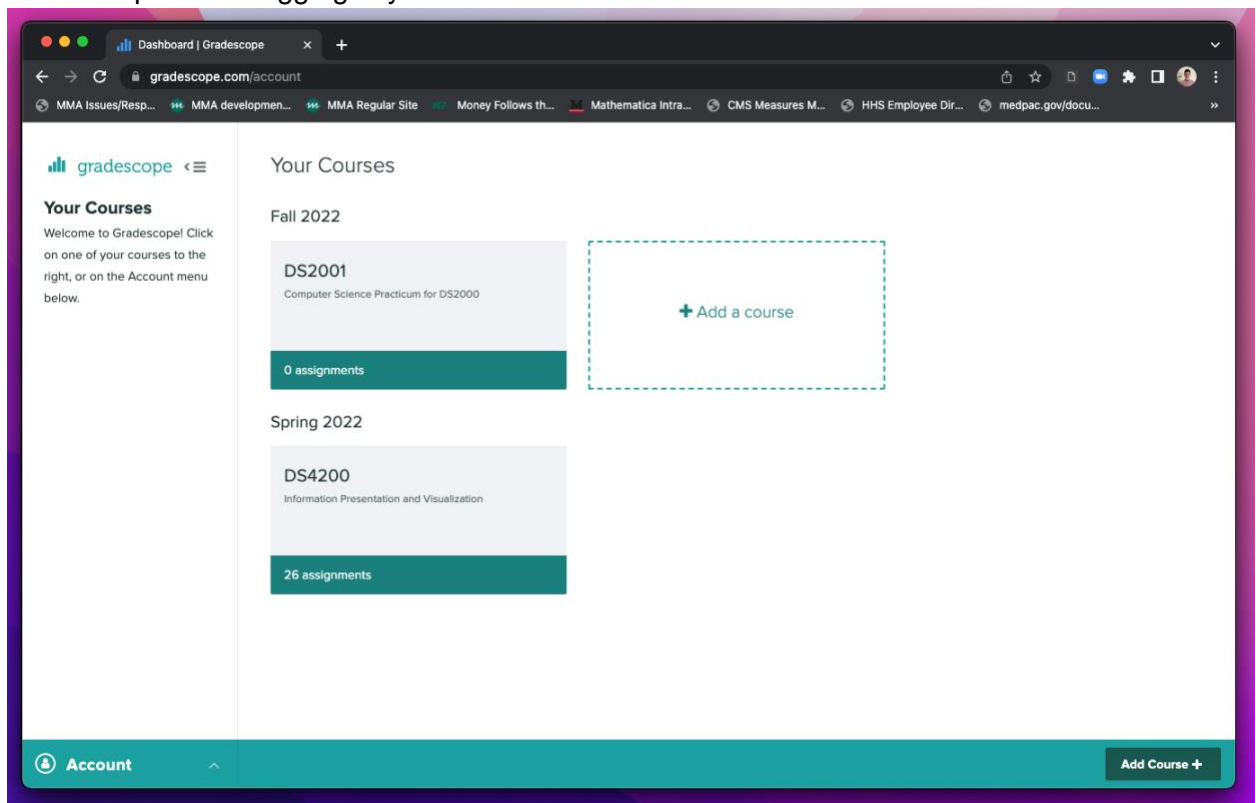
## Self-Reflection

Create a PDF with answers to the following questions:

- What was your group's answer to the first ice breaker question?
- What was your group's answer to the second ice breaker question?

## Submitting on Gradescope

1. Gather your code from today and your self-reflection PDF. Then, head over to Gradescope. After logging in you will see a dashboard like this:



- Click on DS2001 to go to the Gradescope page for this practicum. You will see a page that looks similar to this:

The screenshot shows the DS2001 Dashboard on Gradescope. The page title is "DS2001 Fall 2022". The course description is "Computer Science Practicum for DS2000". The dashboard includes a sidebar with links to "Dashboard", "Regrade Requests", "INSTRUCTORS" (Laney Strange, Ab Mosca), and "COURSE ACTIONS" (Leave Course). The main content area displays a table with columns: NAME, STATUS, RELEASED, and DUE (EDT). The table lists "Practicum-0" with a status of "No Submission", a release date of "AUG 12 AT 9:00AM", and a due date of "SEP 08 AT 11:00PM". A progress bar indicates "3 weeks, 6 days left".

NAME	STATUS	RELEASED	DUE (EDT)
Practicum-0	No Submission	AUG 12 AT 9:00AM	SEP 08 AT 11:00PM

- Click on the name of the assignment you want to submit. Then Drag & Drop your code and self-reflection PDF or find them through Browse Files.

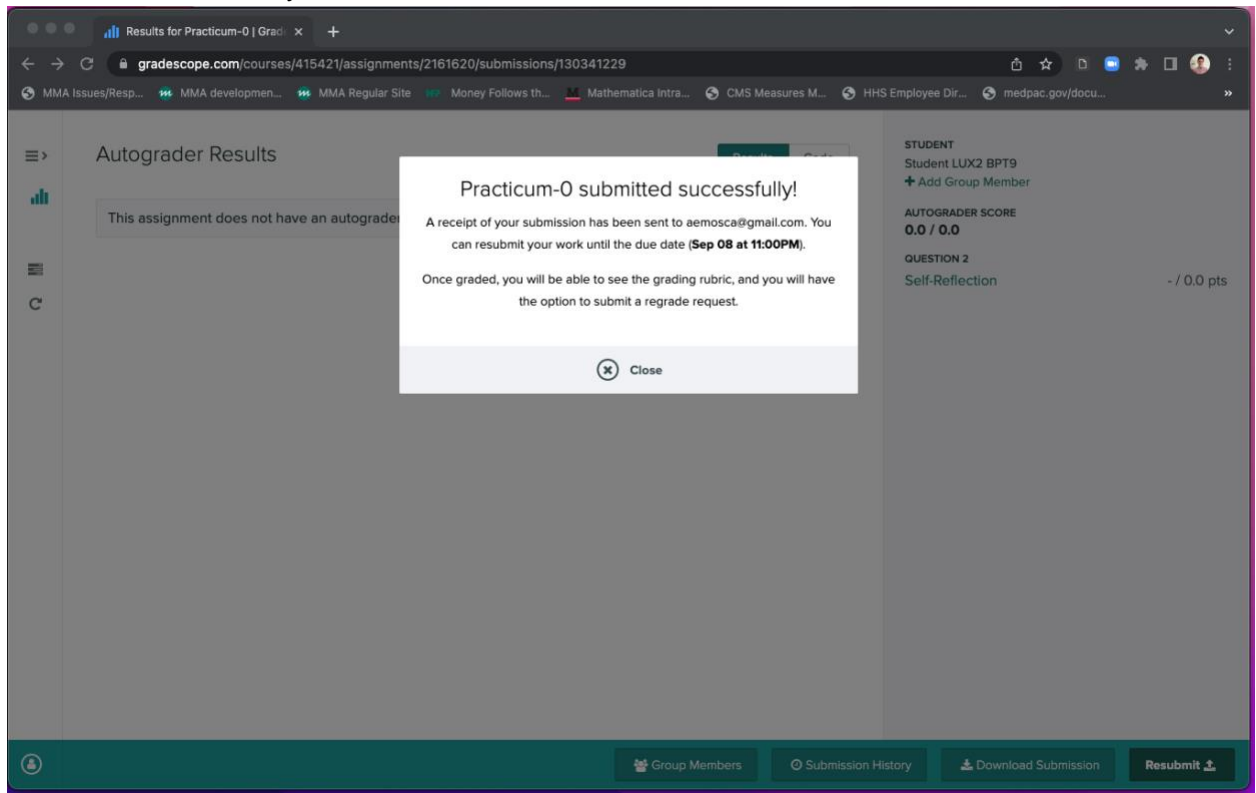
The screenshot shows the DS2001 Dashboard with a "Submit Programming Assignment" modal open. The modal has a title "Submit Programming Assignment" and a subtitle "Upload all files for your submission". It includes a "SUBMISSION METHOD" section with radio buttons for "Upload", "GitHub", and "Bitbucket". Below this, it says "Add files via Drag & Drop or Browse Files." and displays a table of files being uploaded:

NAME	SIZE	PROGRESS
p0.py	36 b	<div></div>
p0_self-reflection.pdf	21.9 KB	<div></div>

At the bottom of the modal are "Upload" and "Cancel" buttons. The background dashboard is dimmed, showing the same course details as the previous screenshot.



4. Click Upload. You should then see a modal window saying your assignment was submitted successfully:



5. You can view your submission by selecting “Code” after closing the successful submission modal window:

The screenshot shows the Gradescope web interface for a submission titled "Submitted Files for Practicum-0". The browser address bar shows the URL: `gradescope.com/courses/415421/assignments/2161620/submissions/130341229?view=files#Question_2`. The interface includes a sidebar on the left with navigation icons. The main content area displays two submitted files:

- p0.py**: A Python file with the following code:

```
1 #Practicum 0
2
3 print("hello world!")
```
- p0\_self-reflection.pdf**: A PDF file displayed as a Microsoft Word document. The content of the PDF is:

Self-Reflection Practicum 0

  - This class is going to be great!

Each file has a "Download" link. Above the file list, there are two buttons: "Results" and "Code". The "Code" button is highlighted with a red rectangle. On the right side of the interface, the student information is displayed:

- STUDENT**: Student LUX2 BPT9
- + Add Group Member**
- AUTOGRADER SCORE**: 0.0 / 0.0
- QUESTION 2**: Self-Reflection - / 0.0 pts

At the bottom of the interface, there is a teal bar with several buttons: "Group Members", "Submission History", "Download Submission", and "Resubmit".