

Practicum 1: Installing Software and Hello, World!

There are two goals for this week's practicum: (1) to install all the software you will need for the rest of the class, and, (2) to write, run, and test your first Python program!

1 Software Installation

Python is a language that comes in many flavors and "distributions." In this class, we will use the distribution Anaconda, which includes all of the most commonly used code libraries that we will need for this class. It also comes with some useful software, including the text editor that we will be using, Spyder, and tools for working with Jupyter notebooks, which we'll use later in the course.

Downloading Anaconda

1. Go to the Anaconda download page: <https://www.anaconda.com/download/>
2. The site should detect which installer you need automatically (MacOS, Windows, or Linux). Right at the start of the page, click "Download." If not, navigate to the installers at the bottom of the page and choose the appropriate one for your computer.
3. When you download Anaconda, a window may come up asking you to set up an account with Anaconda Nucleus. You can ignore this.
4. When the installer is finished downloading, open it. If your computer gives any warnings about installing it, keep proceeding through the install menus. Anaconda can be trusted, it's used by millions of data scientists across the world.

For the most part, simply click "Next" to complete the installation. At the end, you can skip the PyCharm screen.

Windows users: make sure to choose "Add Anaconda to my PATH environment variable." This will make it easier to use Python. Selecting this option may give you a red warning message. This is OK. You should still select this option.

Anaconda is a sizable piece of software, so it may take a little while to finish installing.

Opening Spyder

1. After Anaconda is done installing, you should have a new program called Anaconda-Navigator. Find this program (using your computer's program search if

necessary) and open it.

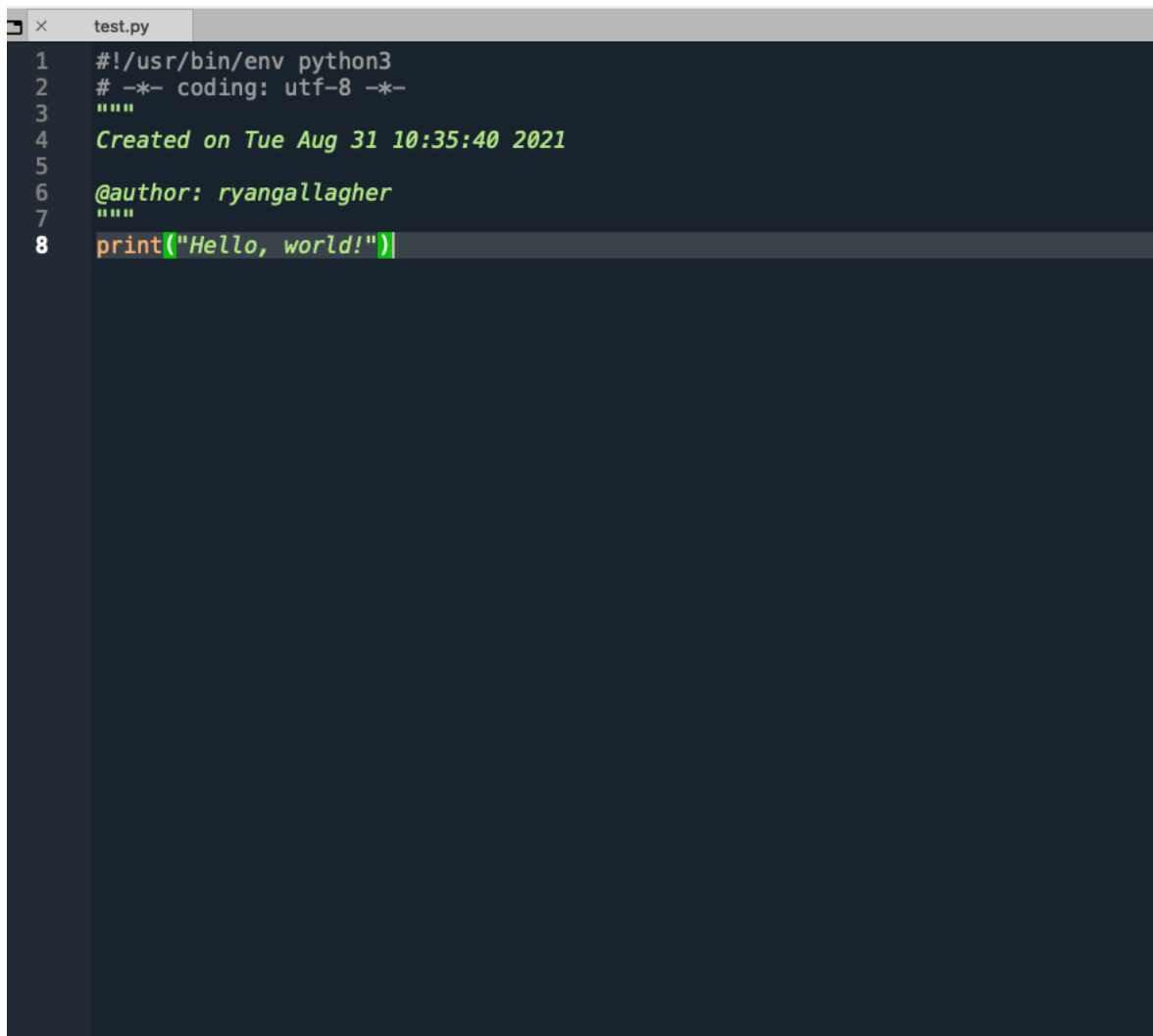
2. After opening Anaconda-Navigator, you will see a number of programs. Find the one that says “Spyder” and launch it.

2 Spyder and Hello, World!

Spyder is an Interactive Development Environment (IDE). IDEs allow us to write code and then execute, debug, and inspect it all in one place.

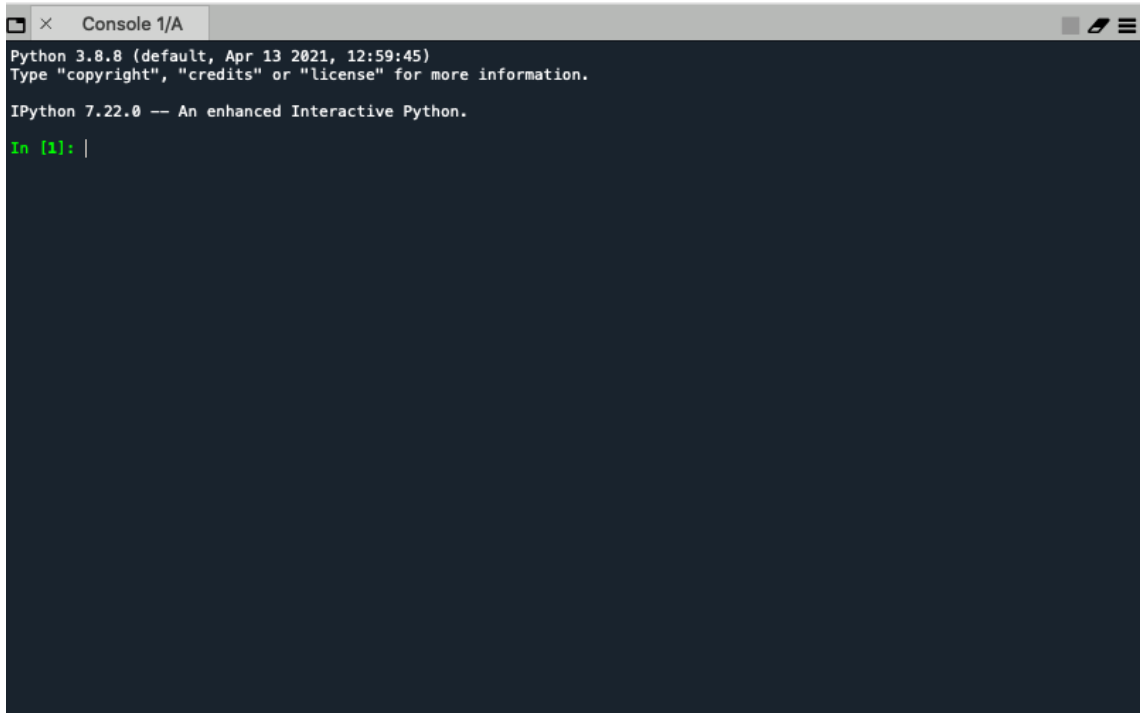
Spyder

- The *text editor* is the largest pane to the left. *This is where you will write all of your code.* We’ll come back to this in a moment.



```
test.py
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  """
4  Created on Tue Aug 31 10:35:40 2021
5
6  @author: ryangallagher
7  """
8  print("Hello, world!")
```

- The *console* is to the bottom right. The console is where commands are entered to run the Python code. In general though, it will be easier for us to use the buttons at the top of the Spyder interface to interact with our code, so you shouldn't have to enter much into the console directly. It also will print any output from our code, including errors.



```
Python 3.8.8 (default, Apr 13 2021, 12:59:45)
Type "copyright", "credits" or "license" for more information.

IPython 7.22.0 -- An enhanced Interactive Python.
>>>
In [1]: |
```

- The top right can be a couple of different things and you can see some options at the bottom of the pane: Variable explorer, Help, Plots, and Files. The variable explorer will keep track of what values we give to different variables as we run our code, so it's useful for debugging (it's OK if that doesn't make sense to you right now). The help tab displays documentation of functions and their arguments (again, it's OK if you don't know what this means, we'll cover functions later in the semester). The plots tab displays any visualizations we make of our data. The files tab gives us an easy way to select what file we want to use for writing and running code.

Hello, World!

At the top you'll see a set of buttons for interacting with the code. Most of these are for debugging code.



1. There should already be a file in the text editor for us to edit, but we're going to make ourselves a new one. You'll want to make a new file for each practicum assignment (and DS2000 homework). Click on the file icon, the button on the very left that looks like a piece of paper. This will create a new file in the text editor.
2. Go to `File > Save`. You want to save the file somewhere where you'll be able to find it later. ***I highly suggest making a folder for this class called something like "DS2001" and saving it there.*** If you are not sure how to navigate your computer's file system, please let me know and I'll be happy to help you save the file somewhere reliable. Save the file with a reasonable name like "practicum01.py". Python files use the .py extension.
3. You'll see that Spyder has already put some information at the top of our file. You can ignore the commands following the pound signs #. After those, you'll see two sets of triple quotes, with some information about when the file was created and who created it in between. This is called a *doc string*. I suggest keeping it there, you'll be required to have something like this on all of your submissions (you'll learn more about that in DS2000 soon).

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  """
4  Created on Mon Sep  6 16:02:54 2021
5
6  @author: ryangallagher
7  """
```

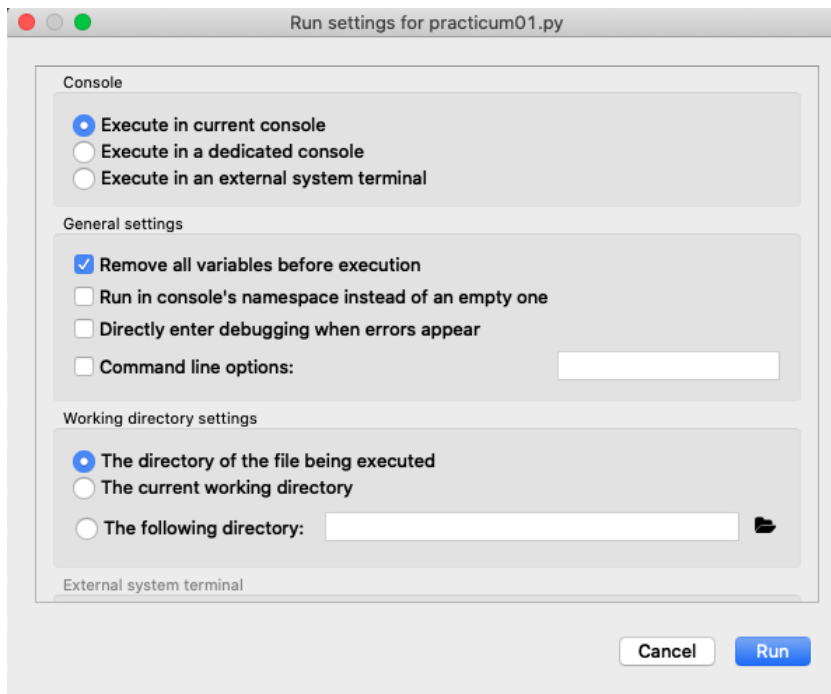
4. After those first few lines, you'll see a blank line. This is where you can start typing code! Your first piece of code will be one that many first time programmers write. On the open line type: `print("Hello, world!")`

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  """
4  Created on Mon Sep  6 16:02:54 2021
5
6  @author: ryangallagher
7  """
8  print("Hello, world!")
```

5. Now we want to run our code. Look at the top bar of buttons and find the Run button, which looks like a green triangle pointing to the right (like a play button). Press that

button to run your code.

6. A screen should come up that looks like this:



Make sure of the following:

- You should select "Execute in current console"
- You should select "Remove all variables before execution"
- You should select "The directory of the file being executed"
- You don't have to show this on the first run of every file, so you can uncheck that box

7. Press "Run" when you've selected all these options. Did you get some errors? *DON'T PANIC!* It's no big deal, and not even a sign that you did something wrong. There is no programmer, no matter how smart and experienced, who writes a whole program without errors. We all get them, we all deal with them. Read the errors and try to figure out what went wrong.
8. Congratulations, you've run your first piece of code! How can you tell? The output is shown in the console at the bottom right:

```
Console 1/A
Python 3.8.8 (default, Apr 13 2021, 12:59:45)
Type "copyright", "credits" or "license" for more information.

IPython 7.22.0 -- An enhanced Interactive Python.

In [1]: runfile('/Users/ryangallagher/teaching/DS2001/Practicum01/practicum01.py', wdir='/Users/ryangallagher/teaching/DS2001/Practicum01')
Hello, world!

In [2]:
```

See that little “Hello, world!”? That’s from our code! We told the computer to output or “print” the sentence “Hello, world!” and there it is.

9. Let’s get a taste of what you’ll be covering in DS2000 over the next week. Try typing the following commands (note the space after the comma in “Hi, “):

```
8 name = input("What's your name?\n")
9 print("Hi, ", name)
```

Run the code (press the green triangle Play button at the top) and look at the console. It’s asking for your name, so type it in and press Enter or Return on your keyboard.

10. You’ll learn more about what’s going on here in these lines of code over the next few classes of DS2000. Here are some questions to ask yourself for now.

- How does the code know your name after you put it in? Where is it getting saved?
- What is that “\n” at the end of the input statement? What happens if you remove it and run the code?
- Look closely at my print statement and notice that I have a space after my comma in “Hi, “. Now look at the output, it should look something like “Hi, Ryan” with two spaces between the comma and “Ryan.” What does that tell you about how `print` works when we’re printing more than one thing? Can you fix it so that there’s only one space between the comma and your name?

Optional Challenges

11. (Optional) Challenge: Make your code ask you for your name and your school year (e.g. sophomore, senior, etc.) and print it out all on one line like this:

“Ryan is a 10th year college student :(”

12. *(Optional)* One more thing to think about: what if in the challenge I had asked you to print “Your name is Ryan. You’re a 10th year college student”? Can you see where you’d run into trouble based on what you’ve done so far?

3 Submission

Lastly, you should submit your work to Canvas.

1. First make a zip file of your work (in this case, you only wrote one program, but you have more in later practicums). Go to where you save your files (this is why you need to know where you’re saving things) and right-click:
 - On MacOS, choose: `Compress practicum01.py`
 - On Windows, choose: `Send To > Compressed (zipped) folder`
2. Go to Canvas and submit the zip folder to Practicum 1 under Assignments.

Submit this practicum to Canvas by Friday September 10th at 11:59pm.

Note: sometimes you may make mistakes uploading files and send the wrong thing to Canvas. When I download the code then, it may not run like you thought it would. After each submission, I recommend you download the submitted work, unzip it, inspect it, and run the code *from that file* to simulate what I’ll do when I’m grading.