SSEP 2024: Intro to Programming with Python

Summer 2024

Assignment 01: Welcome and Set Up

*This is a **pair programming assignment find another person to work with on the assignment.***

Assignment Goals:

- Install Anaconda
- Write a Python program with Spyder

Notes

It is your responsibility to write a program that is well structured (i.e. modular), and easy to read. Your file should start with a header that has your name(s), the date, and a brief description of what your program does. Variable names should be descriptive. Comments should be used appropriately to document your code.

Quick Programming Tips:

- ***Not sure what to do?*** Talk through the assignment instructions with your partner and write out in English what specific tasks your program needs to do. Then, pseudocode.
- ***Stuck on how to program your solution?*** Try out 2 different ideas then ask for help if you are still stuck!
- ***Have a seemingly invisible bug?*** Use print statements throughout your code. Before running the code, think about what you expect to print if the code is working. Then see if what you expected is what prints.

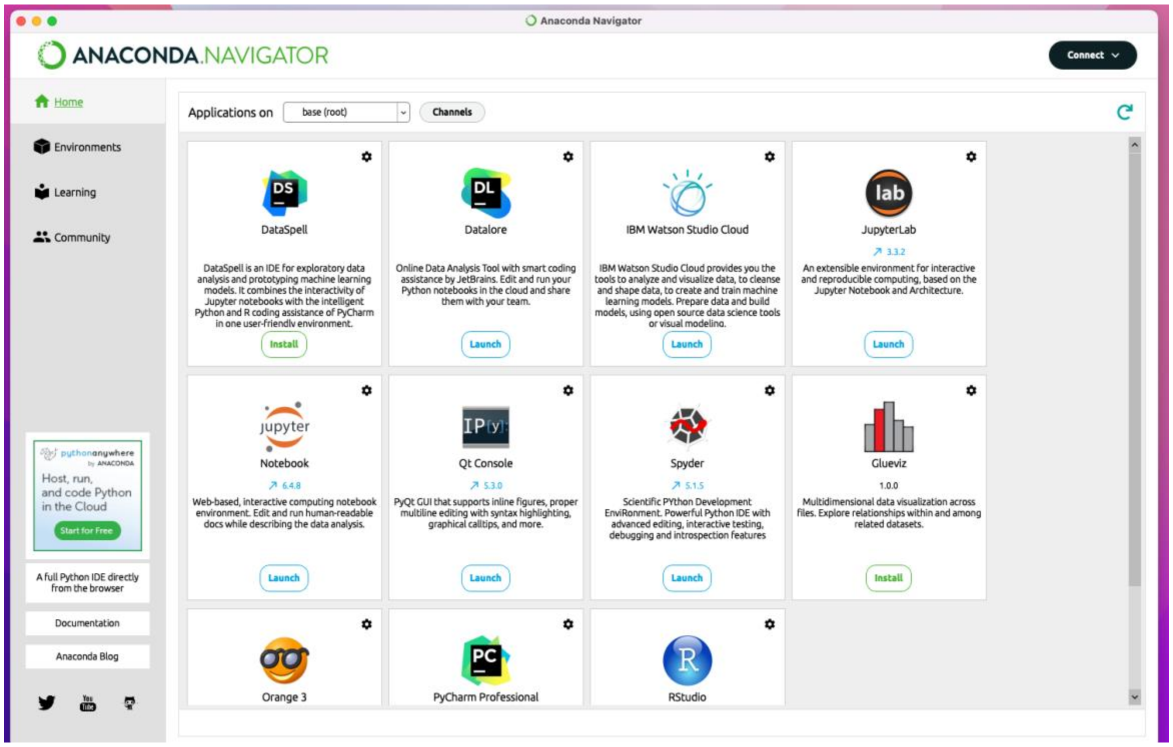
TODO

1. **Install Anaconda**
    - Anaconda is a distribution of the Python (and R) programming language that will make your life simpler as you learn Python.
    - Navigate to https://www.anaconda.com/ and follow the link to download Anaconda.

- Once your Anaconda .pkg is downloaded, open it up and your computer's installer should launch automatically and walk you through the install process.
- If you run into any trouble ask Ab, or Hodan for help (we want to help!)

2. Explore Anaconda

- Find "Anaconda Navigator" among your installed apps and open it.
- If you get a message that "Anaconda Navigator needs to be updated." Follow the prompts to update before moving on.
- Your Anaconda Navigator will look something like this:



DataSpell is an IDE for exploratory data analysis and prototyping machine learning models. It combines the interactivity of Jupyter notebooks with the intelligent Python and R coding assistance of PyCharm in one user-friendly environment.

Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

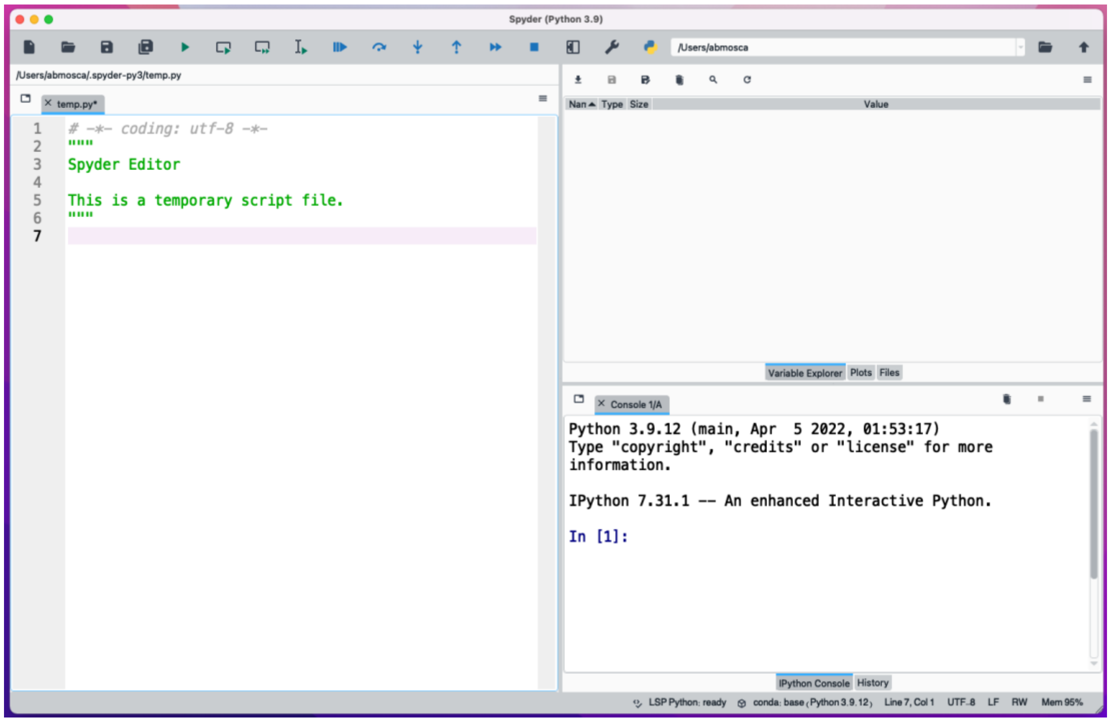
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Multidimensional data visualization across files. Explore relationships within and among related datasets.

- 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.
- Launch Spyder by clicking the "launch" button (second row, third to the right in the screenshot above).
- If you get a message that Spyder needs to be updated ho ahead and update as instructed before moving on. (Spyder's instructions can be confusing and annoying – we can help you figure them out!)

- Spyder will open, and looks something like this:

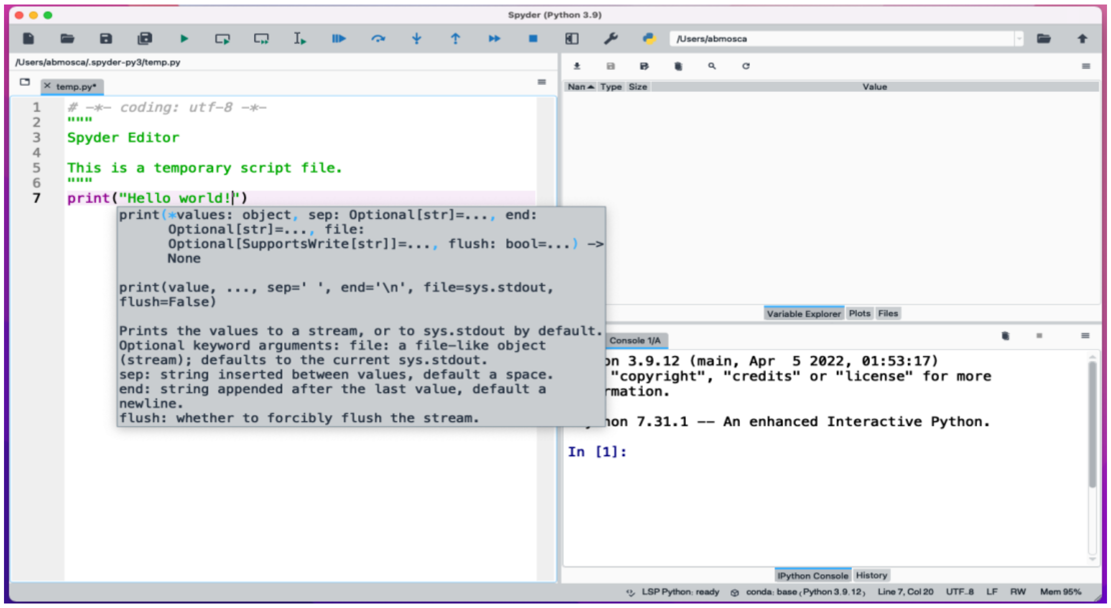

Spyder (Python 3.9)
/Users/abmosca
/Users/abmosca/.spyder-py3/temp.py
temp.py*
Nan Type Size Value
Variable Explorer Plots Files
Console 1/A
IPython Console History
LSP Python: ready conda: base (Python 3.9.12) Line 7, Col 1 UTF-8 LF RW Mem 95%

```
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""

```

```
Python 3.9.12 (main, Apr  5 2022, 01:53:17)
Type "copyright", "credits" or "license" for more
information.

IPython 7.31.1 -- An enhanced Interactive Python.

In [1]:
```

3. Write a Python Program

- 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.

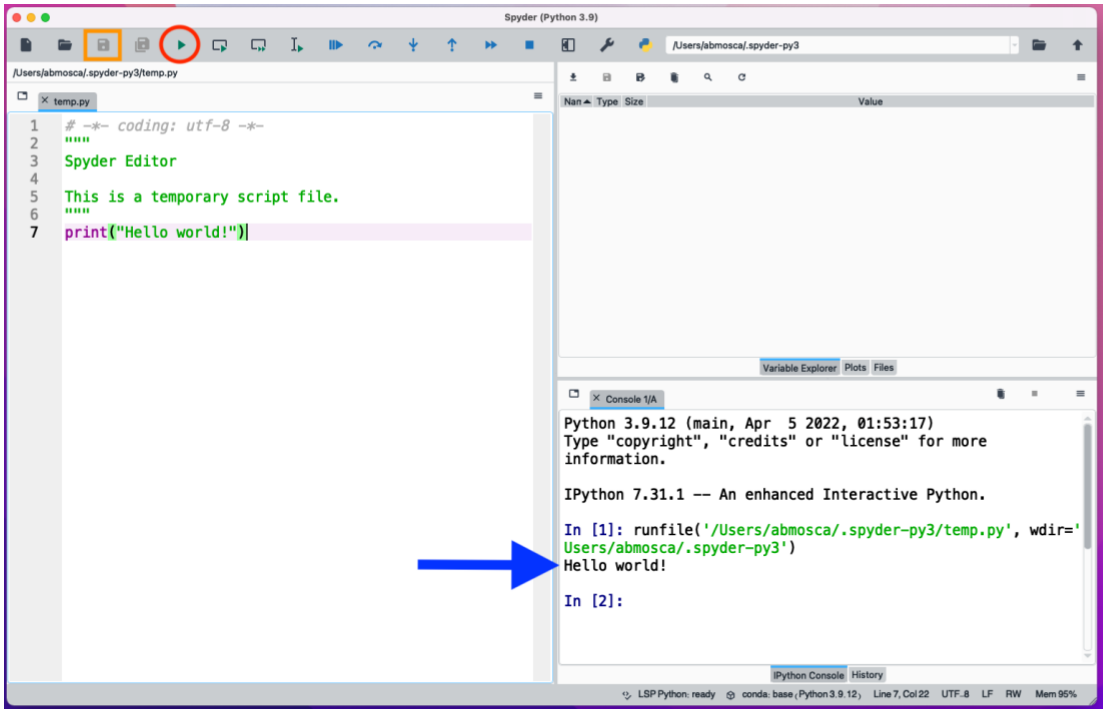

Spyder (Python 3.9)
/Users/abmosca
/Users/abmosca/.spyder-py3/temp.py
temp.py*
Nan Type Size Value
Variable Explorer Plots Files
Console 1/A
IPython Console History
LSP Python: ready conda: base (Python 3.9.12) Line 7, Col 20 UTF-8 LF RW Mem 95%

```
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""
print("Hello world!")
```

```
print(*values: object, sep: Optional[str]=..., end:
      Optional[str]=..., file:
      Optional[SupportsWrite[str]]=..., flush: bool=...) ->
      None

print(value, ..., sep=' ', end='\n', file=sys.stdout,
flush=False)

Prints the values to a stream, or to sys.stdout by default.
Optional keyword arguments: file: a file-like object
(stream); defaults to the current sys.stdout.
sep: string inserted between values, default a space.
end: string appended after the last value, default a
newline.
flush: whether to forcibly flush the stream.
```

```
on 3.9.12 (main, Apr  5 2022, 01:53:17)
 "copyright", "credits" or "license" for more
rmation.

hon 7.31.1 -- An enhanced Interactive Python.

In [1]:
```

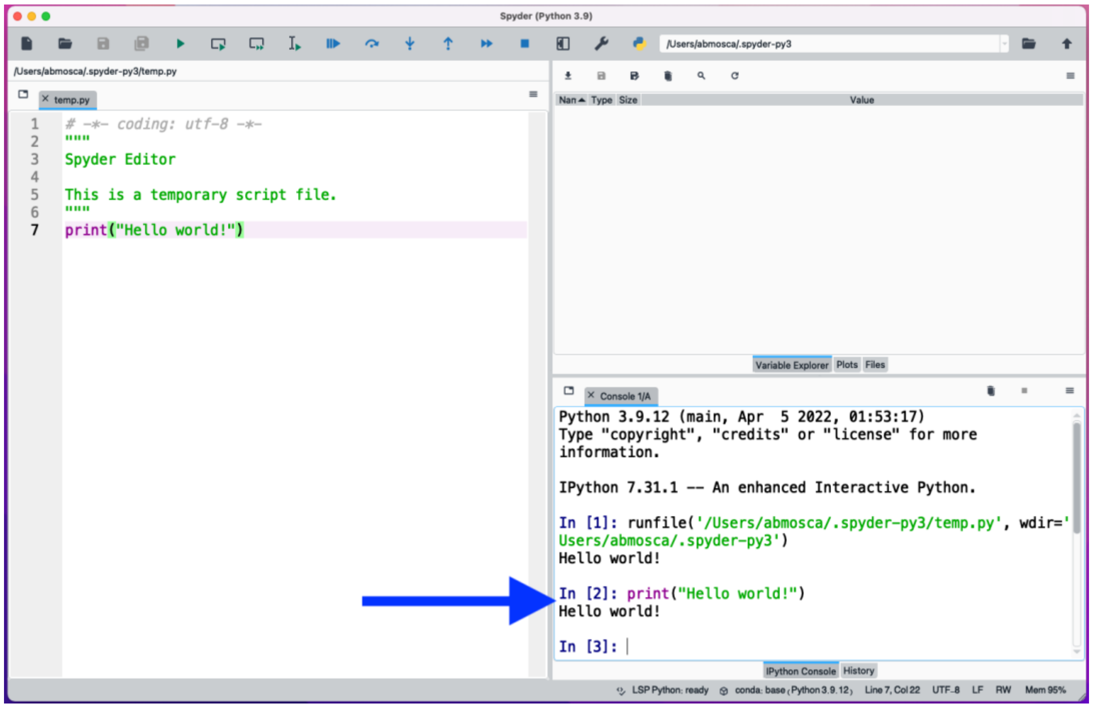
- To save your file, go to File > Save As. In the save dialogue box you can rename the file if you want, and choose where you want it saved. Notice that the default folder is not the most intuitive place to save the file if you want to find it again later. We recommend creating a folder on Desktop or Documents specifically for this class. After 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).
- 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 and hit "Run". Notice what happens in the bottom right of the window after you hit Run.



```
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""
print("Hello world!")
```

```
Python 3.9.12 (main, Apr  5 2022, 01:53:17)
Type "copyright", "credits" or "license" for more
information.

IPython 7.31.1 -- An enhanced Interactive Python.

In [1]: runfile('/Users/abmosca/.spyder-py3/temp.py', wdir='
Users/abmosca/.spyder-py3')
Hello world!

In [2]:
```

- Cool! The code `print("Hello world!")` prints the phrase "Hello world!" in the bottom right window! 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.


Spyder (Python 3.9)
/Users/abmosca/.spyder-py3
/Users/abmosca/.spyder-py3/temp.py
temp.py
Nan▲ Type Size
Value
Variable Explorer Plots Files
Console 1/A
IPython Console History
LSP Python: ready conda: base (Python 3.9.12) Line 7, Col 22 UTF-8 LF RW Mem 95%

```
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""
print("Hello world!")
```

```
Python 3.9.12 (main, Apr  5 2022, 01:53:17)
Type "copyright", "credits" or "license" for more
information.

IPython 7.31.1 -- An enhanced Interactive Python.

In [1]: runfile('/Users/abmosca/.spyder-py3/temp.py', wdir='
Users/abmosca/.spyder-py3')
Hello world!

In [2]: print("Hello world!")
Hello world!

In [3]: |
```

- 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. 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 your assignment!) you should get in the habit of creating `.py` files in the file editor and running them to test your code.
- 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.

4. All Done?

- Find someone who isn't finished and help them out!

Submission

Be prepared to demonstrate and discuss your code with the class. After giving every one time to work we will come together and discuss:

1. Solutions you came up with
2. Roadblocks you hit along the way
3. What you would do differently if you were to do the assignment again

Curious to know how your code would be graded in a college class? Here's an example rubric:

	Missing / Not Complete (0)	Approaching (2)	Meets (4)	Exceeds (5)
Readability	Assignment is unreadable or not submitted.	Assignment includes formatting, but significant improvements could be made. For example, headers, more documentation (comments), descriptive variable names.	Assignment includes formatting, but minor improvements could be made. For example, headers, more documentation (comments), descriptive variable names.	Assignment is well formatted and easy to read. Headers, documentation (comments), and descriptive variable names are all included.
Computational Problem Solving	No code is included in the assignment, or the code included is unreadable.	Problem solving approach could use significant improvements. Specifically, better decomposition of the problem, and/or increased modularity.	Problem solving approach is solid but minor improvements could be made with respect to decomposition of the problem, and/or increased modularity.	Problem solving approach is solid. Problem is decomposed into manageable pieces and code is modular.
Implementation	Nothing has been implemented, or most of assignment has not been done.	Code does not run consistently or efficiently. Some outputs match expected outputs. All parts of the assignment are completed except for a few small parts.	Code mostly runs consistently and efficiently. Most outputs match expected outputs, and all parts of the assignment are completed.	Code runs consistently and efficiently. Outputs match expected outputs, and all parts of the assignment are completed.