Free Resources and Curricula for Teaching Scientific Computing:

DSECOP, PICUP, PY4E

Deva O'Neil, Bridgewater, VA
doneil@bridgewater.edu

April 2023

Scientific Computing in a Physics Curriculum

- Part of a regular physics course
- Stand-alone physics course (Computational Physics)
- Separate course requirement (Introductory Programming)
- Research projects

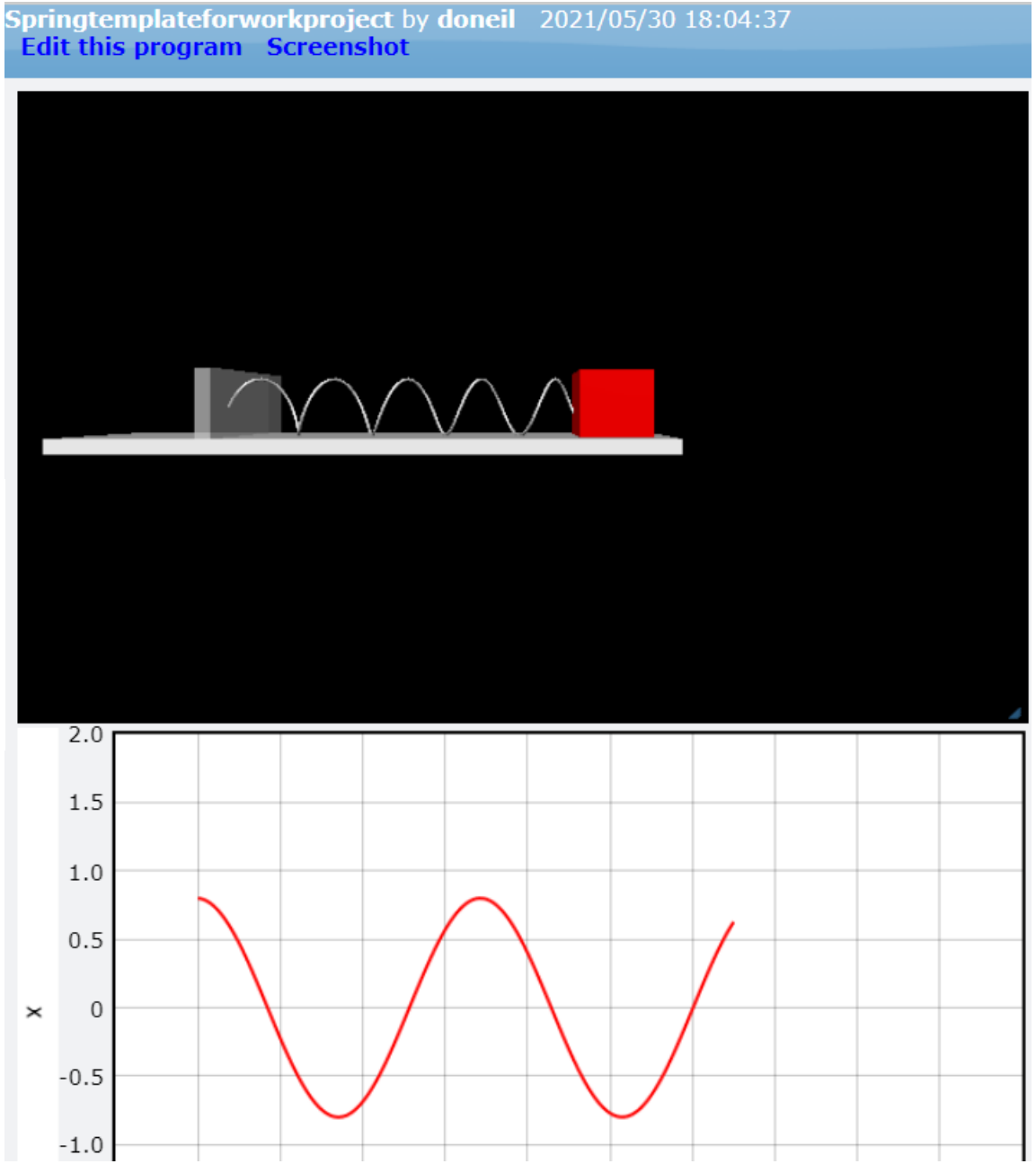
What does scientific computing with free software look like?

- Glowscript (now Web VPython)
- Trinket
- Jupyter Notebooks

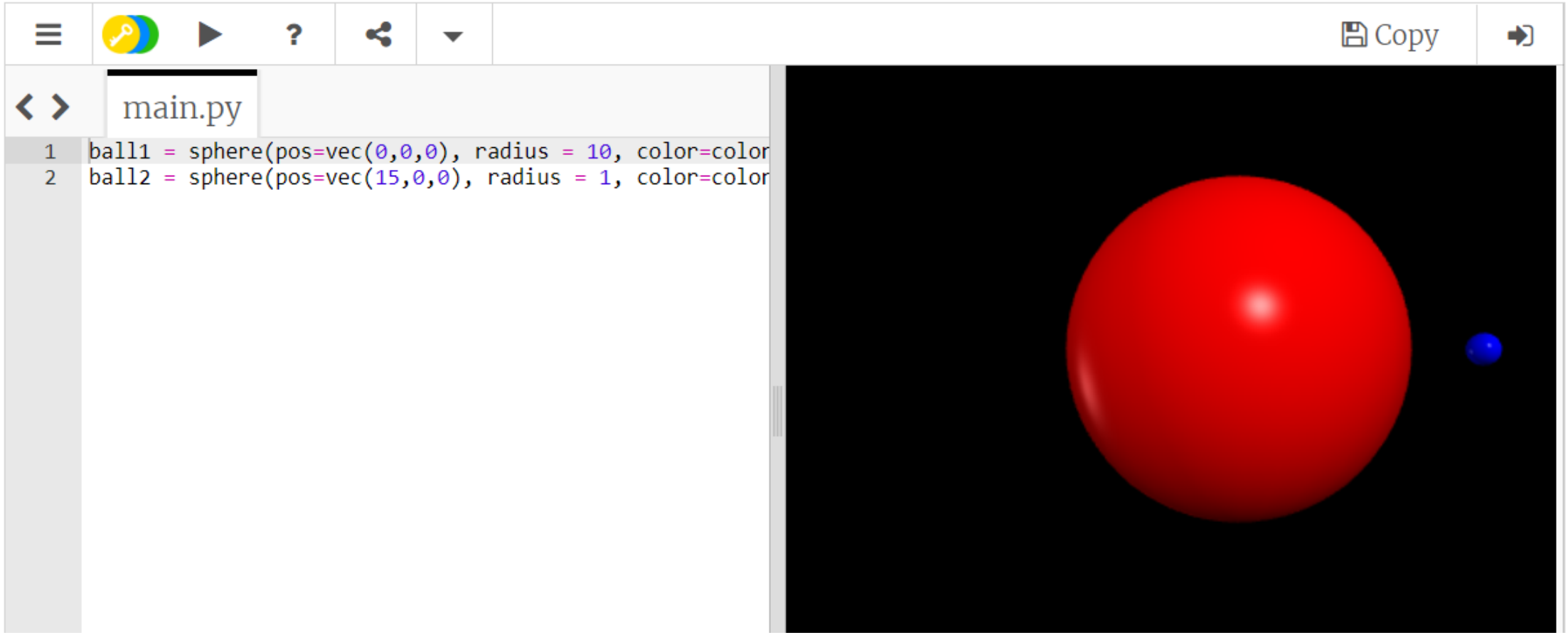
  (Python or Octave kernels)
- Octave (MATLAB workalike)

Glowscript
(now Web VPython)

```
Fspring = - k*X
Fnet = Fspring #the vertical forces cancel
pBlock = pBlock + Fnet*delta_t
Block.pos = Block.pos + (pBlock/mass) * delta_t
Spring.axis = Block.pos - Wall.pos
```



Trinket (runs in browser)

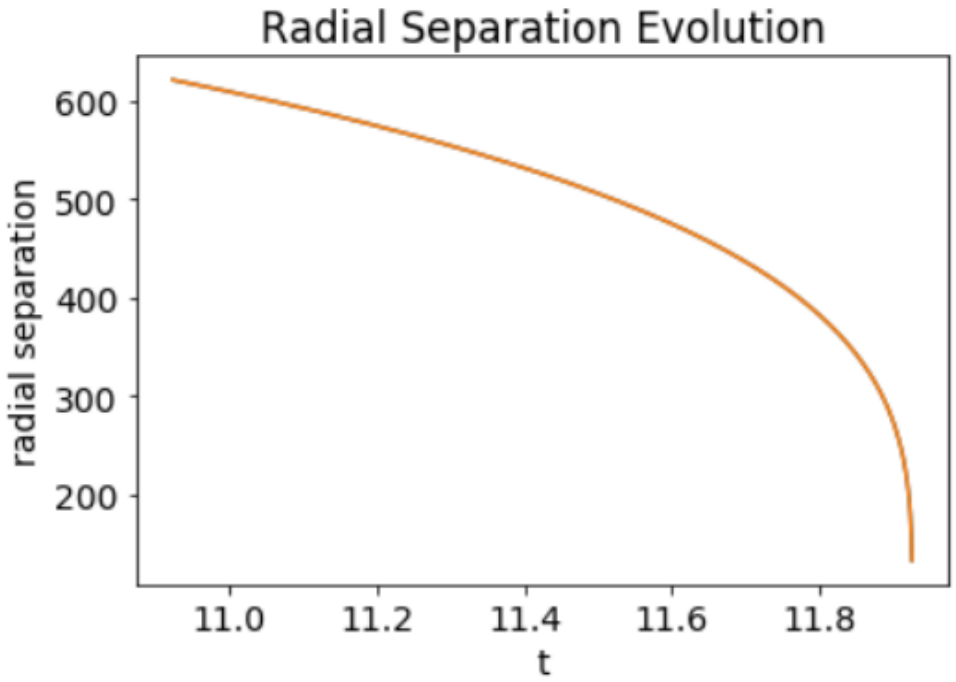


Jupyter Notebooks

Use in Google colab or have students install an IDE (eg, VSCode)

```
plt.xlabel("t")
plt.plot(tscaled, rs)
plt.title('Radial Separation Evolution')
```

Out[17]: Text(0.5, 1.0, 'Radial Separation Evolution')



Calculation of Inspiral Waveform

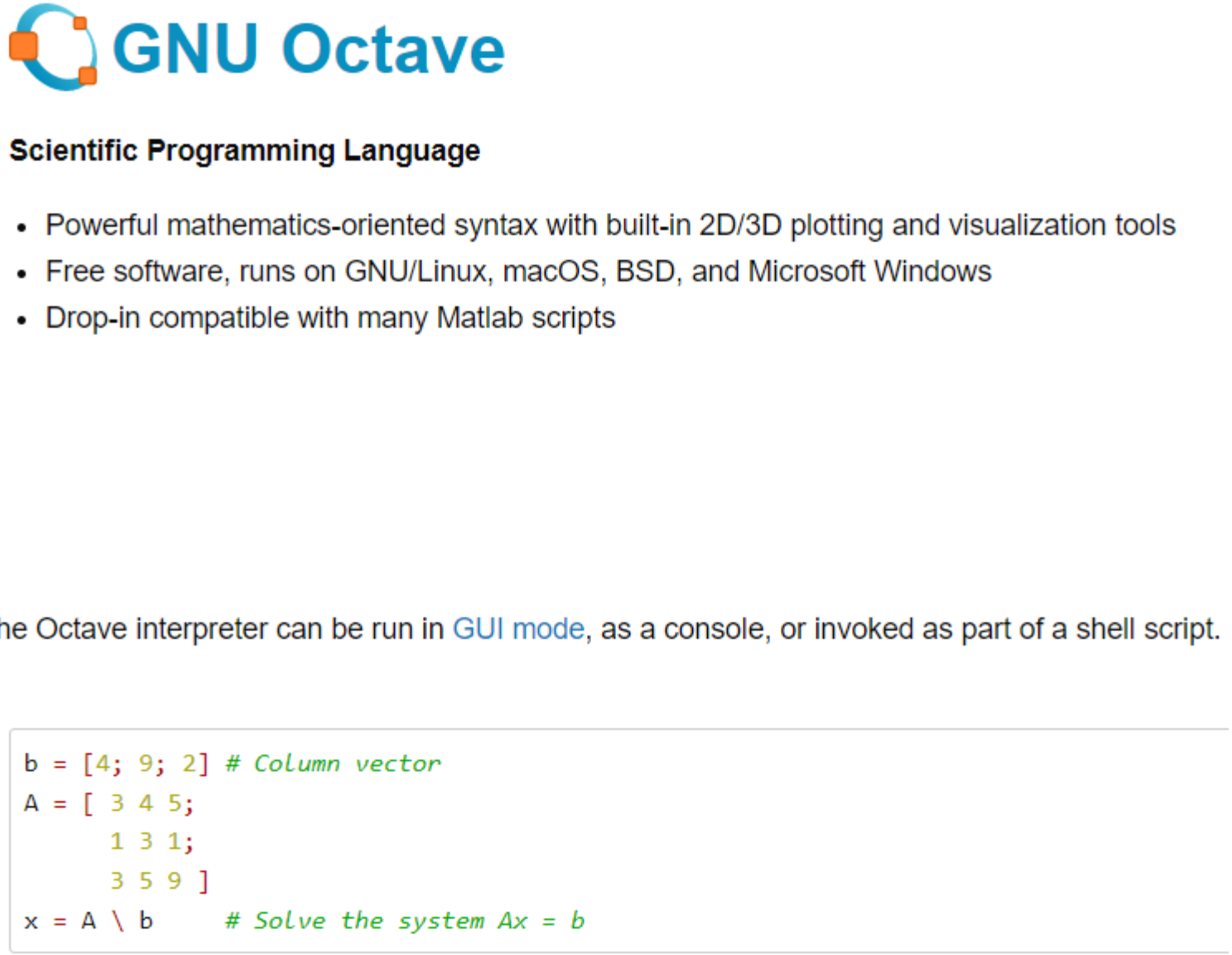
Just like electromagnetic radiation, gravitational waves have two different polarizations. Ho
radiation's polarizations differ by $\pi/2$ while gravitational waves have polarizations the diffe
cross-polarized strains given by the general formula (Huerta et al., 4' Gopakumar and Iyer, 4

$$h_+ = -\frac{M\eta}{r}\left[(\cos^2\theta + 1)\left[\left(-\dot{r}^2 + r^2\dot{\Phi}^2 + \frac{M}{r}\right)\cos 2\Phi\right.\right.$$

Octave
(like MATLAB,
but free)

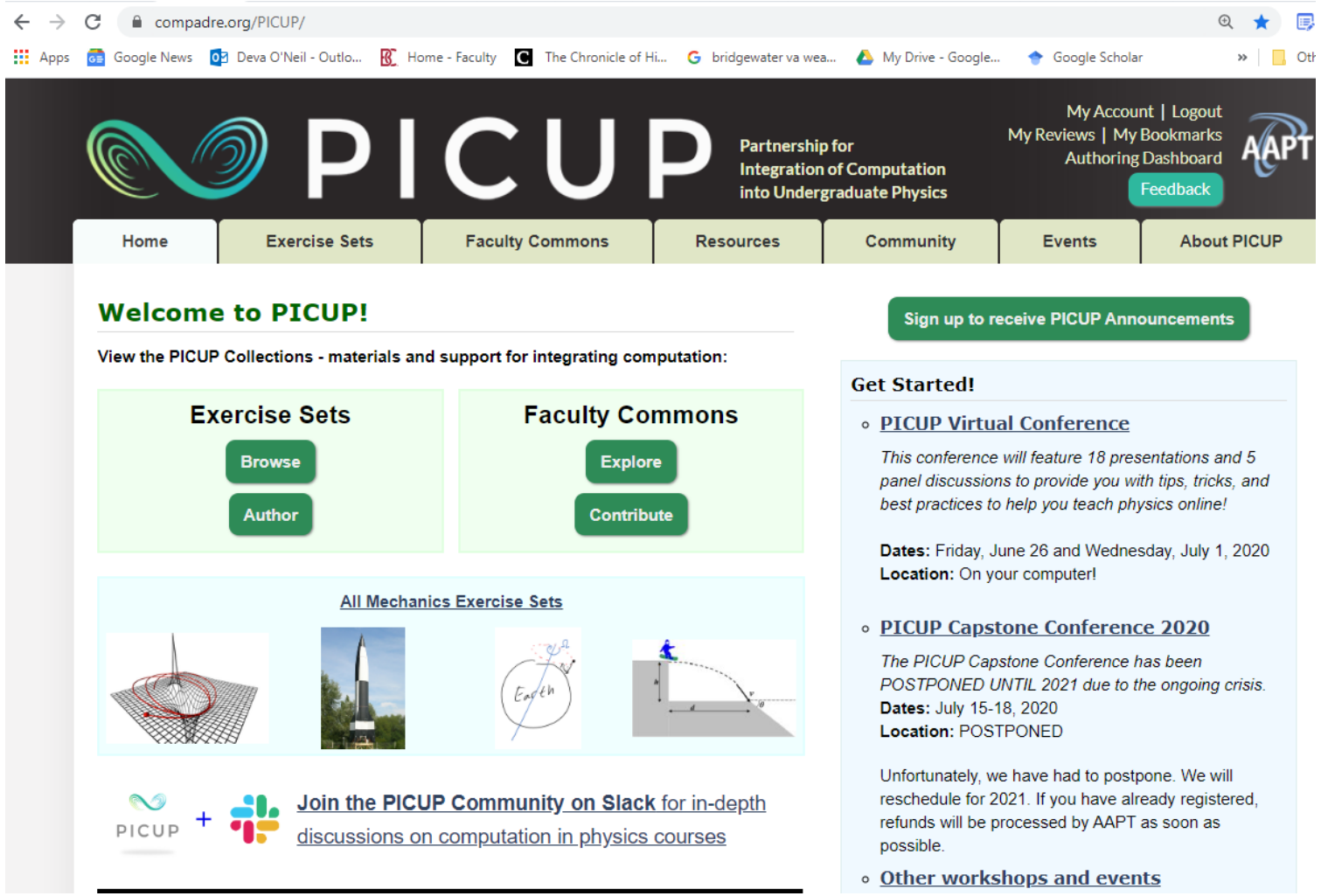
Use in browser
(Octave Online)
or have students
install GNU
Octave

Image from:
https://octave.org/



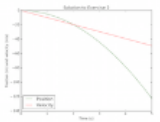
Curriculum Collections

PICUP (gopicup.org)
Good for standard physics courses



Curriculum Collections

PICUP (gopicup.org)
Good for standard physics courses

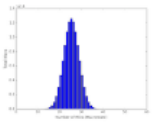


Using Python in Introductory Physics

First Year Programming Introductions

Developed by Eric Ayars

Specialized Programming Language: Python

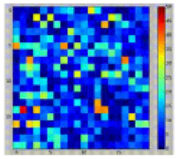


2-State Systems: Statistical Mechanics in Sports and the Story of the Purple Pandas

Beyond the First Year and Advanced Thermal & Statistical Physics

Developed by Brandon Lunk

Example Implementations: Python and Spreadsheet

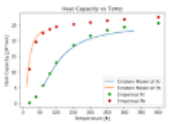


Boltzmann distribution, the statistical way

First Year and Beyond the First Year Thermal & Statistical Physics

Developed by Jay Wang

Example Implementations: Python and IPython/Jupyter Notebook



Einstein Solids: Equilibrium, Temperature and Heat Capacity

Beyond the First Year Thermal & Statistical Physics

Developed by Brandon Lunk

Example Implementations: IPython/Jupyter Notebook and Spreadsheet

Projectile Motion: Experiment and Computational Model

Developed by Todd Zimmerman - Published August 2, 2017

DOI: 10.1119/PICUP.Exercise.proj_mot

Connecting computation to experiment is at the heart of physics. This set of exercises requires students to create a computational model for a ball launched into the air to determine the distance traveled and total time in the air. The students then must perform the experiment by launching the ball with a spring launcher. Data from the experiment must be entered into the computational model and the experimental results are compared to the computer model.

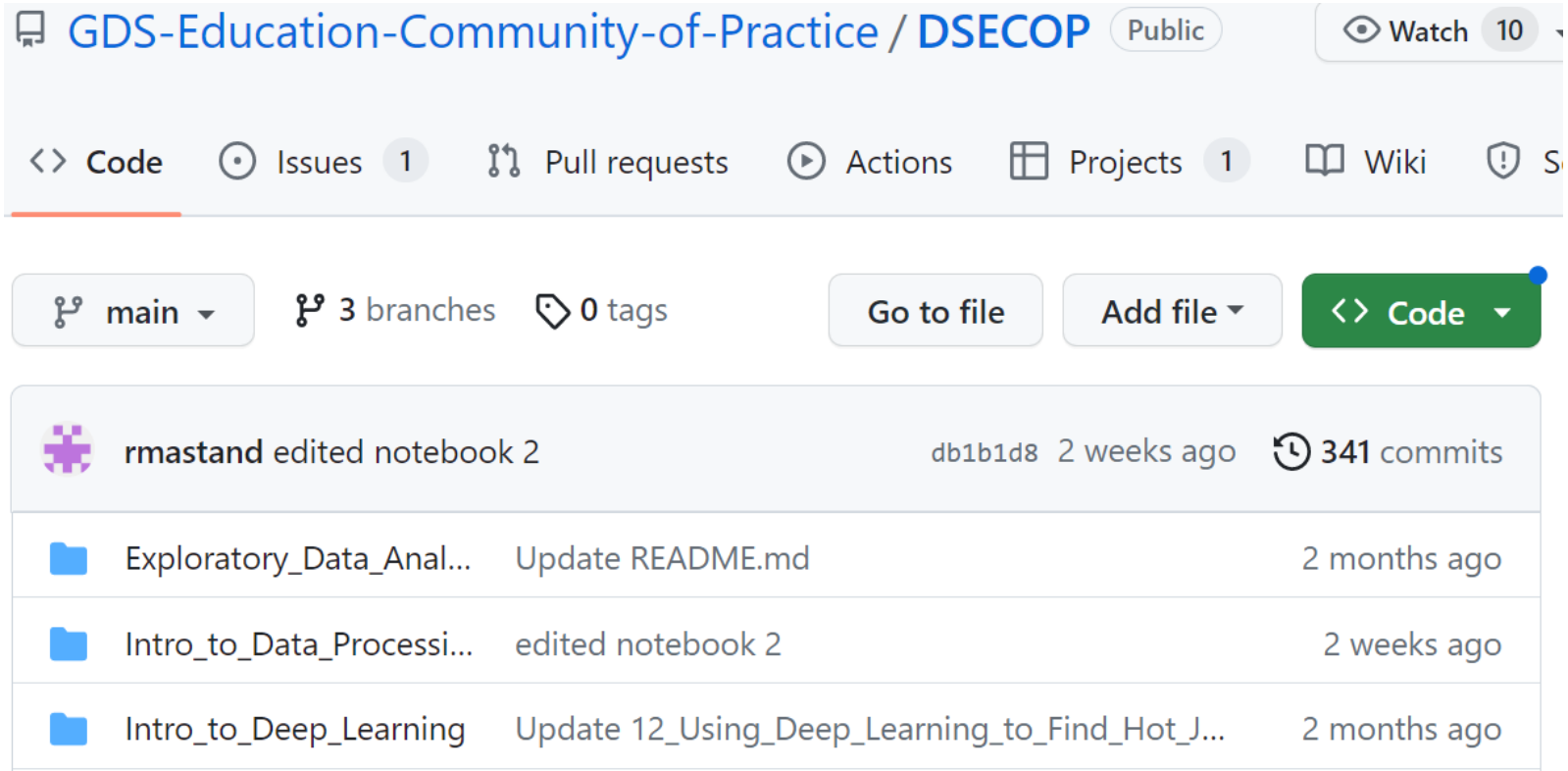
Subject Areas	Mechanics and Experimental / Labs
Level	First Year
Available Implementations	Glowscript, IPython/Jupyter Notebook, Sage Worksheet, and Easy Java Simulations
Learning Objectives	• Explain that the time an object is in the air depends only on motion in the y-direction (**Exercise 4**) • Relate the initial velocity of a launched ball to the horizontal velocity when the ball is launched horizontally (**Exercise 2**)

Curriculum Collections

Data Science Education Community of Practice (DSECOP)

Labs and special projects (Jupyter Notebooks only)

https://dsecop.org/.



Curriculum Collections

DSECOP

Table of Contents

Curriculum Collections

PY4E - free curriculum for python programming

You can embed videos and autograded homework into LMS

If you don’t know python, you can use this to teach yourself.

Curriculum Collections

PY4E - free curriculum for python programming

You can embed videos and autograded homework into LMS

HW for Ch 4

In the past week, we learned built-in functions (Wednesday of week 2) and how to define your own functions (Friday of week 2). The following 2 videos corresponds to those lectures:

https://www.youtube.com/watch?v=5Kzw-0-DQAk

https://www.youtube.com/watch?v=AJVNYRqn8kM

Autograded Homework

4.6 Write a program to prompt the user for hours and rate per hour using input to compute gross pay. Pay should be the normal rate for hours up to 40 and time-and-a-half for the hourly rate for all hours worked above 40 hours. Put the logic to do the computation of pay in a function called **computepay()** and use the function to do the computation. The function should return a value. Use 45 hours and a rate of 10.50 per hour to test the program (the pay should be 498.75). You should use **input** to read a string and **float()** to convert the string to a number. Do not worry about error checking the user input unless you want to - you can assume the user types numbers properly. Do not name your variable sum or use the sum() function.

Check Code **Reset Code**

```
1 def computepay(h, r):
2     return 42.37
3
4 hrs = input("Enter Hours:")
5 p = computepay(10, 20)
6 print("Pay", p)
```

Your Output

Desired Output

```
Pay 498.75
```

Interactive Textbook
has Trinkets embedded

Python for Everybody

by Charles Severance

Python for Everybody is an introduction to the basics of coding in Python 3 with an emphasis on practical usag
intended as a foundation for students who are looking to apply Python within other academic subjects as well
preparation for the serious study of computer science.

Summary: Curriculum Collections

- PICUP – everything physics
- DSECOP – data science
- PY4E - python

Common environments for python programming (free)

Anaconda – ecosystem with multiple environments (eg, Pycharm, Jupyter Notebooks)

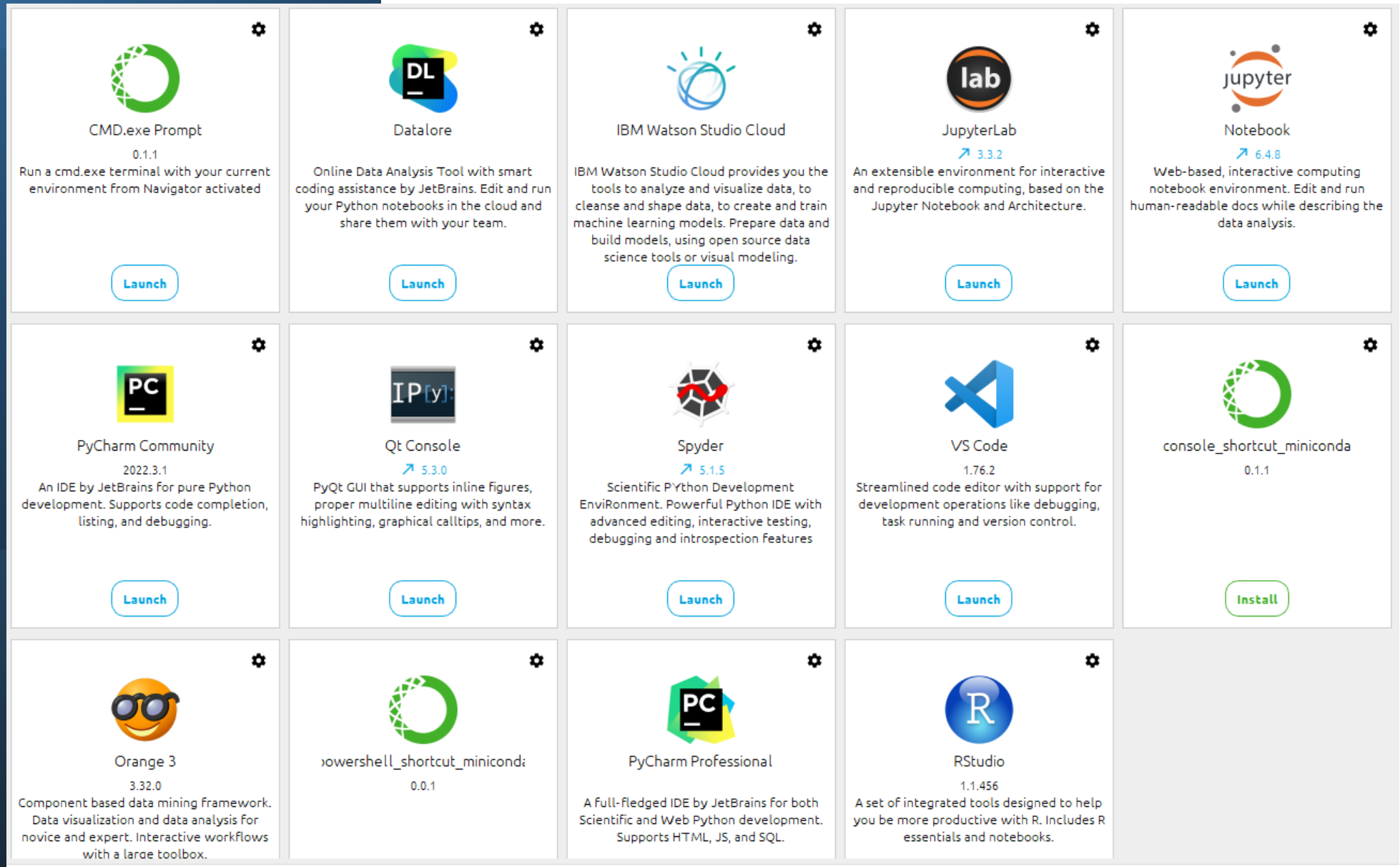
Stand-alone installations:

PyCharm – free version does not support Jupyter notebooks

VSCode – free version supports Jupyter

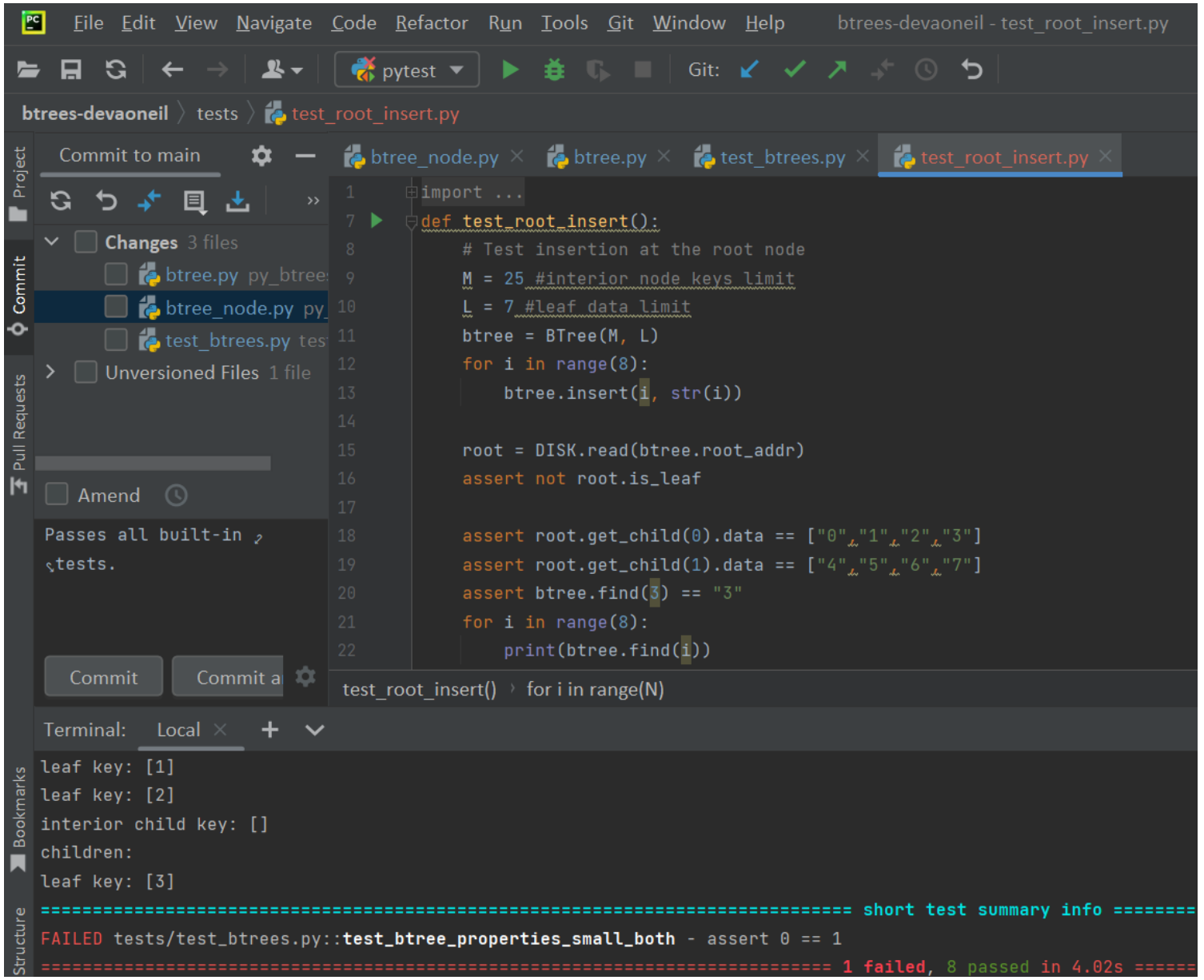
IDLE – built-in to python upon install

Anaconda


CMD.exe Prompt
0.1.1
Run a cmd.exe terminal with your current environment from Navigator activated
Launch
Datalore
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.
Launch
IBM Watson Studio Cloud
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.
Launch
JupyterLab
3.3.2
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
Launch
Notebook
6.4.8
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Launch
PyCharm Community
2022.3.1
An IDE by JetBrains for pure Python development. Supports code completion, listing, and debugging.
Launch
Qt Console
5.3.0
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Launch
Spyder
5.1.5
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Launch
VS Code
1.76.2
Streamlined code editor with support for development operations like debugging, task running and version control.
Launch
console_shortcut_miniconda
0.1.1
Install
Orange 3
3.32.0
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.
owershell_shortcut_minicond
0.0.1
PyCharm Professional
A full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.
RStudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Common environments for python programming (free)

PyCharm (makes version control easy)



IDLE

oodles_web_scraping_template.py - C:/Users/doneil/Box/doneil-migrated/Classes/PYTHON/scripts/oodles_web_scraping_template.py (3

File Edit Format Run Options Window Help

```
import re
import urllib.request
#visit the webpage:
hand = urllib.request.urlopen('https://www.gutenberg.org/ca

oodles = []
for line in hand:
    line = line.decode().strip()
    oodle_result = re.findall( #YOUR REGEX CODE HERE, line)
    if oodle_result  != []: #remember that you get an empty
     #Add the new list (oodle_result) to the main list (ooc

print("Found from the web: ")
print(oodles)
```

A few other free resources

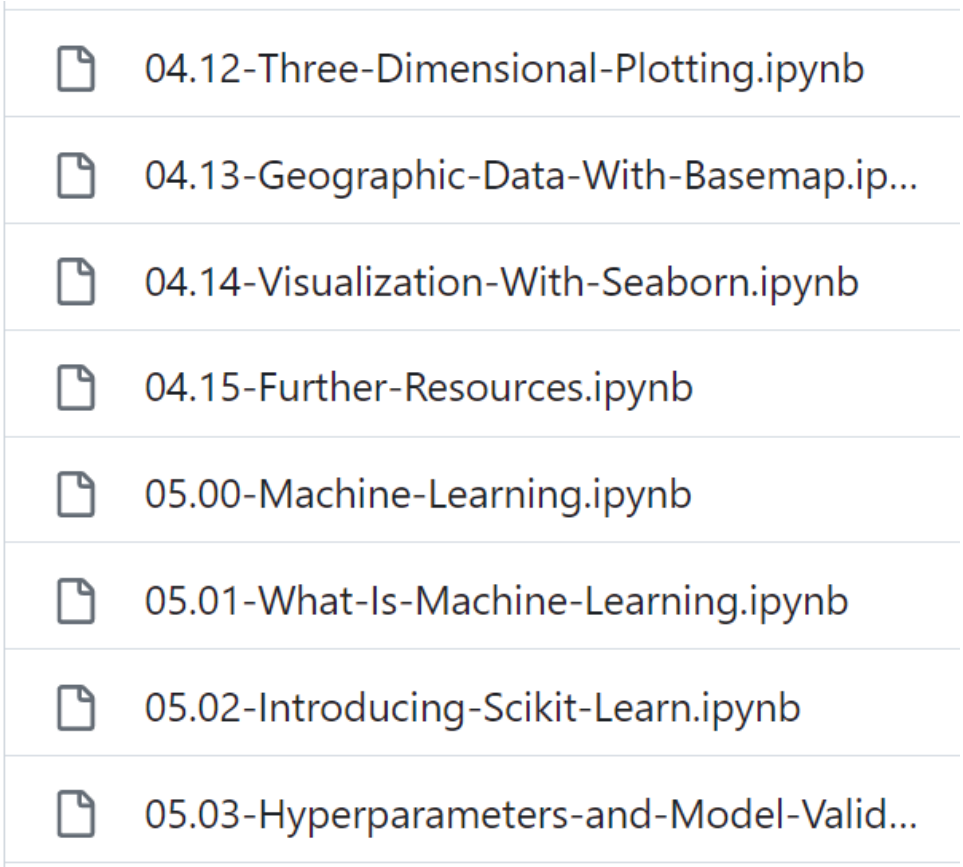
The Data Science Handbook

Free on github. Jupyter notebooks.

A few other free resources

The Data Science Handbook

Includes numpy, pandas, machine learning, etc.



A few other free resources

Daniel Schroeder's Website
Scientific Computing Resources
https://physics.weber.edu/schroeder/

Computational Physics e-text (Python/Jupyter)

Quantum Mechanics e-text (Mathematica)

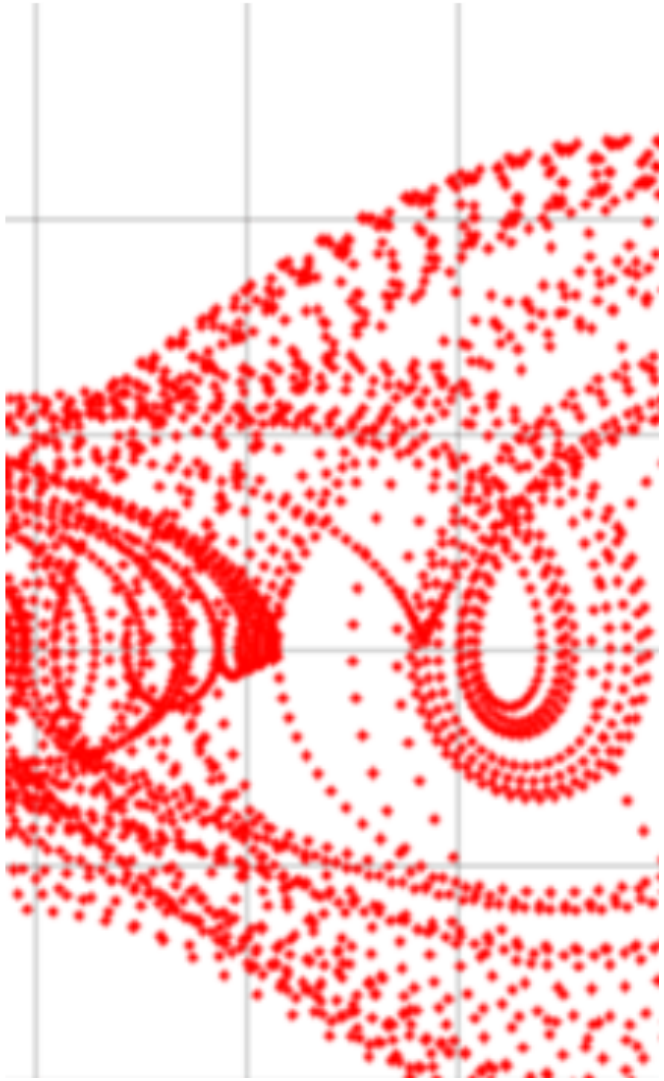
Thermal Physics book
(not free, but has free computational tutorials on website)

Scientific Computing (Physics 2300)

Daniel V. Schroeder, Department of Physics, Weber State University

- Syllabus
- *Physics Simulations in Python* (lab manual)
- LaTeX source and figure files for lab manual

- GlowScript home page
- VPython documentation
- Quick reference sheet and pdf version
- Download Anaconda
- Matplotlib.pyplot documentation
- Python resources for beginners



Contact for more resources :

doneil@bridgewater.edu (can find on BC website)

Data Science Education Community of Practice (DSECOP)
https://github.com/GDS-Education-Community-of-Practice/DSECOP
https://dsecop.org/

Partnership for Integration of
Computation into Undergraduate Physics (PICUP):
gopicup.org

Python for Everybody (PY4E.com)

Scientific Computing Resources
https://physics.weber.edu/schroeder/