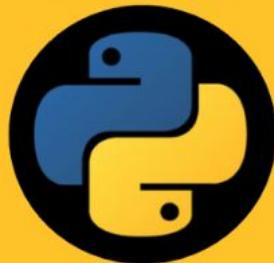


PROGRAMMING WITH PYTHON



GO.GWU.EDU/LIBWORKSHOPS

Objectives

- Gain familiarity with one environment for using Python (Google Colab), and awareness of others
- Learn Python language basics
- Learn how to look things up, how to interpret errors
- Gain confidence to try things we didn't learn today!

What about AI?!



Workshop

Feb
12

Thu 12PM - 2PM

[Using A.I. for Quantitative Data Analysis: Tools, Ethics, and Effective Practices](#)

Explore how A.I. can support quantitative data analysis, from data cleaning and visualization to drawing inferences and generating insights.



What about AI?!

Two things you will want to eventually learn how to do:

1. Use AI effectively as a companion to help you write code



Start coding or generate with AI.

2. Write code that uses AI models to accomplish things

Access Popular LLMs via Google-Colab-AI Without an API Key

Users with Colab's paid plans have free access to most popular LLMs via google-colab-ai Python library. For more details, refer to the [getting started with google colab ai](#).

```
from google.colab import ai
response = ai.generate_text("What is the capital of France?")
print(response)
```

Why Python?

- Free
- General purpose
- Easy to learn
- Readable*
- Community-developed / Open Source
- Widely used and documented
- Good built-in and contributed **libraries**



Different ways to use Python



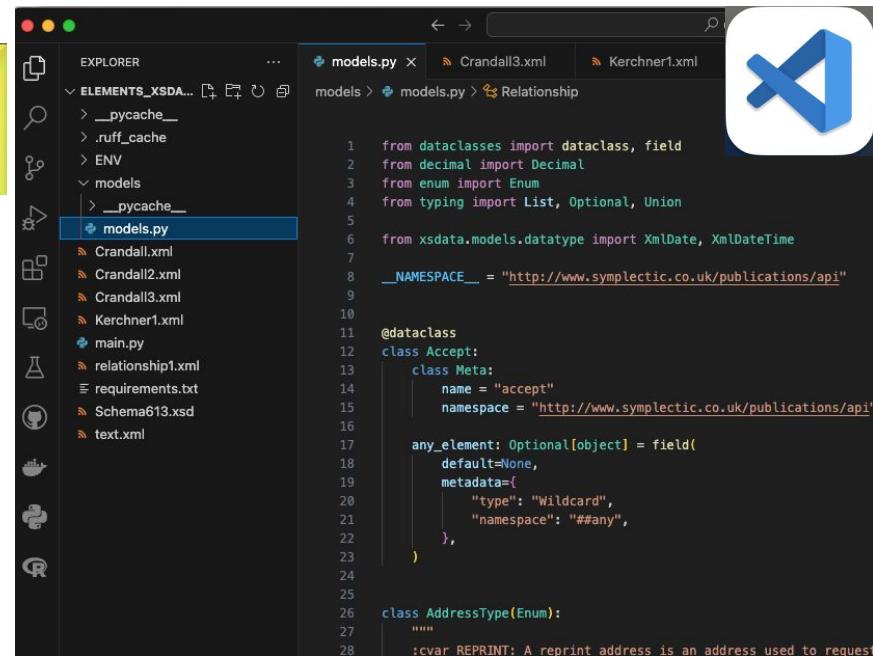
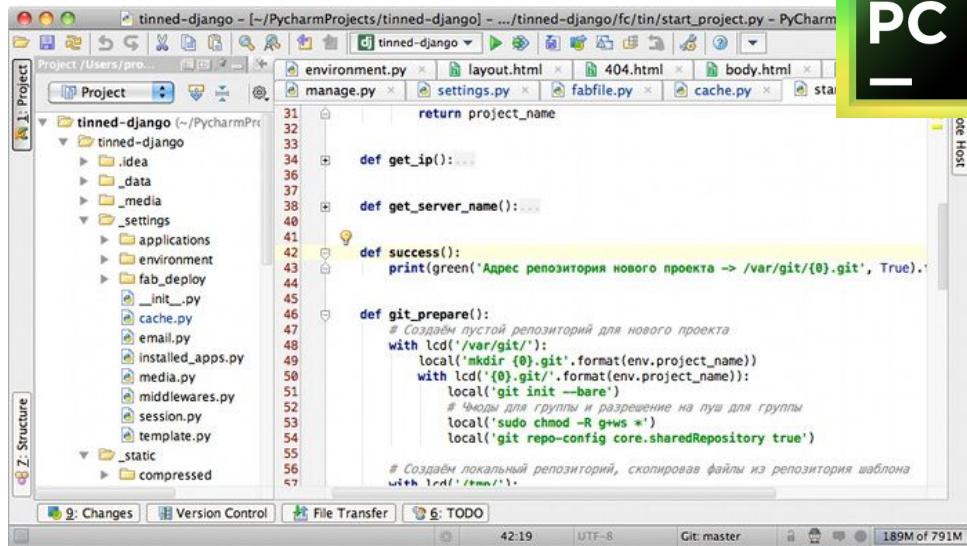
- Command line/REPL

```
Python 3.11.2 (v3.11.2:878ead1ac1, Feb  7 2023, 10:02:41) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> opinion = "This workshop is awful!"
>>> opinion == True
False
>>> █
```

Different ways to use Python

Integrated Development Environment (IDE)

pyCharm, VSCode, ...



Different ways to use Python (continued)

"Python Notebooks"

Jupyter/JupyterLab

Google Colab

(in your Google Drive!)

DataSpell

Kaggle

Deepnote

The screenshot shows the Deepnote interface. At the top, there's a navigation bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', 'Help', and a status message 'All changes saved'. Below the navigation bar is a search bar and a sidebar with project navigation. The main area displays a notebook titled 'Cookbook_Tutorial.ipynb' with code cells and output. One cell shows a DataFrame named 'fixed_df' with data for 'Berri 1'. Another cell contains a line plot of 'sepal_length' over time from January to February 2012. To the right of the notebook is a file browser showing a directory structure with files like 'waveome_overview.ipynb', 'my_notebook.ipynb', and 'libraries_by_python.ipynb'. A large pie chart in the bottom right corner illustrates the distribution of various Python libraries. The chart is divided into several segments, with the largest being 'numpy' (blue), followed by 'matplotlib' (orange), 'pandas' (green), and 'scikit-learn' (red). Other visible segments include 'requests', 'json', 'collections', 'warnings', 're', 'datetime', 'keras', 'IPython', 'sys', 'math', 'seaborn', 'scipy', 'os', and 'time'.

Even more ways to use Python

Anaconda = Python (and R) plus:

- Jupyter notebooks
- lots of libraries
 - data science
 - analytics
 - scientific computing
 - including:
Polars and Panda
 - (for working with data frames)



When you use a Python notebook...

A screenshot of a web browser window titled "colab.research.google.com". The page displays a "More memory" section with text about high-memory VMs and a code cell that prints RAM usage. A large callout bubble labeled "Your Python kernel (session)" is positioned to the right of the browser window, with a double-headed arrow indicating communication between the two.

```
import psutil
ram_gb = psutil.virtual_memory().total / 1e9
print('Your runtime has {:.1f} gigabytes of available RAM\n'.format(ram_gb))

if ram_gb < 20:
    print('Not using a high-RAM runtime')
else:
    print('You are using a high-RAM runtime!')
```



A screenshot of a web browser window titled "localhost:8888". The page displays a "More memory" section with text about high-memory VMs and a code cell that prints RAM usage. A large callout bubble labeled "Your Python kernel (session)" is positioned to the right of the browser window, with a double-headed arrow indicating communication between the two.

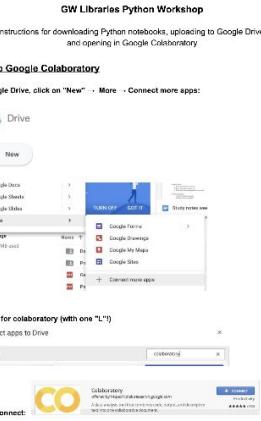
```
import psutil
ram_gb = psutil.virtual_memory().total / 1e9
print('Your runtime has {:.1f} gigabytes of available RAM\n'.format(ram_gb))

if ram_gb < 20:
    print('Not using a high-RAM runtime')
else:
    print('You are using a high-RAM runtime!')
```



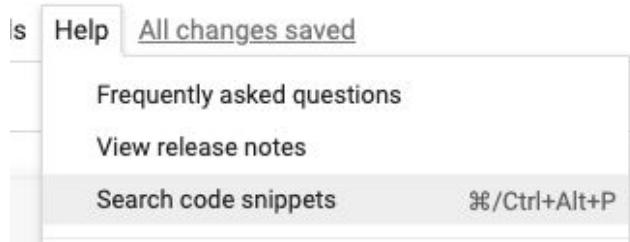
Setup

- Google Colaboratory
colab.research.google.com



* *To do later:* If you aren't already seeing this  , then sign up for [free] Google Colab Pro for Education!
colab.research.google.com/signup

Uploading Data (and doing other things) in Google Colab



The 'Code snippets' sidebar is open. At the top is a search bar with the word 'upload'. Below the search bar is a list of options: 'Open files from your local file system', 'Saving data to Google Drive', 'Saving data with gsutil', and 'Saving data with the Cloud Storage Python API'. Each option has a '+' sign next to it.

Use "Code Snippets" (searchable!) →

A code snippet for file upload is shown. It starts with 'from google.colab import files' and ends with 'print("User uploaded file "{name}" with length {length} bytes')'. The code is highlighted in purple, red, and blue. An 'Insert' button is visible at the top right of the code block.

```
from google.colab import files  
  
uploaded = files.upload()  
  
for fn in uploaded.keys():  
    print('User uploaded file "{name}" with length {length} bytes')  
    name=fn, length=len(uploaded[fn]))
```

Some recommendations

- Write assuming your code will be read (incl. by Future You)
- Version your code  GitHub
- Learn to be "Pythonic" in your style
- Isolate your projects from each other
 - Use venv, conda, and/or pixi.sh
- Stuck? Try an Internet search
- Which version of Python are you using?
- Find good code examples and make them work
- Keep learning!

Some Python libraries/frameworks

Building web applications	Django Flask
Scientific/numerical	Numpy Scipy Pandas
Machine Learning	scikit-learn, tensorflow
Data Visualization (check out www.pythongraphgallery.com)	matplotlib bokeh ggplot (like ggplot2 in R) plotly (<- interactive) seaborn

Things we learned today in Python that most coding languages also share (Part 1)

- variables
- different data types: numeric, text, logical, etc.
- data structures for holding more than a single value: lists/arrays/matrices/etc.
- loops
- conditional logic (if/then)
- functions
- libraries/packages for bringing in extra functionality

Data analysis you can perform using Pandas

- loading in (reading in) a data set
- subsetting based on columns and/or rows based on data criteria
- exploring data variables, both numerical and text/categorical
- merging/joining data frames
- plotting data (e.g., with matplotlib and with ggplot2)

To Learn More

- PyFlo pyflo.net ← NEW!
- Kaggle: kaggle.com/learn
- learnpython.org
- [Software Carpentry](http://software-carpentry.org), [Data Carpentry](http://datacarpentry.org) (not just Python)
- docs.python.org/3/tutorial (and docs.python.org)
- [GW Online: Get data off the ground with Python](#)
- [Upcoming Python workshops @ GW Libraries](#)
- LinkedIn learning it.gwu.edu/linkedin-learning courses
 - 253 Python, 6 Pandas
- More on Pandas:
 - Pandas cookbook: github.com/jvns/pandas-cookbook
 - pandas.pydata.org/pandas-docs/stable/10min.html
 - pandas.pydata.org/pandas-docs/stable/tutorials.html
 - pandas.pydata.org/pandas-docs/stable/cookbook.html
 - www.datacarpentry.org/python-ecology-lesson/

Contact us:

Coding Consultations (with Dan & colleagues):

calendly.com/gwul-coding – Python, R, HTML/CSS/JavaScript

Stats Appointments (with Stats grad students):

go.gwu.edu/dataconsulting

Workshop Materials: go.gwu.edu/pyw

kerchner@gwu.edu