Immersive Python Workshop

January 9 and 10th, 2025

Day 1 (Virtual)

TEXAS Libraries
The University of Texas at Austin
University of Texas Libraries

Presented by the
UT Libraries &
Open Source Program Office (UT-OSPO)

UT-OSPO
UT Austin Open Source
Program Office

Sloan grant number: G-2023-20944

Itinerary for the Day

Time:	
1:00 PM	Welcome
1:10 PM	Instruction: Intro to Python Basics
2:00 PM	Break / Repeat of welcoming remarks for more advanced attendees
2:10 PM	Instruction: Python Essentials
3:00 PM	Instruction: Managing and processing files
4:00 PM	Python and VSCode local installation help (if required)
5:00 PM	Close

Land Acknowledgement

We would like to acknowledge that we are meeting on the Indigenous lands of Turtle Island, the ancestral name for what now is called North America.

Moreover, (I) We would like to acknowledge the Alabama-Coushatta, Caddo, Carrizo/Comecrudo, Coahuiltecan, Comanche, Kickapoo, Lipan Apache, Tonkawa and Ysleta Del Sur Pueblo, and all the American Indian and Indigenous Peoples and communities who have been or have become a part of these lands and territories in Texas.

Logistics

- Shared notes document (https://bit.ly/ipw-jan2025)
  - Important URLs will be provided here
- Asking questions
- Timing of breaks
- Workshop materials and slides will be posted on GitHub
- Further logistics for our in-person day tomorrow will be shared at the start of the Friday session

QR code for shared notes document

Utilizing the Scholars Lab

- **Data Lab**
  - Data and Donuts
  - Digital Humanities Workshops
  - DH Day
  - More

- **Project Rooms**
  - Digital Scholarship related projects
  - Reservable

- **Scan Tech Studio**
  - Supports Optical Character Recognition and Handwritten Text Recognition
  - Scanners and PC
  - Reservable

- **Digital Scholarship Consultation**

Instruction Team

Instructors:

Michael Shensky (Head of Research Data Services)

Jeremy Thompson (Digital Processing Archivist)

Alex Marden (GIS and Geospatial Data Coordinator)

Ian Goodale (European Studies Librarian)

Bryan Gee (Open Research Coordinator for Data and Software)

Karina Sanchez (Scholars Lab Librarian)

Helpers:

Grant Hardaway (Psychology and Life Sciences Librarian)

Meryl Brodsky (Information & Communication Librarian)

Code of Conduct

- Use welcoming and inclusive language
- Be respectful of different viewpoints and experiences
- Gracefully accept constructive criticism
- Focus on what is best for the community
- Show courtesy and respect towards other community members

Contact any Helper or Instructor for assistance

Day 1

Intro to Python

Goals for the Intro to Python Section of the Workshop

- ➔ Introduce the fundamental knowledge required to help you get started using Python
- ➔ Provide information about related resources you can use to learn more about Python moving forward
- ➔ Showcase the benefits of using Python for reproducible research
- ➔ Learn how to use Python without installing any software - this workshop will use Google Colab
  - ◆ We will focus on using a local Python installation during tomorrow's session

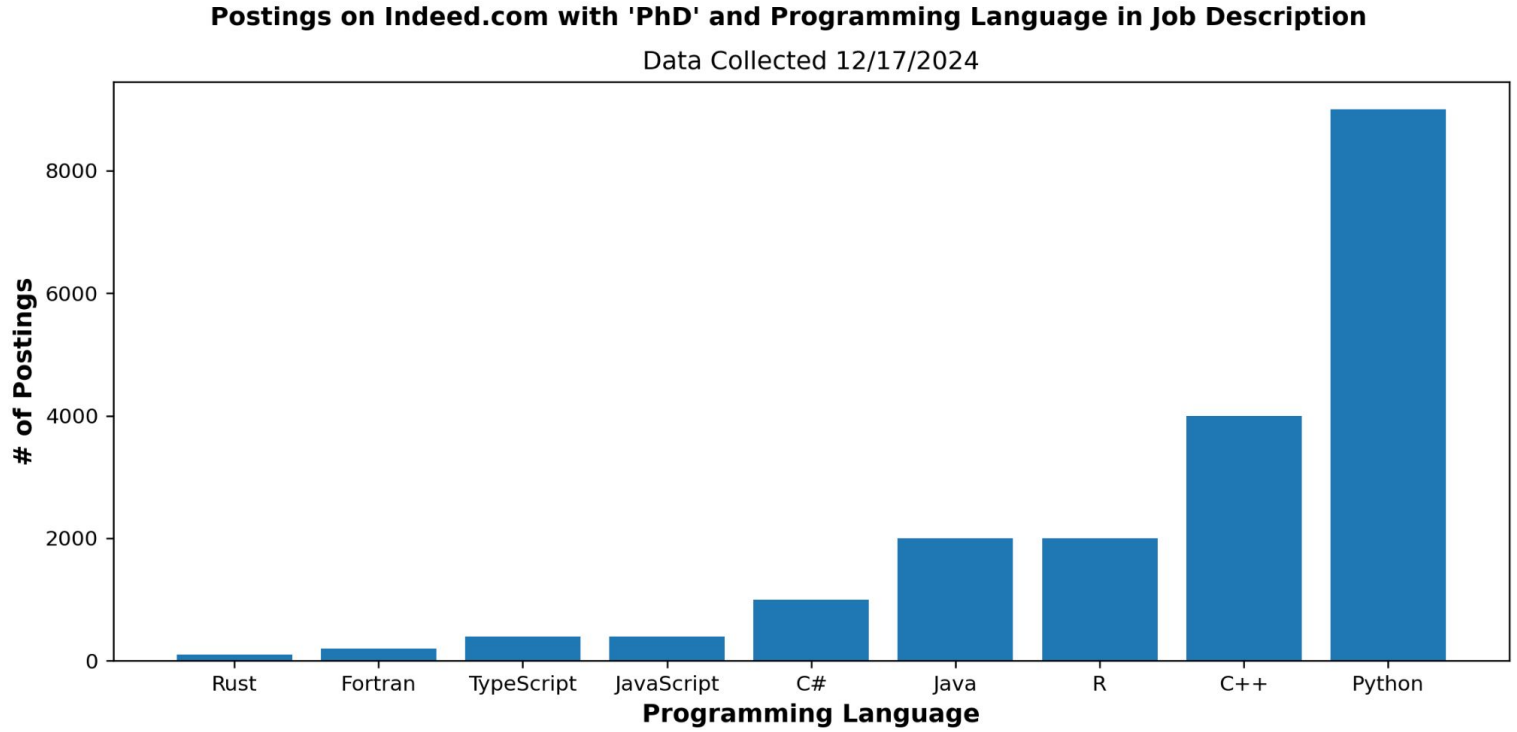
What is Python?

- ➔ Open source, interpreted programming language
- ➔ Cross platform compatible (Windows, MacOS, & Linux)
- ➔ Extensive use in a variety of fields
- ➔ Large ecosystem of open source packages
- ➔ Can be used for file management, analyzing data, editing data, visualizing data, and more!

Why is Python Important?

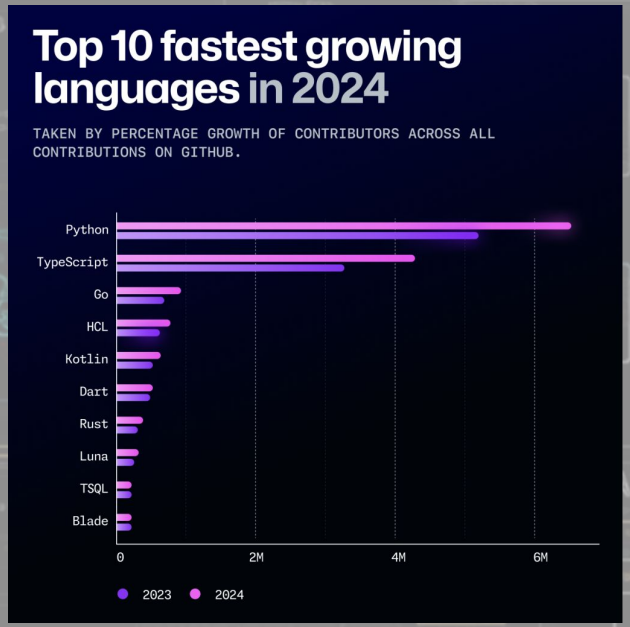
- **Efficiency** — Automated process can save hours, days, or even weeks of time
- **Reproducibility** — Code can be shared with other researchers so they can verify your results and learn from your methods
- **Collaboration** — Code can be shared and collaboratively developed on platforms like GitHub and GitLab
- **Scheduled tasks** — Processes can be set to run at specific times or can be triggered by set events
- **Creativity** — Usually more interesting to write code to automate a tedious process than do it manually

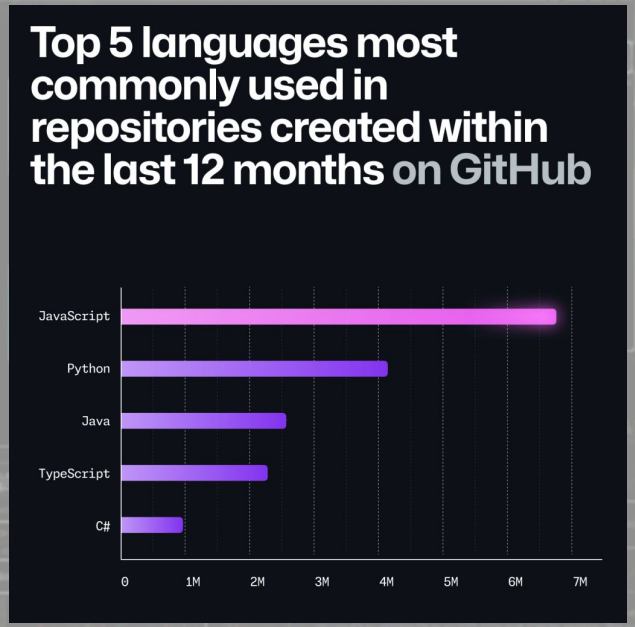
Why Focus on Python Specifically?



Why Focus on Python Over Other Languages?

Python was the most popular language for open source software projects on GitHub in 2024, and is still continuing to grow in popularity





"The rise in Python usage correlates with large communities of people joining the open source community from across the STEM world rather than the traditional community of software developers."

Source for data and graphs: https://github.blog/news-insights/octoverse/octoverse-2024/

Using Python for Data Management

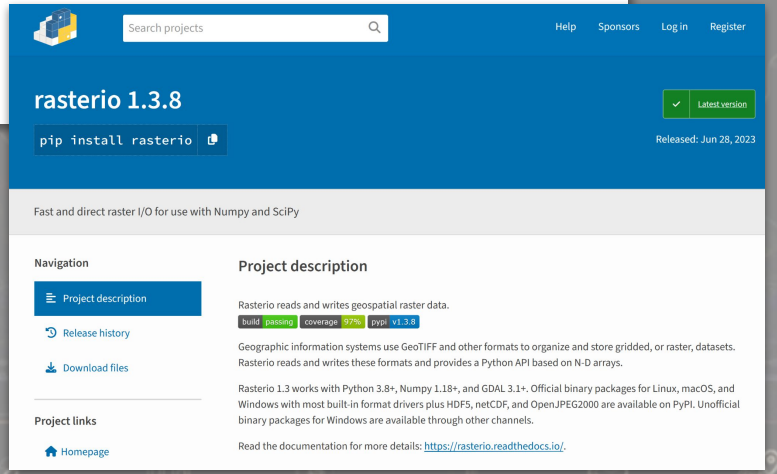
+ Look through all files in a directory
+ Open files and edit content
+ Bulk create, delete, move, and rename files and directories
+ Work with data in large spreadsheets
+ Retrieve data from external sources using an API
+ Schedule scripted tasks like automated data backup or publishing routines
+ Many more useful operations!

Installing Python Locally

- Might already be on your computer due to it being included with your OS or other software
- Be careful if you have multiple Python versions installed…
- You can install Python using the default installer or you can use a package manager like Anaconda
- Once Python is installed, you will likely want to install extra packages to extend the its functionality
  - You can do this with **pip** or with **Anaconda**
- Using a virtual environment can help manage Python and package versions on your local machine

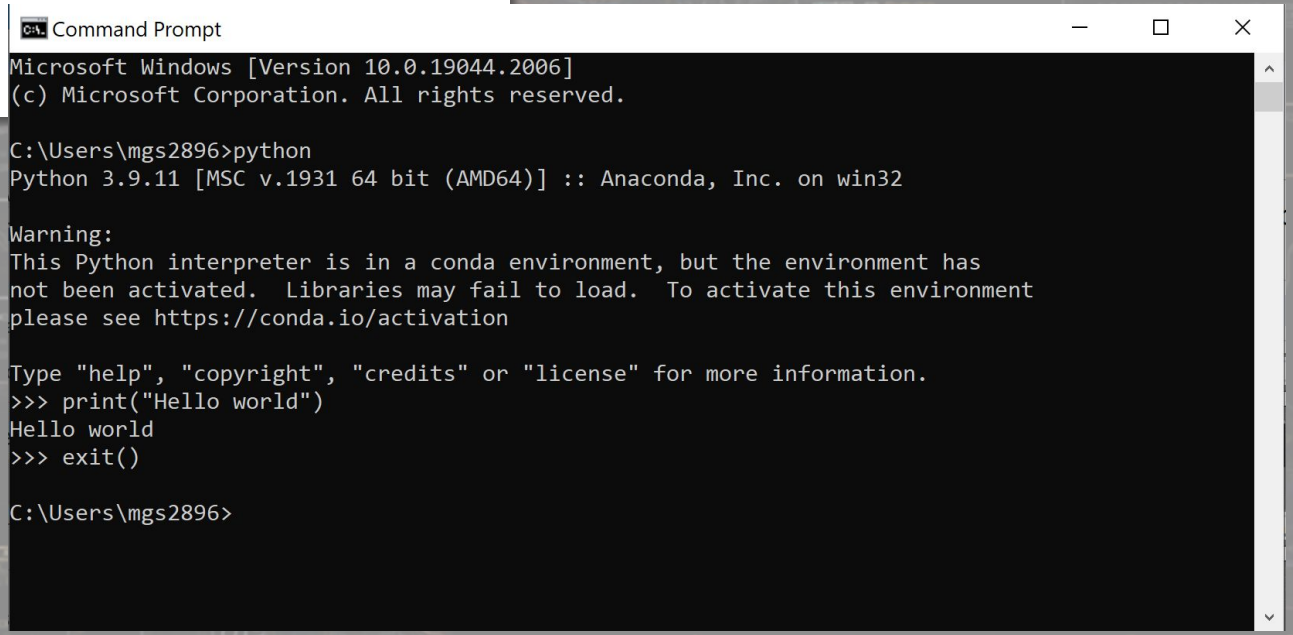
Installing New Python Packages

- **pip** is the default package installer for Python
- You can use pip to install packages from the Python Package Index and other indexes
- Although generally safe, packages should always be vetted before installation and use
- pip can be used to install packages if you are running Python locally or in Google Colab
- Installing and uninstalling packages is simple and fast



Integrated Development Environments (IDEs)

You **can** write Python code and run it from your Command Prompt or Terminal…



Integrated Development Environments (IDEs)

You **can** write Python code and run it from your Command Prompt or Terminal…**but** using an IDE makes writing code much easier

An **IDE (integrated development environment)** is a software solution designed to facilitate writing code

- **Jupyter** (https://jupyter.org/)
- **VS Code** (https://code.visualstudio.com/)
- **Pulsar** (https://pulsar-edit.dev/)
- **PyCharm** (https://www.jetbrains.com/pycharm/download/)
- **Spyder** (https://www.spyder-ide.org/)

```python
import json
import numpy as np
import os
import pandas as pd
import re
import requests
from datetime import datetime, timedelta

test = False

## setting timestamp at start of script to calculate run time
startTime = datetime.now()
## creating variable with current date for appending to filenames
startDate = datetime.now().strftime("%Y-%m-%d")

current_year = datetime.now().year
previous_year = current_year - 1

if test:
    if os.path.isdir("test"):
        print("test directory found - no need to recreate")
    else:
        os.mkdir("test")
        print("test directory has been created")
    os.chdir('test')
    if os.path.isdir("outputs"):
        print("test outputs directory found - no need to recreate")
    else:
        os.mkdir("outputs")
        print("test outputs directory has been created")
else:
    if os.path.isdir("outputs"):
        print("outputs directory found - no need to recreate")
    else:
        os.mkdir("outputs")
        print("outputs directory has been created")

print("Beginning to define API call parameters.")
```

Using VS Code as your Python IDE

➔ VS Code is a great IDE for Python because it is:

- Free
- Open source
- Cross platform compatible
- Widely utilized
- Extensible and customizable

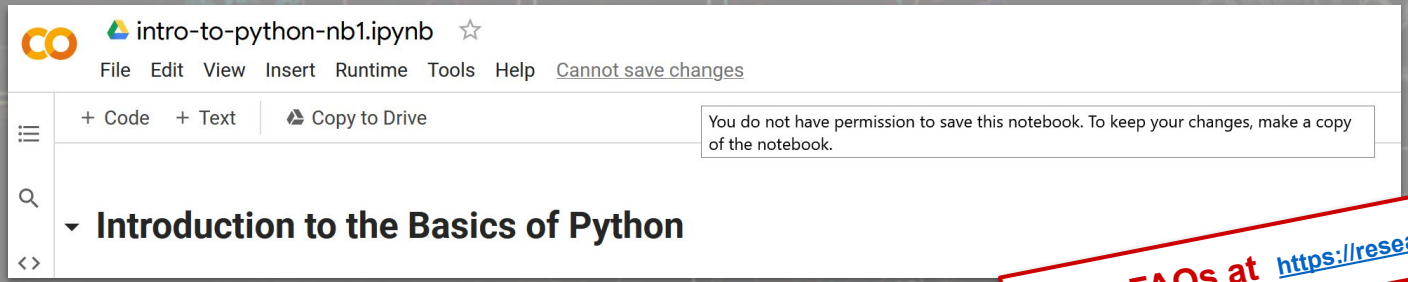
Visual Studio Code

Download VS Code at
https://code.visualstudio.com/

```python
import json
import numpy as np
import os
import pandas as pd
import re
import requests
from datetime import datetime, timedelta

test = False

## setting timestamp at start of script to calculate run time
startTime = datetime.now()
## creating variable with current date for appending to filenames
startDate = datetime.now().strftime("%Y-%m-%d")

current_year = datetime.now().year
previous_year = current_year - 1

if test:
    if os.path.isdir("test"):
        print("test directory found - no need to recreate")
    else:
        os.mkdir("test")
        print("test directory has been created")
    os.chdir('test')
    if os.path.isdir("outputs"):
        print("test outputs directory found - no need to recreate")
    else:
        os.mkdir("outputs")
        print("test outputs directory has been created")
else:
    if os.path.isdir("outputs"):
        print("outputs directory found - no need to recreate")
    else:
        os.mkdir("outputs")
        print("outputs directory has been created")

print("Beginning to define API call parameters.")
```

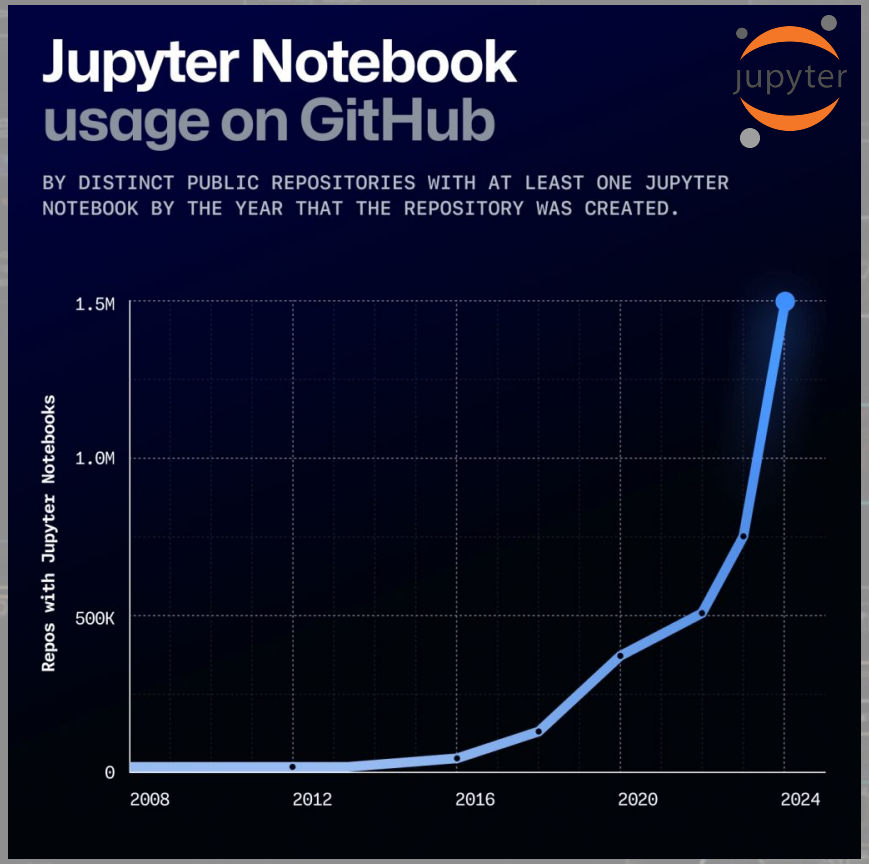
Google Colab

- **Colab is a free Google service that allows you to create and run Jupyter Notebooks in the cloud**
- **Allows you to write Python code and text notes in compartmentalized cells within a notebook**
  - Code cells can be run individually to allow for previewing outputs and troubleshooting issues
- **Notebooks are stored in Google Drive and can access other files in Google Drive**



Colab FAQs at https://research.google.com/colaboratory/faq.html

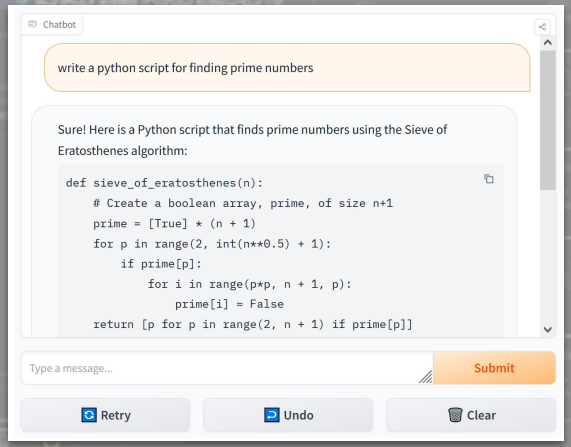
Google Colab



Further UTL Python Learning Opportunities

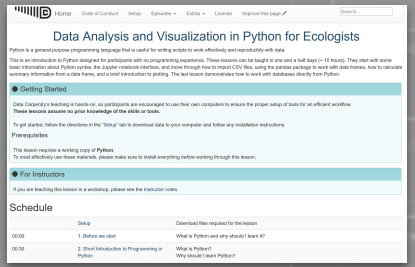
What	Workshops covering research data practices and software
When	On the following dates: 1/31, 2/11, 2/12, 2/13, 2/14, and 2/28
Time	12pm - 1:15pm
Where	Zoom (all dates) / PCL Scholars Lab (select dates)
More info	https://guides.lib.utexas.edu/data-and-donuts

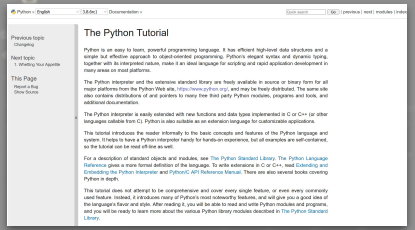
What About AI for Generating Code?



- LLMs like Llama Code, ChatGPT, or Microsoft Copilot can generate code in response to a prompt
- The code they generate may work, but often will not be as specific to your research questions as you may prefer and may need fixing or customization.
- Having a strong knowledge of what you're trying to accomplish and how to write and structure your own code is essential before using AI tools to supplement your programming.

Python Resources, Documentation, and Help



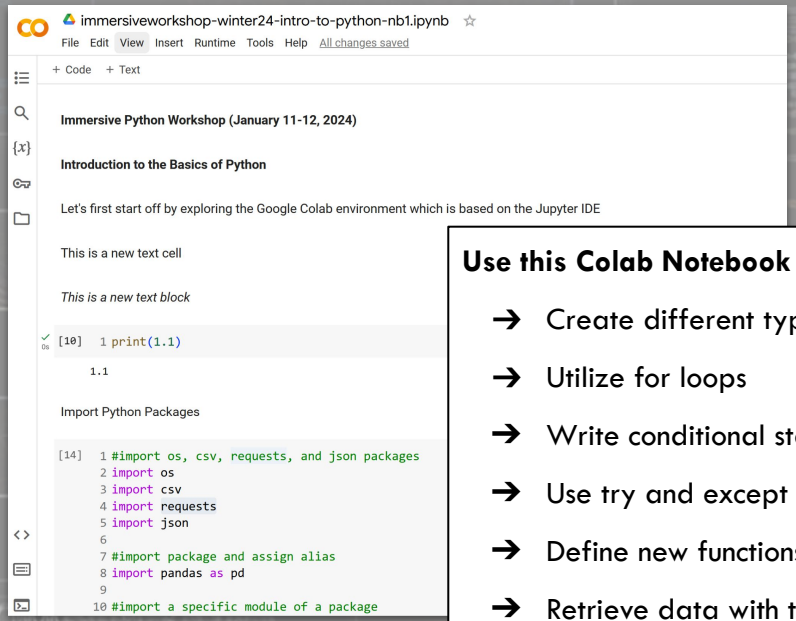


- ★ **Stack Overflow** (https://stackoverflow.com/)
- ★ **Data Carpentry** (https://datacarpentry.org/python-ecology-lesson/)
- ★ **Python Documentation** (https://docs.python.org/3/library/index.html)
- ★ **Python Tutorial** (https://docs.python.org/3/tutorial/index.html)
- ★ **LinkedIn Learning** (https://www.linkedin.com/learning/topics/python)
- ★ **UT Statistics Short Courses** (https://stat.utexas.edu/training/software-short-courses)
- ★ **Google Colab: Intro to Python** (https://colab.research.google.com/notebooks/intro.ipynb)

DEMO: Basics of Python for Reproducible Research

Python Basics

- ★ Packages and modules
- ★ Variables and object types
- ★ Operators
- ★ Methods and functions
- ★ Syntax
- ★ Comments
- ★ Conditional statements
- ★ Loops
- ★ Handle errors gracefully



Use this Colab Notebook to practice using Python to...

- ➔ Create different types of Python objects
- ➔ Utilize for loops
- ➔ Write conditional statements
- ➔ Use try and except to catch error
- ➔ Define new functions
- ➔ Retrieve data with the requests module

Link to publicly shared Google Colab Notebook:

https://colab.research.google.com/drive/1v_INLljzlZmxgPG_2bgMT-qZB546NgU7#scrollTo=NO_W7wUspkUM

Software Installation Support

Software Installation Support

★ **Python (https://www.python.org/downloads/)**

- Make sure you can install new Python packages using the Terminal (MacOS) or Command Prompt (Windows)
- Try the following command on your PC: **python -m pip install pandas**
- Try the following command on your Mac: **python3 -m pip install pandas**

★ **VSCode (https://code.visualstudio.com/download)**

- Create a new file with the extension .py (to designate the file as a Python script)
- Try clicking the play/run button in the upper right corner of VSCode to run the script

★ **Git (https://git-scm.com/downloads)**

★ **Github Desktop (https://desktop.github.com/download/)**

See next slide for troubleshooting common issues

Software Installation Common Issues

★ Mac

- **ISSUE**: you cannot run a Python script in VS Code
- **CAUSE**: the *Python default interpreter path* is not set correctly
- **SOLUTION**: In VSCode go to *File* > *Preferences* > *Settings and search for the Python default interpreter path* setting - when you find it, update the the value to the correct Python interpreter path. If you don't know the path to the interpreter, open your Terminal, type python3, import sys, and the run sys.executable

★ PC

- **ISSUE:** trying to *run python* in the command prompt opens the Windows Store
- **CAUSE:** the directory containing your Python interpreter is not correctly listed in your system Environment Variables' *path* listing
- **SOLUTION:** use the full path to your Python interpreter in the Command Prompt. If you don't know the path to the interpreter search on your file system until you find out where it is installed and then use the full path wrapped in quotation marks in the command prompt (e.g. use 'C:\\Program Files\\python\\python.exe' but note the exact path may be different on your computer)

Helpers can stay past 4pm today to help resolve issues - having everything working by start of Day 2 is important