

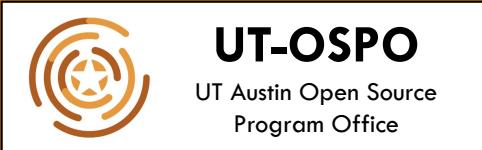
Immersive Python Workshop

August 15-16, 2024

Day 1



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



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.

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**



Logistics

- Asking questions
- Food
- Timing of breaks
- Bathroom locations
- Logging in to the computers
- Shared notes document (<https://bit.ly/python-aug24-notes>)
 - ◆ Important URLs will be provided here



QR code for shared
notes document

Instruction Team

Instructors:

Michael Shensky (Head of Research Data Services)

Jeremy Thompson (Digital Archivist)

Alex Marden (GIS and Geospatial Data Coordinator)

Ian Goodale (European Studies Librarian)

Helpers:

Bryan Gee (Open Research Coordinator for Data and Software)

Karina Sanchez (Scholars Lab Librarian)

Hosts:

Meryl Brodsky (Communications Librarian)

Allyssa Guzman (AD for Research Support and Digital Initiatives)

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 Host, 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 online 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](#)

What is Python?

- Open source, interpreted programming language
 - Cross platform compatible (Windows, MacOS, & Linux)
 - Current version is 3.12.5 (as of Aug. 2024)
 - Extensive use in a variety of fields
 - Large ecosystem of open source packages
 - Can be used for file management, analyzing data, edi



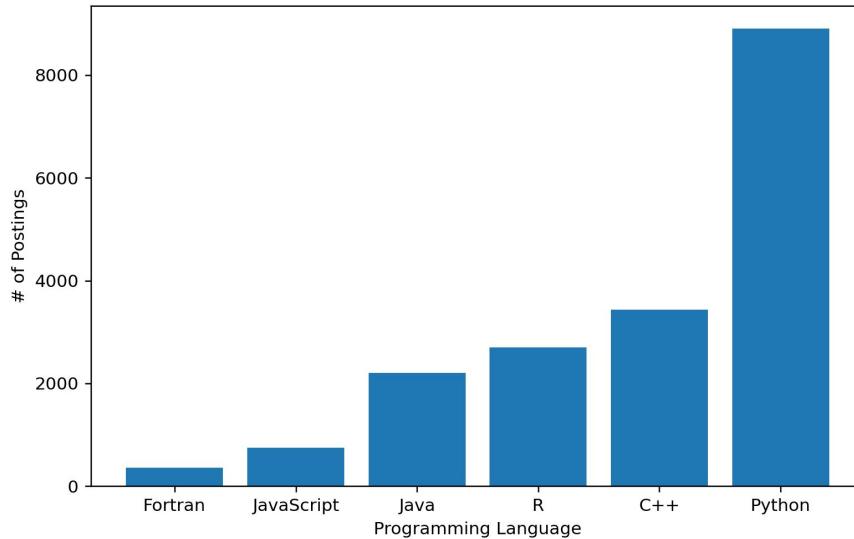
Why is Python Important?

- **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?

Postings on Indeed.com with 'PhD' and Programming Language in Job Description

Data Collected 1/8/2024



```
import os
import matplotlib.pyplot as plt

categories = []
counts = []

languageJobsdata = []

languageJobsdata.append({"name": "Python", "jobcount": 8803})
languageJobsdata.append({"name": "C++", "jobcount": 34391})
languageJobsdata.append({"name": "R", "jobcount": 2700})
languageJobsdata.append({"name": "Java", "jobcount": 2205})
languageJobsdata.append({"name": "JavaScript", "jobcount": 752})
languageJobsdata.append({"name": "Fortran", "jobcount": 369})
# languageJobsdata.append({"name": "Perl", "jobcount": 303})
# languageJobsdata.append({"name": "GoLang", "jobcount": 89})
# languageJobsdata.append({"name": "Kotlin", "jobcount": 577})
# languageJobsdata.append({"name": "Rust", "jobcount": 46})

languageJobsdata.reverse()

categories = [lang['name'] for lang in languageJobsdata]
counts = [lang['jobcount'] for lang in languageJobsdata]

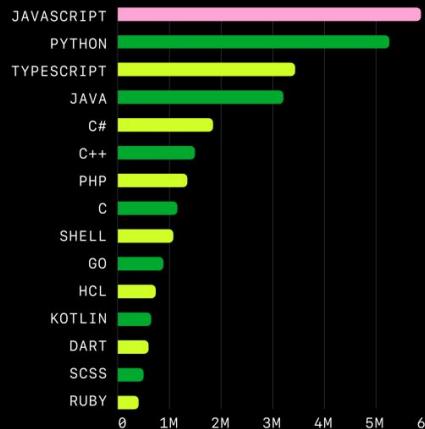
fig, ax = plt.subplots(figsize=(8, 5))
ax.bar(categories, counts)
ax.set_title("Data Collected 1/8/2024")
ax.set_ylabel("# of Postings")
ax.set_xlabel("Programming Language")
fig.suptitle("Postings on Indeed.com with 'PhD' and Programming Language in Job Description", fontsize=12, fontweight='bold')
plt.show()
```

Why Focus on Python Over Other Languages?

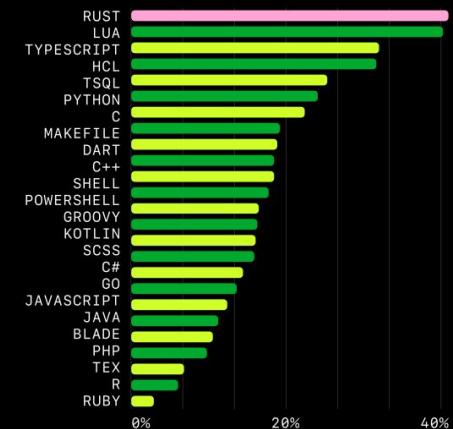
Python was the 2nd most popular language for open source software projects on GitHub in 2023

It is still continuing to grow in popularity with usage on GitHub up significantly year over year

The top languages in 2023 by usage



Top languages by percentage growth of contributors



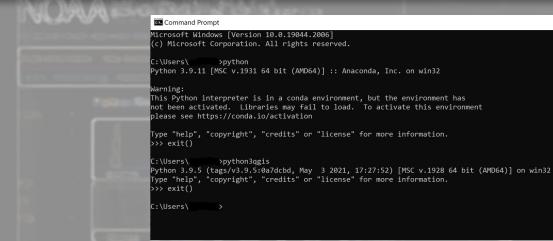
Source for data and graph: <https://github.blog/news-insights/research/the-state-of-open-source-and-ai/#the-most-popular-programming-languages>

Using Python for Data Management

- + Look through all files in a directory
- + Open files and edit content
- + Work with data in large spreadsheets
- + Create, delete, move, and rename files and directories
- + 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



```
Microsoft Windows [Version 10.0.19044.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\...> python
python 3.9.11 [MSC V1293] 64 bit (AMD64) :: Anaconda, Inc. on win32
Warning:
The Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>> exit()

C:\Users\...> python3.9
Python 3.9.11 (tags/v3.9.11/d4d83c07, May 3 2021, 17:37:32) [GCC v10.2.0 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()

C:\Users\...
```

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

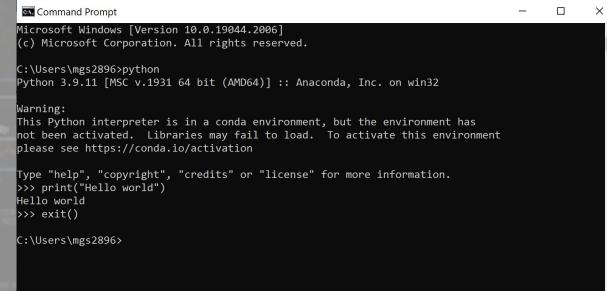
A screenshot of the Python Package Index website. The top navigation bar includes links for "Search projects", "Help", "Sponsors", "Log in", and "Register". A search bar shows the query "pip install rasterio". The main content area displays the project page for "rasterio 1.3.8". The page header says "Fast and direct raster I/O for use with NumPy and SciPy". It features a "Project description" section with a "Project description" button, a "Release history" button, a "Download files" button, and a "Project links" button. The "Project description" section contains detailed information about the rasterio library, its compatibility with Python 3.8+, NumPy 1.18, and GDAL 3.1.1, and its use of various file formats like GeoTIFF and JPEG2000. It also mentions that rasterio works with both in-memory and disk-based datasets. The "Project links" section includes a "Homepage" link and a note to read the documentation for more details at <https://rasterio.readthedocs.io>.

Jupyter and Other Free and Open Source 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/>)



```
C:\Users\mgs2896>python
Microsoft Windows [Version 10.0, Build 19044.2006]
(c) Microsoft Corporation. All rights reserved.

Python 3.9.11 [MSC v.1931 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello world")
Hello world
>>> exit()

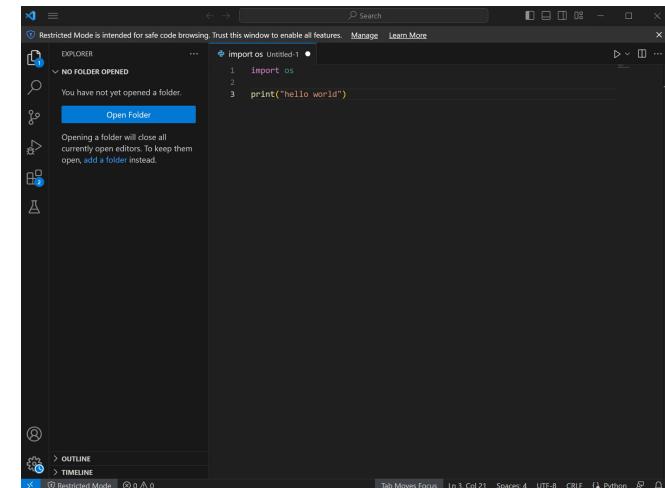
C:\Users\mgs2896>
```



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/>

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**



CO intro-to-python-nb1.ipynb ☆

File Edit View Insert Runtime Tools Help Cannot save changes

+ Code + Text Copy to Drive

You do not have permission to save this notebook. To keep your changes, make a copy of the notebook.

☰

▼ Introduction to the Basics of Python

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

Python Resources, Documentation, and Help



This is an introduction to Python designed for participants with no programming experience. These lessons can be taught in one day (a half day = 10 hours). They start with some basic concepts of what Python is and how it works, then move on to more advanced topics such as data structures, loops, functions, and classes. Participants will learn how to extract summary information from a data frame, and brief introduction to plotting. The last lesson demonstrates how to work with databases directly from Python.

Getting Started
Data Carpentry's teaching is hands-on, so participants are encouraged to use their own computer to ensure the proper setup of tools for an efficient workflow. These lessons assume no prior knowledge of the skills or tools.

To get started, follow the directions in the "Setup" tab to download data to your computer and follow any installation instructions.

For Instructors
This lesson includes a working copy of Python. To most effectively use these materials, please make sure to install everything before working through this lesson.

Schedule

Setup	Download files required for the lesson
00:00	1. Before we start What is Python and why should I learn it?
00:30	2. Short Introduction to Programming in Python What is Python? Why should I learn Python?

The Python Tutorial
Python is a simple, powerful, elegant programming language. It has efficient dynamic data structures and a simple but effective syntax. To emphasize readability, Python's elegant syntax and dynamic typing, together with its interpreted nature, make it ideal language for scripting and rapid application development in many domains.

The Python interpreter and the extensive standard library are freely available in source or binary form for most platforms. Python is also available in many forms of pre-compiled modules and packages, and even complete distributions of end products in many form that partly Python modules, programs and tools, and even entire systems.

The Python interpreter is easily extended via new functions and data types implemented in C (or other languages called from C). Python also includes an extension language for customize applications.

This tutorial introduces the reader informally to the basic concepts and features of the Python language and system. It tries to present the Python language in a way that makes it easy for beginners to understand, but it also touches on more advanced topics in an attempt to give the reader a well-rounded view of what Python has to offer.

For a description of selected objects and modules, see the Python Standard Library. The Python Language Reference provides a detailed description of the language. To learn more about the Python language and system, see the Python Interpreter and Python API Reference Manual. There are also several books covering Python in depth.

The tutorial does not attempt to be comprehensive and cover every single feature, or even every commonly used feature. It is not intended to replace any of the other resources mentioned here, but rather to give you a feel for the language's flavor and style. After reading it, you will be able to read and write Python modules and programs, and you will be ready to look more closely at the various Python library modules described in The Python Standard Library.

★ **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>)

Further UTL Python Learning Opportunities

Data & Donuts

What Workshops covering research data practices and software

When On the following Fridays: 9/13, 9/27, 10/11, 10/25

Time 12pm - 1:15pm

Where Zoom (all dates) / [PCL Scholars Lab](#) (select dates)

More info <https://guides.lib.utexas.edu/data-and-donuts>

UT GIS Day



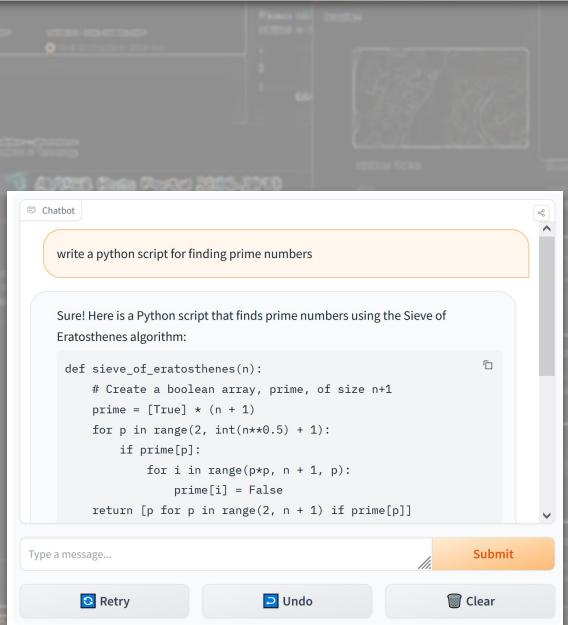
What Celebrate and learn about GIS and geospatial research at UT

When Wednesday, November 20th, time TBA

Where [PCL Scholars Lab](#)

More info <https://guides.lib.utexas.edu/gis/ut-gis-day>

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.

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

The screenshot shows a Google Colab notebook interface. The title bar reads 'immersiveworkshop-winter24-intro-to-python-nb1.ipynb'. The notebook contains several code cells and text cells. One code cell shows the command `print(1.1)`. Another code cell imports packages like os, csv, requests, and json. Text cells provide introductory text about the workshop and the Colab environment.

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/172m3_k2IVD-NQyuDN8Av4oAScQbXMIEZ#scrollTo=pNQyVAOBhB1g

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)
 - The command should be: python -m pip install packagename
- ★ Github Desktop (<https://desktop.github.com/download/>)
- ★ 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