July 19, 2023 Think-a-Thon

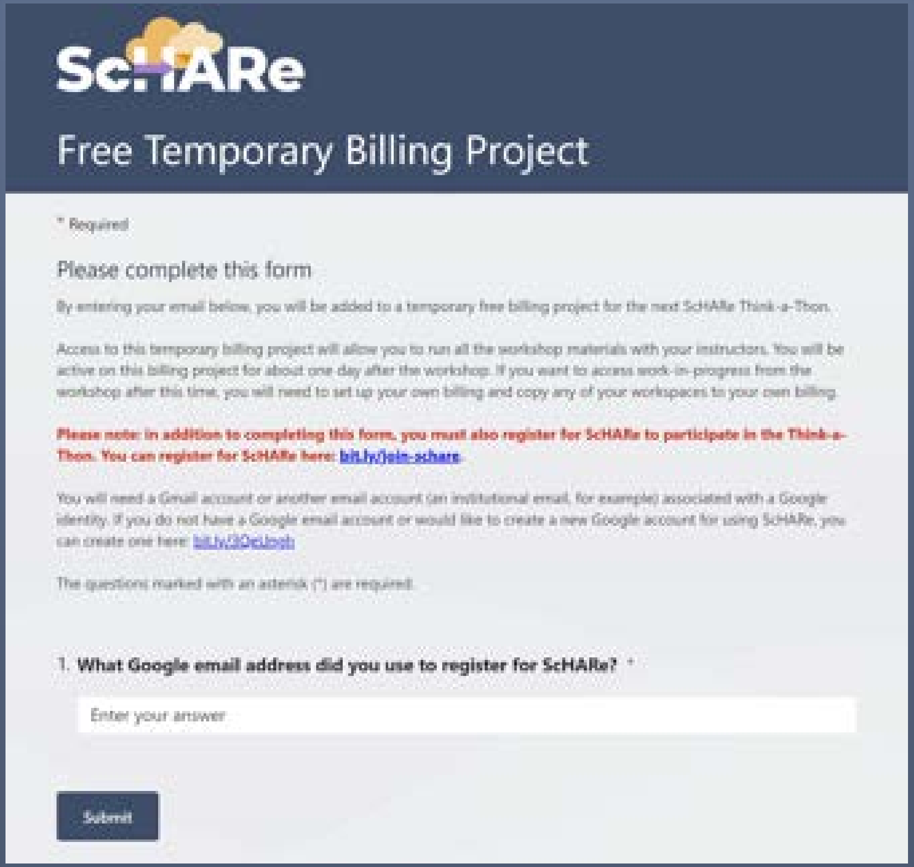
ScHARe

An Introduction to Python for Data Science

Deborah Duran, PhD and Luca Calzoni, MD MS PhD Cand. | NIMHD

Sign up for free temporary billing

Have you filled out the 1-question form on the Think-a-Thon registration confirmation email?



Free Temporary Billing Project

* Required

Please complete this form

By entering your email below, you will be added to a temporary free billing project for the next ScHARe Think-a-Thon.

Access to this temporary billing project will allow you to run all the workshop materials with your instructors. You will be active on this billing project for about one day after the workshop. If you want to access work-in-progress from the workshop after this time, you will need to set up your own billing and copy any of your workspaces to your own billing.

Please note: in addition to completing this form, you must also register for ScHARe to participate in the Think-a-Thon. You can register for ScHARe here: bit.ly/join-schare.

You will need a Gmail account or another email account (an institutional email, for example) associated with a Google identity. If you do not have a Google email account or would like to create a new Google account for using ScHARe, you can create one here: bit.ly/3OeUogh

The questions marked with an asterisk (*) are required.

1. **What Google email address did you use to register for ScHARe?** *

Enter your answer

Sign up for free temporary billing

If not, please fill it out now here:

bit.ly/try-schare

Sign up for free temporary billing

If not, please fill it out now here:

bit.ly/try-schare

You will be:

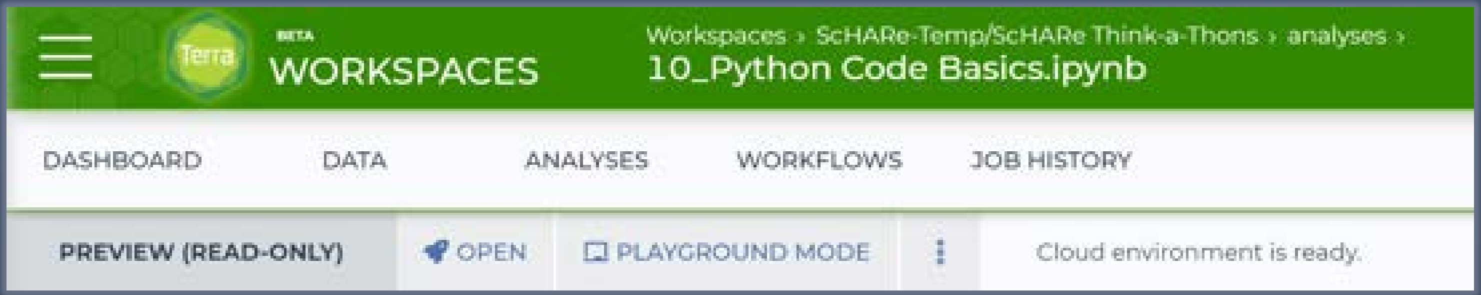
- **registered for ScHARe**
- **added to a free temporary billing project** that will allow you to run the event materials with your instructors

➢ You will be active on this billing project for the duration of the Think-a-Thon. If you want to access work-in-progress after this time, you will need to set up your own billing and copy your workspaces to it

In preparation for the Think-a-Thon

Let’s make sure that everyone:

1. has provided their email and has been registered for ScHARe
2. has created a Terra account
3. can access the tutorial we will be using today at: bit.ly/start-notebook
4. can run the tutorial in playground mode:


Terra
BETA
WORKSPACES
Workspaces › ScHARe-Temp/ScHARe Think-a-Thons › analyses ›
10_Python Code Basics.ipynb
DASHBOARD
DATA
ANALYSES
WORKFLOWS
JOB HISTORY
PREVIEW (READ-ONLY)
OPEN
PLAYGROUND MODE
Cloud environment is ready.

ScHARe

Science
collaborative for
Health disparities and
Artificial intelligence bias
Reduction

ScHARe

NIH National Institute on Minority Health and Health Disparities

+

NIH Office of Data Science Strategy

+

NIH National Institute of Nursing Research

Thank you

NIMHD

Dr. Eliseo Perez-Stable

ODSS

Dr. Susan Gregurick

NIH/OD

Dr. Larry Tabak

NINR

Dr. Shannon Zenk

NINR

Rebecca Hawes
Micheal Steele
John Grason

ORWH

OMH

NIMHD OCPL

Kelli Carrington
Thoko Kachipande
Corinne Baker

BioTeam

STRIDES

Terra

SIDEM

RLA

Broad Institute

CCDE Working Group

Deborah Duran
Luca Calzoni
Rebecca Hawes
Micheal Steele
Kelvin Choi
Paula Strassle
Michele Doose
Deborah Linares
Crystal Barksdale
Gneisha Dinwiddie
Jennifer Alvidrez
Matthew McAuliffe
Carolina Mendoza-Puccini
Simrann Sidhu
Tu Le

Outline

30’ Workshop setup

- Experience poll

5’ ScHARe and Terra overview

- Interest poll

5’ Why Python?

1h45’ Guest Expert: Cindy Sheffield (NIH/OD/ORS)

An introduction to Python for Data Science

5’ Python tutorials and resources

- Think-a-Thon poll

Experience poll

Please check your level of experience with the following:

	None	**Some**	**Proficient**	**Expert**
Python	□	□	□	□
R	□	□	□	□
Cloud computing	□	□	□	□
Terra	□	□	□	□
Health disparities research	□	□	□	□
Health outcomes research	□	□	□	□
Algorithmic bias mitigation	□	□	□	□

ScHARe

Part I
ScHARe and Terra Overview

ScHARe

Phase I

Population Science and SDoH Datasets Tutorials and Resources Think-a-Thons

ScHARe is a cloud-based population science data platform designed to accelerate research in health disparities, health and healthcare delivery outcomes, and artificial intelligence (AI) bias mitigation strategies

ScHARe aims to fill three critical gaps:

- Increase participation of **women & underrepresented populations with health disparities** in data science through data science skills training, cross-discipline mentoring, and multi-career level collaborating on research
- Leverage population science, SDoH, and behavioral Big Data and cloud computing tools to foster a **paradigm shift** in healthy disparity, and health and healthcare delivery outcomes research
- **Advance AI bias mitigation and ethical inquiry** by developing innovative strategies and securing diverse perspectives

ScHARe

nimhd.nih.gov/schare

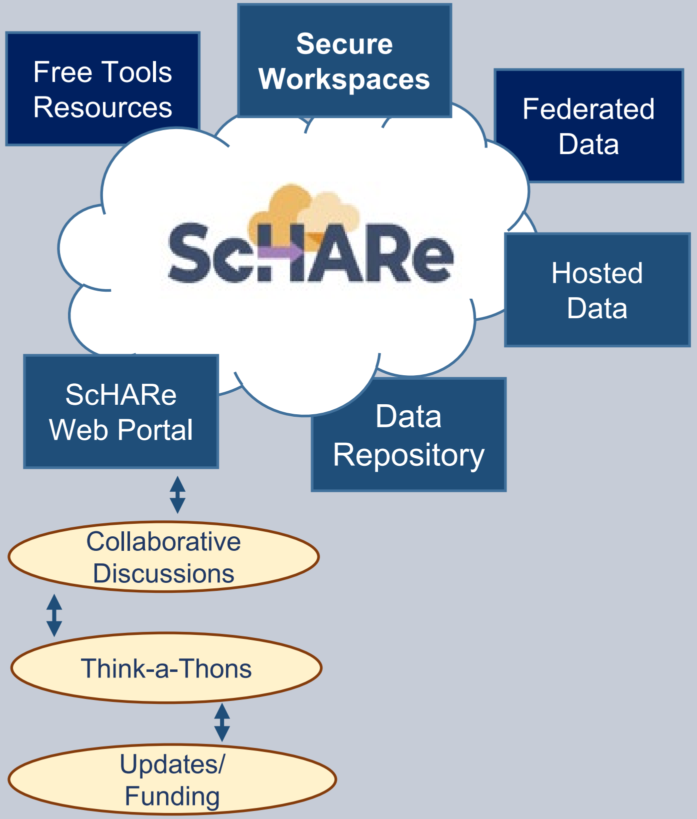
ScHARe Components

ScHARe co-localizes within the cloud:

- **Datasets** (including social determinants of health and social science data) relevant to minority health, health disparities, and health care outcomes research
- **Data repository** to comply with the required hosting, managing, and sharing of data from NIMHD- and NINR-funded research programs
- **Computational capabilities and secure, collaborative workspaces** for students and all career level researchers
- **Tools for collaboratively evaluating and mitigating biases** associated with datasets and algorithms utilized to inform healthcare and policy decisions

Frameworks: Google Platform, Terra, GitHub, NIMHD Web ScHARe Portal

Intramural & Extramural Resource



nimhd.nih.gov/schare

ScHARe Data Ecosystem

Researchers can access, link, analyze, and export **a wealth of datasets** within and across platforms relevant to research about health disparities, health care outcomes and bias mitigation, including:

- **Google Cloud Public Datasets:** publicly accessible, federated, de-identified datasets hosted by Google through the Google Cloud Public Dataset Program

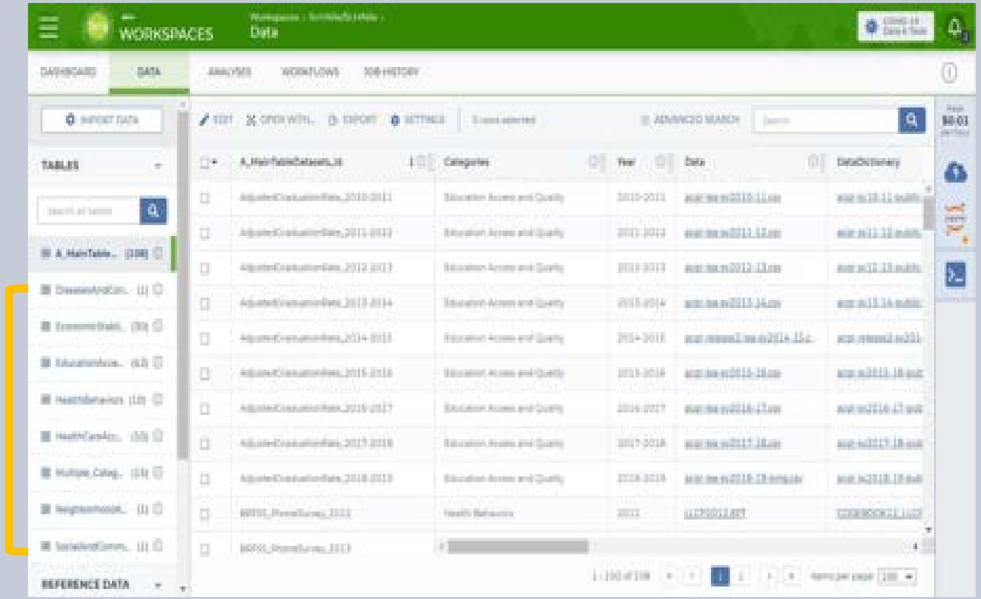
  Example: *American Community Survey (ACS)*

- **ScHARe Hosted Public Datasets:** publicly accessible, de-identified datasets hosted by ScHARe

  Example: *Behavioral Risk Factor Surveillance System (BRFSS)*

- **Funded Datasets on ScHARe:** publicly accessible and controlled-access, funded program/project datasets using Core Common Data Elements shared by NIH grantees and intramural investigators to comply with the NIH Data Sharing Policy

  Examples: *Jackson Heart Study (JHS); Extramural Grant Data; Intramural Project Data*



A_MainTable... (108)
DiseasesAndCon... (1)
EconomicStabi... (30)
EducationAcce... (43)
HealthBehaviors (12)
HealthCareAcc... (15)
Multiple_Categ... (13)
Neighborhood... (1)
SocialAndComm... (1)

A_MainTableDatasets_id	Categories	Year	Data	DataDictionary
AdjustedGraduationRate_2010-2011	Education Access and Quality	2010-2011	[illegible]	[illegible]
AdjustedGraduationRate_2011-2012	Education Access and Quality	2011-2012	[illegible]	[illegible]
AdjustedGraduationRate_2012-2013	Education Access and Quality	2012-2013	[illegible]	[illegible]
AdjustedGraduationRate_2013-2014	Education Access and Quality	2013-2014	[illegible]	[illegible]
AdjustedGraduationRate_2014-2015	Education Access and Quality	2014-2015	[illegible]	[illegible]
AdjustedGraduationRate_2015-2016	Education Access and Quality	2015-2016	[illegible]	[illegible]
AdjustedGraduationRate_2016-2017	Education Access and Quality	2016-2017	[illegible]	[illegible]
AdjustedGraduationRate_2017-2018	Education Access and Quality	2017-2018	[illegible]	[illegible]
AdjustedGraduationRate_2018-2019	Education Access and Quality	2018-2019	[illegible]	[illegible]
BRFSS_PhoneSurvey_2011	Health Behaviors	2011	LLCP2011.ASC	[illegible]
BRFSS_PhoneSurvey_2013				

1 - 100 of 108

On ScHARe, datasets are categorized by content based on the CDC **Social Determinants of Health categories**:

1. Economic Stability
2. Education Access and Quality
3. Health Care Access and Quality
4. Neighborhood and Built Environment
5. Social and Community Context

with the addition of:

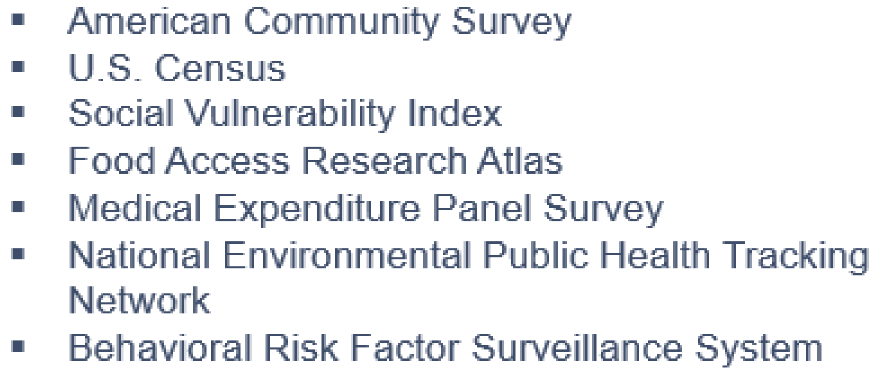
- **Health Behaviors**
- **Diseases and Conditions**

Users will be able to **map and link** across datasets

Access to Population Science datasets

ScHARe Data Ecosystem will offer access to **300+ datasets**, including:

- Google Cloud Public Datasets
- ScHARe Hosted Public Datasets:



- American Community Survey
- U.S. Census
- Social Vulnerability Index
- Food Access Research Atlas
- Medical Expenditure Panel Survey
- National Environmental Public Health Tracking Network
- Behavioral Risk Factor Surveillance System

- **Coming Soon:** Repository for Funded Datasets on ScHARe, in compliance with NIH Data Sharing Policy

Cloud computing strategies

ScHARe

- Uses **workflows** in Workflow Description Language (**WDL**), a language easy for humans to read, for batch processing data
- **Python and R**, including most commonly used libraries
- Enables **customization** of computing environments to ensure everyone in your group is using the same software
- **Big Query** and **Tensorflow** access for advanced machine learning
- Enables researchers to create interactive **Jupyter notebooks** (documents that contain live code) and share data, analyses and results with their collaborators in real time
- For novice users, integration with **SAS** is planned

AI bias mitigation strategies

- Widespread use of AI raises a number of ethical, moral, and legal issues – likely not to go away
- Algorithms often are “black boxes”
- **Biases can result from:**
  - **social/cultural context not considered**
  - **design limitations**
  - **data missingness and quality problems**
  - **algorithm development and model training**
  - **Implementation**
- If not rectified, biases may result in decisions that lead to discrimination, unequitable healthcare, and/or health disparities
- **Lack of diverse perspectives:** populations with health disparities are underrepresented in data science
- **Guidelines** and recommendations emerging from HHS, NIST, White House, etc.

ScHARe

Critical thinking can rectify AI biases

ScHARe was created to:

- foster participation of **populations with health disparities in data science**
- promote the collaborative identification of **bias mitigation strategies** across the continuum
- create a **culture of ethical inquiry** and critical thinking whenever AI is utilized
- build **community confidence** in implementation approaches
- focus on **implementation of AI bias** guidelines and recommendations

SciHARe

Phase II

(in process)

Repository and Data Ecosystem

ScHARe Data Repository

CORE COMMON DATA ELEMENTS

NOVEL CDE FOCUSED REPOSITORY TO FOSTER INTEROPERABILITY

COMPLY WITH DATA SHARING POLICY - HOST PROJECT DATA

DATA ECOSYSTEM

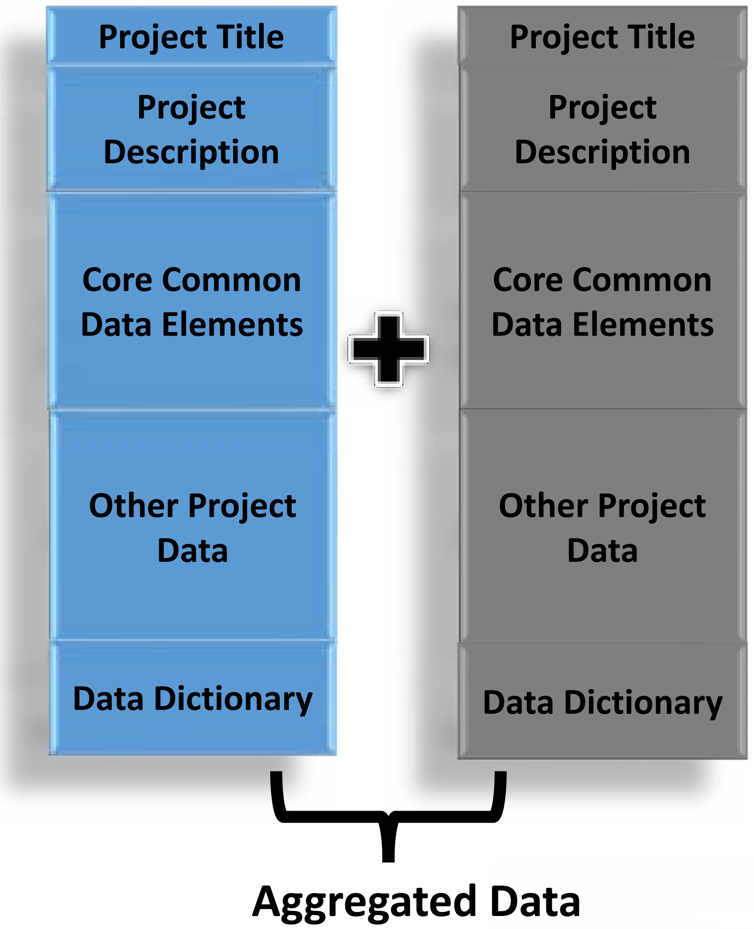
- Map across datasets
- Map across platforms

UPCOMING

ScHARe

Core Common Data Elements
Intramural and Extramural Project Repository

- Complies with **NIH Data Sharing Policy**
- Fosters dataset sharing and interoperability by using or mapping to **Core Common Data Elements**
- Provides resources for **intramural researchers** to work in a secure workspace and host data
- Centralizes **aggregated datasets** for repeat use

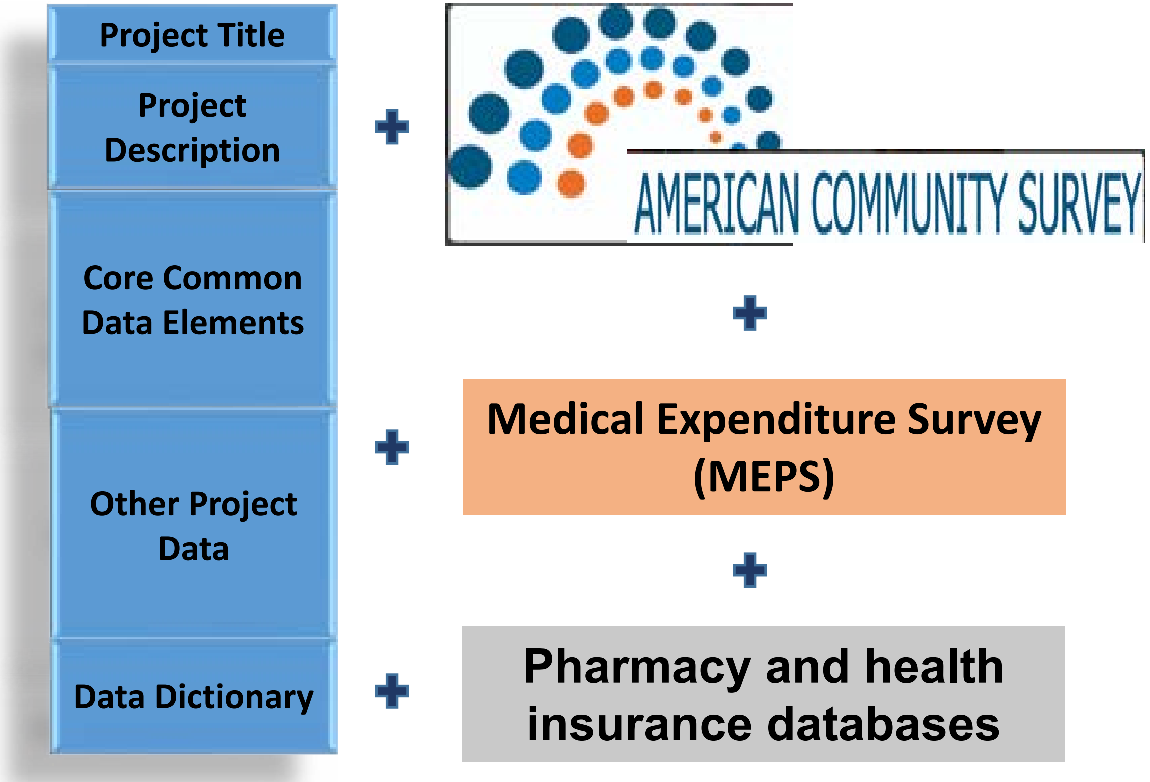


smoking
SDoH
geographic
Community
Interventions
Sleep
cancer
smoking
Mental health
geographic
Applied Clinical
Adherence
Mental health
geographic
Decision making
SDoH
Health Care Delivery
Diabetes
Decision making
SDOH
cancer

UPCOMING FALL

ScHARe

Project & federated dataset mapping



Project Title

Project Description

Core Common Data Elements

Other Project Data

Data Dictionary

+ AMERICAN COMMUNITY SURVEY

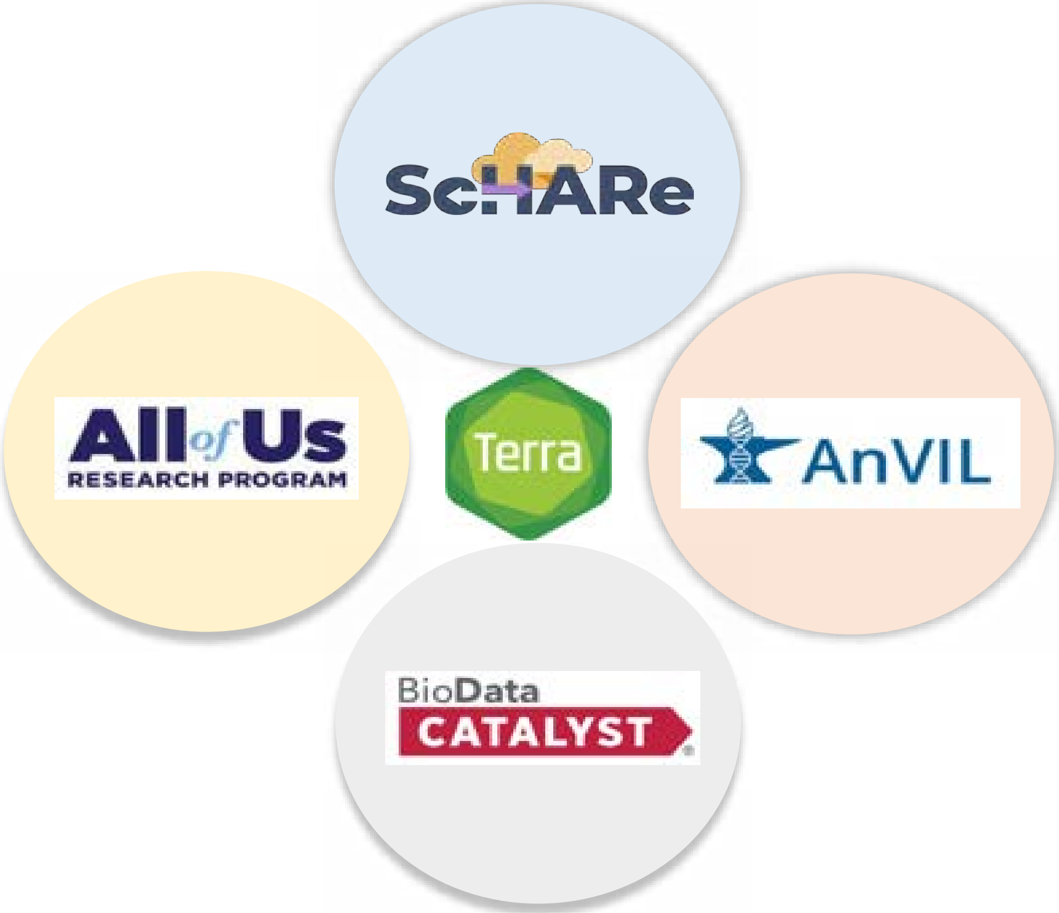
+

+ Medical Expenditure Survey (MEPS)

+

+ Pharmacy and health insurance databases

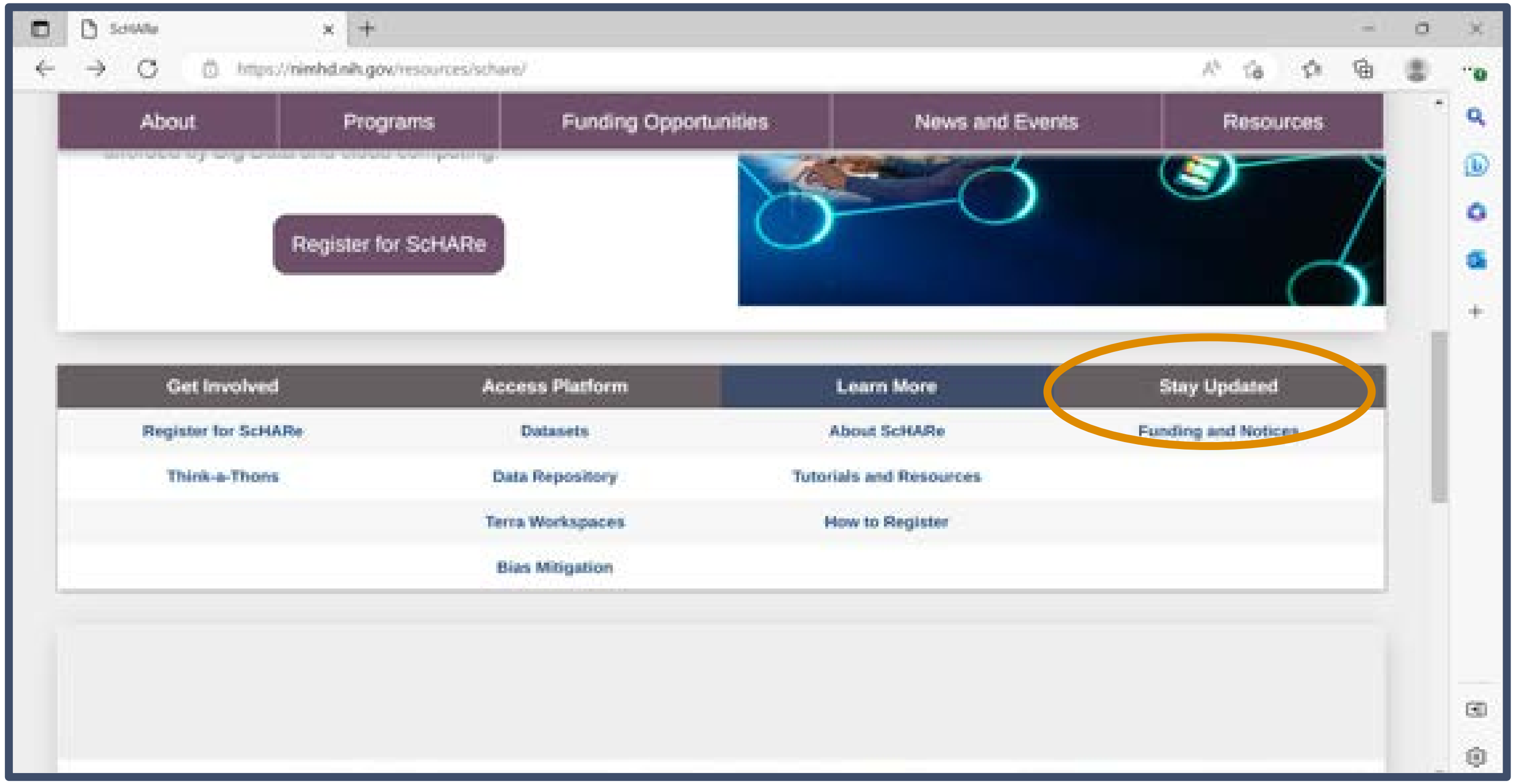
Mapping across cloud platforms



UPCOMING

Two ways to sign up for ScHARe news

Scannable from your screen!



Get Involved	Access Platform	Learn More	Stay Updated
Register for ScHARe	Datasets	About ScHARe	Funding and Notices
Think-a-Thons	Data Repository	Tutorials and Resources	
	Terra Workspaces	How to Register	
	Bias Mitigation		

nimhd.nih.gov/schare

Interest poll

I am interested in (check all that apply):

□ Learning about Health Disparities and Health Outcomes research to apply my data science skills

□ Conducting my own research using AI/cloud computing and publishing papers

□ Connecting with new collaborators to conduct research using AI/cloud computing and publish papers

□ Learning to use AI tools and cloud computing to gain new skills for research using Big Data

□ Learning cloud computing resources to implement my own cloud

□ Developing bias mitigation and ethical AI strategies

□ Other

ScHARe Think-a-Thons (TaT)

- Monthly sessions (2 1/2 hours)
- Instructional/interactive
- Designed for new and experienced users
- Research & analytic teams to:
  - Conduct health disparities, health outcomes, bias mitigation research
  - Analyze/create tools for bias mitigation
- Publications from research team collaboration
- Networking
- Mentoring and coaching
- Focus:

✓ **Instructional**

✓ **Collaboration research teams**

✓ **Bias mitigation**

ScHARe

Think-a-Thon

Artificial Intelligence and Cloud Computing Basics

Terra: Datasets and Analytics

Register:

bit.ly/think-a-thons

SciHARe

Part II

Why Python?

What is Python?

Python is a **computer programming language** used in data science to:

- manipulate and analyze data and conduct statistical calculations
- create data visualizations
- build machine learning algorithms

Python's **data science libraries** are powerful. Examples include:

- **Numpy** - for linear algebra and high-level mathematical functions
- **Pandas** - for handling data structures and manipulating tables
- **SciPy** - for data science tasks like interpolation and signal processing
- **Scikit-learn** - a machine learning library that is useful for classification, regression, and clustering algorithms
- **PyBrain** - for machine learning tasks and to test and compare algorithms

Sources
www.quanthub.com/python-for-data-science/
coursera.org

What is R?

R is a **programming language** for statistical computing and graphics

It is used by data miners, bioinformaticians and statisticians for data analysis

Users have created **packages** to augment its functions

Third-party **graphical user interfaces** are also available, such as Rstudio

Terra supports **both Python and R**

Source
en.wikipedia.org

Why Python?

According to SlashData:

- there are 8.2 million Python users
- **69%** of machine learning developers and data scientists **use Python (vs. 24%** of them **using R)**

Source
stackify.com/learn-python-tutorials/

How to learn Python

How long does it take to learn Python?

It can take **2 to 5 months**, but you can write your first short program in **minutes**

Can you learn Python with no experience?

Python is the **perfect** programming language **for people without any coding experience**, as it has a simple syntax and is very accessible to beginners

Unfamiliar terminology may be a barrier, which today's workshop will hopefully help you overcome

Links to additional **free learning resources** will be provided at the end

SciHARe

Part III

An introduction to Python for Data Science

SciHARe

Guest expert

Cindy Sheffield

NIH/OD/ORS

About Cindy

Cindy is Data Services Librarian at the NIH Library. She began her library career at the Johns Hopkins Medical Institutions with a focus on Evidenced Based Medicine. She progressed within the Welch Medical Library, leaving Hopkins as the Associate Director of Education Services. Cindy has worked at several federal agencies including the Department of Homeland Security, the Department of Defense, and the Department of Health and Human Services. Within DHHS she was worked for both the National Institutes of Health and the Federal Drug Administration.

Her focus has always been on using key resources to identify the best evidence, and then to organize and manage that evidence in a way that makes sense for users. At the NIH she works with various user groups to support literature research and data science. She is the Outreach Librarian for the NIH Clinical Centers, Pain and Palliative Care Team, the Eunice Kennedy Shriver, National Institute of Child and Human Development, the Administration for Children and Families, and the Office of the National Coordinator for Health Information Technology.

National Institutes of Health
Office of Management

ScHARe Think-a-Thon: An Introduction to Python for Data Science

Cindy Sheffield, NIH Library
Data Services

Training Objectives

Why learn Python?

What to know about reading, writing and running Python code?

Popular IDEs for reading, writing and running Python code:

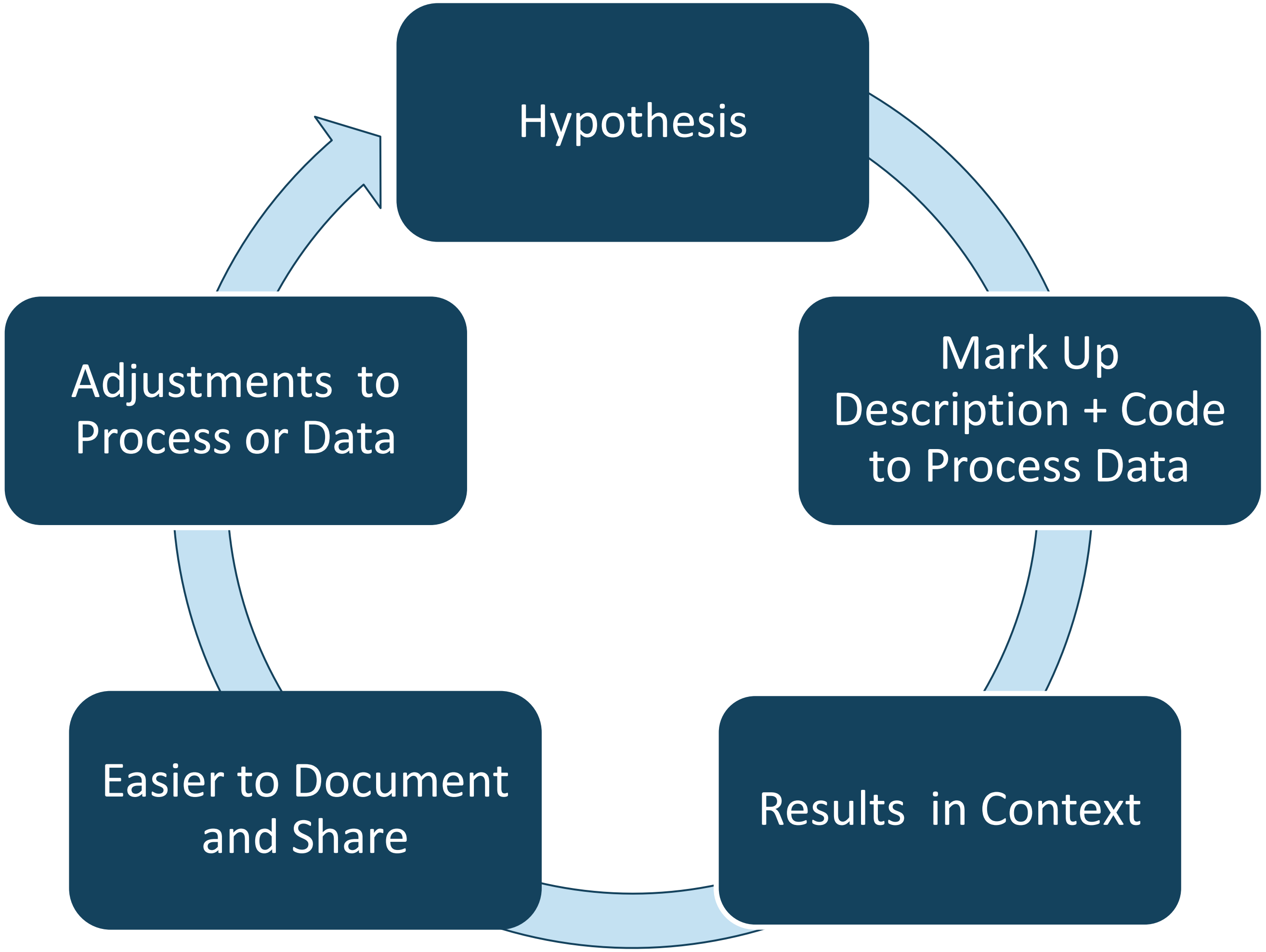
- Google Colab
- Spyder
- Jupyter Notebook
- Jupyter Lab

Why Learn Python?

Why Learn Python?

- **User friendly** - easy to read and easy to learn.
- **Efficient** - facilitates data management, data analysis, process documentation, and visualizations via an ecosystem of libraries.
- **Productivity** – clear syntax, object-oriented design, and code reusability enable developers to write and adapt code efficiently.
- **Dynamic** – Design cycles tend to be short as code can be written and tested
- **Open Source** – Python software and its IDEs are distributed freely for most part. There are inexpensive IDEs. Cost is not a barrier to learning. It simply takes time and effort.

Research Development


Hypothesis
Mark Up Description + Code to Process Data
Results in Context
Easier to Document and Share
Adjustments to Process or Data

What you need to know about reading, writing, and running Python?

What you need to know to read/write/run Python?

- Syntax
- Variables
- Functions
- Data Types
- Conditionals
- Libraries

Syntax

#Comments

Class names = A to Z
(blueprint of an object)

(tuples use parenthesis)

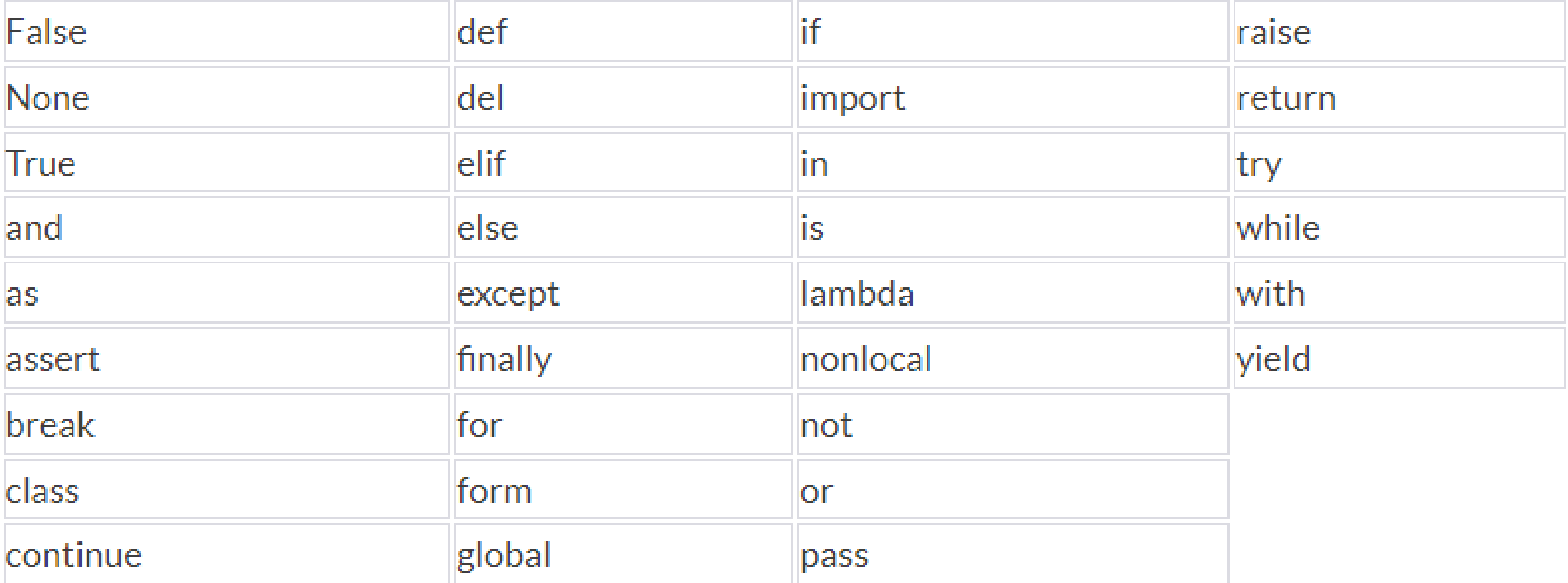
identifiers = a to z

[lists use brackets]

{dictionaries use squiggly brackets}

loops and **if statements** require a colon:
and proper indentation
4 spaces, no less, no more!

Syntax: Reserve Words



False	def	if	raise
None	del	import	return
True	elif	in	try
and	else	is	while
as	except	lambda	with
assert	finally	nonlocal	yield
break	for	not	
class	form	or	
continue	global	pass	

https://flexiple.com/python/python-reserved-words/#section1

Variables

a = “A string of characters”

pi=3.1415

y = (1:5)

x = 2

c = a+b * (3.14)

myfile = “/myfile.csv”

Fruits=(‘apples’, ‘bananas’, ‘cantaloupe’, ‘dates’)

Functions

```
greet(“New Python Coders”)
```

```
def greet (named):
    print(“Hello “ + named + ” !”)
```

```
Hello New Python Coders!
```

In this example ‘greet’ is the Function name and ‘named’ is the Parameter variable, the function headers always begins with ’def’ and ends with a colon.

Python Built-in Functions

– https://www.pythoncheatsheet.org/cheatsheet/built-in-functions

Data Types

Tuples – Sequence Type, parentheses

MyTuple = (parentheses, exponents, multiplication/division, addition/subtraction)

Lists – Sequence Type, square brackets

names = [“Annie”, “Betty”, “Cindy”]
cities = [“Albany”, “Baltimore”, “Cincinnati”]
items = [“apples”, “bananas”, “cantaloupe”]

Dictionaries – Mapping Type, squiggly brackets

MyDictionary = {“names”: “Annie”, “items”: “apples”}

Conditionals

Loops

```
fruit_list=['apples', 'bananas', 'cantaloupe']

for i in range(0,3):
    print(fruit_list[i])
```

```
apples
bananas
cantaloupe
```

If Statements

```
x = 4; y =6

if x < y:
    print(x, ‘is less than’, y)
else:
    print(x, ‘is not less than’, y)
```

```
    4 is less than 6
```

Libraries

Pandas Library

```
import pandas as pd
```

function | library | assign | short for pandas

Enables code:

```
df_alphasong = pd.read_csv(“/alphasong.csv”)
```

Example data frame

alphasong.csv =

NAMES	CITIES	ITEMS
Annie	Albany	Apples
Betty	Baltimore	Bananas
Cindy	Cincinnati	cantaloupe

Libraries

Python Libraries can be found at PyPI.org.

PyPI is the Python Package Index: a repository of software for Python programming

- Statistical Analysis and Visualizations Libraries include:
  - Pandas (data frames)
  - NumPy (math functions for arrays)
  - Statsmodels
  - TensorFlow
  - Scikit-learning
  - Mathplotlib (np + Visualizations)
  - Plotly
  - Seaborn (Visualizations)

```
import pandas as pd
import numpy as np
import seaborn as sb
```

What you need to know to read/write/run Python?

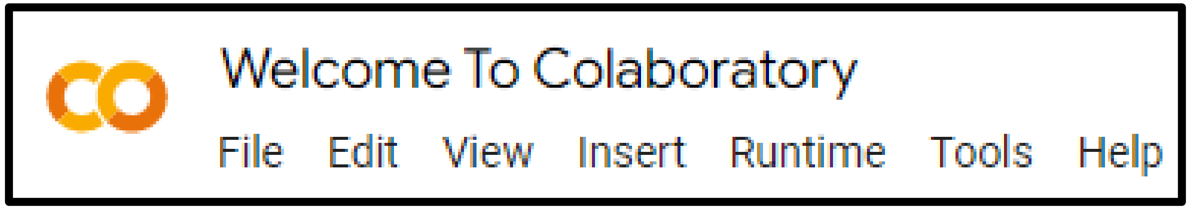
- Syntax
- Variables
- Functions
- Data types
- Conditionals
- Libraries

IDE for Python Code

Popular IDEs for Python

Popular IDEs for reading, writing and running Python code:

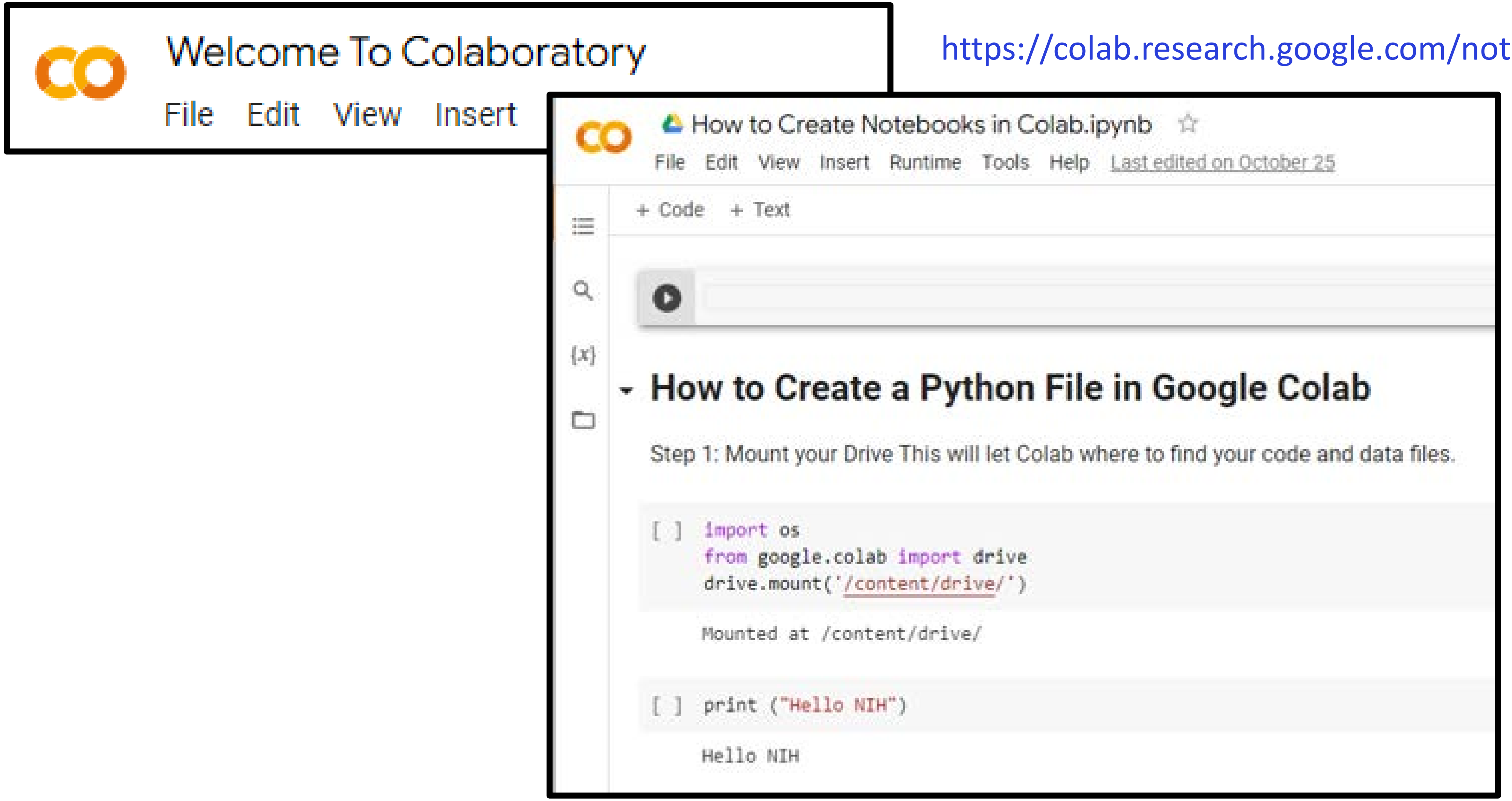
- Google Colab
- Spyder
- Jupyter Notebook
- JupyterLab



Welcome To Colaboratory

Google Colab

https://colab.research.google.com/notebooks/welcome.ipynb



Welcome To Colaboratory
File Edit View Insert
How to Create Notebooks in Colab.ipynb
File Edit View Insert Runtime Tools Help Last edited on October 25
+ Code + Text
{x}

How to Create a Python File in Google Colab

Step 1: Mount your Drive This will let Colab where to find your code and data files.

```
[ ] import os
    from google.colab import drive
    drive.mount('/content/drive/')
```

```
Mounted at /content/drive/
```

```
[ ] print ("Hello NIH")
```

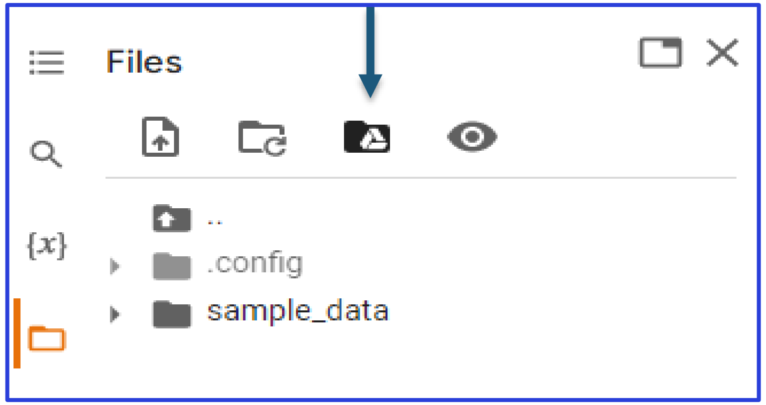
```
Hello NIH
```

Google Colaboratory Setup

- Create a new Google Account for Colab.
- Go: https://colab.research.google.com/notebooks/welcome.ipynb
- Click on Getting Started for a quick overview of Colab
- Use File / Notebook to find online guides.
- Connect to files Google Drive by either:

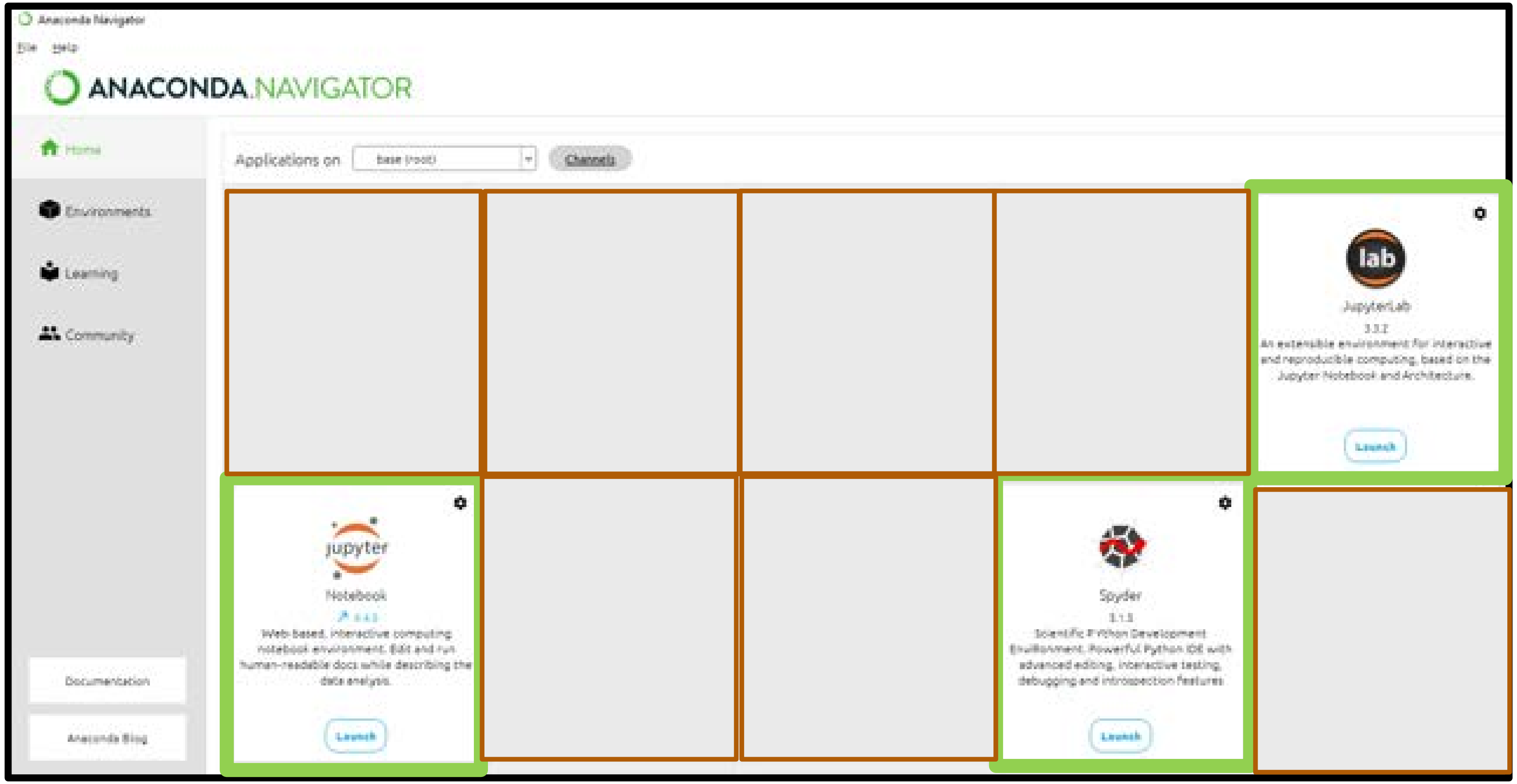
Type text below or select Drive icon under Files

```
import os
from google.colab import drive
drive.mount('/content/drive/')
```



Anaconda

- Jupyter Notebook
- Spyder
- JupyterLab



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

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

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

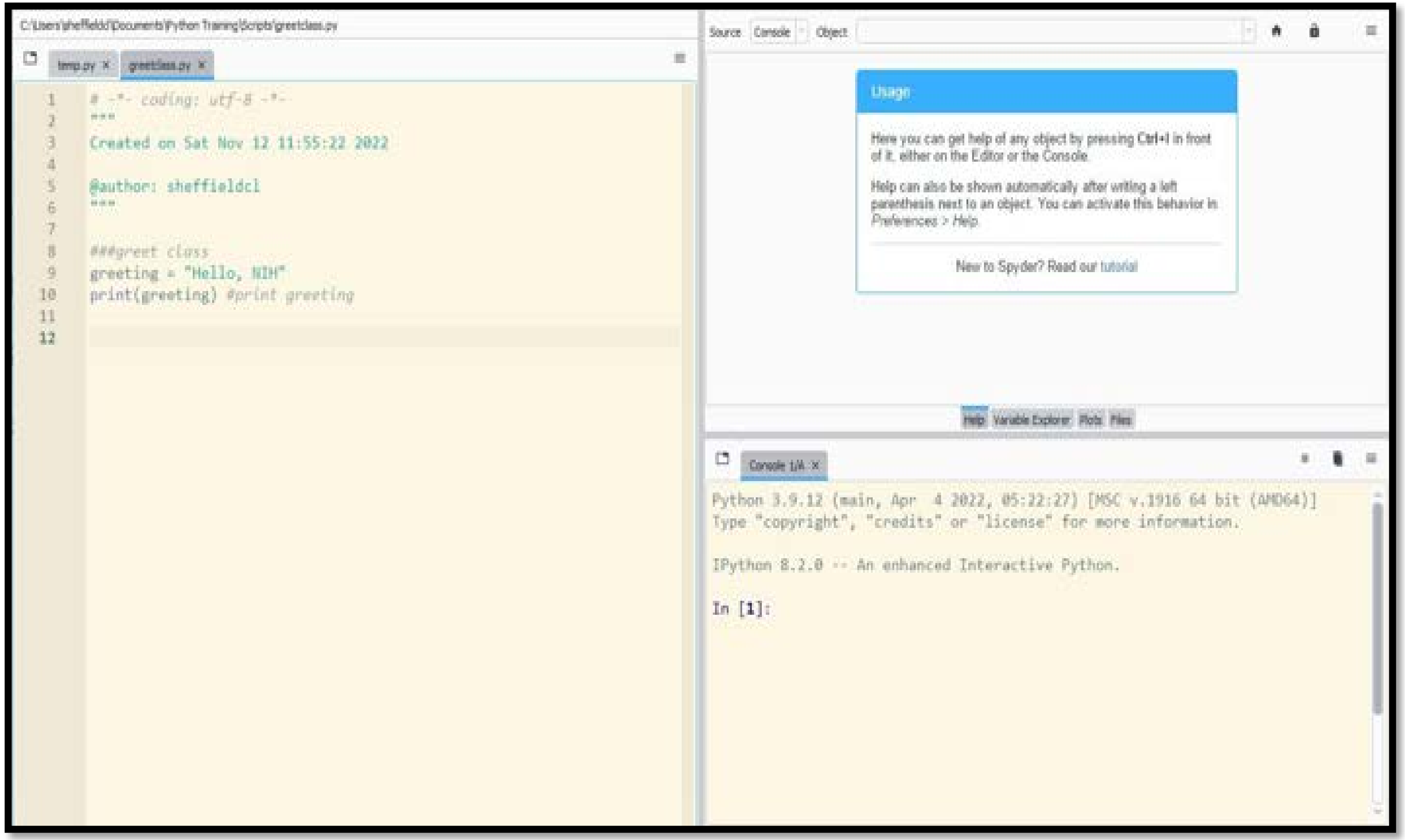
Spyder

Install Anaconda
Launch Spyder
File/Open/*open script*

OR

File/Create/*create new*

- Ctrl-L will give you help.
- There is a tutorial.



```
# -*- coding: utf-8 -*-
"""
Created on Sat Nov 12 11:55:22 2022

@author: sheffieldcl
"""

###greet class
greeting = "Hello, NIH"
print(greeting) #print greeting
```

Usage

Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

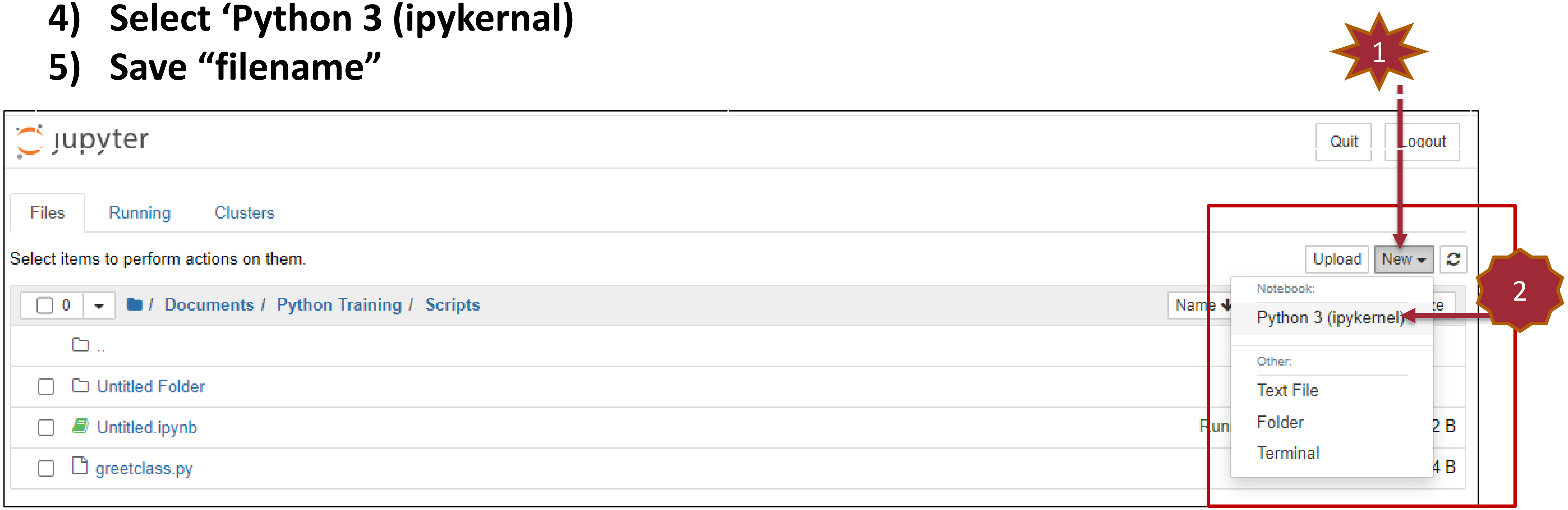
New to Spyder? Read our tutorial

```
Python 3.9.12 (main, Apr  4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.2.0 -- An enhanced Interactive Python.

In [1]:
```

https://docs.spyder-ide.org/current/quickstart.html

Jupyter Notebook

1) Open Anaconda,
2) Launch Jupyter notebook
3) Select ‘New’
4) Select ‘Python 3 (ipykernal)
5) Save “filename”



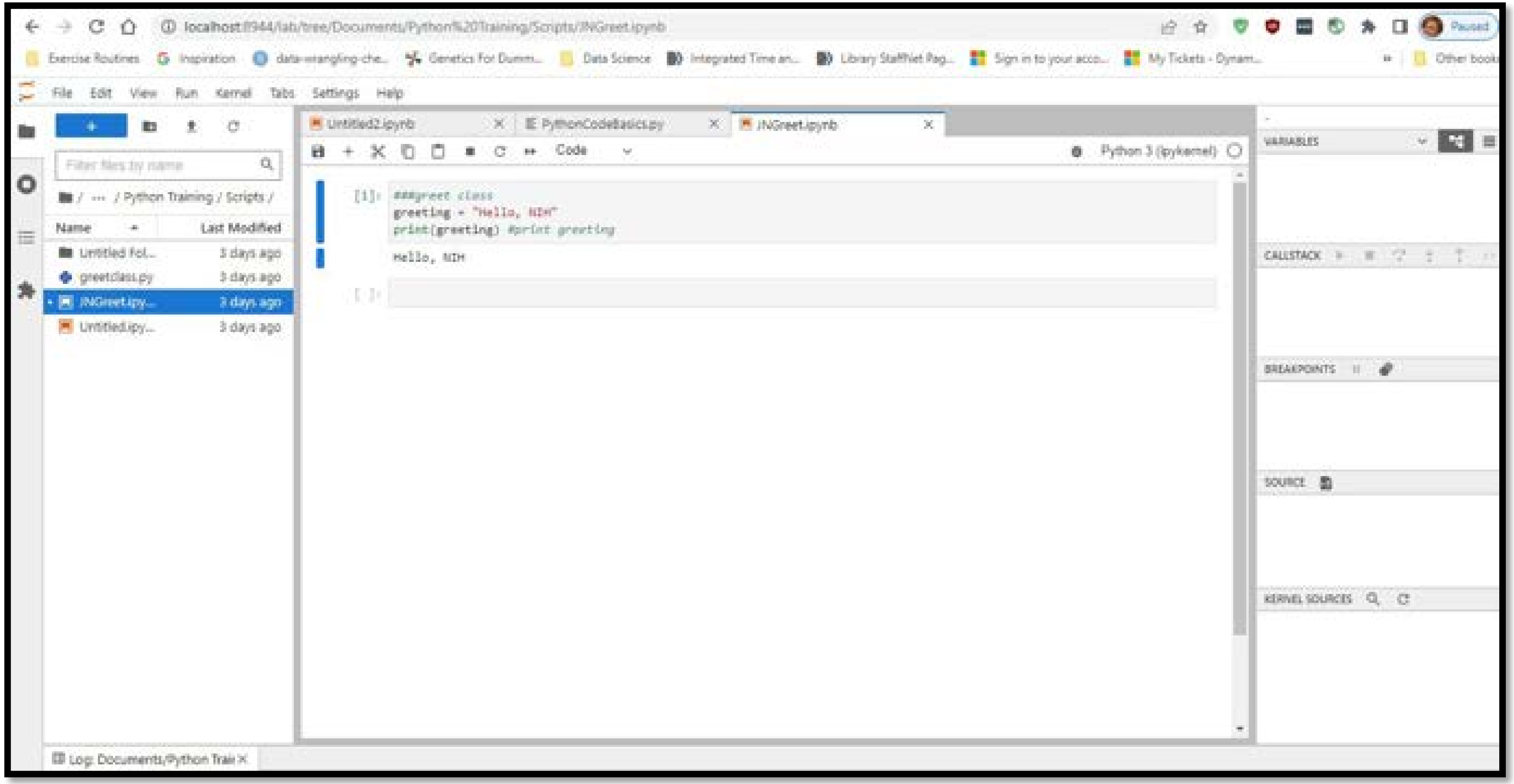
Select items to perform actions on them.

Jupyter Laboratory

- Install Anaconda
- Launch: JupyterLab
- File/Open/*open script*

OR

- File/Create/ *create new*



localhost:8944/lab/tree/Documents/Python%20Training/Scripts/JNGreet.ipynb

Name	Last Modified
Untitled Fol...	3 days ago
greetclass.py	3 days ago
JNGreet.ipy...	3 days ago
Untitled.ipy...	3 days ago

```
[1]: ###greet class
     greeting = "Hello, NIH"
     print(greeting) #print greeting

     Hello, NIH

[ ]:
```

IDE Attributes

- **Console:** - Where code runs
- **Project Files:** - Code and data files
- **Variables:** - One Identifier to assign something to a variable
- **Data View:** - View of the data frame being processed
- **Plots: -** View of the graph being generated
- **History: -** Code history processes in active memory
- **Autocomplete**: - Python feature to help with coding
- **Debugging: -** Python feature to help with finding errors
- **Markdown*:* -** Code to add text around the code

Google Colab IDE Attributes

1. **Console:** Within Colab there isn't a separate console, but the instead the active code and related error messages occur within each cell.
2. **Project Files:** To see where your Notebook is stored use the folder icon on the left to navigate to your MyDrive / Colab Notebooks. "PythonCodeBascis".
3. **Variables:** (Show current variables) We have a notebook here with code, but none of this code is NOT in our active memory, because the only code that has been run is the cell to initialize our PythonCodeBasics notebook and the little bit of code we just recreated. Let's run the code in a few more of these cells.
4. **Data View:** Simply double left click on the data file to view the table on the right side of the Colab screen.
5. **Libraries:** All of the Python libraries are available to Colab so it is a matter of importing each library you want to use and assigning that library to a shortcut or nickname such as import pandas as pd. You will need to have knowledge about the libraries you want to use and why you want to use them.
6. **Plots:** Once you run the code and import the associated data for plots, the plots will appear directly within Colob. There is also a 'View output full screen' feature using the menu to the right.
7. **History:** To view the History of code run during a session select the View menu, and then 'Execute code history'.
8. **Autocomplete**: Note the autocomplete feature will try and complete and function such as 'print' or argument that appears within your active Variables list.
9. **Debugging:** If you have an error in your code, an error message will appear to help with debugging.
10. **Comments and Markdown:** Comments can be added using 3 hash marks on any line of code. Markdown text can be added using the + Text tool.

Spyder IDE Attributes

1. **Console:** The console is immediately viewable in the lower right pane. If I highlight my first line of code in the script and run it using the F9 key, it will run the code and the results from the code in the Console. If I highlight and run the next line, “2*3” and use the F9 key will give me that code and result as well.
2. **Project Files:** We can get to Project Files by using File / Open or Open Recent to find the .py file you choose to use.
3. **Variables:** Can be viewed in the upper right pane using the “Variable Explorer” tab. Note each data type will be a different color.
4. **Data View:** Using the Files tab in the top right pane, identify your data set, click on it to see the content. If you’d like to see the content in a spreadsheet you can right click on the file and use the dropdown menu to select “Open externally”. It will open in Excel outside of Spyder.
5. **Libraries:** Libraries are need for dataframe, calculation and visuals. Import needed libraries.
6. **Plots: Can be viewed in the upper right pane, using the Plots tab.**
7. **History:** The History can be viewed in the lower right pane, using the History tab.
8. **Autocomplete**: Similar to Colab Sypder will autocomplete functions and arguments that are in active memory.
9. **Debugging:** There is a Debug menu at on the top menu bar of Spyder
10. **Comments and Markdown:** Comments can be noted using triple single quotes for large areas of text and the # sign for line comments. Markdown is not an option in Spyder.

Jupyter Notebooks IDE Attributes

1. **Console:** The format is very similar to Google Colab, with the code and related error messages being generated in the main screen.
2. **Project Files:** Can be open via the File dropdown menu. New project files can be created using that option from the dropdown menu.
3. **Variables:** With pandas running, in one of the empty cells run “%whos”
4. **Data View:** To view data frames in one of the empty cells use the function display(<data frame name>)
   Example: display(iris) *may need to specify file path*
5. **Libraries:** Need to be familiar with libraries and import them and assign them short names.
6. **Plots:** Plots will appear directly in the cells with the code to generate them.
7. **History:** In one of the empty cells run “%history”
8. **Autocomplete**: Start typing the first few letters of a function or argument and then hit the Tab key, and Jupyter Notebooks will autocomplete the rest of the code.
9. **Debugging:** Error messages usually show within the cell. Alt + Shift + Enter is another way to see error codes.
10. **Comments and Markdown:** Comments can be created with quotes and # signs. Markdown is available from dropdown menu immediately under the Widgets option on the top menu bar.

Jupyter Labs IDE Attributes

1. Jupyter Labs is very similar to Jupyter Notebooks.
2. The main difference is that you can have multiple tabs open with notebooks or .csv files in this IDE. Multiple consoles can be run as well.   This is helpful if you are trying to run different sets of code with different applications.

Test Your Knowledge

Log on to one of the IDEs.

Type the follow code:

```
name = “#”  #Add your name
str = “ thinks Python is ” #note spaces before and after clause!
descpt = “#” #Add descriptor to indicate how you feel about Python
print (name + str +descpt)
```

Learning Resources

- Google's Python Class
  - https://developers.google.com/edu/python/
- Learn Python – Free Python Courses for Beginners
  - https://www.freecodecamp.org/news/learn-python-free-python-courses-for-beginners/
- Books
  - Learning Python, O'Reilly
  - Head First Python, Paul Barry
  - Python Crash Course, Eric Matthes

References

- **BioPython: freely available Python tools for computational molecular biology and bioinformatics**
  https://academic.oup.com/bioinformatics/article/25/11/1422/330687?login=true
- **Design of Experiments (DOE) with python**
  https://medium.com/mlearning-ai/design-of-experiments-doe-with-python-be88f5c013f5
- **Introduction to Jupyter Notebook | Jupyter Notebook Tutorial**
  https://youtu.be/1A7tea9LSEk
- **JupyterLab Tutorial for Everyone**
  https://youtu.be/mspsHIk_qUQ

NIH National Institutes of Health
Office of Management

SciHARe

Part IV
Python Tutorials and Resources

Python resources

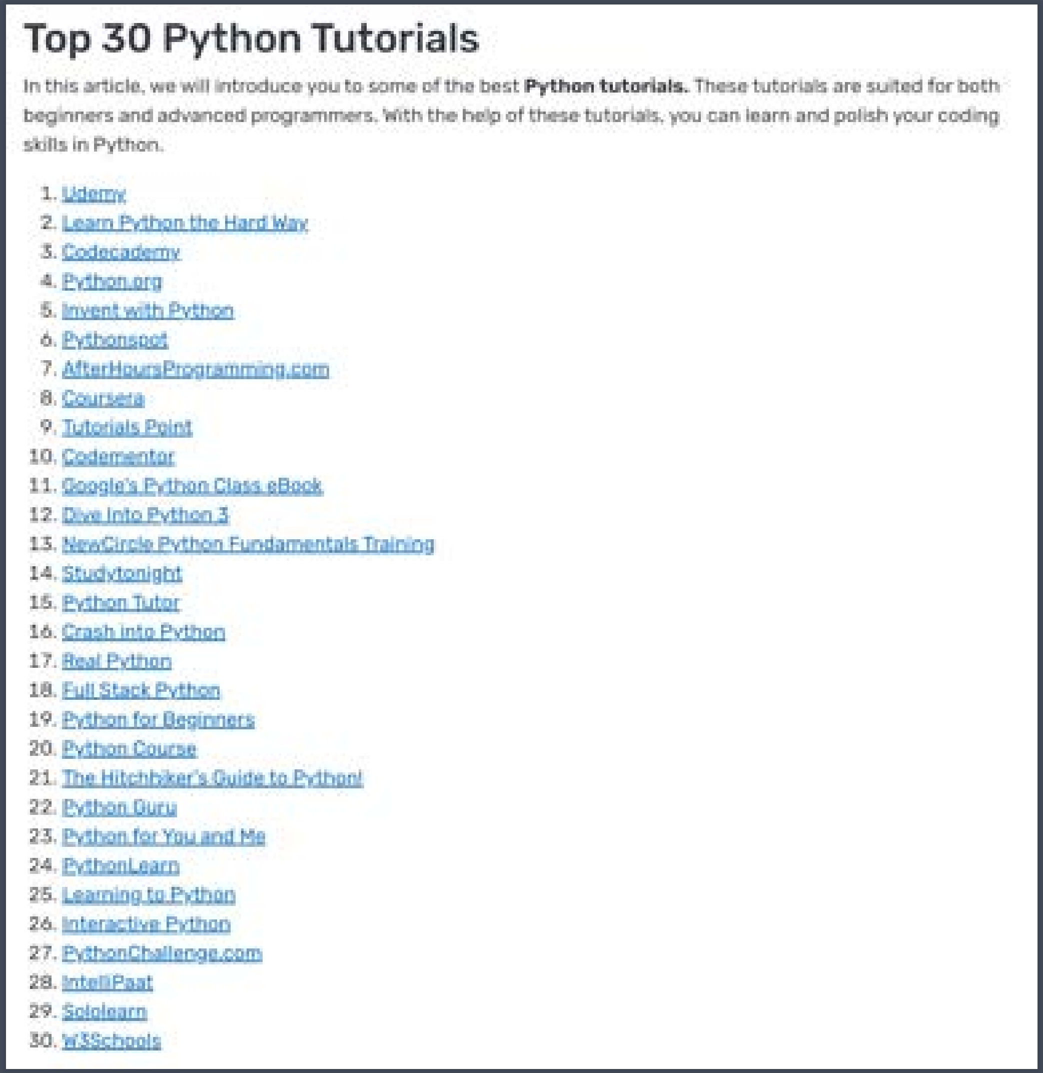
You can take advantage of the dozens of “**Python for data science” online tutorials** for beginners and advanced programmers listed here:

- Stackify - 30+ Tutorials to Learn Python
- FreeCodeCamp - Code Class for Beginners
- Harvard – Free Python Course
- Coursera – Free and Paid Python Courses
- LearnPython – Free Interactive Python Tutorials
- BestColleges – 10 Places to Learn Python for Free

Python resources

Stackify

30+ Tutorials to Learn Python



Top 30 Python Tutorials

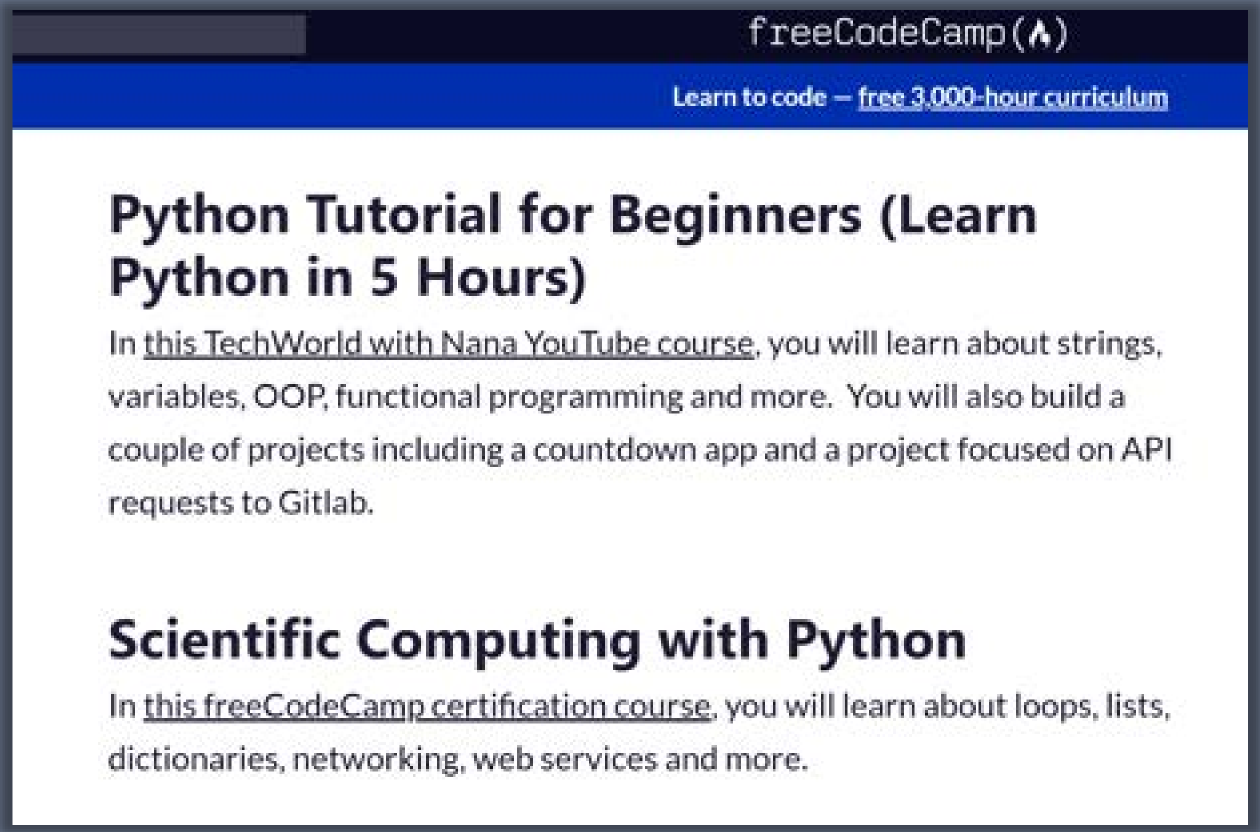
In this article, we will introduce you to some of the best **Python tutorials.** These tutorials are suited for both beginners and advanced programmers. With the help of these tutorials, you can learn and polish your coding skills in Python.

1. Udemy
2. Learn Python the Hard Way
3. Codecademy
4. Python.org
5. Invent with Python
6. Pythonspot
7. AfterHoursProgramming.com
8. Coursera
9. Tutorials Point
10. Codementor
11. Google's Python Class eBook
12. Dive Into Python 3
13. NewCircle Python Fundamentals Training
14. Studytonight
15. Python Tutor
16. Crash into Python
17. Real Python
18. Full Stack Python
19. Python for Beginners
20. Python Course
21. The Hitchhiker's Guide to Python!
22. Python Guru
23. Python for You and Me
24. PythonLearn
25. Learning to Python
26. Interactive Python
27. PythonChallenge.com
28. IntelliPaat
29. Sololearn
30. W3Schools

Python resources

[FreeCodeCamp]

Code Class for Beginners



freeCodeCamp(🔥)

Learn to code — free 3,000-hour curriculum

Python Tutorial for Beginners (Learn Python in 5 Hours)

In this TechWorld with Nana YouTube course, you will learn about strings, variables, OOP, functional programming and more. You will also build a couple of projects including a countdown app and a project focused on API requests to Gitlab.

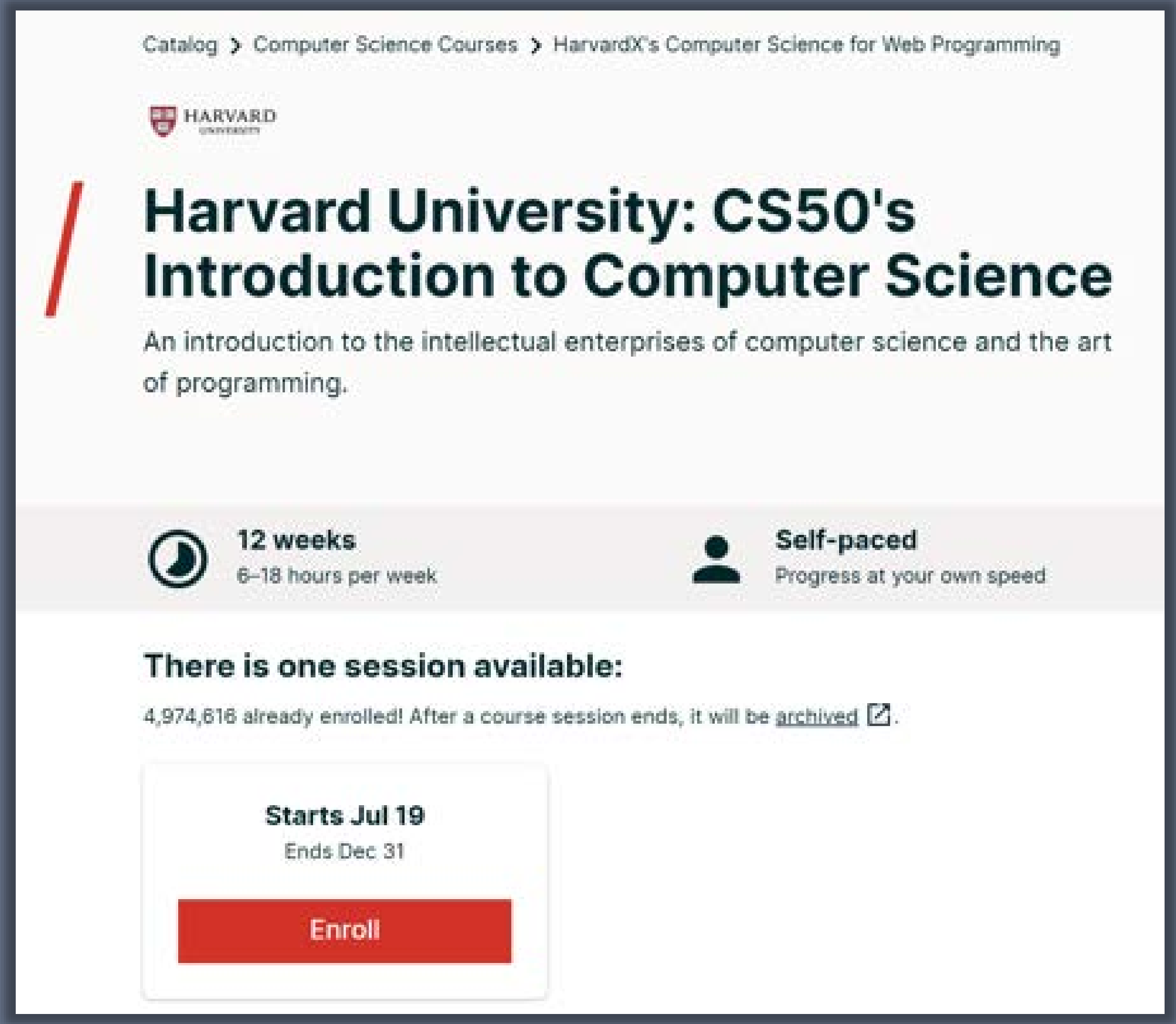
Scientific Computing with Python

In this freeCodeCamp certification course, you will learn about loops, lists, dictionaries, networking, web services and more.

Python resources

Harvard

Free Python Course



Catalog > Computer Science Courses > HarvardX's Computer Science for Web Programming

HARVARD UNIVERSITY

Harvard University: CS50's Introduction to Computer Science

An introduction to the intellectual enterprises of computer science and the art of programming.

12 weeks
6–18 hours per week

Self-paced
Progress at your own speed

There is one session available:

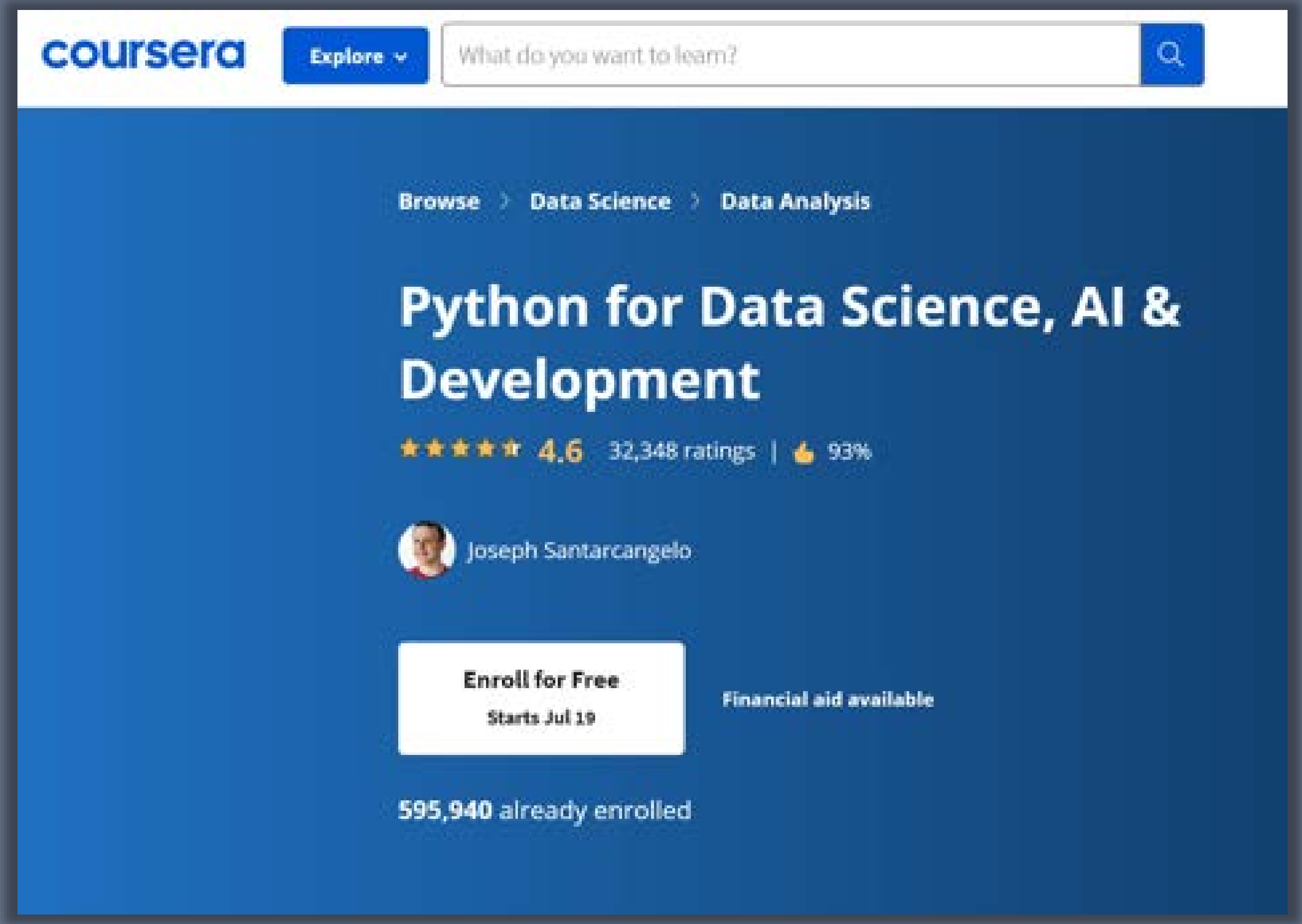
4,974,616 already enrolled! After a course session ends, it will be archived.

Starts Jul 19
Ends Dec 31

Python resources

Coursera

Free and Paid Python Courses



Browse › Data Science › Data Analysis

Python for Data Science, AI & Development

★★★★★ 4.6 32,348 ratings | 👍 93%

Joseph Santarcangelo

Enroll for Free
Starts Jul 19

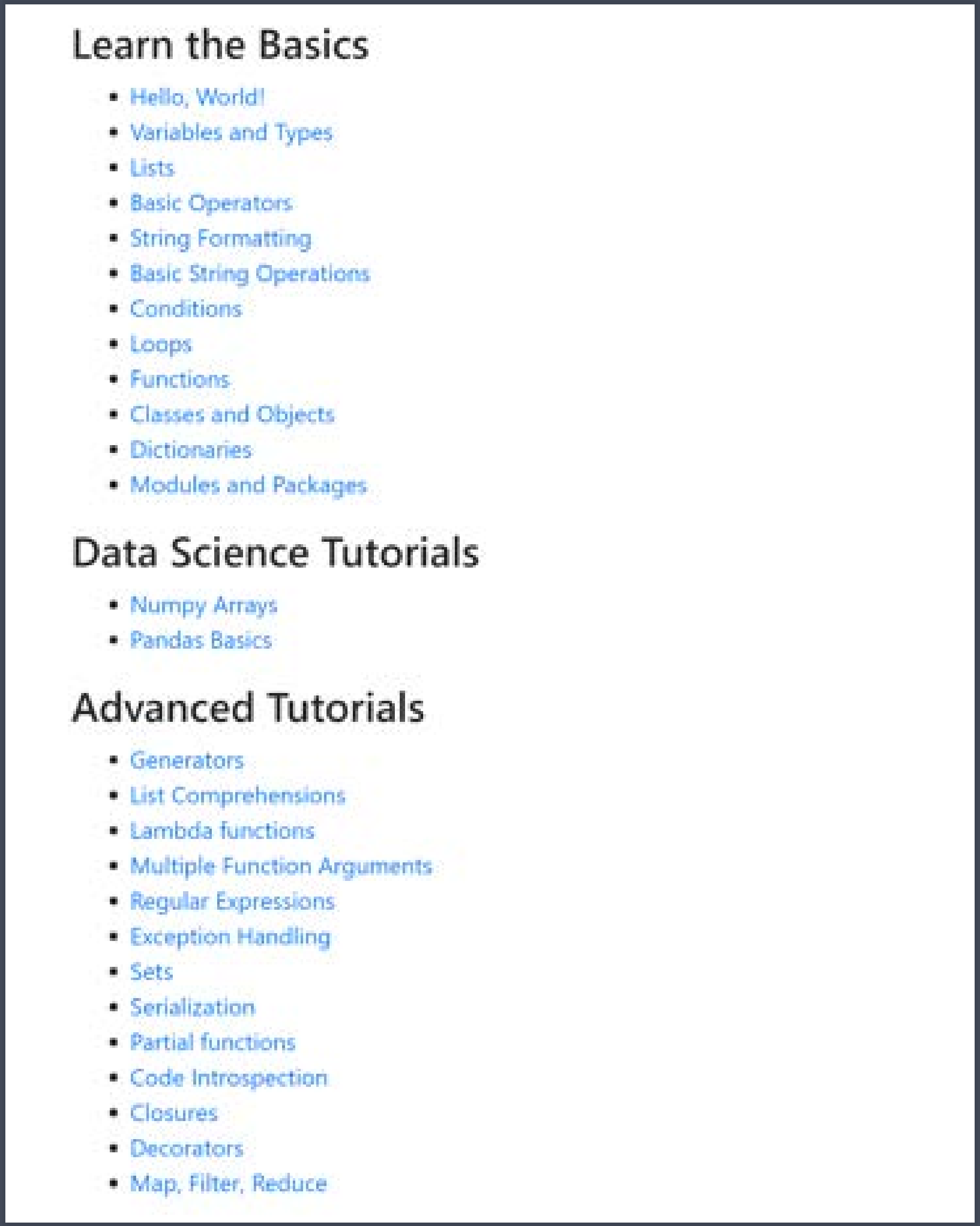
Financial aid available

595,940 already enrolled

Python resources

LearnPython

Free Interactive Python Tutorials



Learn the Basics

- Hello, World!
- Variables and Types
- Lists
- Basic Operators
- String Formatting
- Basic String Operations
- Conditions
- Loops
- Functions
- Classes and Objects
- Dictionaries
- Modules and Packages

Data Science Tutorials

- Numpy Arrays
- Pandas Basics

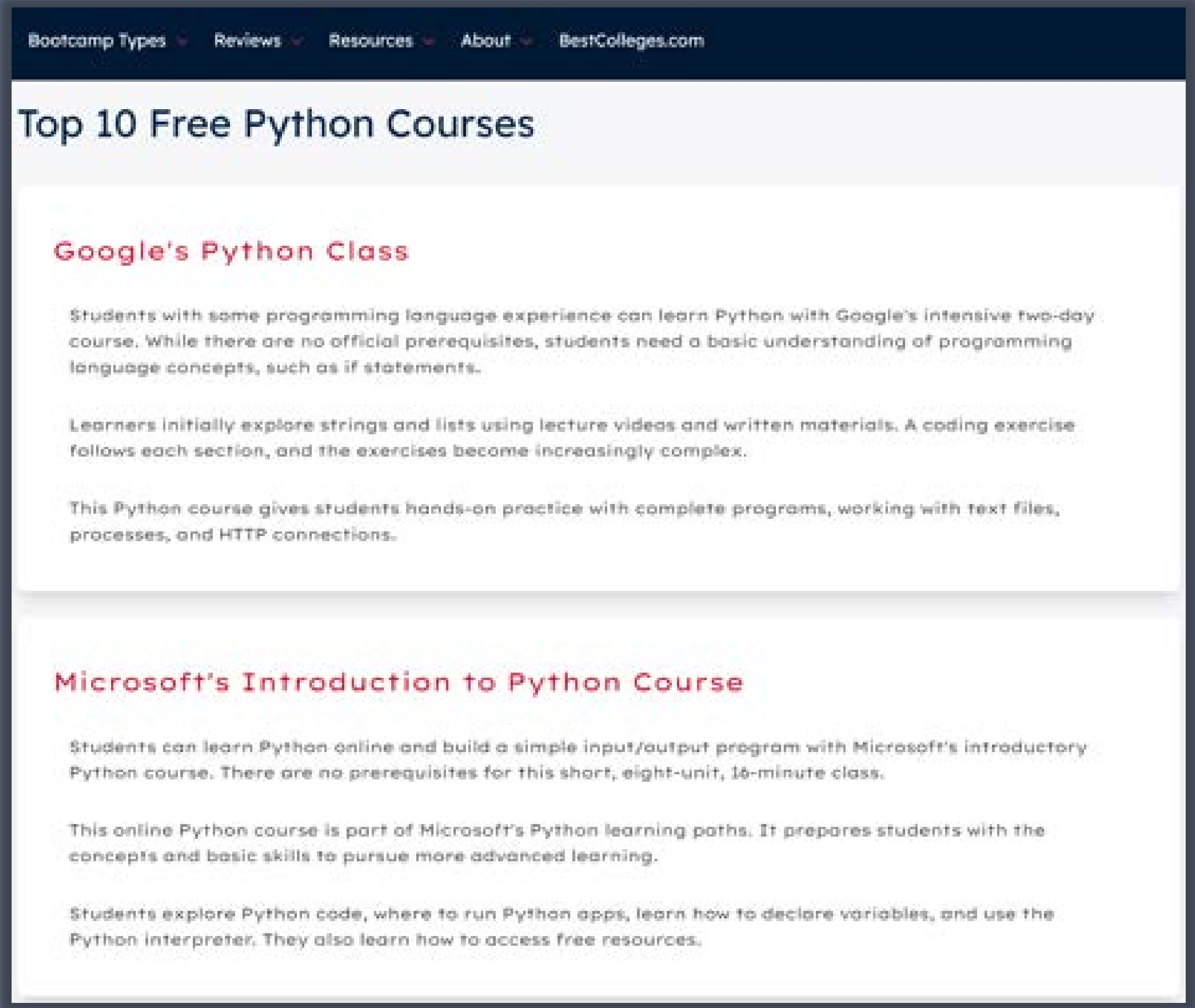
Advanced Tutorials

- Generators
- List Comprehensions
- Lambda functions
- Multiple Function Arguments
- Regular Expressions
- Exception Handling
- Sets
- Serialization
- Partial functions
- Code Introspection
- Closures
- Decorators
- Map, Filter, Reduce

Python resources

[BestColleges]()

10 Places to Learn Python for Free



Top 10 Free Python Courses

Google's Python Class

Students with some programming language experience can learn Python with Google's intensive two-day course. While there are no official prerequisites, students need a basic understanding of programming language concepts, such as if statements.

Learners initially explore strings and lists using lecture videos and written materials. A coding exercise follows each section, and the exercises become increasingly complex.

This Python course gives students hands-on practice with complete programs, working with text files, processes, and HTTP connections.

Microsoft's Introduction to Python Course

Students can learn Python online and build a simple input/output program with Microsoft's introductory Python course. There are no prerequisites for this short, eight-unit, 16-minute class.

This online Python course is part of Microsoft's Python learning paths. It prepares students with the concepts and basic skills to pursue more advanced learning.

Students explore Python code, where to run Python apps, learn how to declare variables, and use the Python interpreter. They also learn how to access free resources.

Terra resources

If you are new to Terra, we also recommend exploring the following resources:

- Overview Articles: Review high-level docs that outline what you can do in Terra, how to set up an account and account billing, and how to access, manage, and analyze data in the cloud
- Video Guides: Watch live demos of the Terra platform's useful features
- Terra Courses: Learn about Terra with free modules on the Leanpub online learning platform
- Data Tables QuickStart Tutorial: Learn what data tables are and how to create, modify, and use them in analyses
- Notebooks QuickStart Tutorial**:** Learn how to access and visualize data using a notebook
- Machine Learning Advanced Tutorial: Learn how Terra can support machine learning-based analysis

ScHARe

Thank you

Think-a-Thon poll

1. **Rate how useful this session was:**

□ Very useful

□ Useful

□ Somewhat useful

□ Not at all useful

Think-a-Thon poll

2. **Rate the pace of the instruction for yourself:**

□ Too fast

□ Adequate for me

□ Too slow

Think-a-Thon poll

3. **How likely will you participate in the next Think-a-Thon?**

□ Very interested, will definitely attend

□ Interested, likely will attend

□ Interested, but not available

□ Not interested in attending any others

Next Think-a-Thons:

bit.ly/think-a-thons

Register for ScHARe:

bit.ly/join-schare

✉ schare@mail.nih.gov