



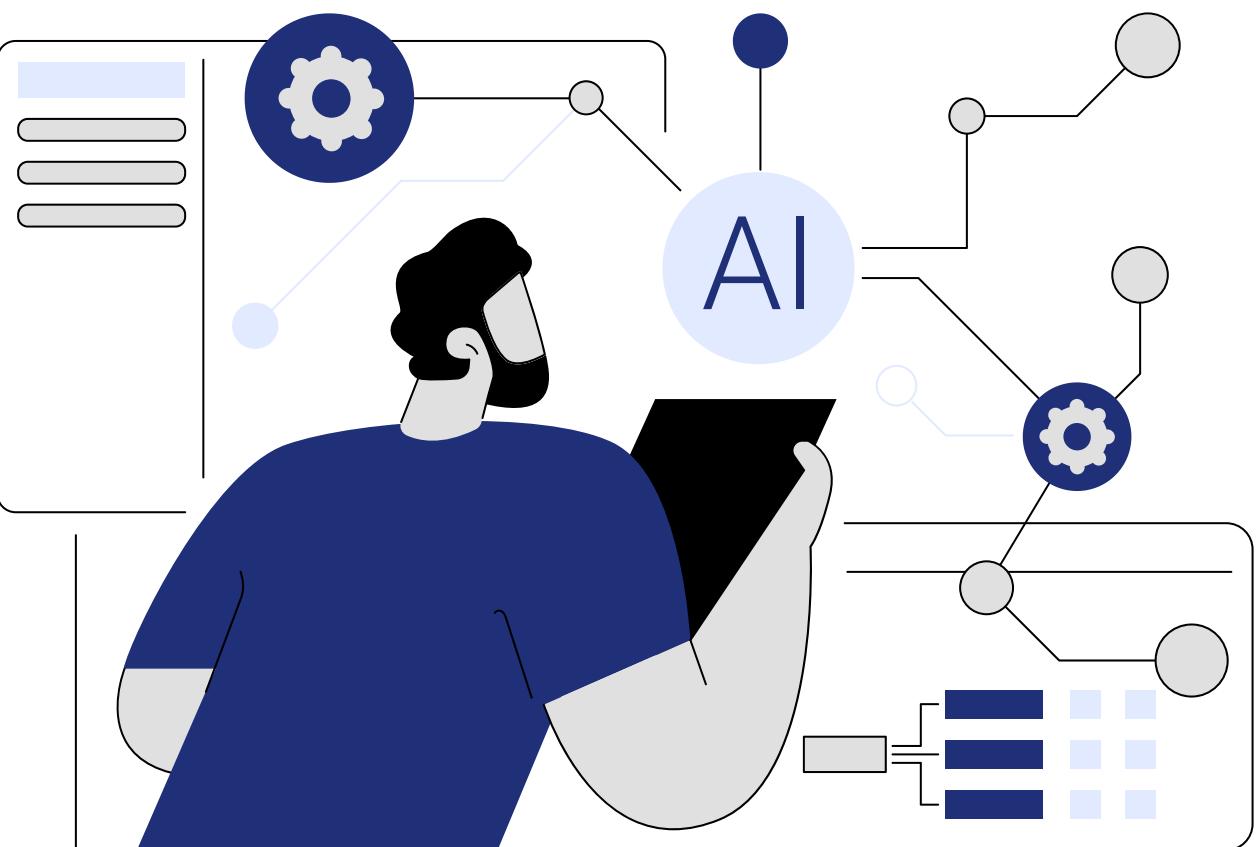
WELLCOME

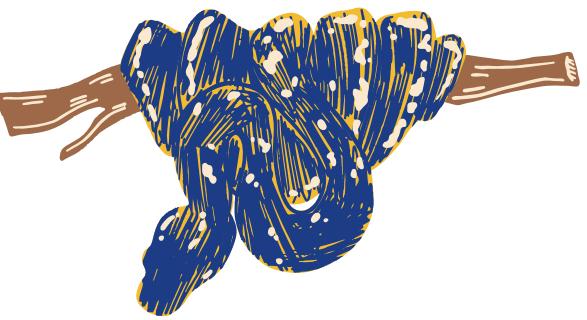


INTRO. TO DATA SCIENCE

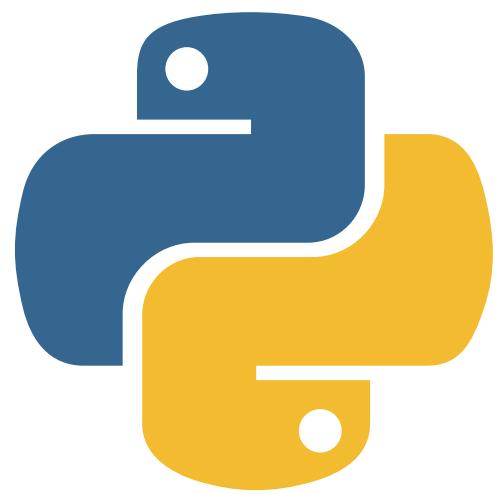
EMBARKING ON A JOURNEY
INTO DATA SCIENCE

YA MANON





Python for Data Science



COURSE STRUCTURE



This is a **project-based** course for students looking for a practical, hands-on, and highly engaging approach to learning, Data Science, Python essentials for data analysis.

Additional resources include:

- ☀️ **Source Code Link:** to serve as a helpful reference when you're offline or on the go
- ☀️ **Quizzes & Assignments** to test and reinforce key concepts, with step-by-step solutions
- ☀️ **Interactive demos** to keep you engaged and apply your skills throughout the course

SETTING EXPECTATIONS



Who this is for:

- Analysts or BI professionals looking to learn **Python for data analysis** and **Data Science**
- Students looking to learn the most popular open-source analytics tool
- Anyone who wants to understand the core fundamentals of the Python language and syntax



Who this is NOT for:

- Experienced Python programmers or advanced users
- Students looking to learn Python for Software or Web Development
- Anyone who would rather copy and paste code or run packages without building the foundational skills

COURSE OUTLINE

1

Why Python?

Introduce the Python analytics ecosystem and key reasons why it's the programming language of choice for many data analysts

2

Jupyter Notebook

Install Anaconda & create your first Jupyter Notebook, a user-friendly Python coding environment designed for data analysis and visualization

3

Data Types

Introduce native Python data types, common use cases, type conversion methods, and key concepts like iteration and mutability

4

Variables

Learn how to name and store values in memory using variables, as well as how to overwrite, delete and track them

5

Numerical Data

Learn how to work with numeric data, and use numeric functions to perform a range of arithmetic operations

6

String

Learn how to manipulate text via indexing and slicing, calculate string lengths, apply various string methods, and print f-strings to include variables

COURSE OUTLINE

7

Conditional Logic

Learn how to use IF statements and Boolean operators to establish conditional logic and control the flow of your programs

8

Sequence Data Types

Learn how to create, modify, and nest lists, tuples, and ranges, all of which allow you to store many values within a single variable

9

Loops

Understand the logic behind For and While loops and learn how to refine loop logic and handle common errors

10

Dictionaries and Sets

Address the limitations of working with lists and explore common scenarios for using dictionaries and sets in their place

11

Functions

Learn how to create custom functions in Python to boost productivity, and how to import external functions stored in modules or packages

12

Data Analytics with Pandas

Learn data manipulation and analysis, including one-dimensional data. Explore DataFrame and apply function for Data Analytics

COURSE OUTLINE

13

Intro Data Visualization

14

Matplotlib Basic

15

OO-ploting

Develop skills in interpreting and analyzing visualizations

16

Chart formatting

Matplotlib and Seaborn library and use it to build & customize several chart types, including line charts, bar charts, pie charts, scatterplots, and histograms

17

Chart types

18

Seaborn

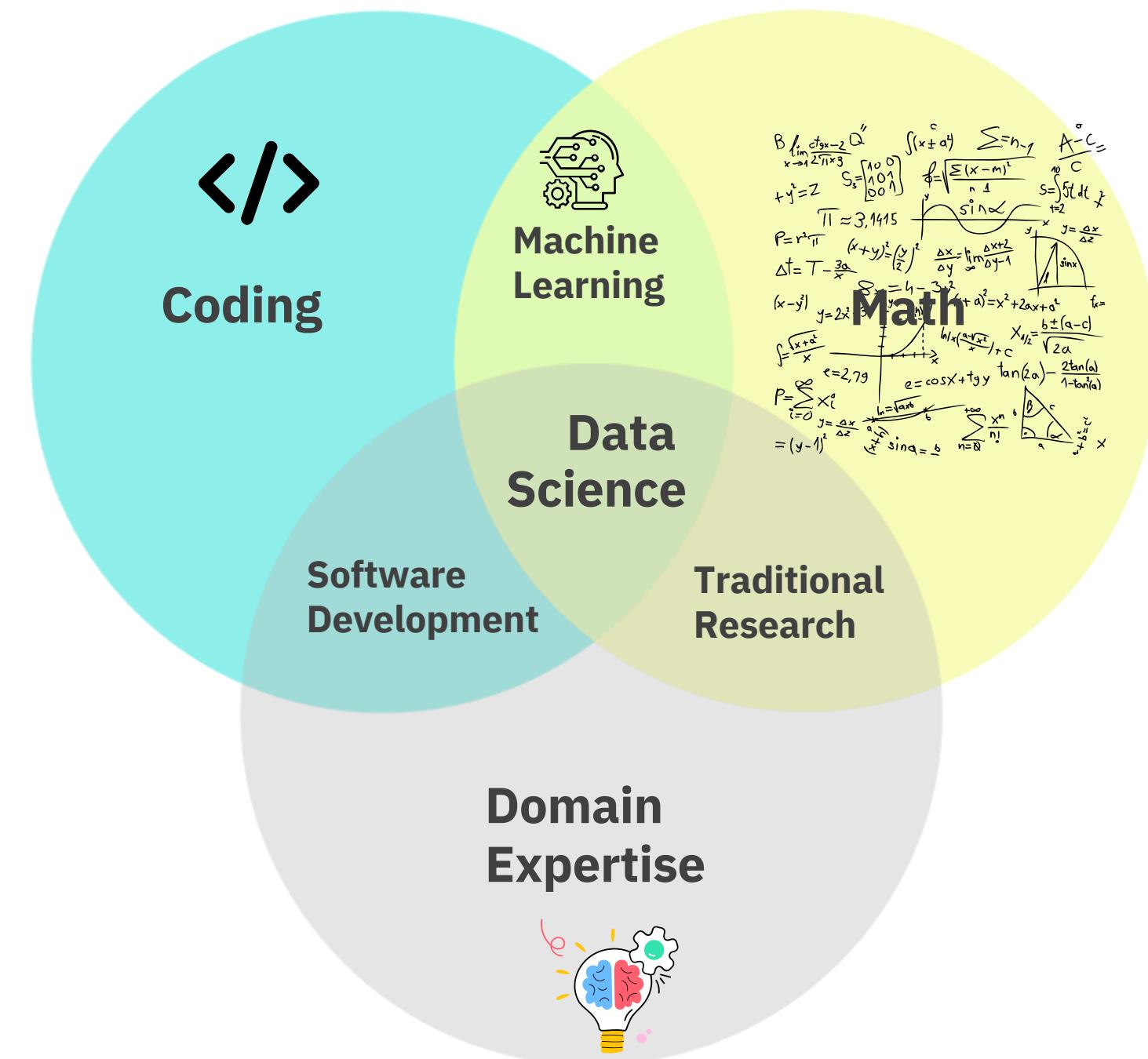
What's Data Science?

What's Data?

- **Data** information, experience, ...
- **Structures data:** Dimension (rows, columns), csv, excel , ..
 - **Example:** phone number, zip codes, identity number, gender,..
- **Unstructured data:** Everything else (no rows, columns)...
 - **Example:** emails, social media post, video, sensor, data...

What's Data Science?

Data science is the study of **data** to extract meaningful insights for **business**.



The key is in applying these along with soft skills like:

- Communication
- Problem solving
- Curiosity & creativity
- Googling prowess

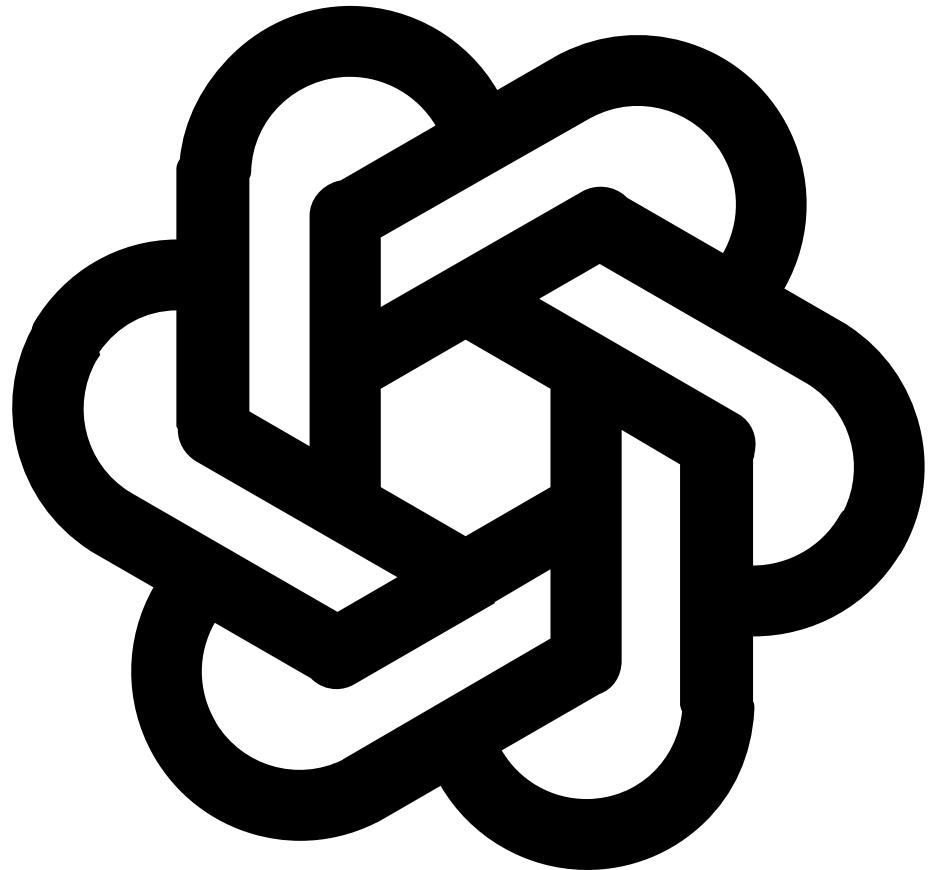


Data scientists & analysts approach problem solving in similar ways, but data scientists will often work with larger, more complex data sets and utilize advanced algorithms

What's Data Science?

Data science is about *using data to make smart decisions.*

What's Data Science?



<https://chatgpt.com>

Real life Application

- **Recommendation system (using machine learning , ai)**
- **Search engine (using machine learning, ai)**



WHY PYTHON?

MEET PYTHON

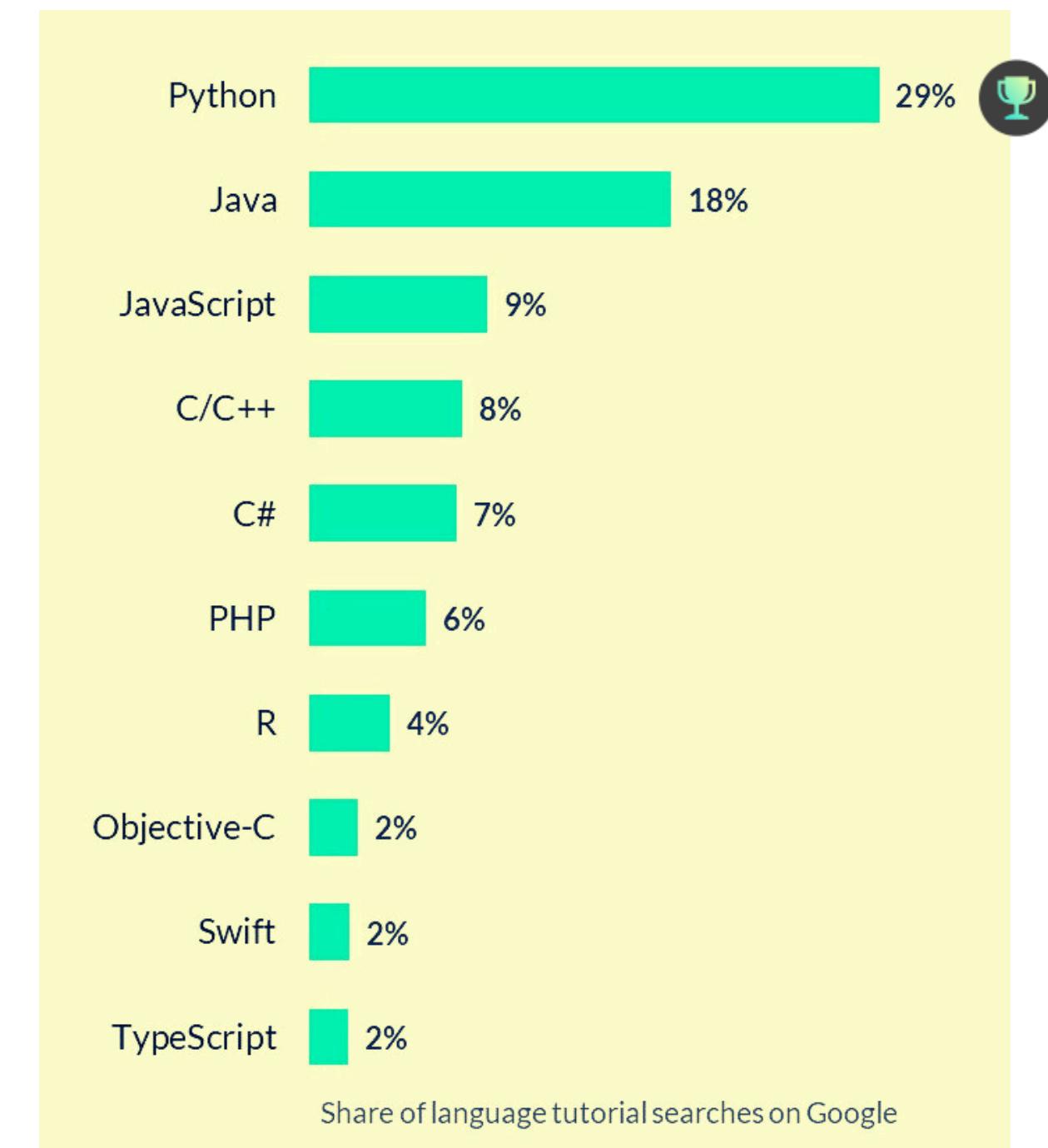


Python is a free, **open-source programming language** that is both powerful and easy to learn.

It has become one of the most popular languages in the world due to its accessibility, flexibility, ease of use, and wide range of applications, including:

- **Data Analytics**
- **Software Development**
- **Machine Learning**
- **Web Scraping**
- **Game Development And**
more!

Popularity of Programming Language (PYPL):



WHY PYTHON FOR ANALYTICS?



Scalability

Unlike most analytics tools or self-service Business Intelligence platforms, Python is **open source, free to use, and built for scale**



Versatility

With powerful libraries and frameworks, Python can add value at **every stage of the analytics workflow**, from data prep and analysis to machine learning and visualization



Community

Become part of a **large and active Python user community**, where you can share resources, get help, offer support, and connect with other users



Automation

Python can **automate complex tasks and workflows out of the box**, without complicated integrations or custom plug-ins



Demand

Python skills are **valuable and highly sought after**, and are becoming increasingly popular among analytics and Business Intelligence professionals



When it comes to data analytics, each tool has unique strengths and weaknesses; while Python shouldn't be the *only* tool in your stack, it can add tremendous value when combined with other tools like **Excel, SQL, Power BI & Tableau**

PYTHON ANALYTICS ECOSYSTEM

General Purpose Programming



Mastering **basePython** will give you a solid foundational understanding of the language, which is essential for using packages and libraries effectively

Data Manipulation & Analysis



Pandas helps us structure our data into dataset formats similar to that which you'd see in SQL or Excel. It also provides us with an arsenal of analytical functions that help us manipulate data and calculate the metrics we need to understand our data

Data Visualization



Matplotlib and **Seaborn** can create a wide array of visually appealing, static visualizations. **Plotly** can be used to create interactive visualizations and dynamic dashboards.

Machine Learning



Scikit learn is among the most popular tools for building and testing machine learning models. **Statsmodels** provides a suite of tools for model building and statistical analysis. **TensorFlow** is the industry standard for developing deep learning models.

DATA ROLES USING PYTHON

BI / DATA ANALYST

Data Analysts often use Python due to its cost effectiveness, or in collaboration with data science teams

Analysts are typically well-versed in base Python, Pandas, and at least one visualization library

→ **19%** of data analyst jobs require Python skills (**25%** in California)

DATA ENGINEER

Data Engineers commonly use Python to automate complex ETL processes or interact with APIs.

DBAs often use Pandas to manipulate data before storing it in a database or data warehouse.

→ **72%** of data engineer jobs require Python skills

DATA VIZ SPECIALIST

Data Visualization Specialists may use Python to design custom visuals that standard templates can't support.

They often utilize integrations with tools like Power BI or Tableau.

→ **33%** of data viz roles mention Python as a required or desired skill



DATA SCIENTIST

Data Scientists are most likely to use the 'full stack' of Python data tools. They leverage packages like Pandas, Scikit learn, Statsmodels & TensorFlow to build and deploy ML models.

→ **71%** of data science and machine learning jobs require Python skills



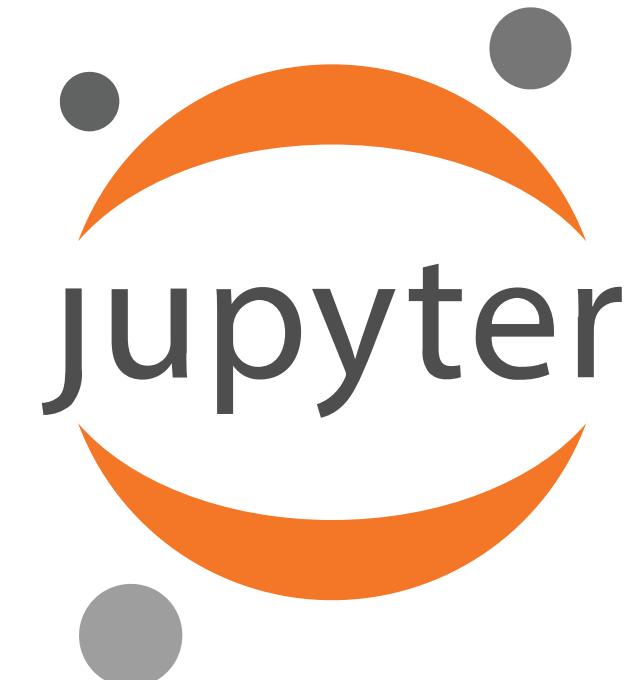
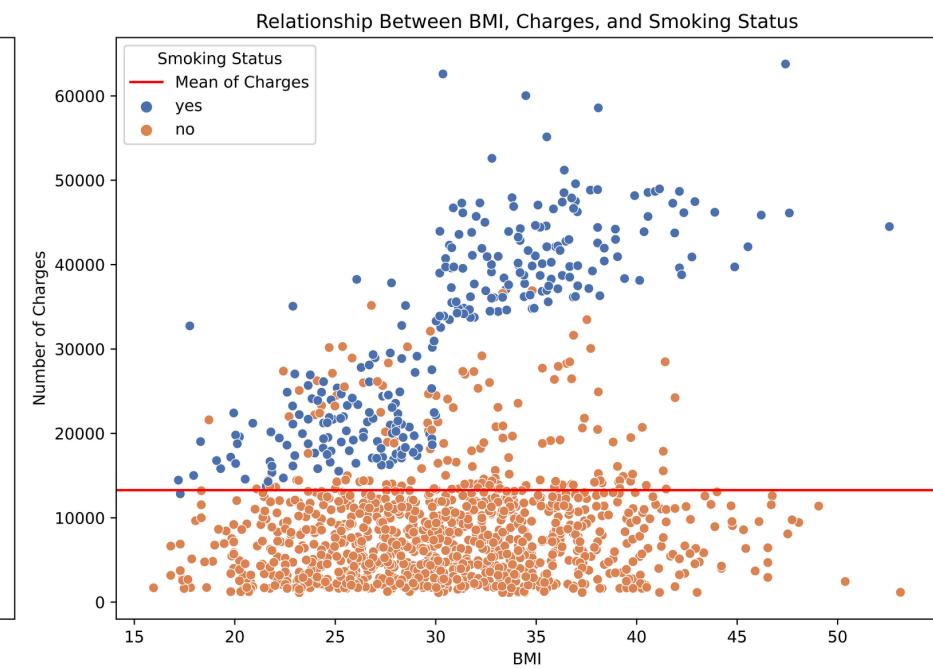
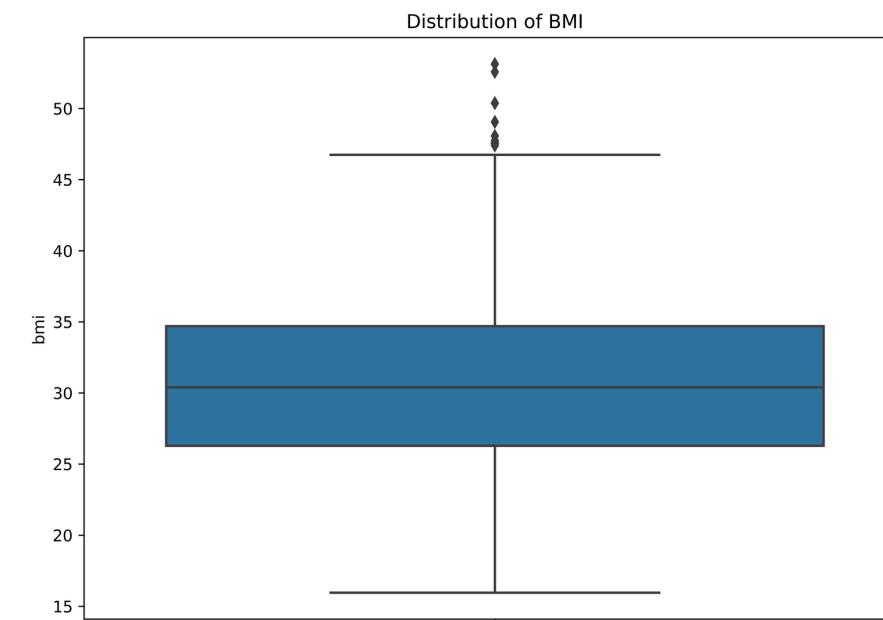
WELLCOME



INTRO. TO DATA SCIENCE

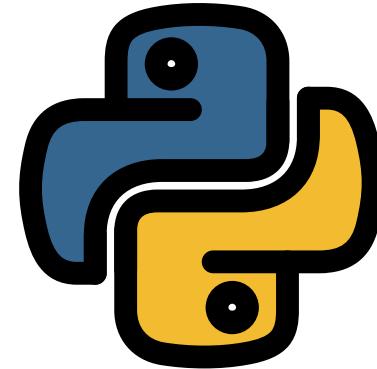
EMBARKING ON A JOURNEY
INTO DATA SCIENCE

YA MANON



JUPYTER NOTEBOOKS

JUPYTER NOTEBOOKS



In this section we'll install Anaconda and introduce **Jupyter Notebooks**, a **user-friendly coding environment** where we'll write our first Python program

TOPICS WE'LL COVER:

Installation & Setup

Notebook Interface

Comments & Markdown

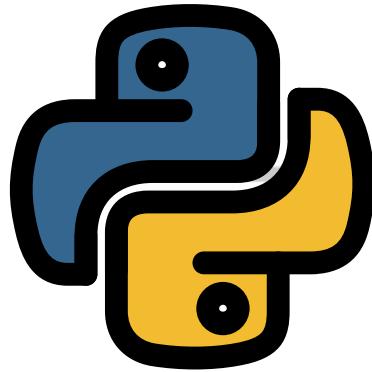
The Print Function

Google Colab

Helpful Resources

GOALS FOR THIS SECTION:

- Install **Anaconda** and **launch Jupyter Notebooks**
- Get comfortable with the Jupyter Notebook environment and interface
- Learn some very basic Python syntax and write our first simple programming



INSTALLING ANACONDA (MAC)

Installation & Setup

Notebook Interface

Comments & Markdown

The Print Function

Google Colab

Helpful Resources

1) Go to anaconda.com/products/distribution and click

Download

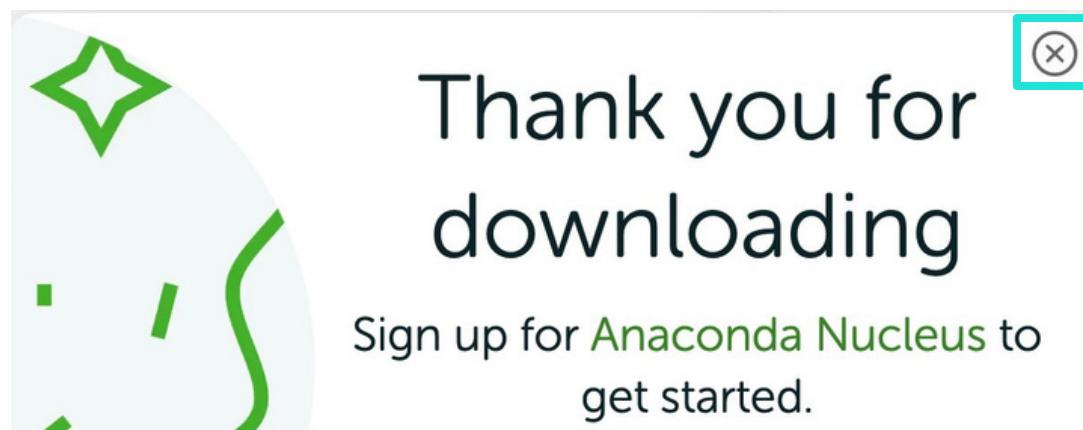
Individual Edition is now

ANACONDA DISTRIBUTION

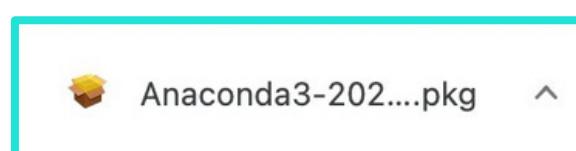
The world's most popular open-source Python distribution platform



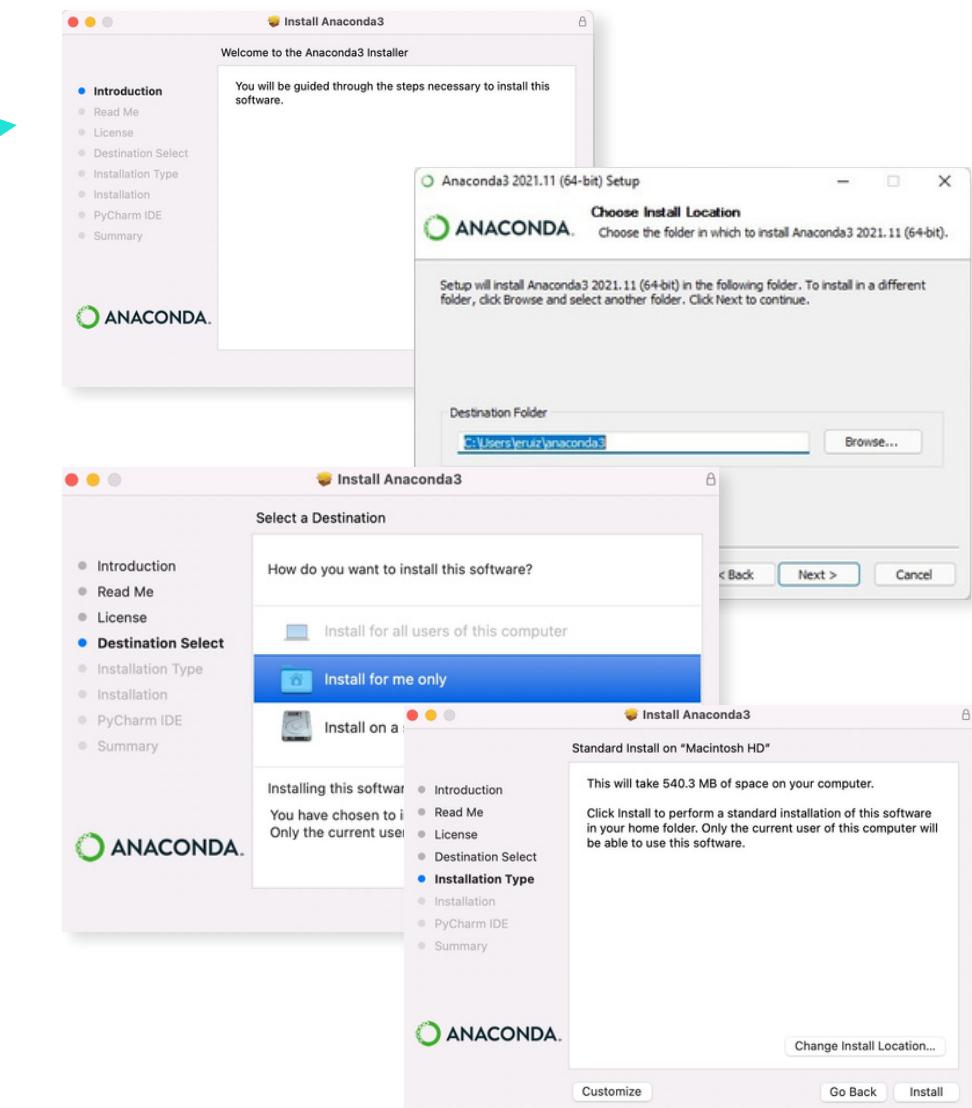
2) Click **X** on the Anaconda Nucleus pop-up
(no need to launch)

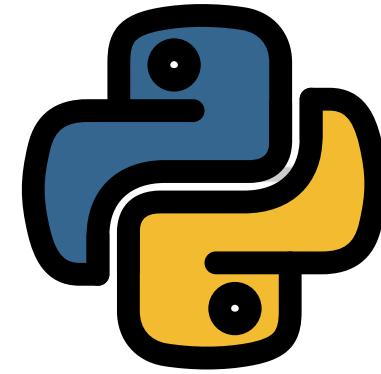


3) Launch the downloaded Anaconda **pkg** file



4) Follow the **installation steps**
(default settings are OK)





INSTALLING ANACONDA (PC)

Installation & Setup

Notebook Interface

Comments & Markdown

The Print Function

Google Colab

Helpful Resources

- 1) Go to anaconda.com/products/distribution and click

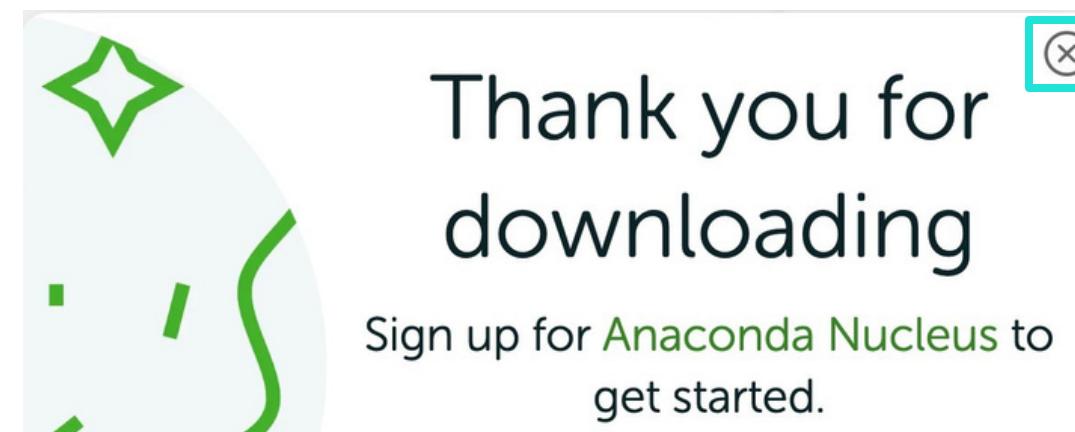
[Download](#)

Individual Edition is now
ANACONDA DISTRIBUTION

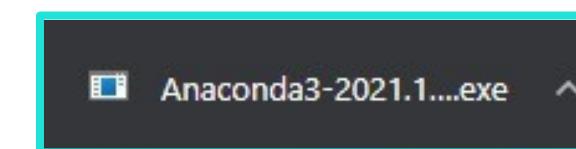
The world's most popular open-source Python distribution platform



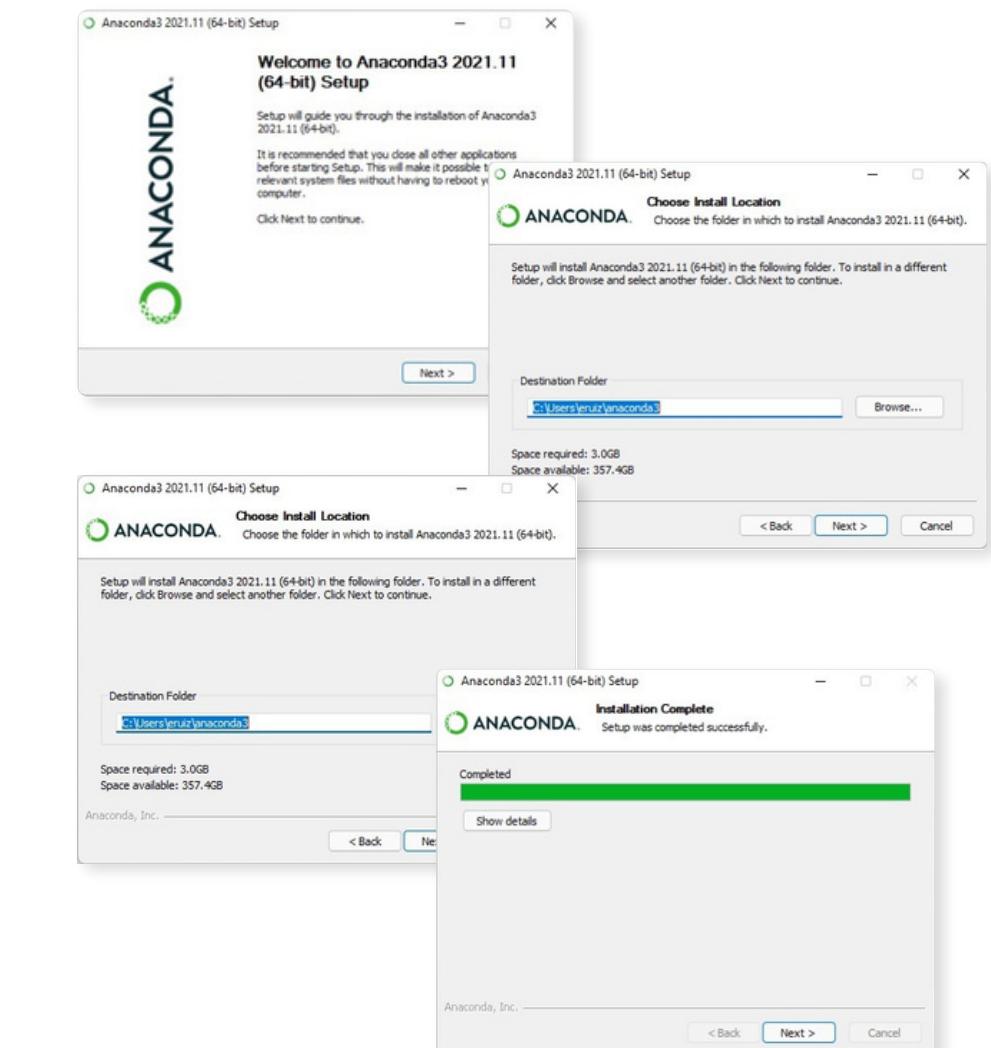
- 2) Click **X** on the Anaconda Nucleus pop-up (no need to launch)



- 3) Launch the downloaded Anaconda **exe** file



- 4) Follow the **installation steps** (default settings are OK)





LAUNCHING JUPYTER

Installation & Setup

Notebook Interface

Comments & Markdown

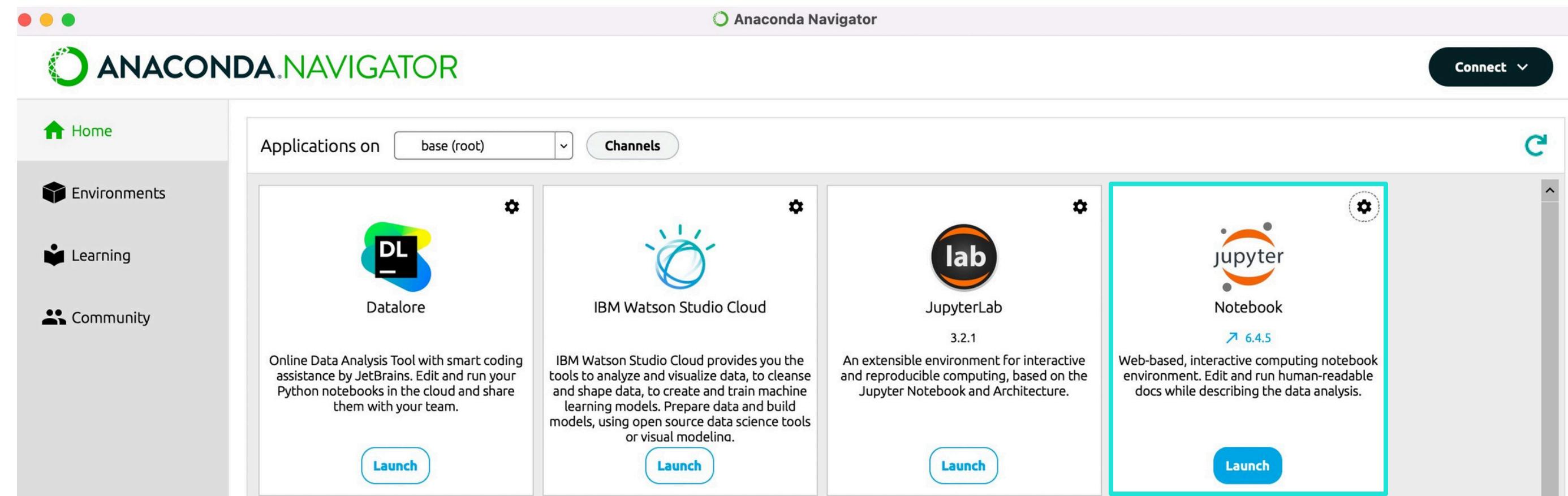
The Print Function

Google Colab

Helpful Resources

1) Launch **Anaconda Navigator**

2) Find **Jupyter Notebook** and click 





YOUR FIRST JUPYTER NOTEBOOK

Installation & Setup

Notebook Interface

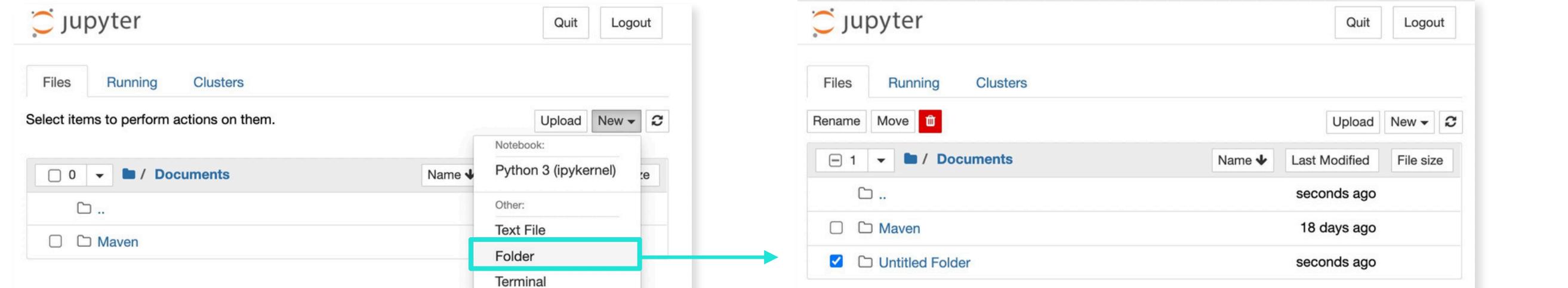
Comments & Markdown

The Print Function

Google Colab

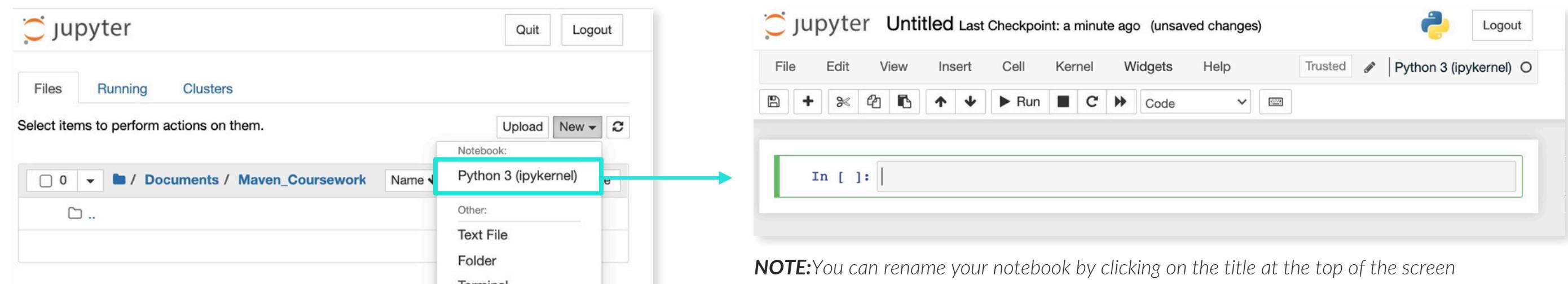
Helpful Resources

1) Once inside the Jupyter interface, **create a folder** to store your notebooks for the course



NOTE: You can rename your folder by clicking "Rename" in the top left corner

2) Open your new coursework folder and **launch your first Jupyter notebook!**



NOTE: You can rename your notebook by clicking on the title at the top of the screen



THE NOTEBOOK SERVER

Installation &
Setup

Notebook
Interface

Comments &
Markdown

The Print
Function

Google Colab

Helpful
Resources

NOTE: When you launch a Jupyter notebook, a terminal window may pop up as well; this is called a **notebook server**, and it powers the notebook interface

```
python
Last login: Tue Jan 25 14:04:12 on ttys002
(base) chrisb@Chriss-MBP ~ % jupyter notebook
[I 2022-01-26 08:45:53.886 LabApp] JupyterLab extension loaded from /Users/chrisb/opt/anaconda3/lib/python3.9/site-packages/jupyterlab
[I 2022-01-26 08:45:53.886 LabApp] JupyterLab application directory is /Users/chrisb/opt/anaconda3/share/jupyter/lab
[I 08:45:53.890 NotebookApp] Serving notebooks from local directory: /Users/chrisb
[I 08:45:53.890 NotebookApp] Jupyter Notebook 6.4.5 is running at:
[I 08:45:53.890 NotebookApp] http://localhost:8888/?token=3159cf032d9e6841d04910e257db2b24b6df6dfc878d6d5f
[I 08:45:53.890 NotebookApp] or http://127.0.0.1:8888/?token=3159cf032d9e6841d04910e257db2b24b6df6dfc878d6d5f
[I 08:45:53.890 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 08:45:53.893 NotebookApp]

To access the notebook, open this file in a browser:
file:///Users/chrisb/Library/Jupyter/runtime/nbsrvr-27175-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=3159cf032d9e6841d04910e257db2b24b6df6dfc878d6d5f
or http://127.0.0.1:8888/?token=3159cf032d9e6841d04910e257db2b24b6df6dfc878d6d5f
[W 08:46:05.829 NotebookApp] Notebook Documents/Maven_Coursework/Python_Intro.ipynb
```



If you close the server window,
your notebooks will not run!

Depending on your OS, and method of launching Jupyter, one may not open. As long as you can run your notebooks, don't worry!



THE NOTEBOOK INTERFACE

Installation &
Setup

Notebook
Interface

Comments &
Markdown

The Print
Function

Google Colab

Helpful
Resources

Menu Bar

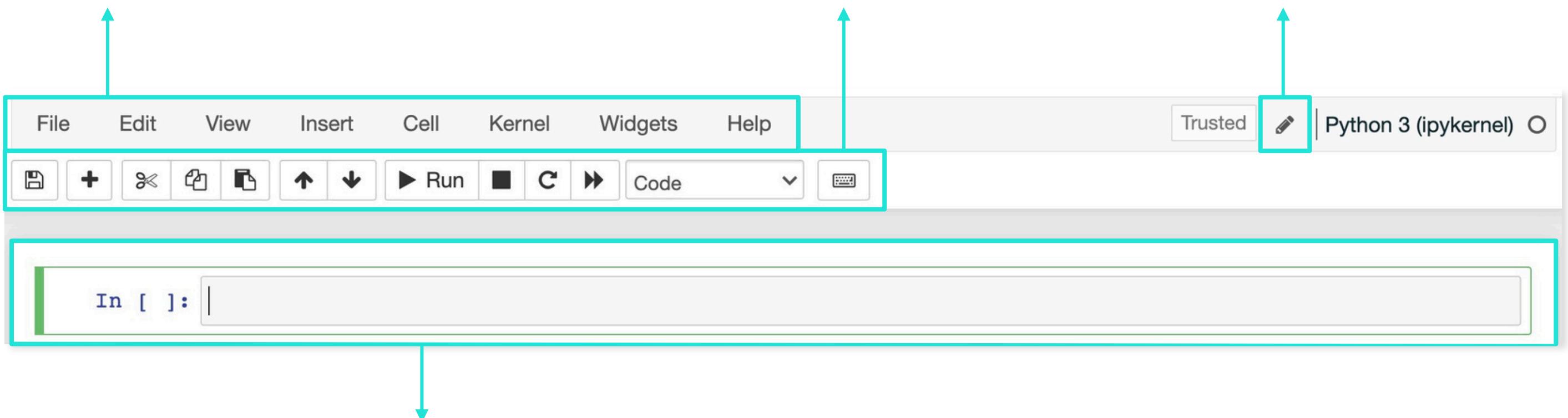
Options to manipulate the way
the notebook functions

Toolbar

Buttons for the most-used
actions within the notebook

Mode Indicator

Displays whether you are in **Edit**
Mode or **Command** Mode



Code Cell

Input field where you will write and
edit new code to be executed



MENU OPTIONS

Installation &
Setup

Notebook
Interface

Comments &
Markdown

The Print
Function

Google Colab

Helpful
Resources

Save or revert, make a copy, open a notebook, download, etc.

- File Browser
- Show Header
- Toggle Zen Mode
- Table of Contents ⌘ ⌘ K
- Show Notifications
- Show Line Numbers ⇧ L
- Match Brackets
- Wrap Words
- Open in JupyterLab
- Collapse Selected Code
- Collapse Selected Outputs
- Collapse All Code
- Collapse All Outputs
- Expand Selected Code
- Expand Selected Outputs
- Expand All Code
- Expand All Outputs
- Render Side-by-Side ⇧ R

Edit cells within your notebook (while in command mode)

- Edit
- View
- Run
- Kernel
- Settings
- Help
- Undo ⌘ Z
- Redo ⇧ ⌘ Z
- Undo Cell Operation Z
- Redo Cell Operation ⇧ Z
- Cut Cell X
- Copy Cell C
- Paste Cell Below V
- Paste Cell Above
- Paste Cell and Replace
- Delete Cell D, D
- Select All Cells ⌘ A
- Deselect All Cells
- Move Cell Up ⌃ ⌄ ↑
- Move Cell Down ⌃ ⌄ ↓
- Split Cell ⌃ ⌄ -
- Merge Selected Cells ⌃ M
- Merge Cell Above ⌃ ⌄ ⌂
- Merge Cell Below ⌃ ⌄ M
- Clear Cell Output
- Clear Outputs of All Cells
- Find... ⌘ F
- Find Next ⌘ G
- Find Previous ⇧ ⌘ G

Edit cosmetic options for your notebook.

- View
- Run
- Kernel
- Settings
- Help
- Activate Command Palette ⌃ ⌄ C
- Open JupyterLab
- File Browser
- Show Header
- Toggle Zen Mode
- Table of Contents ⇧ ⌘ K
- Show Notifications
- Show Line Numbers ⇧ L
- Match Brackets
- Wrap Words
- Open in JupyterLab
- Collapse Selected Code
- Collapse Selected Outputs
- Collapse All Code
- Collapse All Outputs
- Expand Selected Code
- Expand Selected Outputs
- Expand All Code
- Expand All Outputs
- Render Side-by-Side ⇧ R
- Text Editor Syntax Highlighting
- Preview Notebook with Panel
- Open with Panel in New Browser Tab



MENU OPTIONS

Installation &
Setup

Notebook
Interface

Comments &
Markdown

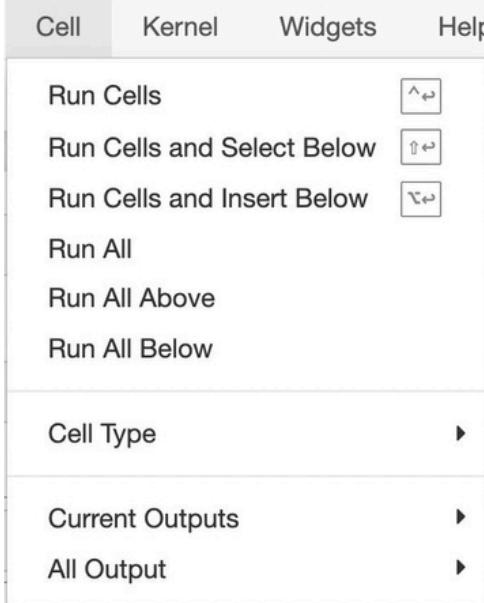
The Print
Function

Google Colab

Helpful
Resources

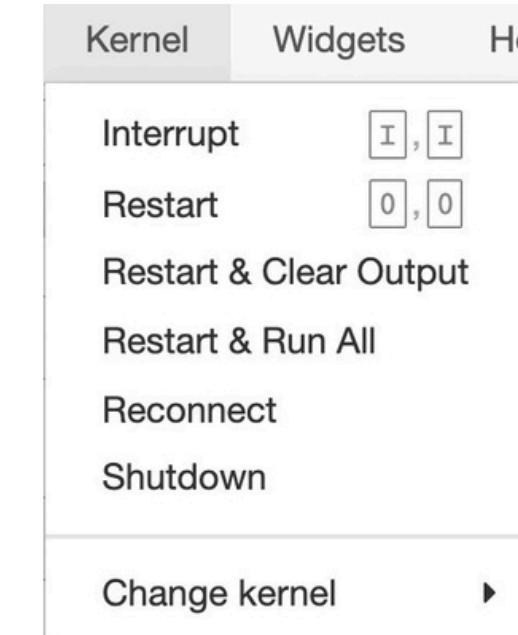
Cell

Access options for
running the cells in your
notebook



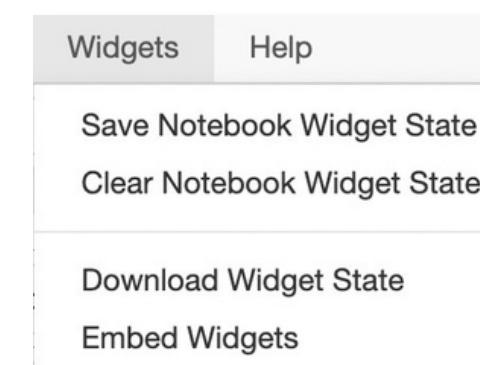
Kernel

Interact with the
instance of Python that
runs your code



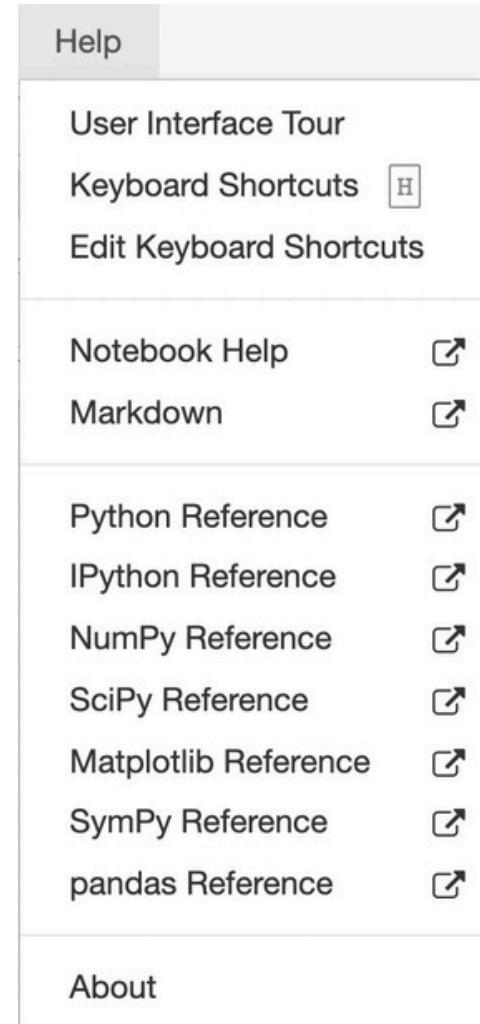
Widgets

Manage interactive
elements, or 'widgets' in
your notebook



Help

View or edit keyboard
shortcuts and access
Python reference
pages





THE TOOLBAR

Installation &
Setup

Notebook
Interface

Comments &
Markdown

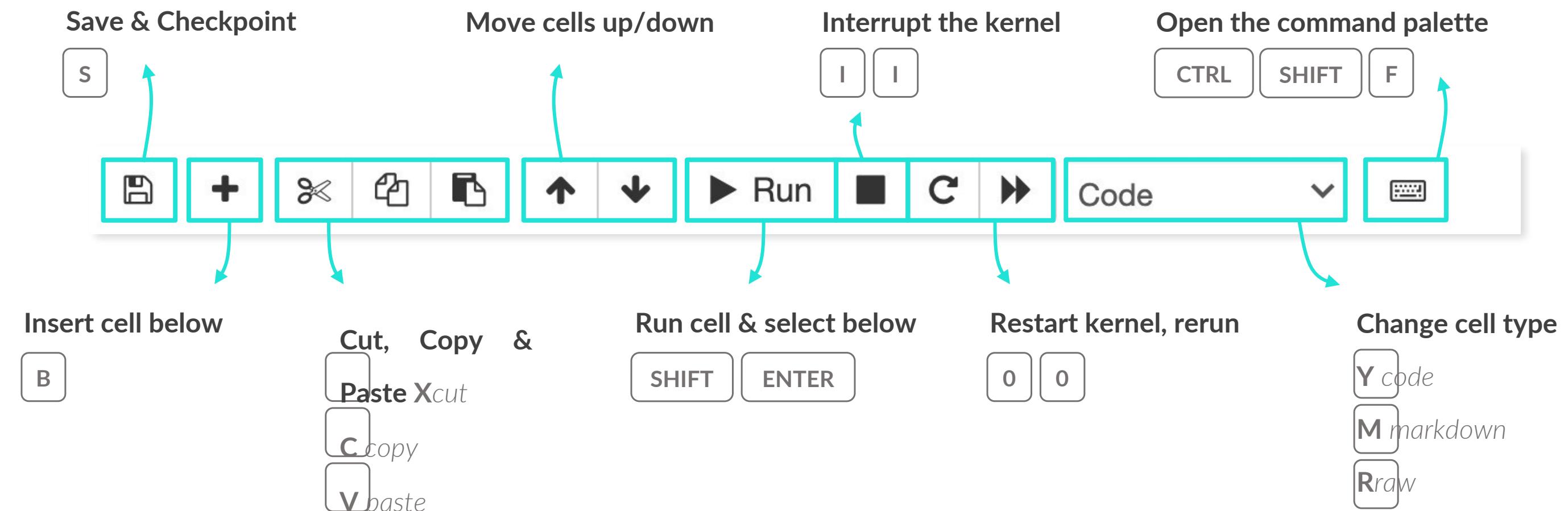
The Print
Function

Google Colab

Helpful
Resources

The **toolbar** provides easy access to the most-used notebook actions

- These actions can also be performed using hotkeys (keyboard shortcuts)



Shortcuts may differ depending on **which mode you are in**



EDIT & COMMAND MODES

Installation &
Setup

Notebook
Interface

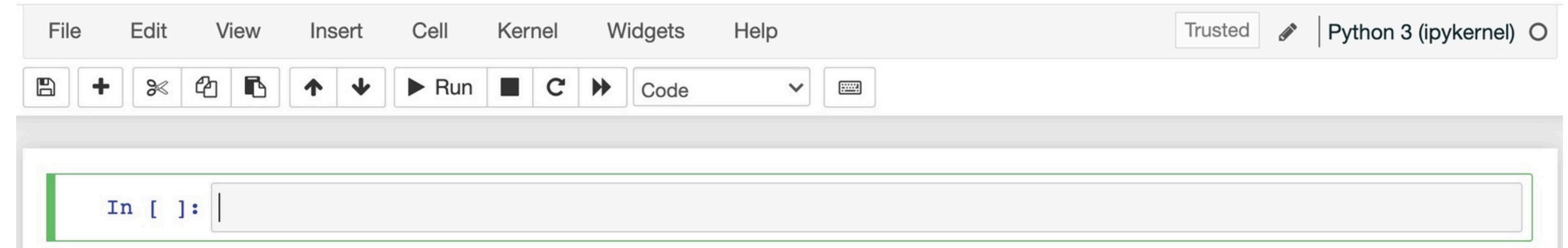
Comments &
Markdown

The Print
Function

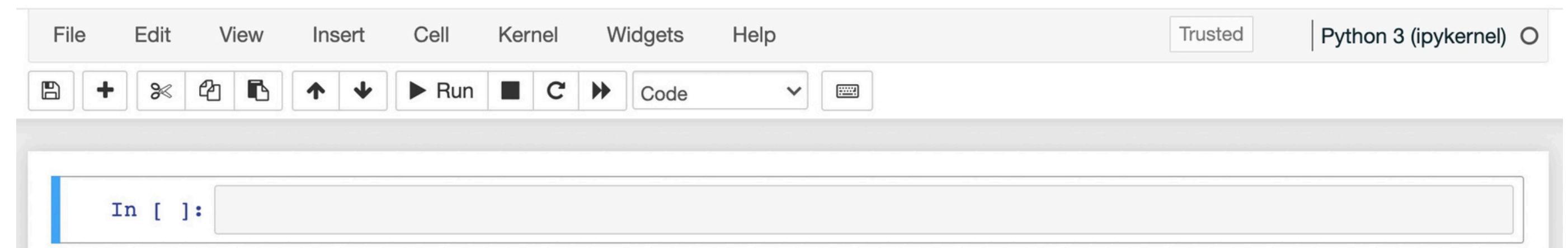
Google Colab

Helpful
Resources

EDIT MODE is for editing **content within cells**, and is indicated by **green** highlights and a icon



COMMAND MODE is for editing **the notebook**, and is indicated by **blue** highlights and no icon





THE CODE CELL

Installation &
Setup

Notebook
Interface

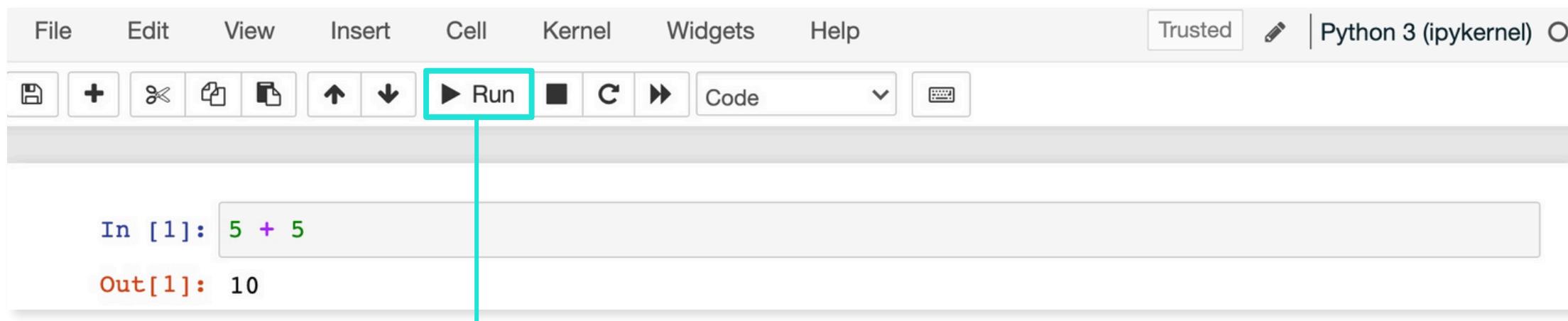
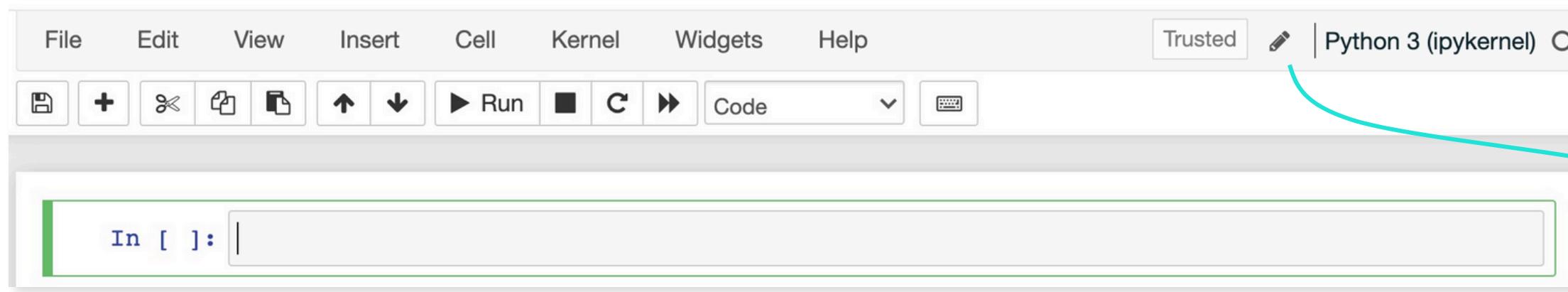
Comments &
Markdown

The Print
Function

Google Colab

Helpful
Resources

The **code cell** is where you'll write and execute Python code



Type some code, and click **Run** to execute

In[]: Our code (*input*)

*Note: not all code has an output!

Out[]: What our code produced (*output*)*



Congratulations, you just became
a Python programmer!



THE CODE CELL

Installation &
Setup

Notebook
Interface

Comments &
Markdown

The Print
Function

Google Colab

Helpful
Resources

The **code cell** is where you'll write and execute Python code

In [1]: `5 + 5`

Out[1]: 10

In [2]: `5 + 5`

Out[2]: 10

Note that our output hasn't changed, but the number in the brackets increased from **1** to **2**.

This is a **cell execution counter**, which indicates how many cells you've run in the current session. If the cell is still processing, you'll see

In[*]

Click back into the cell (or use the up arrow) and press **SHIFT + ENTER** to rerun the code

In [2]: `5 + 5`

Out[2]: 10

In [3]: `5 - 5`

Out[3]: 0

The cell counter will continue to increment as you run additional cells



COMMENTING CODE

Installation &
Setup

Notebook
Interface

Comments &
Markdown

The Print
Function

Google Colab

Helpful
Resources

Comments are lines of code that start with '#' and do not run

- They are great for explaining portions of code for others who may use or review it
- They can also serve as reminders for yourself when you revisit your code in the future

```
In [4]: # I'm subtracting five from five. Add a space between your hash and comment.  
      5 - 5
```

```
Out[4]: 0
```

```
In [5]: 5 - 5 # change the second 5 to a 6 tomorrow
```

```
Out[5]: 0
```

```
# This notebook is about teaching the basics of Jupyter notebook.  
# Should i define what a jupyter notebook is here?  
# I'm subtracting five from five. Add a space between your hash and comment.  
5 - 5 # 5 is the fifth integer greater than zero. It is also the number of fingers on our hand  
# 5 is a very interesting number  
# so is 0, which is the output
```

```
0
```

Think about your audience when commenting your code (you may not need to explain basic arithmetic to an experienced Python programmer)

Be conscious of over-commenting, which can actually make your code even more difficult to read



Comments should explain **individual cells or lines of code**, NOT your entire workflow -we have better tools for that!



MARKDOWN CELLS

Installation &
Setup

Notebook
Interface

Comments &
Markdown

The Print
Function

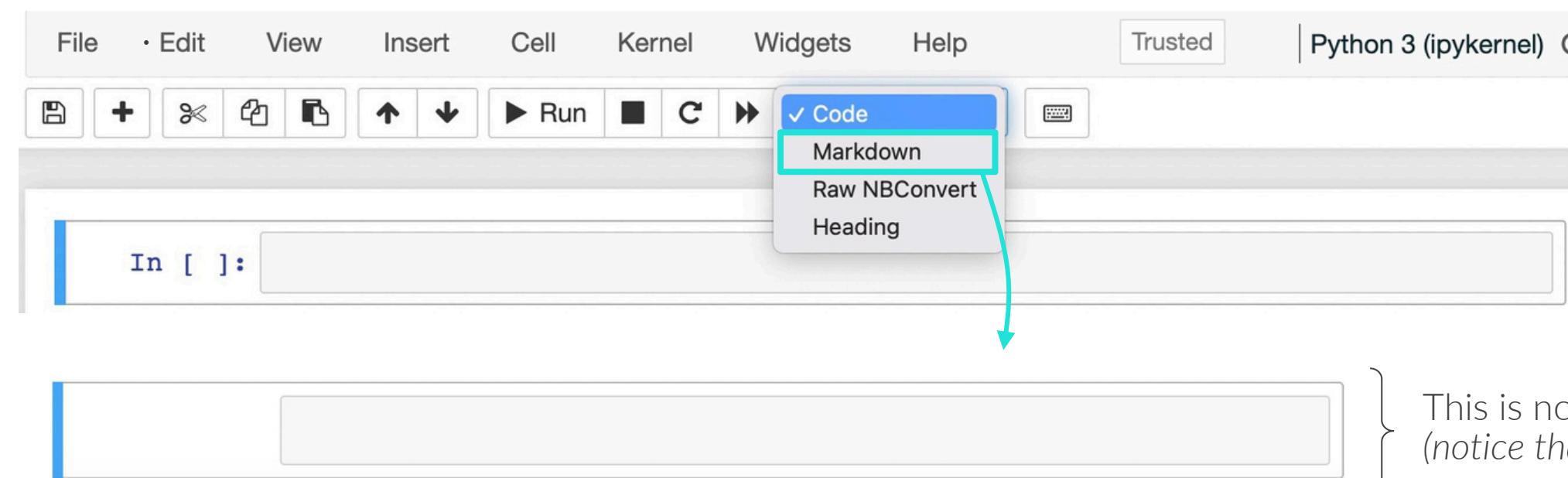
Google Colab

Helpful
Resources

Markdown cells let you write structured text passages to explain your workflow, provide additional context, and help users navigate the notebook

To create a markdown cell:

- 1 Create a new cell above the top cell (press **A** with the cell selected)
- . Select “**Markdown**” in the cell menu (or press **M**)
- 2



} This is now a markdown cell
(notice that the
In[]: disappeared)



MARKDOWN SYNTAX

Installation &
Setup

Notebook
Interface

Comments &
Markdown

The Print
Function

Google Colab

Helpful
Resources

Markdown cells use a special **text formatting syntax**

Jupyter Notebook Intro

Section 1: Markdown Basics

This is body text. I can use this area to provide more in depth explanations of my:

- * Thought process
- * Overall workflow
- * etc

Anything that would be too much text for comments.

To create bulleted lists, begin a line with *.

Numbered lists can be created by beginning a line with 1., 2., and so on.

Markdown has a ****lot**** of capabilities, and could be a course on its own. You will learn more as you build more notebooks and look at the work of others.

The Essentials to get started are:

1. Create headers with # (one is biggest, six is the smallest header)
2. ****Bold****, ***italicize***, *****Bold AND Italic*****
3. Creating bulleted or numbered lists (begin a new line with * for bullets 1. for numbers).
4. Code highlighting e.g. ``my_variable = 5``. Use the backtick, not apostrophe.

To explore further, I highly recommend [\[this guide.\]](https://www.markdownguide.org/basic-syntax/)(<https://www.markdownguide.org/basic-syntax/>)



MARKDOWN SYNTAX

Installation &
Setup

Notebook
Interface

Comments &
Markdown

The Print
Function

Google Colab

Helpful
Resources

Markdown cells use a special **text formatting syntax**

Jupyter Notebook Intro

Section 1: Markdown Basics

This is body text. I can use this area to provide more in depth explanations of my:

- Thought process
- Overall workflow
- etc

Anything that would be too much text for comments.

To create bulleted lists, begin a line with *.

Numbered lists can be created by beginning a line with 1., 2., and so on.

Markdown has a **lot** of capabilities, and could be a course on its own. You will learn more as you build more notebooks and look at the work of others.

The Essentials to get started are:

1. Create headers with # (one is biggest, six is the smallest header)
2. **Bold**, *italicize*, **Bold AND Italic**
3. Creating bulleted or numbered lists (begin a new line with * for bullets 1. for numbers).
4. Code highlighting e.g. `my_variable = 5`. Use the backtick, not apostrophe.

To explore further, I highly recommend [this guide](#).

In [4]: 5 + 5

Out[4]: 10



THE PRINT FUNCTION

Installation &
Setup

Notebook
Interface

Comments &
Markdown

The Print
Function

Google Colab

Helpful
Resources

The **print()** function will display a specified value

In [11]: `print('Hello World!')`

Hello World!

Simply specify the value you want to print inside the parenthesis

In [16]: `print(5, 5 + 5)`

5 10

You can print multiple values by separating them with **commas**

Note that this does not say different from the standard output returned by Python

Out[16]: as a printed output is



Besides `print()`, Python has **many built-in functions** along with tools for **creating our own custom functions** (*more on that later!*)



PRO TIP: Add a “**?**” after a function name to access documentation

In [18]: `print?`

Docstring:

`print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)`

Prints the values to a stream, or
Optional keyword arguments:

file: a file-like object (stream)

sep: string inserted between values

end: string appended after the last value

flush: whether to forcibly flush the stream

Type: `builtin_function_or_method`



ALTERNATIVE: GOOGLE COLAB

Installation &
Setup

Notebook
Interface

Comments &
Markdown

The Print
Function

Google Colab

Helpful
Resources

Google Colab is Google's cloud-based version of Jupyter Notebooks

To create a Colab notebook:

1. Log in to a Gmail account
2. Go to colab.research.google.com
3. Click "new notebook"



Colab is very similar to Jupyter Notebooks (*they even share the same file extension*); the main difference is that you are connecting to **Google Drive** rather than your machine, so files will be stored in Google's cloud

The screenshot shows the Google Colab interface with the 'Recent' tab selected. At the top, there are tabs for 'Examples', 'Recent', 'Google Drive', 'GitHub', and 'Upload'. Below the tabs is a search bar labeled 'Filter notebooks'. The main area displays two recent notebooks: 'Welcome To Colaboratory' (last opened 1:34 PM on January 4) and 'Scratch_Work.ipynb' (last opened January 25 on January 4). Each notebook entry includes a preview icon, the notebook name, its last opening time, the date it was created, and a trash bin icon.

The screenshot shows the Google Drive interface with the 'My Drive' section selected. On the left, there is a sidebar with options like 'New', 'Priority', 'My Drive' (which is highlighted), 'Shared drives', 'Shared with me', 'Recent', 'Starred', and 'Trash'. In the main area, there is a search bar labeled 'Search in Drive'. A preview of a notebook titled '1_Python_Intro.ipynb' is shown, along with a 'CO' logo. At the bottom right, there is a blue button labeled 'New notebook' with a white border, and a 'Cancel' button next to it.



HELPFUL RESOURCES

Installation &
Setup

Notebook
Interface

Comments &
Markdown

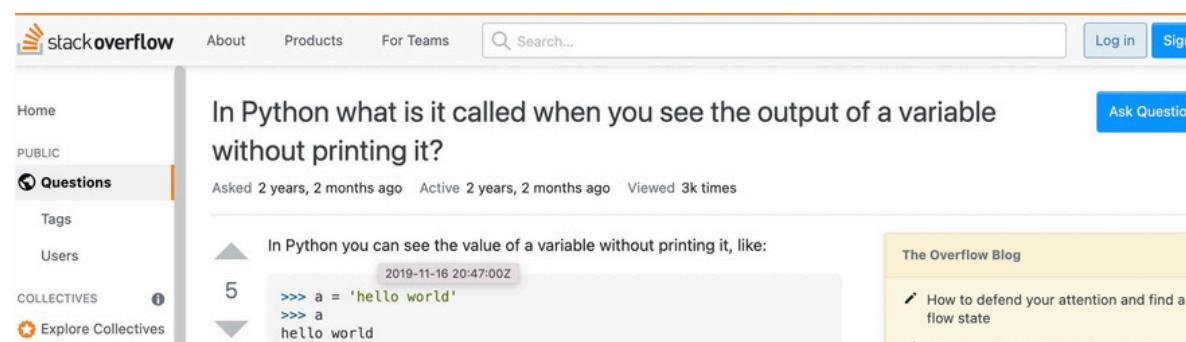
The Print
Function

Google Colab

Helpful
Resources

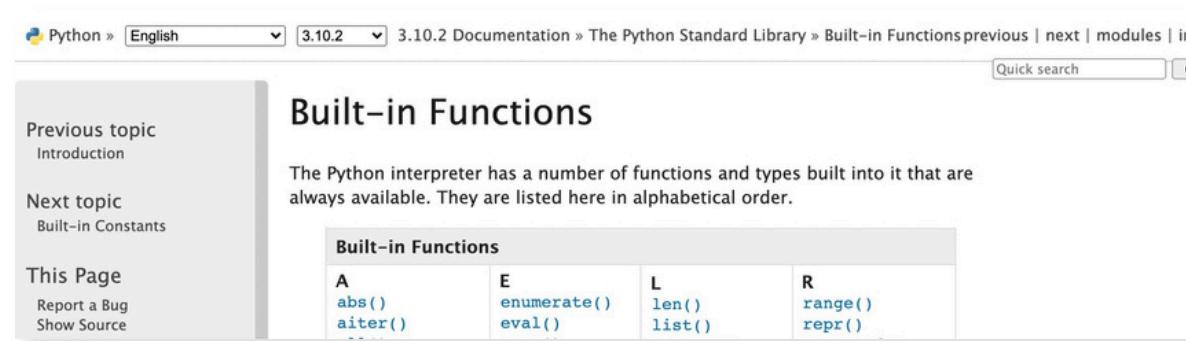


Google your questions – odds are someone else has asked the same thing and it has been answered (*include Python in the query!*)



Stack Overflow is a public coding forum that will most likely have the answers to most of the questions you'll search for on Google

<https://stackoverflow.com/>



The **Official Python Documentation** is a great “cheat sheet” for library and language references

<https://docs.python.org/3/>



There are many quality **Python & Analytics Blogs** on the web, and you can learn a lot by subscribing and reviewing the concepts and underlying code

<https://towardsdatascience.com/>

KEY TAKEAWAYS



JupyterNotebooks are user-friendly coding environments

- Jupyternotebooks are popular among analysts and data scientists, since they allow you to create and document entire analytical workflows and render outputs and visualizations on screen



Code cells are where you write and execute Python code

- Make sure that you know how to run, add, move, and remove cells, as well as how to restart your kernel or stop the code from executing



Use comments and markdowncells for documentation

- Comments should be used to explain specific portions of your code, and markdown should be used to document your broader workflow and help users navigate the notebook



Google Colab is a popular cloud-based alternative to JupyterNotebooks

- Colab and Jupyternotebooks are very similar and share the same file extension, but Colabfiles are stored in Google Drive instead of on your machine