

INTRO TO DATA SCIENCE

INTRO TO DATA SCIENCE



In this section we'll **introduce the field of data science**, discuss how it compares to other data fields, and walk through each phase of the data science workflow

TOPICS WE'LL COVER:

What is Data Science?

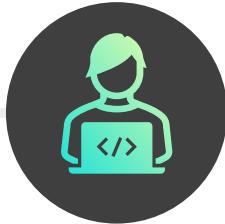
Essential Skills

What is Machine Learning?

Data Science Workflow

GOALS FOR THIS SECTION:

- Compare roles under the broader data analytics umbrella
- Discuss essential skills of a data scientist
- Compare data science and machine learning
- Introduce supervised and unsupervised learning, and commonly used algorithms
- Review each phase of the data science workflow



WHAT IS DATA SCIENCE?

What is Data Science?

Essential Skills

What is Machine Learning?

Data Science Workflow

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



Wait, isn't that **business intelligence** ?

Yes! The differences lie in the **types of problems** you solve, and **tools and techniques** you use to solve them:

What happened?

- Descriptive Analytics
- Data Analysis
- Business Intelligence

What's going to happen?

- Predictive Analytics
- Data Mining
- **Data Science**



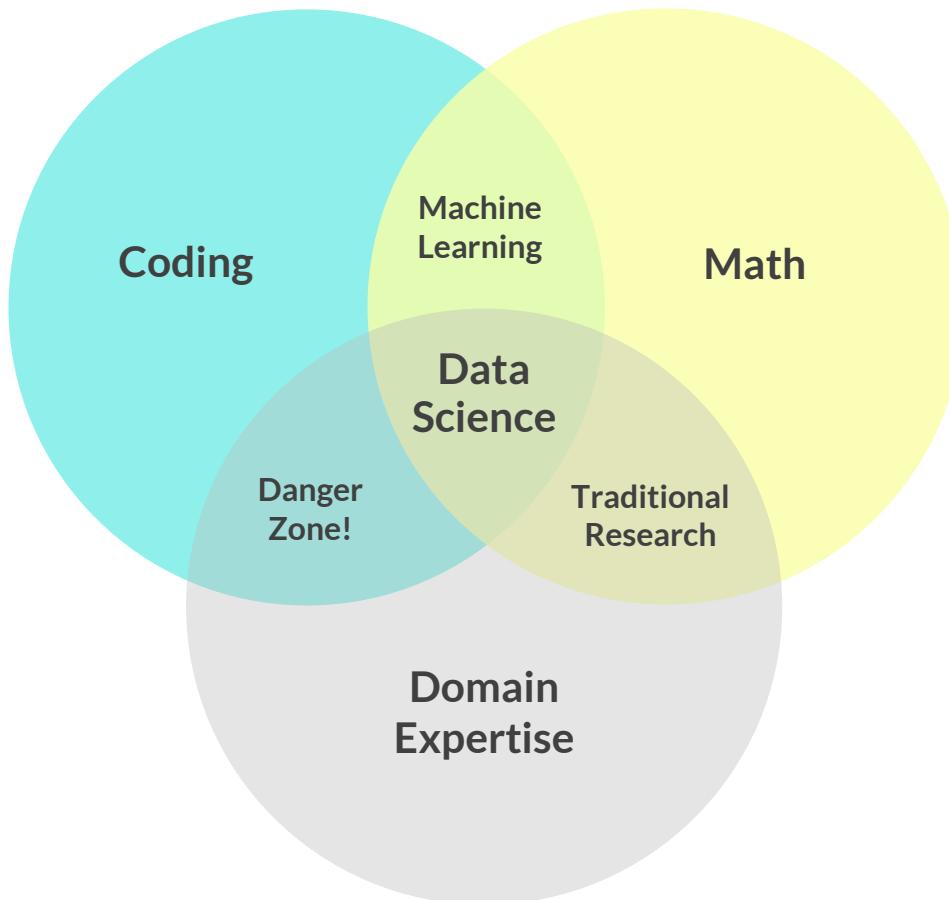
DATA SCIENCE SKILL SET

What is Data Science?

Essential Skills

What is Machine Learning?

Data Science Workflow



Data science requires a blend of **coding**, **math**, and **domain expertise**

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

- Communication
- Problem solving
- Curiosity & creativity
- Grit
- 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 IS MACHINE LEARNING?

What is Data Science?

Essential Skills

What is Machine Learning?

Data Science Workflow

Machine learning enables computers to learn and make decisions from data



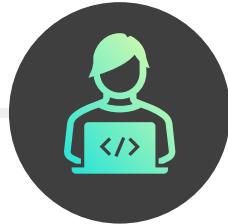
How can a computer learn?

By using **algorithms**, which is a set of instructions for a computer to follow



How does this compare with data science?

Data scientists know how to apply algorithms, meaning they're able to tell a computer how to learn from data



SUPERVISED VS. UNSUPERVISED LEARNING

What is Data Science?

Essential Skills

What is Machine Learning?

Data Science Workflow

Machine learning algorithms fall into two broad categories:
supervised learning and **unsupervised learning**

Supervised Learning

Using historical data to predict the future



What will house prices look like for the next 12 months?



How can I flag suspicious emails as spam?

Unsupervised Learning

Finding patterns and relationships in data



How can I segment my customers?



Which TV shows should I recommend to each user?



COMMON ALGORITHMS

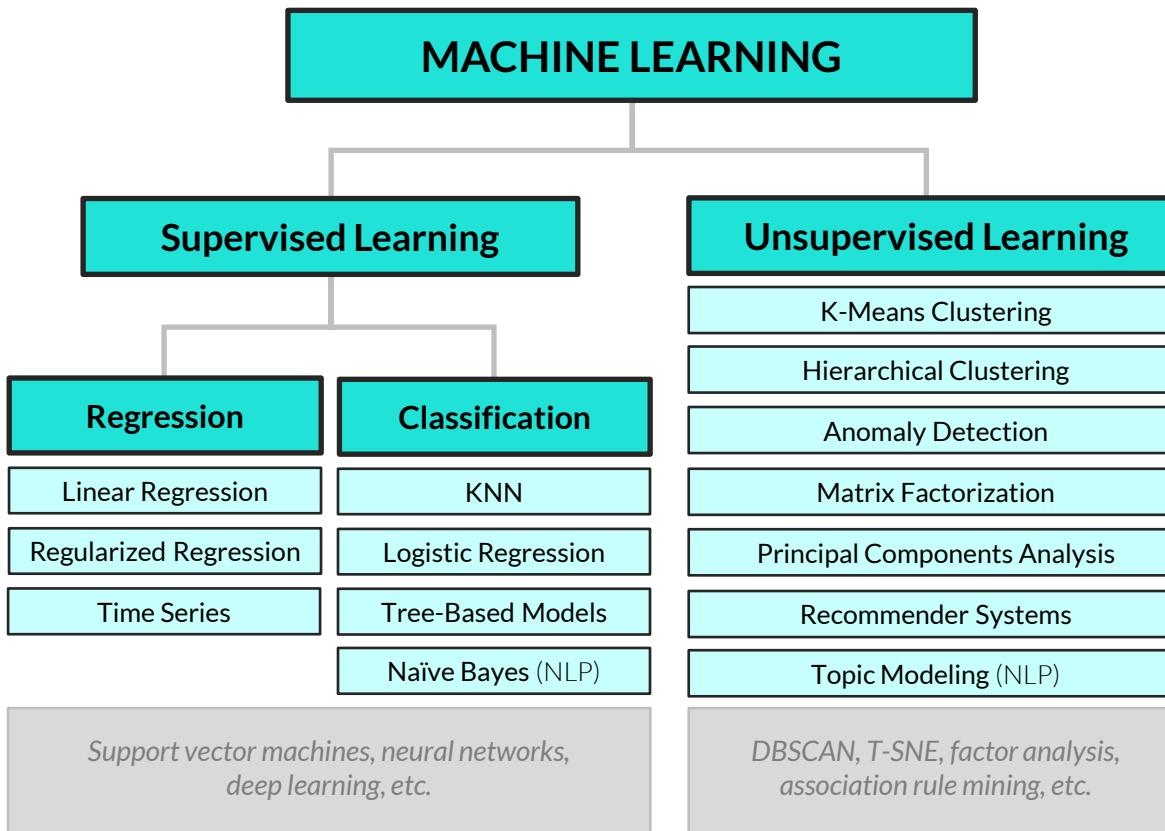
What is Data Science?

Essential Skills

What is Machine Learning?

Data Science Workflow

These are some of the most **common machine learning algorithms** that data scientists use in practice



Another category of machine learning is called **reinforcement learning**, which is more commonly used in robotics and gaming

Other fields like **deep learning** and **natural language processing** utilize both supervised and unsupervised learning techniques



DATA SCIENCE WORKFLOW

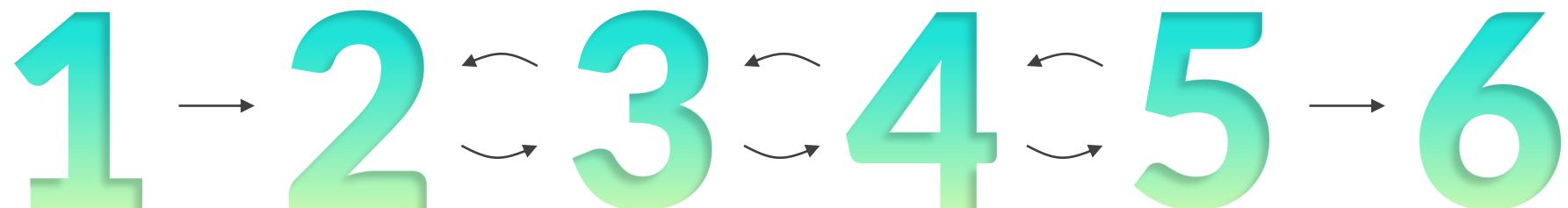
What is Data Science?

Essential Skills

What is Machine Learning?

Data Science Workflow

The **data science workflow** consists of scoping a project, gathering, cleaning and exploring the data, building models, and sharing insights with end users



This is not a linear process! You'll likely go back to further gather, clean and explore your data



STEP 1: SCOPING A PROJECT

What is Data Science?

Essential Skills

What is Machine Learning?

Data Science Workflow



Projects don't start with *data*, they start with a **clearly defined scope**:

- Who are your end users or stakeholders?
- What business problems are you trying to help them solve?
- Is this a supervised or unsupervised learning problem? (*do you even need data science?*)
- What data do you need for your analysis?



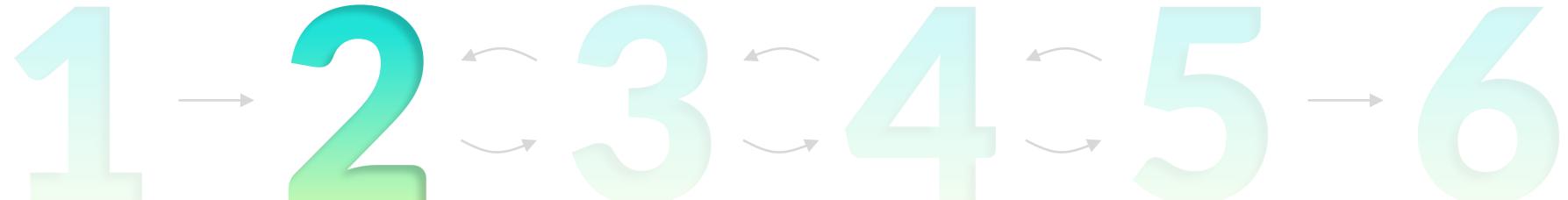
STEP 2: GATHERING DATA

What is Data Science?

Essential Skills

What is Machine Learning?

Data Science Workflow



Scoping a Project

Gathering Data

Cleaning Data

Exploring Data

Modeling Data

Sharing Insights

A project is only as strong as the underlying data, so **gathering the right data** is essential to set a proper foundation for your analysis

Data can come from a variety of sources, including:

- Files (flat files, spreadsheets, etc.)
- Databases
- Websites
- APIs



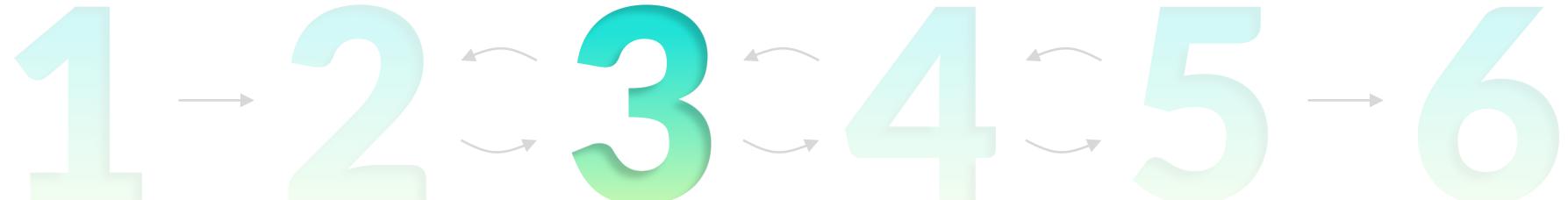
STEP 3: CLEANING DATA

What is Data Science?

Essential Skills

What is Machine Learning?

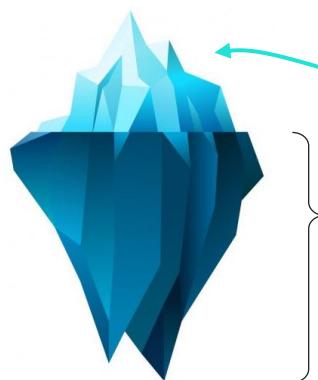
Data Science Workflow



A popular saying within data science is “garbage in, garbage out”, which means that **cleaning data** properly is key to producing accurate and reliable results

Data cleaning tasks may include:

- Correcting data types
- Imputing missing data
- Dealing with data inconsistencies
- Reformatting the data



Building models
The flashy part of data science

Cleaning data
Less fun, but very important
(Data scientists estimate that around 50-80% of their time is spent here!)



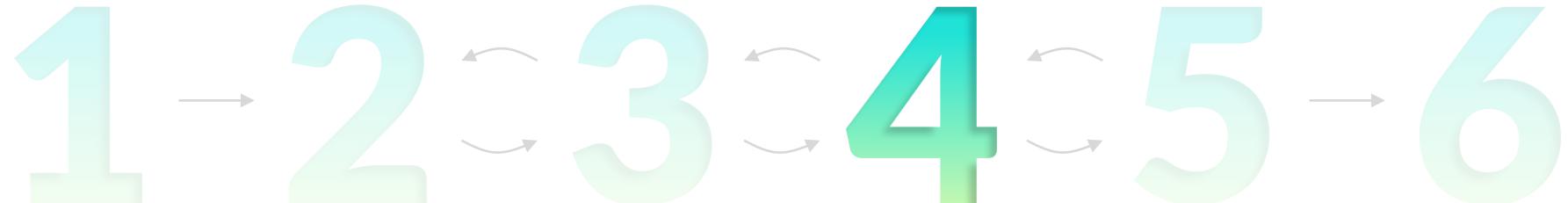
STEP 4: EXPLORING DATA

What is Data Science?

Essential Skills

What is Machine Learning?

Data Science Workflow



Scoping a Project

Gathering Data

Cleaning Data

Exploring Data

Modeling Data

Sharing Insights

Exploratory data analysis (EDA) is all about exploring and understanding the data you're working with before building models

EDA tasks may include:

- Slicing & dicing the data
- Summarizing the data
- Visualizing the data



A good number of the **final insights** that you share will come from the exploratory data analysis phase!



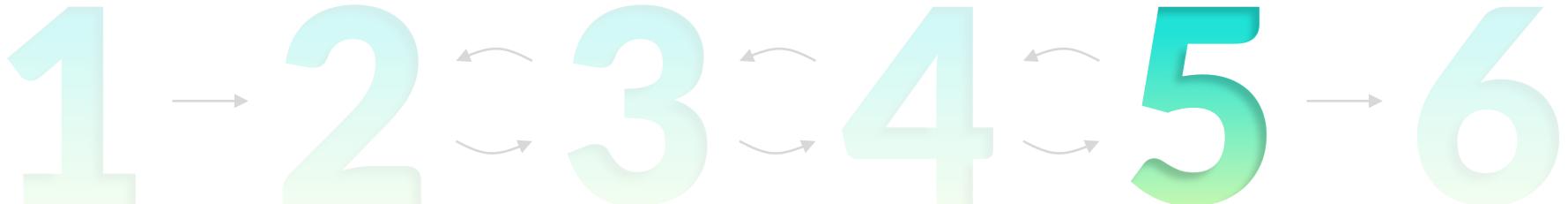
STEP 5: MODELING DATA

What is Data Science?

Essential Skills

What is Machine Learning?

Data Science Workflow



Scoping a Project

Gathering Data

Cleaning Data

Exploring Data

Modeling Data

Sharing Insights

Modeling data involves structuring and preparing data for specific modeling techniques, and applying algorithms to make predictions or discover patterns

Data modeling tasks may include:

- Restructuring the data
- Feature engineering (*adding new fields*)
- Applying machine learning algorithms



With fancy new algorithms introduced every year, you may feel the need to learn and apply the latest and greatest techniques

In practice, **simple is best**; businesses & leadership teams appreciate solutions that are easy to understand, interpret and implement



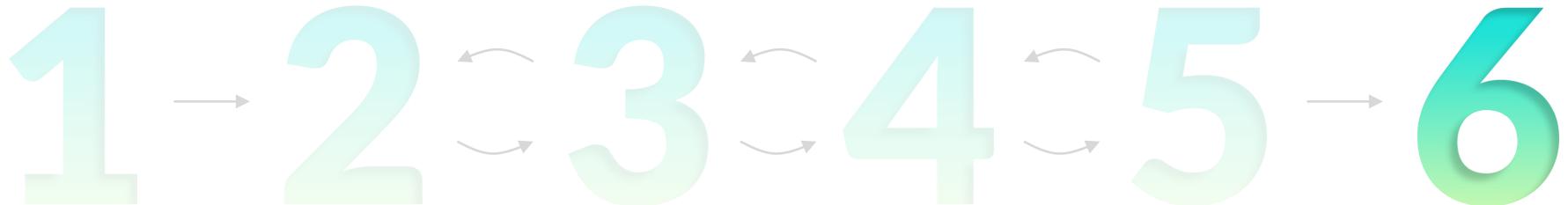
STEP 6: SHARING INSIGHTS

What is Data Science?

Essential Skills

What is Machine Learning?

Data Science Workflow



Scoping a Project

Gathering Data

Cleaning Data

Exploring Data

Modeling Data

Sharing Insights

The final step of the workflow involves summarizing your key findings and **sharing insights** with end users or stakeholders:

- Reiterate the problem
- Interpret the results of your analysis
- Share recommendations and next steps
- Focus on potential impact, not technical details



Even with all the technical work that's been done, it's important to remember that the focus here is on **non-technical solutions**

NOTE: Another way to share results is to deploy your model, or put it into production



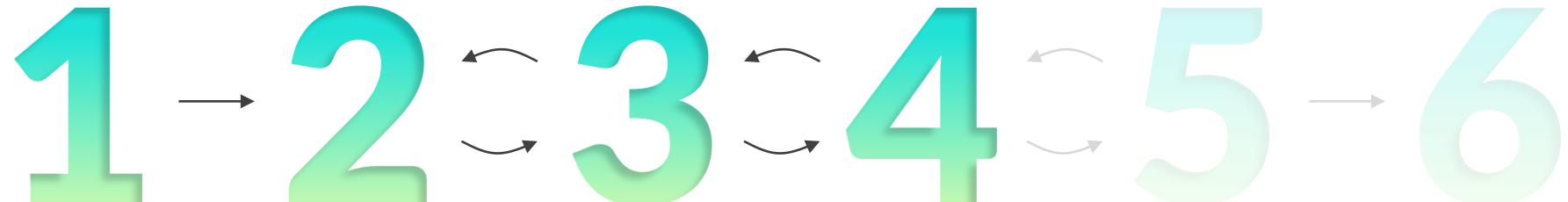
DATA PREP & EDA

What is Data Science?

Essential Skills

What is Machine Learning?

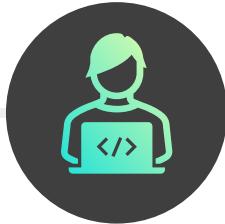
Data Science Workflow



DATA PREP & EDA

REGRESSION, CLASSIFICATION,
UNSUPERVISED LEARNING, NLP

Data prep and EDA is a critical part of every data science project, and should always come first before applying machine learning algorithms



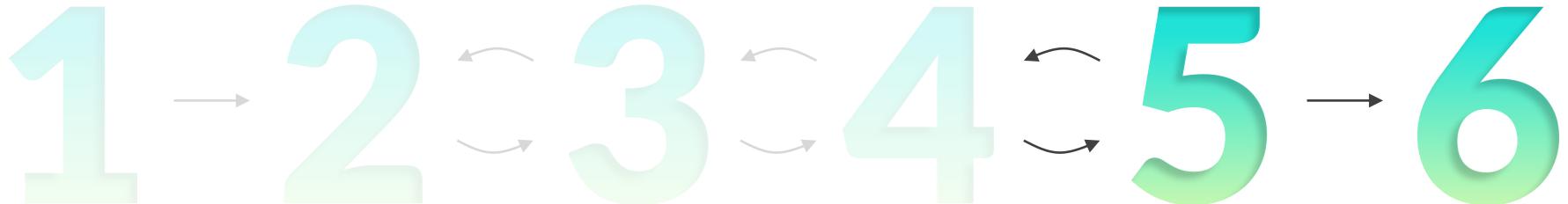
REGRESSION MODELING

What is Data Science?

Essential Skills

Machine Learning

Data Science Workflow



Scoping a Project

Gathering Data

Cleaning Data

Exploring Data

Modeling Data

Sharing Insights

DATA PREP & EDA

REGRESSION

Regression models are used to predict the value of numeric variables

Even though regression models fall into the final two steps of the workflow, data prep & EDA should *always* come first to help you get the most out of your models



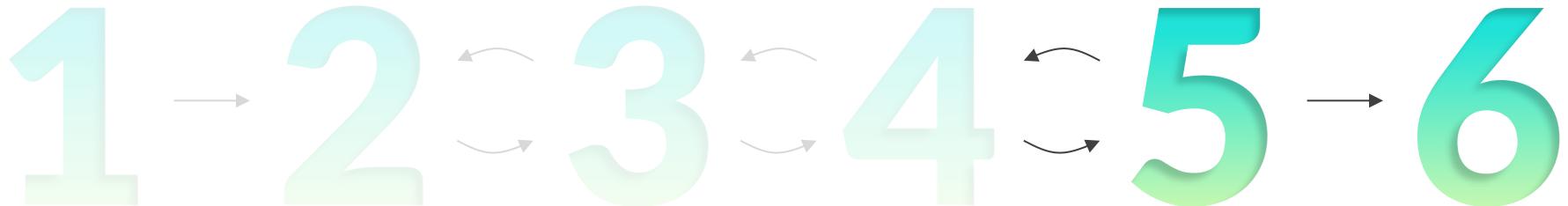
CLASSIFICATION MODELING

What is Data Science?

Essential Skills

Machine Learning

Data Science Workflow



DATA PREP & EDA

CLASSIFICATION

Classification models are used to predict the value of categorical variables

Even though classification models fall into the final two steps of the workflow, data prep & EDA should *always* come first to help you get the most out of your models

WHY PYTHON?

MEET PYTHON

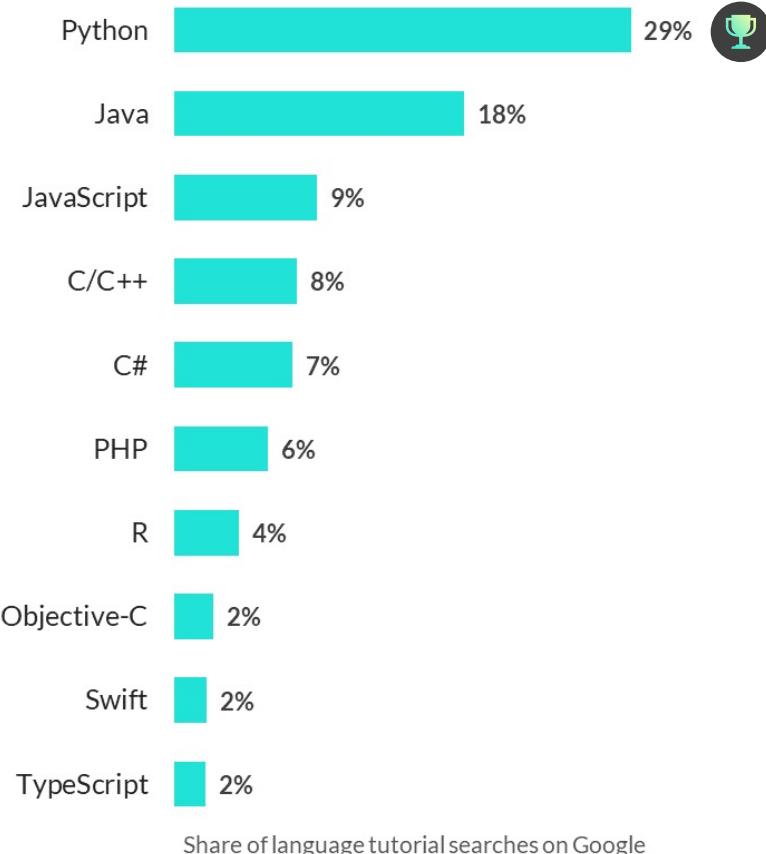


Python is a free, open-source programming language that is both powerful and easy to learn.

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):

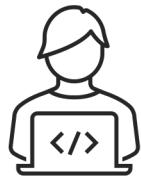


WHY PYTHON FOR ANALYTICS?



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 **base Python** 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



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 program



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](#)

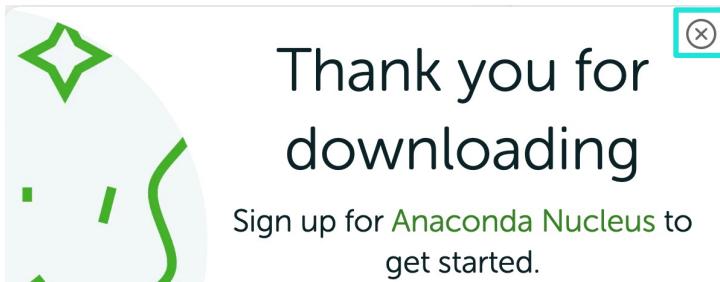
Individual Edition is now

ANACONDA DISTRIBUTION

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



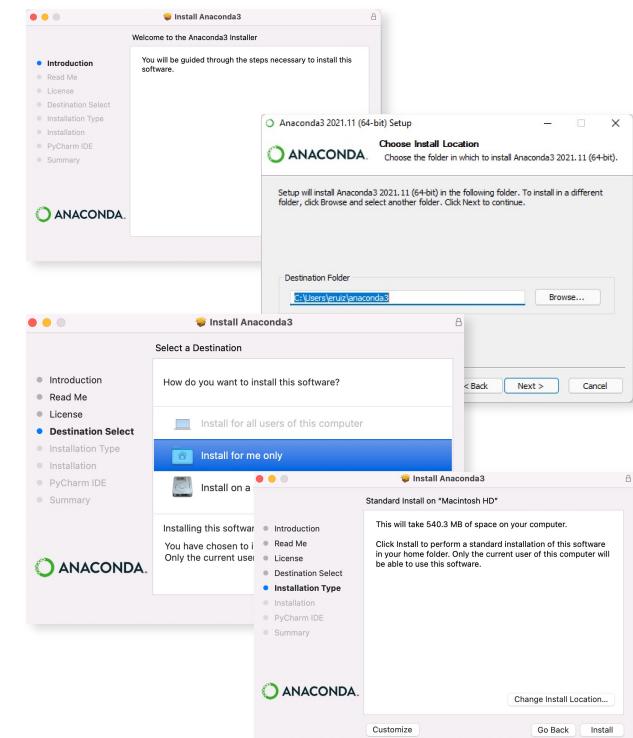
2) Click **X** on the Anaconda Nucleus pop-up
(no need to launch)



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

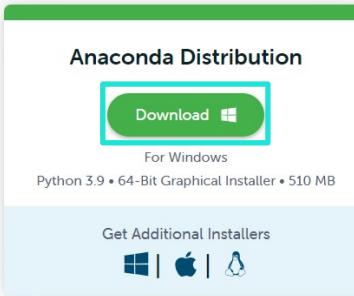
Helpful Resources

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

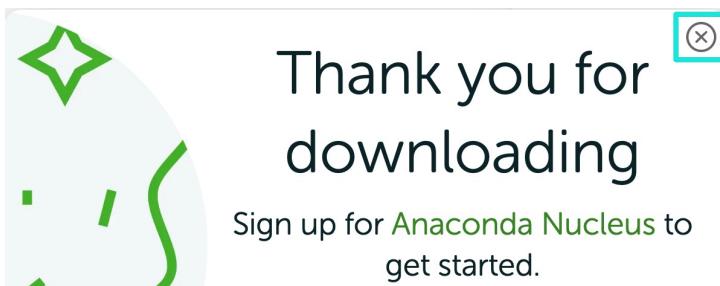
Individual Edition is now

ANACONDA DISTRIBUTION

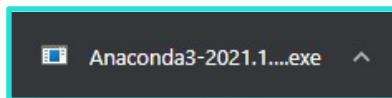
The world's most popular open-source Python distribution platform



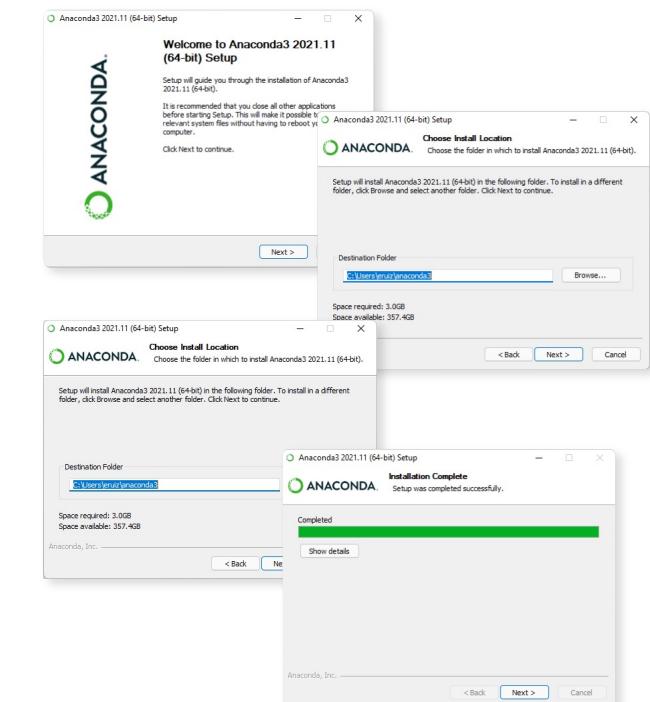
2) Click **X** on the Anaconda Nucleus pop-up
(no need to launch)



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



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





LAUNCHING JUPYTER

Installation &
Setup

Notebook
Interface

Comments &
Markdown

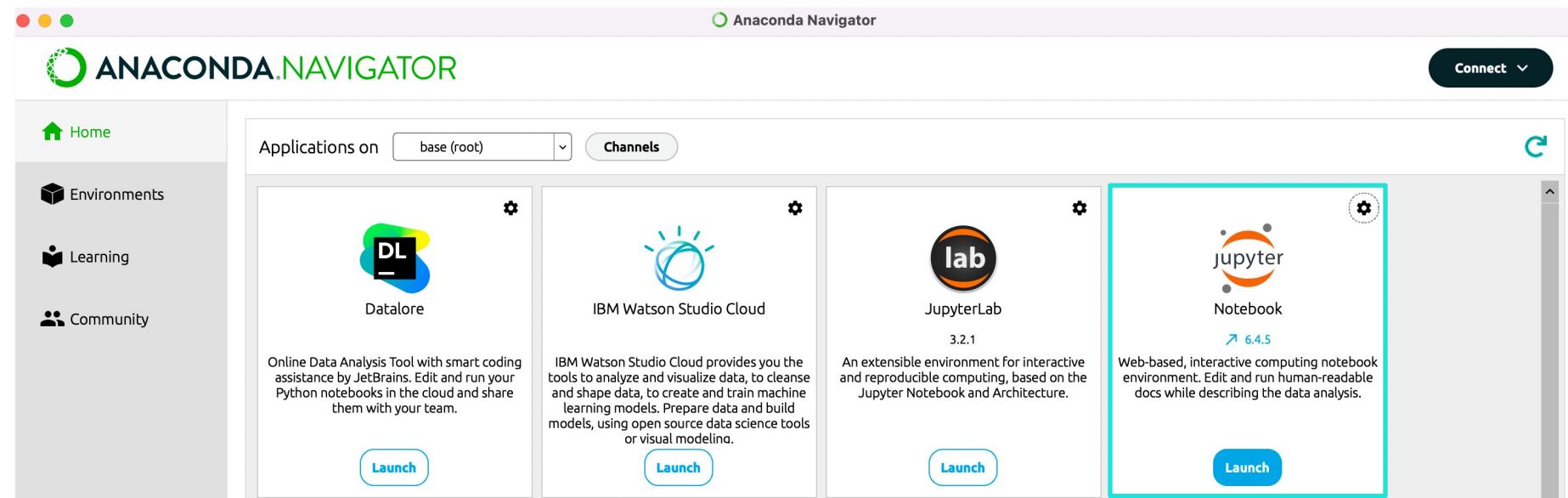
The Print
Function

Google Colab

Helpful
Resources

1) Launch **Anaconda Navigator**

2) Find **Jupyter Notebook** and click 



The screenshot shows the Anaconda Navigator interface. On the left, there's a sidebar with links for Home, Environments, Learning, and Community. The main area is titled "ANACONDA.NAVIGATOR" and shows a grid of application cards. One card for "Jupyter Notebook" is highlighted with a teal border. The card for Jupyter Notebook includes a brief description: "Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.", a version number "6.4.5", and a "Launch" button.



YOUR FIRST JUPYTER NOTEBOOK

Installation & Setup

Notebook Interface

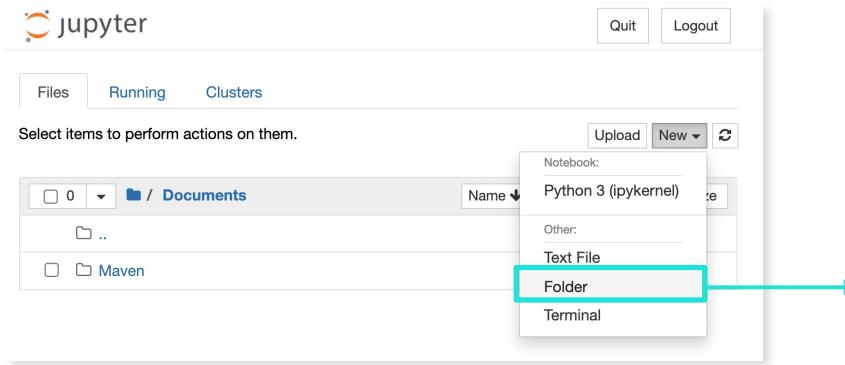
Comments & Markdown

The Print Function

Google Colab

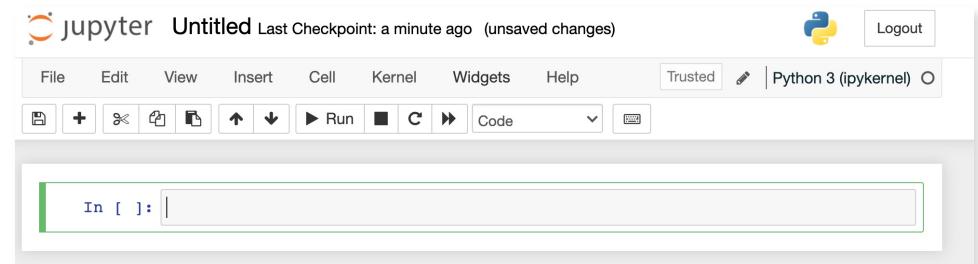
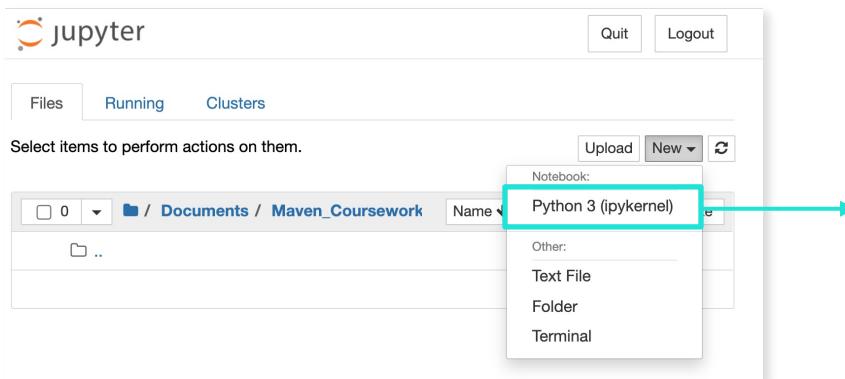
Helpful Resources

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

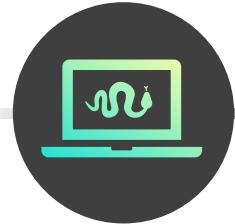


NOTE: You can rename your folder by clicking "Rename" in the top left corner

- 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

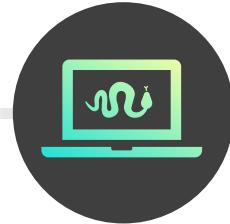
```
Last login: Tue Jan 25 14:04:12 on ttys002
(base) chrisb@Chriss-MBP ~ % jupyter notebook
[I 2022-01-26 08:45:53.886 LabApp] JupyterLab extension loaded from /Users/chrisb/opt/anaconda3/lib/python3.9/site-packages/jupyterlab
[I 2022-01-26 08:45:53.886 LabApp] JupyterLab application directory is /Users/chrisb/opt/anaconda3/share/jupyter/lab
[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/nbserver-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

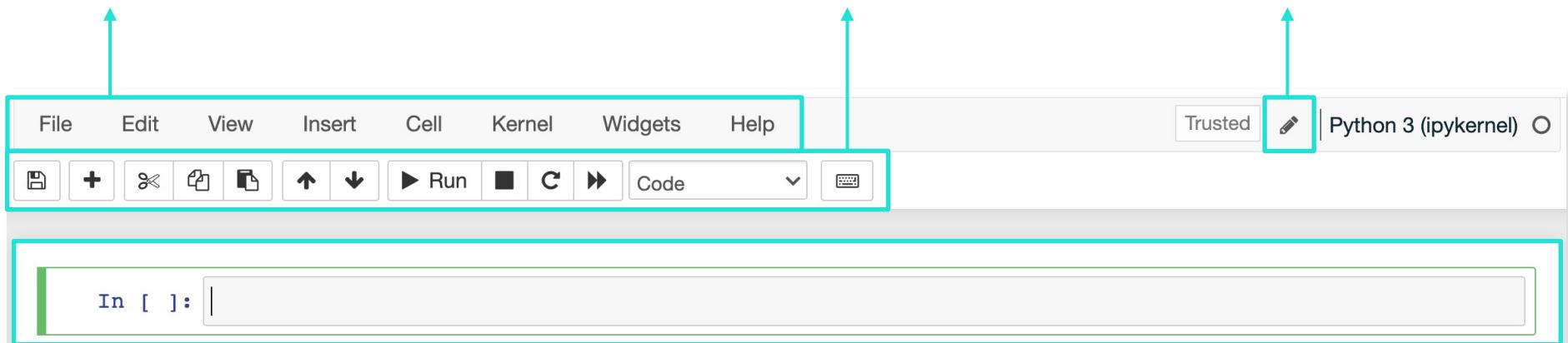
Options to manipulate the way
the notebook functions

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

File

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

- File Edit View Insert
- New Notebook ▶
- Open...
- Make a Copy...
- Save as...
- Rename...
- Save and Checkpoint ⌘S
- Revert to Checkpoint ▶
- Print Preview
- Download as ▶
- Trusted Notebook
- Close and Halt

Edit

Edit cells within your notebook (while in command mode)

- Edit View Insert
- Cut Cells X
- Copy Cells C
- Paste Cells Above ↑V
- Paste Cells Below V
- Paste Cells & Replace
- Delete Cells D, D
- Undo Delete Cells Z
- Split Cell ⌘-minus
- Merge Cell Above
- Merge Cell Below
- Move Cell Up
- Move Cell Down
- Edit Notebook Metadata
- Find and Replace

View

Edit cosmetic options for your notebook.

- View Insert Cell ▾
- Toggle Header
- Toggle Toolbar
- Toggle Line Numbers ↑L
- Cell Toolbar ▶

Insert

Insert new cells into your notebook

- Insert Cell Kernel
- Insert Cell Above A
- Insert Cell Below B

MENU OPTIONS



Installation &
Setup

Notebook
Interface

Comments &
Markdown

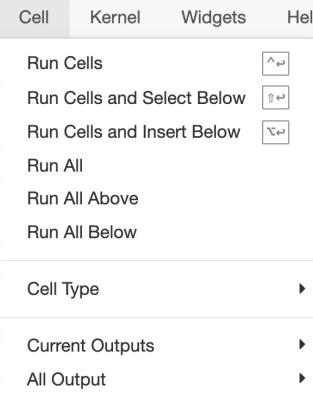
The Print
Function

Google Colab

Helpful
Resources

Cell

Access options for
running the cells in your
notebook



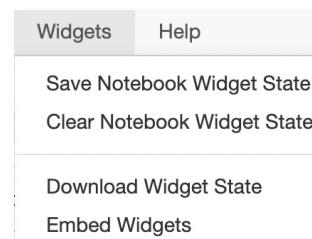
Kernel

Interact with the
instance of Python that
runs your code



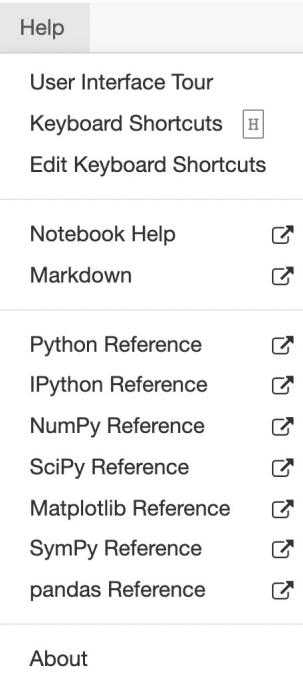
Widgets

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

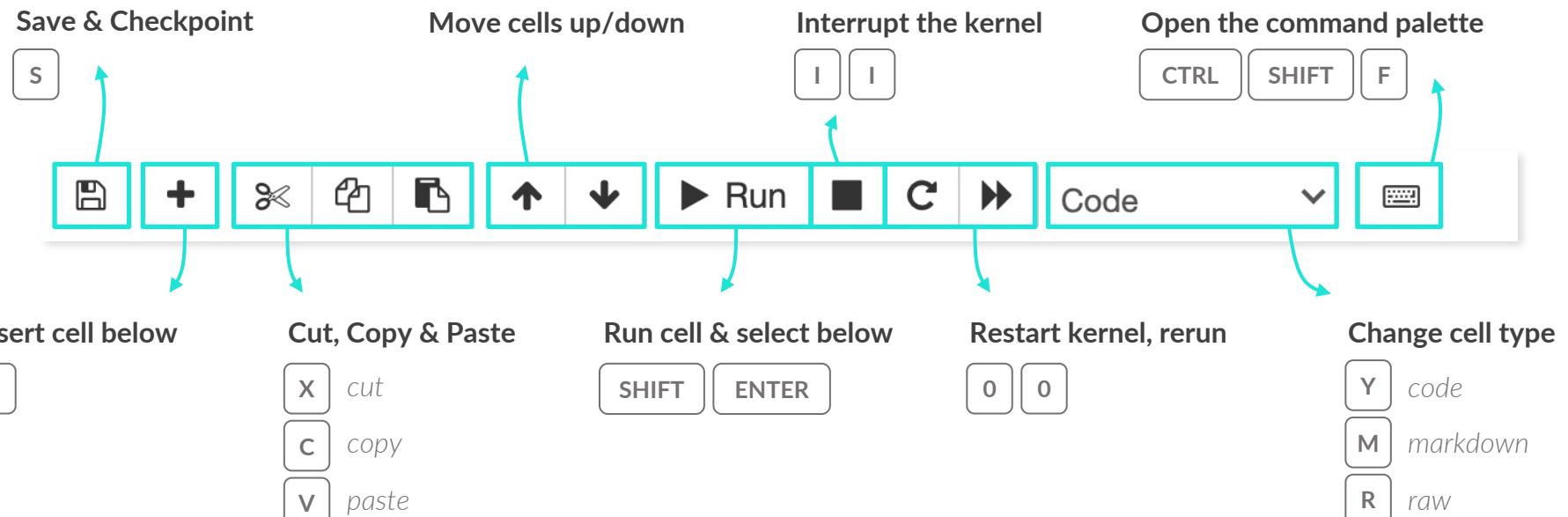
The Print
Function

Google Colab

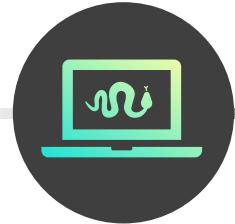
Helpful
Resources

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

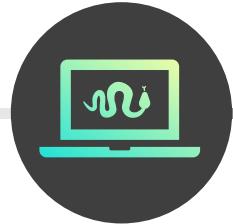
Helpful
Resources

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

A screenshot of the Jupyter Notebook interface in Edit mode. The top navigation bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. On the right, there are buttons for Trusted, a pen icon, and Python 3 (ipykernel). Below the menu is a toolbar with icons for file operations like Open, Save, and New, along with Run, Cell, and Code selection. A code cell is visible, starting with "In []: |". The entire cell area is highlighted with a green border.

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

A screenshot of the Jupyter Notebook interface in Command mode. The top navigation bar and toolbar are identical to the Edit mode screenshot. A code cell is visible, starting with "In []: |". The entire cell area is highlighted with a blue border.



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

This screenshot shows a Jupyter Notebook interface. At the top, the menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. To the right of the menu is a status bar showing 'Trusted' and 'Python 3 (ipykernel)'. Below the menu is a toolbar with various icons for file operations like opening, saving, and deleting, as well as cell execution options like Run, Kernel, and Cell Type. A green arrow points from the text above to the 'In []:' input cell, which is highlighted with a green border. Inside the cell, there is a small cursor icon.

In **edit mode**, the cell will be highlighted **green** and a pencil icon will appear

This screenshot shows a Jupyter Notebook interface after running some code. The 'Run' button in the toolbar is highlighted with a red box and a blue arrow pointing down to it from the text below. Below the toolbar, the 'In [1]:' cell contains the code '5 + 5'. Underneath it, the 'Out[1]:' cell displays the result '10'. The status bar at the bottom right shows 'Trusted' and 'Python 3 (ipykernel)'.

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

- **In []:** Our code (*input*)
- **Out[]:** What our code produced (*output*)*

*Note: not all code has an 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

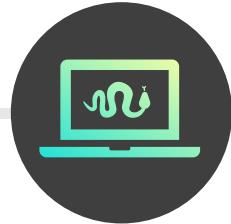
In [2]: `5 + 5`

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

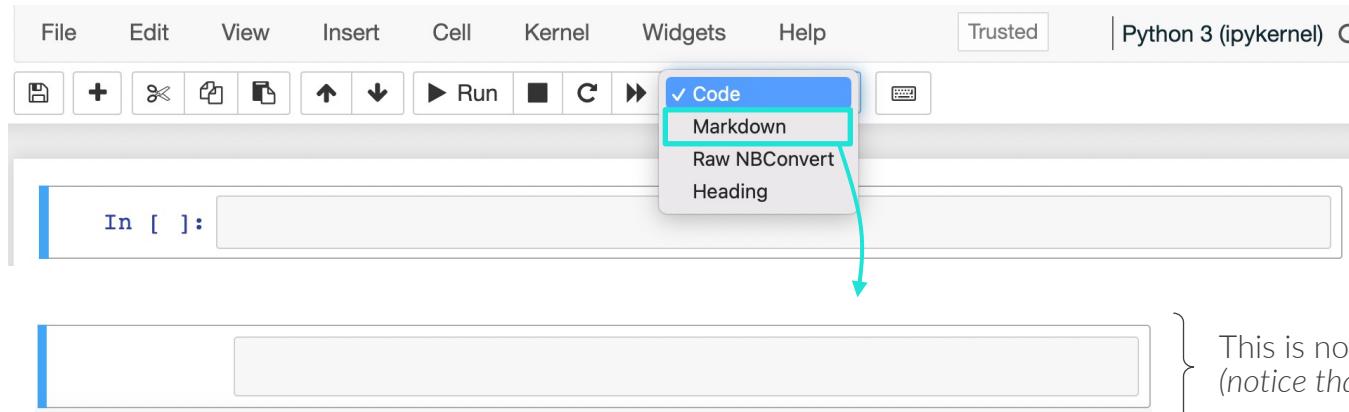
Google Colab

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)
2. Select “**Markdown**” in the cell menu (or press **M**)





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 *.

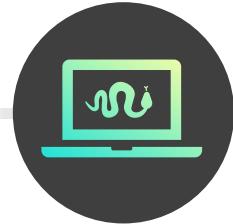
Numbered lists can be created by beginning a line with 1., 2., and so on.

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.

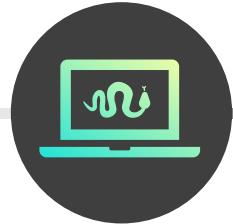
The Essentials to get started are:

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 **Out[16]**: as a printed output is
different from the standard output returned by Python



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 screenshots illustrate the workflow for creating a new Colab notebook. The top image shows the Colab dashboard with recent notebooks. The bottom image shows the Google Drive interface where a new notebook is being created, with the 'New notebook' button highlighted.



HELPFUL RESOURCES

Installation & Setup

Notebook Interface

Comments & Markdown

The Print Function

Google Colab

Helpful Resources

why is my output not printing python

In Python what is it called when you see the output of a variable without printing it?

Built-in Functions

Creating a better dashboard with Python, Dash, and Plotly

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



Jupyter Notebooks are user-friendly coding environments

- Jupyter notebooks 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 markdown cells 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 Jupyter Notebooks

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