What Is Data Science?

(aka what is this buzzword anyway?)

Today’s Learning Objectives

- List a few skills that all conglomerate into data science
- Explain the types of problems that data science can solve
- Describe the data science process

Find out for yourself:

Pick one, read through it - then we'll discuss!

1. Battle of the Data Science Venn Diagrams (blog post, 2 pages)
2. The 5 Questions Data Science Can Answer (5 min video)
3. Preparing for the Transition to Data Science (blog post)
4. How to Become a (Good) Data Scientist (blog post)
5. The Data Science Process (blog post)

Let’s discuss!

Across these articles/blogs/videos, what are the main skills you need to be a data scientist?

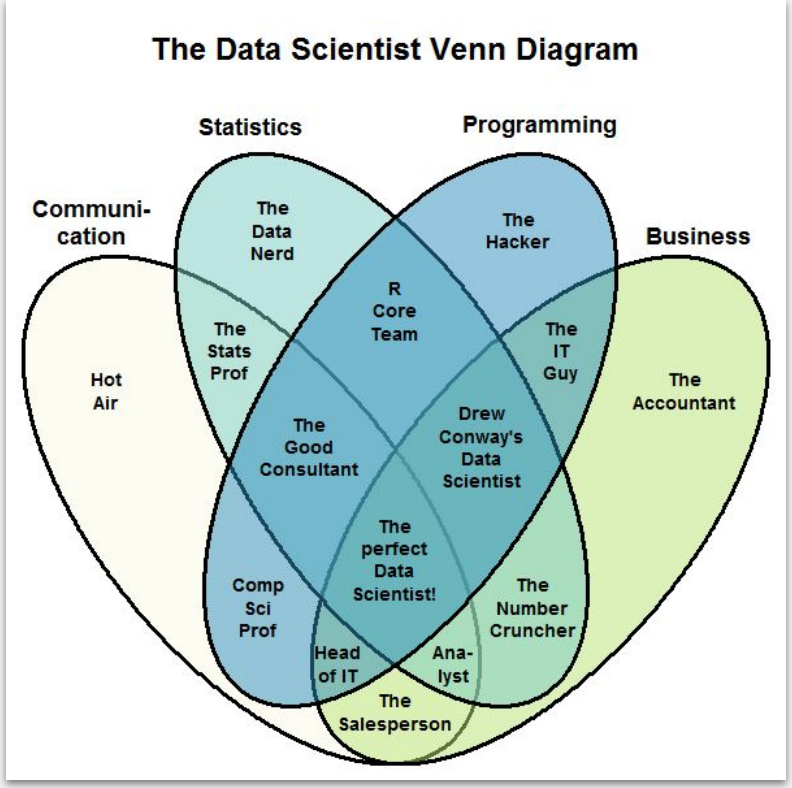
What things are in dispute? In other words, what isn’t consistent across these sources, or what do some talk about that others don’t?

What does a ‘data scientist’ even do?

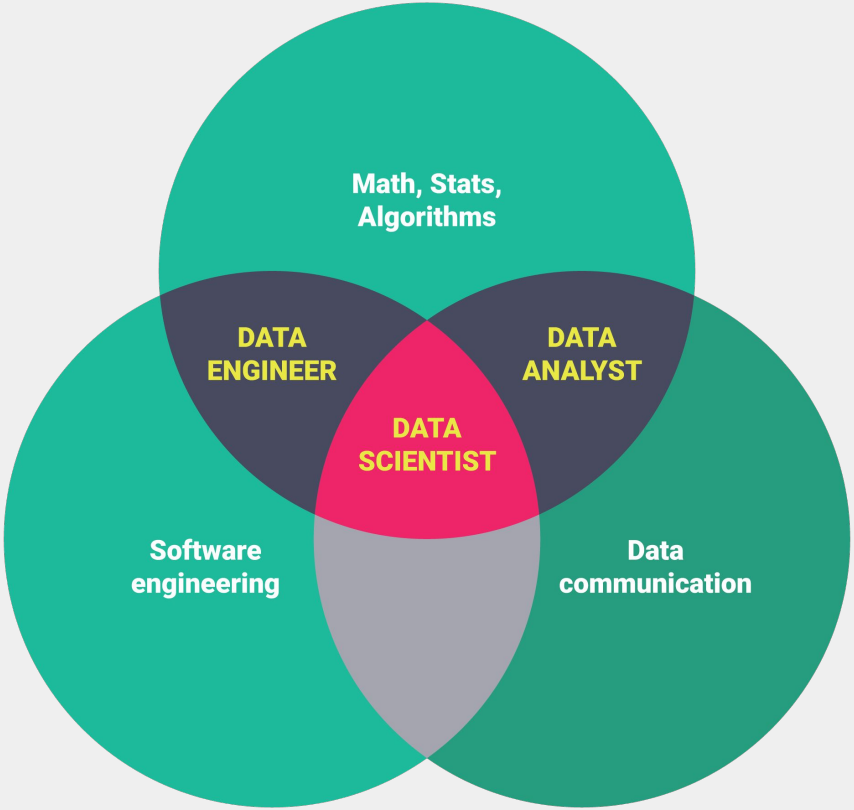
“A data scientist does model-driven analyses of our data; analyses to improve our planning, increase our productivity, and develop our deeper levels of subject matter expertise. A data scientist works at the tactical, operational, and strategic levels, sharing insights with the business.”

— Chris Pehura, Practice Director, Management Consultant at C-SUITE DATA

Overview

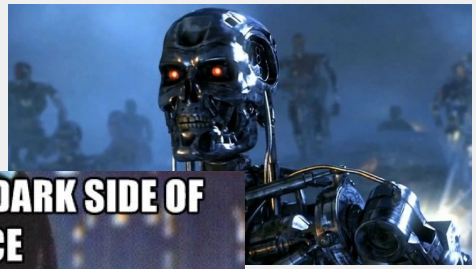

The Data Scientist Venn Diagram
Statistics
Programming
Communi-
cation
Business
The
Data
Nerd
The
Hacker
R
Core
Team
The
Stats
Prof
The
IT
Guy
Hot
Air
The
Accountant
The
Good
Consultant
Drew
Conway's
Data
Scientist
The
perfect
Data
Scientist!
Comp
Sci
Prof
The
Number
Cruncher
Head
of IT
Ana-
lyst
The
Salesperson

More simply...

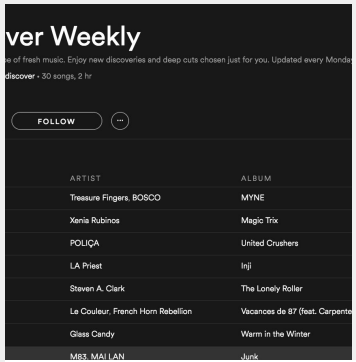

Math, Stats,
Algorithms
DATA
ENGINEER
DATA
ANALYST
DATA
SCIENTIST
Software
engineering
Data
communication

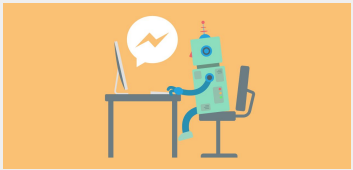
More specifically…

Not this (mostly…)



Yes this



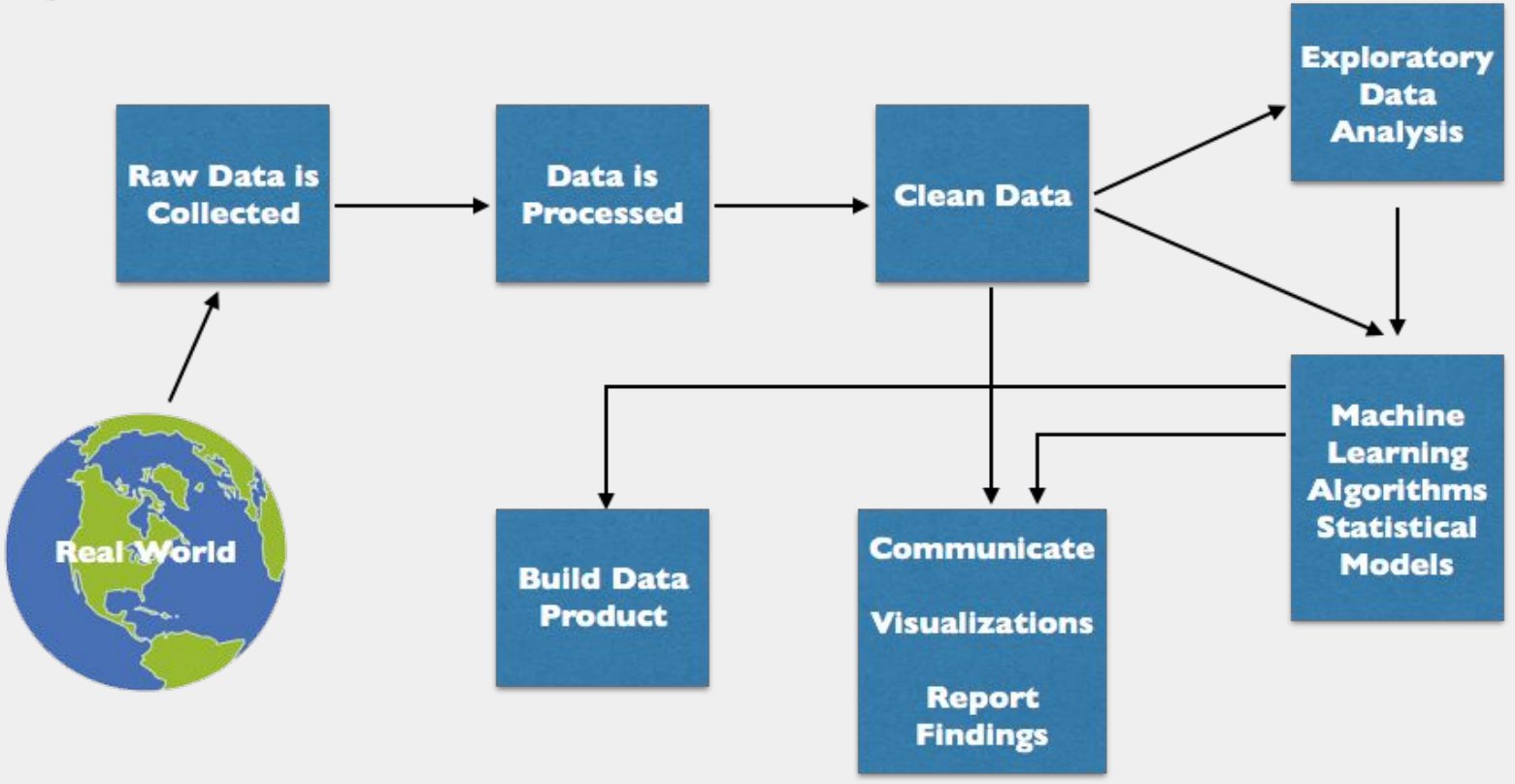


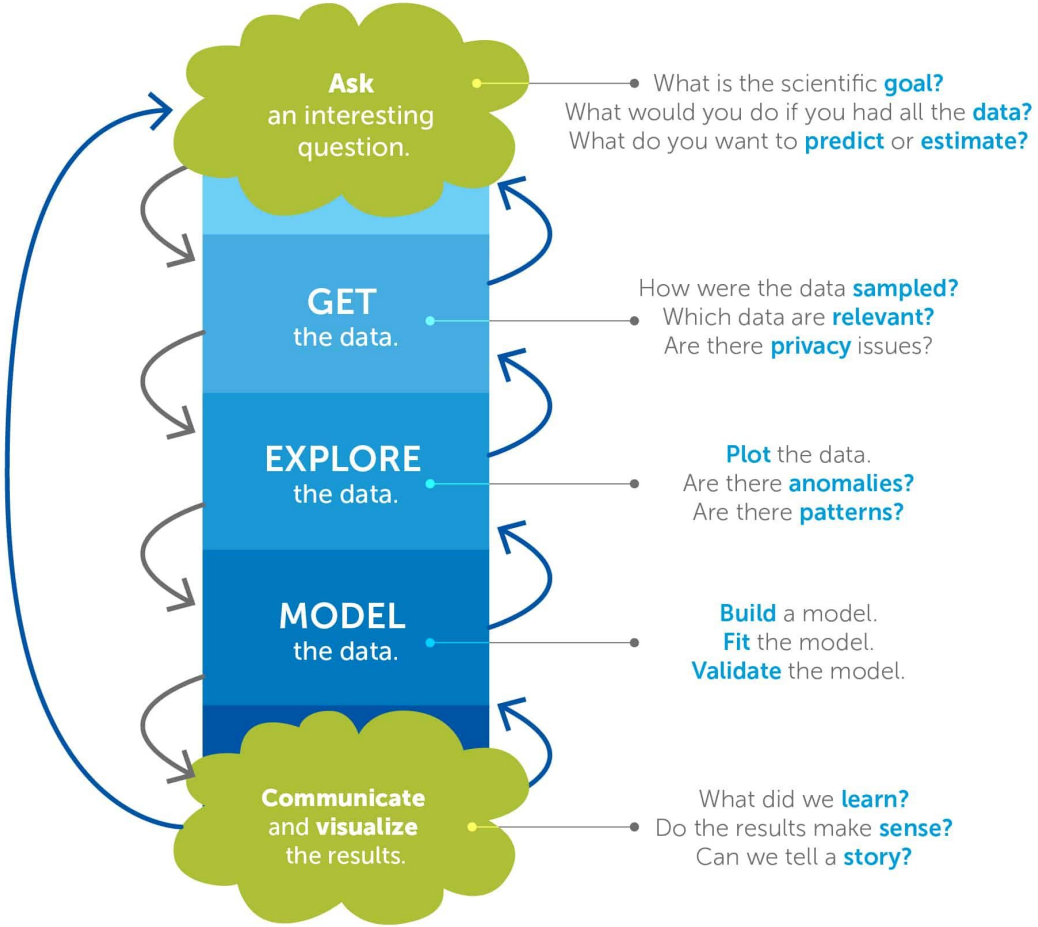
It might surprise you, but there are only five questions that data science answers:

- **Is this A or B?**
- **Is this weird?**
- **How much – or – How many?**
- **How is this organized?**
- **What should I do next?**

— Microsoft's ML Studio:
Data Science for Beginners

Data Science Process

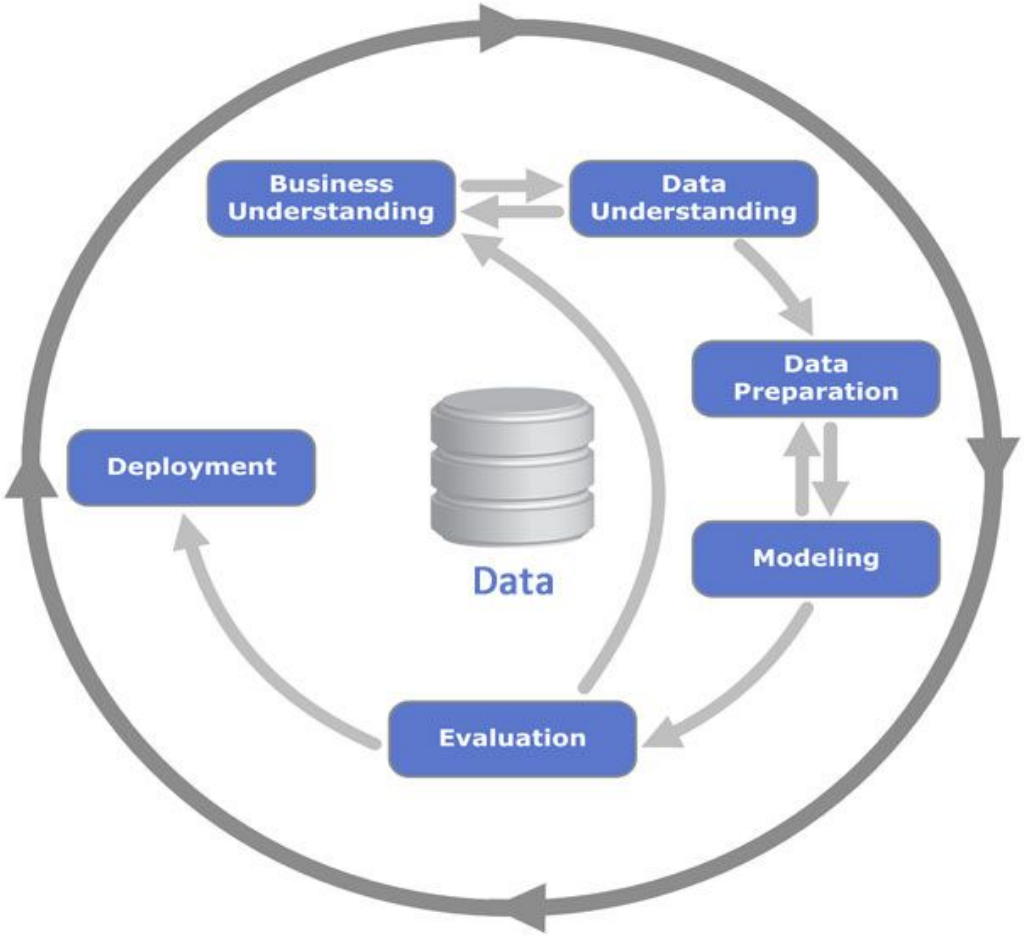




Derived from the work of Joe Blitzstein and Hanspeter Pfister,
originally created for the Harvard data science course http://cs109.org/.

Data Science Deconstructed

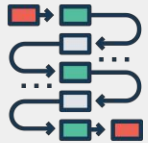
CRISP-DM Process Diagram



Source: Kenneth Jensen

Data Science Process

LEAD



OBTAIN	SCRUB	EXPLORE	MODEL	INTERPRET
O	**S**	**E**	**M**	**N**
Gather data from relevant sources	Clean data to formats that machine understands	Find significant patterns and trends using statistical methods	Construct models to predict and forecast	Put the results into good use

Originally by Hilary Mason and Chris Wiggins

Takeaways

- Question First versus Data First
- Data Cleaning/Preparation/Processing/Exploration is key
- Communication (of question and of results) is also key
- Have to ask questions that models/machines can answer
- The process is iterative: you don’t just finish a step, but go back and forth repeatedly


And now for something completely different.

What is *Python?*

A coding language used extensively by data scientists

Easter Egg

```
In [1]: 1 import this
```

```
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

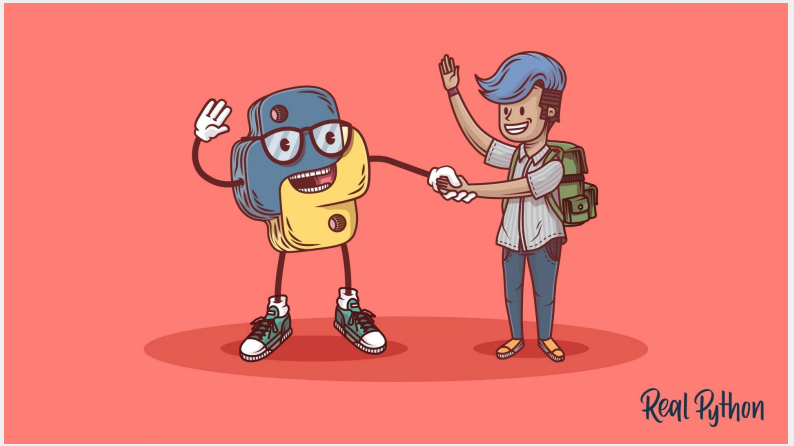
PEP 8

Python

“Python, named after the British comedy group Monty Python, is an interpreted, interactive, object-oriented programming language.

Its flexibility allows it to do many things, both big and small.

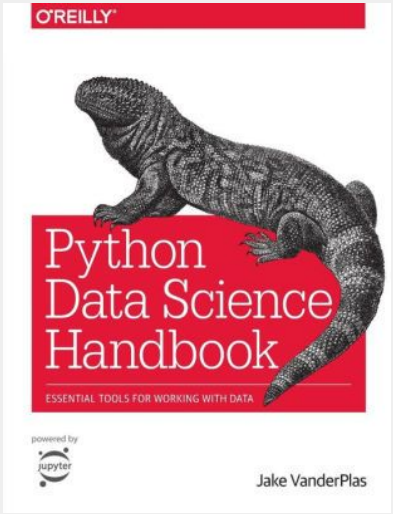
Python can be used to write simple programs, but it also possesses the full power required to create complex, large-scale enterprise solutions.” - Derrick Kearney



Python for Data Science

“The usefulness of Python for data science stems primarily from the large and active ecosystem of third-party packages:

- *NumPy* for manipulation of homogeneous array-based data;
- *Pandas* for manipulation of heterogeneous and labeled data;
- *SciPy* for common scientific computing tasks;
- *Matplotlib* for publication-quality visualizations;
- *[Jupyter]* for interactive execution and sharing of code;
- *Scikit-Learn* for machine learning, and many more tools...”
  - Jake VanderPlas



What is ***Anaconda?***

“The open-source Anaconda Distribution is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 15 million users worldwide, it is the industry standard for...enabling individual data scientists to:

- *Quickly download 1,500+ Python/R data science packages*
- *Manage libraries, dependencies, and environments with Conda”*

— *Anaconda Distribution*
- Package List

Anaconda (Conda)

- Conda is an open source package management system and environment management system that runs on Windows, macOS and Linux.
- Conda quickly installs, runs and updates packages and their dependencies.
- Conda easily creates, saves, loads and switches between environments on your local computer.
- You'll create conda environments to share, collaborate on, and reproduce projects with specific versions of particular packages.

What is ***Jupyter?***

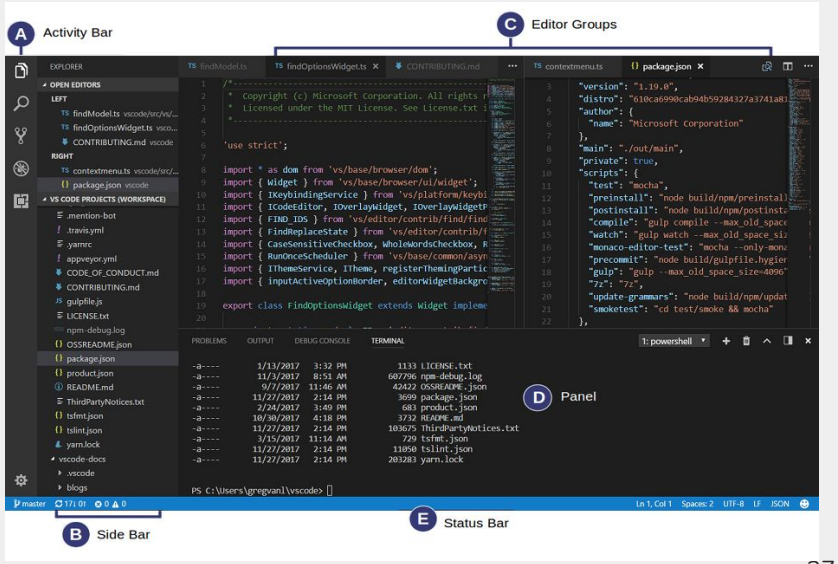
Jupyter

- Project Jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.
- Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text.
  - Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.
- JupyterLab is a next-generation web-based user interface
- Share notebooks using nbviewer

What is ***Visual Studio Code***?

Visual Studio (VS) Code

- Visual Studio Code is an open-source text editor created by Microsoft
- Navigate directory structure, make/ remove files, and direct access to the Terminal/Command Line
- Allows you to write text files (`.py`, `README.md`, etc.) and recently, [VS Code allows you to edit Jupyter Notebooks directly]
- Easy to switch between conda environments and lint code



Pycharm: An alternative to VS Code

- Open source IDE (Integrated Development Environment) developed specifically for Python
- Powerful but heavy. Long load time and memory consumption
- Community edition is free
- Specialized features like support for django

Choose the tools that work for you

What is ***Git***?

Git

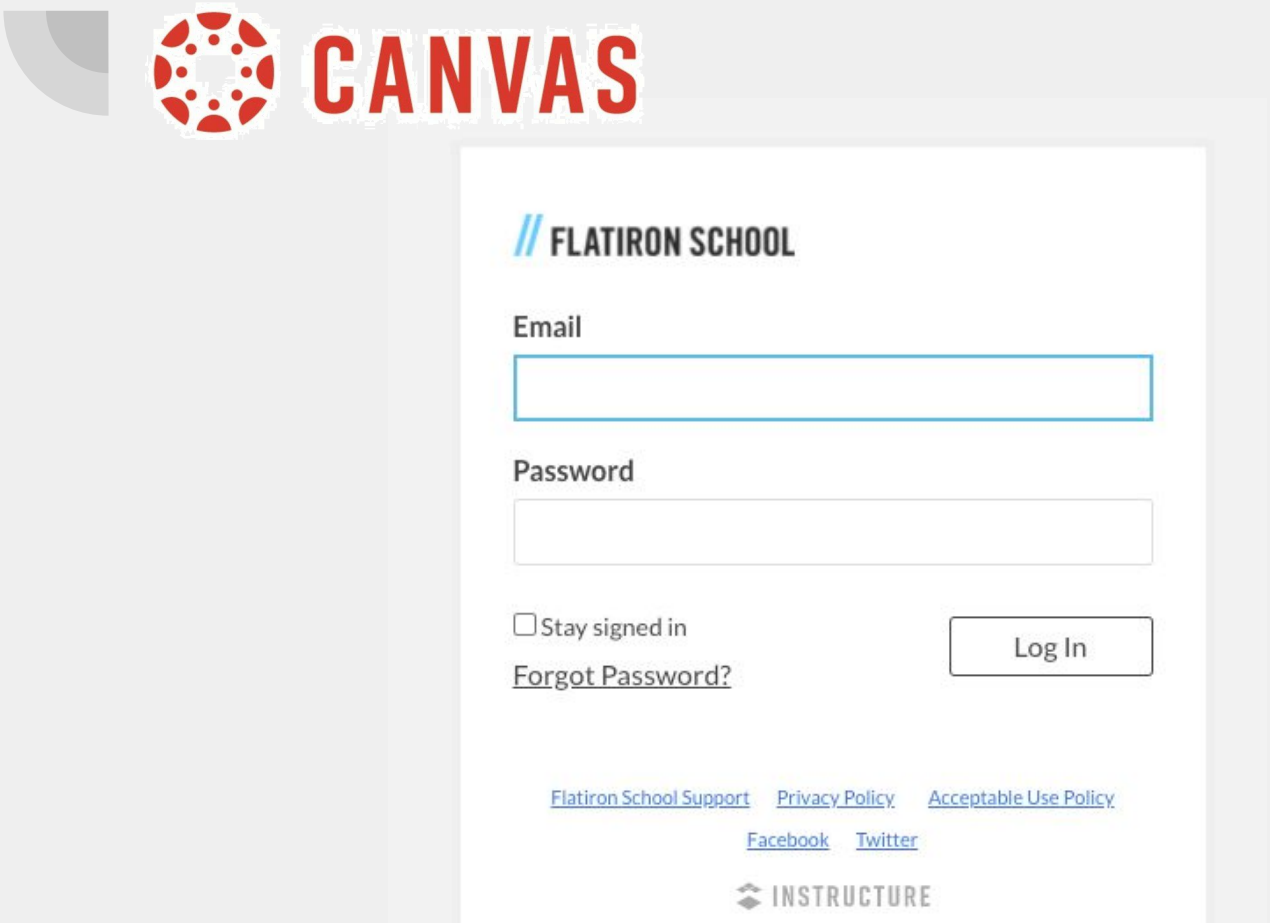
- Git is a version control system.
- It’s a way of keeping track of all the changes made across your project.
- Think of it like “track changes” in Word - but with the ability to track changes across multiple documents.

What is ***GitHub***?

GitHub

- GitHub is a free software platform that hosts over 40 million developers code
- You'll primarily use GitHub to collaborate with others, document your projects, and build your portfolio to showcase your abilities as a data scientist
- You can also use GitHub for any of the following tasks:
  - Code hosting
  - Code review
  - Project management
  - Team management
  - Documentation

Flatiron Systems: Canvas & IllumiDesk

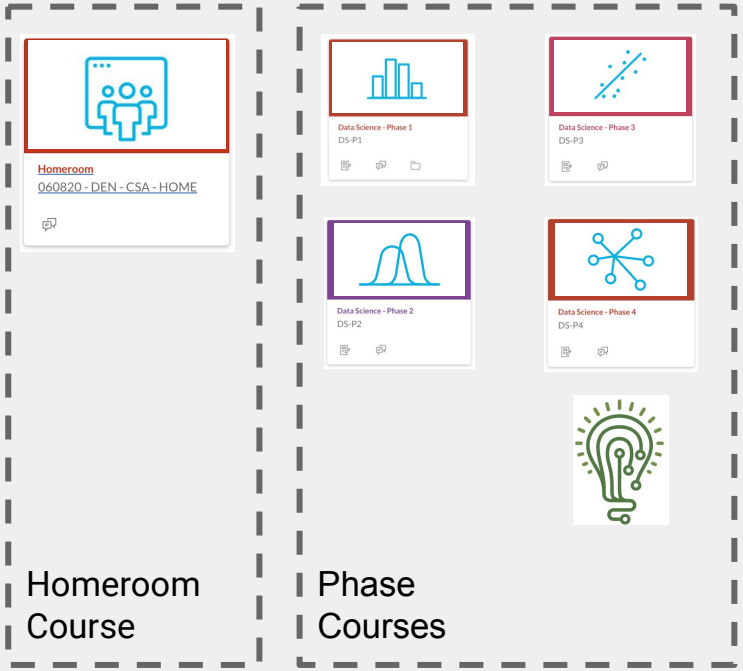

CANVAS
FLATIRON SCHOOL
Email
Password
Stay signed in
Forgot Password?
Log In
Flatiron School Support
Privacy Policy
Acceptable Use Policy
Facebook
Twitter
INSTRUCTURE

Two groups of Canvas courses

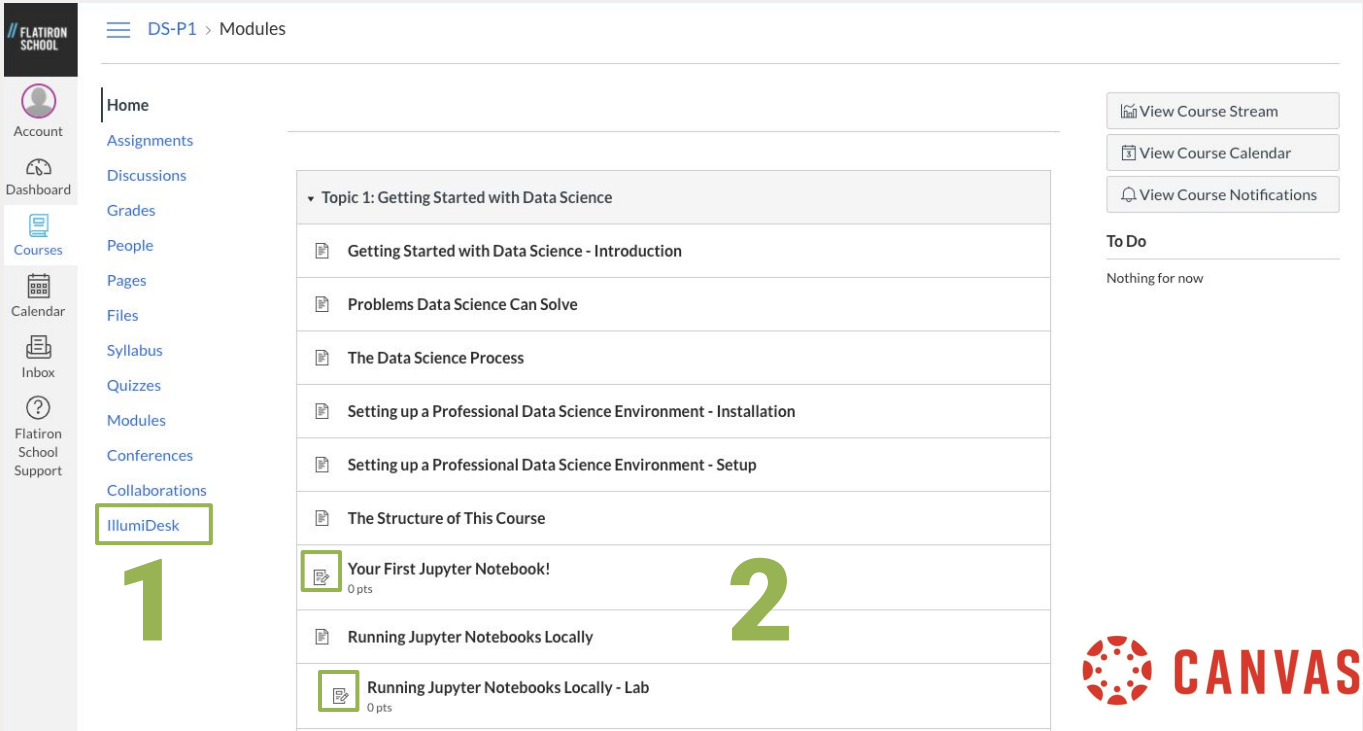
illumiDesk

Is used in the Phase courses

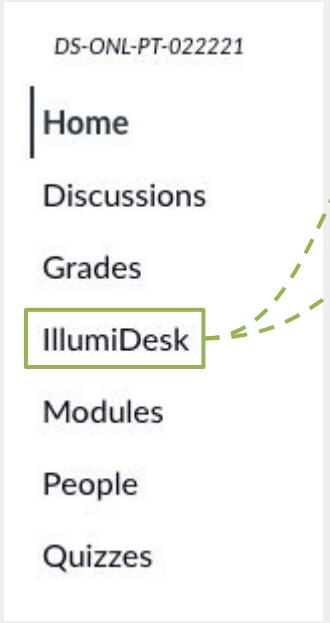
CANVAS

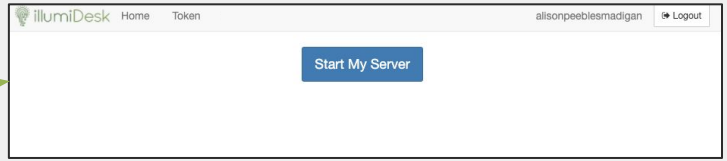


Two ways to access



IllumiDesk through the IllumiDesk link



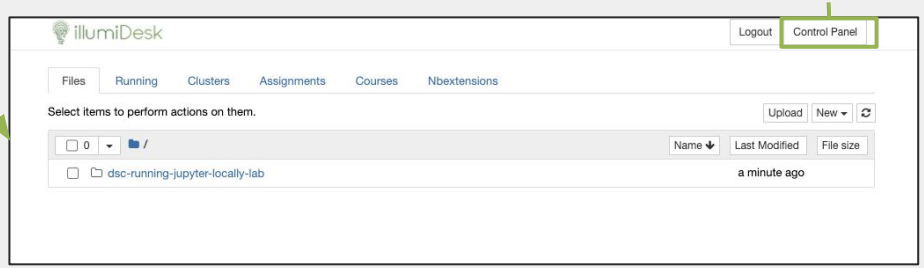


First time you access IllumiDesk in a Canvas session, the you will need to start your server at the Control Panel

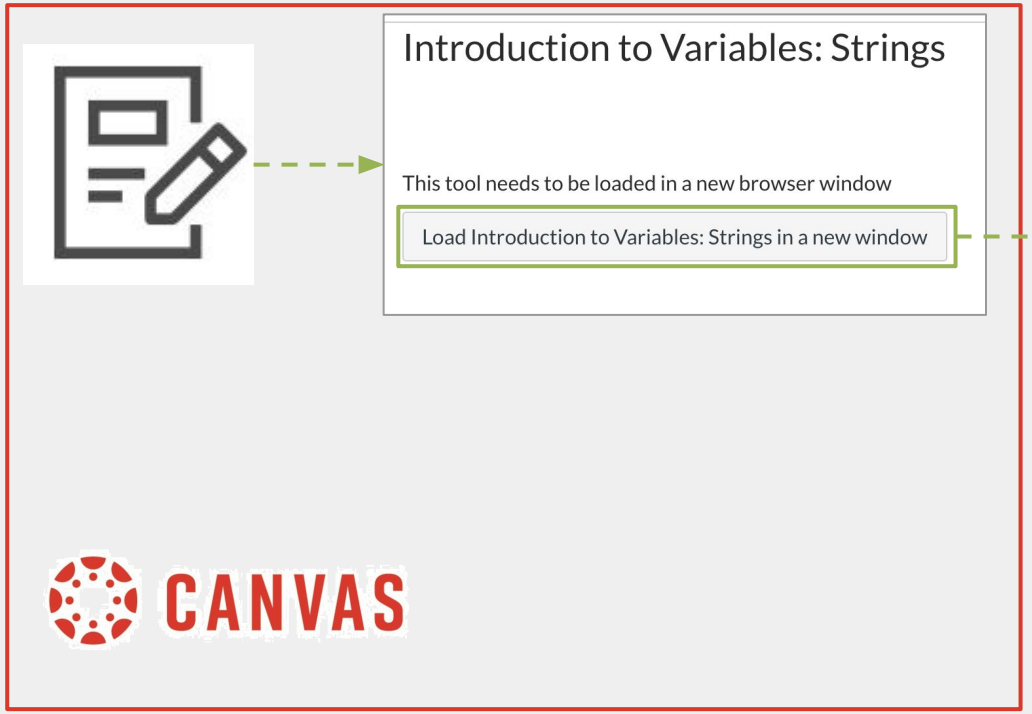
After starting the server the Control Panel has multiple options

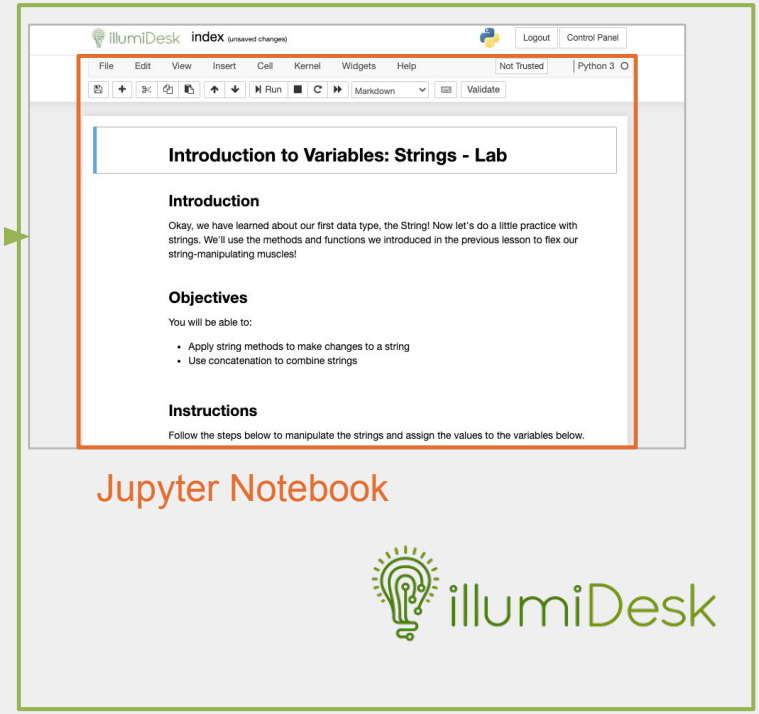
Either blue button will direct you to your Jupyter environment

The Control Panel button navigates back to the Control Panel, also known as “Home”



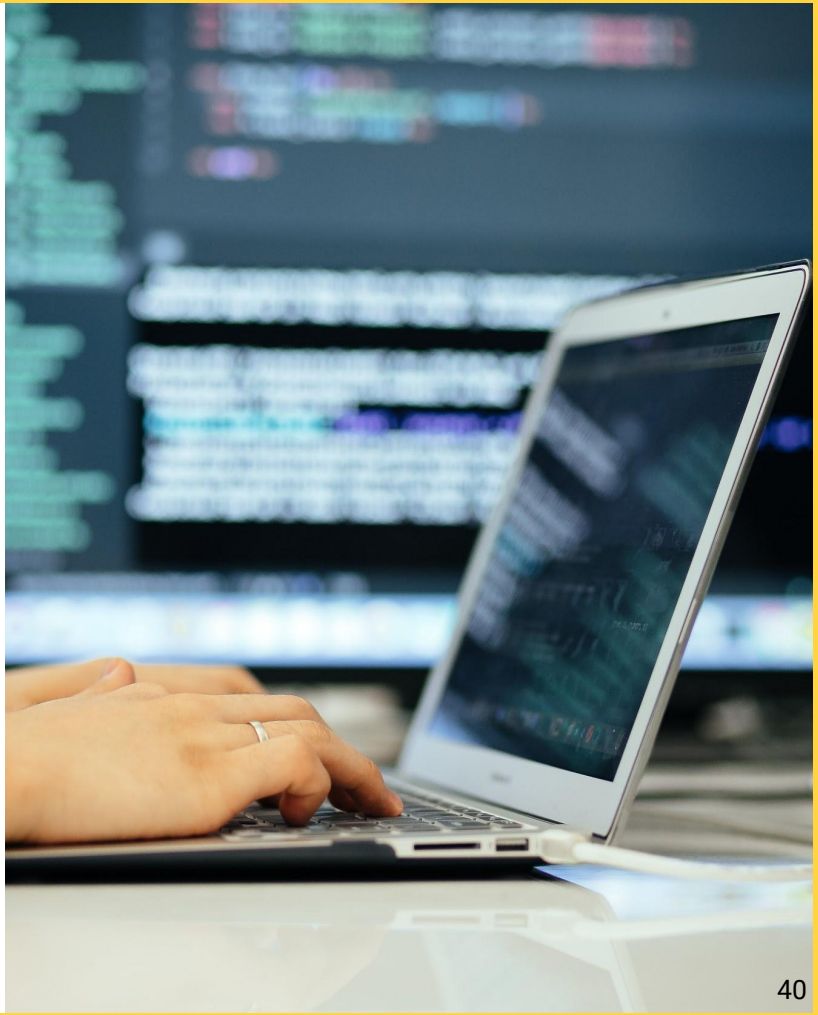
IllumiDesk through Assignment links



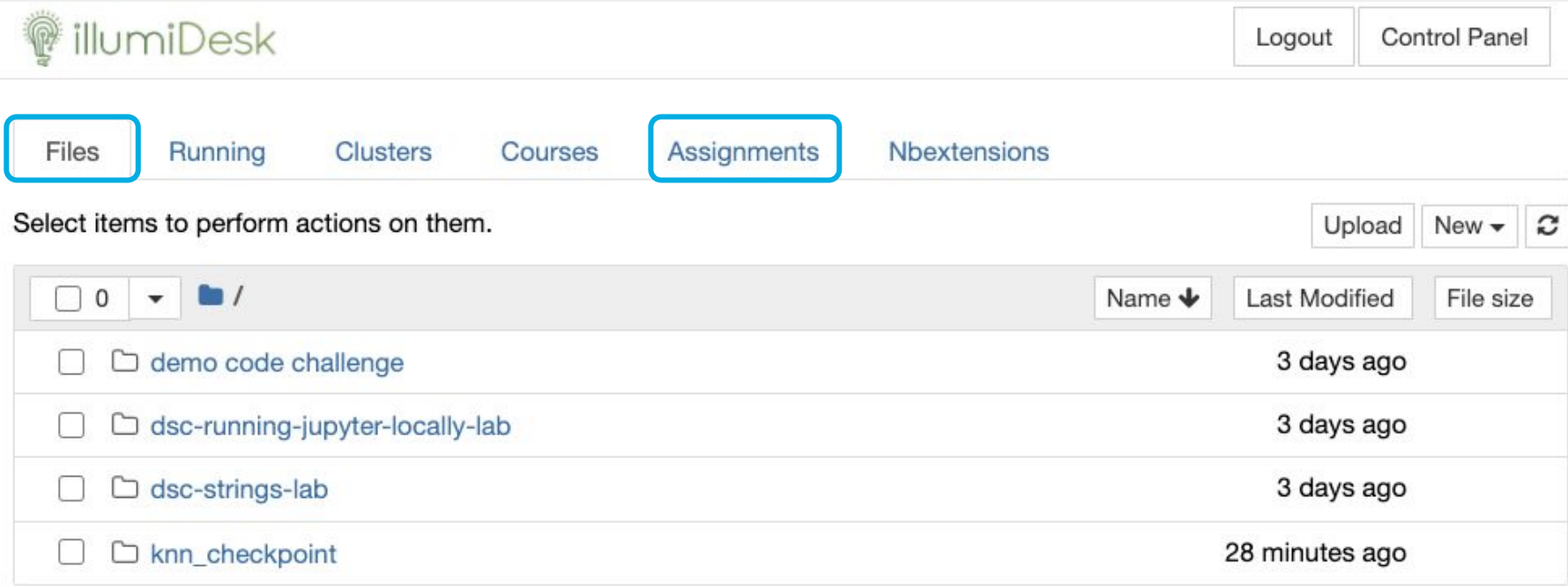


IllumiDesk Structure

Every Jupyter Notebook opened through an assignment link lives in your Files tab, and will save your work.



Only two tabs matter for now



Wait - should we do labs in IllumiDesk, or clone them to a local environment?

Advantages of GitHub

Employers look for comfort using git
A “green” robust github commit history
Content accessible after the program
It is what you will be using in the real world
Built for collaboration

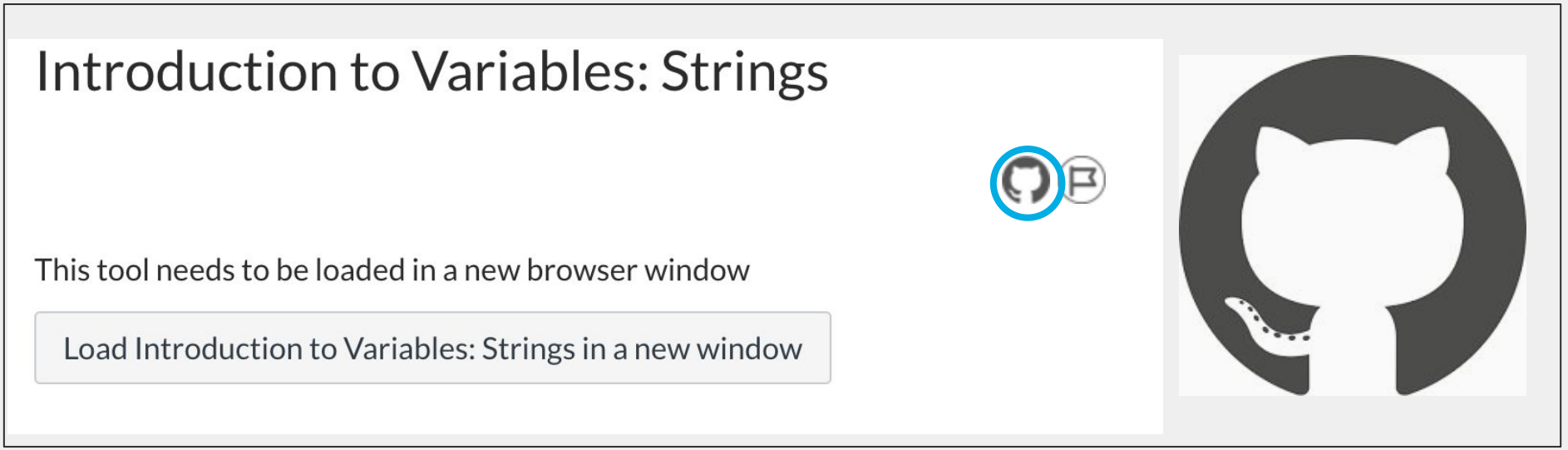
Advantages of IllumiDesk

Ease of use
No environment issues
Fully integrated into Canvas

You will use both

GitHub	IllumiDesk
Projects	Labs & Code Lessons

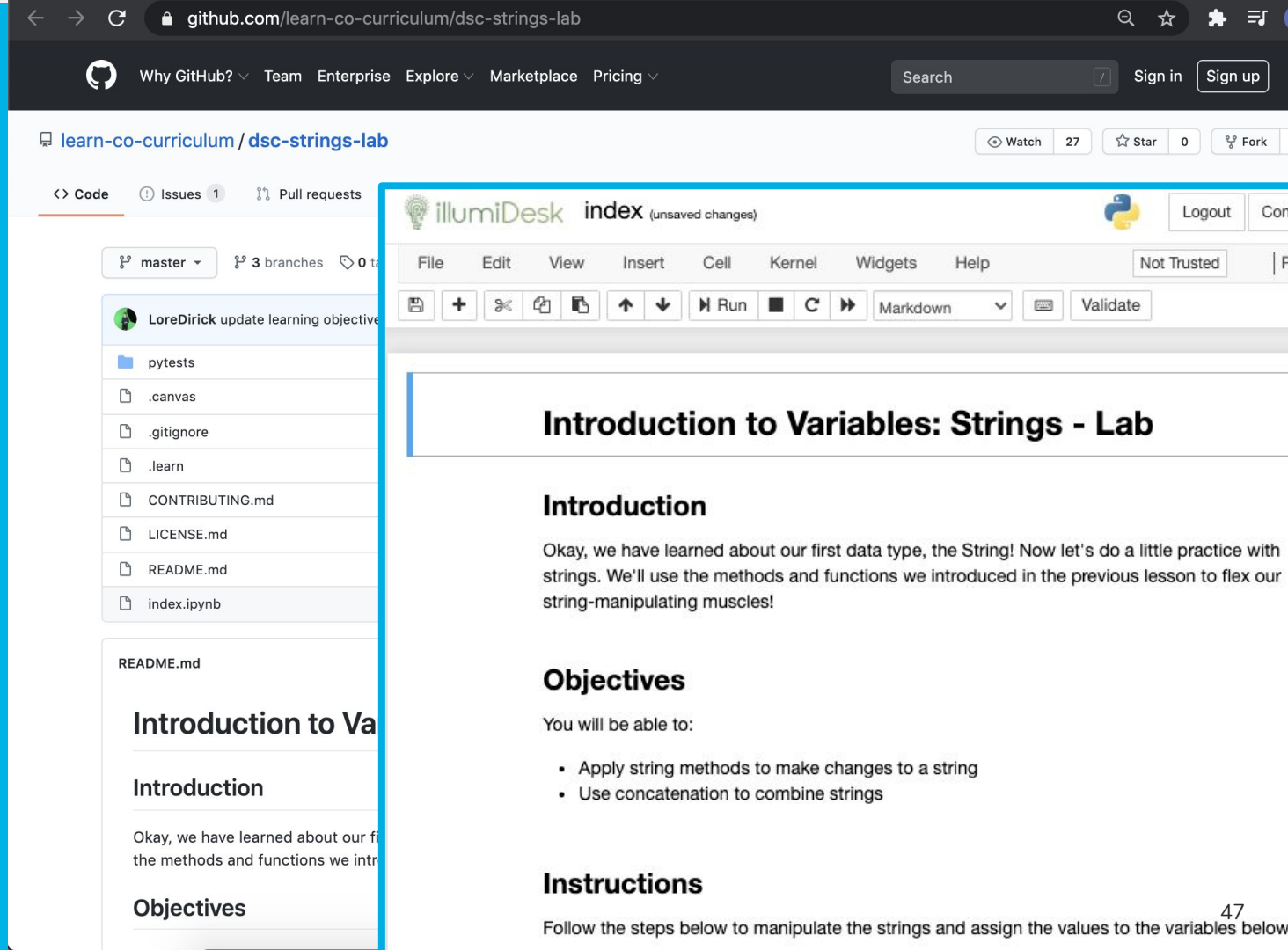
Every lesson with code is stored on GitHub



The GitHub logo in Canvas will navigate to the lesson's GitHub repository

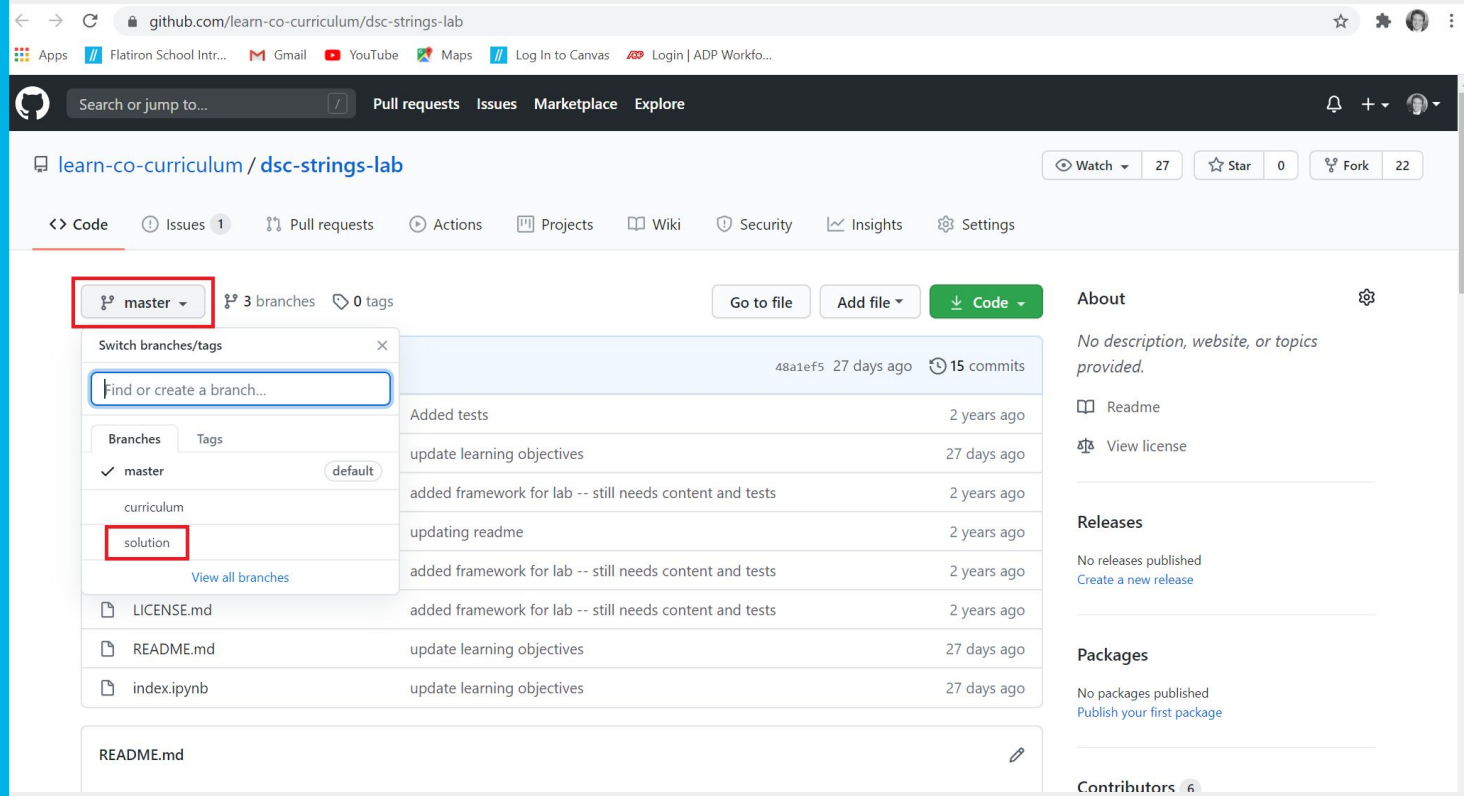
You fork and clone the source material from GitHub to your own machine.

(instructions to come)



Lab solutions are on the “solution” branch of each repository.

(we will teach you what that means soon)



Putting it All Together

Go to the “Topic 1: Getting Started with Data Science” Module in the Phase 1 course on Canvas and work through the “Setting up a Professional Data Science Environment” lessons that are appropriate for your OS!