Introduction to Python Programming

An APS workshop for REU students
June 8, 2022

hosted by

Victoria "Cat" Catlett

(they/them)
Software Engineer I
Green Bank Observatory

I am a Software Engineer at the Green Bank Observatory

Today, we will go over the basics of Python

Outline:

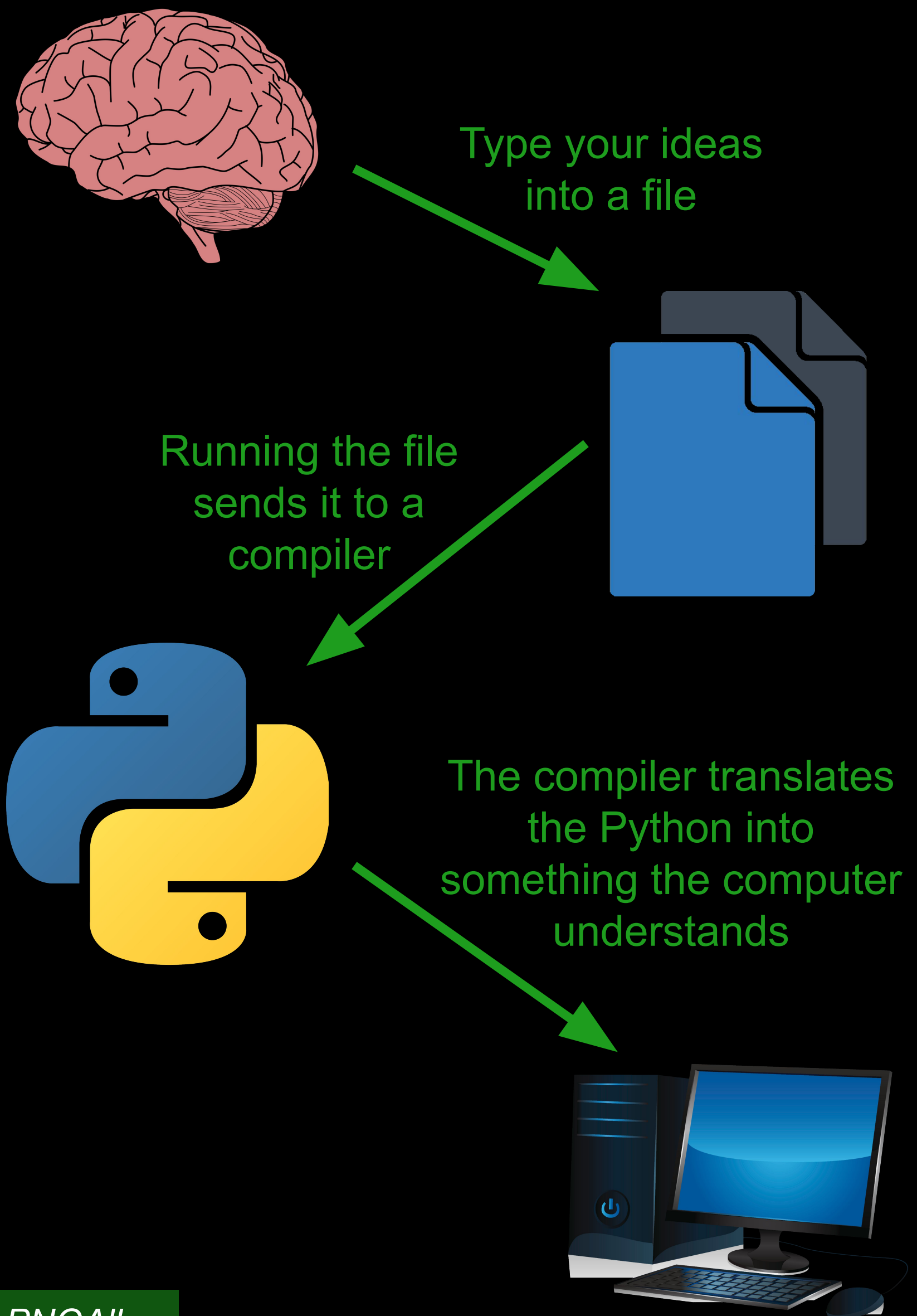What is Python?

Getting Started

Learning by Examples:
1. Syntax
2. Data Structures
3. Functions
4. Conditionals
5. Libraries

Moving Forward

What is Python?

Python is a high-level programming language

Type your ideas into a file

Running the file sends it to a compiler

The compiler translates the Python into something the computer understands

Python files can be .py or .ipynb

```python
48    import pandas as pd
49
50    # Go back 2 folders, into the "files" folder, and identify Excel sheet
51    path2data = '../../files/sample_data.xlsx'
52
53    # Use pandas to load the Excel data
54    data = pd.read_excel(path2data)
55
56    # Make the Excel column called "x" be the x-values for our plot
57    x = data['x']        # or x = data.x
58
59    # Make the Excel column called "sin_x" be the y-values for our plot
60    y = data['sin_x']  # or y = data.sin_x
61
62    # Plot the data
63    plt.figure()
64    plt.plot(x,y)
```

.py

Here are two things you should notice:

- We have to pay attention to **indentation** now
- If your `condition` is always `True`, the loop will run forever!

The cell below will show you how to construct a while loop, and the example will add consecutive numbers until the sum is greater than 10.

```python
[ ]  current_number = 0 # Start with 0
     current_sum = 0 # Start with a sum of 0

     while current_sum < 10:
         print('%i + %i = %i'%(current_sum, current_number, current_sum + current_number))

         current_sum = current_sum + current_number # Add the number to the sum
         current_number = current_number + 1 # Increase the number by 1

     # This next line won't execute until the loop is done
     print("The sum is %i, and the last number is %i"%(current_sum, current_number))
```

It's possible to make a loop that runs forever, which will eventually crash your computer. That's not ideal. To make sure your while loop stops even if you made a mistake, add another stopping condition based on the number of iterations. Consider the example below. It would run forever without the extra stopping condition because `7 < 8` is always `True`.

```python
[ ]  iteration_max = 1000
     iteration_count = 0

     while 7 < 8 and iteration_count < iteration_max:
       iteration_count += 1
       if iteration_count == iteration_max:
         print('WARNING: Maximum number of iterations reached. Loop terminated.')
```
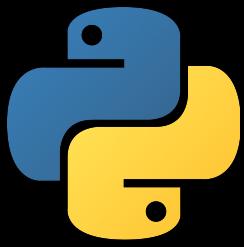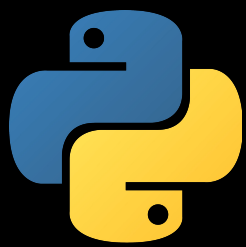
.ipynb
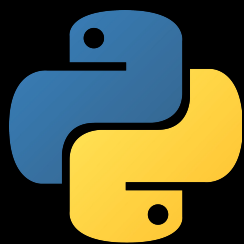
Python is named after…

David Clode

Python is named after…

David Clode

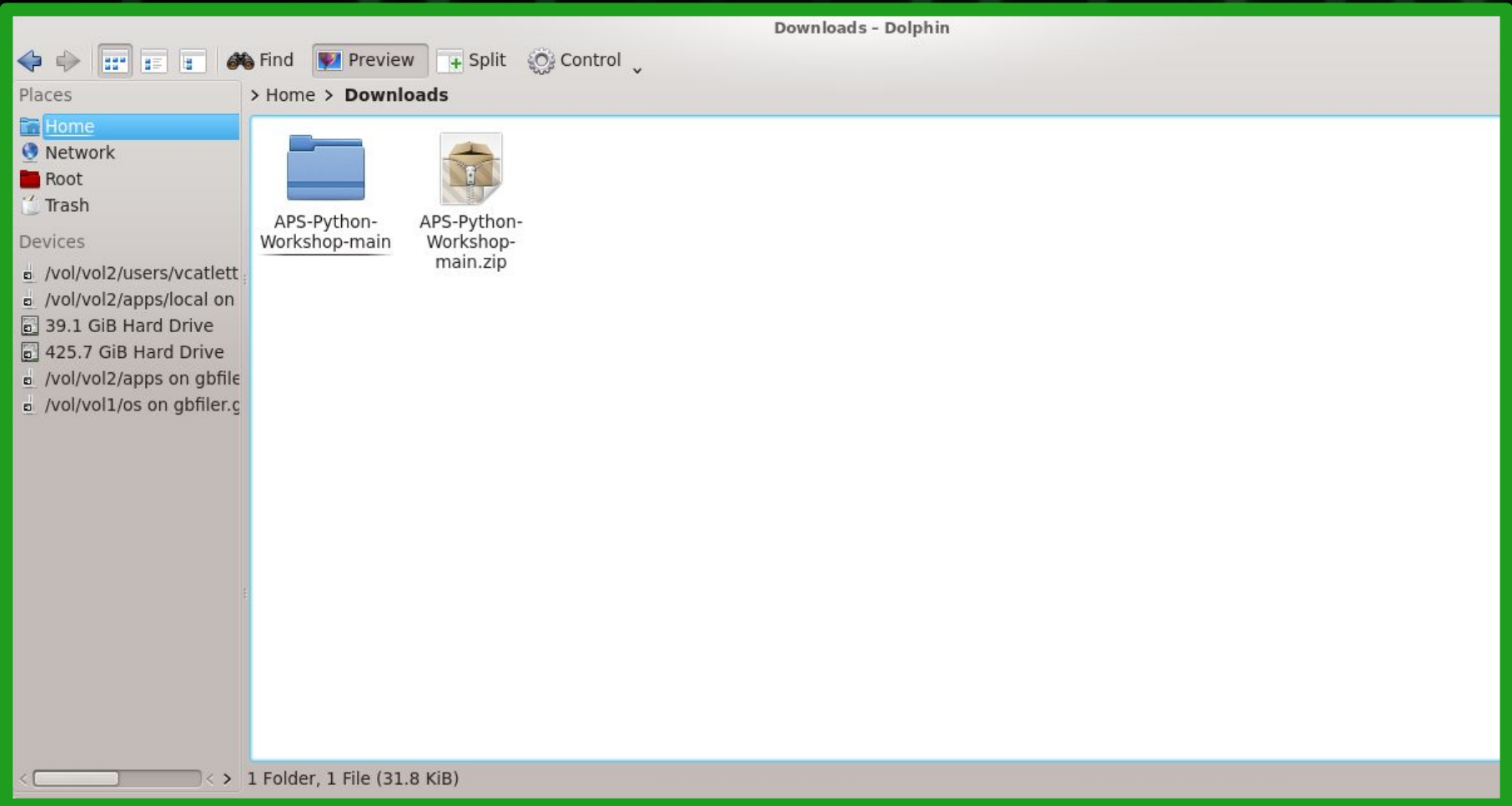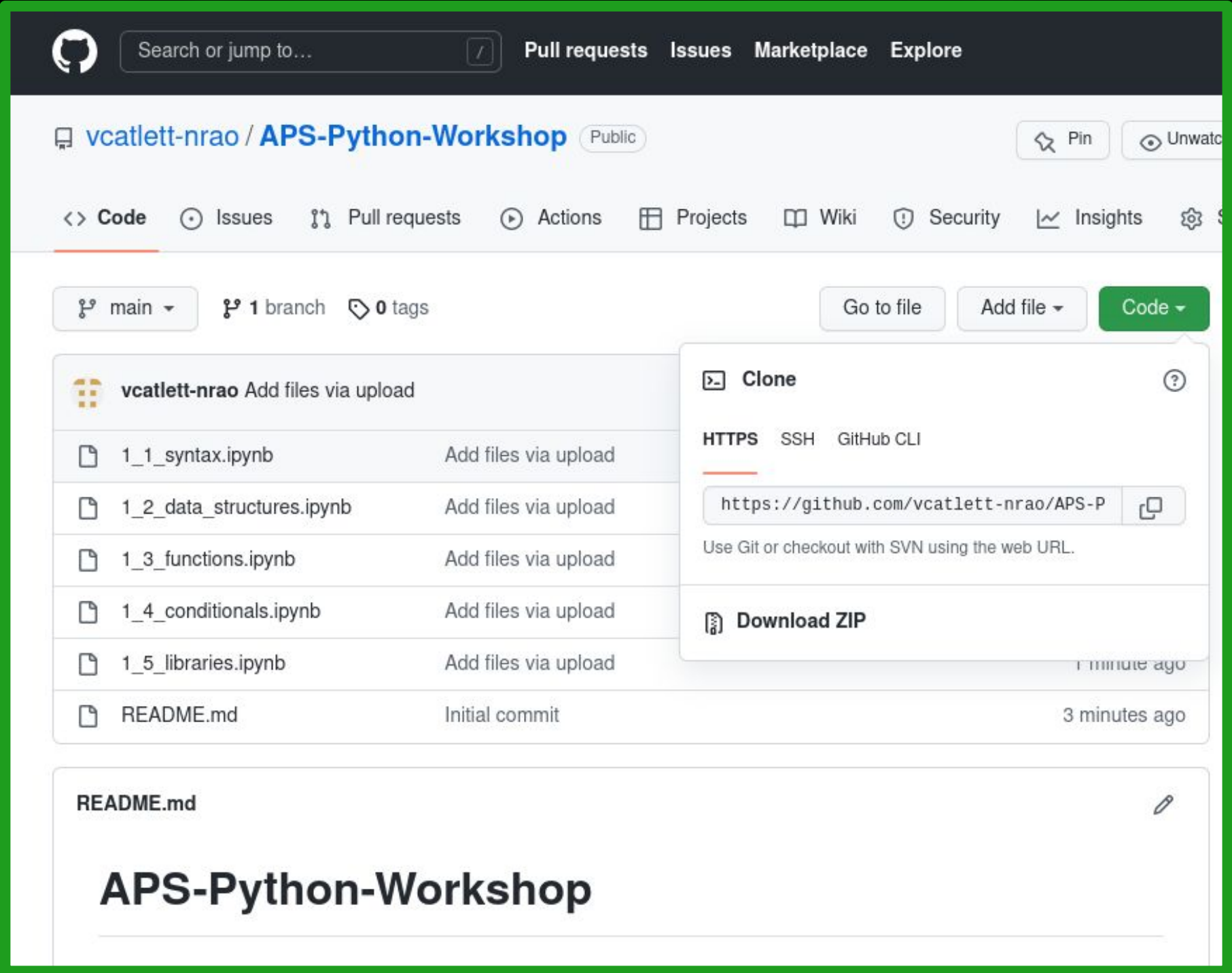**Python is named after...**

David Clode

Netflix

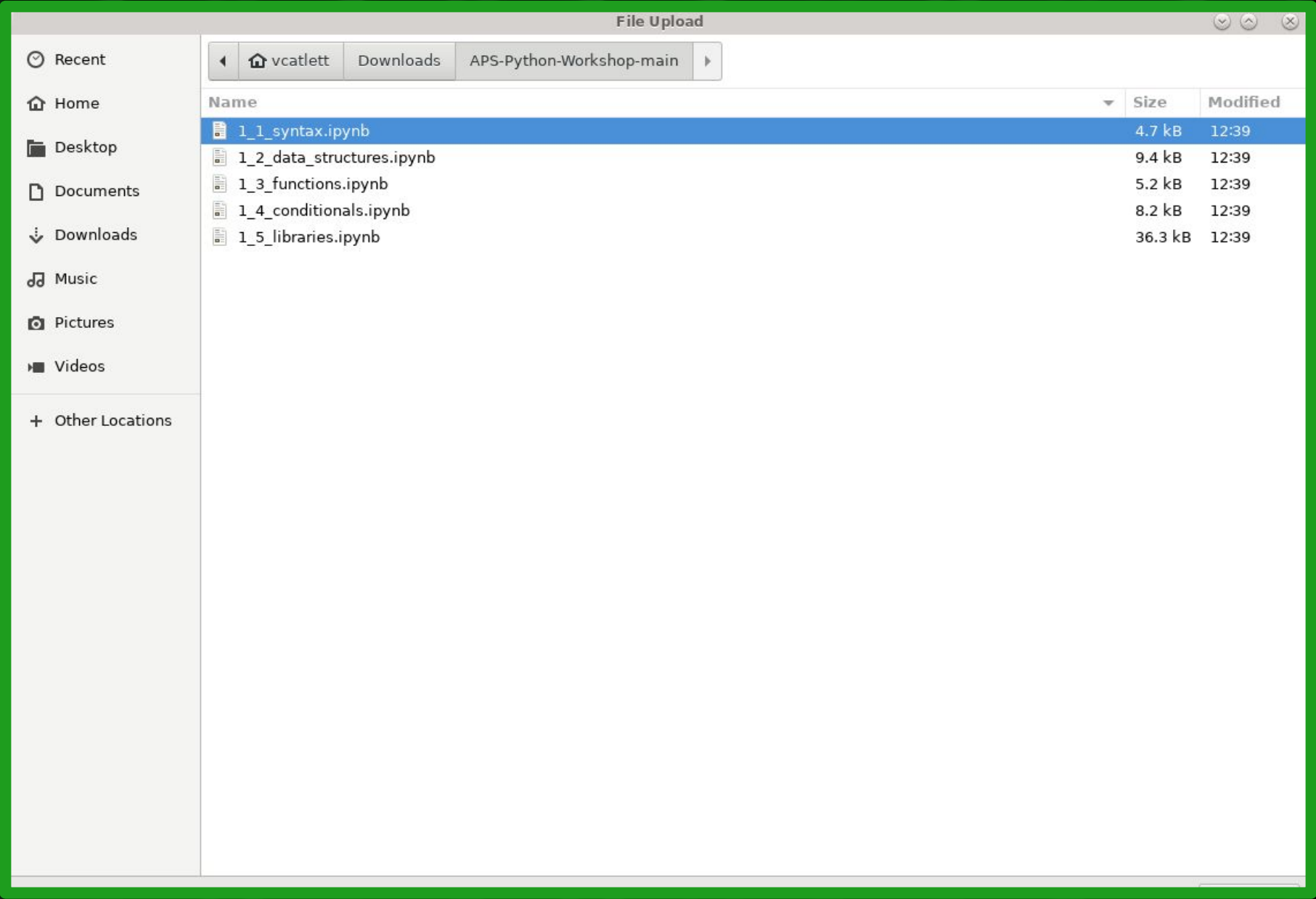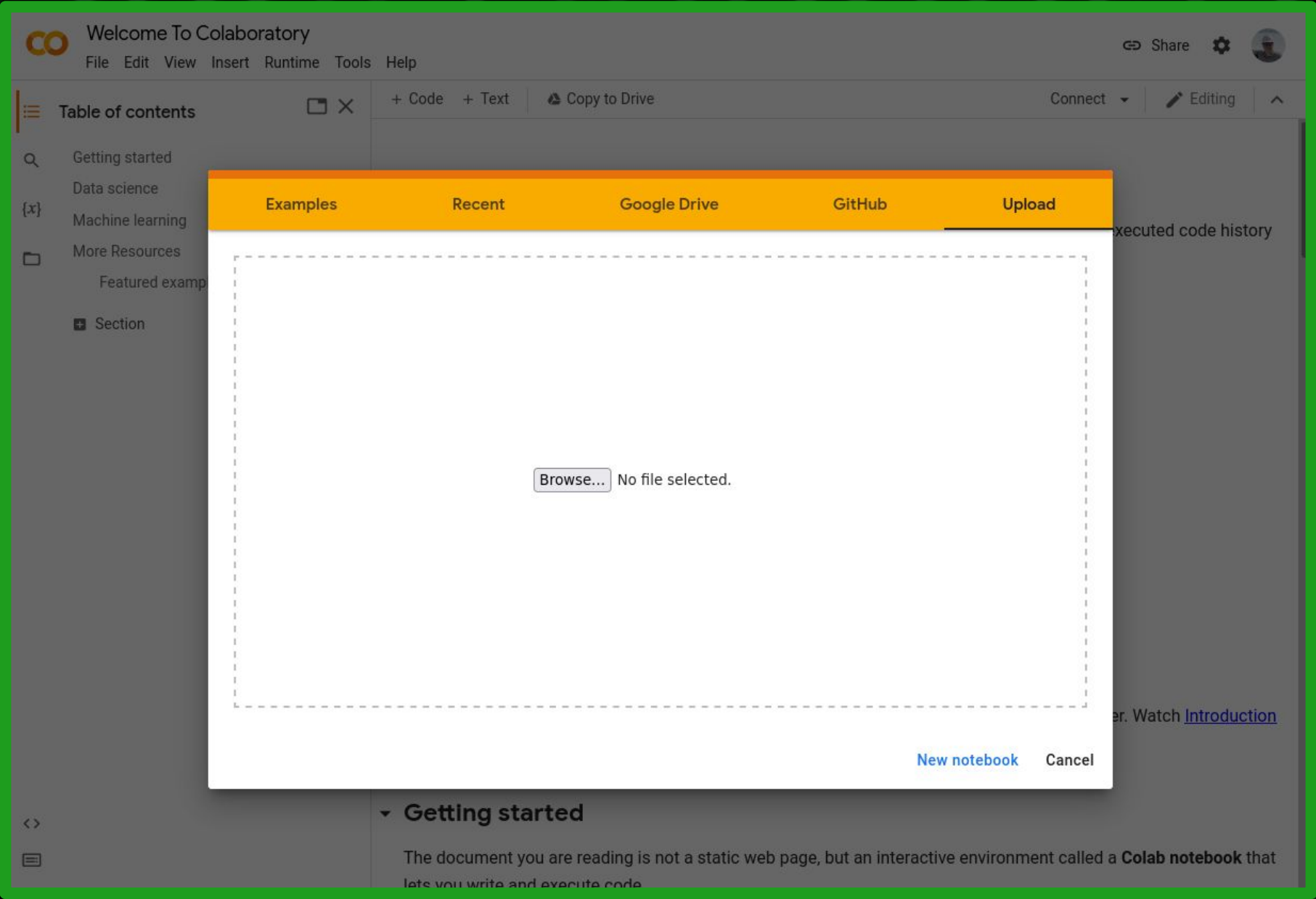# Getting Started

Download the session materials from my GitHub and unzip them

Upload the .ipynb files
into Google Colab

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

🔗 Share ⚙ 👤

:≡ Table of contents ☐ ✕

+ Code   + Text   ⬆ Copy to Drive

Connect ▾   ✏ Editing ⌃

🔍 Getting started

Data science

{x} Machine learning

📁 More Resources

Featured examp

⊞ Section

Examples	Recent	Google Drive	GitHub	Upload

Browse... No file selected.

New notebook   Cancel

⟨⟩

☰

▾ **Getting started**

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that

---

File Upload   ⌄ ⌃ ✕

🕐 Recent   ◀ 🏠 vcatlett | Downloads | APS-Python-Workshop-main ▶

🏠 Home   **Name**   ▾   Size   Modified

📁 Desktop   📄 1_1_syntax.ipynb   4.7 kB   12:39
                📄 1_2_data_structures.ipynb   9.4 kB   12:39
📄 Documents   📄 1_3_functions.ipynb   5.2 kB   12:39
⬇ Downloads   📄 1_4_conditionals.ipynb   8.2 kB   12:39
                📄 1_5_libraries.ipynb   36.3 kB   12:39
🎵 Music

📷 Pictures

▶ Videos
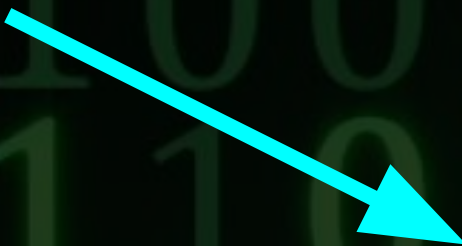
+ Other Locations

Syntax

1_1_syntax.ipynb

Data Structures

1_2_data_structures.ipynb

Functions

1_3_functions.ipynb

A code function is like a math one: it takes inputs and gives outputs

$$f(x) = |x|$$

y = abs(x)

$$f(x) = x^2$$
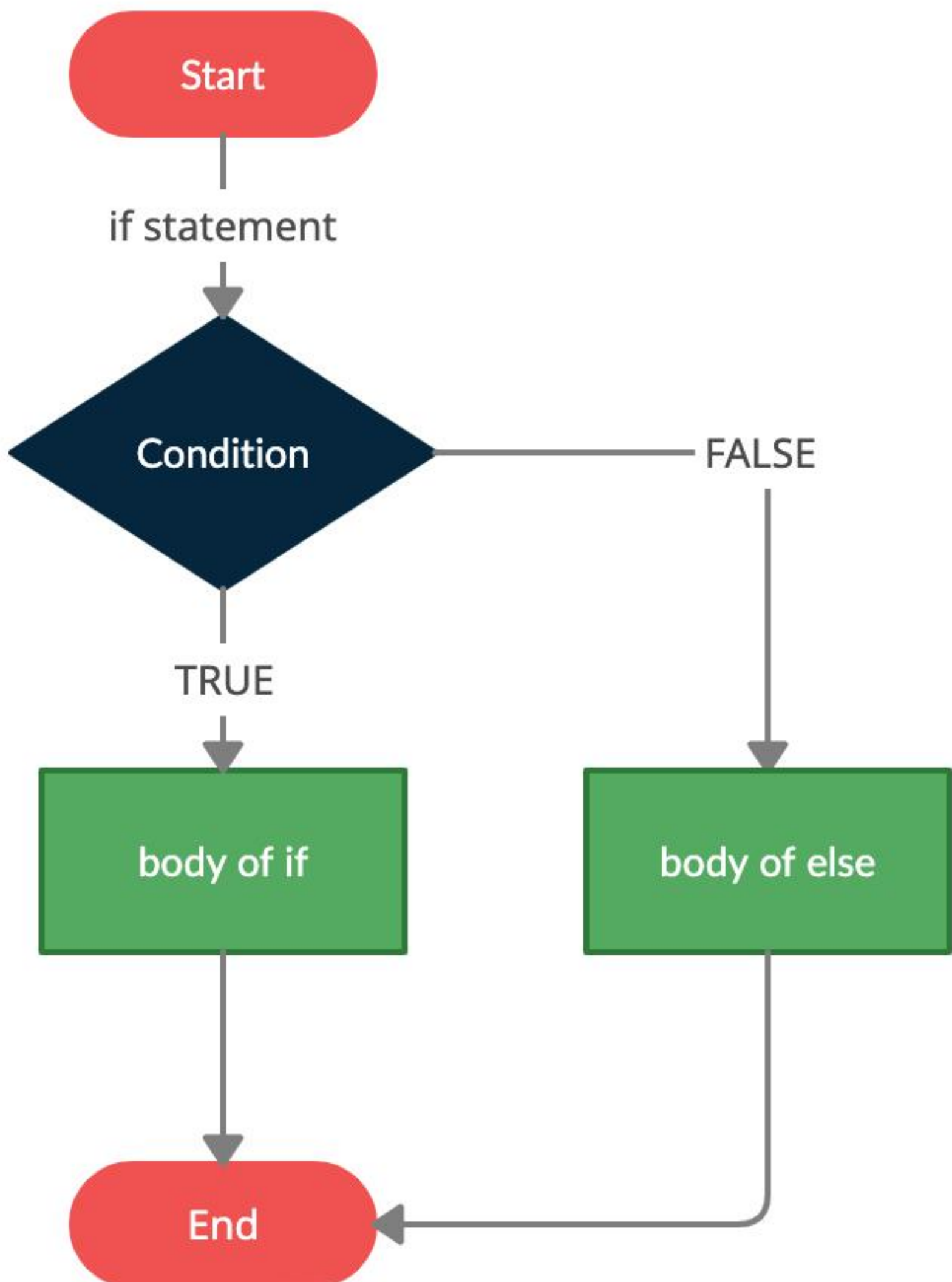
def f(x):
y = x**2
return y

They can be
built-in, **user-defined**, or imported

Conditionals

1_4_conditionals.ipynb

A **conditional** lets you evaluate blocks of code *only if* something is True


```
Start
  |
if statement
  |
  v
Condition --- FALSE ---
  |                    |
TRUE                   v
  |              body of else
  v                    |
body of if             |
  |                    |
  v                    |
End <------------------
```

toppr

Libraries

1_5_libraries.ipynb

Real Python

python™
Package
Index

It's all managed
by **pip**

Python Software Foundation

We will use
NumPy, Matplotlib, and SciPy

NumPy
Contains lots of math functions
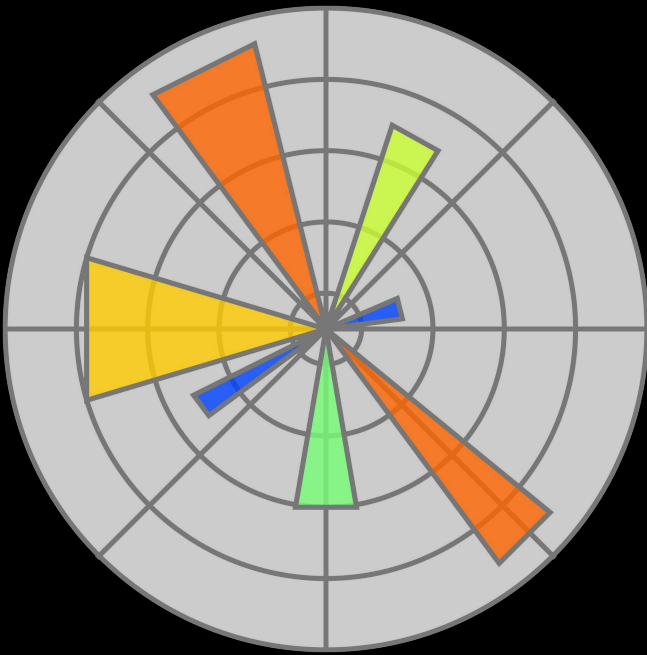
Matplotlib
Lets you make nice plots of data
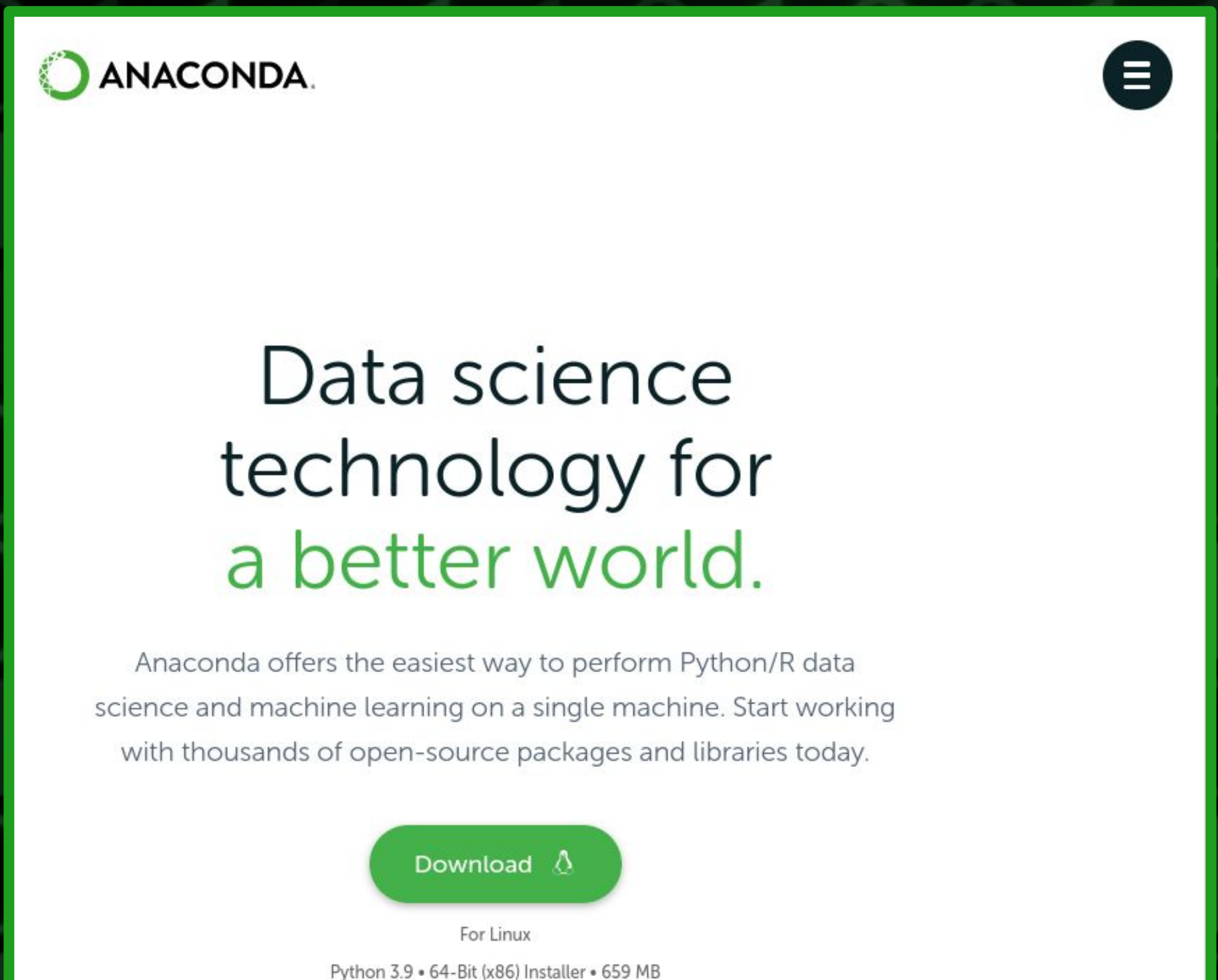
SciPy
Even more math functions than NumPy

Now What?

You can use online editors or download Python onto your computer