COMS BC1016

Introduction to Computational Thinking and Data Science

Lecture 2: Introduction to Python

Sept. 3, 2025

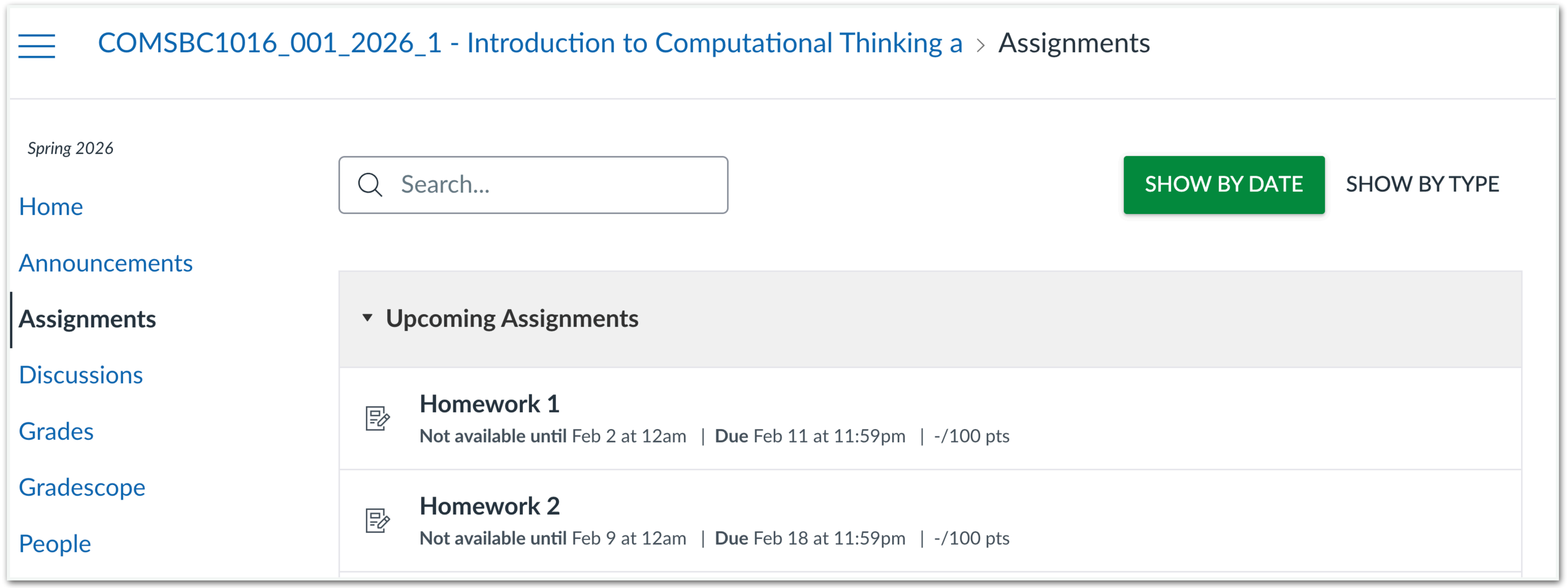
January 26, 2025

Office Hours

- Office hours begin next week (first week of February)
- **Monday**:
  - Elena Lukac, 1:30-3pm in Milstein 503
  - Eysa Lee, 3-5pm in Milstein 512
- **Tuesday**: Nami Jain, 4-5:30pm in Milstein 503
- **Wednesday**: Madeline Gutierrez, 5:30-7pm in Milsten 503
- **Thursday**: Sathya Raman, 4-5:30pm in Milstein 503

Assignments

- Labs and HWs located on Courseworks
    - Download the files from the assignment page
    - HWs posted to 1016, Labs to 1017
- Submit HWs as .ipynb
- Submit Labs as PDFs



COMSBC1016_001_2026_1 - Introduction to Computational Thinking a › Assignments

Spring 2026

Home
Announcements
Assignments
Discussions
Grades
Gradescope
People

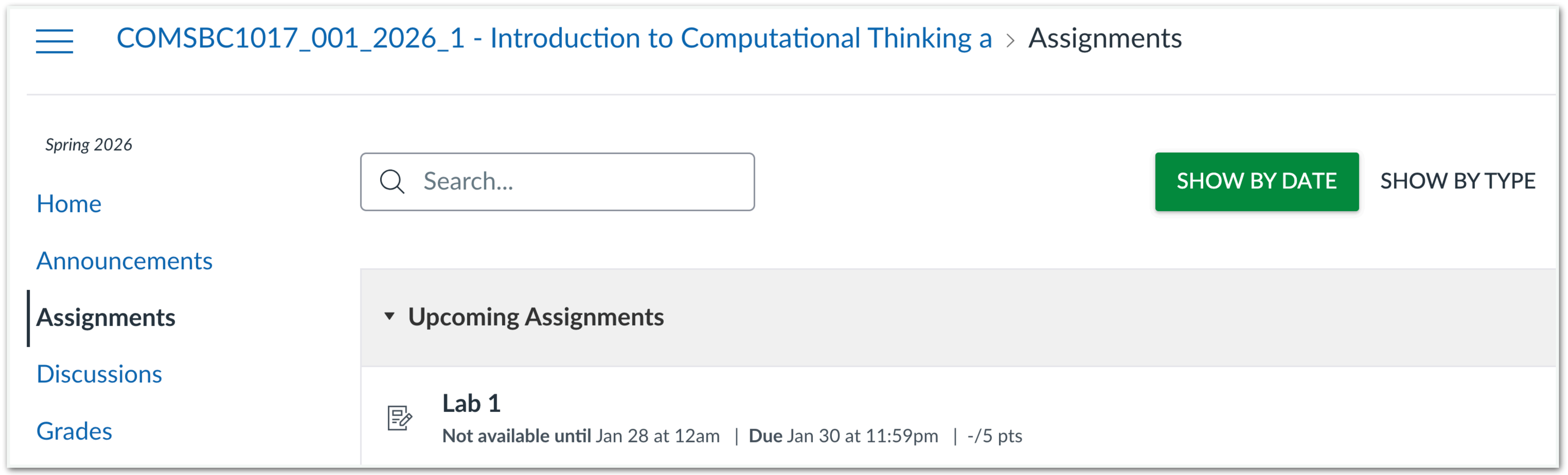
Search...

SHOW BY DATE SHOW BY TYPE

▾ Upcoming Assignments

Homework 1
Not available until Feb 2 at 12am | **Due** Feb 11 at 11:59pm | -/100 pts

Homework 2
Not available until Feb 9 at 12am | **Due** Feb 18 at 11:59pm | -/100 pts



COMSBC1017_001_2026_1 - Introduction to Computational Thinking a › Assignments

Spring 2026

Home
Announcements
Assignments
Discussions
Grades

Search...

SHOW BY DATE SHOW BY TYPE

▾ Upcoming Assignments

Lab 1
Not available until Jan 28 at 12am | **Due** Jan 30 at 11:59pm | -/5 pts

More on Homeworks

- Homeworks will be released on Mondays and due the following Wednesday
  - HW 1 will be released next week Monday
- Autograder tests for common mistakes and type errors
  - We will be checking more than just what tests are in the autograder!
  - True/False, short answer, and multiply choice are all manually graded
- Homeworks are submitted on Gradescope via Courseworks

Lab Reminders

- **Reminder: You must be enrolled in a 1017 section!**
  - As of Sunday night, there are more people enrolled in 1016 than 1017
- Labs begin this week
- **Email your TA if you'll be late or missing!**
  - 50% of your lab grade is attendance!
  - One unexcused absence + lowest lab dropped

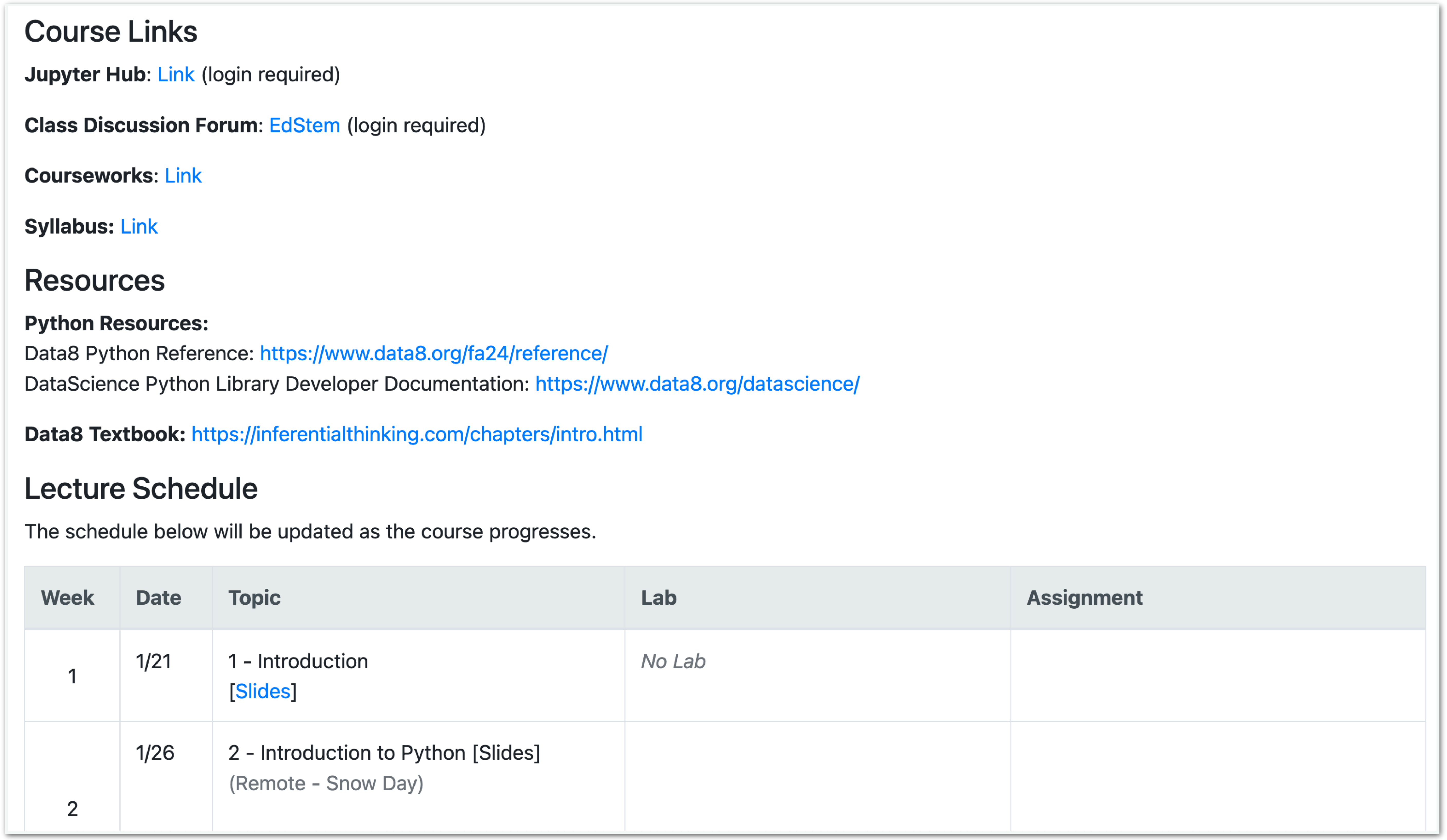
Reminder: Midterm Exam

- Paper exam happens during class **Wednesday, March 11, 2026**
  - This is the week before Spring Recess
- You will be allowed a note sheet to use as a reference during the exam
  - It will be submitted along with your exam
- If you need particular accommodations, please contact CARDS

Course Website

Slides, emails, and helpful links are on the course website:

https://www.eysalee.com/courses/s26/bc1016.html



Course Links

Jupyter Hub: Link (login required)

Class Discussion Forum: EdStem (login required)

Courseworks: Link

Syllabus: Link

Resources

Python Resources:
Data8 Python Reference: https://www.data8.org/fa24/reference/
DataScience Python Library Developer Documentation: https://www.data8.org/datascience/

Data8 Textbook: https://inferentialthinking.com/chapters/intro.html

Lecture Schedule

The schedule below will be updated as the course progresses.

Week	Date	Topic	Lab	Assignment
1	1/21	1 - Introduction [Slides]	*No Lab*	
2	1/26	2 - Introduction to Python [Slides] (Remote - Snow Day)		

Python

Python Intro

- Popular programming for software development

- Especially popular for data science

- Learning programing is about learning how to think computational & transfers to other languages

How to Approach Programming in BC1016

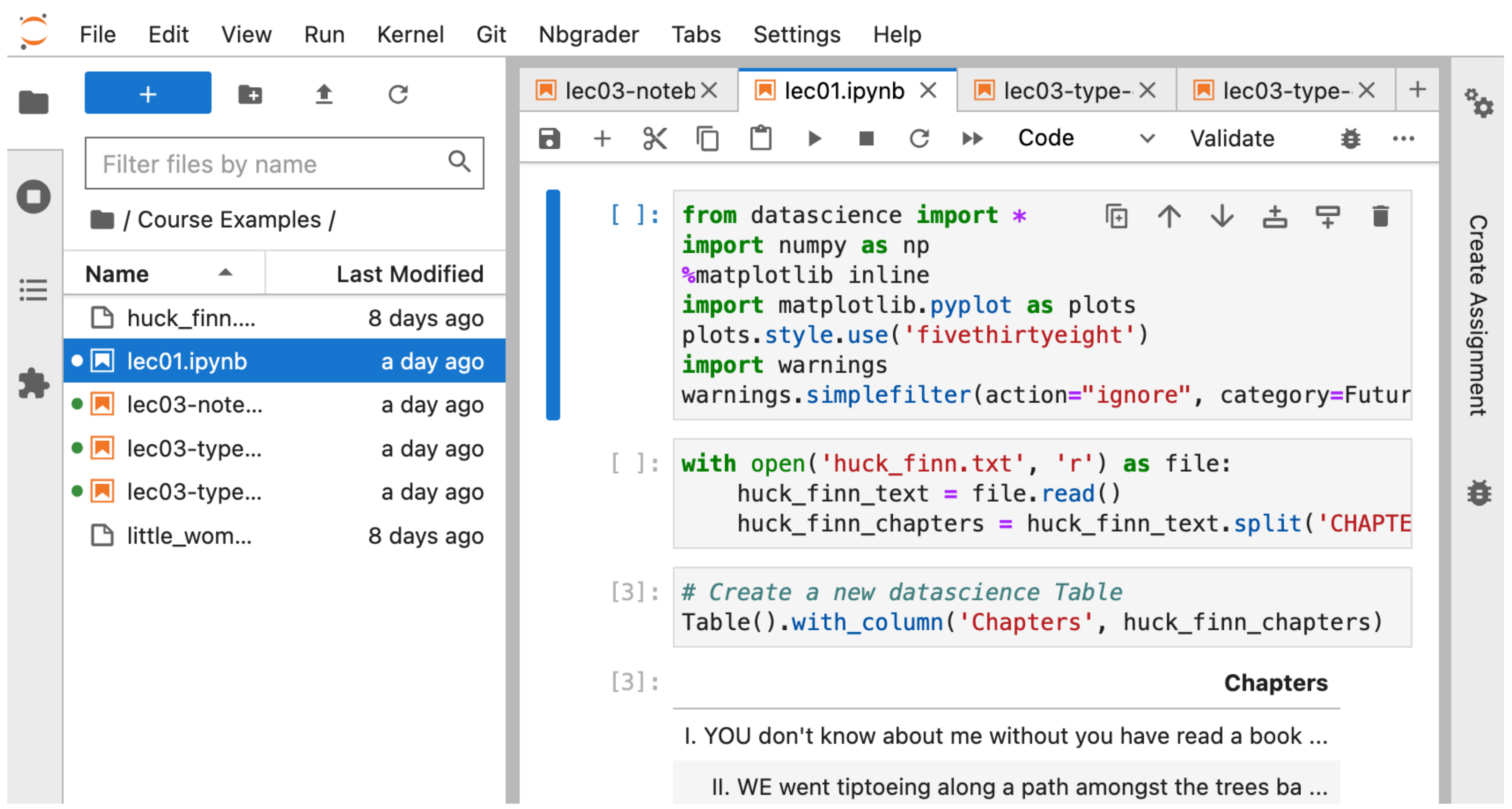
Programming is all about practice

- Class: Demos introduce terms & rules
- Labs & Homework: Try out programming yourself

Goal: writing your own code that can solve new problems!

Jupyter Notebooks

- Can be run locally or in cloud-based environments
    - For this class: JupyterHub Server
- Benefits of cloud-based
    - Access anywhere
    - Expandable compute resources



Name	Last Modified
huck_finn....	8 days ago
lec01.ipynb	a day ago
lec03-note...	a day ago
lec03-type...	a day ago
lec03-type...	a day ago
little_wom...	8 days ago

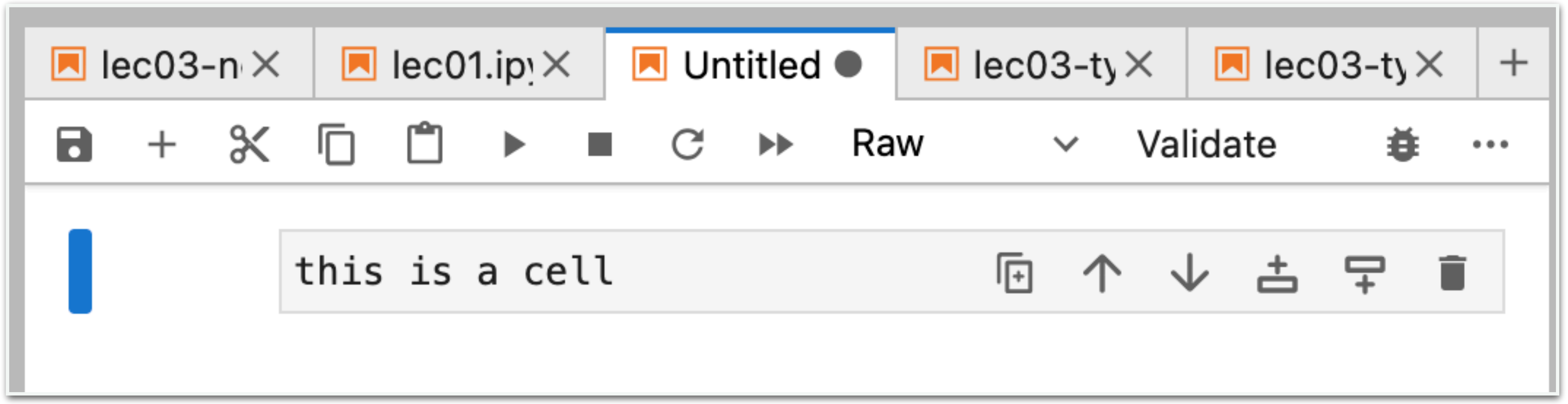
```
[ ]: from datascience import *
     import numpy as np
     %matplotlib inline
     import matplotlib.pyplot as plots
     plots.style.use('fivethirtyeight')
     import warnings
     warnings.simplefilter(action="ignore", category=Futur

[ ]: with open('huck_finn.txt', 'r') as file:
         huck_finn_text = file.read()
         huck_finn_chapters = huck_finn_text.split('CHAPTE

[3]: # Create a new datascience Table
     Table().with_column('Chapters', huck_finn_chapters)

[3]:                                          Chapters
     I. YOU don't know about me without you have read a book ...
          II. WE went tiptoeing along a path amongst the trees ba ...
```

Let's poke around in Jupyter…

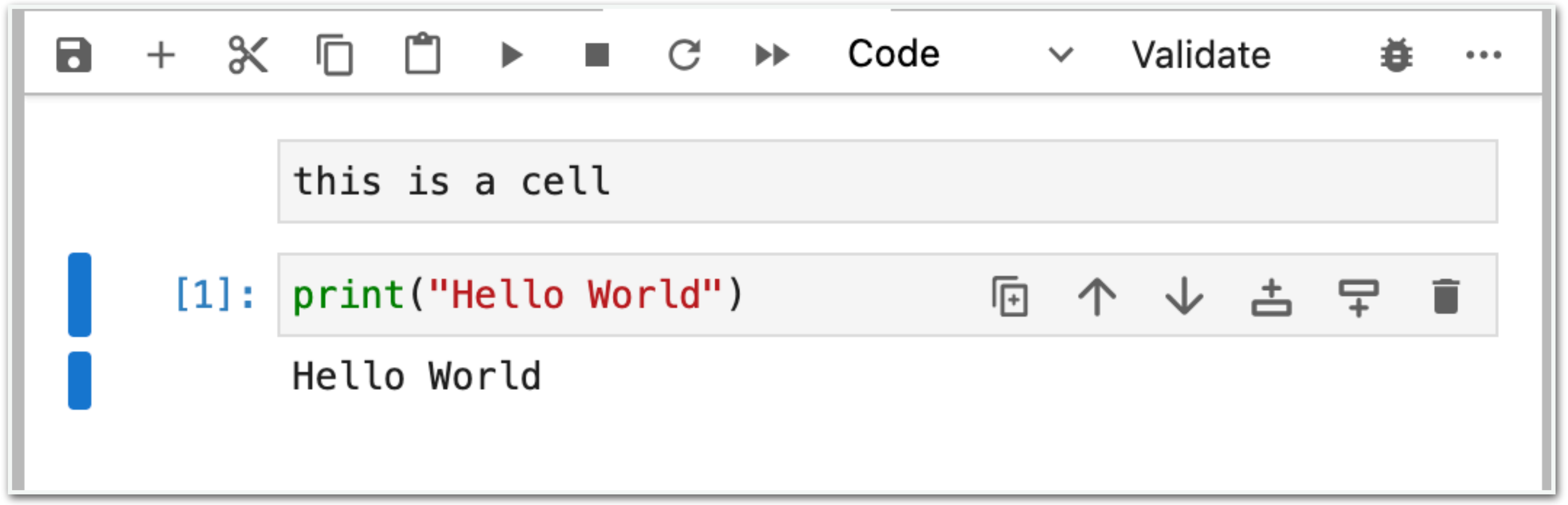
Jupyter Notebooks: Terms

- **Cells**: block or section for writing code / notes / etc

- **Kernel**: executes code in cells (when you run a “cell”)



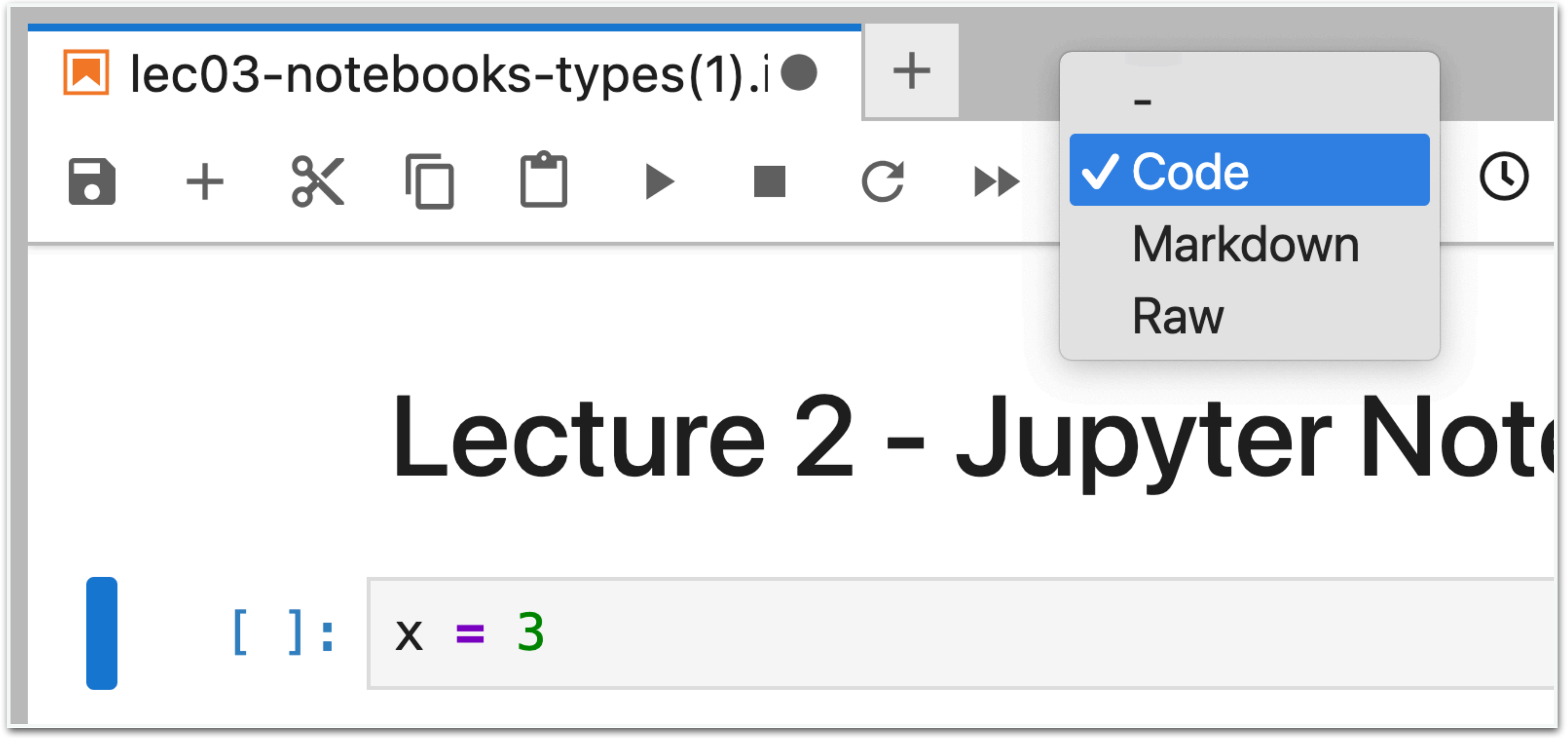
```
this is a cell
```



```
this is a cell

[1]: print("Hello World")

Hello World
```

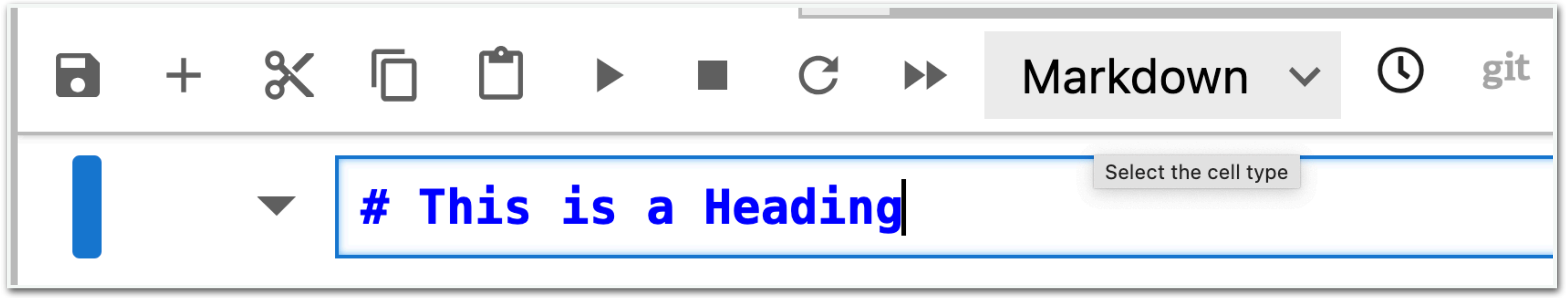
Jupyter Notebooks: Cell Types

- **Code**: For writing and running Python code
- **Markdown**: Markup language for formatting plaintext
  - You've likely come across it on the internet before (`#` for headings, `*text*` for italics, `**text**` for bold, …)
- **Raw**

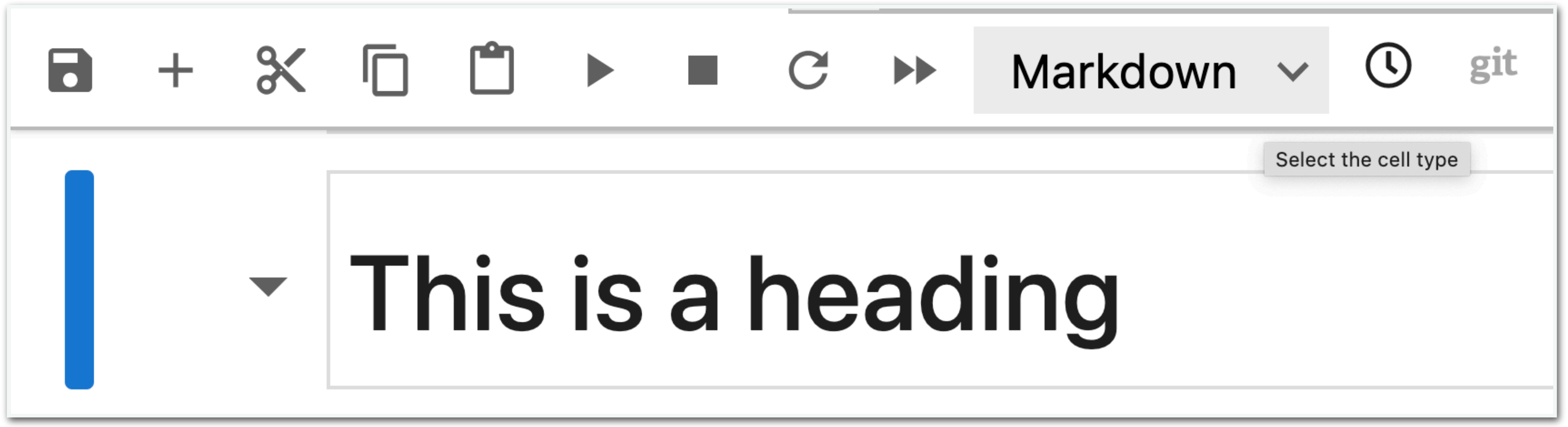


Lecture 2 - Jupyter Not

```
[ ]: x = 3
```



```
# This is a Heading
```



This is a heading

Data Types

Numbers

- **Integers**: Whole numbers
  - e.g., `3, -10, 25`
- **Floats**: Anything with decimals
  - e.g., `3.1, -10.2, 2.0`
- Basic calculations
  - e.g., `+, -, *, /`

Numbers

- **Integers**: Whole numbers
  - e.g., `3, -10, 25`
- **Floats**: Anything with decimals
  - e.g., `3.1, -10.2, 2.0`
- Basic calculations
  - e.g., `+, -, *, /`

Some questions:

1. What happens if we add different number types?
   a. Multiply or divide?
2. What if I wanted 2 raised to the power of 4?

Strings

- Text in python
- You can specify a string using either single quotes or double quotes
  - `“a”`
  - `‘This is a sentence’`
  - `“This is another sentence. Wow!”`

Some questions:

1. What happens if you start a string with a double quote and end it with a single quote?
2. What if I want to write the following sentence as a string using single quotes?
   a. `That’s weird`

Booleans

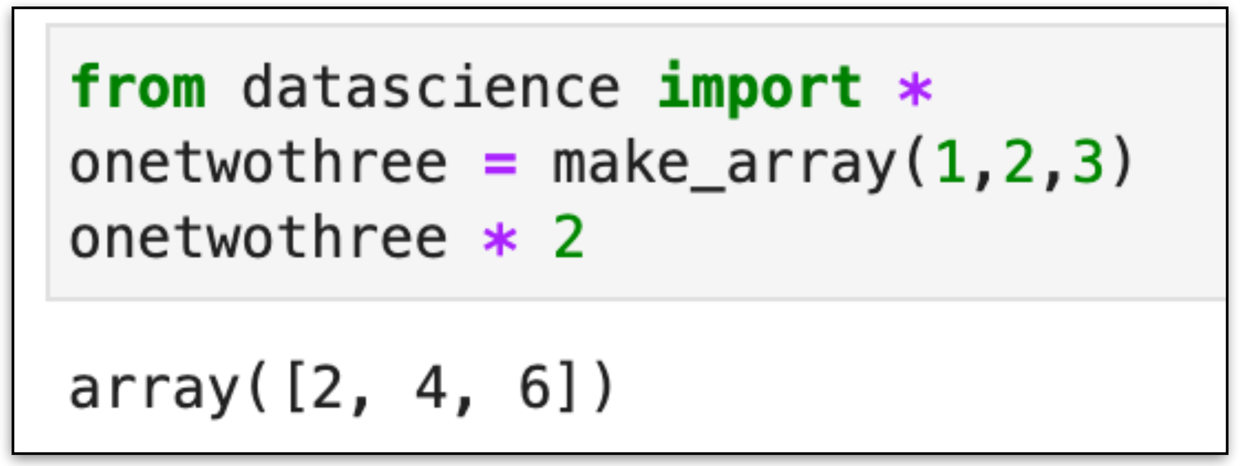
- True or False values
- Useful for conditional statements
  - Usually of the form “if some statement is true, then execute some code. Otherwise, do something else.”
- Use == to check equivalence
- We will talk more about Booleans in a few weeks

Lists

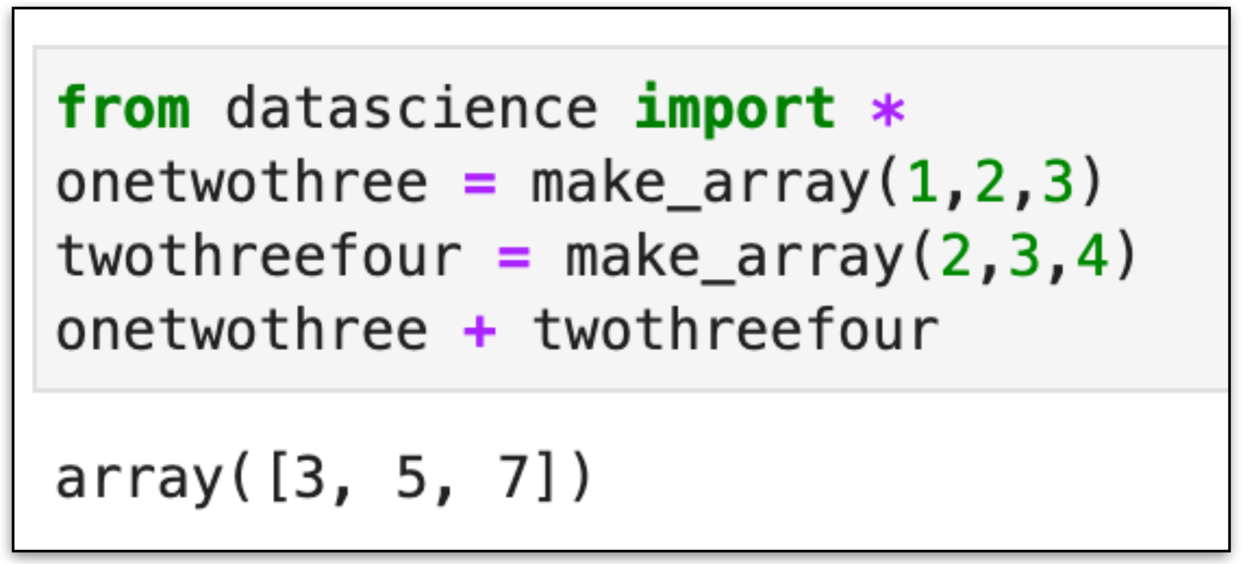
- Lists are sequences of values
- Start with brackets and each element is separated by commas
  - [2,4,6,8]
  - [‘apples’, ‘bananas’, ‘oranges’]
  - [‘apples’, ‘bananas’, ‘apples’, ‘apples’]
- Lists are zero-indexed
  - The first element is the 0th and the last is length-1

`datascience` **Arrays**

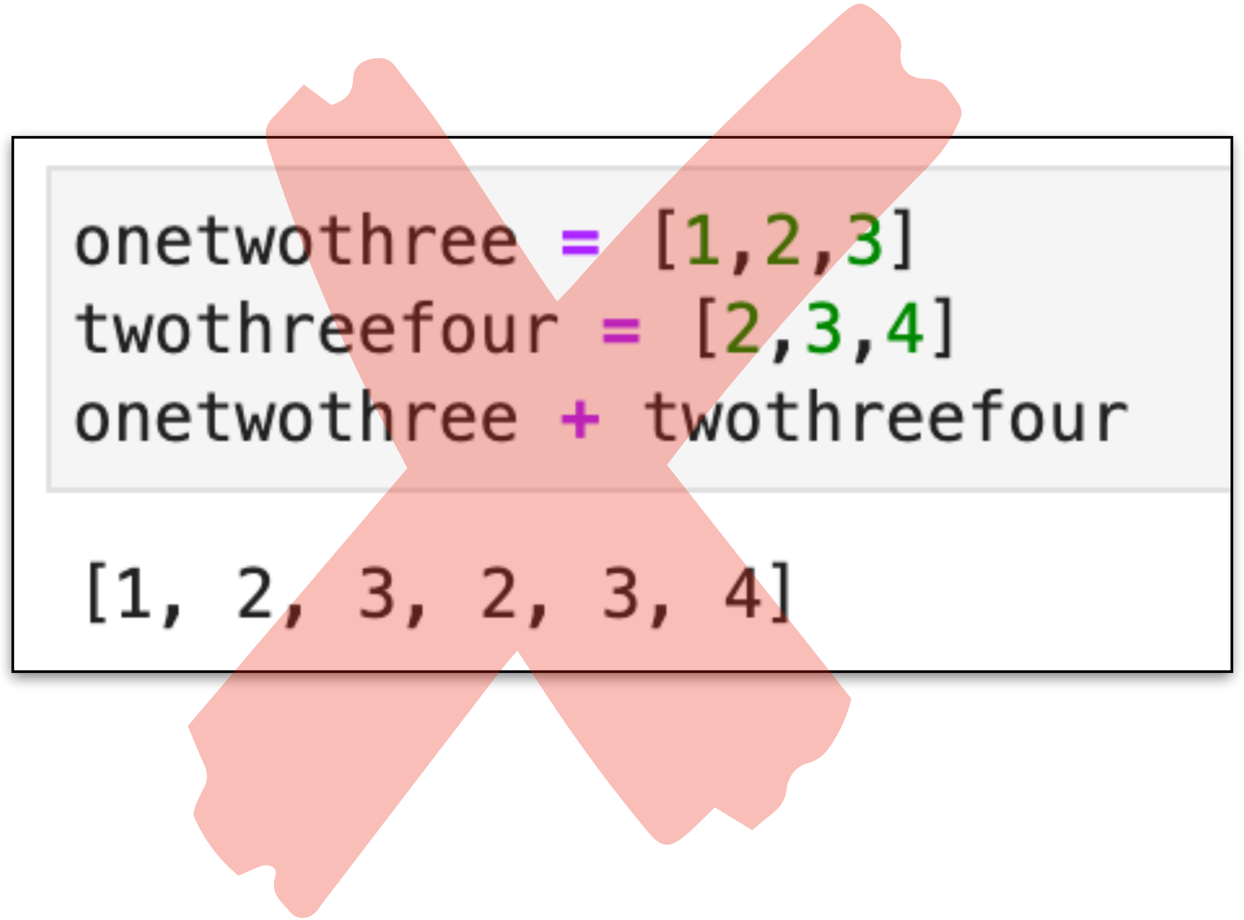
- In this class we'll mostly use NumPy arrays
    - Can make arrays using `datascience.make_array` or `numpy.array`
- Elements of an array should have the same type
    - e.g., `["Mystery", "Abby", "Jinu", "Baby", "Romance"]` or `[1,2,3,5]`
- First element is index 0 and the last is the length-1
- Can perform component-wise arithmetic
    - Note this only works for numpy arrays but not built-in Python lists!



```
from datascience import *
onetwothree = make_array(1,2,3)
onetwothree * 2

array([2, 4, 6])
```



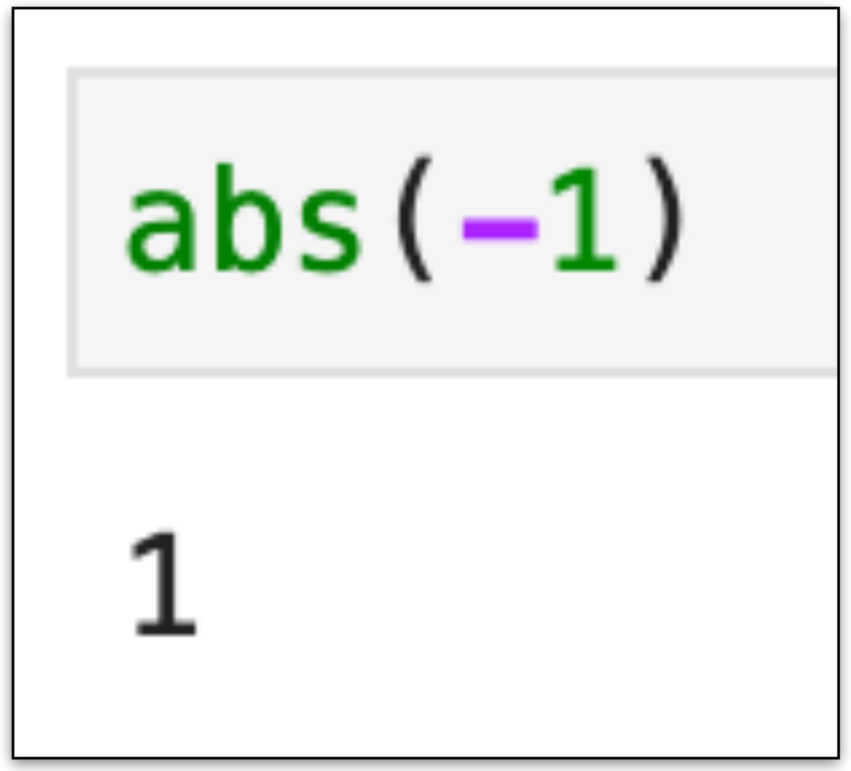
```
from datascience import *
onetwothree = make_array(1,2,3)
twothreefour = make_array(2,3,4)
onetwothree + twothreefour

array([3, 5, 7])
```



```
onetwothree = [1,2,3]
twothreefour = [2,3,4]
onetwothree + twothreefour

[1, 2, 3, 2, 3, 4]
```

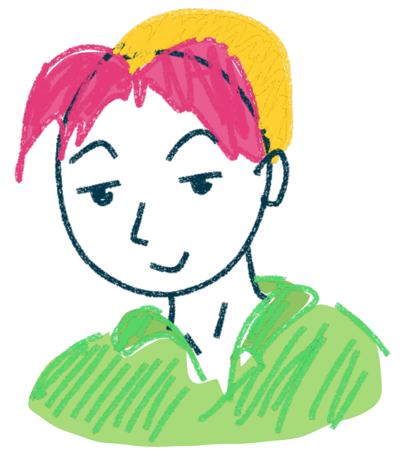
Functions

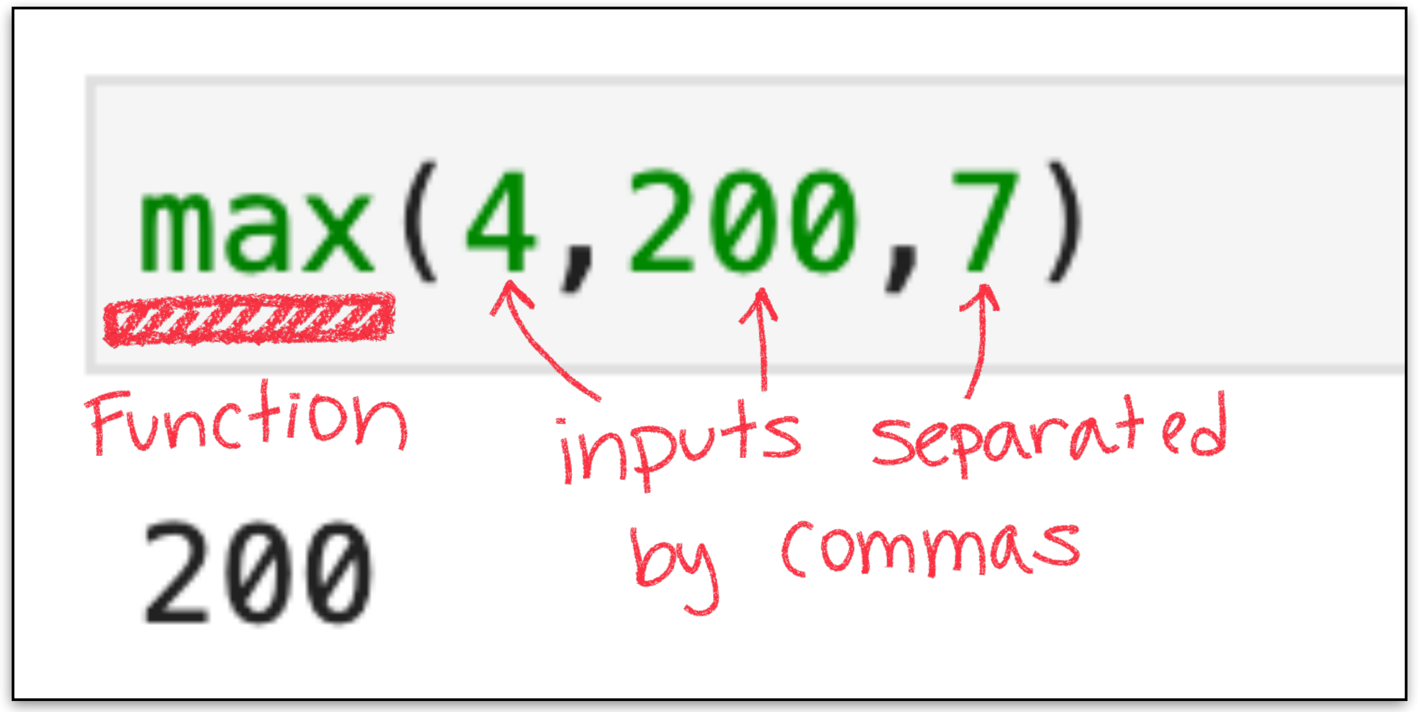
- Type of abstraction for pre-defined set of code or instructions
  - Commonly takes inputs, performs a computation, and produces output
  - Many useful things built in!
  - Some useful functions aren't built in, so we need to import them



```
abs(-1)
```

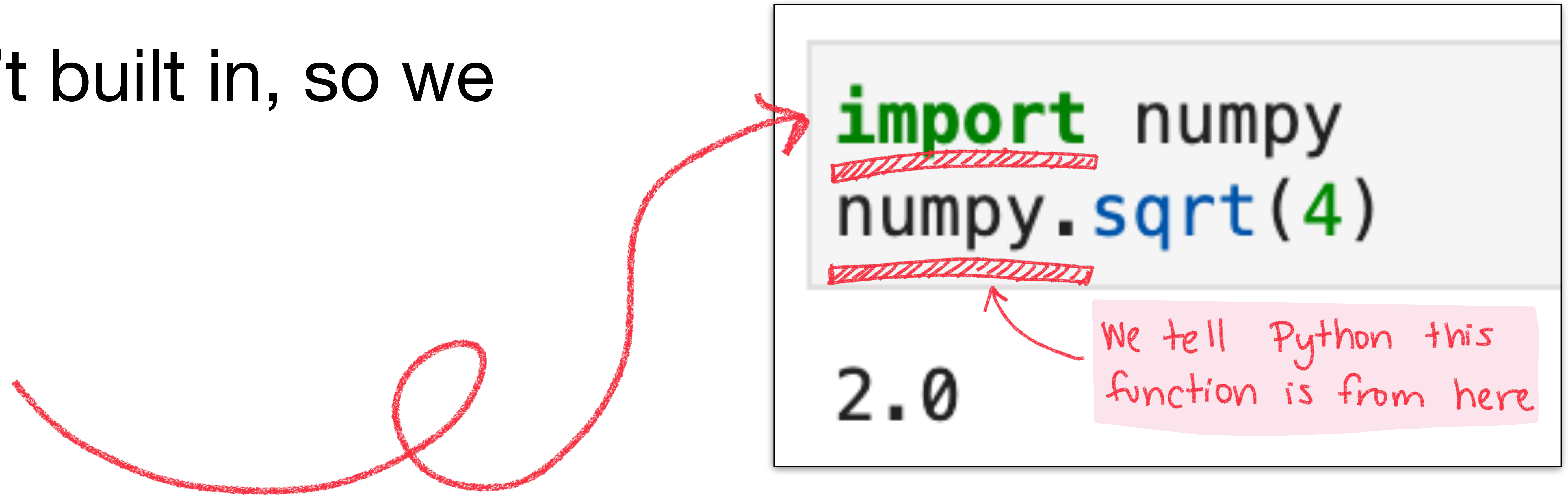
```
1
```





```
max(4,200,7)
```

```
200
```



```
import numpy
numpy.sqrt(4)
```

```
2.0
```

Built-in conversions

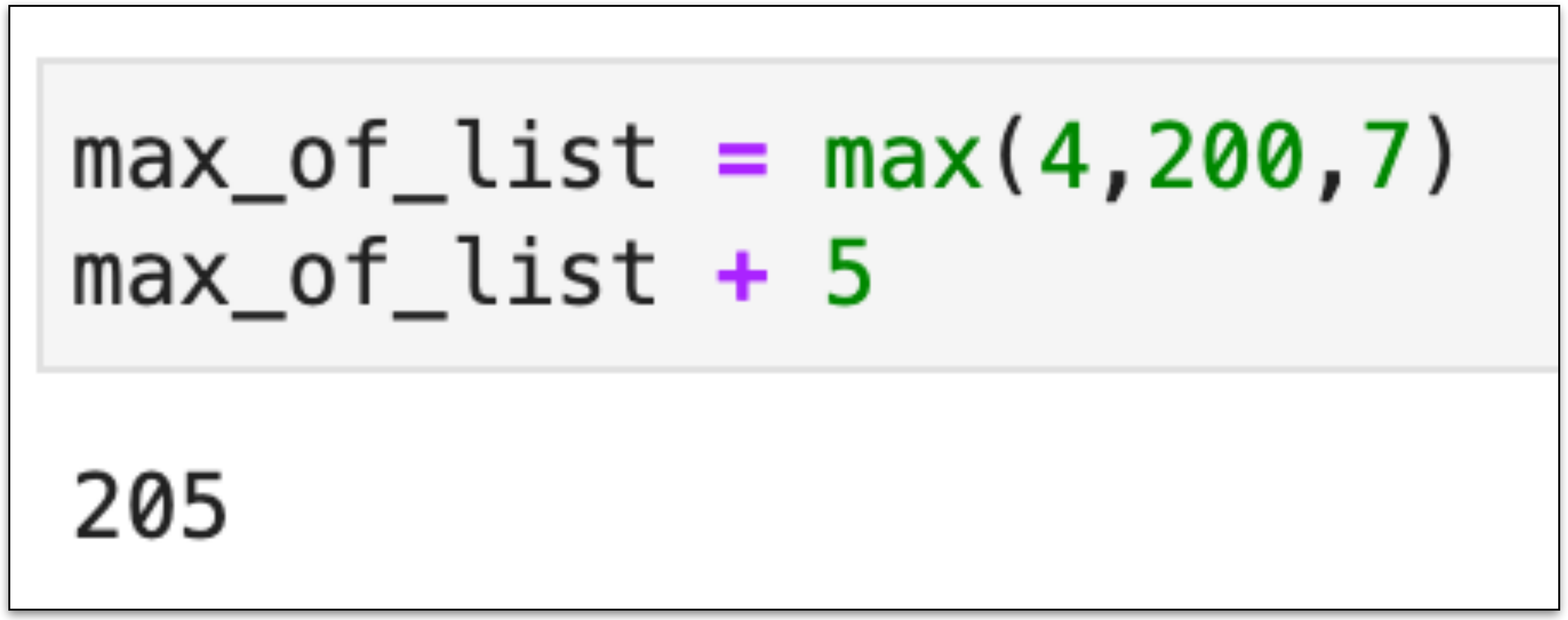
- You can convert values to a string using `str(..)`
  - `str(5)` becomes “5”
- You can convert strings of numbers to numbers
  - `int(’12’),float(‘1.2’)`

Questions:

1. What happens if you try to convert a float to an integer?
2. What happens if you try to convert non-numbers to a number?

Variables and Assignments

- **Assignments** change the meaning of the name to the left the = symbol
- **Variables** are values you can assign values to
    - "Variable" because they can change
- You can assign outputs of functions and operations to variables

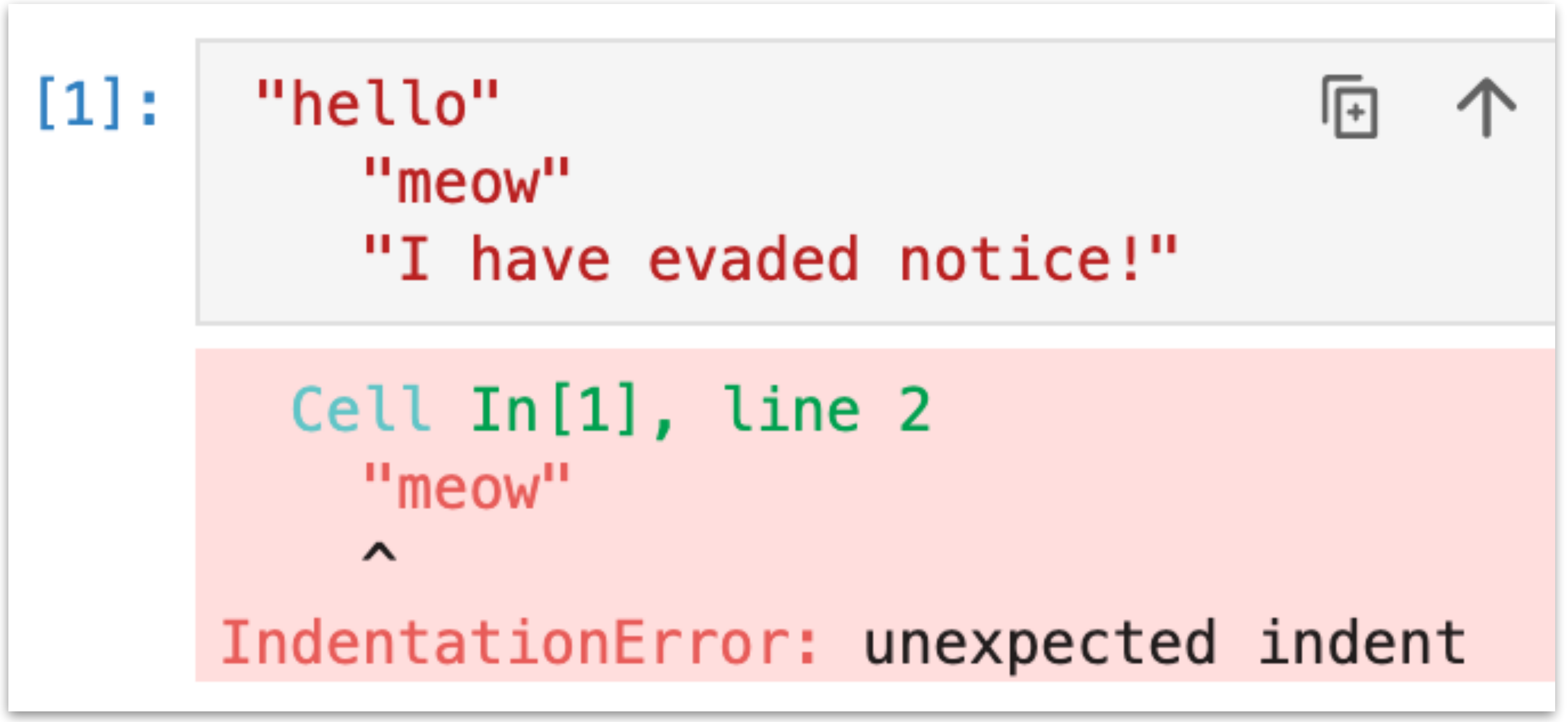


```
max_of_list = max(4,200,7)
max_of_list + 5
```

```
205
```

Python quirks

- Indentation and new lines matter
    - Be careful not to add extra spaces or indentations at the beginning of the line!
- Python runs line-by-line
    - It'll stop as soon as it runs into an issue and tell you what's wrong
- Lines starting with # are comments and are ignored



```
[1]: "hello"
        "meow"
        "I have evaded notice!"
```

```
  Cell In[1], line 2
    "meow"
    ^
IndentationError: unexpected indent
```

Types

Type	**Example**
Int	**3**
Float	3.0
String	‘three’
Boolean	True
Arrays	[1, 2, 3, 4]
Functions	abs(-5)

Type Exercise

Let’s say you have defined the following variables in your notebook

```
x = 3

y = ‘4’

z = ‘5.6’
```

What would the source of the error in these examples?

How could you fix it?

```
1. x + y

2. x + int(y + z)

3. str(x) + int(y)

4. y + float(z)
```

Type Exercise

Let’s say you have defined the following variables in your notebook

```
x = 3

y = ‘4’

z = ‘5.6’
```

What would the source of the error in this example:

```
x + y
```

How could you fix it?

Type Exercise

Let’s say you have defined the following variables in your notebook

```
x = 3

y = ‘4’

z = ‘5.6’
```

What would the source of the error in this example:

```
x + int(y + z)
```

How could you fix it?

Type Exercise

Let’s say you have defined the following variables in your notebook

```
x = 3

y = ‘4’

z = ‘5.6’
```

What would the source of the error in this example:

```
str(x) + int(y)
```

How could you fix it?

Type Exercise

Let’s say you have defined the following variables in your notebook

```
x = 3

y = ‘4’

z = ‘5.6’
```

What would the source of the error in this example:

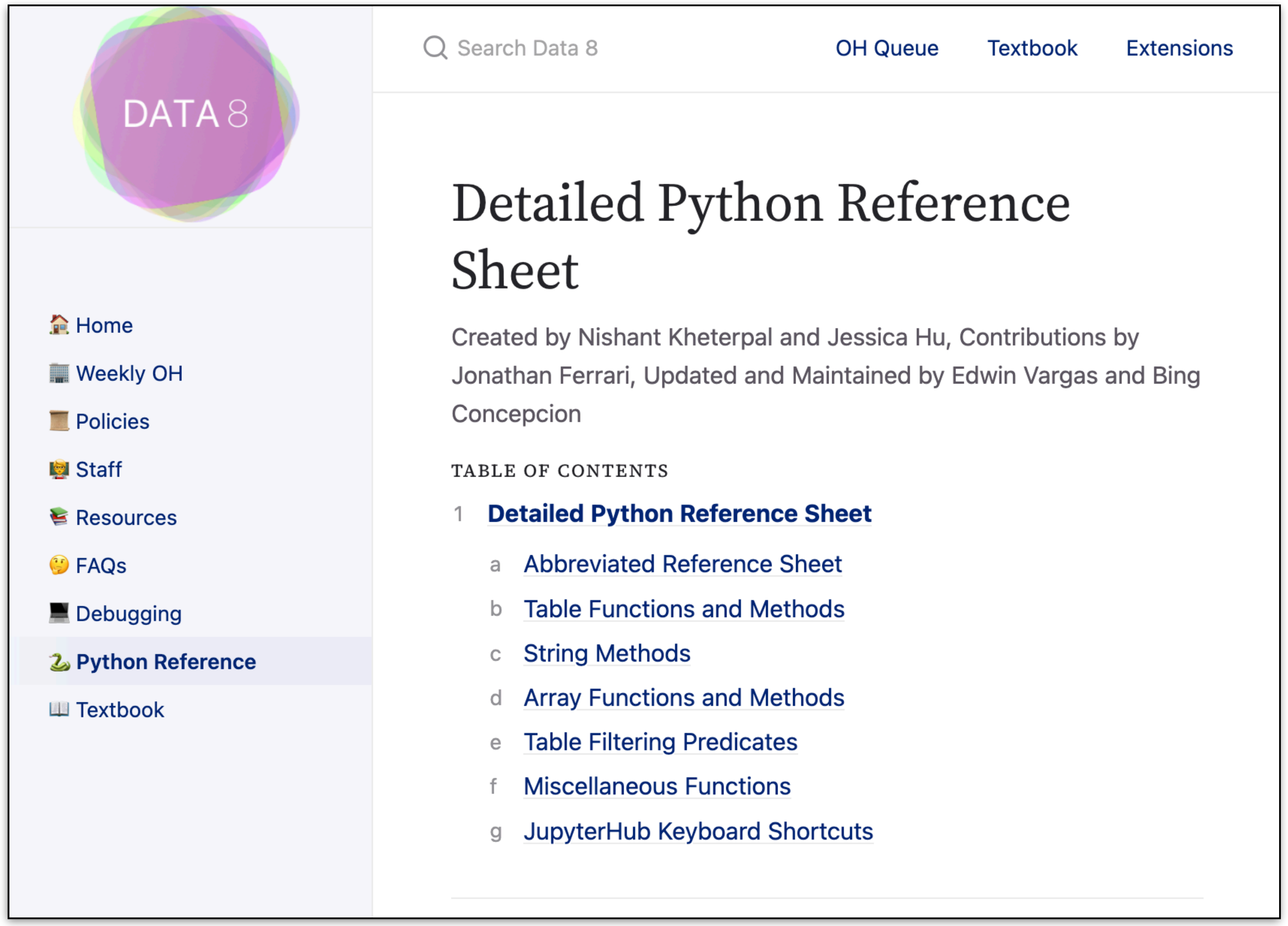
```
y + float(z)
```

How could you fix it?

Python Reference

https://www.data8.org/sp25/reference/

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



Search Data 8

OH Queue Textbook Extensions

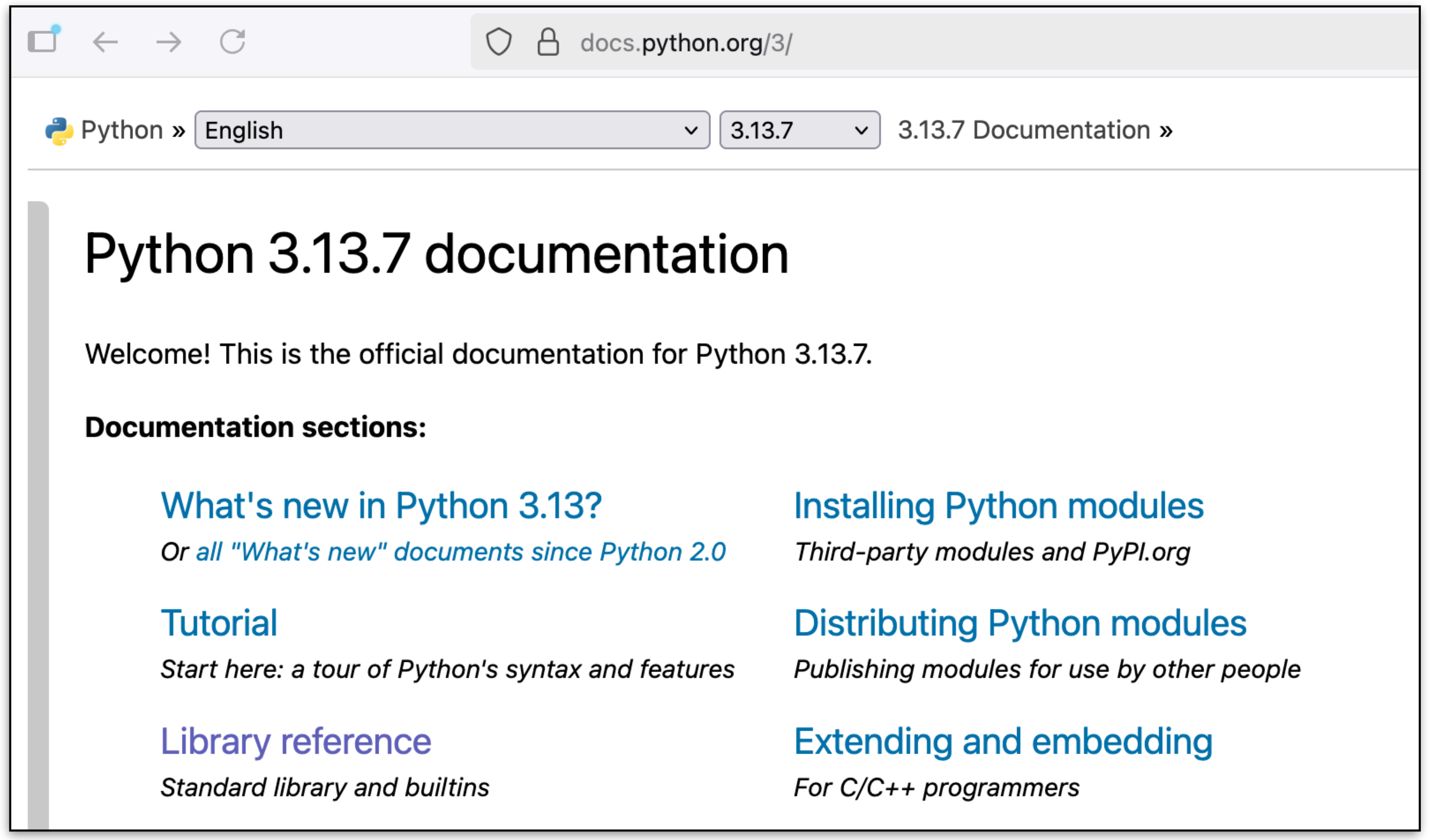
DATA 8

Home
Weekly OH
Policies
Staff
Resources
FAQs
Debugging
Python Reference
Textbook

Detailed Python Reference Sheet

Created by Nishant Kheterpal and Jessica Hu, Contributions by Jonathan Ferrari, Updated and Maintained by Edwin Vargas and Bing Concepcion

TABLE OF CONTENTS



docs.python.org/3/

Python » English 3.13.7 3.13.7 Documentation »

Python 3.13.7 documentation

Welcome! This is the official documentation for Python 3.13.7.

Documentation sections:

What's new in Python 3.13?
Or all "What's new" documents since Python 2.0

Installing Python modules
Third-party modules and PyPI.org

Tutorial
Start here: a tour of Python's syntax and features

Distributing Python modules
Publishing modules for use by other people

Library reference
Standard library and builtins

Extending and embedding
For C/C++ programmers

Next Class

- Today
  - Jupyter Notebooks
  - Data Types
- Wednesday
  - Tables
- Monday
  - Charts & Visualization