

Welcome to Python!

April 3, 2023



parth sarin (they/them)



tara jones (she/her)



Will Fang



Chase Joyner



Arpit Ranasaria

Sign up for sections on Canvas next Monday
(when you have a group)

Case Study: Hello World

Case Study: Hello World

// Java

Case Study: Hello World

```
// Java  
public class HelloWorld {  
    public static void main(String[ ] args) {  
        System.out.println("Hello world!");  
    }  
}
```

Case Study: Hello World

```
// Java  
  
public class HelloWorld {  
    public static void main(String[ ] args) {  
        System.out.println("Hello world!");  
    }  
}
```

```
$ javac HelloWorld.java  
$ java HelloWorld  
Hello world!
```

Case Study: Hello World

Case Study: Hello World

// C++

Case Study: Hello World

```
// C++
#include <iostream>
using namespace std;

int main() {
    cout << "Hello world!" << endl;
    return 0;
}
```

Case Study: Hello World

```
// C++
#include <iostream>
using namespace std;

int main() {
    cout << "Hello world!" << endl;
    return 0;
}
```

```
$ g++ helloWorld.cpp
$ ./a.out
Hello world!
```

Case Study: Hello World

Case Study: Hello World

```
print("Hello world!")
```

Case Study: Hello World

```
print("Hello world!")
```

```
$ python helloworld.py
Hello world!
```

Case Study: Int Size

Case Study: Int Size

// C++

Case Study: Int Size

```
// C++
#include <iostream>
using namespace std;

int main() {
    cout << sizeof(41) << endl;
    return 0;
}
```

Case Study: Int Size

```
// C++
#include <iostream>
using namespace std;

int main() {
    cout << sizeof(41) << endl;
    return 0;
}
```

```
$ g++ intSize.cpp
$ ./a.out
4
```

Case Study: Int Size

Case Study: Int Size

```
print((41).__sizeof__())
```

Case Study: Int Size

```
print((41).__sizeof__())
```

```
$ python intsize.py  
28
```

```
>>> import this
```

```
>>> 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.

```
>>> import this
```

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!

what this class *is*

- Deep dive into the Python programming language
 - Implementation details
 - Language design
 - Etc.
- Project-based learning – build things (physical and digital) with Python
- An introductory course :) – 106A and 106B / programming maturity are the prerequisites

what this class *is not*

- An introduction to programming – take 106A!
- A class about "low-level" vs. "high-level" programming – take 107!
- A class about artificial intelligence – take 229, 230n, etc.!

Learning Goals

Learning Goals

After CS 41, students will be able to...

1. Compare and contrast Python's language design with other languages they've seen before
2. Determine whether or not Python is an appropriate language to write a program in
3. Design the architecture of a Python project and implement it knowing which resources to consult
4. Work with a team more effectively by using pair programming resources and balancing work appropriately
5. Interact with and augment others' digital or physical creations using Python

Logistics

Logistics

Website

<https://stanfordpython.com>

Logistics

Website

<https://stanfordpython.com>

Ed

Discussion, Ed workspaces, link on the website

Logistics

Website

<https://stanfordpython.com>

Ed

Discussion, Ed workspaces, link on the website

Canvas

Section sign-up, link on the website ([assignments tab](#))

Logistics

Website

<https://stanfordpython.com>

Ed

Discussion, Ed workspaces, link on the website

Canvas

Section sign-up, link on the website ([assignments tab](#))

Groups

Link on the website ([assignments tab](#))

Logistics

Website	https://stanfordpython.com
Ed	Discussion, Ed workspaces, link on the website
Canvas	Section sign-up, link on the website (assignments tab)
Groups	Link on the website (assignments tab)
Assignments	3 assignments + a final project A0 is individual, the rest are in groups

Logistics

Website	https://stanfordpython.com
Ed	Discussion, Ed workspaces, link on the website
Canvas	Section sign-up, link on the website (assignments tab)
Groups	Link on the website (assignments tab)
Assignments	3 assignments + a final project A0 is individual, the rest are in groups
Grading	No grades on any assignments, feedback only Overall course grade is based on completion

Logistics

Website	https://stanfordpython.com
Ed	Discussion, Ed workspaces, link on the website
Canvas	Section sign-up, link on the website (assignments tab)
Groups	Link on the website (assignments tab)
Assignments	3 assignments + a final project A0 is individual, the rest are in groups
Grading	No grades on any assignments, feedback only Overall course grade is based on completion
Labs	Class on Tuesday, Lab on Thursday

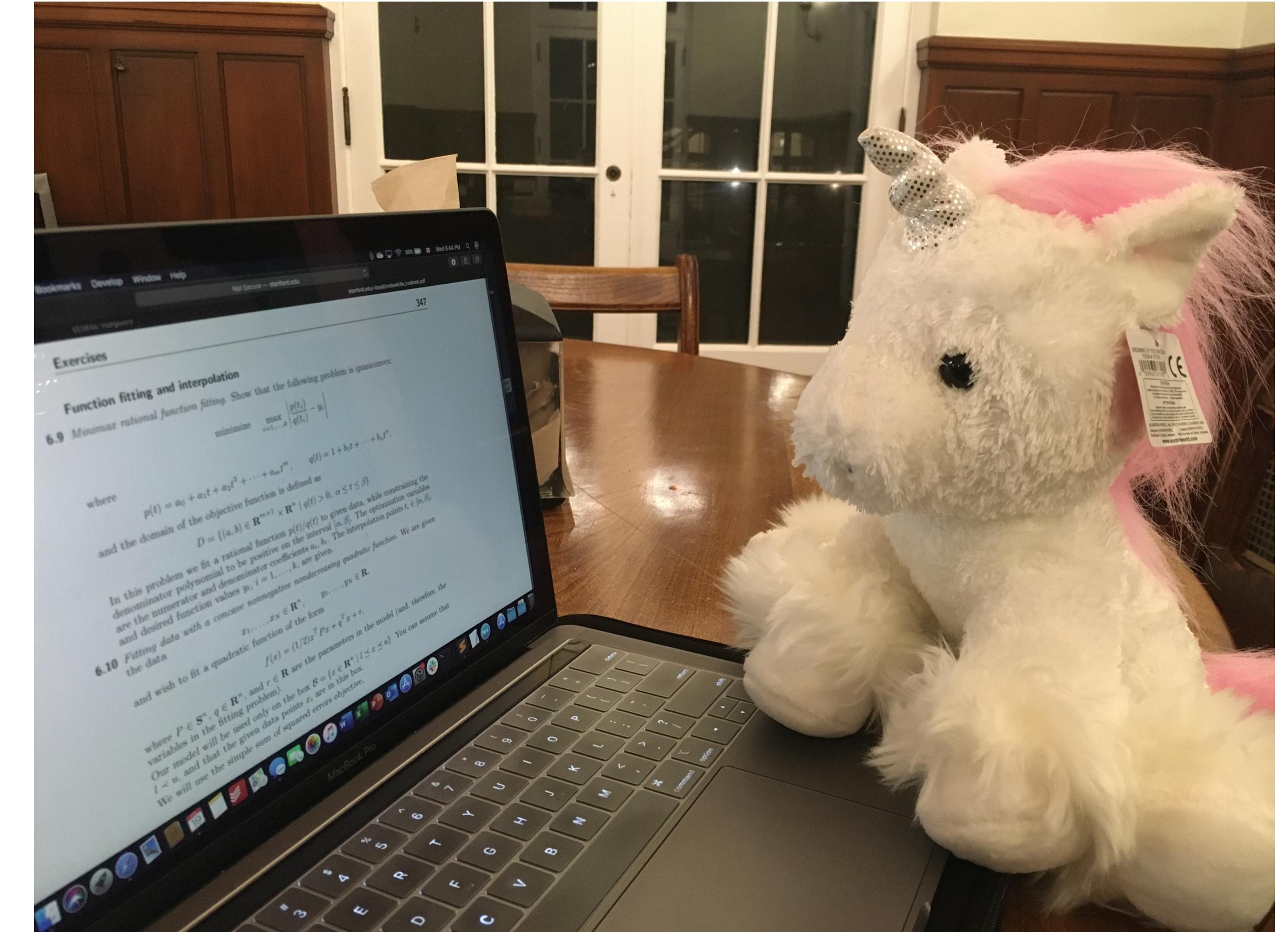
Logistics

Website	https://stanfordpython.com
Ed	Discussion, Ed workspaces, link on the website
Canvas	Section sign-up, link on the website (assignments tab)
Groups	Link on the website (assignments tab)
Assignments	3 assignments + a final project A0 is individual, the rest are in groups
Grading	No grades on any assignments, feedback only Overall course grade is based on completion
Labs	Class on Tuesday, Lab on Thursday
Playlist	Link on the website

Logistics

Website	https://stanfordpython.com
Ed	Discussion, Ed workspaces, link on the website
Canvas	Section sign-up, link on the website (assignments tab)
Groups	Link on the website (assignments tab)
Assignments	3 assignments + a final project A0 is individual, the rest are in groups
Grading	No grades on any assignments, feedback only Overall course grade is based on completion
Labs	Class on Tuesday, Lab on Thursday
Playlist	Link on the website
Masking	Masks recommended, safety is the priority

Unicornelius (any/all)



The Road Ahead



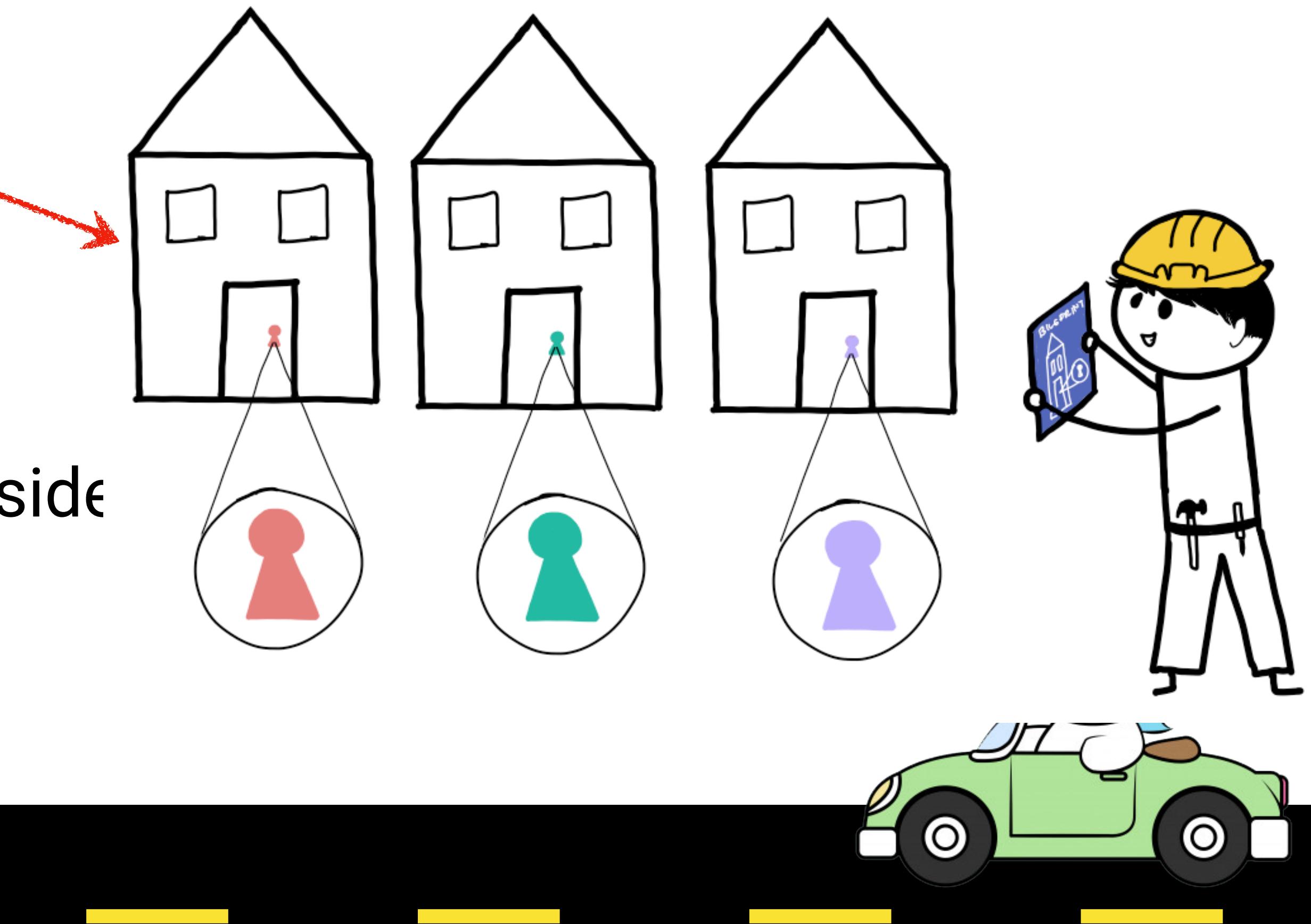
The Road Ahead

- Week 1:** Welcome & Basics
- Week 2:** Data Structures & OOP
- Week 3:** Electronics
- Week 4:** Python & the Web, APIs
- Week 5:** Python & the Web, Server-side



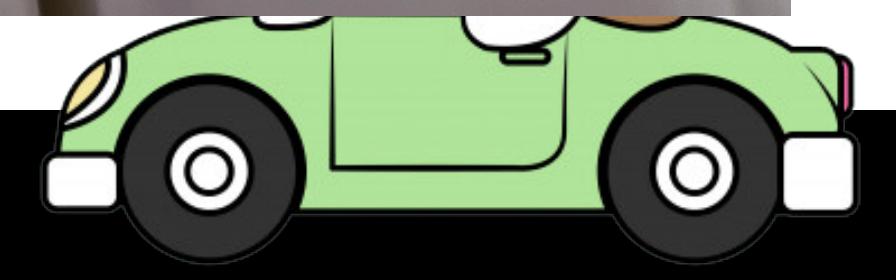
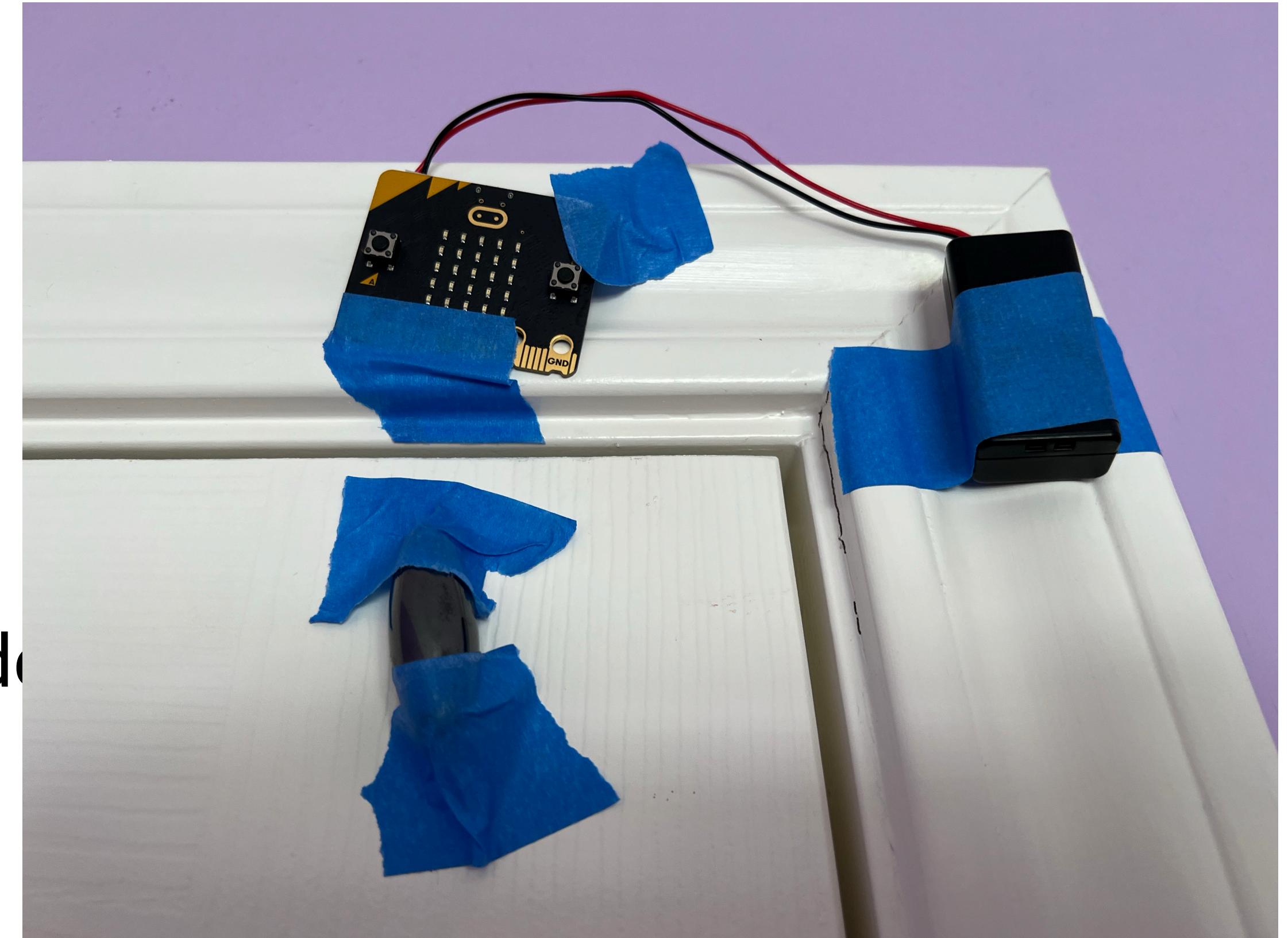
The Road Ahead

- Week 1:** Welcome & Basics
- Week 2:** Data Structures & OOP
- Week 3:** Electronics
- Week 4:** Python & the Web, APIs
- Week 5:** Python & the Web, Server-side



The Road Ahead

- Week 1:** Welcome & Basics
- Week 2:** Data Structures & OOP
- Week 3:** Electronics 
- Week 4:** Python & the Web, APIs
- Week 5:** Python & the Web, Server-side



The Road Ahead

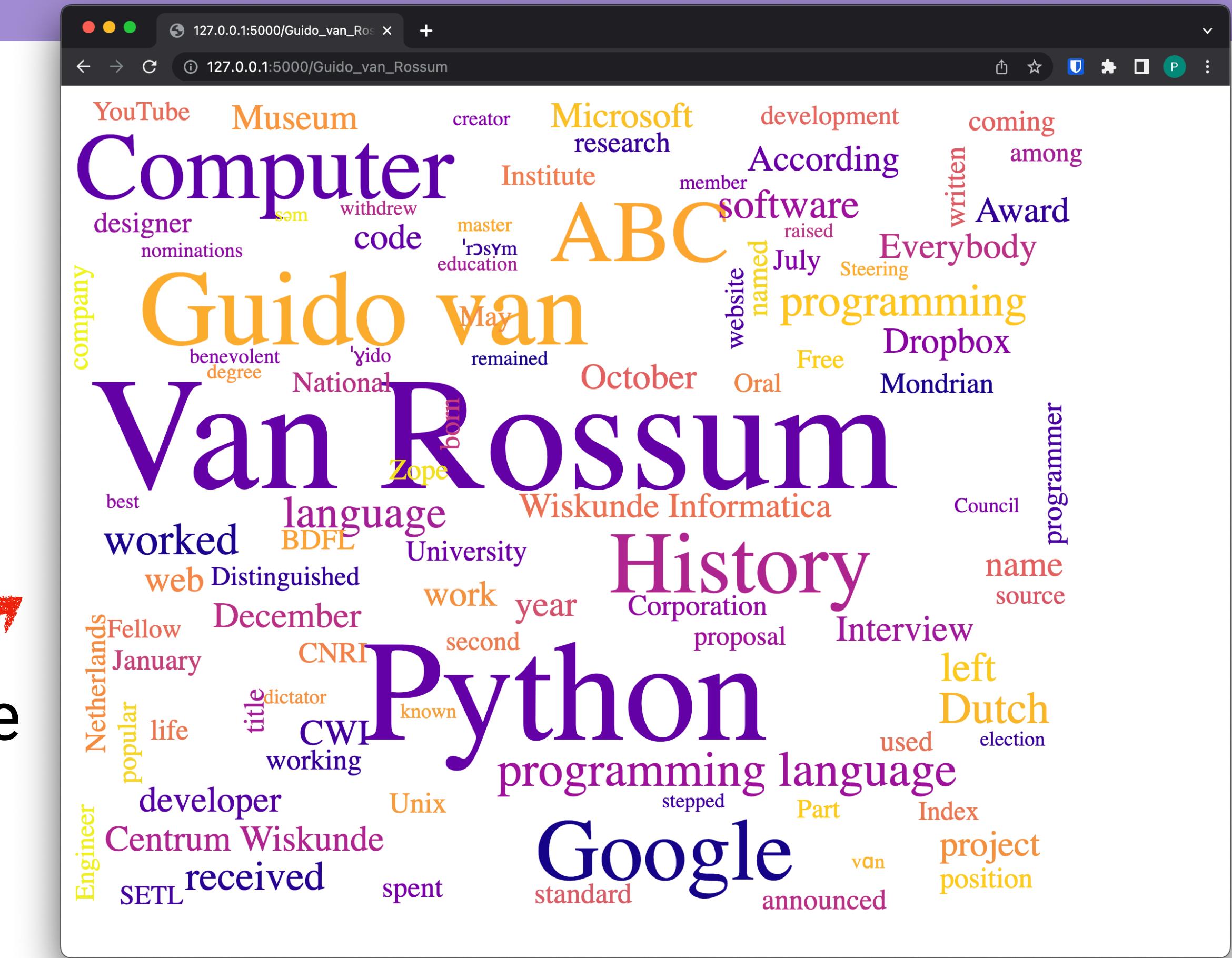
- Week 1:** Welcome & Basics
- Week 2:** Data Structures & OOP
- Week 3:** Electronics
- Week 4:** Python & the Web, APIs
- Week 5:** Python & the Web, Server-side

```
import requests
r = requests.get(
    'https://karl joke.herokuapp.com/jokes/random'
)
# This is raw bytes data...
r.content
# => b'{"id":86,"type":"general","setup":"Did you hear about the bread
factory burning down?", "punchline":"They say the business is toast."}'
# ...this is a string
# => '{"id":86,"type":"general","setup":"Did you hear about the bread
factory burning down?", "punchline":"They say the business is toast."}'
r.json()
# => {'id': 86, 'type': 'general', 'setup': 'Did you hear about the
bread factory burning down?', 'punchline': 'They say the business is
toast.'}
# ...and this is a dictionary
```



The Road Ahead

- | | |
|----------------|-------------------------------|
| Week 1: | Welcome & Basics |
| Week 2: | Data Structures & OOP |
| Week 3: | Electronics |
| Week 4: | Python & the Web, APIs |
| Week 5: | Python & the Web, Server-side |



The Road Ahead

- Week 1:** Welcome & Basics
- Week 2:** Data Structures & OOP
- Week 3:** Electronics
- Week 4:** Python & the Web, APIs
- Week 5:** Python & the Web, Server-side



The Road Ahead

- Week 6:** Low-Level & Embedded Python
- Week 7:** Functions & Functional Programming
- Week 8:** Standard & Third-Party Libraries
- Week 9:** Gaming
- Week 10:** Final Project Presentations



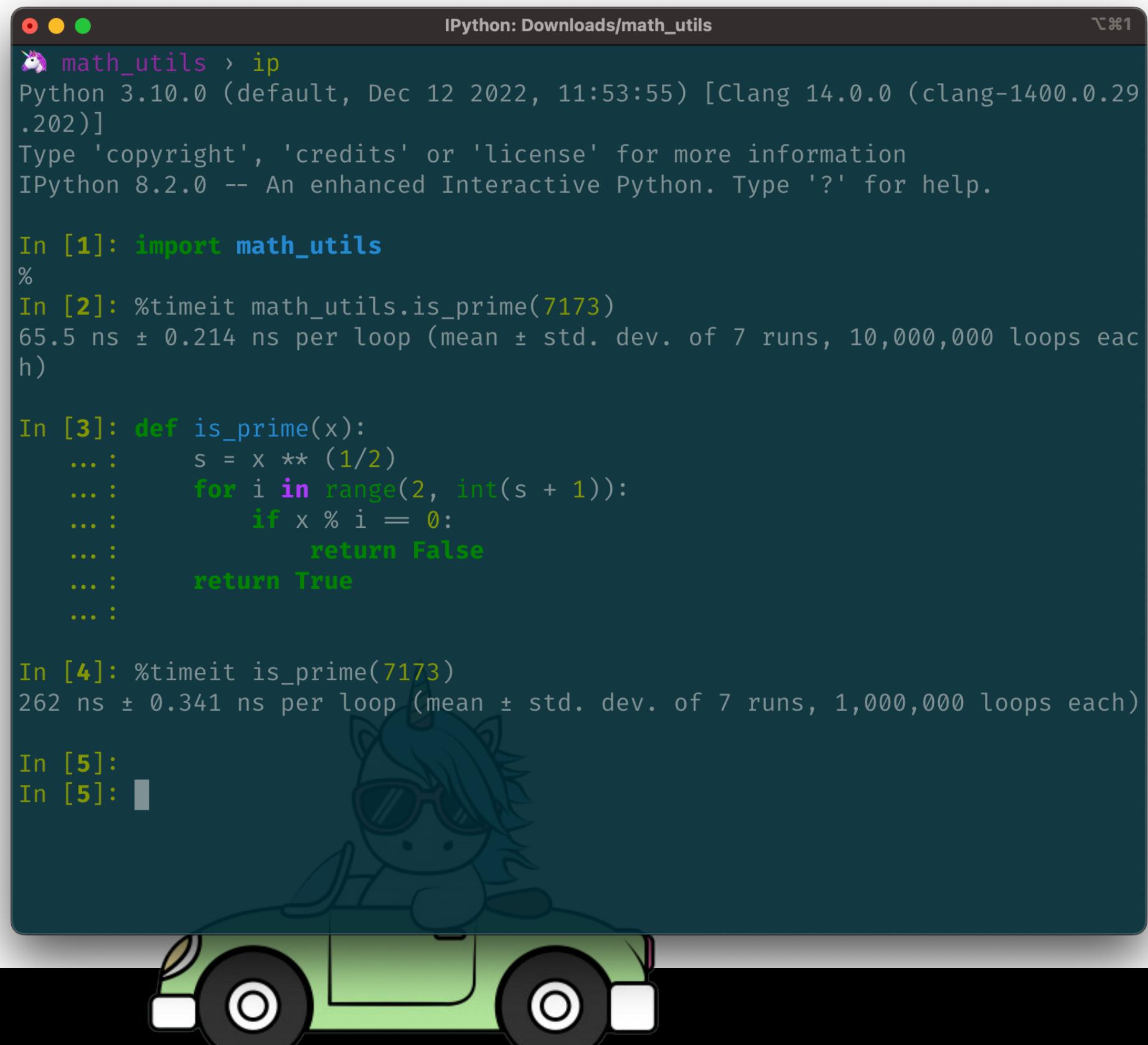
Is it finally time to remove the Python GIL?

The global interpreter lock is both a key component of the Python runtime and a major obstacle to multithreading. These are the plans to get around it or get rid of it.



By **Serdar Yegulalp**

Senior Writer, InfoWorld | FEB 27, 2023 2:00 AM PST



```
IPython: Downloads/math_utils
```

```
math_utils > ip
Python 3.10.0 (default, Dec 12 2022, 11:53:55) [Clang 14.0.0 (clang-1400.0.29.202)]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.2.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: import math_utils
%
In [2]: %timeit math_utils.is_prime(7173)
65.5 ns ± 0.214 ns per loop (mean ± std. dev. of 7 runs, 10,000,000 loops each)

In [3]: def is_prime(x):
...:     s = x ** (1/2)
...:     for i in range(2, int(s + 1)):
...:         if x % i == 0:
...:             return False
...:     return True
...:

In [4]: %timeit is_prime(7173)
262 ns ± 0.341 ns per loop (mean ± std. dev. of 7 runs, 1,000,000 loops each)

In [5]:
```

Road Ahead

Week 6:

Low-Level & Embedded Python

Week 7:

Functions & Functional Programming

Week 8:

Standard & Third-Party Libraries

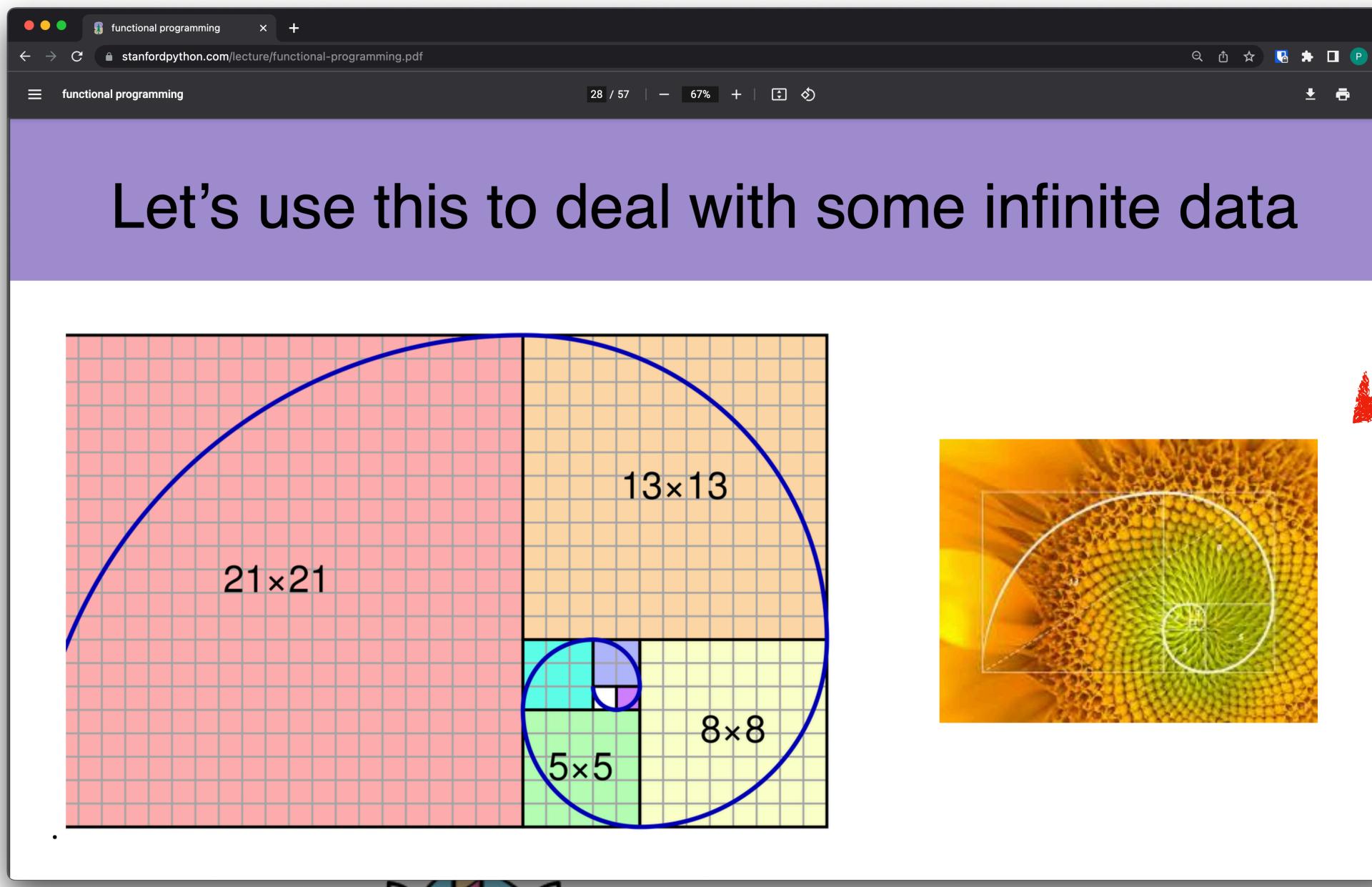
Week 9:

Gaming

Week 10:

Final Project Presentations

The Road Ahead



- Week 6:** Low-Level & Embedded Python
- Week 7:** Functions & Functional Programming
- Week 8:** Standard & Third-Party Libraries
- Week 9:** Gaming
- Week 10:** Final Project Presentations

The Road Ahead

The screenshot shows the Python pickle module documentation page. The page includes a table of contents, a note about serialization vs persistence, and sections on the module interface, constants, and functions. A red arrow points from the 'Week 8' section towards the 'pickle' module documentation.



Week 6:

Week 7:

← Week 8:

Week 9:

Week 10:

Low-Level & Embedded Python

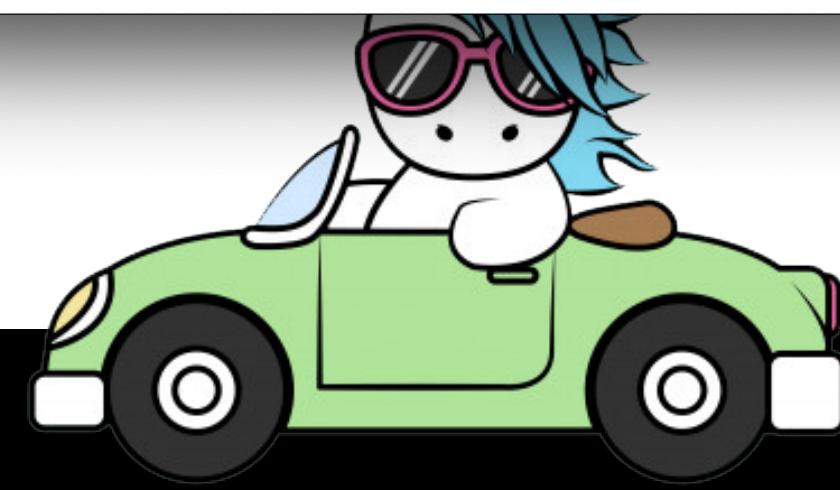
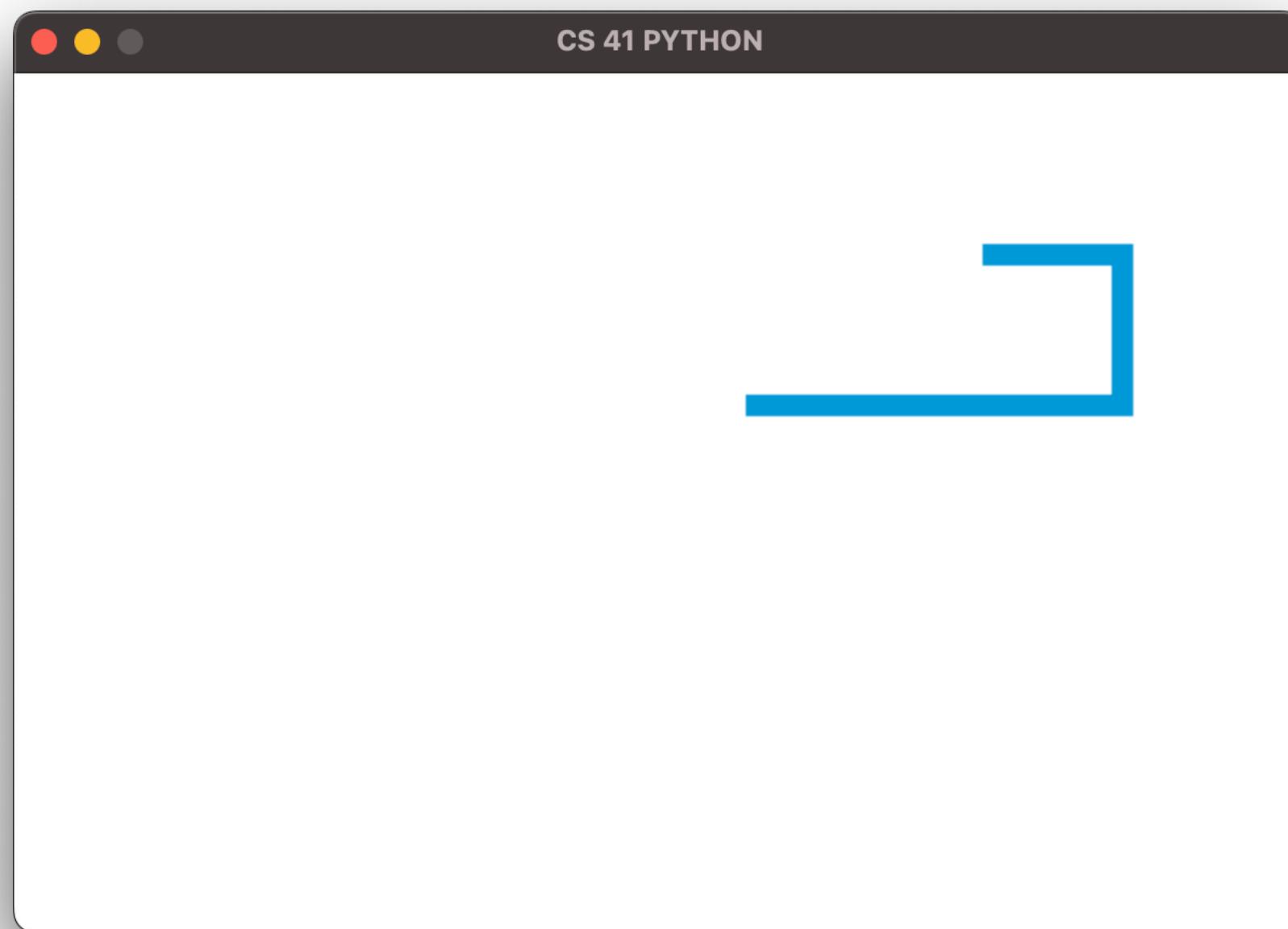
Functions & Functional Programming

Standard & Third-Party Libraries

Gaming

Final Project Presentations

The Road Ahead



- Week 6:** Low-Level & Embedded Python
- Week 7:** Functions & Functional Programming
- Week 8:** Standard & Third-Party Libraries
- Week 9:** Gaming
- Week 10:** Final Project Presentations

The Road Ahead

- Week 6:** Low-Level & Embedded Python
- Week 7:** Functions & Functional Programming
- Week 8:** Standard & Third-Party Libraries
- Week 9:** Gaming
- Week 10:** Final Project Presentations

