Introduction to Python
Wi24 Section B

Welcome!

Brian Hempel - bhempel@ucsd.edu - January 9, 2024

Instructor

Dr. Brian Hempel
Direct Manipulation
Programming
Interfaces

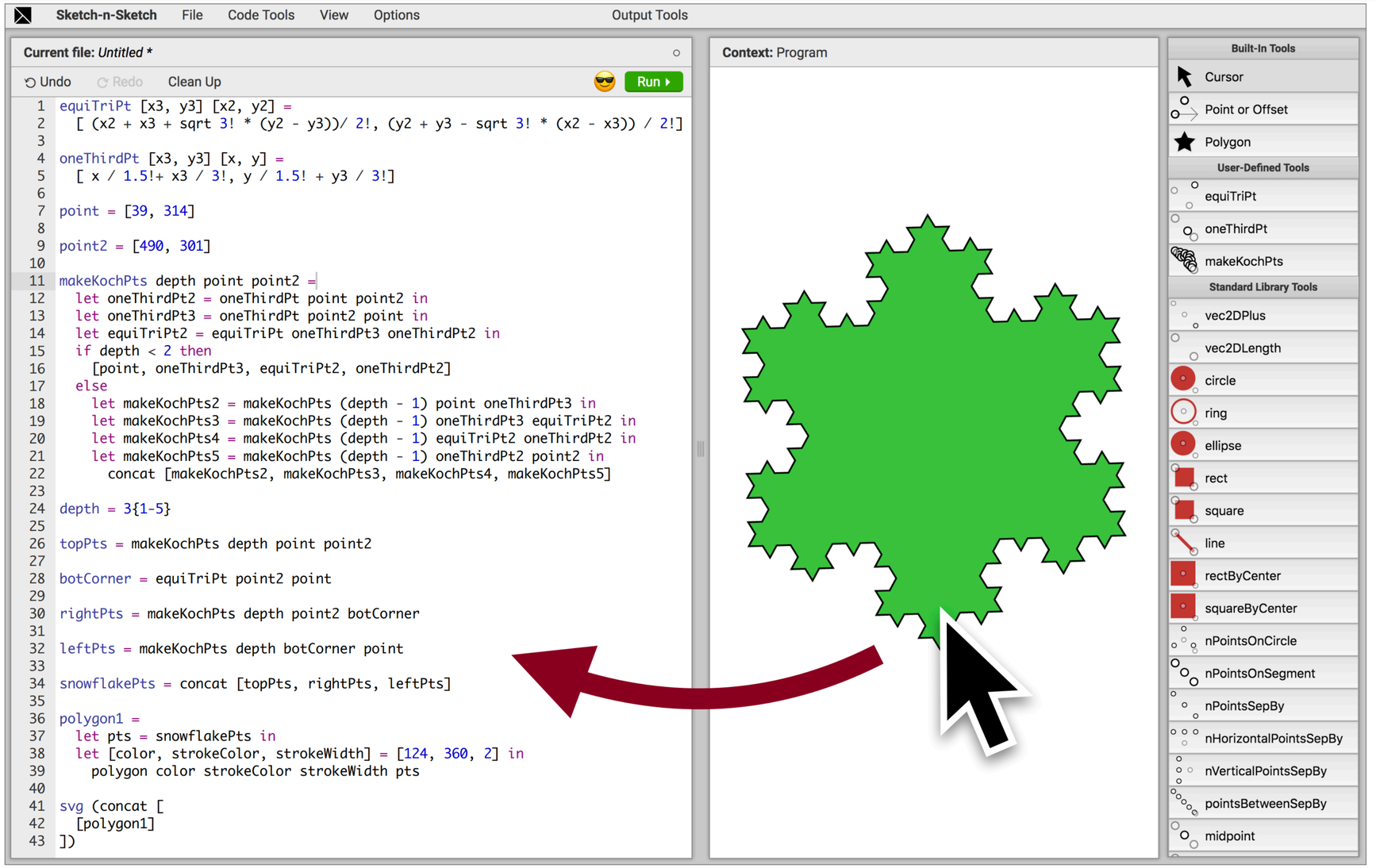
TAs

Unay Shah - 2nd Year ECE Masters
Sihan Yang - 2nd Year CogSci PhD
Hanqing “Helen” Zhao - 2nd Year CSE Masters

IAs

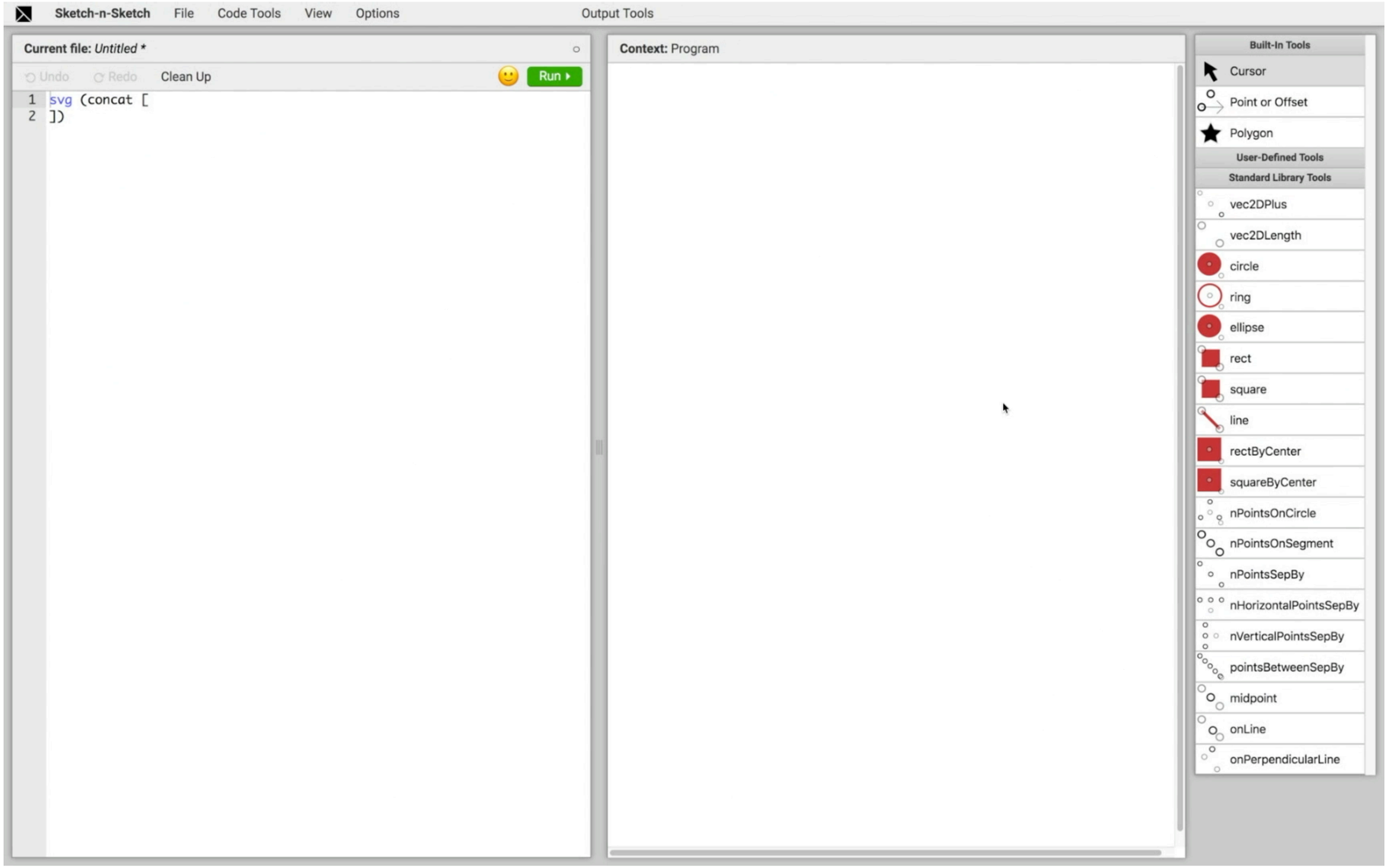
TBA

My Research: Direct Manipulation Programming

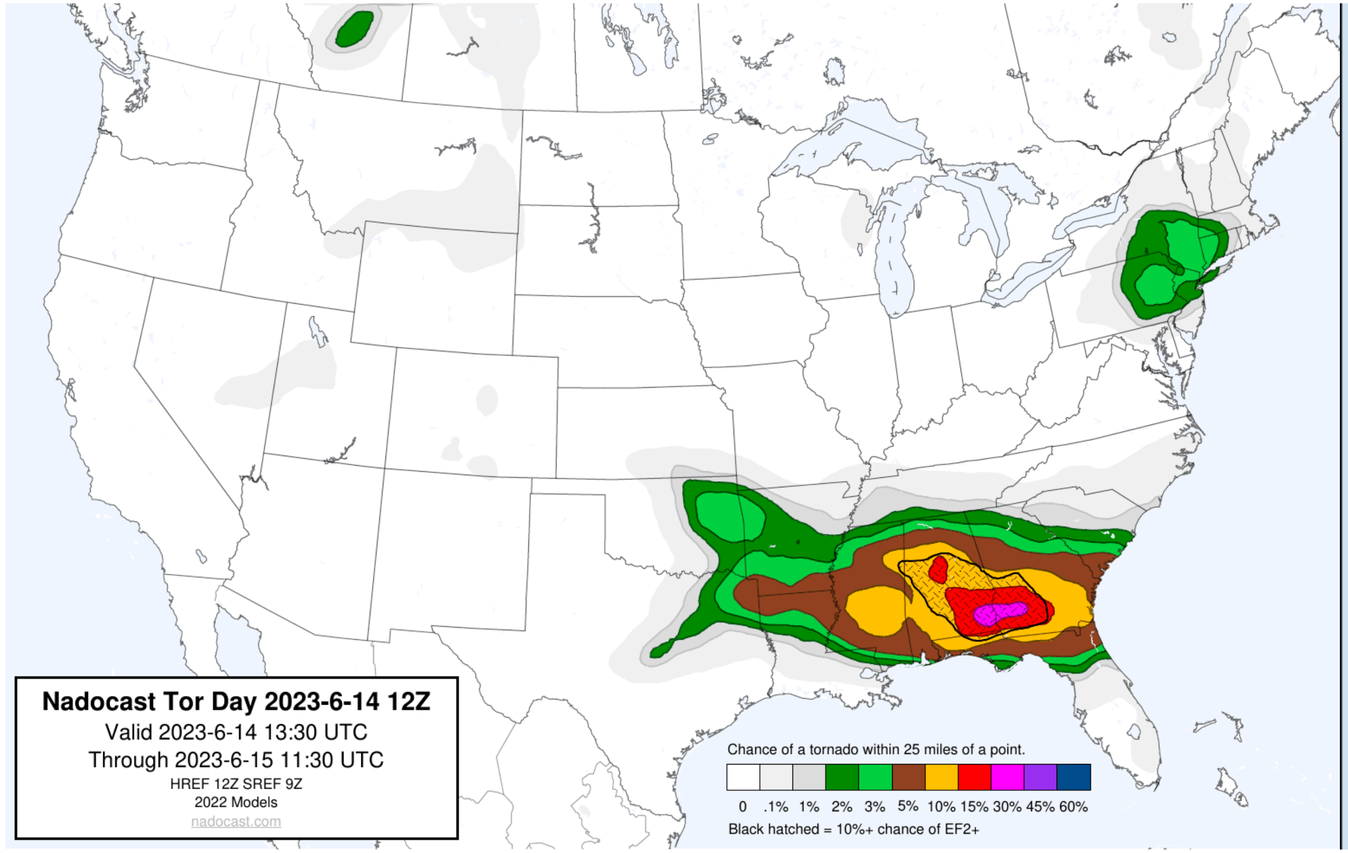


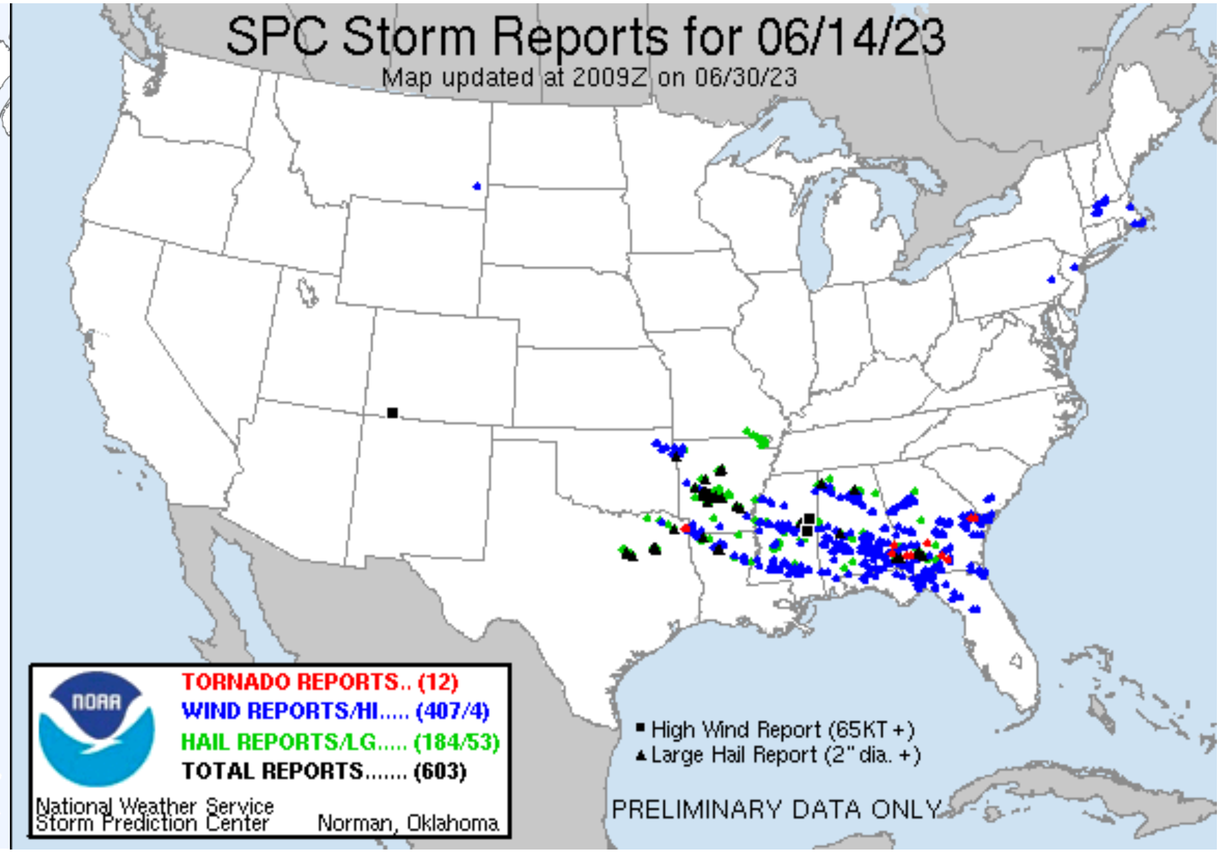

```
equiTriPt [x3, y3] [x2, y2] =
  [ (x2 + x3 + sqrt 3! * (y2 - y3))/ 2!, (y2 + y3 - sqrt 3! * (x2 - x3)) / 2!]

oneThirdPt [x3, y3] [x, y] =
  [ x / 1.5!+ x3 / 3!, y / 1.5! + y3 / 3!]

point = [39, 314]

point2 = [490, 301]

makeKochPts depth point point2 =
  let oneThirdPt2 = oneThirdPt point point2 in
  let oneThirdPt3 = oneThirdPt point2 point in
  let equiTriPt2 = equiTriPt oneThirdPt3 oneThirdPt2 in
  if depth < 2 then
    [point, oneThirdPt3, equiTriPt2, oneThirdPt2]
  else
    let makeKochPts2 = makeKochPts (depth - 1) point oneThirdPt3 in
    let makeKochPts3 = makeKochPts (depth - 1) oneThirdPt3 equiTriPt2 in
    let makeKochPts4 = makeKochPts (depth - 1) equiTriPt2 oneThirdPt2 in
    let makeKochPts5 = makeKochPts (depth - 1) oneThirdPt2 point2 in
      concat [makeKochPts2, makeKochPts3, makeKochPts4, makeKochPts5]

depth = 3{1-5}

topPts = makeKochPts depth point point2

botCorner = equiTriPt point2 point

rightPts = makeKochPts depth point2 botCorner

leftPts = makeKochPts depth botCorner point

snowflakePts = concat [topPts, rightPts, leftPts]

polygon1 =
  let pts = snowflakePts in
  let [color, strokeColor, strokeWidth] = [124, 360, 2] in
    polygon color strokeColor strokeWidth pts

svg (concat [
  [polygon1]
])
```

My Research: Direct Manipulation Programming



My Hobby Research: Severe Weather Prediction





Tornado Prediction
June 14, 2023

Red Dots = Actual Tornados

Introduction to Python

Section B

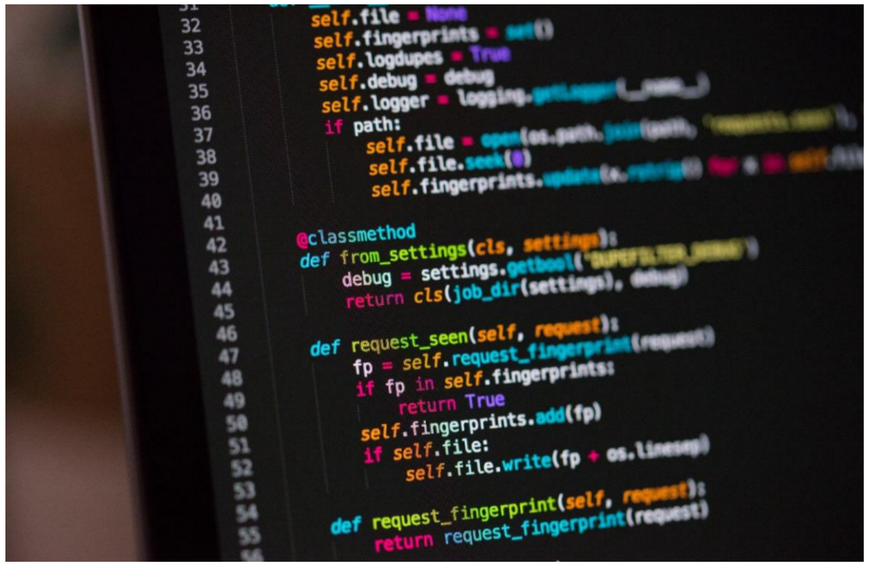
Waitlist

I do not handle it!

- Email cogsadvising@ucsd.edu
- Typically ~3-5 students from each section are enrolled by our staff.
- Enrollment cannot exceed the number of seats in the classroom.
- The waitlist clears at the end of week 2.

Success

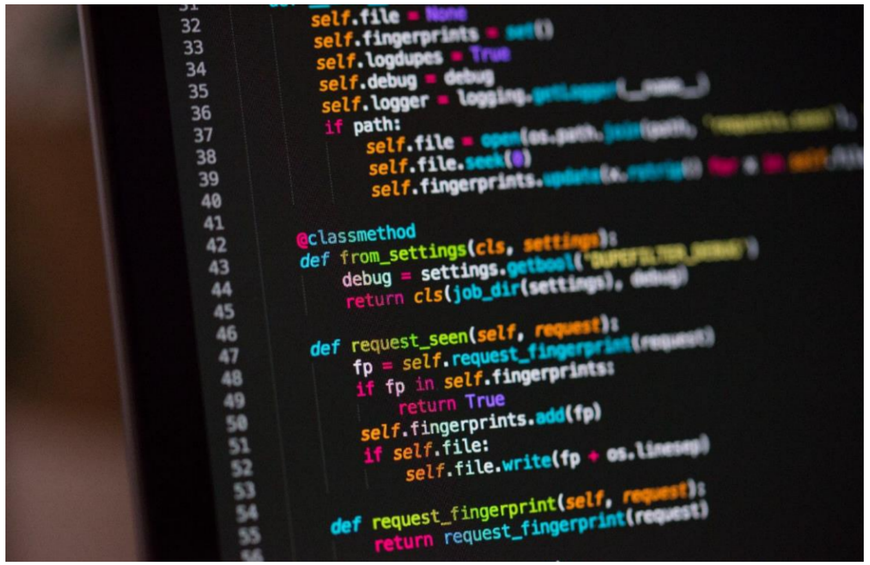
You, too, will learn Python.



The only thing that is slightly predictive of success in an intro programming course is...

Success

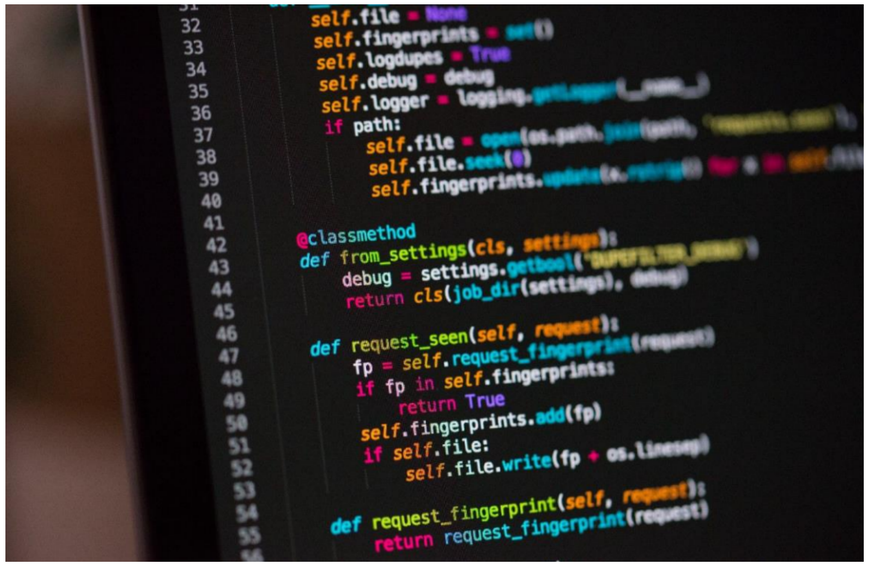
You, too, will learn Python.



The only thing that is slightly predictive of success in an intro programming course is...
how successful the student thinks they will be.

Success

You, too, will learn Python.



The only thing that is slightly predictive of success in an intro programming course is...
how successful the student thinks they will be.

Things that do NOT predict success:

- gender
- age
- personality
- **math ability**

Programming is Concepts You Already Know...

...just expressed precisely for the computer.

Naming stuff

if you can slap a Post-It note on something, you can program

“Yes” and **“No”**

Numbers

addition, subtraction, *occasionally* multiplication

Text

Lists

If *something* **Then** *do something* **Otherwise** *do something else*

Repeating stuff

Doing the same thing, but with a slight variation

It takes years to become an expert.

But, by the end of this course, you will be able to program at an introductory level.

That is my goal.

We are learning Python! It is:

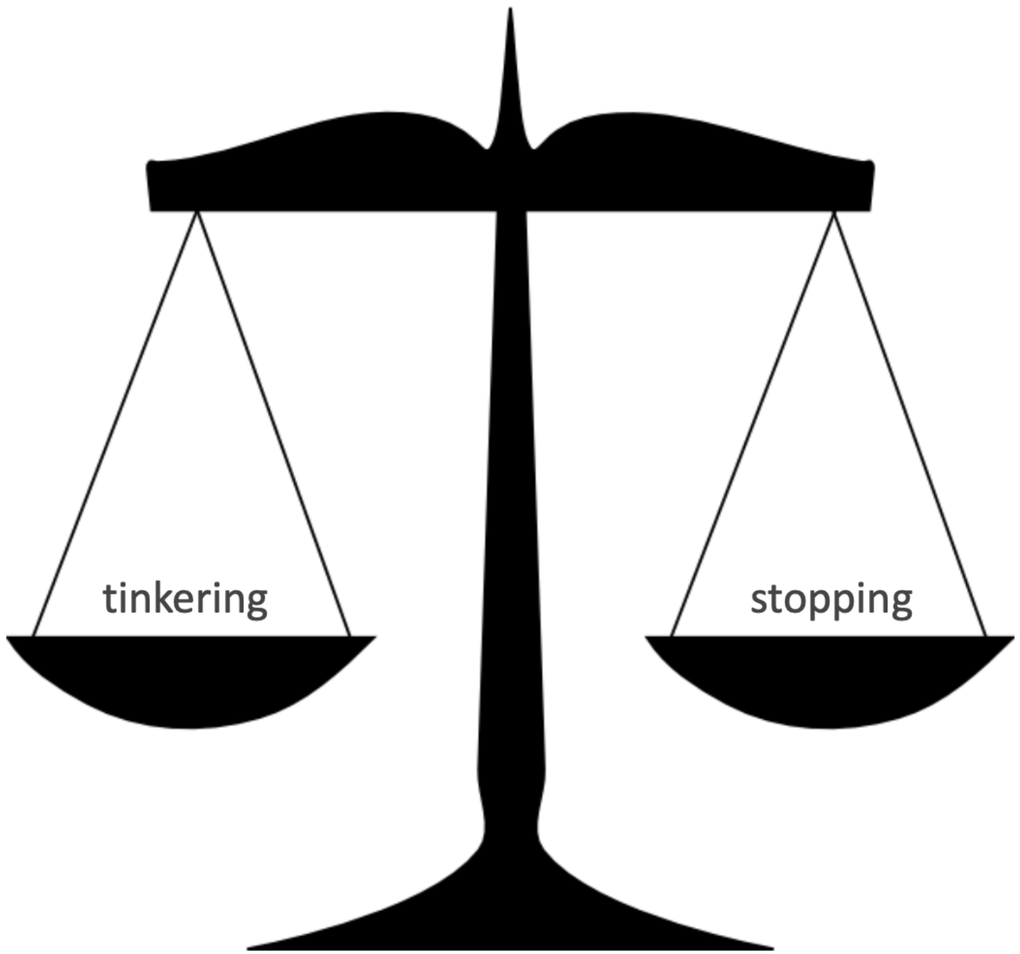
- **Widely used:** de facto language for scientific computing.
- **Simple.**
- **Not bad.**
  “It’s not the best language for anything, but it’s the second best for everything.” -Brad Voytek

Programming is *not* book learning.

The only way is practice, practice, practice.

To avoid the common pitfalls of intro programming courses, we're going to take the following approach:

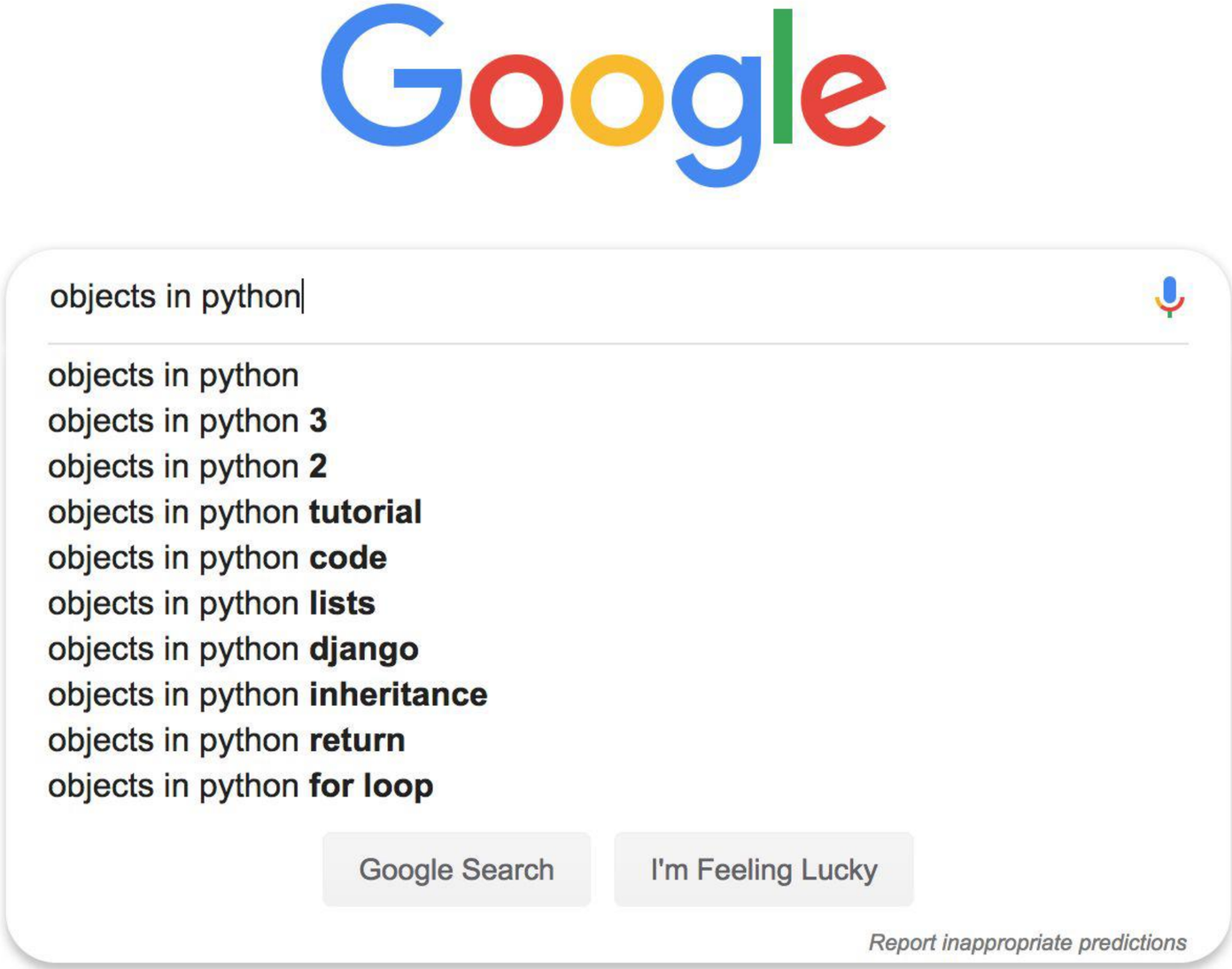
1. First 2/3 of course: basic concepts
2. In-class practice (no stakes)
   - In-notebook questions for comprehension
3. Coding Labs (low stakes)
   - Notebooks provided
   - Staff/classmates there to help
   - Checked for effort, not correctness
4. Assignments (mid stakes)
   - After every lecture
   - Completed individually (can work together)
   - Programmatically graded
5. Exams (high stakes)
   - Two parts: conceptual (in-class) + technical (take-home)
   - Completed totally individually



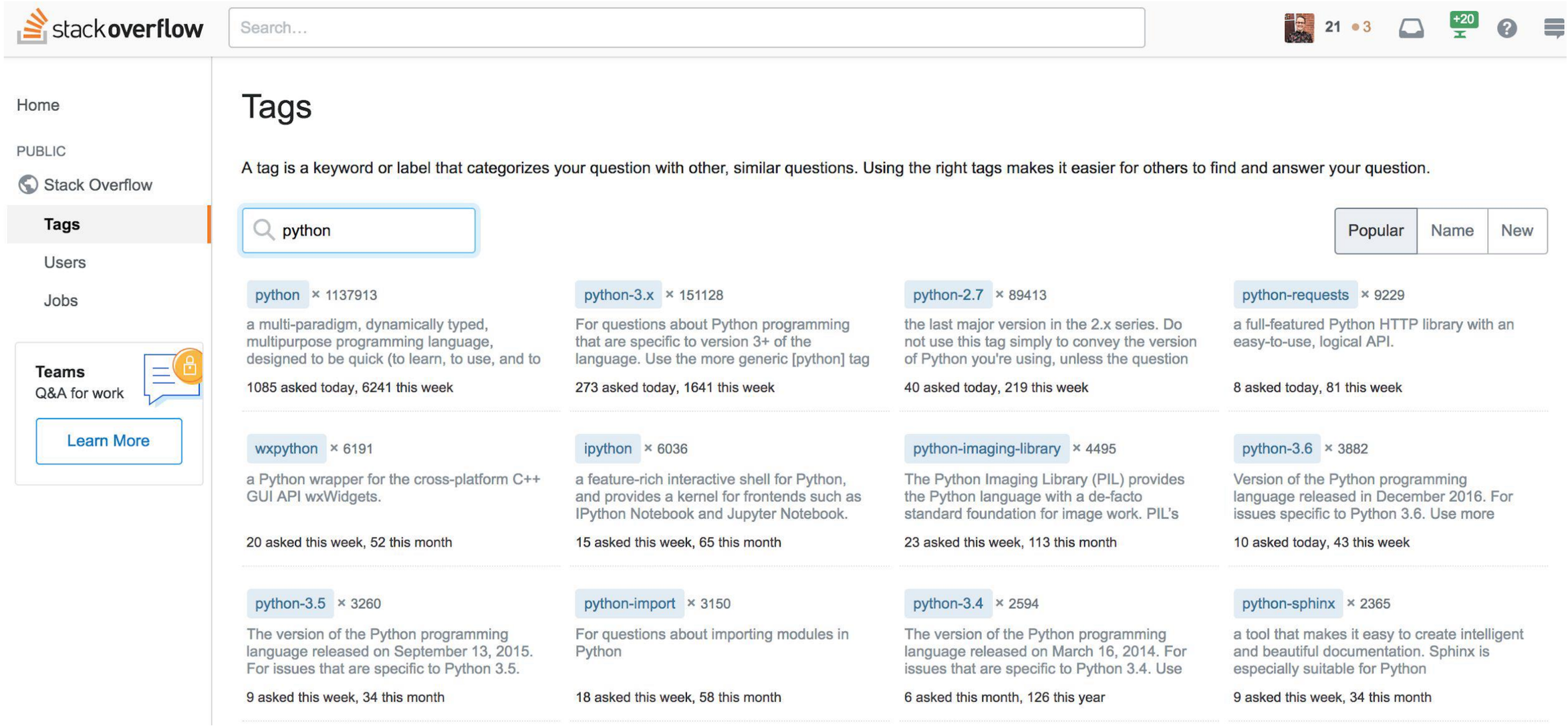
Be a mover: Make forward progress. Strike a balance between just stopping and tinkering forever.

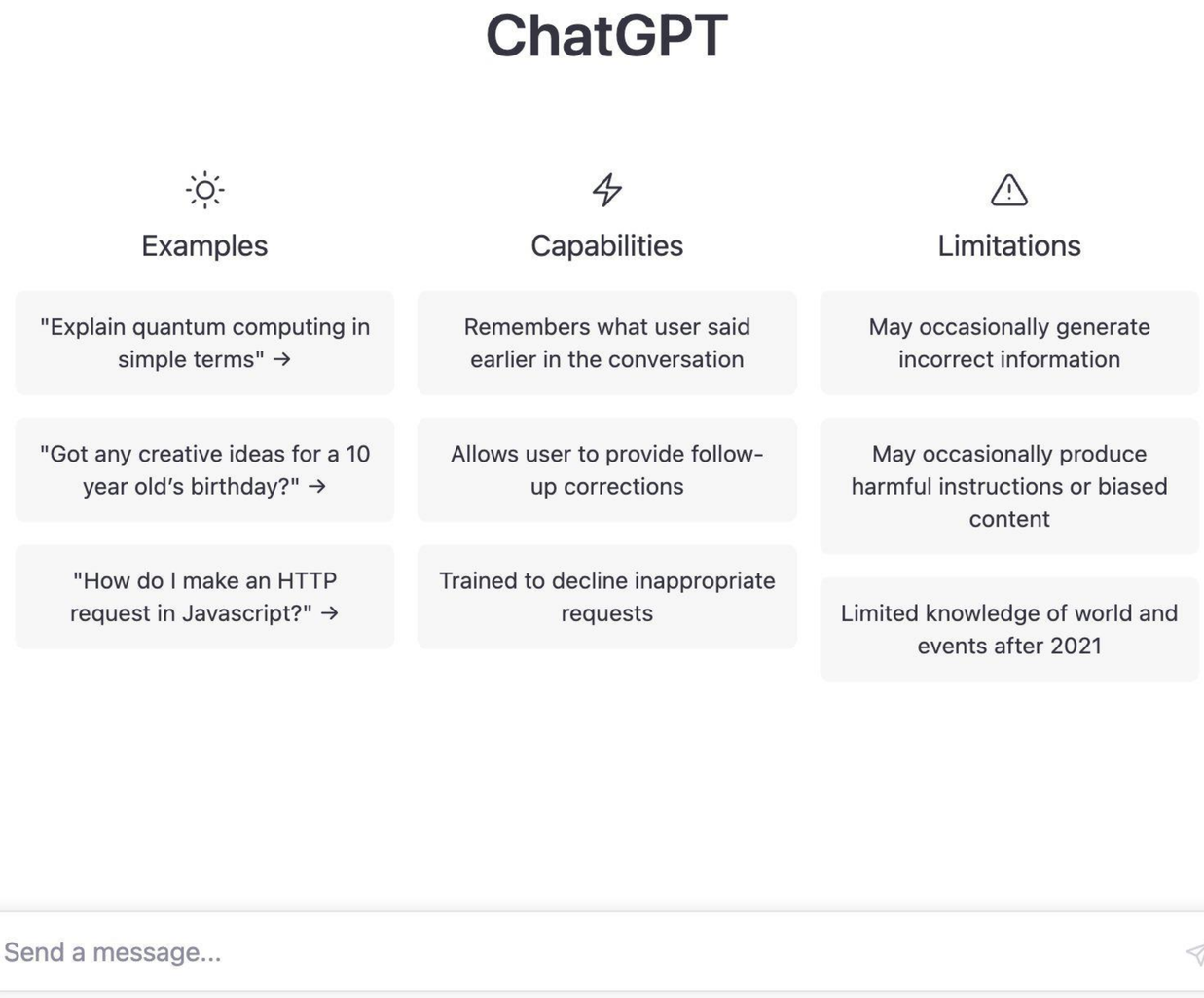
- Try for 30min.
- Take a walk.
- Try again.
- Seek help!

Including “in python” in your Google search can be magic



StackOverflow probably has the answer to your question





ChatGPT

A conversational LLM that will produce prose and code. It has pros (you can get your questions answered!) and cons (it can be confidently wrong).

Good ChatGPT prompt: “What is an example of _____ in Python?”

What is an example of variables in Python?

In Python, variables are used to store and manipulate data. Here's an example of variables in Python:

python Copy code

```python
# Integer variable
age = 30

# String variable
name = "John Doe"

# Floating-point variable
salary = 50000.50
```

Don’t ask ChatGPT what you wouldn’t ask a classmate.

You wouldn’t ask a classmate to just do a question you are stuck on.

Just use “What is an example of…” queries to ChatGPT.

There are also COGS18-specific avenues when looking for help

Questions in CodingLabs, coming to office hours, talking to your classmates, or reaching out for help on Piazza are all options for you. You're encouraged to help one another on Piazza!

YESTERDAY

Instr **Welcome to Piazza!** 06:40 PM
Hello COGS 18 Students!Piazza is an incredible resource for technical classes. It gives you a place to post questions an

Welcome to Piazza! 06:31 PM
Piazza is a Q&A platform designed to get you great answers from classmates and instructors fast. We've put together thi

note @1 1 view

Actions

Welcome to Piazza!

Piazza is a Q&A platform designed to get you great answers from classmates and instructors fast. We've put together this list of tips you might find handy as you get started:

1. **Ask questions!**

The best way to get answers is to ask questions! Ask questions on Piazza rather than emailing your teaching staff so everyone can benefit from the response (and so you can get answers from classmates who are up as late as you are).

2. **Edit questions and answers wiki-style.**

Think of Piazza as a Q&A wiki for your class. Every question has just a single **students' answer** that students can edit collectively (and a single **instructors' answer** for instructors).

3. **Add a followup to comment or ask further questions.**

To comment on or ask further questions about a post, start a **followup discussion**. Mark it resolved when the issue has been addressed, and add any relevant information back into the Q&A above.

4. **Go anonymous.**

Shy? No problem. You can always opt to post or edit anonymously.

5. **Tag your posts.**

It's far more convenient to find all posts about your Homework 3 or Midterm 1 when the posts are tagged. Type a "#" before a key word to tag. Click a blue tag in a post or the question feed to filter for all posts that share that tag.

6. **Format code and equations.**

Adding a code snippet? Click the **pre** or **tt** button in the question editor to add pre-formatted or inline teletype text.
Mathematical equation? Click the **Fx** button to access the LaTeX editor to build a nicely formatted equation.

7. **View and download class details and resources.**

Click the **Course Page** button in your top bar to access the class syllabus, staff contact information, office hours details, and course resources—all in one place!

Contact the Piazza Team anytime with questions or comments at **team@piazza.com**. We love feedback!

In technical classes, Piazza is a particularly helpful resource

There are rules:

1. No duplicates.
2. Include Assignment & Question in Summary line.
3. Posts must include your question, what you've tried so far, and resources used.
4. Public posts are best.
5. Helping one another is encouraged.
6. No assignment code in public posts.
7. We're not robots.

A message for first-gen students, transfer students, and those who don't have older siblings/friends who have attended UCSD/university

If you are struggling, come to office hours. Ask questions on Piazza. Reach out to me to ask for better approaches. Your classmates ARE doing this. And, you're not alone.

If you need a bit longer on something b/c you fell sick, a family thing came up, work called you in for an extra shift, etc., ask for an extension. Your classmates ARE doing this.