

Welcome to COGS 18:

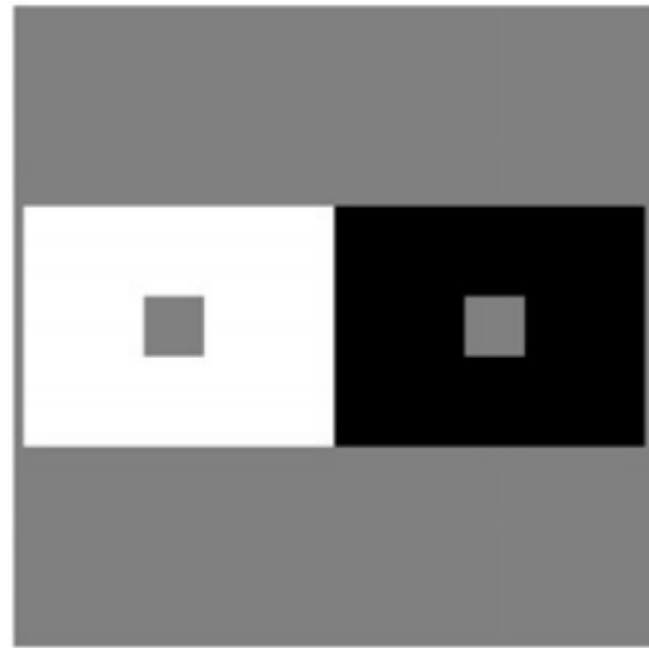
Introduction to Python

Eric Morgan

UC San Diego

Department of Cognitive Science

e1morgan@ucsd.edu





Nimit

Office hours

Th 3:30p to 4:30p

Location: TBD



Siddarth

Office hours

Tu 3:30p to 4:30p

Location: TBD

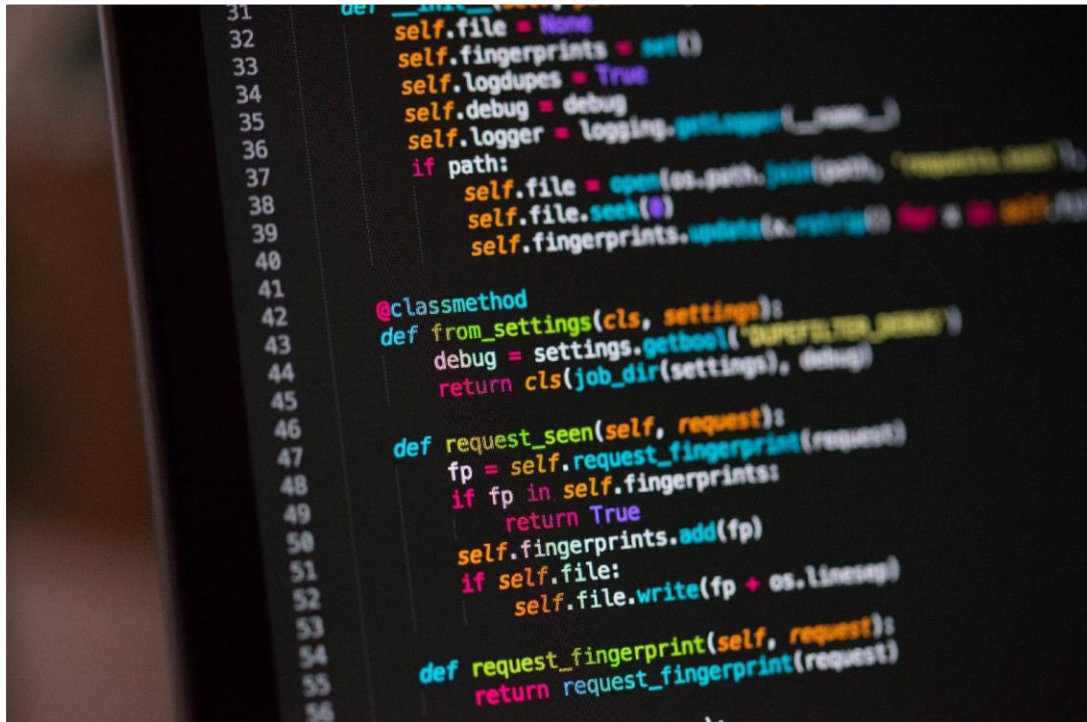
The (dreaded) waitlist

1. I do not handle the waitlist - our staff (cogsadvising@ucsd.edu) do
2. I do not have access to the waitlist nor the system that enrolls students from the waitlist.
3. Typically ~3-5 students from each section are enrolled by our staff.
4. Enrollment cannot exceed no. of seats in the classroom.
5. The waitlist clears at the end of week 2.

If you email me about the waitlist or your specific circumstance/need to take this course this quarter, I will point you to cogsadvising@ucsd.edu.

Intro Programming courses are often **thought of as difficult** and are courses with the **highest dropout rates**

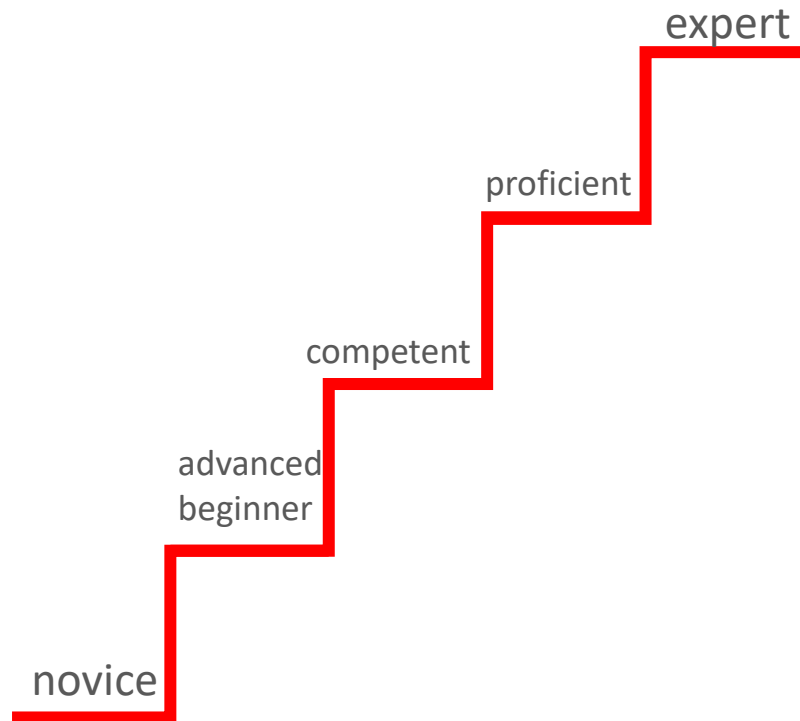
....yet, the only thing that is slightly predictive of success in an intro programming course is...**how successful the student thinks they will be**



```
31 def __init__(self):
32     self.file = None
33     self.fingerprints = set()
34     self.logdupes = True
35     self.debug = debug
36     self.logger = logging.getLogger(__name__)
37     if path:
38         self.file = open(os.path.join(path, "requests.log"),
39                         "a")
40         self.file.seek(0)
41         self.fingerprints.update(e.request() for e in self.requests)
42
43 @classmethod
44 def from_settings(cls, settings):
45     debug = settings.getbool("SUPERFINGER_DEBUG")
46     return cls(job_dir(settings), debug)
47
48 def request_seen(self, request):
49     fp = self.request_fingerprint(request)
50     if fp in self.fingerprints:
51         return True
52     self.fingerprints.add(fp)
53     if self.file:
54         self.file.write(fp + os.linesep)
55
56 def request_fingerprint(self, request):
57     return request_fingerprint(request)
```

Things that do NOT predict success:

- gender
- age
- personality
- math ability



My goal is to have you all be able to **program at an introductory level**

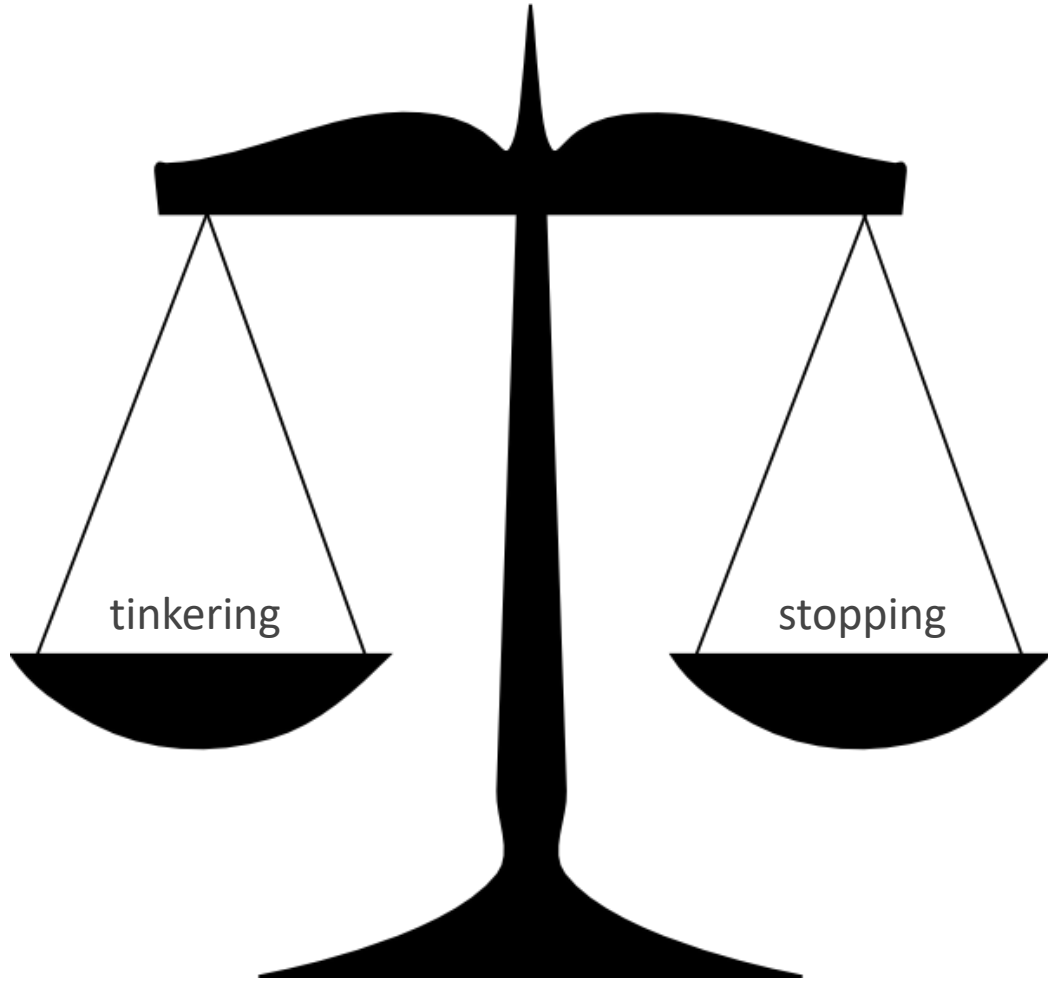
It's generally accepted that it takes people **10 years to move from novice to expert programmer**. But there are lots of steps in between! We're working to move you further away from novice (& in the direction of expert) than you are right now.



Mixed Messages: We tell people learning to program will be **tough and frustrating** but that **if you're not having fun, you're doing it wrong.**



Building Blocks: Too often, we also tell people to “just try things out” without explaining basic concepts. Other courses aren’t taught this way...



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

If you're not moving forward, consider the **2-hour rule**.

If you're trying to figure something out and struggling to move forward at all, consider the 2-hour rule: If you're stuck, **work on the problem for an hour**. If you're still stuck, walk away & **take a 30 min break**. Then, **try again for another 30 minutes** or so. If you're **still completely stuck, stop and contact us** (come to office hours, post on Piazza). If you're not even sure what your question is, include what information that you do have - what you're stuck on, what you've tried, error messages you've received, etc.

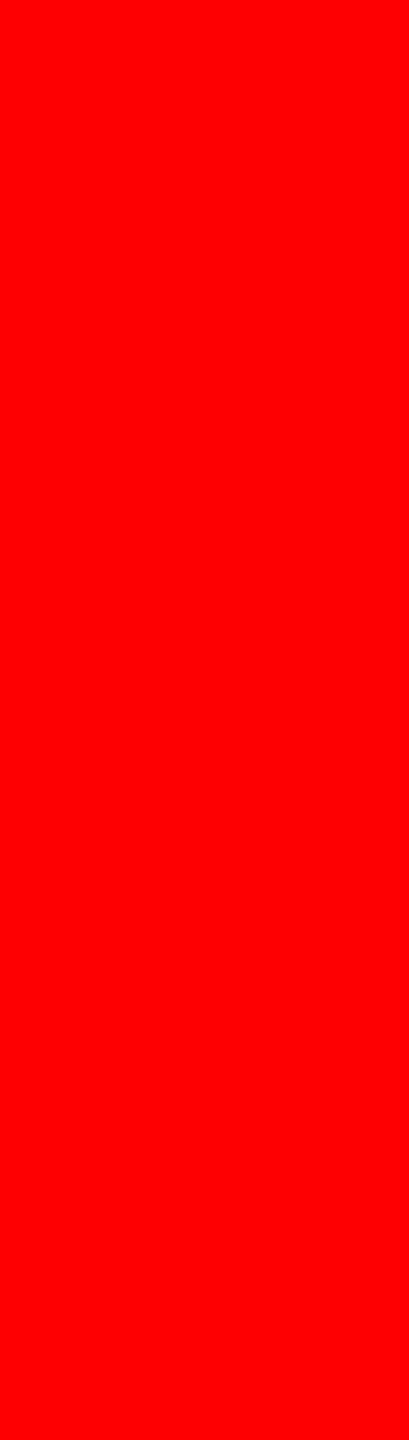
Why Python?

- simple(r) syntax
- widely-used
- Jupyter Notebooks

“It’s not the best language for anything, but it’s the second best for everything”

-Brad Voytek





COGS 18: How
this course is
going to work

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)
 1. iClicker questions for comprehension
 2. Time to apply what was just explained
3. **Coding Labs** (low stakes)
 1. Notebooks provided
 2. Staff/classmates there to help
 3. Checked for effort, not correctness
4. **Assignments** (mid stakes)
 1. Completed individually (can work together)
 2. Programmatically graded
5. **Exams** (high stakes)
 1. Two parts: conceptual (in-class) + technical (take-home)
 2. Completed totally individually

COGS 18: How You'll Be Evaluated

	% of Grade	Requirement
Coding Labs	15%	Participate In 8 Coding Labs
Assignments	35%	Complete 5 assignments
Midterm	25%	1 Midterm
Final	25%	Complete Final Exam

CodingLabs:
apply concepts
discussed in
lecture using
coding labs
(15%). Practice
makes progress.

Attempt for full credit (~2% each)

- Have to make a concerted effort to complete labs
- Coding Labs will be submitted on datahub
- Answers will be sent out the following week
- Encouraged to work with others

(5) Assignments
(40%) : Jupyter
notebooks that are
completed
individually &
graded
programmatically.

Assignments always be due @ 11:59 PM.

Assignment	Due Date	Median Time Spent (hours)
A1		2
A2		4
A3		4
A4		5
A5		5

Assignment Submission @ Datahub: <https://datahub.ucsd.edu>

DATA SCIENCE / MACHINE LEARNING PLATFORM

UC San Diego

Information Technology Services - Educational Technology Services

Help Options ▾



Log In

Registered Users
"username@ucsd.edu"

UC San Diego Jupyterhub (Data Science) Platform

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.

Midterm (25%):
will be completed
individually.

Two parts:

- In-person: conceptual
- Take-home: technical
(open-book/open
Google/open ChatGPT)

Each part will be **completed on
your own**. These will include a
combination of types of questions.

Final Exam (25%):
will be completed
individually and
Submitted electronically
on the day of the final.

Must be completed on your
own. You do not have to show
up anywhere on the day of
the actual final.

All exam and due dates
are listed on the schedule
in the course syllabus and
are in Canvas

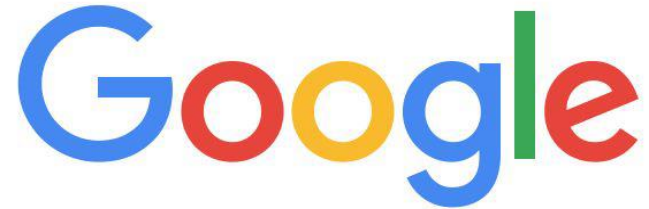


Your point of contact for COGS
18 will be the course website:

<https://cogs18.github.io>

Where to turn for help
and practice when
learning to program?

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



objects in python|




objects in python
objects in python **3**
objects in python **2**
objects in python **tutorial**
objects in python **code**
objects in python **lists**
objects in python **django**
objects in python **inheritance**
objects in python **return**
objects in python **for loop**


Google Search

I'm Feeling Lucky

Report inappropriate predictions


StackOverflow probably has the answer to your question







21

3




+20



Home

PUBLIC

 Stack Overflow


Tags

Users

Jobs

Teams

Q&A for work



Learn More

Tags

A tag is a keyword or label that categorizes your question with other, similar questions. Using the right tags makes it easier for others to find and answer your question.

Popular

Name

New

<div>python × 1137913</div> <div>a multi-paradigm, dynamically typed, multipurpose programming language, designed to be quick (to learn, to use, and to</div> <div>1085 asked today, 6241 this week</div>	<div>python-3.x × 151128</div> <div>For questions about Python programming that are specific to version 3+ of the language. Use the more generic [python] tag</div> <div>273 asked today, 1641 this week</div>	<div>python-2.7 × 89413</div> <div>the last major version in the 2.x series. Do not use this tag simply to convey the version of Python you're using, unless the question</div> <div>40 asked today, 219 this week</div>	<div>python-requests × 9229</div> <div>a full-featured Python HTTP library with an easy-to-use, logical API.</div> <div>8 asked today, 81 this week</div>
<div>wxpython × 6191</div> <div>a Python wrapper for the cross-platform C++ GUI API wxWidgets.</div> <div>20 asked this week, 52 this month</div>	<div>ipython × 6036</div> <div>a feature-rich interactive shell for Python, and provides a kernel for frontends such as IPython Notebook and Jupyter Notebook.</div> <div>15 asked this week, 65 this month</div>	<div>python-imaging-library × 4495</div> <div>The Python Imaging Library (PIL) provides the Python language with a de-facto standard foundation for image work. PIL's</div> <div>23 asked this week, 113 this month</div>	<div>python-3.6 × 3882</div> <div>Version of the Python programming language released in December 2016. For issues specific to Python 3.6. Use more</div> <div>10 asked today, 43 this week</div>
<div>python-3.5 × 3260</div> <div>The version of the Python programming language released on September 13, 2015. For issues that are specific to Python 3.5.</div> <div>9 asked this week, 34 this month</div>	<div>python-import × 3150</div> <div>For questions about importing modules in Python</div> <div>18 asked this week, 58 this month</div>	<div>python-3.4 × 2594</div> <div>The version of the Python programming language released on March 16, 2014. For issues that are specific to Python 3.4. Use</div> <div>6 asked this month, 126 this year</div>	<div>python-sphinx × 2365</div> <div>a tool that makes it easy to create intelligent and beautiful documentation. Sphinx is especially suitable for Python</div> <div>9 asked this week, 34 this month</div>

ChatGPT



Examples

"Explain quantum computing in simple terms" →

"Got any creative ideas for a 10 year old's birthday?" →

"How do I make an HTTP request in Javascript?" →



Capabilities

Remembers what user said earlier in the conversation

Allows user to provide follow-up corrections

Trained to decline inappropriate requests



Limitations

May occasionally generate incorrect information

May occasionally produce harmful instructions or biased content

Limited knowledge of world and events after 2021

ChatGPT

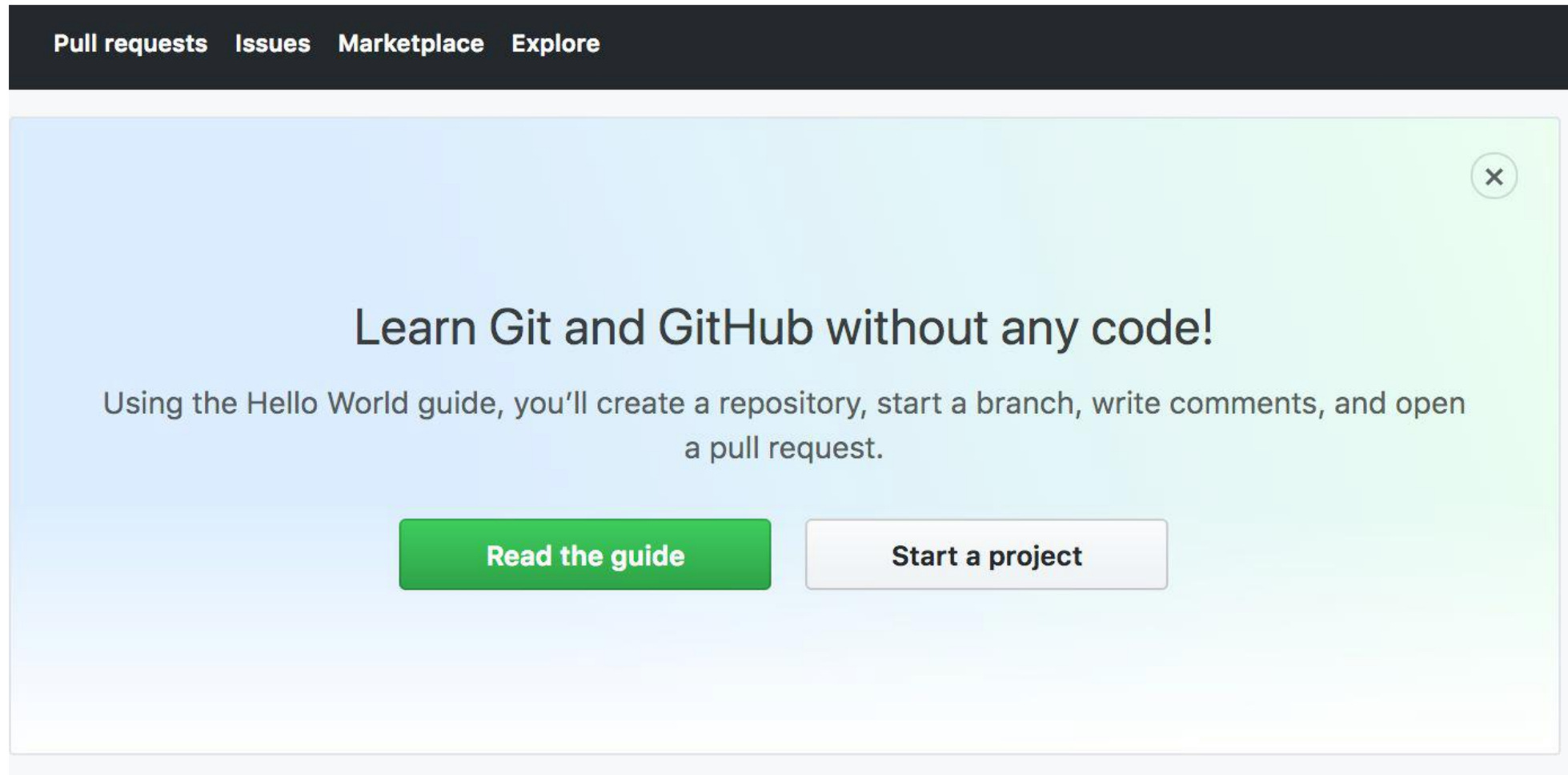
A conversational LLM that will produce prose and code. It has **pros** (you can get your questions answered!) and **cons** (it can be confidently wrong).

Send a message...



GitHub: programmers' social media platform

Code is shared on GitHub. In the beginning, it may be intimidating, but I encourage you to familiarize yourself with the platform and share code you write on GitHub.



The screenshot shows the GitHub homepage. At the top is a dark navigation bar with links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below this is a large, light-colored banner with a blue-to-green gradient. In the top right corner of the banner is a circular close button with an 'x' icon. The banner contains the following text and buttons:

Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

[Read the guide](#) [Start a project](#)

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!

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.



Today I used a slideshow, but every other day of class, **lecture notes** will be presented in a **Jupyter notebook**