

Welcome to COGS 18: Introduction to Python

COGS 18

Shannon E. Ellis, PhD

UC San Diego

Department of Cognitive Science

✉ sellis@ucsd.edu

hello
my name is

Professor Ellis



Teaching Assistants (TA)



Shreenivas



Devendra

IAs

Chau
Duolan
Byungkwon
Andrew
Titan
Zijan
Sarah
Severine*
Stephen*

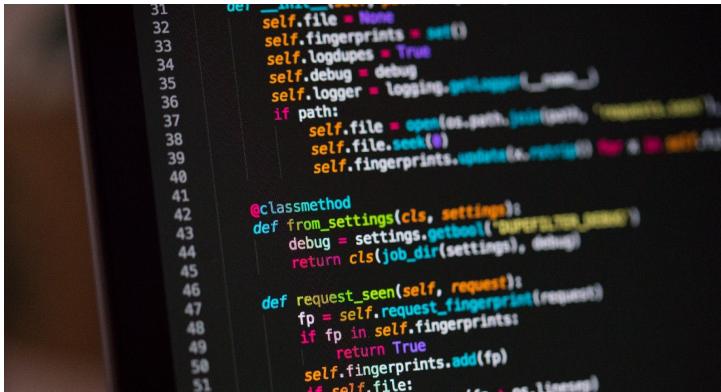
*Unofficial IAs. They'll hold office hours and you may see them in CodingLab, but they won't be answering on Piazza.



Let's chat: Teaching &
Learning Programming

o o o

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

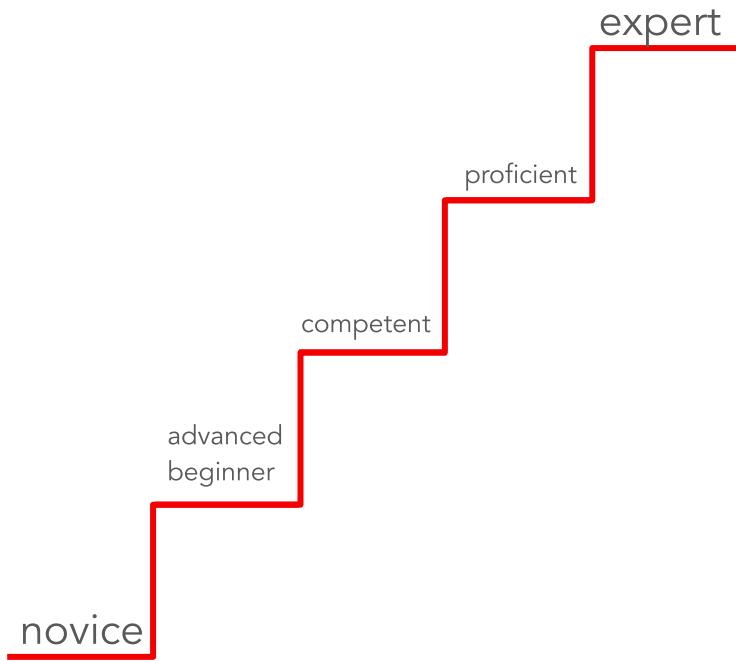


```
31     self._init()
32     self.file = None
33     self.fingerprints = set()
34     self.logdups = True
35     self.debug = debug
36     self.logger = logging.getLogger(__name__)
37     if path:
38         self.file = open(os.path.join(path, "fingerprint"))
39         self.file.seek(0)
40         self.fingerprints.update([line.strip() for line in self.file])
41
42     @classmethod
43     def from_settings(cls, settings):
44         debug = settings.getbool("DEBUG", False)
45         return cls(job_dir(settings), debug)
46
47     def request_seen(self, request):
48         fp = self.request_fingerprint(request)
49         if fp in self.fingerprints:
50             return True
51         self.fingerprints.add(fp)
52         # self.file: ... os.linesep)
```

....yet, 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



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

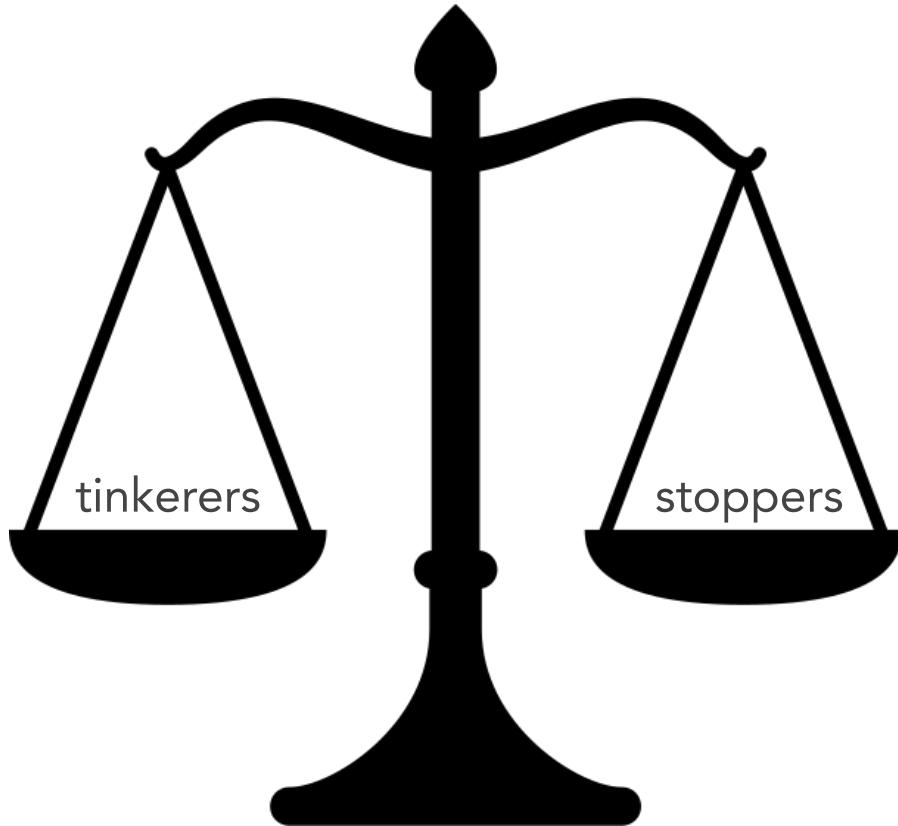
It's generally accepted that it takes people **10 years to move from novice to expert programmer**. But, there are lots of steps in between! We're working to move you further away from novice (& in the direction of expert) than you are right now.



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?

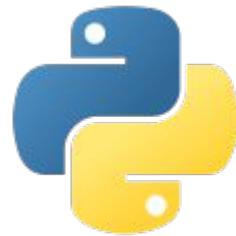
simple(r) syntax

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

o o o

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

1. First half of course: **basic concepts**
2. **In-class practice**
 - a. iclicker questions for comprehension
 - b. time to apply what was just explained
3. **Coding Labs (discussion section)**
 - a. Notebooks provided in section
 - b. TAs/IAs/classmates there to help
4. **Assignments**
 - a. Completed individually
 - b. Programmatically graded

COGS 18: How You'll Be Evaluated

	% of Grade	Requirement
Coding Labs	15%	Attend & Participate In 6 Code Labs
Assignments	30%	Complete 4 assignments
Exams	30%	Two, written in-class exams
Final Project	15%	Submit final project
Lecture Attendance	10%*	Attend 75% of all lectures

*if you fail to attend 75% of all lectures, each exam will automatically be an extra 5% toward your final grade

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

Week 1-9: must **attend 6** for full credit

- Have to attempt section **coding labs**
- Answers will be sent out the following week
- Can work with others

Week 10: get help & code review projects (attendance
not required, but encouraged)

You **should attend the section to which you're assigned.** You
can attend a different section. However, if one section
becomes too crowded each week, we'll revisit this policy.

(4) Assignments (30%) : Jupyter notebooks that are completed individually & graded programmatically.

Assignments will always be **due on Friday by 11:59 PM**. Due dates are in the first seven weeks of the course. The second half of the course becomes project-based.

Assignment	Due Date (11:59 PM)	Week
A1	10/11	Week 2
A2	10/18	Week 3
A3	11/1	Week 5
A4	11/15	Week 7

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

DATA SCIENCE / MACHINE LEARNING PLATFORM

UC San Diego

Information Technology Services - Educational Technology Services

Help Options ▾



UC San Diego Jupyterhub (Data Science) Platform

o o o

In technical
classes, Piazza is
a particularly
helpful resource

There are rules:

1. No duplicates.
2. Include Assignment & Question in Summary line.
3. Public posts are best.
4. Helping one another is encouraged.
5. No assignment code in public posts.
6. We're not robots.

University of California, San Diego

(change school)

Are you a professor?
Click here to create & join classes

Welcome to Piazza!

Piazza is a free platform for instructors to efficiently manage class Q&A. Students can post questions and collaborate to edit responses to these questions. Instructors can also answer questions, endorse student answers, and edit or delete any posted content.

Piazza is designed to simulate real class discussion. It aims to get high quality answers to difficult questions, fast!

The name Piazza comes from the Italian word for plaza—a common city square where people can come together to share knowledge and ideas. We strive to recreate that communal atmosphere among students and instructors.

Selected Term: Fall 2019

Fall 2019

Class 1: COGS 18: COGS 18: Introduction to Python (edit)

Instructors: Shannon Ellis · 7 Enrolled

→ **Join as:** Student *Instructor self-enrollment has been disabled for this class.*

Class 2: X

Class 3: X

Class 4: X

Class 5: X

Add Another Class

Join Classes

<https://piazza.com/ucsd/fall2019/cogs18>



(1) Exams (30%): Exams are closed-book and taken on paper. Each will include a **combination of types of questions** (multiple choice, fill in the blank, short answer, matching, etc.). All you need to bring on exam days is a writing utensil.

will be closed book
and completed
in-class on 10/21
and 11/13.

(1) Final Project (15%):
will be completed
individually and
submitted
electronically on the
day of the final (12/11)
by 11:59 PM.

Your final project can build on an assignment from the course to provide it with additional functionality or it can be on a completely new topic of your choosing. However, it **must include original code that you've written** for this project. You do not have to show up anywhere on the day of the actual final.



Your point of contact for COGS18
will be the course website:
<https://cogs18.github.io>

Any questions about
course logistics?

• • •



Why even learn to
program in **Python**?

o o o



The University of St Andrews, founded in 1411, is Scotland's oldest university ([Source](#))



Introduction

The IT Services department at the University of St Andrews, Scotland, develops and maintains software systems used in a variety of capacities throughout the university.

I had several years of experience working with Perl when I took my first serious look at Python back in 1999. Our team's projects were becoming bigger and more complex, and it was obvious that we needed to bring to them more structure and clarity. I had been looking at Java for some time, but its potential benefits seemed to come at the cost of a steep learning curve, and an overall increase in development time. In contrast, Python appeared to offer the prospect of having both clarity *and* productivity at the same time. And if we ever needed to make use of Java's class libraries there was always Jython, an implementation of Python for the JVM. The increasing number of Python books being published testified to the language's growing popularity, and the number of available libraries was beginning to rival Perl's. This convinced me to give Python a try.

<https://www.python.org/about/success/st-andrews/>



In 1999, Python becomes mainstay of IT systems

Use Case: Matching students with a class

- Select preferences each semester
- Validate who is in which course
- Handles concurrent users well

Why Python:

- Reduced amount of programming needed
- Quickly learned by staff





VISUAL EFFECTS

ART DEPARTMENT

SERVICES

ILMxLAB

COMPANY

BLOG



<http://www.ilm.com/>



Background

Industrial Light & Magic (ILM) was started in 1975 by filmmaker George Lucas, in order to create the special effects for the original Star Wars film. Since then, ILM has grown into a visual effects powerhouse that has contributed not just to the entire Star Wars series, but also to films as diverse as Forrest Gump, Jurassic Park, Who Framed Roger Rabbit, Raiders of the Lost Ark, and Terminator 2. ILM has won numerous Academy Awards for Best Visual Effects, not to mention a string of Clio awards for its work on television advertisements.

In 1996, Python unifies ILM

- Used Python to code (and re-code) the programs that controlled the production pipeline
 - Ease-of use
 - Quickly learnable
 - Replace Unix shell scripting
- What Python accomplished
 - Streamlined production
 - Provided the needed flexibility
 - Reduced hardware costs
 - Stay on top of the competition
- Since 1996, no better system found

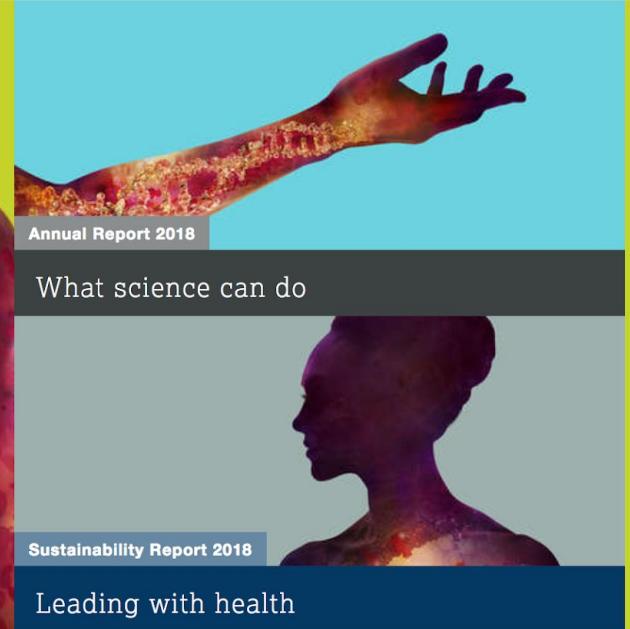


What science can do ▾ Our science ▾ Our focus areas ▾ Our company ▾ Careers ▾ Investors ▾ Partnering ▾ Media ▾ Sustainability ▾



Entering a new era in vascular and cardiac regeneration research

[Read more](#) 



We believe in what science can do

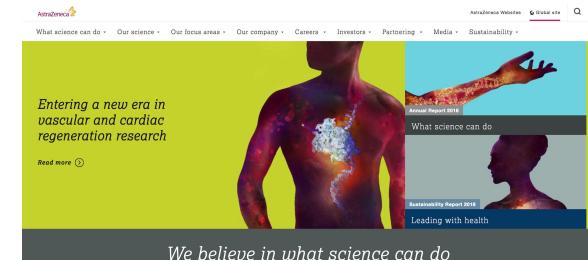
Introduction

AstraZeneca is one of the world's leading pharmaceutical companies. With over 54,000 employees worldwide, it provides innovative, effective medicines designed to fight cancer, provide pain control, heal infection, and fight diseases of the cardiovascular, central nervous, gastrointestinal, and respiratory systems.

To save time and money on laboratory work, experimental chemists use computational models to narrow the field of good drug candidates, while also verifying that the candidates to be tested are not simple variations of each other's basic chemical structure.

At AstraZeneca, Python enabled collaboration

- Drug Discovery is key
- Computational chemists weren't the only ones who could do analysis
 - web-based interface called H2X, initially written in Perl
 - Experimentalists could make predictions on their own
 - In 2001, further developed in Python: PyDrone
 - 3 months of development time + 3 months of QA + 3 weeks of documentation time to produce about 5,600 lines of finished Python code.
 - Chosen for its ease-of-use & interpretability
 - Can solve real-world problems
 - Error handling made product more robust



AIR TRAFFIC MANAGEMENT

Keeping 3 billion passengers safe each year



VOICE COMMUNICATIONS

NETWORKS

REMOTE VIRTUAL TOWER

ATC TOWER

SURVEILLANCE

AIM

AMHS

ATM/UTM INTEGRATION

Keeping 3 billion passengers safe each year

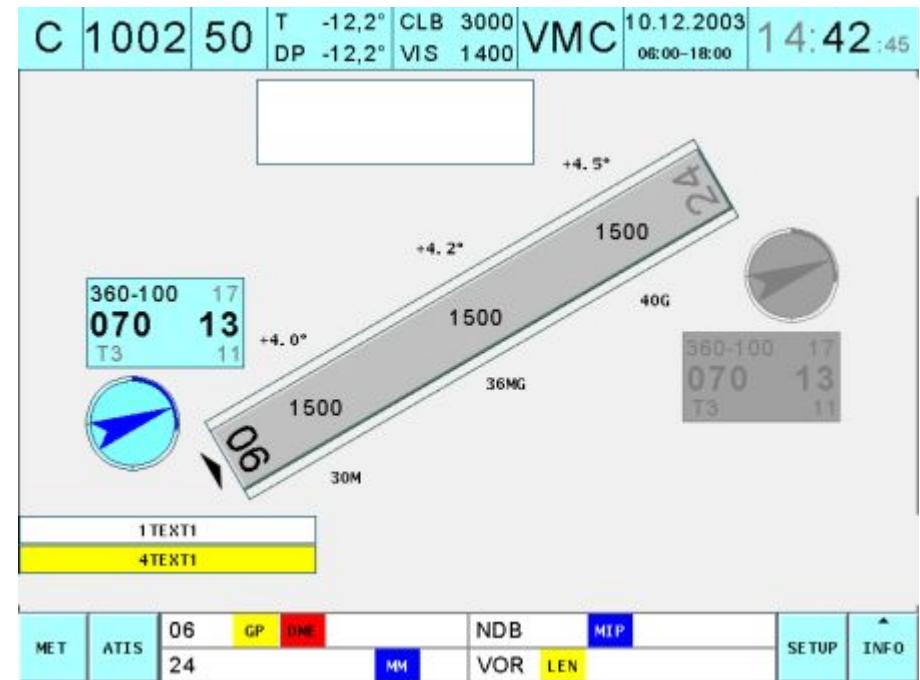
Air Traffic Management (ATM) solutions leverage more than seventy years of ATC experience showcasing true leadership in this market, with deep cross-industry experience to provide functionality in support of the needs of today's Air Navigation Service Providers (ANSP). Embracing digitalisation, virtualisation and innovation, these solutions are an essential enabler to lead today's Air Navigation Service Providers through future evolutions of their infrastructure.

Frequentis is one of the world's leading providers for safety-critical solutions in the field of Air Traffic Management and Public Safety & Transport. With over 500 employees world-wide, it provides innovative, user-centered solutions to its customers.

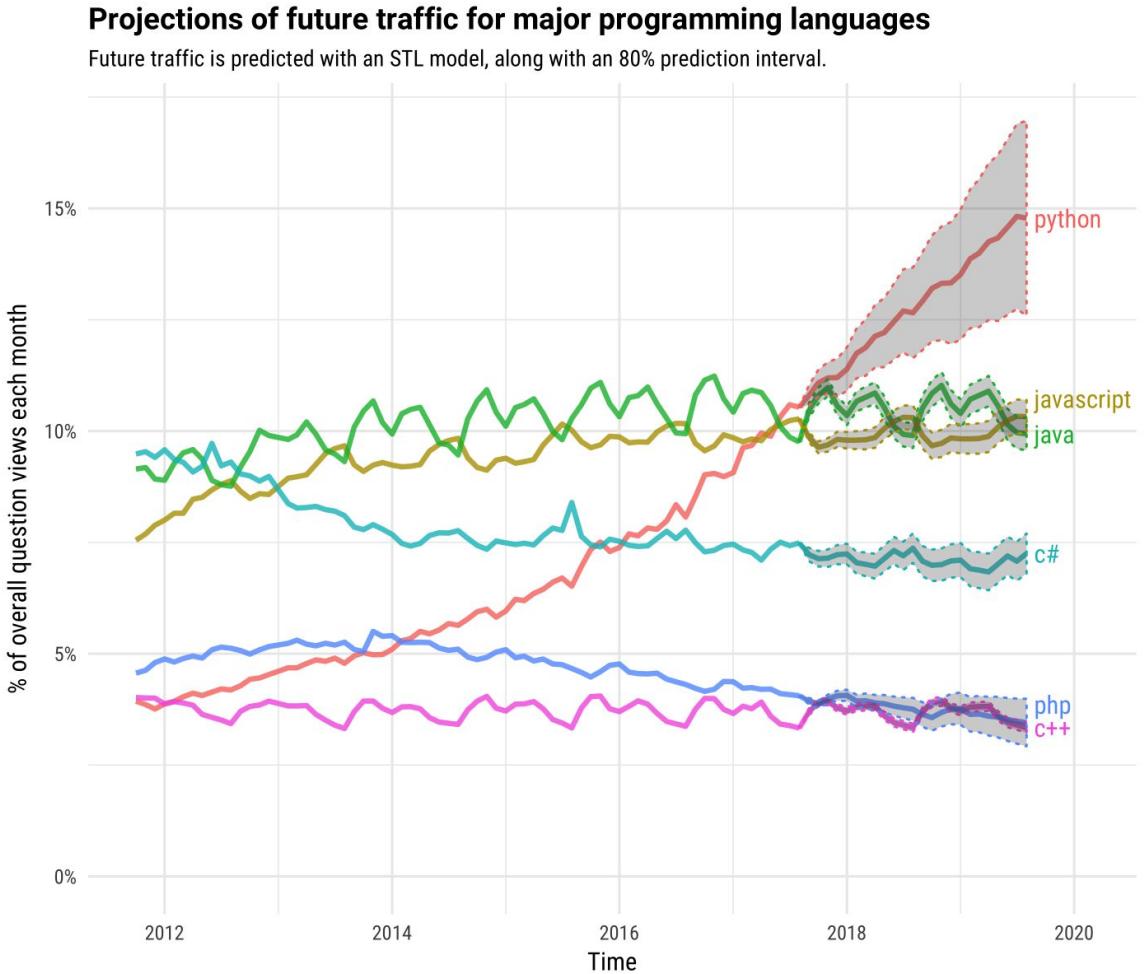
Frequentis has been using Python in its TAPtools® product family, which focuses on the *Tower and Airport Tools* segment of Air Traffic Control. These tools are used by air traffic controllers to track weather conditions, control runway lighting, and to monitor and control navigational aid instruments.

Python enables tool to work in front of the customer

- Each airport is unique
- TAPtools enables interface unique to airport to be built
 - Rapidly developed in front of the user
 - Minimizes back and forth
- Python advantages
 - Easy to read and maintain
 - Flexible
 - Faster to write (3X faster)



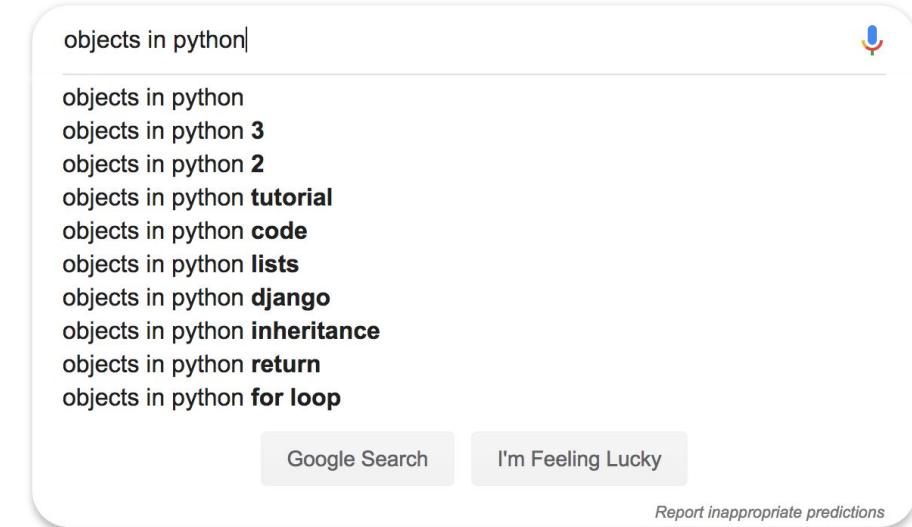
Python is only
gaining in
popularity as a
programming
language



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

o o o

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



A screenshot of a Google search interface showing suggestions for the query "objects in python". The suggestions are:

- 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**

Below the suggestions are two buttons: "Google Search" and "I'm Feeling Lucky". At the bottom, there is a link to "Report inappropriate predictions".

StackOverflow probably has the answer to your question

stackoverflow Search... 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.

python

Popular Name New

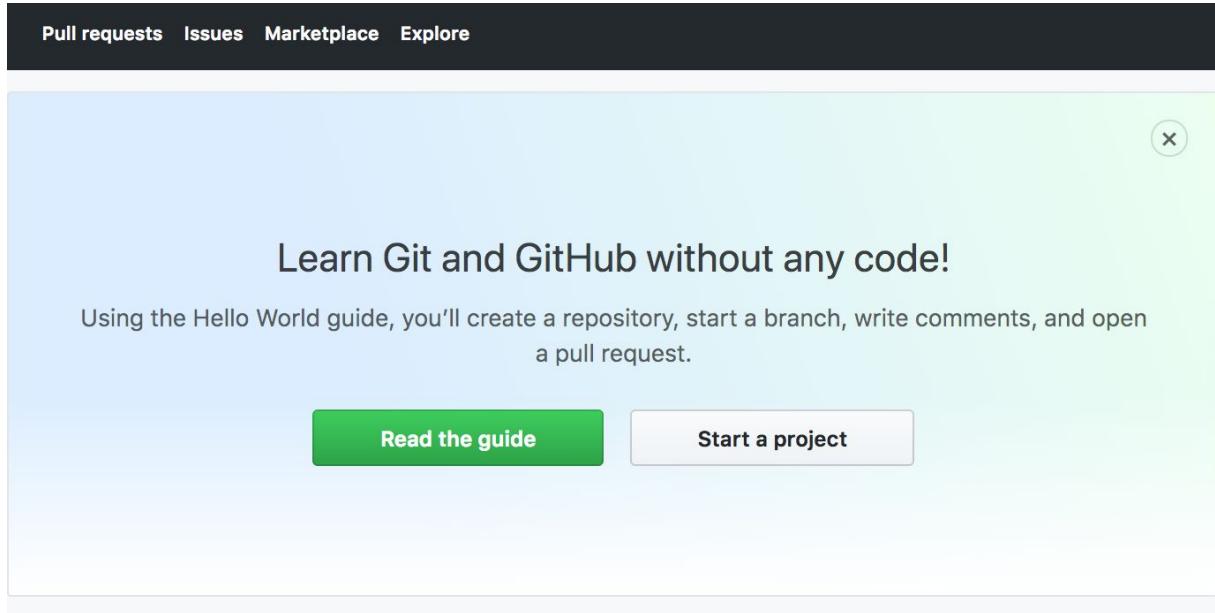
python × 1137913 a multi-paradigm, dynamically typed, multipurpose programming language, designed to be quick (to learn, to use, and to 1085 asked today, 6241 this week)	python-3.x × 151128 For questions about Python programming that are specific to version 3+ of the language. Use the more generic [python] tag 273 asked today, 1641 this week	python-2.7 × 89413 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 40 asked today, 219 this week	python-requests × 9229 a full-featured Python HTTP library with an easy-to-use, logical API. 8 asked today, 81 this week
wxpython × 6191 a Python wrapper for the cross-platform C++ GUI API wxWidgets. 20 asked this week, 52 this month	ipython × 6036 a feature-rich interactive shell for Python, and provides a kernel for frontends such as IPython Notebook and Jupyter Notebook. 15 asked this week, 65 this month	python-imaging-library × 4495 The Python Imaging Library (PIL) provides the Python language with a de-facto standard foundation for image work. PIL's 23 asked this week, 113 this month	python-3.6 × 3882 Version of the Python programming language released in December 2016. For issues specific to Python 3.6. Use more 10 asked today, 43 this week
python-3.5 × 3260 The version of the Python programming language released on September 13, 2015. For issues that are specific to Python 3.5. 9 asked this week, 34 this month	python-import × 3150 For questions about importing modules in Python 18 asked this week, 58 this month	python-3.4 × 2594 The version of the Python programming language released on March 16, 2014. For issues that are specific to Python 3.4. Use 6 asked this month, 126 this year	python-sphinx × 2365 a tool that makes it easy to create intelligent and beautiful documentation. Sphinx is especially suitable for Python 9 asked this week, 34 this month

<https://stackoverflow.com>



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 with a dark header bar containing 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the header is a light blue main area. A white modal window is centered, featuring a close button ('X') in the top right corner. The modal's title is 'Learn Git and GitHub without any code!'. The text inside the modal reads: 'Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.' At the bottom of the modal are two buttons: a green 'Read the guide' button and a white 'Start a project' button.

<https://github.com/>

o o o

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!](#)

	Time	Location
Instructor Office Hours	Wed 4-6 PM	CSB 243
TA Office Hours	Th 1-2 PM (Devendra)	TBD
	TBD (Shreenivas)	TBD



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

ooo



I'm excited to have you all in COGS18 this quarter & I'd love to learn more about you:

http://bit.ly/cogs18_survey_fa19