

Welcome to **BILD62**

Dr. Juavinett
jah-vah-nett
(or, Dr. J)



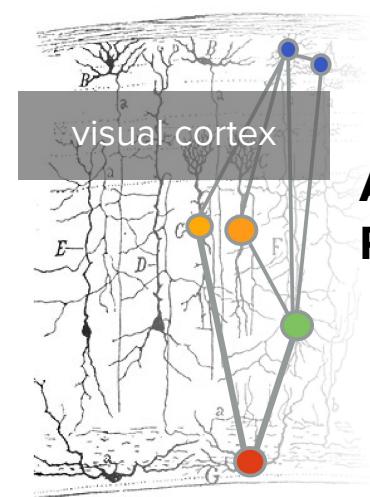
Image: [garetsworkshop/Shutterstock](#)

Objectives for this morning

- Introduce the teaching staff, students, and class
 - Motivate learning how to code as a biology student
 - Discuss course logistics, expectations, & tools
-

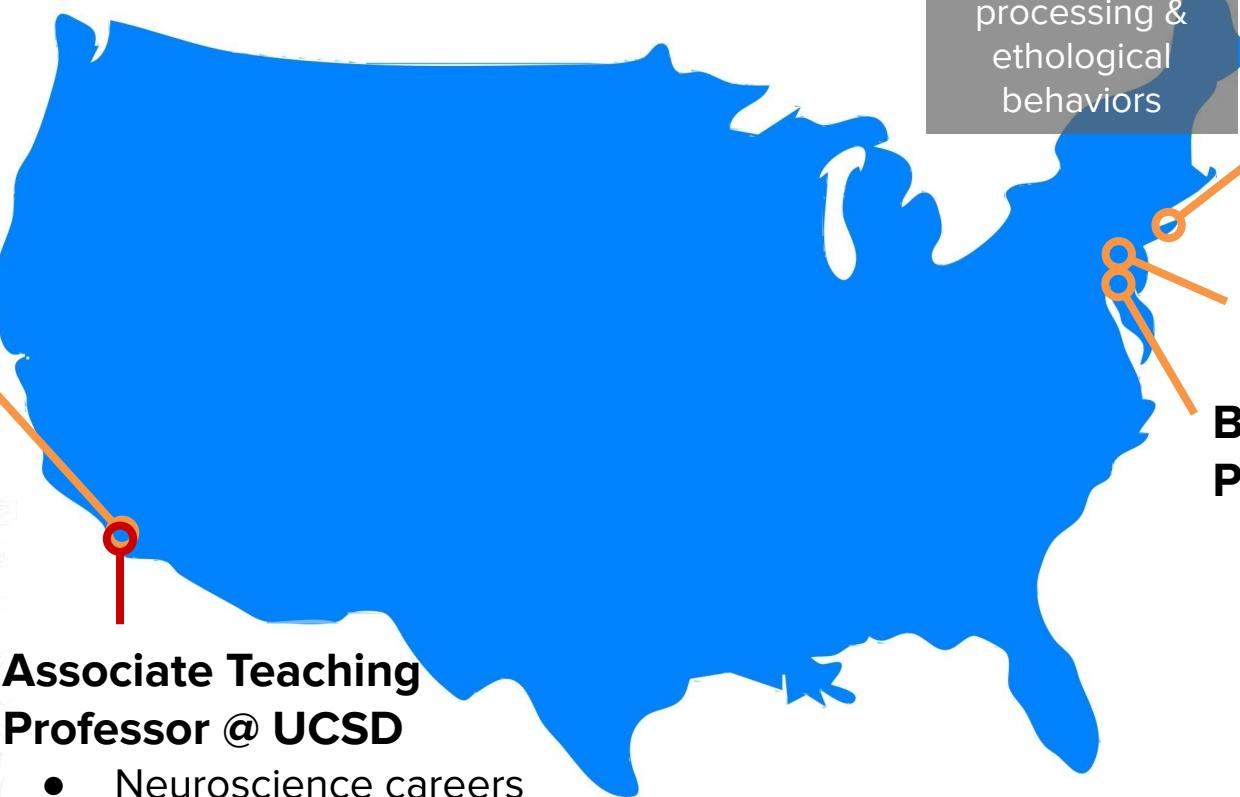


PhD in
Neuroscience
@ UCSD



Associate Teaching Professor @ UCSD

- Neuroscience careers & education
- Open-source data

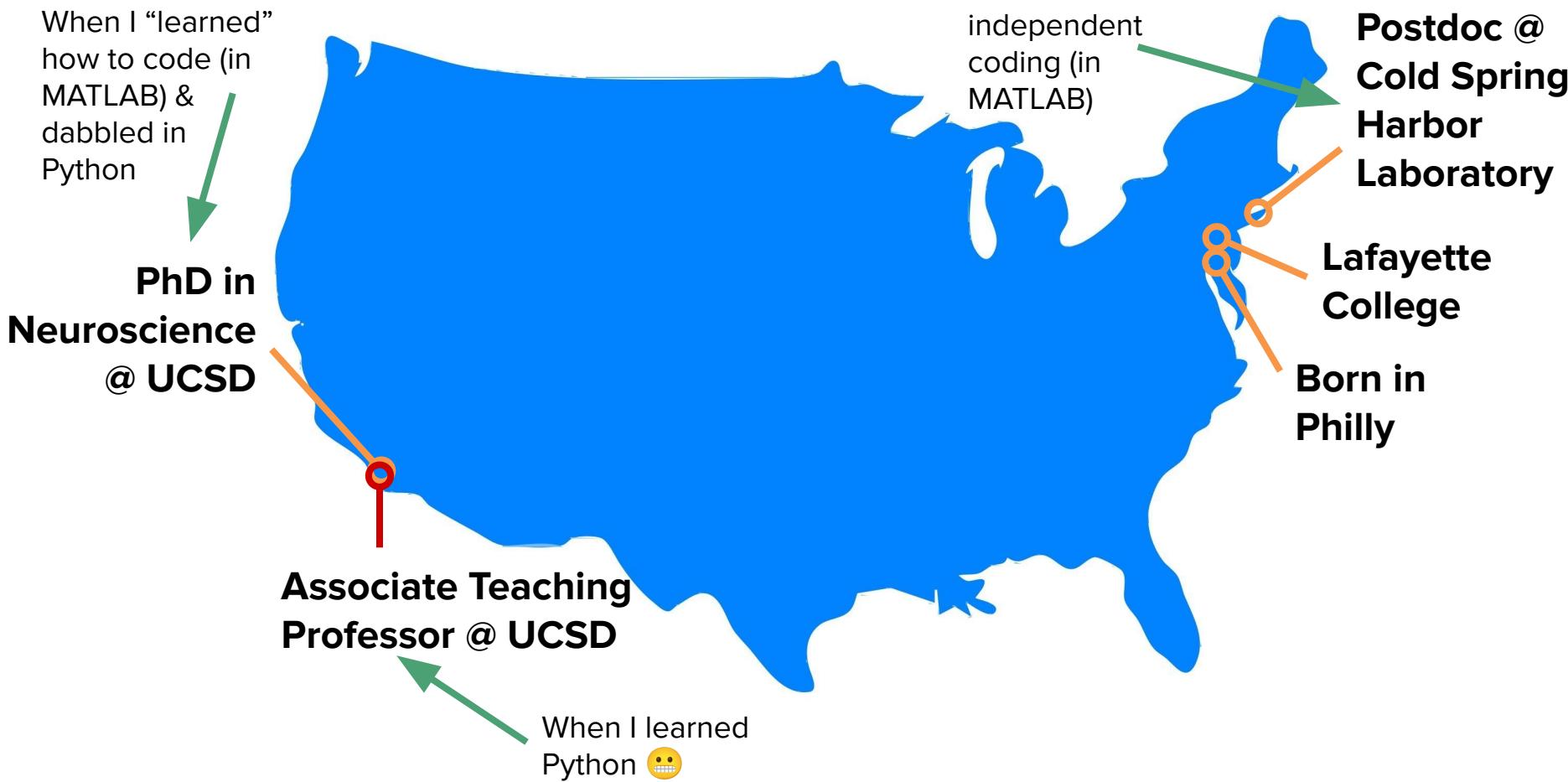


multisensory
processing &
ethological
behaviors

**Postdoc @
Cold Spring
Harbor
Laboratory**

Lafayette
College

Born in
Philly



Introduction to our Instructional Assistants!

Let's be human,
for just a second.

With the folks next to you,
share:

- Your name, major, and preferred pronouns
- Something that brought you joy over the break
- Why you're taking this course



Objectives for this morning

- Introduce the teaching staff, students, and class
 - **Motivate learning how to code as a biology student**
 - Discuss course logistics, expectations, & tools
-

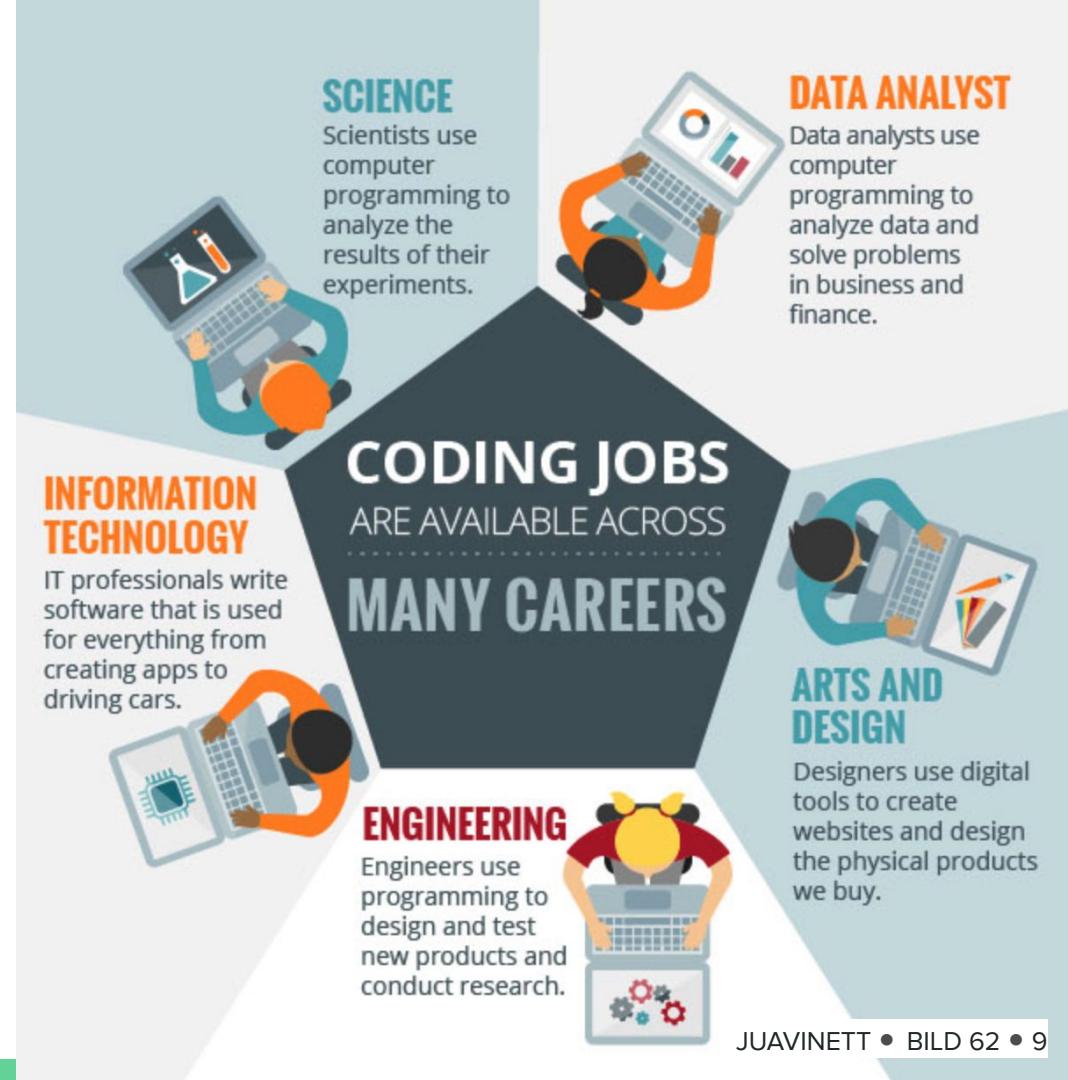
What does coding have to
do with *biology*?

Why *you*, right now?



Why should I learn how to code?

- Coding is useful for:
 - Data acquisition (controlling hardware, image acquisition, etc)
 - Data analysis & visualization
 - Computational modeling
- Beyond research, there are more and more jobs for software engineers, and they pay well
(see report by Burning Glass:
<https://www.burning-glass.com/research-project/coding-skills/>)

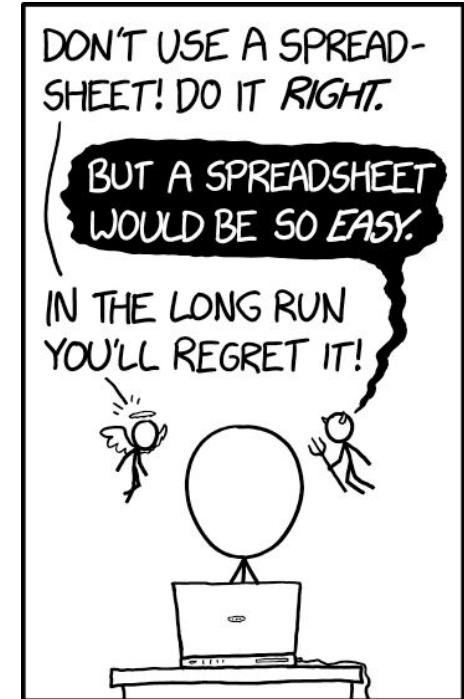


Excel can only handle datasets with **“1 million rows, and 16,000 columns** — many datasets in biology are much larger than this!

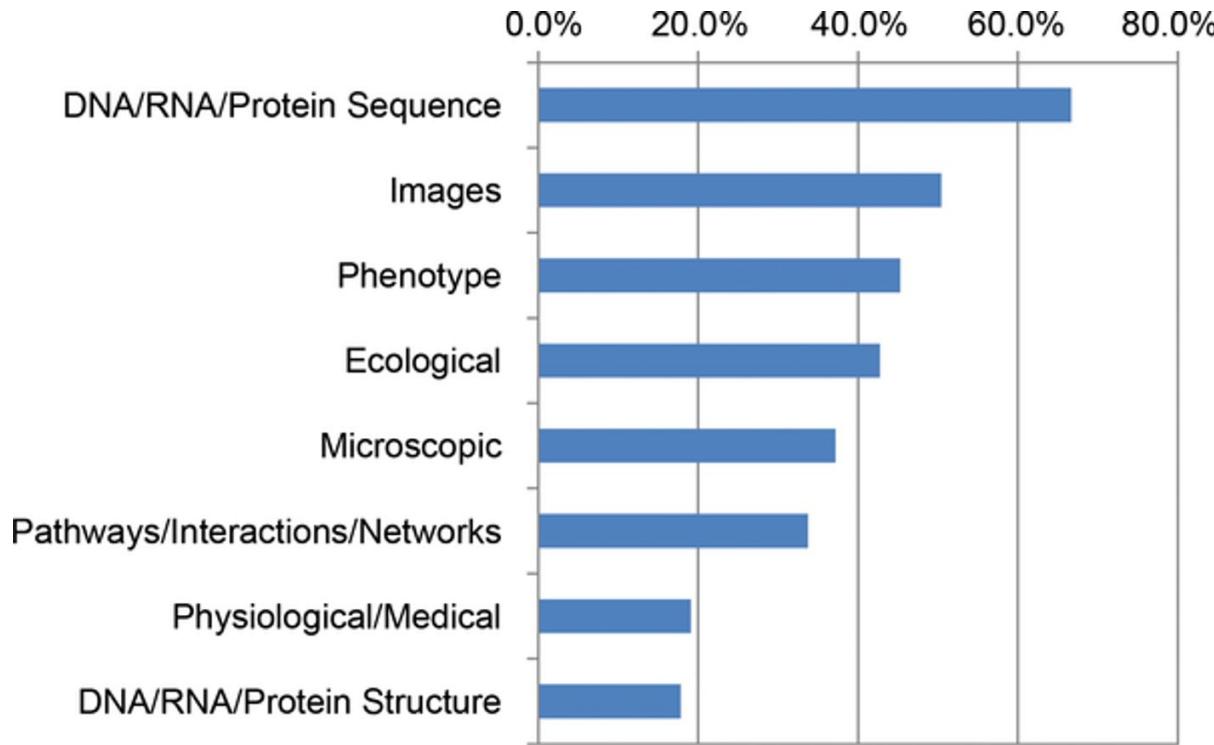
You can automate analyses in Excel, but this is quite limited.

There are also specialized biological data analysis software programs, but often these are limited in how much they can be customized.

Code is *infinitely* customizable.



<https://xkcd.com/2180/>



Major data types used by National Science Foundation (NSF)
Biological Sciences Directorate (BIO) principal investigators (PIs).

PyMOL (membrane.pse)

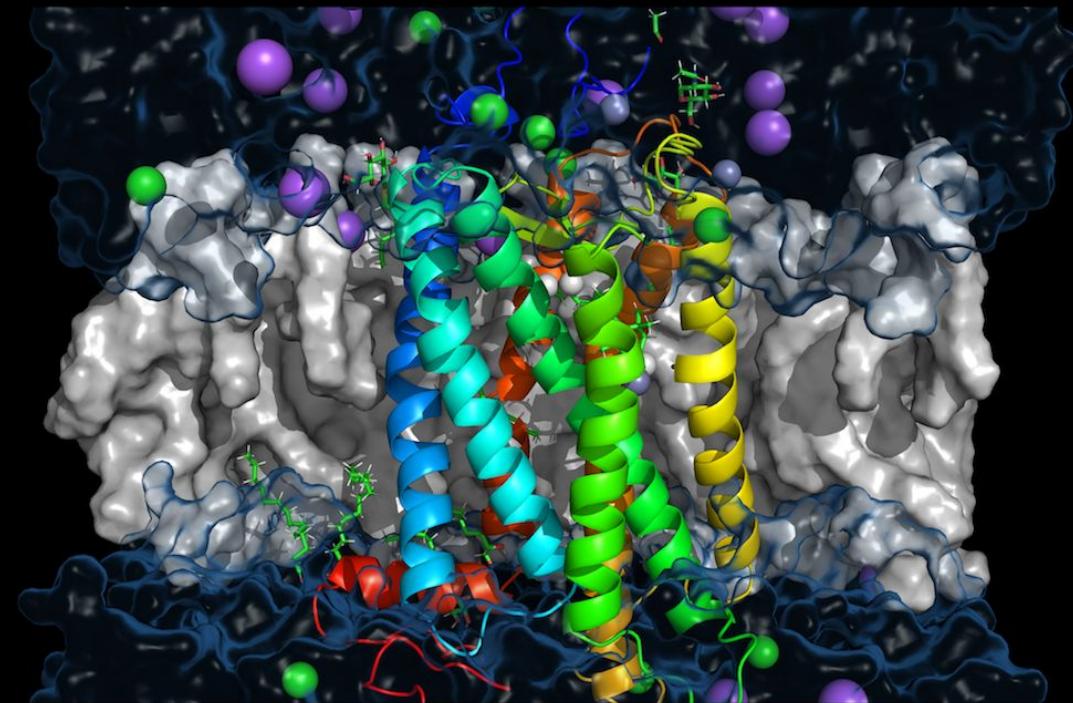
Save: Please wait -- writing session file...
Save: wrote "/Users/piotr/membrane.pse".

PyMOL-ray 2440, 1300

Ray: render time: 208.91 sec. = 17.2 frames/hour (208.91 sec. accum.).

Reset Zoom Orient Draw/Ray
Unpick Deselect Rock Get View
< < Stop Play > > McClear
Builder Properties Rebuild

PyMOL>



all	A	S	H	L	F
ions	1/1	A	S	H	L
(sel)		A	S	H	L
water	1/1	A	S	H	L
protein	1/1	A	S	H	L
membrane1	1/1	A	S	H	L
membrane2	1/1	A	S	H	L

Mouse Mode 3-Button Viewing
& Keys Rota Move MovZ Slab
Shft +Box -Box Clip MovS
Ctrl Move PkRt Pk1 MvS
CtSh Selc Orig Clip MovZ
SnglClk +/- Cent Menu
DblClk Menu - PkRt
Selecting Residues
State 1/ 1

PyMOL>-

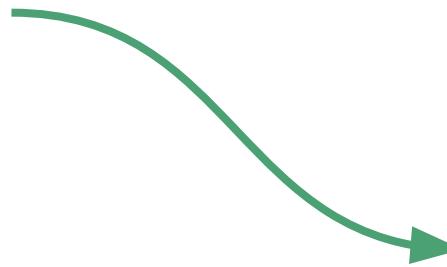
Open-Source Philosophy

PyMOL is a commercial product, but we make most of its source code freely available under a permissive license. The open source project is maintained by [Schrödinger](#) and ultimately funded by everyone who purchases a PyMOL license.

Open source enables open science.
This was the vision of the original PyMOL author Warren L. DeLano.

[Visit the Open-Source Project](#)

[Become a sponsor](#)



Schrodinger / [pymol-open-source](#) Public

Code Issues 55 Pull requests 1 Discussions Actions Security Insights

master 3 branches 4 tags Go to file Add file Code

 speleo3 iterate: Add explicit_valence and explicit_degree (#227) abc3077 19 days ago 5,076 commits

 .github/workflows Cl: Use ubuntu-18.04 16 months ago

 contrib PYMOL-3722 Fix gro file reading 7 months ago

 data PYMOL-3793: Fix for Lighting Plugin on Mac 22 days ago

 examples Fix remaining string module uses 2 years ago

 include pymol::invoke & pymol::apply last month

 layer0 Remove orthoCGO defines; fix warnings 22 days ago

 layer1 iterate: Add explicit_valence and explicit_degree (#227) 19 days ago

 layer2 iterate: Add explicit_valence and explicit_degree (#227) 19 days ago

 layer3 Fix broken group parenting 22 days ago

 layer4 Fix broken group parenting 22 days ago

About
Open-source foundation of the user-sponsored PyMOL molecular visualization system.

[pymol.org/](#)
[Readme](#) [View license](#) [634 stars](#) [32 watching](#) [166 forks](#)

Releases
4 tags

AND many software packages for biologists can be modified... if you know how to code!

By taking this class, you're ahead of the game!

Many researchers learn to code really informally, and relatively late in their careers



ashley, ahem, dr. juavinett
@analog_ashley



Neuroscientists of Twitter, when did you learn* how to code?

*Let's say, when you felt reasonably capable writing your own simple code (e.g. reading data and plotting, or communicating with an Arduino)

19% High school or earlier

30% College

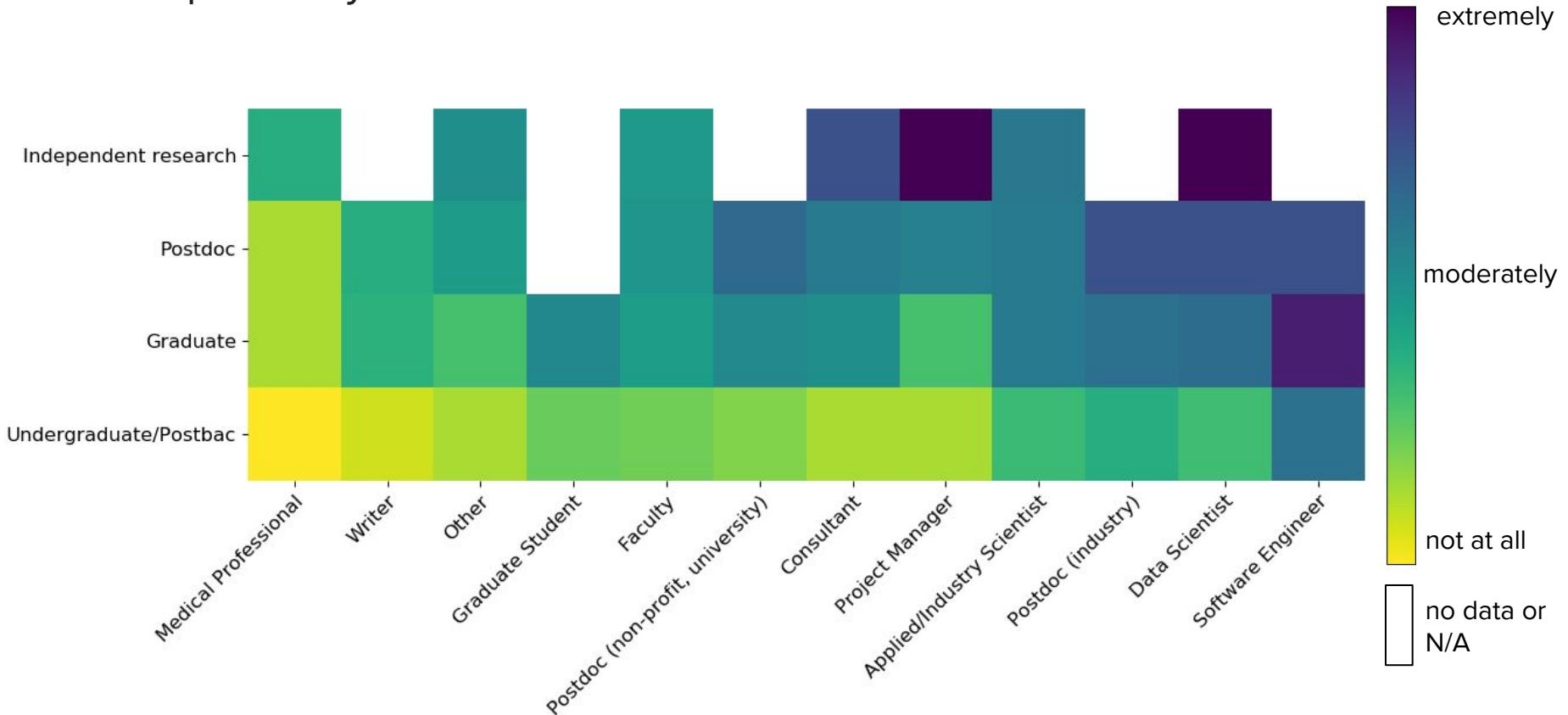
36% Graduate school

15% After graduate school

313 votes • Final results

+ many comments that they *still* hadn't learned how, and wanted to!

How comfortable did/do you feel working with code at this point in your career?



Objectives for this morning

- Introduce the teaching staff, students, and class
 - Motivate learning how to code as a biology student
 - **Discuss course logistics, expectations, & tools**
-

First step: let's drop our ideas of what it means to be a *coder*.

Programming, like learning a language, *takes time*.



Your language brain matters more for learning programming than your math brain

New research contradicts long held assumptions about coding



Amy Nippert

Neuroscience

University of Minnesota

May 12, 2020



2 peer comments



Christina Morillo on Wikimedia Commons.

<https://massivesci.com/articles/programming-math-language-python-women-in-science/>, summarizes this article: <https://www.nature.com/articles/s41598-020-60661-8>

Previous studies have shown that math and logic problems seem to rely mainly on the multiple demand regions in the left hemisphere, while tasks that involve spatial navigation activate the right hemisphere more than the left. The MIT team found that reading computer code appears to activate both the left and right sides of the multiple demand network, and ScratchJr activated the right side slightly more than the left. This finding goes against the hypothesis that math and coding rely on the same brain mechanisms.

<https://news.mit.edu/2020/brain-reading-computer-code-1215>
about this study: <https://elifesciences.org/articles/58906>

What will help you succeed in this course?

Things that predict success:

- How successful you *think* you'll be
- Completing assignments on time
- Asking questions when you have them
- Attending discussion sections & office hours

Things that **do not** predict success:

- Gender
- Age
- Personality
- Math ability



29A



@StuxnetStudios · 14h



New programming student:

"I'm not very good at this. When I type out the code, I have to fix lots of errors. And I have to look up how to do most of it."

Instructor:

"You're doing it right."

29

275

1.4K



*Historical sidenote: why is it called a **bug**?*

In 1947, computer scientist & legend **Grace Hopper** found a *literal bug* in their computer, causing it to produce many errors.



Interview with Grace Hopper:
<https://www.youtube.com/watch?v=QA33wW5LaNY>

Photo # NH 96566-KN (Color) First Computer "Bug", 1947

192

9/9

0800 anchor started { 1.2700 9.037 847 025
 1000 " stopped - anchor ✓ } 9.037 846 995 correct
 13° 00' (03) MP - MC ~~1.30476415 (-3)~~ 4.615 925 059 (-2)
 (03) PRO 2 2.130476415

Concert 2.1306764/s
Relays 6-2 m 033 failed special speed test
in relay 11.000 test.

1100 Started Cosine Tape (Sine check)
1525 Started Multi Adder Test.

1545



Relay #70 Panel F
(moth) in relay.

1700 First actual case of bug being found.
antagonist started.
1700 closed down.

[https://www.nationalgeographic.org/thisday/
sep9/worlds-first-computer-bug/](https://www.nationalgeographic.org/thisday/sep9/worlds-first-computer-bug/)

What is programming, anyway?

- Programming is the way humans communicate with **computers**
 - It's a language!



Wait, what's a computer?

Hardware: the physical parts of the computer (CPU, hard drive, etc.)

Memory:

Primary: fast, temporary storage

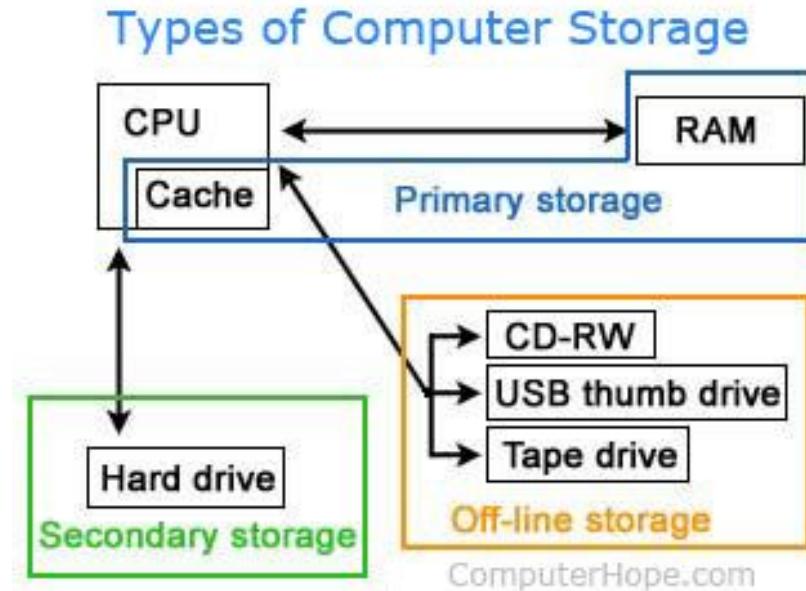
Secondary: slower, long-term storage

Tertiary: cloud storage

All computer memory is stored in **binary**.

[More information about memory](#)

[Great explainer about binary memory](#)



Wait, what's a computer?

Interface: software, the operating system,
what you see

File structure

cats = files

Each has their own name.

boxes = folders/directories

Where you store the cats.

You can put a box into another box,
but you can't put a box into a cat.



What is programming, anyway?

- Programming is the way humans communicate with computers
 - It's a language!
- The instructions we give the computer are taken **literally** and **sequentially**

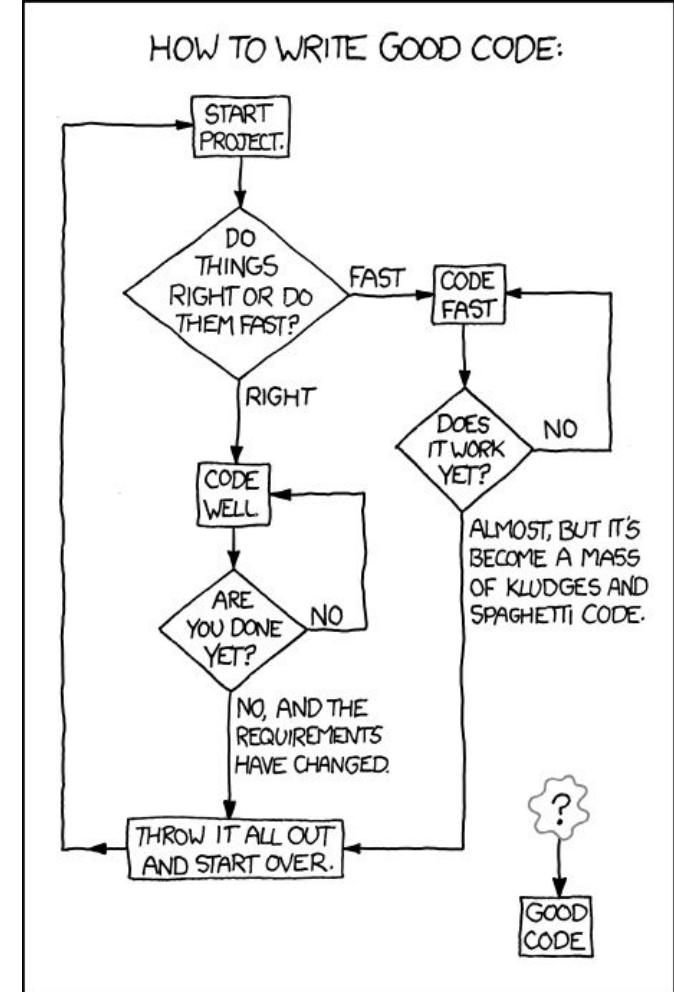
Capitalization matters:
`print()` ≠ `Print()`

`b = a * 2`
`a = 2`

computer: what is a?

The path to writing good, efficient code

1. Make it **work**
2. Make it **right**
3. Make it **fast**



XKCD, <https://xkcd.com/844/>

The path to writing good, efficient code

1. Make it **work**
2. Make it **right**
3. Make it **fast**

Our goal is to get to this step

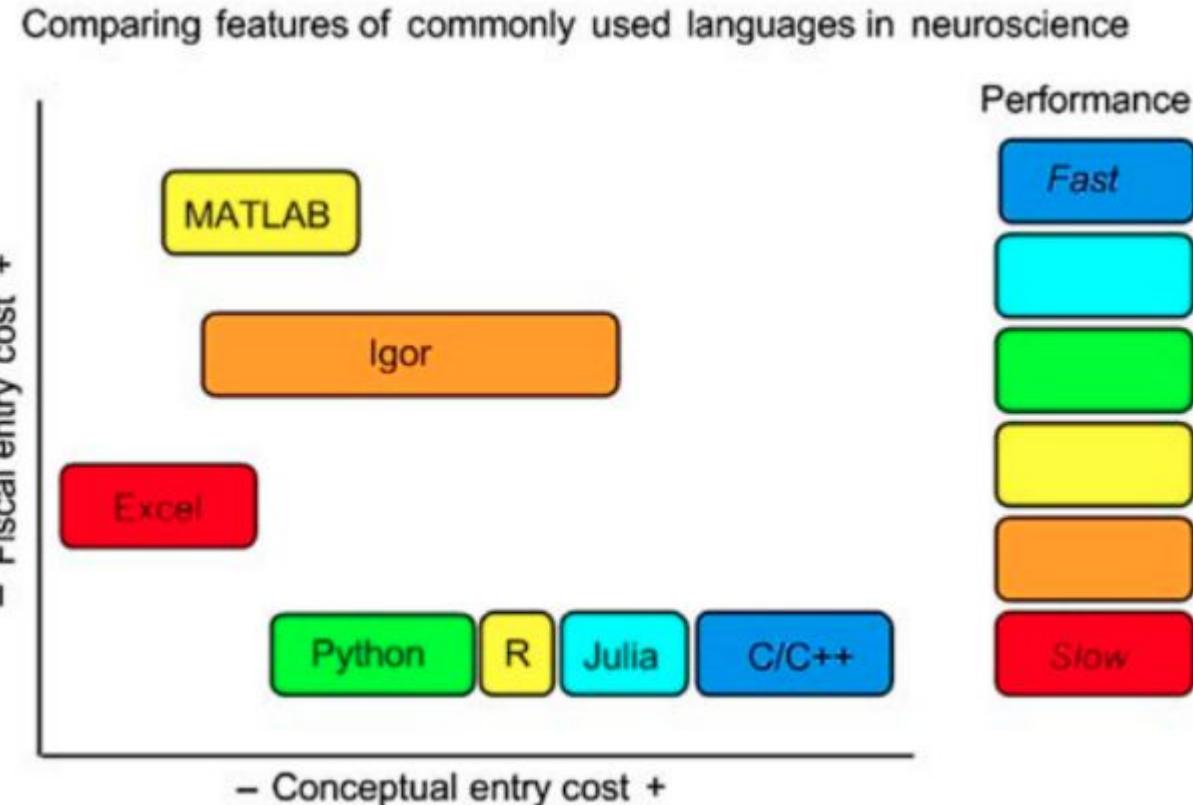
If you ultimately became a back-end programmer, you'd care about step 3.

For most problems biologists face, step 3 isn't paramount.



Considerations for choosing a programming language

- Fiscal & conceptual entry
- Usage in particular field or profession



From Wallisch (2017)

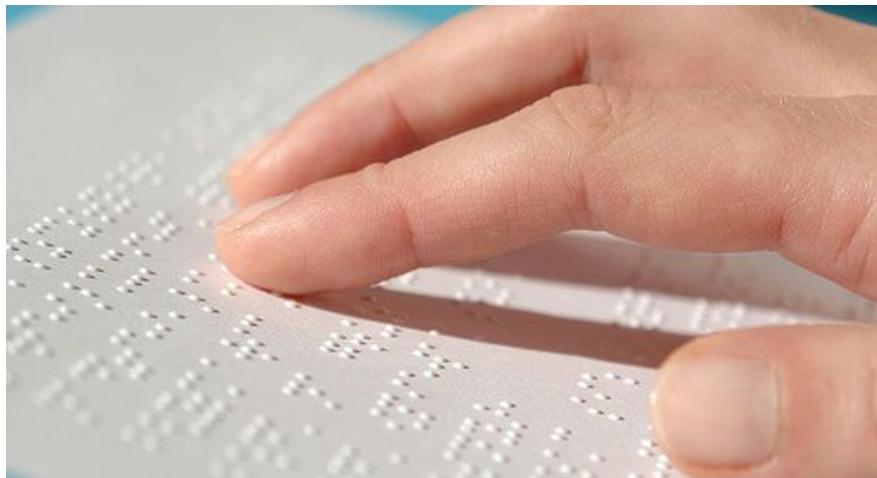
All coding languages eventually need to talk to the computer in binary:

01001000 01100101 01101100 01101100 01101111 00100001

(hello)

[Learn How To Write Your Name In Binary Code](#)

There are many types of binary code, beyond computers



Braille

<https://www.afb.org/blindness-and-low-vision/braille/what-braille>

A • -	J • - - -	S • • •
B - • • •	K - • -	T -
C - • - •	L • - • •	U • • -
D - • •	M --	V • • • -
E •	N - •	W • - -
F • • - •	O - - -	X - • • -
G - - •	P • - - •	Y - • - -
H • • • •	Q - - • -	Z - - • •
I • •	R • - -	

Morse code

https://www.discoveryworld.org/about/blog/discover_at_home/morse-code/

In this class, we'll use Python

- Programming language, development led by Python Software Foundation (www.python.org)
- Uses concise structure & wording similar to human language
- An **interpreted** language — it doesn't speak *directly* to the computer
- Can be used for many purposes, from web programming, to creating games, to analyzing & visualizing data
 - Extension: '.py'
- We'll also work in **Jupyter Notebooks**
 - Extension '.ipynb'



Course logistics

“What is one thing you’re hoping to gain from this course?”

Course Objectives

- Read and run basic Python programs, recognizing the structures used (i.e. variables, conditionals, loops, functions) and explaining how they work
- Manipulate and create objects in Python, including data structures and classes
- Write, edit, and execute Python code in Jupyter Notebooks as well as the command line
- Visualize and run hypothesis-testing on simple datasets in Python
- Implement common algorithms for analyzing biological data (e.g., time series, images) and determine when such computations are appropriate

Grading breakdown

In-class work & participation (10%)

Assignments (50%)

Midterm (15%)

Final project (25%)

Notes:

- Assignments & project components lose 10% each day they are late.

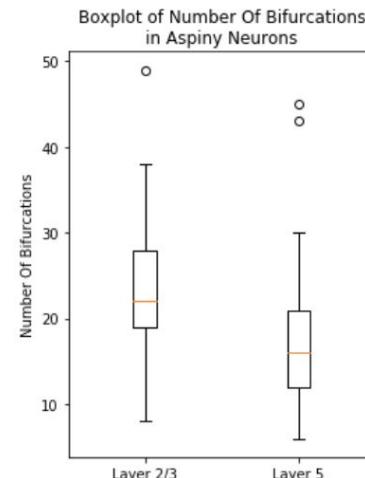
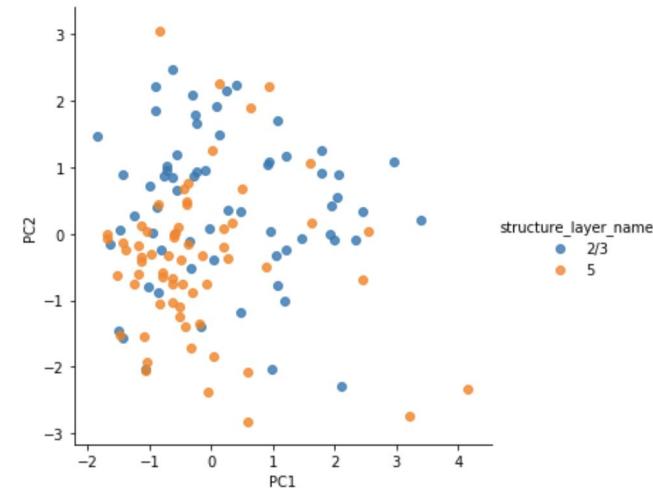


Assignments

- Due **every Tuesday at 5 pm**
- Worth 2.5-10% each
- Completed individually
- Programmatically graded (via Datahub/NBGrader)
- This week, we'll walk through how to submit these.

Project, groups of 2-3

- Includes the project proposal, code, and deliverables.
- Your final project will take some sort of raw biological data and analyze it to draw conclusions.
- We will discuss possibilities for your project as we move through the course.



Organization of content in this course

- **Lectures:** Information that I present in class; many PDF, some via Jupyter Notebooks, or a mix of the two
- **Materials:** Jupyter Notebooks that we will manipulate in class, and that will be useful sandboxes for you
- **Resources:** Additional resources that can help provide more background information to supplement your learning
 - See syllabus as well as links at the end of lectures

END OF YEAR SALE - SAVE 50%

0 1
Days0 6
Hours1 1
Minutes0 6
Seconds[VIEW PLANS](#)

DATAQUEST

[COURSES](#)[STUDENT STORIES](#)[WE'RE HIRING](#)[BLOG](#)[START LEARNING](#)[LOG IN](#)

Learn Data Science

Whether you're new to the field or looking to take a step up in your career, Dataquest can teach you the data skills you'll

Take a FREE course!

 Email Password

You can also sign up for **Stepik** (<https://stepik.org/course/56730/>) or **DataQuest** (free!) & complete lessons in parallel with our course.

Python Basics for Data Analysis (Skill Path) or Data Scientist in Python (Career Path)

Office hours

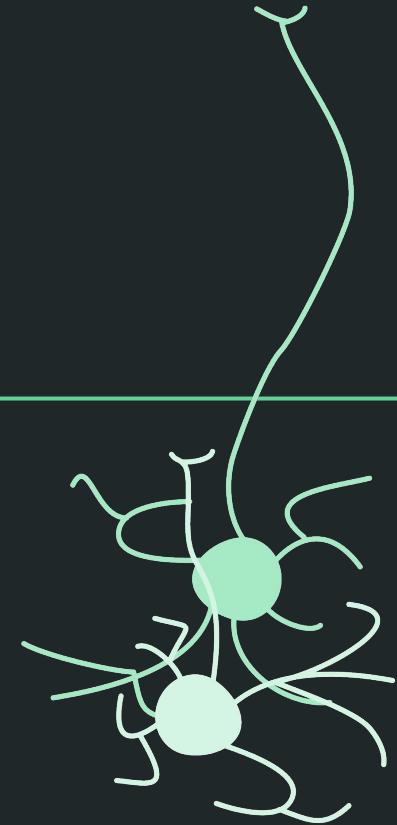
On Zoom, where it's easier to troubleshoot!

Why should you come to office hours?

- You have clarifying questions about the course or its content
- You have concerns about the course and your progress
- You'd like to talk about career paths in biology or neuroscience



Tools for this class



Course Tools



Submitting non-coding assignments & managing grades

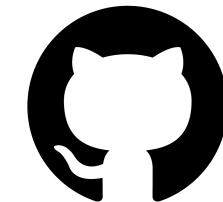


jupyterhub

Coding exercises & assignments



Sharing private course content



GitHub

Sharing public course materials
<https://github.com/BILD62>

Interacting with course materials



You can find all of our course materials on either Canvas or the course GitHub: https://github.com/BILD62/BILD62_WI23

Lectures

In other words, PDF slides shown during class.

Hosted on GitHub in the Lectures folder

If I use both a PDF and a Jupyter Notebook during lecture, these numbers will match

Materials

Jupyter Notebooks

You can pull these locally or to DataHub, or look at them online via GitHub or Colab/ Binder



Assignments

Jupyter Notebooks, submitted through **Assignments** tab

Answers posted in the Assignments folder on Github



THE MAGIC LINK FOR THIS
COURSE:

[https://datahub.ucsd.edu/hub/user-redirect/
git-sync?repo=https://github.com/BILD62/BILD62_WI23](https://datahub.ucsd.edu/hub/user-redirect/git-sync?repo=https://github.com/BILD62/BILD62_WI23)



THE MAGIC LINK FOR THIS COURSE:

Sync with your datahub:



[https://datahub.ucsd.edu/hub/user-redirect/
git-sync?repo=https://github.com/BILD62/BILD62_WI23](https://datahub.ucsd.edu/hub/user-redirect/git-sync?repo=https://github.com/BILD62/BILD62_WI23)



*Where our course
content lives*

To clone Materials to DataHub:

1. Click on the magic link.
2. Log in to DataHub as prompted.
3. You'll be in our course folder now!
4. If you want, save your own copy by adding your initials to the end of the file name. **DO NOT DO THIS FOR ASSIGNMENTS!**
5. Next time you click the link, you'll have a fresh copy, plus your copy.

To interact with Jupyter Notebooks on your computer

OPTIONAL

1. Install Anaconda with Python 3.7 for your operating system.
2. If you're using Windows, [download git](http://www.github.com/BILD62/BILD62_WI23.git).
3. In Terminal (Mac) or the Anaconda Prompt (Windows), clone the repository by running the following command:
`git clone http://www.github.com/BILD62/BILD62_WI23.git`
4. Open Jupyter Notebook. There are two ways to open:
 - o In Terminal (Mac) or the Anaconda Prompt (Windows), type **jupyter notebook**
 - o Open Anaconda Navigator and launch jupyter notebook
5. On the Jupyter landing page, navigate to the notebook and open it.
 - o It will open in a browser but is *not* using an internet connection.

This is a new(ish) class —
thanks for your patience!

We'll also be trying to learn
about how this is going with
some ***education research***.



Exempt Information Sheet

You are being invited to participate in a research study titled Coding Attitudes of Biology Students Coding Attitudes of Biology Students. This study is being done by Ashley Juavinett, PhD from UC San Diego. You were selected to participate in this study because you are a student in a class that is being studied.

The purpose of this research study is to create knowledge that has the potential to improve the learning and educational experience of students at UC San Diego and beyond, and specifically to understand student attitudes towards coding

Your participation in this research should last approximately 15 minutes. If you agree to take part in this study, you will be asked to take pre- and post-course surveys containing about 20 multiple choice items and several open-ended questions.

Your participation in this study is completely voluntary and you can withdraw at any time. Choosing not to participate or withdrawing will result in no penalty or loss of benefits to which you are entitled, and will not adversely impact your standing with the instructional staff of your course. You are free to skip any question that you choose.

Complete both pre- and post-surveys on Qualtrics, gain 1% extra credit

Introduction to the UCSD DataHub & Jupyter Notebooks

A screenshot of a Jupyter Notebook interface. At the top, there's a navigation bar with icons for file operations, a Python logo, 'Logout', and 'Control Panel'. Below it is a toolbar with icons for file, edit, view, insert, cell, kernel, widgets, help, and various cell execution and validation buttons. The main area shows the title 'jupyter ProgrammingFundamentals' and the status 'Last Checkpoint: 11/05/2019 (autosaved)'. The bottom right corner shows 'Not Trusted' and 'Python 3'.

About Jupyter Notebooks

- Jupyter is a loose acronym for Julia, Python, and R
- Run in a web browser!
- Usefully, it will show plots directly in the notebook as you work your way through, performing analyses in real-time
- Two main components:
 - **Kernel:** the engine that runs the code
 - **Dashboard:** landing page where you can see the notebooks you've created

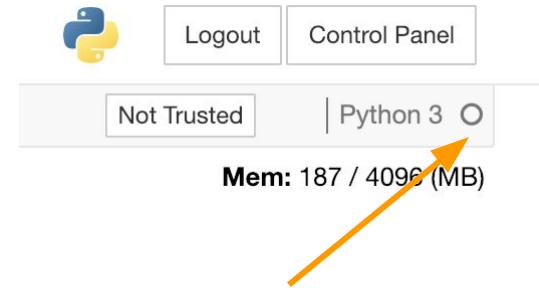


Using Jupyter Notebooks

- **Cell:** the main organizational structure of the notebook
 - Use **Shift+Enter** to run a cell (or press Run)
 - You can run cells out of order, and move cells around!
 - Cells can be **code** (the default) or **markdown** (descriptive text or images)
 - Code cells have **In[]** :
 - If there is a star (**In[]* :**), that means your cell is running
 - Change between code & markdown using dropdown menu (or keyboard shortcuts)
 - Turns **green** in edit mode

Using Jupyter Notebooks (continued)

- You can clear your **namespace** and get a fresh start by restarting the kernel



- If you change anything in the cell, you need to re-run it.
- For help:
 - Help > User Interface tour
 - Help > Keyboard Shortcuts

You can tell if the kernel is busy by whether or not the circle next to Python 3 (upper right corner) is filled or not. (filled = busy)

Let's get into a Jupyter notebook!
Use the [magic link](#) to sync up
your DataHub with our Materials
folder, and open notebook 01.

Before next class...

- Take the entry survey <https://forms.gle/RPN26BKDbwqvyz9B9>
- Take the computing attitudes survey
https://ucsd.co1.qualtrics.com/jfe/form/SV_doRycCO2ZCnvBb0
- Access Canvas (canvas.ucsd.edu) & DataHub (datahub.ucsd.edu)
- (Optional) Sign up for Stepik and/or DataQuest
- (Optional) Install Python 3.7 (via the Anaconda distribution) on your computer (<https://www.anaconda.com/distribution/>)

You only *really* need access to the DataHub, but having the ability to use Python & Jupyter Notebooks on your local computer *may* be useful!