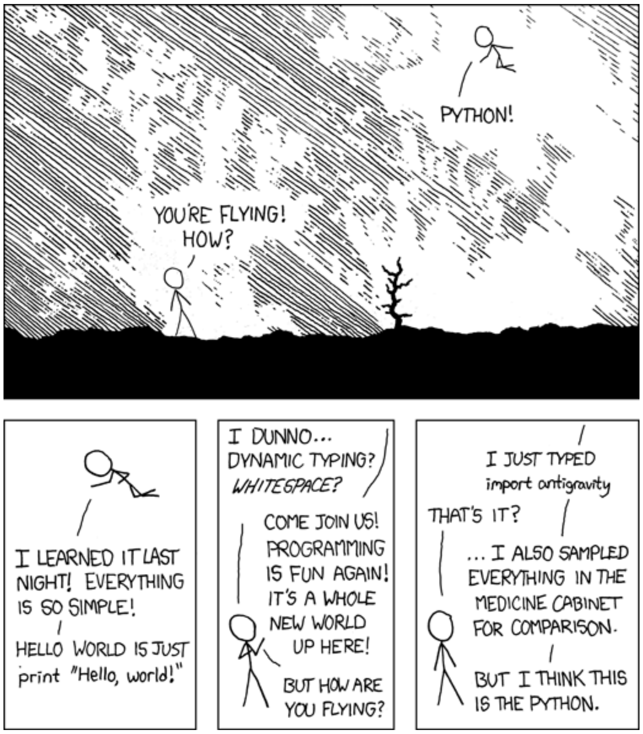
BILD 62 | Introduction to Python for Biologists

WINTER 2024


PYTHON!
YOU'RE FLYING! HOW?
I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"
I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?
I JUST TYPED import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.

Instructor Ashley Juavinett

Instructional Assistants
Felix Taschbach
Ber Al Jaibaji

Office hours
Dr J: Wednesdays @ 10 am on Zoom (link on Canvas)
Felix: Wednesdays @ 2 pm
Ber: Mondays @ 10 am in H&SS 1145L

Class Schedule
Lectures: Tues/Thurs, 12:30 -1:50 pm in TATA 3201
Discussion: Tuesdays at 2 pm on Zoom

Course GitHub: https://github.com/BILD62/BILD62_WI24

Course Description: Introductory class for biology students interested in using Python for data analysis and visualization. Course covers the basics of programming in Python and introduces students to various implementations of Python analyses for biological data such as time series and images.

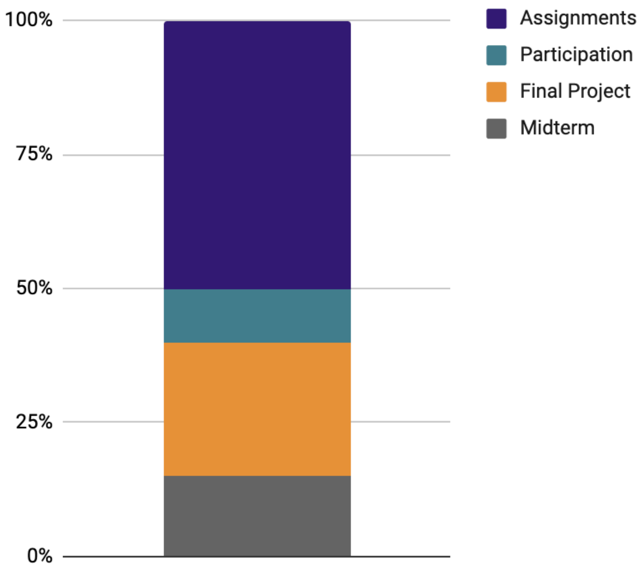
By the end of this course, you will be able to:

- Read and run basic Python programs, recognizing the structures used (i.e. variables, conditionals, loops, functions) and explaining how they work
- Write, edit, and execute Python code in Jupyter Notebooks as well as the command line
- Manipulate and create objects in Python, including data structures and classes
- 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

- **In-class work & participation (15%)**:
  - Attendance (5%): If you attend >80% of discussion sections (8 out of 10), you'll receive all 50 points. You may optionally assign these points to the midterm (you need to let me know before the midterm if you like to do this).
  - In-class work (10%): We will complete several activities in class for credit. If you miss these, you'll need to contact the IAs and make them up before Friday at 5 pm.

- **Assignments (45%)**: Weekly take-home coding assignments will support your progression through the course topics. Assignments will be submitted through the DataHub (http://datahub.ucsd.edu) and graded automatically using a tool called NBGrader.
  - All assignments are released by Thursday at 5 pm, due the following **Wednesday at 5 pm**, and are worth 2.5-5% each.
  - These assignments should be completed individually and without chatGPT *unless instructed*. They should take you about 1-2 hours each.
  - At the end of the quarter, **I will drop your lowest assignment grade.**



- **Final Projects (25%)** Includes the project proposal, code, and deliverables. During finals week, we'll have a final project presentation where you'll share your project with classmates and visitors.

- **Midterm Exam (15%)** About two thirds through our course, we will have an in-person exam that will test you on the fundamentals of Python and computational thinking.

Additional notes about grading:

- We will be using Canvas (http://canvas.ucsd.edu) to manage grades and assignments.
- **Late policy**: Assignments and projects will lose -10% for each day they are late.
- **Grading Scheme:** Final scores will be converted to letter grades, where A=100-90%, B=89-80%, C=79-70%, D=69-60%, and F=59-0%. For positive and minus grades, A+ = 97-100, A = 93-96.99, A- = 90-92.99, B+ = 87-89.99, B = 83-86.99, and so on.
- Final scores are as you see them on Canvas, once all of your assignments are graded. There is no rounding up to the closest score.

Course Resources

Computing Resources

You will need access to a computer and an internet connection for our course. It will also significantly help if you have a laptop (or an iPad + keyboard) for both lectures and discussions. If you need a laptop for the quarter, you can request a loaner laptop by filling out this form: https://support.ucsd.edu/students?id=sc_home.

Textbook

There is no official textbook for this course. However, we'll be relying on several online resources:

- Mansuri & Moshiri, **Intro to Computer Science & Python Stepik Course**
- VanderPlas, **Whirlwind Tour of Python**
- Software Carpentry, **Plotting and Programming in Python**
- Libeskind-Hadas & Bush, **Computing for Biologists** **(Note:** only available for purchase, optional).
- Python Game (made by Niema Moshiri, a UCSD CS professor)
  - Learn Programming: Python - Retro on Steam
  - Learn Programming: Python - Remake on Steam

Course Philosophy

A note on our course's environment

We'll be working together to create an equitable and inclusive environment of mutual respect, in which we all feel comfortable to share our moments of confusion, ask questions, and challenge our understanding. Everyone should be able to succeed in this course. If you do not feel that is the case please let me know.

Course accommodations

If you need accommodations for this course due to a disability, please contact the Office for Students with Disabilities (osd@ucsd.edu) for an Authorization for Accommodation letter. Please speak with me in the first week of class if you intend to apply for accommodations. For more information, visit http://disabilities.ucsd.edu.

This course, and the work it entails, is for *you*

You won't benefit if others (including AI assistants) do your work. If you're unclear about what constitutes cheating in this course, please ask. Cases of academic dishonesty or cheating will be first handled by me, and then by the Academic Integrity Office. If you become aware of cheating in this class, you can anonymously report it.

We'll be relying a lot on other people's code as we learn. Here are some guidelines as to how you should use other code in the process of writing your own, as well as how you can talk to your classmates about the code we're working with in class:

Do explain the thought process behind your code.
Do share the general steps you took to solve a problem.
Do describe your code to others, either verbally or in writing.
Do use examples on the internet to inform your code.

Do not screenshot someone else's code.
Do not directly share your code with others, either in text or image format.
Do not directly copy 5+ lines of code from examples on the internet or chatGPT.
Do not share the values of variables that are explicitly asked for in the validation of the question.

We will be discussing how we can best use generative AI tools such as chatGPT to help us learn how to code. You're encouraged to use these tools to help you learn, but *not* to complete assignments. One of our goals in this class is for you to learn how to read and inspect code, including code written by an AI assistant. I reserve the right to ask you to explain any code you have written on an assignment or on the final project, and your midterm will not permit the use of an AI assistant.

Syllabus

(subject to change!)

Date	Topic	Assignments
Week 1	To set the foundation for this course, we'll introduce the approaches and tools that we'll use throughout, as well as the motivation for learning how to code as a biology student.	
Jan 9	**#1: Welcome to BILD 62!** Introduction to the course, people & tools *Discussion*: Submitting assignments	
Jan 11	**#2: Where Python lives, and how to talk to it** *Note*: In-Class Activity, due Monday @ 5 pm	*Due Friday at 5 pm: In-Class Activity* *Before Monday:* Take the computing attitudes survey for extra credit

Week 2 Fundamental coding skills in Python

Jan 16	**#3: Data structures** (lists, tuples, and dictionaries) & troubleshooting *Discussion*: Coding with chatGPT	*Due Wednesday at 5 pm*: **a0**
Jan 18	**#4: Functions, booleans & conditionals**	

Week 3 Fundamental coding skills in Python (continued)

Jan 23	**#5: For Loops** *Discussion*: Week 2 recap	*Due Wednesday at 5pm*: **a1**
Jan 25	**#5** (continued)**: More loops and the power of mindset**	

Week 4

Jan 30	**#6:** Object-oriented programming *Discussion*: Week 3 recap & error handling	*Due Wednesday at 5pm*: **a2**
Feb 1	**#7:** NumPy *Note*: In-Class Reflection, due Friday @ 5 pm	

Week 5 Scientific Computing

Feb 6	**#8:** Visualizing data *Discussion*: Week 3 recap	*Due Wednesday at 5pm*: **a3**
Feb 8	**#9:** Data analysis & computational thinking	

Week 6

Feb 13	**#10:** Pandas *Note*: In-Class Activity, due Friday @ 5 pm *Discussion*: Midterm review & BILD 62 Bingo [Questions]	*Due Wednesday at 5pm*: **a4**

Feb 15	**Information & group formation for final projects**	

Week 7

Feb 20	**Midterm**	*Due Wednesday at 5pm*: **a5**
Feb 22	**#11**: Linear algebra: Intuitions & implementation in Python	

Week 8

Feb 27	**#11:** Linear Algebra (continued)	*Due Wednesday at 5pm*: **a6**
Feb 29	Signal & image processing in Python	*Due Friday at 5pm*: **Project proposal**

Week 9 _______ - *informatics*

Mar 5	**Neuroinformatics: Using Python in literature searches**	*Due Wednesday at 5pm*: **a7**
Mar 7	**Neuroinformatics: Using Python in literature searches**	

Week 10 Wrapping up

Mar 12	Documentation, version control, and collaborating on code *Note*: In-Class Code Review Activity, due Friday @ 5 pm	*Due Wednesday at 5pm*: **a8**
Mar 14	Next steps in bioinformatics, biological data science & computational approaches to big data	

Final Project Showcase: Tuesday, March 19, 11:30 am - 3:30 pm