CMSC 398z
Effective use of AI Coding Assistants and Agents

Bill Pugh and Derek Willis

Sept 5th, 2025

brief
Instructor
Introductions

State of AI Coding tools

brief thoughts now, more next week

Official class goals

Learn effective use of AI coding tools

Discuss and adopt ~~best~~ good practices for using them

Learn to use AI tools as coaches and thinking partners

Evaluate different LLMs for coding

Discuss challenges and weaknesses of AI coding tools

Learn to use Python and to write code that make calls to LLMs

Improve software development skills

Discuss how AI changes what professional software developers do

Updating the UMD CS curriculum

Discussions are starting about how to update the UMD CS curriculum

Discussions happening at many universities.

- See NYTimes article in reading for next week
- Also discussions about the impact on the entire undergrad curriculum

Leo Porter from UCSD will be visiting Oct 8th and talking about how they updated their CS1 course to use Python & Copilot.

- We will be reading about his work before his visit

You all can be advocates for this effort

Hoped for course byproducts

Gain insight as to how the CS curriculum could be updated

- To adjust to the use of AI coding tools
- How to use AI tools as coaches
- To provide more software development skills and experience

Evaluate ways to bring more active learning into the classroom

Understand how AI coding tools could be used across the entire undergraduate curriculum

classroom expectations, pairing, grading, etc.

Slides and video of lectures will be made available after class

Most of the required coding for the course will be done in pairs in class

Lots of discussions around each table and work with partners

    Be nice and supportive

You may need to pay to get access to services we will use

Switch up tables and partners each class. Try to pair with as many people as possible over the course of the semester

Required coursework should require working in class + 2 hours outside of class

Grading based on what you learn, not what you complete.

Markdown text

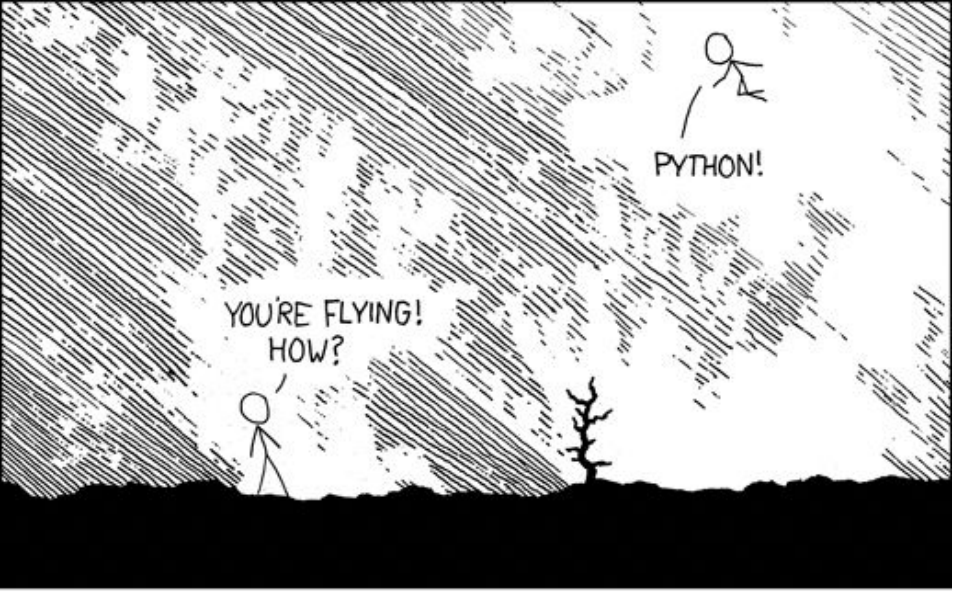
Very standard way to have text that is both readable directly and can be formatted for easier reading. Can include bullet items, links, etc.

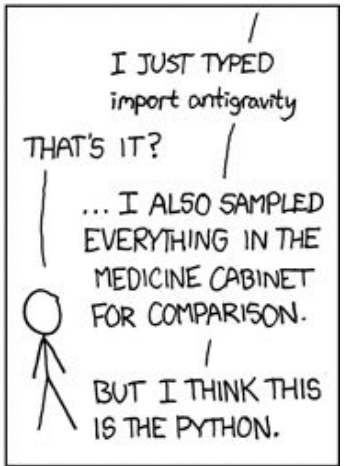
Git repositories generally contain a README.md file that describes the repository

The first project we will be doing as a README.md file with instructions.

In VSCode, Command-Shift V show a preview of the currently selected Markdown file (Ctrl-Shift V on Windows)

In providing context to GenAI tools, it often helps to convert
PDFs and web pages to Markdown first.





Python

We'll be doing much of our programming in Python

Bill never did much of any programming in Python before starting to prepare for this course

https://xkcd.com/353/

Python Core concepts

Untyped programming language

Common built-in types:

`None`, `bool`, `int`, `float`, `string`, `range`, `list`, `dict` and `tuple`

`lists` are similar to Java ArrayLists

with lots of syntax for things like slices and list comprehensions

`tuples` are immutable fixed length arrays

`dicts` are like Java HashMaps

variables and fields aren't declared. You just assign to them

Libraries - one of Python's superpowers

Lots of libraries and standard repositories for them

How do you make them accessible to your code?

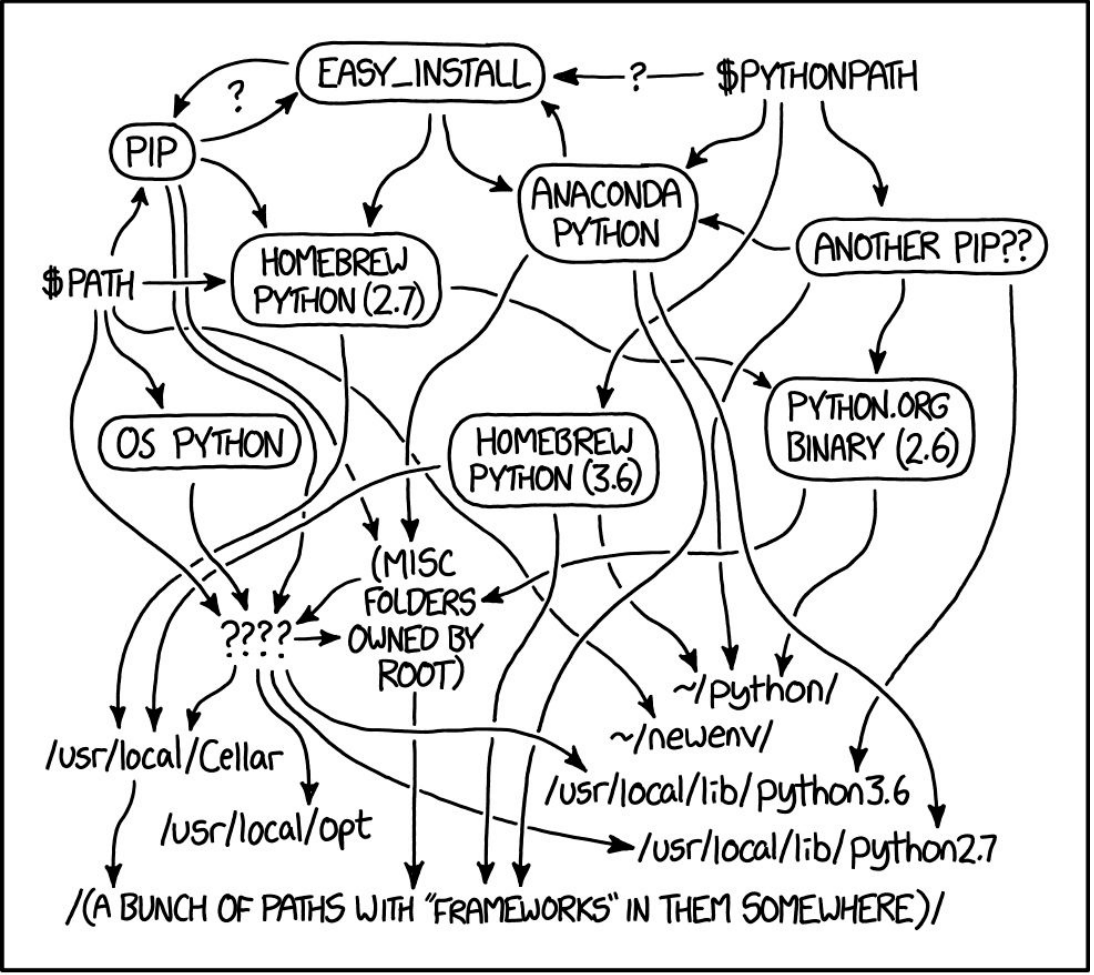
- You define a text file that names the libraries you need
- Various tools to download libraries and make them available to your code
- Some libraries are automatically available, you just need to import them

The oldest and most common way to handle libraries is pip

- We aren't going to use pip

Python install hell

HTTPS://XKCD.COM/1987/



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

uv for python library management

Lightning fast, written in rust

Can also put `python` in your path, so you don't have to type `python3`

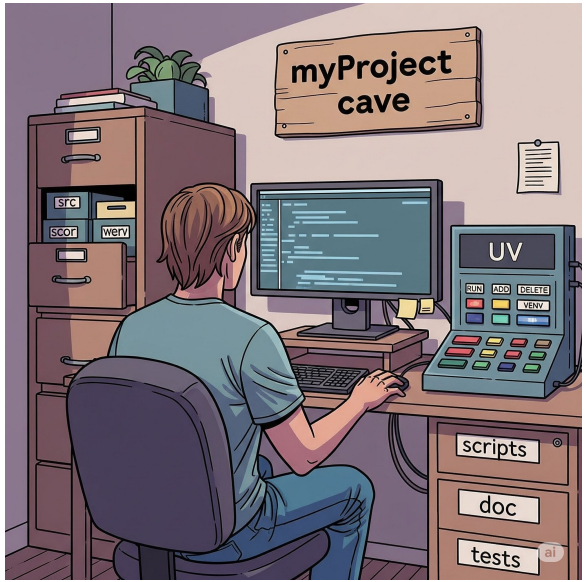
```
uv python install --default  --preview
```

Provides commands for managing your dependencies

Python code generally lives in a project folder, which include files that describe the library dependencies

`uv` gives you 3 ways to run python code from the command line

Use the uv command within a project directory



You can use

uv run *myscript.py*

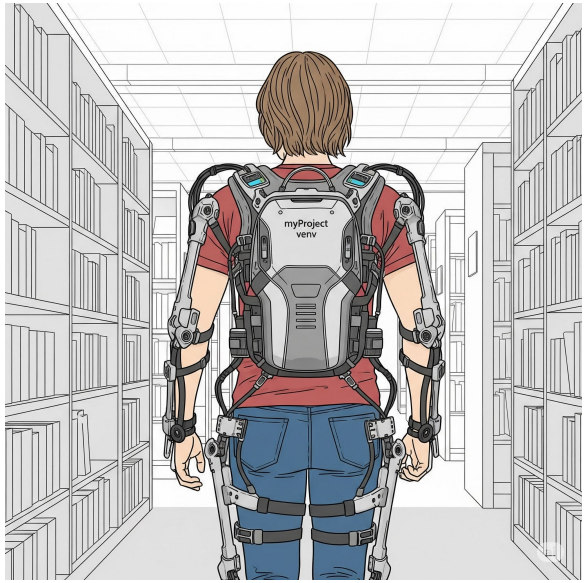
to run any scripts or tools associated with the project or any libraries used by the project

Create a virtual environment you can put on and take anywhere

uv venv builds the venv - the exoskeleton

VSCode uses the venv to run python code when using uv

Both `python` and `<toolname>` are added to your path



Use uvx to execute a tool associated with a library

You can be anywhere, and decide that you suddenly need a cow to say something.

uvx cowsay -t "More grass plz."

First break from lecture coming up

After we present the next slide on getting your computer setup

Introduce yourself other at your table

- Give your name and favorite programming language or fun fact about yourself
- Briefly describe your experience with python and with the tools described on the next page

Everyone work to complete setup, help others as useful

Setting the stage

Load notes for week 1 from course web page for links

On MacOS, install xcodetools

For Windows, install Windows Subsystem for Linux (WSL 2)

Install uv

Install VS Code

Install Python extension for VS Code

Install and enable Copilot

Don't worry if you can't get all of this done today

Using Copilot in coaching mode

We are going to spend a few weeks doing projects designed to get you leveled up in using AI coding tool to create Python

For the project we are going to have you do first, when I typed the first line of a key function into VS Code, it autocompleted the entire function, > 20 lines

Going to see if we can use Copilot as a coach,
rather than an answer machine

Claude Code added an output mode with that functionality,
we are trying to bring it to VS Code & Copilot

Using the coach

Understands that you already know Java and C, may not know Python

You can complete project by just typing natural language into agent window

Think about the how to solve the problem

Decide what functions you want to create, what they should do.

Decide functionality of small blocks of code (e.g., a loop, perhaps with a nested conditional), ask it to write the python code for that

You can jailbreak the coach, get it to write all the code. But that isn't the point of using a coach

Bill and Derek will live code a project

Bill found this project in a Intro to Python textbook

- Thought about it a bit, but hasn't tried writing any code for it.

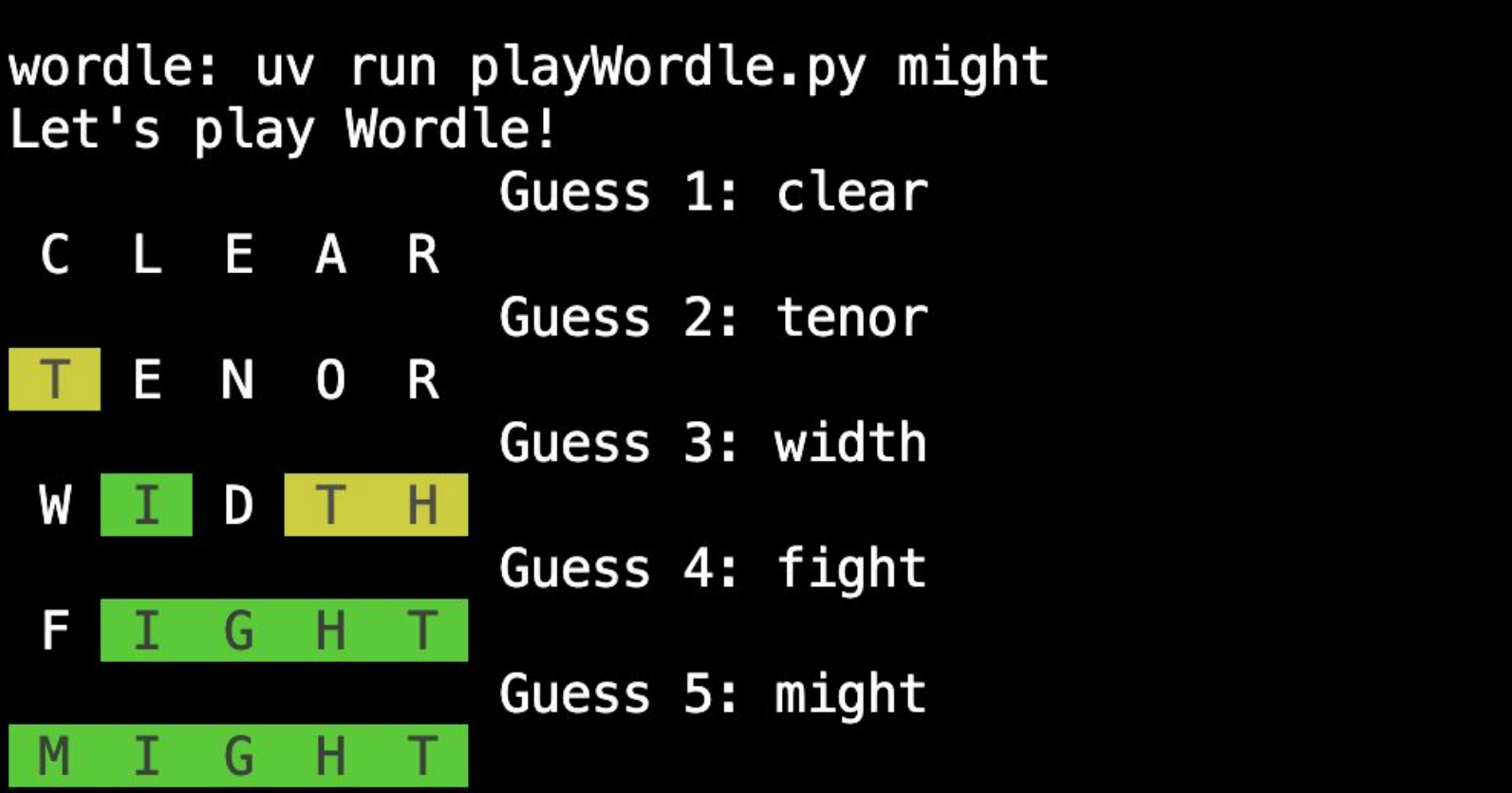
Derek has no idea what it is

- Derek will be on the keyboard

It is a little more complicated than the project we are going to ask you all to do

Live coding demo

Project for you all to work on



Getting ready for project

Shuffle seats if needed to try to get an even number of people at each table

Create folder for project

Get project downloaded, download links on week 1 web page

Open zip file in a folder

Open folder in VS Code

Perform uv setup described in README

```
uv run testColor.py
```

Pair up

First discuss plan and questions for project

Ideas for functional decomposition

Strategy for figuring out right place and wrong place matches

Do parallel 5 minute sprints, each on own computer

- alternative: traditional pair programming, share a laptop

Submit using submit server

- Must include your python code and your chat dialog.
  - Save chat in chat.md file
- Enter the name of your partner in a comment at the top of playWordle.py