



**Welcome to HWRS 401/501 aka:**

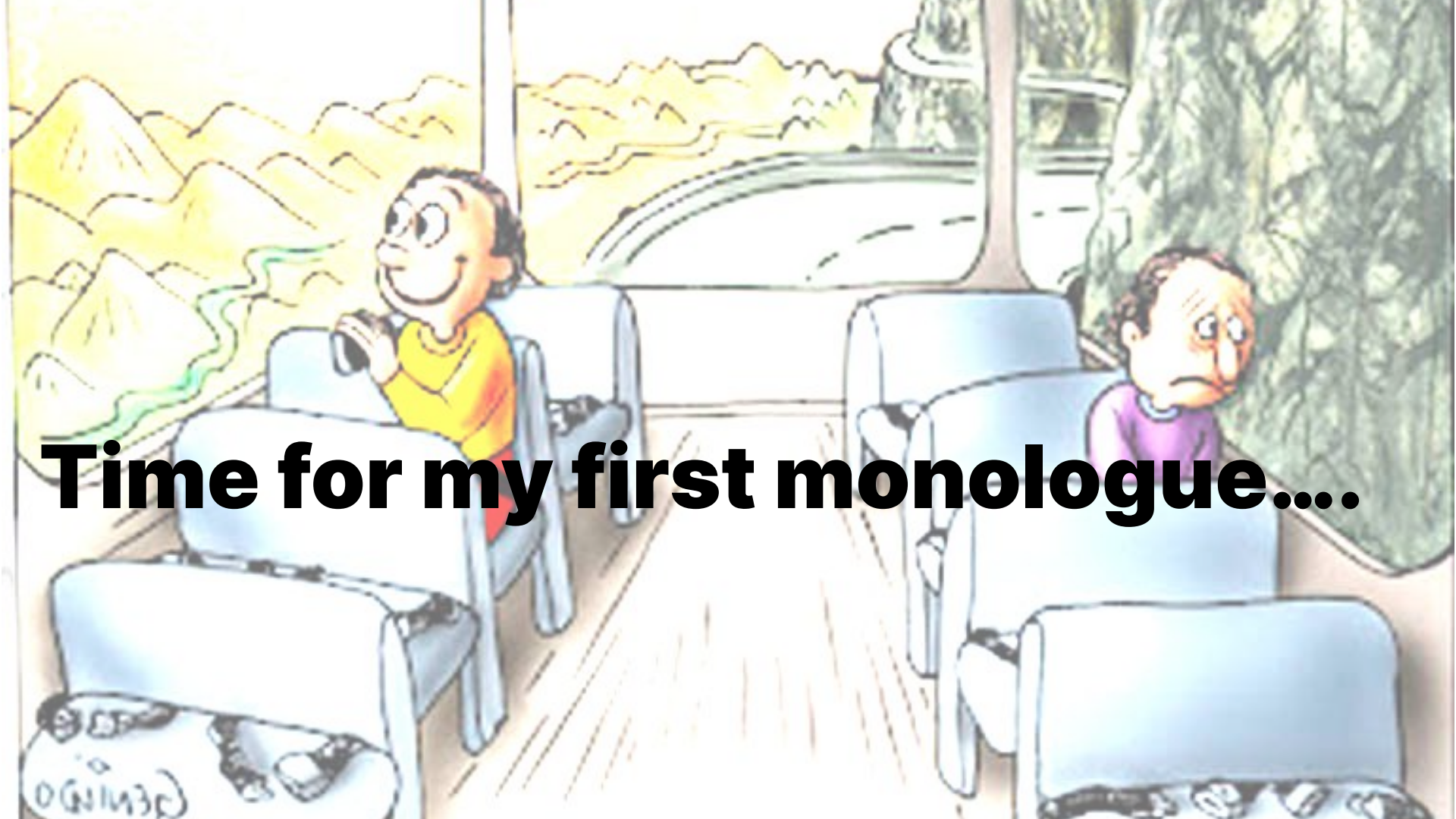
**“Tools for Data Handling  
and Analysis in Water,  
Weather, & Climate”**

A cartoon illustration of a train interior. On the left, a man in a yellow shirt sits happily, looking out the window at a bright, hilly landscape. On the right, a man in a purple shirt sits sadly, looking out the window at a dark, dense forest. The train tracks and a rainbow are visible through the windows. The text 'YOU' is above the happy man, 'PEOPLE NOT IN HWRS 401/501' is above the sad man, and 'DOING RESEARCH OR ANALYSIS' is in the center.

**YOU**

**PEOPLE  
NOT IN  
HWRS  
401/501**

**DOING  
RESEARCH  
OR ANALYSIS**



**Time for my first monologue...**

# **At a high level, this is what you can expect to learn:**

- A rough and basic history of why computing is important in environmental, water, and climate research

# **At a high level, this is what you can expect to learn:**

- A rough and basic history of why computing is important in environmental, water, and climate research
- Basic programming skills and concepts

# **At a high level, this is what you can expect to learn:**

- A rough and basic history of why computing is important in environmental, water, and climate research
- Basic programming skills and concepts
- An understanding of the overall scientific python data stack

# **At a high level, this is what you can expect to learn:**

- A rough and basic history of why computing is important in environmental, water, and climate research
- Basic programming skills and concepts
- An understanding of the overall scientific python data stack
- Common data structures for environmental and climate data



# **At a high level, this is what you can expect to learn:**

- A rough and basic history of why computing is important in environmental, water, and climate research
- Basic programming skills and concepts
- An understanding of the overall scientific python data stack
- Common data structures for environmental and climate data
- How to find & use publicly accessible data



# **At a high level, this is what you can expect to learn:**

- A rough and basic history of why computing is important in environmental, water, and climate research
- Basic programming skills and concepts
- An understanding of the overall scientific python data stack
- Common data structures for environmental and climate data
- How to find & use publicly accessible data
- How to structure and share reproducible research

# **At a high level, this is what you can expect to learn:**

- A rough and basic history of why computing is important in environmental, water, and climate research
- Basic programming skills and concepts
- An understanding of the overall scientific python data stack
- Common data structures for environmental and climate data
- How to find & use publicly accessible data
- How to structure and share reproducible research
- The value & process of community and collaboration in data analysis

# **At a high level, this is what you can expect to learn:**

- A rough and basic history of why computing is important in environmental, water, and climate research
- Basic programming skills and concepts
- An understanding of the overall scientific python data stack
- Common data structures for environmental and climate data
- How to find & use publicly accessible data
- How to structure and share reproducible research
- The value & process of community and collaboration in data analysis
- Considerations for ethical and equitable computing in your research

# **At a high level, this is what you can expect to learn:**

- A rough and basic history of why computing is important in environmental, water, and climate research
- Basic programming skills and concepts
- An understanding of the overall scientific python data stack
- Common data structures for environmental and climate data
- How to find & use publicly accessible data
- How to structure and share reproducible research
- The value & process of community and collaboration in data analysis
- Considerations for ethical and equitable computing in your research
- Who might be collaborators for your future research

# Making the case for data science literacy

- We live in unprecedented times in terms of data, compute, and tooling for environmental, climate, and Earth sciences
- I believe that scientists in such fields can use these facts to better understand the Earth system and advance both science and policy
- Taking this approach seriously requires researchers to take computing seriously as an approach to scientific inquiry
- Taking this approach seriously requires researchers to take a critical eye to where their data comes from and understand approaches to discovering errors and/or limitations

# Let's talk about you

- What's your background?
- What are you interested in?
- What do you already know?
- What do you want to know?
- Do you have cool pets, good recipes, or random thoughts to share with the class?



# Let's talk about me

- You can tell that picture is me by the pixels
- I'm a postdoc working with Laura Condon
- Largely my research interests are focused around hydrologic modeling, machine learning, and understanding how meteorologic data is used in making hydrologic predictions
- I'm also interested in open source science broadly
- Just call me Andrew





# Syllabus time!!!

- Yeah, again, it's 8am - not my choice. What's your preference for congregation/community?
- We can flip the classroom, I am happy to record lectures and post on youtube or whatever and spend class time on exercises
- Grades will be posted via canvas, but all other materials will be shared via github (to be explained)
- Office hours - I set some but what works for y'all?
  - Mon 2-3pm & Thurs 9:30-10:30am
- I want this to be collaborative - we have a curriculum, but if needs/interests arise let's respond and adjust!!!

# Grading

Item	Grade %
Participation	20
Forecast submissions	40
Cheat Sheets	14
Code review	6
Submitted scripts	12
Forecast evaluation	8

# Cheat sheets

- You'll submit one cheat sheet per module
- These are distilled versions of your notes
- Should give something for you to refer back to
- Should give me something to make sure I know ya'll are following along
- Format is free form, can be as simple as organized and formatted bullet points but feel free to go nuts and make infographic, sketches, or whatever

Beginner's Python Cheat Sheet	
<b>Variables and Strings</b> <i>Variables are used to store values. A string is a series of characters, surrounded by single or double quotes.</i>	<b>Lists (cont.)</b>
Hello world <pre>print("Hello world!")</pre>	List comprehensions <pre>squares = [x**2 for x in range(1, 11)]</pre>
Hello world with a variable <pre>msg = "Hello world!" print(msg)</pre>	Slicing a list <pre>finishers = ['sam', 'bob', 'ada', 'bea'] first_two = finishers[:2]</pre>
Concatenation (combining strings) <pre>first_name = 'albert' last_name = 'einstein' full_name = first_name + ' ' + last_name print(full_name)</pre>	Copying a list <pre>copy_of_bikes = bikes[:]</pre>
<b>Lists</b> <i>A list stores a series of items in a particular order. You access items using an index, or within a loop.</i>	<b>Tuples</b> <i>Tuples are similar to lists, but the items in a tuple can't be modified.</i>
Make a list <pre>bikes = ['trek', 'redline', 'giant']</pre>	Making a tuple <pre>dimensions = (1920, 1080)</pre>
Get the first item in a list <pre>first_bike = bikes[0]</pre>	<b>If statements</b> <i>If statements are used to test for particular conditions and respond appropriately.</i>
Get the last item in a list <pre>last_bike = bikes[-1]</pre>	Conditional tests <pre>equals      x == 42 not equal   x != 42 greater than x &gt; 42 or equal to x &gt;= 42 less than   x &lt; 42 or equal to x &lt;= 42</pre>
Looping through a list <pre>for bike in bikes:     print(bike)</pre>	Conditional test with lists <pre>'trek' in bikes 'surly' not in bikes</pre>
Adding items to a list <pre>bikes = [] bikes.append('trek') bikes.append('redline') bikes.append('giant')</pre>	Assigning boolean values <pre>game_active = True can_edit = False</pre>
Making numerical lists <pre>squares = [] for x in range(1, 11):     squares.append(x**2)</pre>	A simple if test <pre>if age &gt;= 18:     print("You can vote!")</pre>
	If-elif-else statements <pre>if age &lt; 4:     ticket_price = 0 elif age &lt; 18:     ticket_price = 10 else:     ticket_price = 15</pre>
	<b>Dictionaries</b> <i>Dictionaries store connections between pieces of information. Each item in a dictionary is a key-value pair.</i>
	A simple dictionary <pre>alien = {'color': 'green', 'points': 5}</pre>
	Accessing a value <pre>print("The alien's color is " + alien['color'])</pre>
	Adding a new key-value pair <pre>alien['x_position'] = 0</pre>
	Looping through all key-value pairs <pre>fav_numbers = {'eric': 17, 'ever': 4} for name, number in fav_numbers.items():     print(name + " loves " + str(number))</pre>
	Looping through all keys <pre>fav_numbers = {'eric': 17, 'ever': 4} for name in fav_numbers.keys():     print(name + " loves a number")</pre>
	Looping through all the values <pre>fav_numbers = {'eric': 17, 'ever': 4} for number in fav_numbers.values():     print(str(number) + " is a favorite")</pre>
	<b>User input</b> <i>Your programs can prompt the user for input. All input is stored as a string.</i>
	Prompting for a value <pre>name = input("What's your name? ") print("Hello, " + name + "!")</pre>
	Prompting for numerical input <pre>age = input("How old are you? ") age = int(age)  pi = input("What's the value of pi? ") pi = float(pi)</pre>
	<b>Python Crash Course</b> <i>Covers Python 3 and Python 2</i> <a href="http://nostarchpress.com/pythoncrashcourse">nostarchpress.com/pythoncrashcourse</a>



# **Forecasting on the Verde River**



# About the Verde

- The 192-mile river begins as springs near Paulden
- 40 miles designated National Wild and Scenic River
  - Riparian oasis surrounded by arid land
  - Supports 50+ threatened or endangered species
  - Critical flyway for migratory birds
- Free-flowing except for 2 dams around mile 137
- Supplies ~40% of the surface water SRP delivers annually to Phoenix for municipal and agricultural use

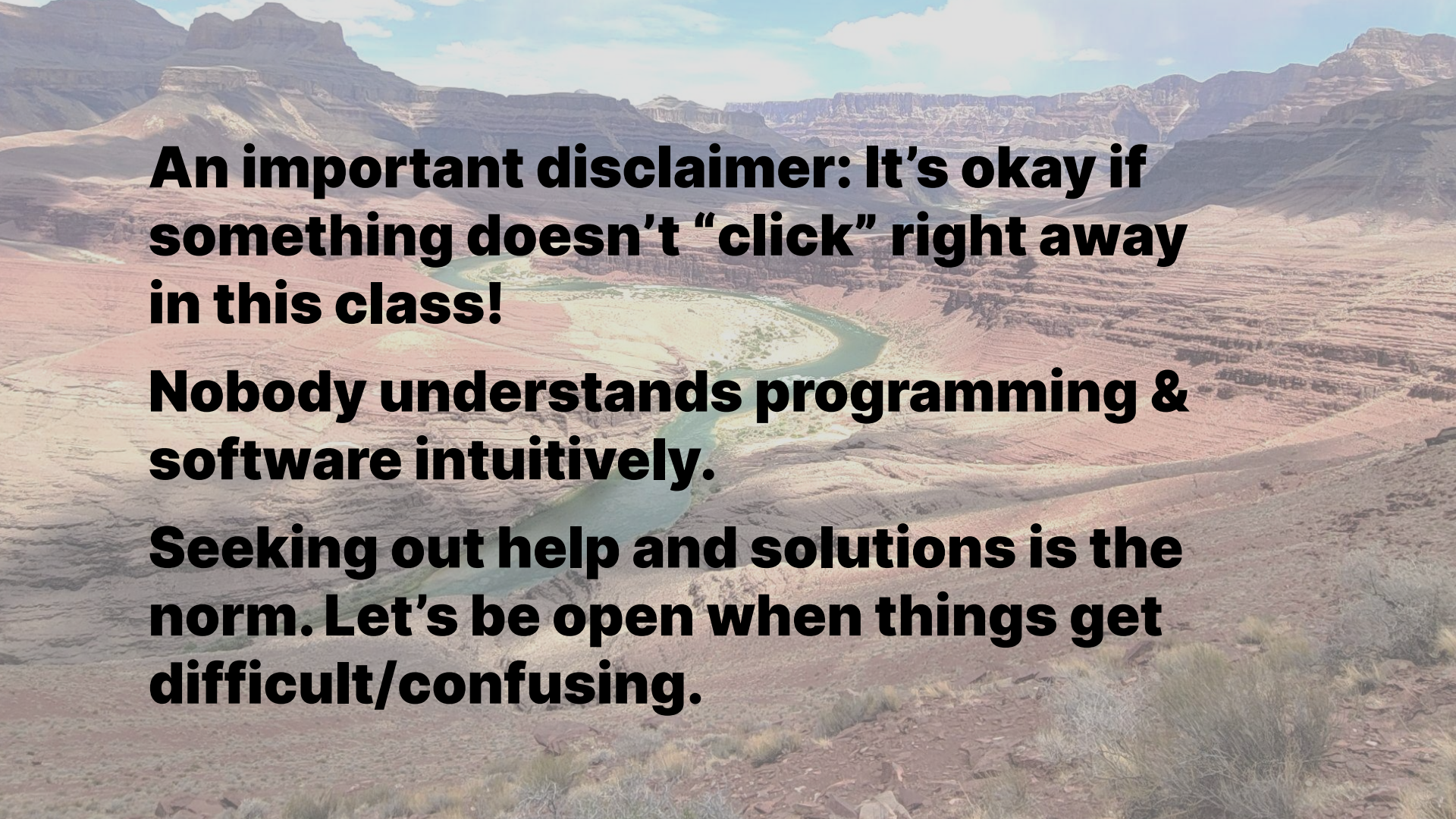


# Verde River Near Camp Verde, AZ (USGS Gauge 09506000)

- Each week I'll ask you to produce a 1 week and 2 week forecast of how much water will flow through the Verde river @ Camp Verde
- Specifics of the assignment will be revealed Thursdays and submissions will be due the following Monday
- You will not be graded for accuracy of results
- You will be graded for completion and explanation of how you used the methods from the week to produce your forecast
- There will be a competition where I track who has the best forecasts overall. 1st place will receive a 5% grade boost. 2nd and 3rd will receive 3% boosts.
- There might be cool trophies too...





A wide-angle photograph of the Grand Canyon, showing the Colorado River winding through the deep, layered rock formations under a clear blue sky with some light clouds.

**An important disclaimer: It's okay if something doesn't "click" right away in this class!**

**Nobody understands programming & software intuitively.**

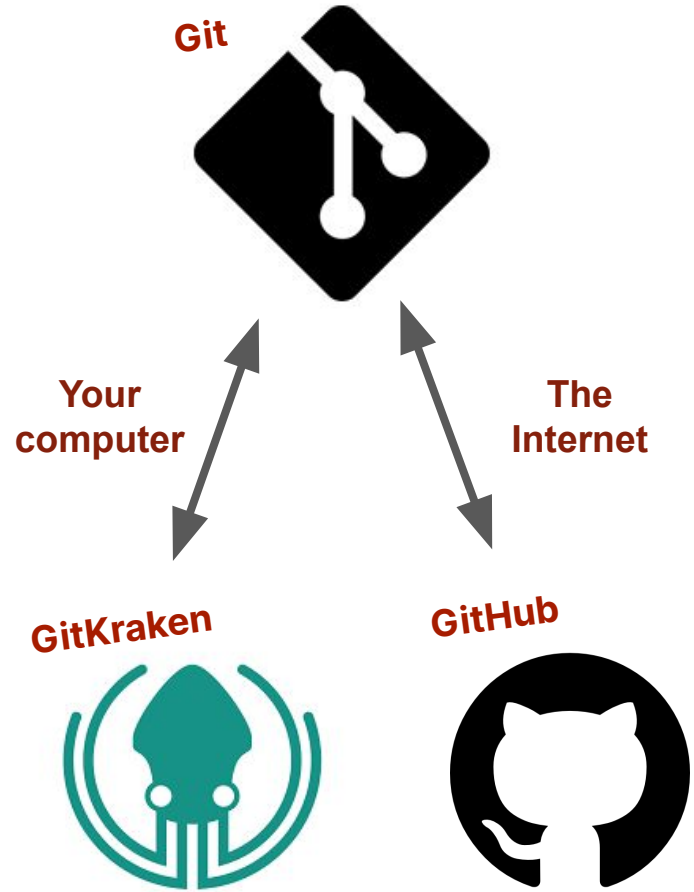
**Seeking out help and solutions is the norm. Let's be open when things get difficult/confusing.**



# Let's see how far we can get...

## Introducing git + GitHub

- Git is software that helps track changes in code/files. Kinda like track changes in word, but on steroids.
- But it's also super f'n hard to use. We'll try to simplify things as much as possible though
- Git is a particular software, but GitHub is a company and web interface built around the software
- We will use GitHub to track assignments and content, as well as help build community
- We will use GitKraken as an app to interact with git and GitHub
- Ignoring how git works for a moment, let's all create a github account.



# Let's see how far we can get...

## Creating github profiles

- My account is here:  
<https://github.com/arbennett>
- The class organization is here:  
<https://github.com/HAS-Tools-Fall2022>
- Once you create your account write your username on the board and I will add you to the class
- When you are added, please navigate to the CourseMaterials22 link and go to the “Discussions” tab and introduce yourself in the “Class role call” thread

# Let's see how far we can get...

## Installing Git

- Please bear with me... I'm still not going to describe git until next session probably
- But, let's try to "git" it installed. Go to <https://git-scm.com/>
- Try to cluster into groups of Windows, MacOS, and (if existing) linux users and walk through steps together

# Let's see how far we can get...

## Installing python

- Python is a complicated programming ecosystem. We'll dive into it more next class.
- For now, we'll be using the anaconda python ecosystem.
- Let's all try to download it via miniconda:  
<https://docs.conda.io/en/latest/miniconda.html>
- Try to cluster into groups of Windows, MacOS, and (if existing) linux users and walk through steps together

# Let's see how far we can get...

## Installing GitKraken

- GitKraken just makes git easier to use.
- Let's all install it from here:  
<https://www.gitkraken.com/>
- Once installed let's log in via our GitHub credentials.
- You should be able to “clone” the class resources at this point
- Depending on time I might return to this later.

# Let's see how far we can get...

## Installing vscode

- VSCode is a code editor. If you already have something else you know how to use, feel free to stick to it
- Basic download instructions here:  
<https://code.visualstudio.com/>
- Once you have it installed, boot things up, and install the python extension. We will probably walk through this together.