Day 1:

* Scholarships for people who are new to coding?
* Go to hack BU to help yourself learn code.

**Syllabus:**

**Required Texts**

Charles R. Severance, [*Python for Everybody*](https://eng.libretexts.org/Bookshelves/Computer_Science/Programming_Languages/Book%3A_Python_for_Everybody_(Severance)) (2020; 2016) – this is an OER, that is, it’s free to read & use. The link I provide is for the online version, if you want a PDF, Kindle, or hardcopy, go here <https://www.py4e.com/book>.

**Required Technology Installations & Packages (Don’t worry, we will download these together!)**

* Python 3.10: <https://www.python.org/downloads/>
* Jupyter Notebook (Anaconda Distribution 64-bit): <https://www.anaconda.com/products/individual>
* Google Colab (for students with small storage): <https://colab.research.google.com/>

**Recommended Resources**

* The Python Tutorial: <https://docs.python.org/3/tutorial/>
* W3Schools Python Tutorial: <https://www.w3schools.com/python/>
* Geeks4Geeks: <https://www.geeksforgeeks.org/python-programming-language/>
* Codecademy: <https://www.codecademy.com/>
* Kaggle Notebooks: <https://www.kaggle.com/notebooks>
* Learn Python: <https://www.learnpython.org/>
* Python Cheatsheet: <https://www.pythoncheatsheet.org/>
* Stack Overflow: <https://stackoverflow.com/>
  + Lectures
    - LECTURES: Every week we have a lecture for 1.5hrs. These provide essential context for the course and we will often deepen skills that you learn in labs. Please note, the room has no video recording option. If you miss a lecture you will need to make it up in office hours.
  + Coding in the wild
    - How to look at the world through coding
    - Its similar to packback in the sense where you’re making posts to see the coding classes
    - After learning to code you will use and see the world differently.
    - Describe your experience
    - You only need five posts
      * You can only post per week tho
    - You only have one shot to post under coding in the wild.
* Labs and lab exercises has tests and you have to keep up every week to konw what is going on in class.
  + If you happen to have a bad week, you can drop two the
  + You will complete 13 labs over the course of the semester. In each lab, you will work on new concepts, and at the end of each lab, you have to complete lab exercises. I will often provide bonus exercises in these labs to help grow your skills. Lab exercises are due 11:59PM the night before your next lab unless otherwise specified.
  + You’ll have time to get the exercises done
  + ***Plan deadline of lab assignments.***
* ***DON”T finish ater the key is out.*** 
  + There is an O credit here.
* Record your video about your game project.
* **Oral presentation critiques:** 
  + Provide feedback to your peers.
  + **Your must respond to the presenter’s discussion board during class b/c it’s graded pass/fail.**
* **Game Project (15%)** 
  + Around the midpoint of the semester, you will start building a game with a partner. This partner should be in your lab section. (There will likely need to be one group of 3 per section). If you do not work with a partner the highest grade you can make is 50/100.
* **Data Project (10%)**
  + For the final coding assignment in this class, you will create a data-based project. You will find an open data set, then clean, analyze, and visualize the data. You can work on this with a partner or alone.
* **Sign-up sheets for the presentation is on the syllabus**
* **Public speaking is a very important skill**
* **Data project:** 
  + Is on the
* ***Emphasizes this:*** 
  + **Office hours are very important because if provides you effective help**
  + **Knowing to troubleshoot yourself is very important.**
  + **GO TO OFFICE HOURS**

**MUST TALK BEFORE DEADLINE OR SHORTLY AFTER DEADLINE:**

| **Late Work Policy Guide** | | |
| --- | --- | --- |
| *Assignment* | *Late Policy* | *Why?* |
| Lectures | * Remember, **two/2 lowest attendance will be dropped** at the end of the semester! * You can **make up a lecture** to get attendance credit, if you have legitimate reasons, by making an **in-person appointment** with me in office hours **within 2 weeks of the original lecture date** | The best way to learn is be physically and mentally in the classroom |
| Labs | * **Any late submissions will get an automatic zero/0** (we will release the corresponding lab key the next day) * Any other late labs require permission from Prof. Xie **prior** to deadline * Remember, **two/2 lowest labs will be dropped** at the end of the semester! | Labs build essential skills; if you miss too many you won’t pass the class;  Doing the labs on time is the best way to keep up with the course. So I strongly suggest that you turn in whatever you have by the deadline! |
| Lab Peer Critiques | * **No late submissions accepted** | You can only complete them if you are in lab and you only need to critique 60% of all in-lab presentations |
| Game Project & Peer Review | * **No late submissions accepted** * Any other late labs require permission from Prof. Xie **prior** to deadline | You will have multiple weeks to work on this project before it is due |
| Coding in the Wild posts | * **No late submissions accepted** | You have 16 weeks to make 5 posts |
| In-lab Presentation | * Requires permission from Prof. Xie **prior** to deadline * **If you fail to present after an extension or rescheduling, you will get an automatic 0** | These are time sensitive and must be completed close to the original lab date |
| Data Project | * **No late submissions accepted** | The class is over |
| Course Reflection | * **No late submissions accepted** | The class is over |

* Do not plagiarise do not write code that is the same as other people.
* Do not use ai to write the code for you.
* The AI will give you solutions, but you won’t understand it.
  + You won’t be able to tell what a good solution is and what a bad solution is
  + **You must build the foundation and think about what you’re looking for.**
* Finish the assignments based on the assignments you learned.
  + **There is a two-strike system**
  + **If you plagiarize once,**
* **No exams or quizzes and lab exercises**
* **What do lab exercises typically look like?** 
  + From lab 2 and onward you will be learning a new coding concept and you’ll be solving questions and problems that test your understanding of the concept.
  + There will be bonus questions on the concept and be on the next stage
    - ***What will the bonus questions look like? Are they based on new info or just more complicated take on info we’ve used?***
    - ***Both; more likely the second***
    - ***We will be providing allthe information you’ll need.***
* ***Starting tips:*** 
  + *New concepts each week so don’t fall behind*
  + *Use office hours*
  + *Charge and bring your computer save and backup your stuff*

***Questions on the survey:***

* *I also recognize that it is better to turn in a partially complete lab rather than a late lab so I do not fall behind.*
* *I recognize that labs are due at 11:59PM the night before my lab section unless otherwise instructed*
* *I understand that attendance is taken in labs via the presentation critiques, and that if I do not post enough of these critiques (60% of total minimum), I will get a zero for the total grade.*

**Week 2:**

* <https://docs.google.com/presentation/d/1Hy_oOgvuxy16eEMRd2XjMGH_BZMN89_nXLcMGoHPwMk/edit?usp=sharing>
* Coding in the wild. In discussions
* More on in-lab presentation

More on in-lab presentations:

* Sign up early
* Design one question
* Demonstrate key concepts
* Enough for 5 min presentation

Data project:

* Some conceptualization
* Mostly skills and techniquest to do data analysis
* Topics are wide open!

Concerns i nclass:

* memorizing ?
  + She says learning by doing is what will help you the most .
  + If you forget it, you can just google it or search it up.

What are interfaces:

* Graphical user Interface (GUI) = visual interface we interact with to run software instead of a text-based commend line interface (CLI)
* We’re using jupyter notebook, which is an Integrated Development Environment (IDE), a type of GUI
  + Other IDEs you may one day use
  + Jupyter is used widely to publish and visualize data.
* Other IDes you may one day use: visual studio code; Google Colab; Atom; PyCharm; spyder; sublime text 3; IDLE
* Jupyter is used widely to publish and visualize data

Digital Literacy: what is it and how good are you?

* You need code and ou need digital literacy

**Super helpful if you have some second language learning experience:**

* Language learning! – top 1 (writing)
* Math & logics - secondary
* Everyone has different coding style and has their own edges
* Creativity is endless

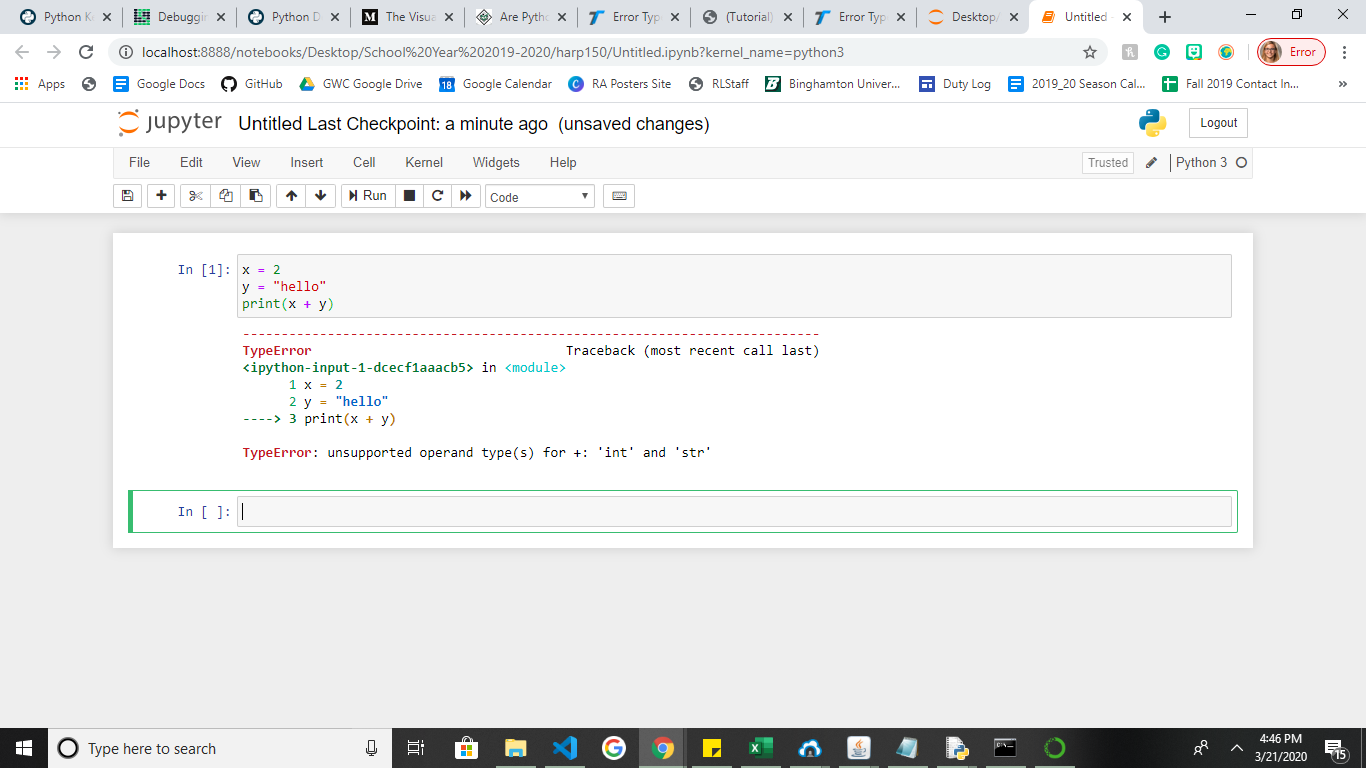
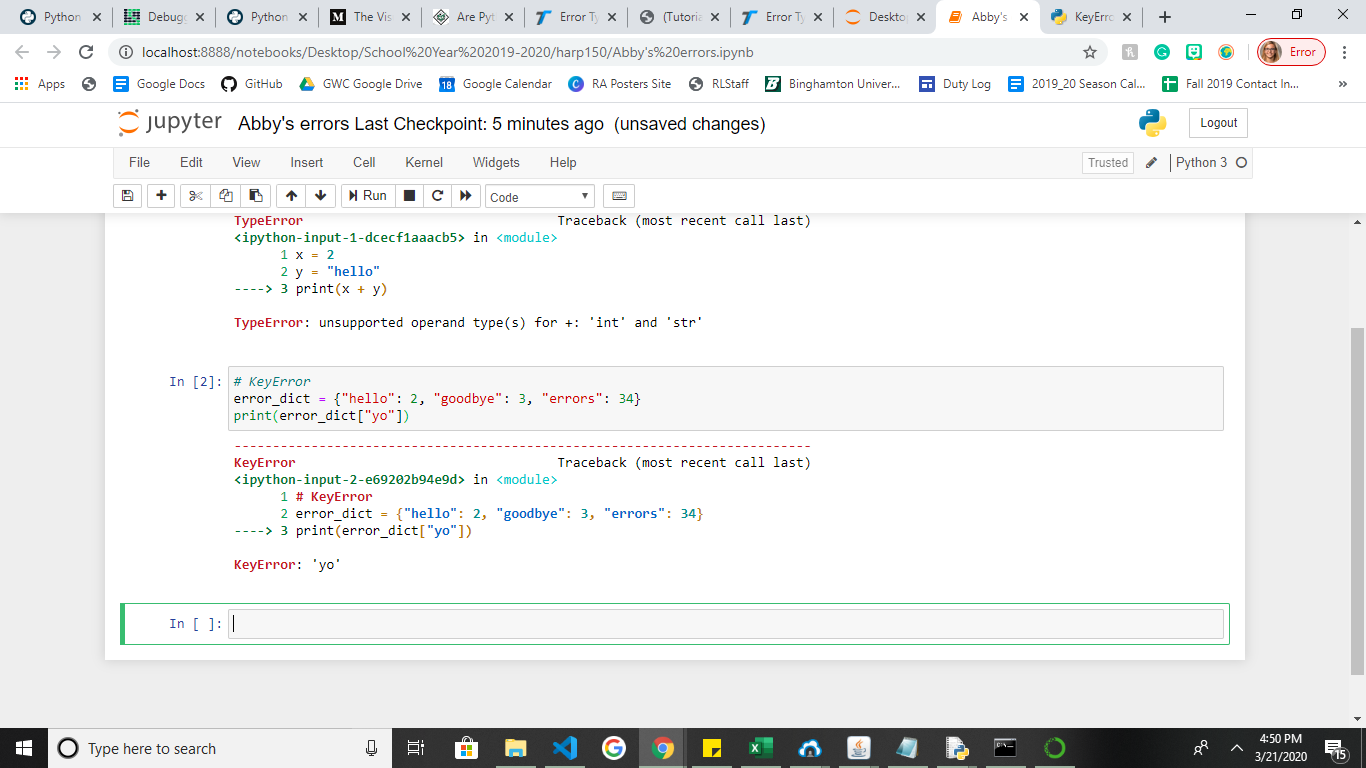
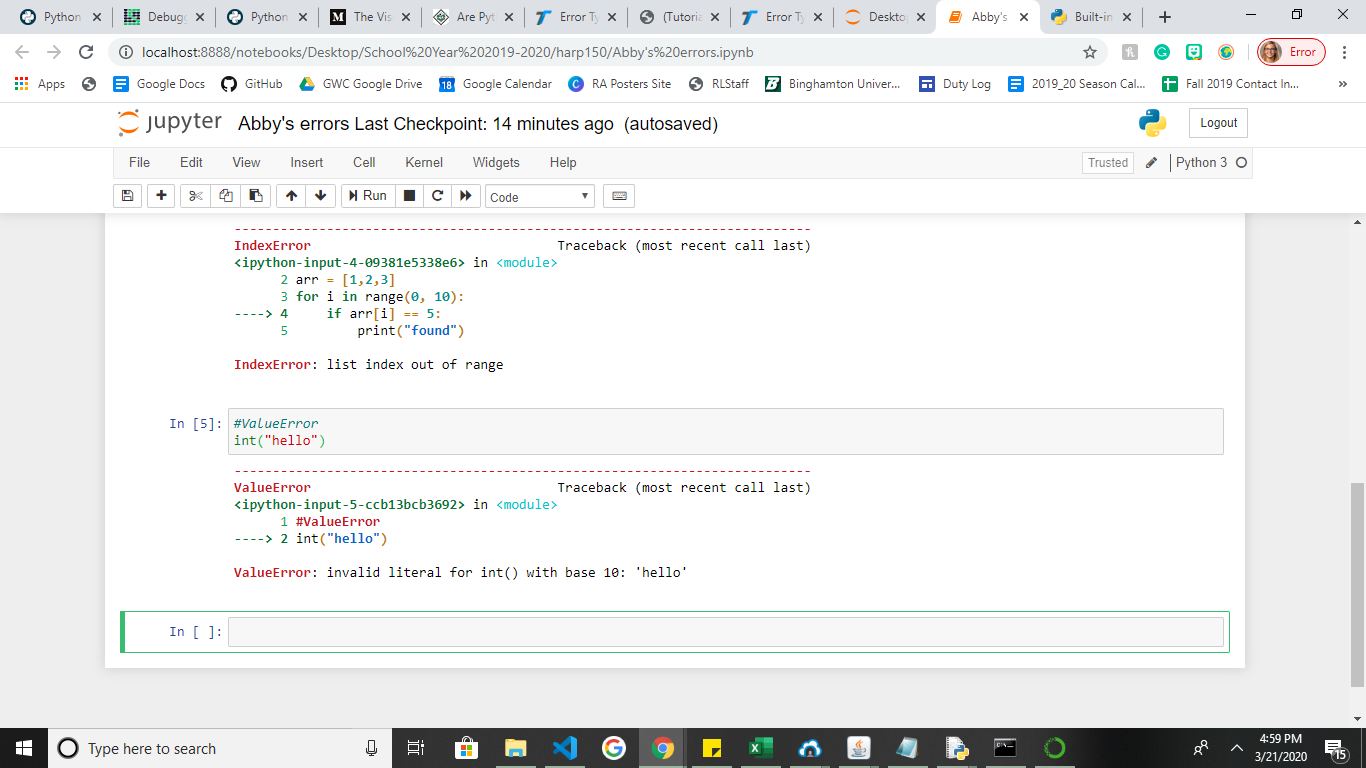
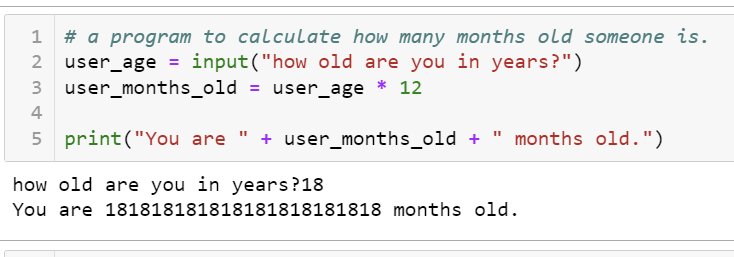
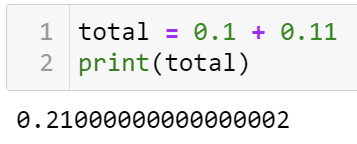
**Let’s talk syntax and semantics - the building blocks of a programming language**

* He were ! very nice (syntax errors)
* That cute tadpole was a volcano (semantic/logical error)
* Most common syntax error in Python: wrong indentation, punctuation, or spelling

**Week 3 1/30/2024 lecture over zoom:**

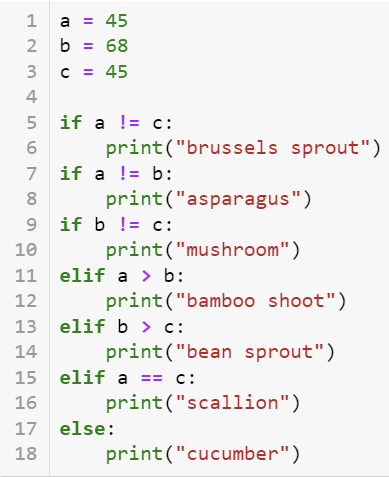
* <https://docs.google.com/presentation/d/1gn8QBsCayyRSD-Ee0Or5CSwGm_ss36X-5e7sYXTH5CE/edit#slide=id.g20242417527_0_216>
* Troubleshooting
* Concatenating = link together in a series or chain
* **Most common error types you’ll face so far:**
* **(see full list** [**here**](https://www.tutorialsteacher.com/python/error-types-in-python)**)**

| NameError | When a variable is not found |  |
| --- | --- | --- |
| KeyError | KeyError is thrown when a key is not found in the dictionary.  Ex: Indexing by a key in the dictionary, but the key does not exist. |  |
| TypeError | TypeError is thrown when an operation or function is applied to an object of an inappropriate type.  **Ex:** Adding a string to a integer |  |
| SyntaxError | When you have an error in syntax |  |
| ValueError | ValueError is thrown if a function’s argument is of an invalid type.  Ex: Int() is a function that converts a decimal to a whole number. Here, we call it with a string. |  |
|  |  |  |

* **Type error example:**
* 
* **Key errors** 
  + If you’re trying to pick up specific terms
  + ****
* **Get() function, which returns TRue if teh key exists and false if it doesn’t exist.**
* 4 basic hard rules for writing variables:
* Must be one word (no spaces allowed)
* Only letters, numbers and the underscore punctuation (\_)
* Cannot begin with a number
* Cannot use keywords
* **ValueError :** the values are unable to be provided b/c the values are different.
  + ****
* **She says that it’s important for you to understand something by doing it, not memorizing it verbatim.**
* **Error messages are useful, but they can be tricky**
  + Sometimes, it reports where the error was detected, but not where it actually occurred! - see slide 18 example
  + Error message is outright confusing, especially in the beginning - because error message is written from the perspective of the computer, instead of the human programmers!
  + No error message does not necessarily mean your code is working as intended.
* ****
* Beautiful is better than ugly.  
  Explicit is better than implicit.  
  Simple is better than complex.  
  Complex is better than complicated.  
  Flat is better than nested.  
  Sparse is better than dense.  
  Readability counts.  
  Special cases aren't special enough to break the rules.  
  Although practicality beats purity.  
  Errors should never pass silently.  
  Unless explicitly silenced.  
  In the face of ambiguity, refuse the temptation to guess.  
  There should be one-- and preferably only one --obvious way to do it.  
  Although that way may not be obvious at first unless you're Dutch.  
  Now is better than never.  
  Although never is often better than \*right\* now.  
  If the implementation is hard to explain, it's a bad idea.  
  If the implementation is easy to explain, it may be a good idea.  
  [Namespaces](https://realpython.com/python-namespaces-scope/#the-global-namespace) are one honking great idea -- let's do more of those!
* “weird” behavior of float.
  + Some floats don’t round properly.
  + If you use the round() function, you can define the digits you want to round it up to.
  + 
  + Ex. round(total,2)
    - Means you round the float to **two decimal places.**
* What do lab presentations look like?
  + Only design one exercise question using one of the important concepts you learned in lab 2.
  + Common errors when you solve the problem.

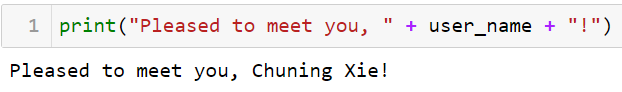
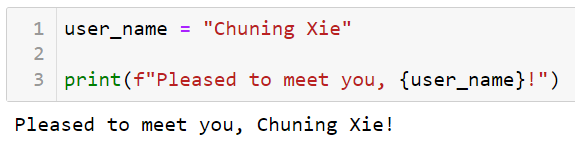
2/13/2024

* **According to her, all the labs can be completed with the knowledge from the labs and lecture.** Find that.



* Asparagus and mushrooms would be printed because the **if statements come first. When all the if statements are completed it doesn’t move on to elif statements.**
  + *The purpose of elif statements is to perform functions if none of the if statements are*

An f string starts with the letter f or F, followed by quotation marks, and automatically concatenate strings with variables enclosed in curly brackets, like {user\_name}.



* The purpose of this is so you can **quickly**

***Check the slide show to find out the different companies that use python in different ways.***

* Data analysis: pandas, numpy
* Data visualization: matplotlib, seaborn
* Text analysis: NLTK, Spacy
* They operate in python and have their own language to them,
  + also , you think
* <https://docs.google.com/presentation/d/12chxlPX0twlNIgnzncvM_h5AXUBMYJe8_QD0DkJ8KwM/edit?usp=sharing>
* Originally coded in Lisp, but re-coded to Python in

December 2005

* Web framework, [web.py](https://webpy.org/), is now an open source project

**Spotify**

Primarily uses Python for back end

services and data analysis

Why Python?

* Speed of development using Python in the pipeline
* The communicative Python community
* Check out some of their [open-source code](https://spotify.github.io/#python)

Spotify sponsors many Python conferences each year

Uses over 6000 individual Python processes that work together

**Netflix**

Mainly uses Python for data analysis, server

maintenance (similar to Spotify), and to build apps

that track changes in history and security

This is how recommendations are made for you!

Why Python?

* Developers choice! - Netflix allows software

Engineers to select the language they code in

Why Python?

* Wide range of code libraries
* Flexibility of code for developers
* Readability of the code
* If you cannot figure something out with your

own code, there are others who have shared similar

code out there that can be borrowed

Guido van Rossum worked here as a developer!)

* Used with the desktop client connected to their cloud-based

System

* Make use of various libraries on both Mac and Windows

machines to allow a unified experience. This is because Python

does not come preinstalled on Windows and depending on your

Mac, your Python version will vary

Learn more:

* Info on their API and what the developers do: <https://www.dropbox.com/developers/documentation/python>
* Interviews with the engineers who are writing and working with the Python code: <https://talkpython.fm/episodes/transcript/30/python-community-and-python-at-dropbox>
* Uses Python for infrastructure management, and to process images (particularly resizing)
* At least 20% of their infrastructure is built using Python

Why Python?

* Ease of use, simplicity
* Not requiring much code to write or maintain
* Popularity
* Speed with development

Open source projects published by Facebook:

Facebook Ads API:  
<https://github.com/facebook/facebook-python-business-sdk>

Python Async IRCbot Framework: <https://github.com/facebook/pyaib>

**Week 12: data 101**

**Data project:**

* Full rubric/explanation/data websites on pg. 17-19 of syllabus
* Choosing a dataset to analyze
* Requirements:
  + Must include clear research questions
  + Must clean dataset
  + Must use at least 6 visualizations
* Can work with a partner or alone
* Expectation is that you will have to teach yourself a few concepts – this is part of the goal of the assignment
* Due May 10th @ 11:59PM no exceptions barring major emergency

**Thinking about data:**

You might hear the terms:

* Raw/unprocessed data
* Datasets
* Full rubric/explanation/data websites on pg. 17-19 of syllabus
* Choosing a dataset to analyze
* Requirements:
  + Must include clear research questions
  + Must clean dataset
  + Must use at least 6 visualizations
* Can work with a partner or alone
* Expectation is that you will have to teach yourself a few concepts – this is part of the goal of the assignment
* Due May 10th @ 11:59PM no exceptions barring major emergency

**Some places you can find open datasets!**

[Kaggle](https://www.kaggle.com/datasets) [Open Access Directory](http://oad.simmons.edu/oadwiki/Data_repositories)

[GitHub Awesome Datasets](https://github.com/awesomedata/awesome-public-datasets) [NYC Open Data](https://data.cityofnewyork.us/browse)

[Data.gov](https://www.data.gov/) [Google Dataset Search](https://datasetsearch.research.google.com/)

[Buzzfeed Data](https://github.com/BuzzFeedNews/everything) [Reddit Datasets](https://www.reddit.com/r/datasets/)

[List of 33 sites for datasets](https://www.forbes.com/sites/bernardmarr/2016/02/12/big-data-35-brilliant-and-free-data-sources-for-2016/#6ac39a47b54d) [AWS Open Data](https://registry.opendata.aws/)

[World Bank Open Data](https://data.worldbank.org/) [Data.World](https://data.world/datasets/open-data)

[List of Free Open Data Sources](https://www.freecodecamp.org/news/https-medium-freecodecamp-org-best-free-open-data-sources-anyone-can-use-a65b514b0f2d/)

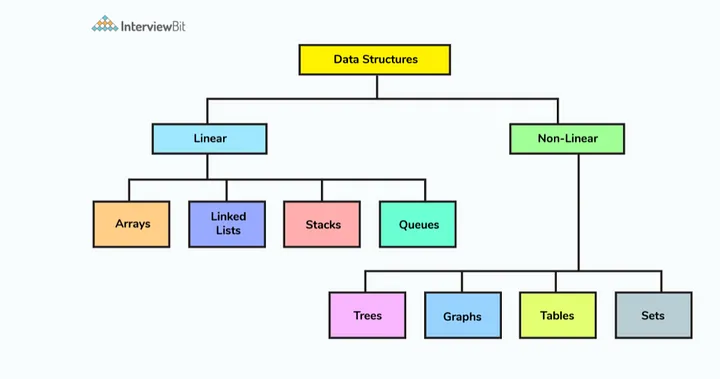
[List of Datasets and Projects to try](https://www.dataquest.io/blog/free-datasets-for-projects/)

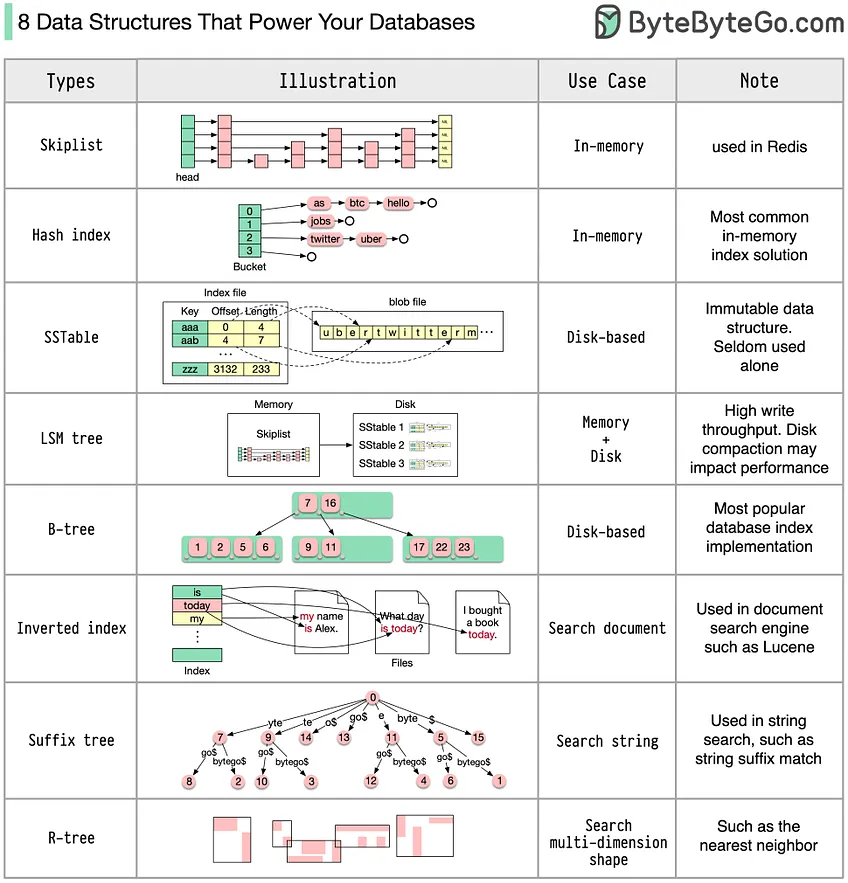
**Choosing dataset:**

* When choosing a dataset/s for the data project, consider the following:
  + What is your major?
  + What are your interests?
  + Have you already done research in a field/on a topic that would be good for data analysis?
  + Can you find a dataset that has lots of information to work with?
  + Do you understand all the elements in the dataset?

[**https://docs.google.com/presentation/d/1h4N5JhDTS8LF2EY4utJniSCch6P9jJYqh4wRjHXlqWI/edit?usp=sharing**](https://docs.google.com/presentation/d/1h4N5JhDTS8LF2EY4utJniSCch6P9jJYqh4wRjHXlqWI/edit?usp=sharing)

* Can work with a partner for data project.
* How do you **get good at coding?**
  + **There’s a gap between what you do in labs and what you do IRL**
  + **We give you a lot of instructions**
  + You’re told what you’re supposed to do with the methods.
* Once you learn to do that, how do you use these concepts to make your own projects?
  + When you do that, you may need to get better by making your own projects?
    - This is a true test
  + When you get stuck when coding, how do you make it?
* How do you know when you make the test, you’re supposed doing the right thing?
  + You need to know the basics of OOP and basic
  + Some people learn the basics here, but they immediately do it to learn their own project, and their foundation is a bit shaky
  + How do you know if you’re doing the right thing?
* The best exercise is to just give you a problem and let you figure it out, but they can’t do that.
  + When you do your own project, you’re not sure where to start, and you get stuck
  + How do you know if your solution is right was the problem I faced?
  + If your solution is works, it works, don’t worry about it.
    - As long as if you aren’t making a project that has to deal with security concerns
* How to get better at code outside of **making a new project and how to** ?
  + Search online to self-learn **and look on Google and make beginner projects and try to do something that**
  + **Once you build tutorials on how to do something, you can look at their code and then get some ideas from youtube, and find the ones that are interesting and useful for you.**
  + **It is an important way to**
* **Since you’ve seen other people have done something like that, you can find out how what is a good way to code.**
  + You can also post online on stack overflow
    - Help you debug
    - Help you give suggestions to solve problems
    - Depending on what your preferences is
* Tricky thing about self-learning
  + How do you know if someone’s suggestions are good?
    - You don’t **you need to consult a CS prof or a very good CS student .**
  + You can also test things yourself, and just **guide them .**
  + **You can get info from this stage, so its important to double-check, and so you know what works or doesn’t work in your eyes.**
  + The best way to get better is to test solutions and get
  + Ask people on campus.
* IT depends on what you’re trying o get
  + Full course or tutorials
  + And you might lack some knowledge on tutorials
  + Look at courses that form institutions form free
  + Don’t always recommend to do buy from courses
    - Should buy courses when you’re looking for something specific.
  + **All kinds of resources on all kinds of courses**
    - **What are the things she recommends?**
      * You will still get used to github is a big plus if you wanna get into coding.
    - **CS 50:** 
      * **Half of it is not in the course**
      * **Cs 50 = harp 151?**
      * **Harp 151 has more classes**
      * Harp 150 and harp 151 are to expose more people to coding, is more realistic to build more projects.
        + Is 151 offered over the summer
    - Would harp 150: is just
      * Cs110 doesn’t have data analysis
      * Data frame and



* + - * She doesn’t mean this?
      * <https://medium.com/codechef-vit/how-to-identify-which-data-structure-to-use-5a1c66ad2742>
  + I think Prof. Mooe is talking about this, Dr. Xie was mad and confused when I told her that CS 110 has data structures in it:
    - 
    - <https://blog.bytebytego.com/p/8-data-structures-that-power-your>

**Harp 150: Week 14: Intro to Data Viz**

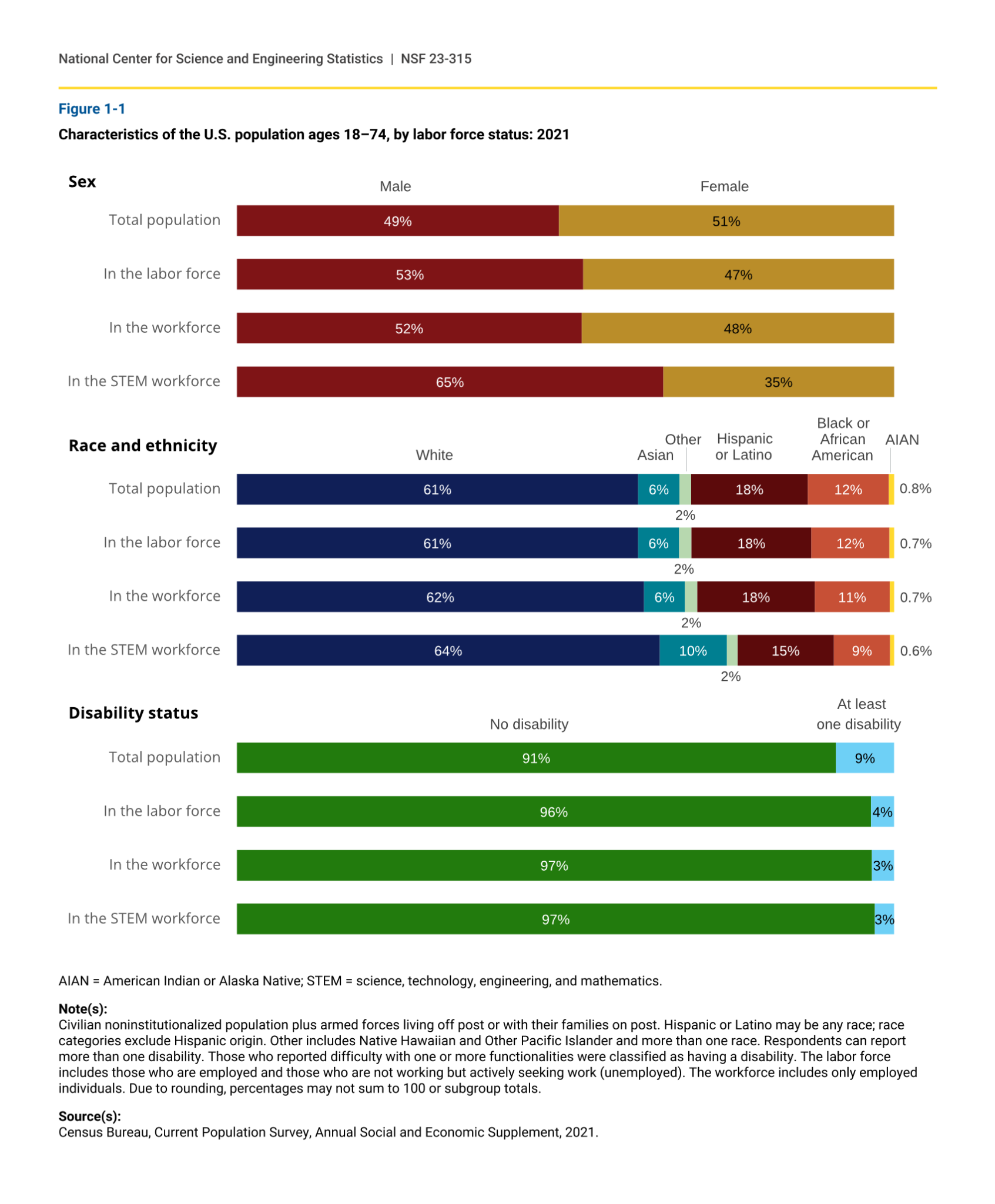
* <https://docs.google.com/presentation/d/1JYMf24iAvWGkWv8CUYF7bZC6NpuNr9YI3rRCFB4sb_Q/edit?usp=sharing>

Misleading graphs come in all kinds of forms:

Discuss -

Which part(s) is misleading?

What information is left out?



Let’s explore [the issue of diversity in the field of STEM](https://ncses.nsf.gov/pubs/nsf23315/report/introduction) (focus on visualizations):

1. What’s your spontaneous insight?
2. What’s your conclusion after exploring all the visualizations?

Things that stand out:

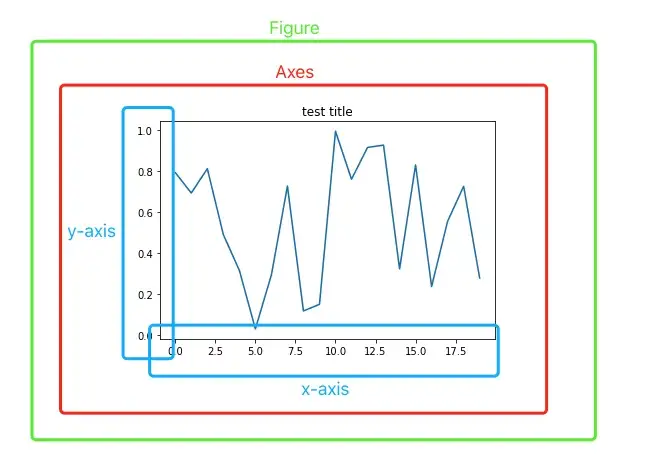
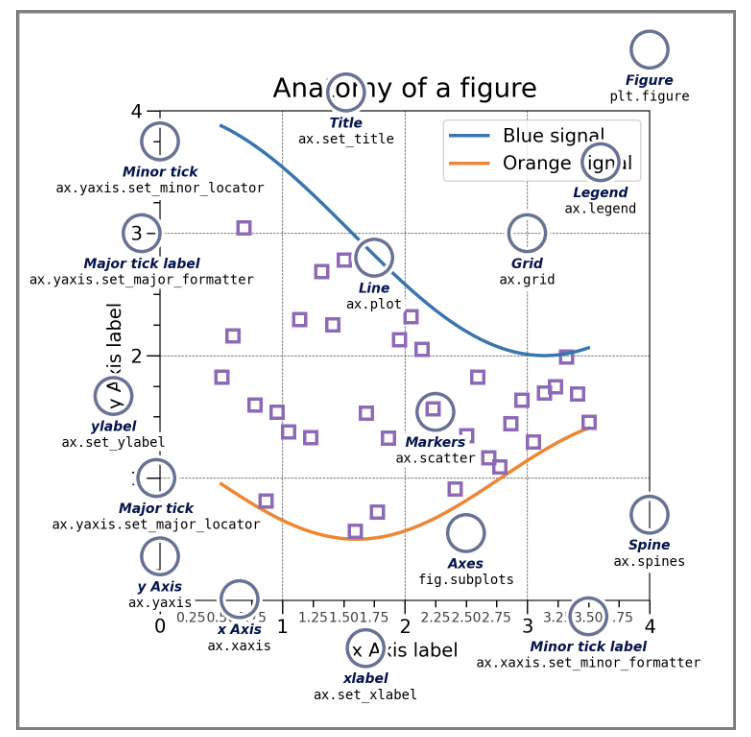
* There is a equal representation in the total population, labor force, and the workforce.
* There are more men in the STEM workforce than women in the STEM workforce
* White people are the major force in the population, labor force, workforce, and STEM workforce.
  + Minority groups are less represented in the total population, labor force, workforce, and STEM workforce ‘’
* **You can also see that the gender gap is bigger than the race and ethnicity gap**

**After looking at all of the graphs on the website, what did you find:**

**In pepole who have associate’s degeree recipents**

**There is a greater race discrepancy in certain degrees instead of**

**Structure of a matplotlib graph:**

* 
* 
* All of the things that are circled have meaning and show something about the data/chart

**Simple Process:**

1. **Which kinds of visualization may be suitable for your purpose:** [**Data Visualization Catalogue**](https://datavizcatalogue.com/)
   1. Good for creating the visualizations that work
2. **How to make those kinds of visualization using matplotlib with examples:** [**Plotting Methods Overview**](https://matplotlib.org/stable/gallery/index.html) 
   1. Good for making the actual code behind everything

Almost everyone doesn’t have a disability in the population, labor force, workforce, and STEM workforce.

**Plotting methods:**

* [**https://matplotlib.org/3.5.3/api/\_as\_gen/matplotlib.pyplot.html**](https://matplotlib.org/3.5.3/api/_as_gen/matplotlib.pyplot.html)
* [**https://matplotlib.org/stable/gallery/lines\_bars\_and\_markers/simple\_plot.html#sphx-glr-gallery-lines-bars-and-markers-simple-plot-py**](https://matplotlib.org/stable/gallery/lines_bars_and_markers/simple_plot.html#sphx-glr-gallery-lines-bars-and-markers-simple-plot-py)

**4/30/2024: Week 16: Last Day & Course Evaluation**

* <https://docs.google.com/presentation/d/1Ra08aCFKI33vxA_4lESQ8YEnII3d8Gq9yYJydEsLyVo/edit?usp=sharing>
* Some guy donated 1 mill
  + Works at google
* What if you don’t take harp 151
  + She said I can be a TA in the future. If i show her other means of competency
  + Harp 151: does look at the different ways you can
* Everything must be submitted by may 16th.
* binghamton alumni donates 1 million dollars to support binghamton codeshttps://www.binghamton.edu/news/story/4821/binghamton-university-set-to-receive-5-million-landmark-gift-to-pursue-ai-research-and-development
  + <https://www.binghamton.edu/news/story/4890/new-york-state-announces-1.2-million-in-nsf-funding-for-binghamton-university-to-bring-in-stem-teachers>