

# WELCOME TO CSCI E-33A WEB PROGRAMMING W/ PYTHON AND JAVASCRIPT SECTION 2 PYTHON/DJANGO

TF FOR THIS SECTION: GLENN LANGDON

GLL939@G.HARVARD.EDU

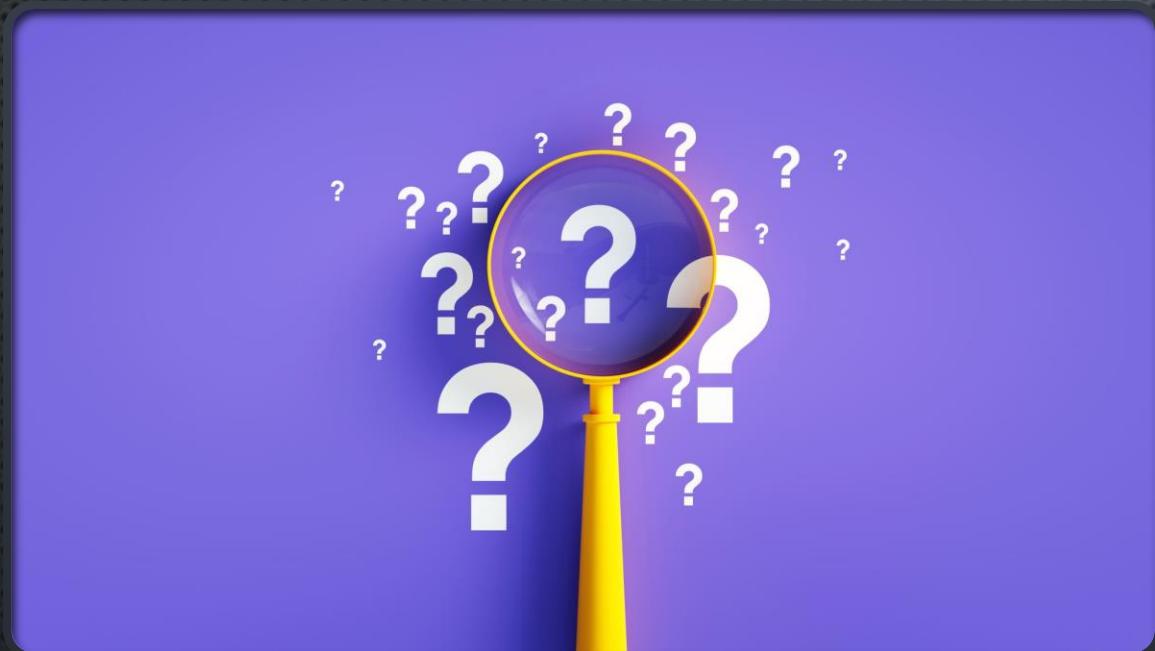




# PROGRAM FOR TODAY

- GRADING ON 3 AXES FOR REMAINING PROJECTS
- IDENTIFYING STYLING ERRORS
- PROGRAMMING IN PYTHON
  - METHODS AND TECHNIQUES USEFUL IN THIS COURSE
- DJANGO
  - NAVIGATING A DJANGO APP

# QUESTIONS?



- ANYTHING FROM PREVIOUS SECTION
- ABOUT ASSIGNMENT 0, SEARCH
- ABOUT THE COURSE



WHEN CAN I  
SEE THE GRADE  
FOR MY  
ASSIGNMENT?



# STARTING WITH THIS PROJECT WE'RE GRADING ALONG 3 AXES

- CORRECTNESS (x3)
- DESIGN (x2) (IMPORTANT! THIS SCORE IS ADDITIVE, NOT SUBTRACTIVE)
- STYLE (x1)



## SPEAKING OF STYLE

- STYLE POINTS CAN BE SOME OF THE EASIEST TO EARN FOR EVERY PROJECT
- IN THE CHAT SECTION IS A LINK TO A PYTHON FILE, DOWNLOAD IT
- IN BREAKOUT ROOMS, INTRODUCE YOURSELF TO YOUR CLASSMATES
- IDENTIFY AND COUNT THE NUMBER OF STYLE ERRORS IN THE FILE (DO NOT USE ANY SOFTWARE OR ONLINE AIDS, IT'S A LEARNING EXPERIENCE FOR ALL)
- DESIGNATE A CLASSMATE AS A SPOKESPERSON WHO WILL SPEAK FOR THE GROUP WHEN WE RETURN FROM THE BREAKOUT ROOM
- THIS IS NOT A TEST! ENJOY THE EXPERIENCE.



# TO THE RESCUE: PYCODESTYLE

A tool to check python code against standardized style conventions

\$ PIP INSTALL PYCODESTYLE

FINAL WORD ON STYLE.  
IT IS GENERALLY  
RECOMMENDED THAT  
THERE ARE:

- **TWO BLANK LINES** BEFORE AND AFTER TOP-LEVEL FUNCTION AND CLASS DEFINITIONS.
- **ONE BLANK LINE** BETWEEN METHOD DEFINITIONS WITHIN A CLASS.
- USE BLANK LINES SPARINGLY TO SEPARATE GROUPS OF RELATED STATEMENTS, LIKE INSIDE A FUNCTION OR METHOD
- NO BLANK LINES BETWEEN SIMPLE, CONSECUTIVE STATEMENTS WITHIN THE SAME BLOCK.



# PYTHON REVIEW





IN THE CHAT IS A  
LINK TO SOME  
DATA STRUCTURES  
I'LL BE USING FOR  
THE FOLLOWING  
EXAMPLES

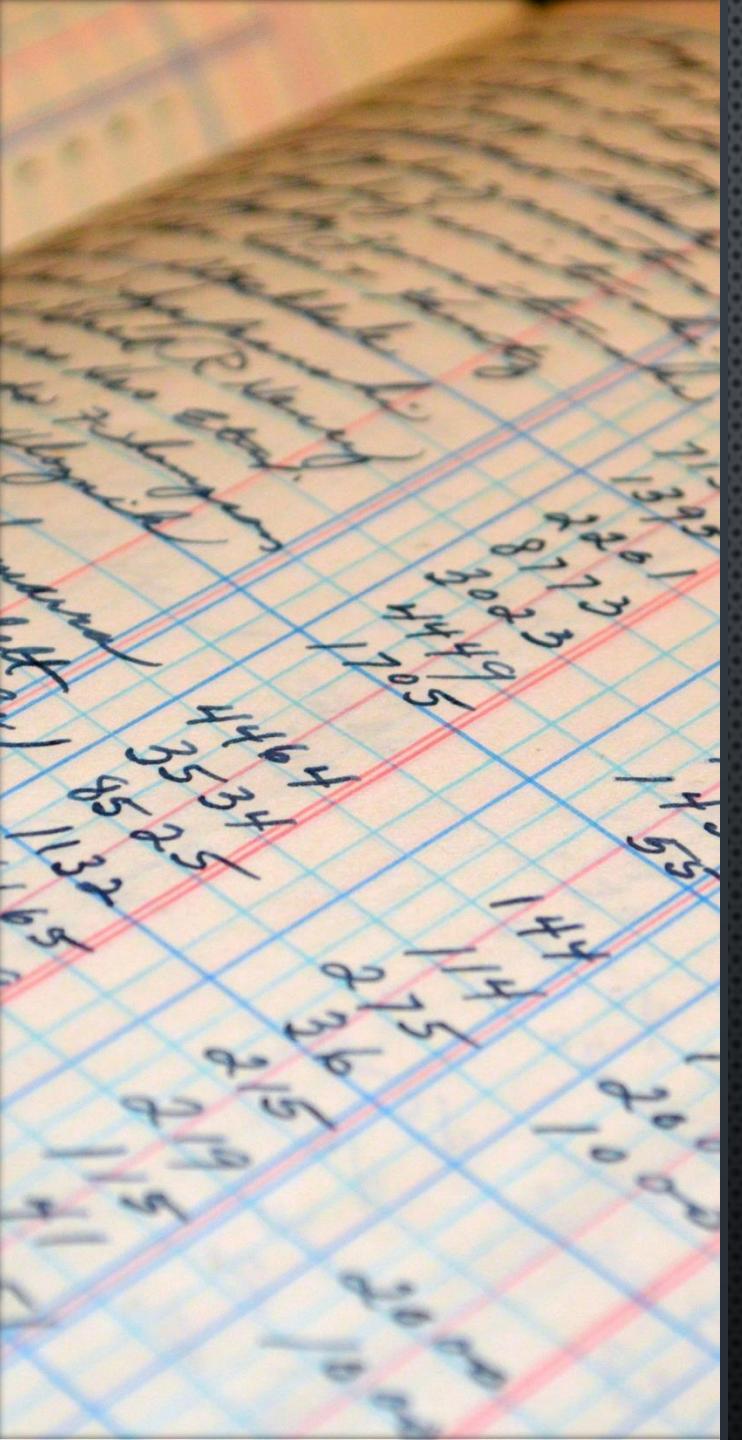


# WORKING IN PYTHON

- LISTS
- LIST COMPREHENSION
- DICTIONARIES
- TUPLES
- LAMBDA

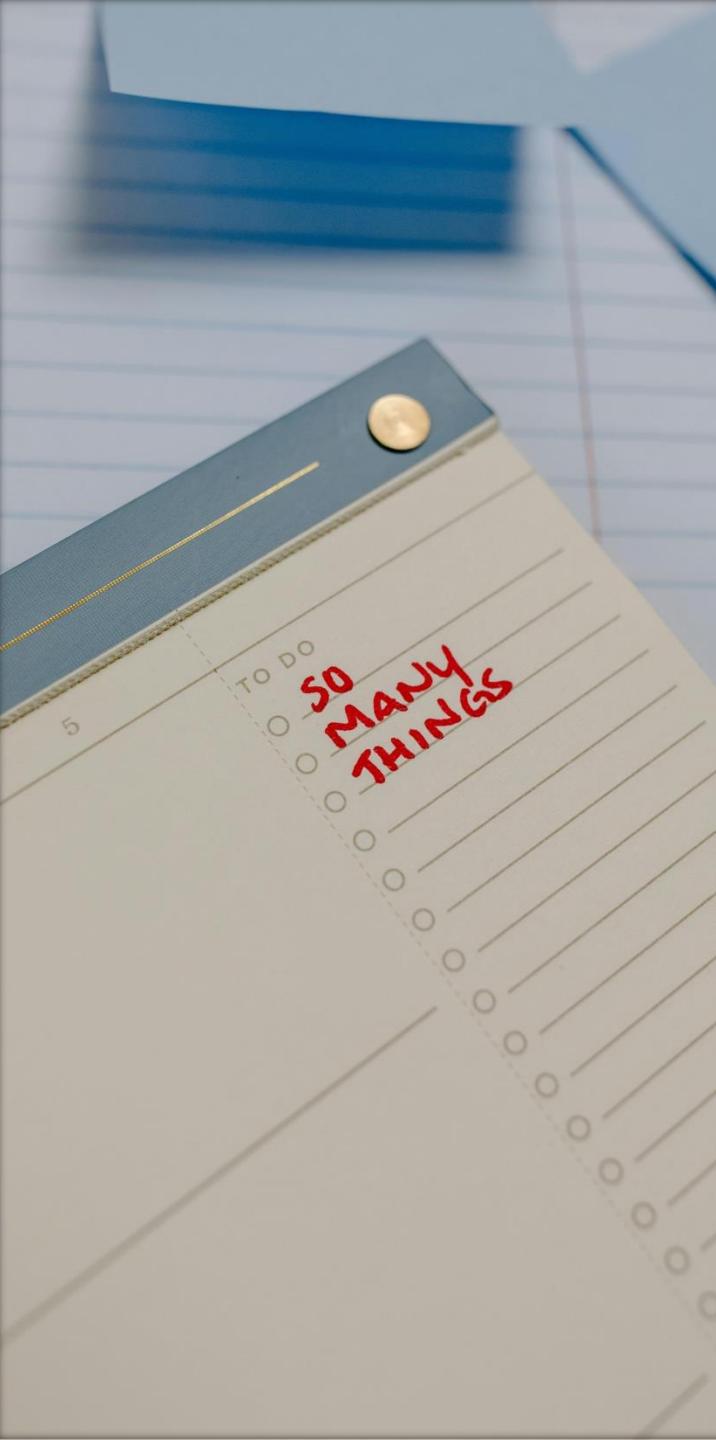
# PYTHON

- WE'RE GOING TO BRIEFLY REVIEW WHAT BRIAN HAS DEMONSTRATED IN THE PREVIOUS LECTURE  
AND
- GO A LITTLE FURTHER WITH SOME EXAMPLES THAT WILL SERVE YOU AS YOU GET INTO YOUR PROJECTS.



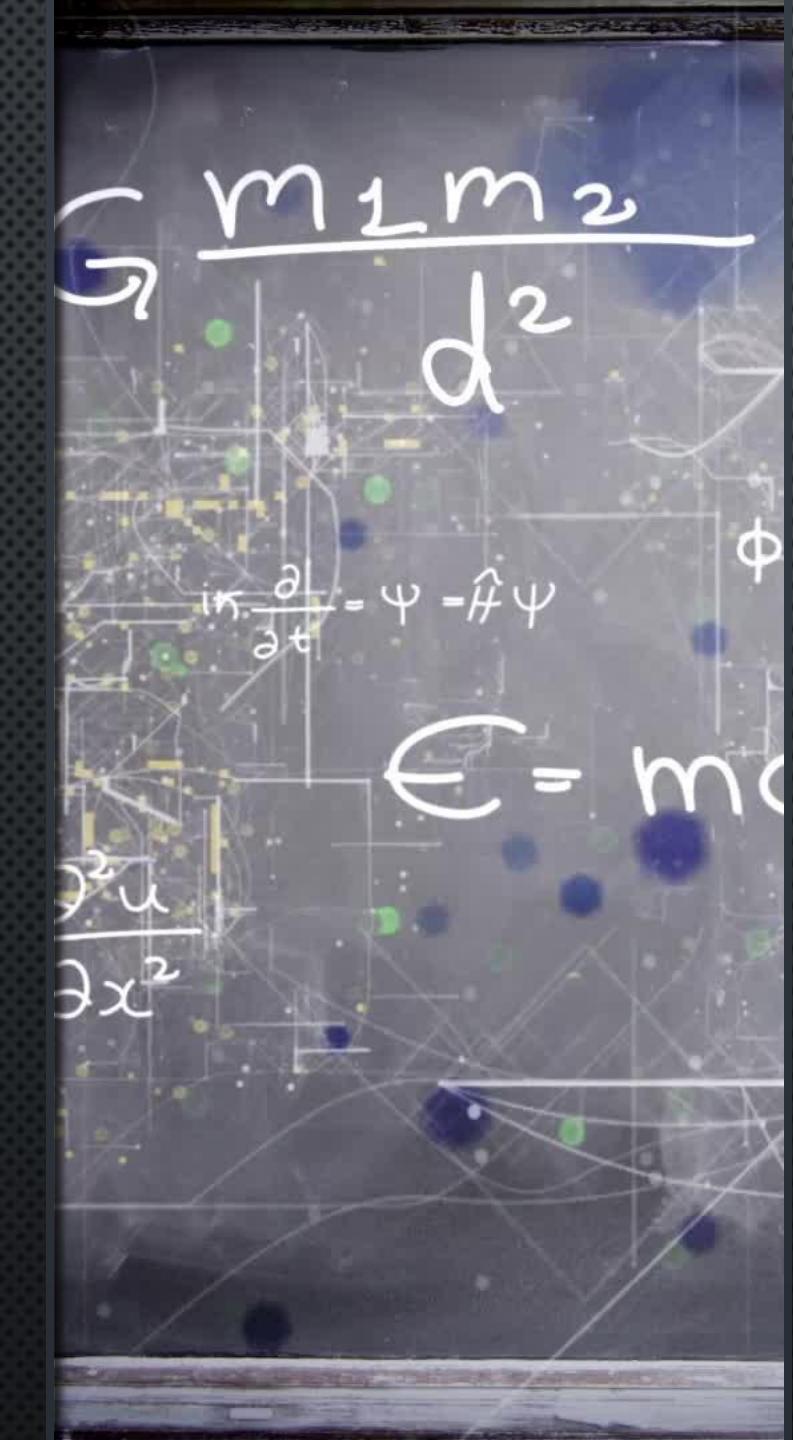
# LISTS

- A PYTHON LIST IS A BUILT-IN DATA STRUCTURE THAT IS MUTABLE, ORDERED, AND ALLOWS FOR THE STORAGE OF A COLLECTION OF ITEMS.
- LISTS CAN CONTAIN ELEMENTS OF DIFFERENT DATA TYPES, INCLUDING NUMBERS, STRINGS, OTHER LISTS, AND MORE.
- THEY ARE DEFINED BY ENCLOSING A COMMA-SEPARATED SEQUENCE OF ELEMENTS IN SQUARE BRACKETS --- [ ].
- HERE ARE SOME LIST METHODS...



# LIST COMPREHENSION

- PYTHON LIST COMPREHENSIONS PROVIDE A CONCISE WAY TO CREATE LISTS.
- THEY ARE A PYTHONIC WAY TO APPLY TRANSFORMATIONS AND FILTERS TO SEQUENCES, SUCH AS LISTS, TUPLES, OR STRINGS, TO GENERATE NEW LISTS.
- LIST COMPREHENSIONS ARE OFTEN MORE READABLE AND EFFICIENT THAN TRADITIONAL METHODS OF ITERATING AND APPENDING TO LISTS.

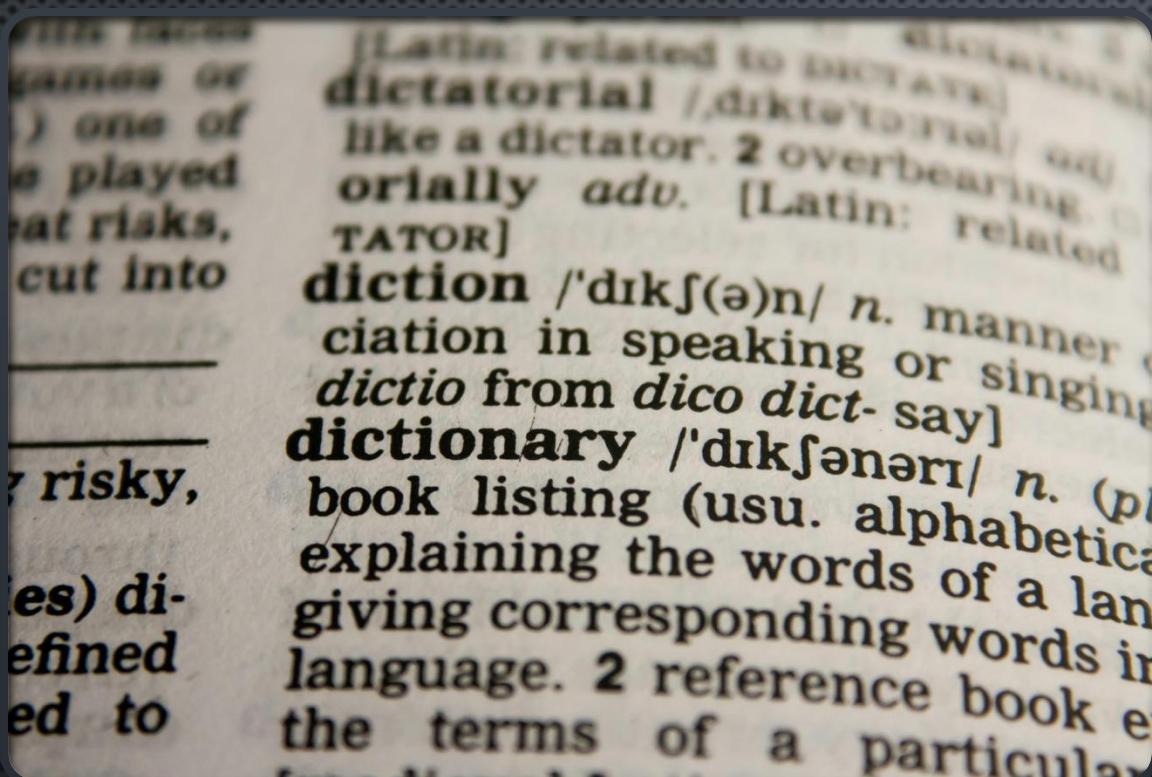


# LIST COMPREHENSION EXPLANATION

- **[EXPRESSION FOR ITEM IN ITERABLE (IF CONDITION)]**
- 1) EXPRESSION: EXPRESSION TO EVALUATE AND APPEND TO THE NEW LIST
- 2) ITEM: THE VARIABLE REPRESENTING EACH ELEMENT IN THE ITERABLE (E.G., A LIST, TUPLE, OR STRING).
- 3) ITERABLE: THE SEQUENCE OR ITERABLE TO ITERATE OVER
- 4) CONDITION (OPTIONAL): AN EXPRESSION TO FILTER ELEMENTS BASED ON A CONDITION.

# DICTIONARIES

- IN PYTHON, DICTIONARIES ARE DEFINED USING CURLY BRACES {} AND CONSIST OF KEY-VALUE PAIRS SEPARATED BY COLONS ::
- KEYS ARE UNIQUE WITHIN A DICTIONARY.
- VALUES CAN BE OF ANY DATA TYPE.
- DICTIONARY ITEMS ARE MUTABLE AND DYNAMIC
- CAN BE NESTED
- HERE ARE SOME METHODS FOR PYTHON DICTIONARIES...



# TUPLES

- MOST IMPORTANTLY, HOW DO I PRONOUNCE THIS PYTHON NATIVE TYPE?
- TUPLES ARE IMMUTABLE, AS OPPOSED TO LISTS
- A SEQUENCE OF VALUES
- **USE LISTS** WHEN YOU NEED A COLLECTION THAT CAN CHANGE OVER TIME.
- **USE TUPLES** WHEN YOU NEED A COLLECTION THAT SHOULD REMAIN CONSTANT AND NEED TO ENSURE ITS INTEGRITY.





# LAMBDAS

- A LAMBDA IS A SMALL ANONYMOUS FUNCTION (A FUNCTION WITHOUT A NAME)
- IT CAN TAKE ANY NUMBER OF ARGUMENTS, BUT ONLY HAS ONE EXPRESSION
  - ARGUMENT IS THE PLACEHOLDER
  - EXPRESSION IS THE CODE YOU WANT TO EXECUTE
- IN THE COMING PROJECTS YOU WILL BE ABLE TO MAKE YOUR CODE MORE READABLE BY UTILIZING LAMBDAS IN CERTAIN SPECIFIC CASES
- HERE ARE SOME EXAMPLES...

# THE BASIC STRUCTURE OF ANY DJANGO APP

A DJANGO PROJECT CAN HAVE MULTIPLE APPS

# DIRECTORY STRUCTURE OF A DJANGO PROJECT

# DIRECTORY STRUCTURE OF A DJANGO PROJECT WITH AN APP

```
myproject/
|-- myproject/
|   |-- boards/                                <-- our new django app!
|   |   |-- migrations/
|   |   |   +-- __init__.py
|   |   |   |-- __init__.py
|   |   |   |-- admin.py
|   |   |   |-- apps.py
|   |   |   |-- models.py
|   |   |   |-- tests.py
|   |   |   +-- views.py
|   |   |-- myproject/
|   |   |   +-- __init__.py
|   |   |   |-- settings.py
|   |   |   |-- urls.py
|   |   |   |-- wsgi.py
|   |   +-- manage.py
+-- venv/
```

# A WORD ON VIRTUAL ENVIRONMENTS (VENV)

Virtual environment: consists of 1) the python interpreter that the environment runs on and 2) a folder containing 3<sup>rd</sup> party packages

Virtual environments create isolated contexts to keep dependencies required by different projects separate from one another, so they don't interfere with each other.

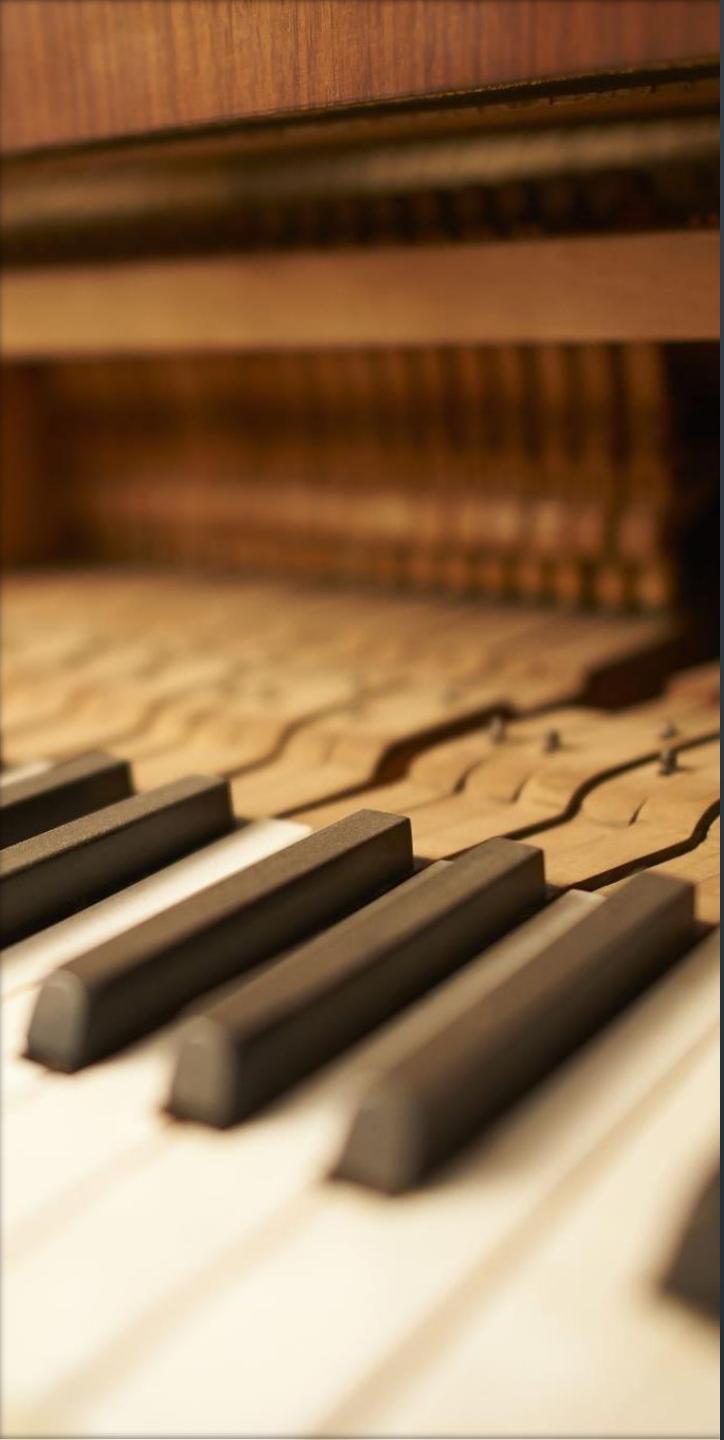
Python can only run one version of a dependency at a time

All virtual environments are separated from each other and can contain different versions of python and different versions of packages

No limitations on the number of environments one has as they are just folders containing the python interpreter and python packages

# DO I NEED TO USE A VIRTUAL ENVIRONMENT FOR THIS CLASS?

- IN A WORD, NO
- SINCE THEIR IMPORTANCE DEPENDS ON ISOLATING DIFFERENT VERSIONS OF THE SAME PACKAGE ON THE SAME COMPUTER, IT IS DOUBTFUL THAT YOU WILL RUN INTO ANY DEPENDENCY CONFLICTS DURING THE SEMESTER
- HOWEVER, IT IS CONSIDERED BEST PRACTICE AND IS SOMETHING YOU SHOULD BE AWARE OF IF YOU CONTINUE IN YOUR PYTHON JOURNEY AFTER THIS CLASS.



# DJANGO

- WE'RE GOING TO LOOK AT A PIANO INVENTORY APP. WHICH WILL STORE PIANOS IN INVENTORY AND ALLOW FOR ADDITIONS AND DELETIONS FROM THE INVENTORY.
- THIS APP HAS MANY FEATURES SIMILAR TO WHAT YOU'RE GOING TO BE BUILDING THIS WEEK.
- LET'S HAVE A LOOK!



## PROJECT1 SEARCH

- WE'VE COVERED PRETTY MUCH EVERYTHING YOU'LL NEED TO KNOW (AND MORE) TO FULFILL THE SPECIFICATIONS FOR THIS WEEK'S PROJECT.
- IF YOU HAVE ANY QUESTIONS, DON'T HESITATE TO REACH OUT ON ED OR ME DIRECTLY. MY EMAIL LISTED ON THE COURSE WEBSITE.

A photograph of two flutes filled with champagne. One flute has a black ribbon wrapped around its middle with a small gold star. The background is blurred with warm, glowing lights.

# CONGRATULATIONS

- WE'VE COVERED A LOT!
- ADDING VIEWS, ADDING ROUTES TO FOR YOUR VIEWS, PASSING THE VIEW CONTEXT TO A TEMPLATE AND DISPLAYING THAT INFORMATION IN THE TEMPLATE WILL BE YOUR BREAD AND BUTTER THIS SEMESTER
- THE MORE COMFORTABLE YOU ARE WITH THE PIPELINE IN DJANGO, THE BETTER YOU'LL BE ABLE TO EFFICIENTLY DO YOUR ASSIGNMENTS