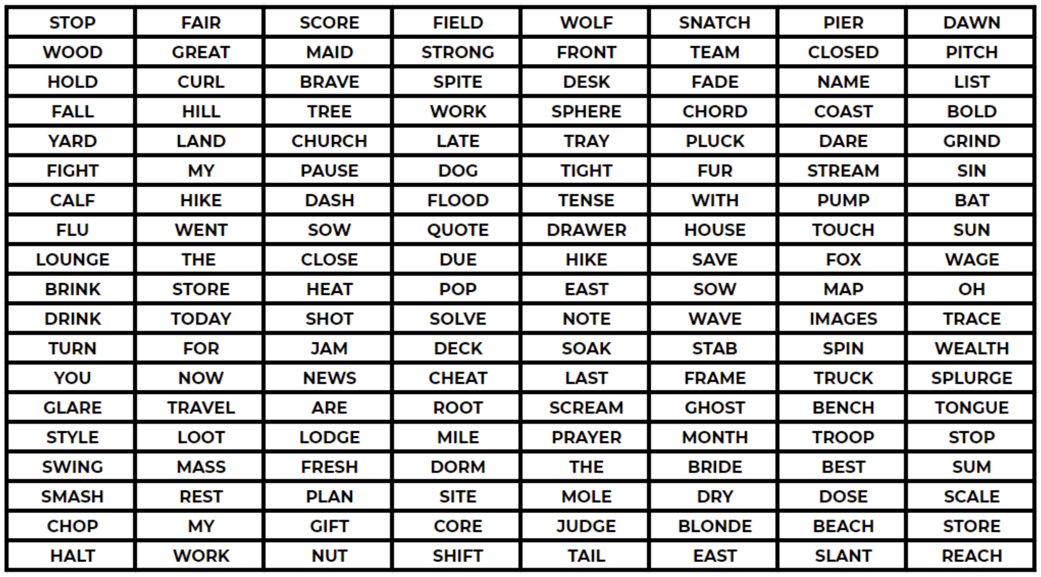
In-class 3

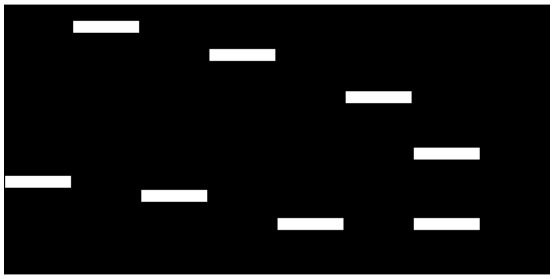
There are two images we will be working with:

- word_matrix.png
- mask.png

The word_matrix is a .png image that contains a spreadsheet of words with a hidden message in it. Your task is to use the mask.png image to reveal the **hidden message** inside the word_matrix.png.



STOP	FAIR	SCORE	FIELD	WOLF	SNATCH	PIER	DAWN
WOOD	GREAT	MAID	STRONG	FRONT	TEAM	CLOSED	PITCH
HOLD	CURL	BRAVE	SPITE	DESK	FADE	NAME	LIST
FALL	HILL	TREE	WORK	SPHERE	CHORD	COAST	BOLD
YARD	LAND	CHURCH	LATE	TRAY	PLUCK	DARE	GRIND
FIGHT	MY	PAUSE	DOG	TIGHT	FUR	STREAM	SIN
CALF	HIKE	DASH	FLOOD	TENSE	WITH	PUMP	BAT
FLU	WENT	SOW	QUOTE	DRAWER	HOUSE	TOUCH	SUN
LOUNGE	THE	CLOSE	DUE	HIKE	SAVE	FOX	WAGE
BRINK	STORE	HEAT	POP	EAST	SOW	MAP	OH
DRINK	TODAY	SHOT	SOLVE	NOTE	WAVE	IMAGES	TRACE
TURN	FOR	JAM	DECK	SOAK	STAB	SPIN	WEALTH
YOU	NOW	NEWS	CHEAT	LAST	FRAME	TRUCK	SPLURGE
GLARE	TRAVEL	ARE	ROOT	SCREAM	GHOST	BENCH	TONGUE
STYLE	LOOT	LODGE	MILE	PRAYER	MONTH	TROOP	STOP
SWING	MASS	FRESH	DORM	THE	BRIDE	BEST	SUM
SMASH	REST	PLAN	SITE	MOLE	DRY	DOSE	SCALE
CHOP	MY	GIFT	CORE	JUDGE	BLONDE	BEACH	STORE
HALT	WORK	NUT	SHIFT	TAIL	EAST	SLANT	REACH



Suggestions:

- You may need to make changes to the mask.png for this to work because the mask is currently not the desired size.

- You may need to change the transparency of the mask when it lays over the word matrix. Transparency and masking documentation can be found here: https://pillow.readthedocs.io/en/stable/

Requirements:

As output, you will need to provide/show the following:

1. An image showing mask properly overlay the word matrix to reveal the hidden message.
2. Write out the **hidden message** in a markdown box.

Submit your file as a finished Jupyter Notebook.

The grading rubrics is as follows:

Criteria	**Full Marks (100%)**	**High Partial Marks (75%)**	**Middle Partial Marks (50%)**	**Low Partial Marks (25%)**	**No Marks (0%)**
Functionality and Correctness (50 points)	The code successfully generates expected outputs.	The code mostly works but misses a few outputs or occasionally encounters errors that do not significantly impact the overall functionality.	The code partially works, but fails to generate all expected outputs or frequently encounters errors.	The code has limited functionality, with significant issues.	The code does not function correctly; it fails to the expected outputs.
Code Efficiency and Quality (20 points)	The code is highly efficient, with no redundancy, and follows best practices in Python programming.	The code is mostly efficient but contains some unnecessary parts or could be optimized for better performance.	The code works but is inefficient or does not follow best practices, making it less effective or slower than necessary.	The code has significant inefficiencies or poor practices that heavily impact its performance and effectiveness.	The code is inefficient, poorly written, and does not adhere to basic programming practices.
Error Handling and Data Validation (20 points)	The code includes robust error handling and data validation to gracefully manage exceptions and ensure only images are saved.	The code includes some error handling and data validation, but there are minor gaps that could be improved.	The code has minimal error handling or data validation, leading to potential issues with exception management or incorrect file saving.	The code lacks sufficient error handling and data validation, resulting in frequent errors or inappropriate file saving.	The code does not include error handling or data validation, making it prone to failure and incorrect data handling.
Documentation and Code Readability (10 points)	The code is excellently documented with clear, concise comments explaining each major step. Variable names are meaningful, enhancing readability.	The code is well-documented with good comments and variable names, but there are sections that could be clearer or more descriptive.	The code has some documentation and readability efforts, but comments are sparse, or variable names are not consistently meaningful.	The code has minimal documentation or effort toward readability, making it difficult to understand the logic and flow.	The code lacks documentation and readability, with no comments and poor variable naming, making it challenging to follow.