Group Name: Dothraki
Group URL: https://github.com/JoeSGomes/animated-meme.git
Group Commit: 48b62fb0c52a0f3ad7a92d898884b44d061b3140
Group File:

- Joe_check-in1.py
- Kishan_check-in1.py → used jeopardy.txt file to run the class and make sure it works
- Walter_check-in1.py
- naila_check-in1.py
- ariana_check-in1.py

Kishan S:

Problem: I had to try to figure out how to take all the subject, points,question, answer values from the "jeopardy.txt" file and sort them into dictionaries that would be accessible in the jeopardy game

Solution: I made a JeopardyCatalog class that will open up the txt file with the information, and I converted it so that it sorts the questions by category (either History, Pop Culture, Or Math). Then, inside those dictionaries, I made another dictionary, where the key is the specific points value assigned to each question-answer combination, and the value for that key is a tuple of (question, answer). This way, the information is easily accessible to be extracted into the future methods that are made (such as get_question(), get_answer() and get_points().

Which file contains your solution and any instructions on how to run the solution:
File with Solution: kishan_check-in1.py (supplemental → need jeopardy.txt to run code)

How to run Solution:
You will need the jeopardy.txt file to run this method
In the vscode terminal, just type "python3 kishan_check-in.py" and it will print out the dictionary that took the information from the jeopardy.txt file. This dictionary will be used in future methods as was described in the "solution" explanation section.

Ariana S:

Problem: The major problem I encountered was figuring out how to display the words and the questions. At first, I ran all the words at once and asked the questions and the word recitation prompts after all of them were displayed. However, this would cause a major problem with the user as it would have been better to display the words first and then have them answer questions afterwards. However, I did not find a way to do this that would have been done efficiently.

Solution:

Which file contains your solution and any instructions on how to run the solution:
File with a solution: ariana_check-in1.py

How to run Solution:

1. Open up ariana_check-in1.py.
2. Open a terminal.
3. In the terminal, write “python3 ariana_check-in1.py”
4. Run this

Naila N:

Problem: The problem I had to figure out was the logic of how many tries and hints each player can use while guessing a number. This was dependent on how many times a player had to guess the right number. If they run out of tries, they can guess to get close to the correct number.

Solution: The way I approached this problem was that there were 10 tries and 3 hints in total for the game. If a player uses all the hints, they don’t have any hints left and then they can use the guesses. The computer will choose a number within a certain range using randint(). If correctly guessed, it will give a message "Correct guess! You won!" if not it will print incorrect guess.If a player runs out of tries it will show them that "You ran out of tries; better luck next time!".

Which file contains your solution and any instructions on how to run the solution:
File with a solution: naila_check-in1.py

How to run Solution:
Step 1: Open the naila_check-in1.py.

Step 2: Open up a terminal.
Step 3: Type in “Python3 naila_check-in1.py”
Step 4: Click enter and run it.

Walter G:

Problem: I had to figure out how to get hints to pop out for the player if in case the player had a hard time guessing the number. I also had to figure out how many hints the player would be allowed to use without it being a handicap.

Solution: What I figured would be the best way to execute these hints was to add a counter for each time the player would like a hint. Once the player reaches the third hint, they will be prompted with the third hint every time after the player asks for a hint again. The total amount of hints is 3. I will be making a play_game method where the question of ‘do you need a hint (y/n)’ will pop out before every guess the player makes.

Which file contains your solution and any instructions on how to run the solution:
File with a solution:Walter_check-in1.py

How to run Solution:
Step 1: Open the Walter_check-in1.py file.
Step 2: Open up a terminal.
Step 3: Type in ‘Python3 Walter_check-in1.py’
Step 4: Click enter and run it.

Joe G:

Problem: I was set out to explore how to show the available questions left for each category. That dictionary will hold all of the key values of the questions for the jeopardy game and once the value has been chosen, it shows the most updated dictionary. The problem was how can we display the user with the most updated available questions.

Solution: The way I tackled this was I created an empty list and in the list I extracted out the key values which will be the points (500, 400, 300, 200, 100). The values then will be appended to the new list and returned using an f-string corresponding to the specific subject.

Which file contains your solution and any instructions on how to run the solution:
File with a solution: Joe_check-in1.py
How to run Solution:
Step 1: Open the Joe_check-in1.py.

Step 2: Open up a terminal.
Step 3: Type in “Python3 Joe_check-in1.py”
Step 4: Click enter and run it.