Grade 11 Assignment #6 Java Script

Whale Talk

Take a sentence like, 'Turpentine and turtles,' and translate it into its whale talk equivalent: 'UUEEIEE A UUEE'.

There are a few simple rules to translating text to the *whale language*:

1. There are no consonants. Only vowels.
2. The 'u's and 'e's are extra long, so we must double 'u's and 'e's.

Once we have converted text to the whale language, the result is sung slowly, as is a custom in the ocean.

To accomplish this translation, we can use our knowledge of loops. Let's get started!

Tasks:

There are 16 Tasks to complete for this assignment:

1. Using NotePad++, create the code needed for the program called Whale Talk.

2. Create a variable named input, that is equal to any sentence. This variable will contain the text we will translate into whale language.

3. We will also need an array of letters that we are going to look for in the input variable.

   Since whale language only contains vowels, create an array with every vowel inside of it and set it equal to a variable named vowels.

4. Lastly, later when we find a vowel in the input string, we will need somewhere to store it.

   Create a variable named resultArray, and set it equal to an empty array: [].

5. Now that we have an input sentence, and the vowels we're looking for, we need to write logic that compares the input variable's text to the vowels array. Our goal is to find all the vowels in the input string.

   We can do this by going letter-by-letter and checking to see if the first letter of the input is a vowel, then if the second letter is a vowel, and so on...

   To do this, let's utilize a for loop to loop through every letter of the input string.

   Write a for loop that starts at var i = 0, stops at i < input.length, and iterates by i++.

6. Inside of the for loop you just wrote, write another for loop. This loop's purpose is to iterate through the vowels array. Each time the outter for loop runs, the inner for loop will run completely.

This will enable us to check the first letter of input against all the vowels items, then the second letter of input against all the vowels items, etc.

Since we have two for loops, set this for loops iteration variable as j. Start at var j = 0, end at j < vowels.length, and go up by one each time with j++.

7. Inside the second for loop's block, we can write the logic to compare the input letter to every letter in the vowels array.

   Write a if/else statement that checks if input[i] is equal to vowels[j]. If it is, then use push to add input[i] onto the resultArray.

   This will only push the characters from input that are vowels to the resultArray.

8. Now we need to double the 'e's and the 'u's that are found.

   In the first for loop, outside of the inner for loop's block, write an if/else if/else statement that checks if input[i] is equal to 'e'. If so, push input[i] to the resultArray.

9. Repeat step 7, but this time with the letter 'u'.

   Just like in step 7, we need to double a letter, but this time it should be the letter u.

   Modify the if statement to also push the letter u if it matches.

10. At the bottom of the program, use console.log to print the resultArray.

11. Notice the output has commas between each letter. That's because we are printing an array.

    This is ok, but we also want to print out to the screen the word formed in whale language. To put the array items together as a string, we can use this code:

    ```
    resultArray.join('').toUpperCase();
    ```

    Since resultArray is an array, we can make it back into a continuous string with .join('').

    Also, we think whales are pretty enthusiastic, so we also included the toUpperCase() function, which will take any string and make it uppercase.

12. Now, run the program and sing your translation!

    To confirm, if you use the input: 'Turpentine and turtles', the whale version would read: 'UUEEIEEAUUEE'. Say that three times fast!

13. However, you are not finished yet. You need to make your program user friendly by printing-out user friendly messages to the screen welcoming the user and asking them to try out the program while informing them the point of the program and what the output on the screen they see means.

14. It would be helpful to describe to other developers what this small Java program does. Write some comments that describes what this program does.

    a. Use multi-line comments to (/* comment in the middle of */):
        i. Write one at the top of the code (before the main part of the code) that gives a quick intro/description the assignment.
        ii. Below that comment, write one that summarizes the program.
    b. Use a single line comment to (//then comment):
        i. Create 5 comments anywhere you deem necessary or important in the code. Remember the comment is to highlight or explain what is going on or what is being done in a particular way or used and why.
        ii. Ideas for your comments above would be to explain: the for loops, the use of is statements, the variable declarations or the printed out statements to the screen.
        iii. Identify the last line of code in every function or statement (anything that ends with a curly bracket { }) by writing "End of BLAH-BLAH".
        iv. Identify the end of the program.

15. Once your program is complete, make sure to test it using the console (do not submit a program that does not work).

16. Lastly, upload (drag and drop) your assignment to the portal. Look for your name under the Assignment 6 webpage.