**Assignment 6**

Your task is to implement a word guessing app using JavaScript.  The program will 'think' of a word and display it as a row of dashes, with each dash representing a letter in the word.  The user will try to guess the word, by suggesting letters, one letter at a time.   If the user suggests a letter which occurs in the word, that letter has to be displayed within the dashes in its correct positions.

The following screencast shows one implementation of the game.  Your solution has to implement the same logic.



When the web page is loaded, a word is displayed as a series of dashes and the user may enter a letter and press GUESS to play.

When the user guesses the full word, a winning message is displayed.  The user has then to restart the game to get a new word to guess.

After 10 wrong guesses, a message is displayed with the right answer.  The user has to press RESTART to begin playing with a new word.

At any point, the user must be shown all the letters they have already tried.  The correct ones are displayed in the dash-display but they also need to see the wrong ones.  In addition, **they need to have some representation of how many more guesses they have.**

If the user presses GUESS without entering a letter, nothing happens.

If the user enters a correct letter that has already been entered and is in the display, nothing happens.

If the user enters a wrong letter that has been entered before, it is added to the wrong letters (it will show twice) and the count of wrong guesses is incremented.

The game is case insensitive:  letters may be entered in lower case or upper case.

The user may choose to restart the game at any point.  The web page is then reinitialized and a new word is displayed as a series of dashes.

Your program will get the words from a list (array of answers).  The program will cycle through the words in that list, using one after the other. When it runs out, it will start back from the beginning of the list.  Do NOT choose the answer randomly.

You will use the following answersList array in your program but **don’t make any assumptions as to its length or the words it contains.** You may assume its name is game.answersList and it is an array. Your program has to work for any array of answers.

game.answersList = ['JavaScript', 'document', 'element', 'ajax', 'property',

                            'event', 'propagation', 'listener', 'transition', 'animation']

The html source document  and css stylesheet that were used in the screencast demo are available under Resources.  A template JavaScript file guess.js file is also available under Resources.

To submit your assignment, upload your**JavaScript program guess.js, your html source document, guess.html and your stylesheet guess.css**.

**Start early,** ask questions and have fun!

**Grading Rubric:**

Restart works correctly – web page is reinitialized and new word is displayed – 10 points

The game works correctly under the normal conditions (correct letter or wrong letter):  the dash display is updated  accordingly. The input is reinitialized to blank after each guess - 10 points

The game is case insensitive:  letters may be entered in lower case or upper case.  The words provided in the wordList array may also have mixed case - 10 points

Guess works correctly in the following special cases - 10 points

If the user presses GUESS without entering a letter, nothing happens.

If the user enters a correct letter that has already been entered and is in the display, nothing happens.

 If the user enters a wrong letter that has been entered before, it is added to the wrong letters (it will show twice) and the count of wrong guesses is incremented.

Winning the game – message is displayed – 5 points

Losing – message and correct answer displayed – 5 points

Wrong letters are shown correctly – 5 points

Some visualization of the number of incorrect guesses or the number of available guesses (progress bar or such) - 5 points

Program works for any answersList (of any length) and cycles through the words - 10 points

**Answer**

* application/x-javascript[guess.js](https://myetudes.org/access/mneme/content/private/mneme/09ae2205-2717-4bfc-00cf-33f5bdcd7b48/submissions/14984068/b7178838-9f5f-4fea-00cb-cbf5f3f8430a/guess.js)
* text/html[guess.html](https://myetudes.org/access/mneme/content/private/mneme/09ae2205-2717-4bfc-00cf-33f5bdcd7b48/submissions/14984068/31a2fd65-4ee1-4439-80c1-ed49bddf8309/guess.html)
* text/css[guess.css](https://myetudes.org/access/mneme/content/private/mneme/09ae2205-2717-4bfc-00cf-33f5bdcd7b48/submissions/14984068/42598962-36b4-4363-00c3-8f485588d6af/guess.css)

[[https://myetudes.org/ambrosia_library/icons/collapse.gif](https://myetudes.org/portal/tool/acd42055-9bd4-4630-8071-c0425c2388c3/review/14984068/list) Model Answer](https://myetudes.org/portal/tool/acd42055-9bd4-4630-8071-c0425c2388c3/review/14984068/list)

/\*

 \* Word Guessing Game - Solution

 \*

 \*/

'use strict';

// Define a container for the game variables

var game = {

  answerPosition: 0,   // position of the answer in the wordList

  display: '',         // the current dash/guessed letters display

  wrong: '',           // all the wrong letters guessed so far

  answer: '',          // the correct answer

  wrongCount: 0,       // the number of wrong guesses so far

  over: false,         // is the game over?

  answersList: [       // list of answers to cycle through

    'JavaScript',

    'document',

    'element',

    'ajax',

    'property',

    'event',

    'propagation',

    'listener',

    'transition',

    'animation'

  ]

  };

function dashes(number) {

    // this function takes a number as a parameter

    // and returns a string with that many dashes

    var result = '';

    for (var i = 1; i <= number; i++)  {

        result = result + '-';

    }

    return result;

}

function check(letter) {

    // Checks all occurrences of the letter guessed against game.answer.

    // Returns true if the guess is correct and false otherwise.

    // Updates the game dash display variable game.display if applicable.

    var position;

    var result = false;

    if (letter) {   // check that guess is not the empty string

        // Find the first occurrence of guess in the answer

        position = game.answer.indexOf(letter);

        // if the guessed letter is found in the answer

        if (position > - 1) {

            result = true;

        }

        while (position >= 0) {

            // update the dash display and find all remaining occurrences

            game.display = game.display.substring(0, position) + letter + game.display.substring(position + 1);

            // get the next occurrence

            position = game.answer.indexOf(letter, position + 1);

        }

    }

    return result;

}

function restart() {

    // Initialize the game at the beginning or after restart

    // Initialize the game variables - the model

    game.answer = game.answersList[game.answerPosition].toLowerCase(); // get the word for this round

    // use the modulo operator to cycle through the answersList

    game.answerPosition = (game.answerPosition + 1) % game.answersList.length;

    game.display = dashes(game.answer.length);

    game.wrong = '';

    game.wrongCount = 0;

    game.over = false;

    // Initialize the web page - the view

    document.getElementById('indicator').value = 0; // initialize the progress bar

    document.getElementById('display').textContent = game.display;

    document.getElementById('wrong').textContent = '';

    document.getElementById('guessedletter').value = '';

    document.getElementById('guessedletter').focus();

}

function play() {

    // Invoked when the user clicks on GUESS

    if (game.over) {// if the game is over

        document.getElementById('wrong').textContent = 'Press RESTART to play again.';  // user has to restart

    } else {

        //if the game is not over yet

        var guess = document.getElementById('guessedletter').value.toLowerCase();

        if (check(guess)) {

            // if the guess is valid

            document.getElementById('display').textContent = game.display;

        } else if (guess) {

            // If it's a wrong non-empty guess

            game.wrong = guess + ' ' + game.wrong;

            game.wrongCount++;

            document.getElementById('wrong').textContent = game.wrong;

            document.getElementById('indicator').value = game.wrongCount;

        }

        // reinitialize the guess

        document.getElementById('guessedletter').value = '';

        document.getElementById('guessedletter').focus();

        // check for a win or loss

        gameOutcome();

    }

}

function gameOutcome() {

    // check if the game is won or lost

    if (game.answer === game.display) {

        document.getElementById('wrong').textContent = 'Congratulations!  You win';

        game.over = true;  // game is over.  User has to restart to play again

    } else if (game.wrongCount >= 10) {  // 10 guesses only

        document.getElementById('wrong').textContent = 'No more guesses - the answer was ' + game.answer;

        game.over = true;  // game is over.  User has to restart to play again

    }

}

// Main program starts here

restart();

document.getElementById('guessbutton').addEventListener('click', play, false);

document.getElementById('restart').addEventListener('click', restart, false);

**Comments**

Good work overall.  There is a couple of issues:

The wrongCount is not initialized in restart.  It only gets reinitialized if the previous game ended in a win or a loss.

The program stops working when it reaches the end of  the list:

TypeError: game.answer is undefined

It is supposed to  start back from the beginning of the list

Second late submission - 10% penalty.

 1 of 1