

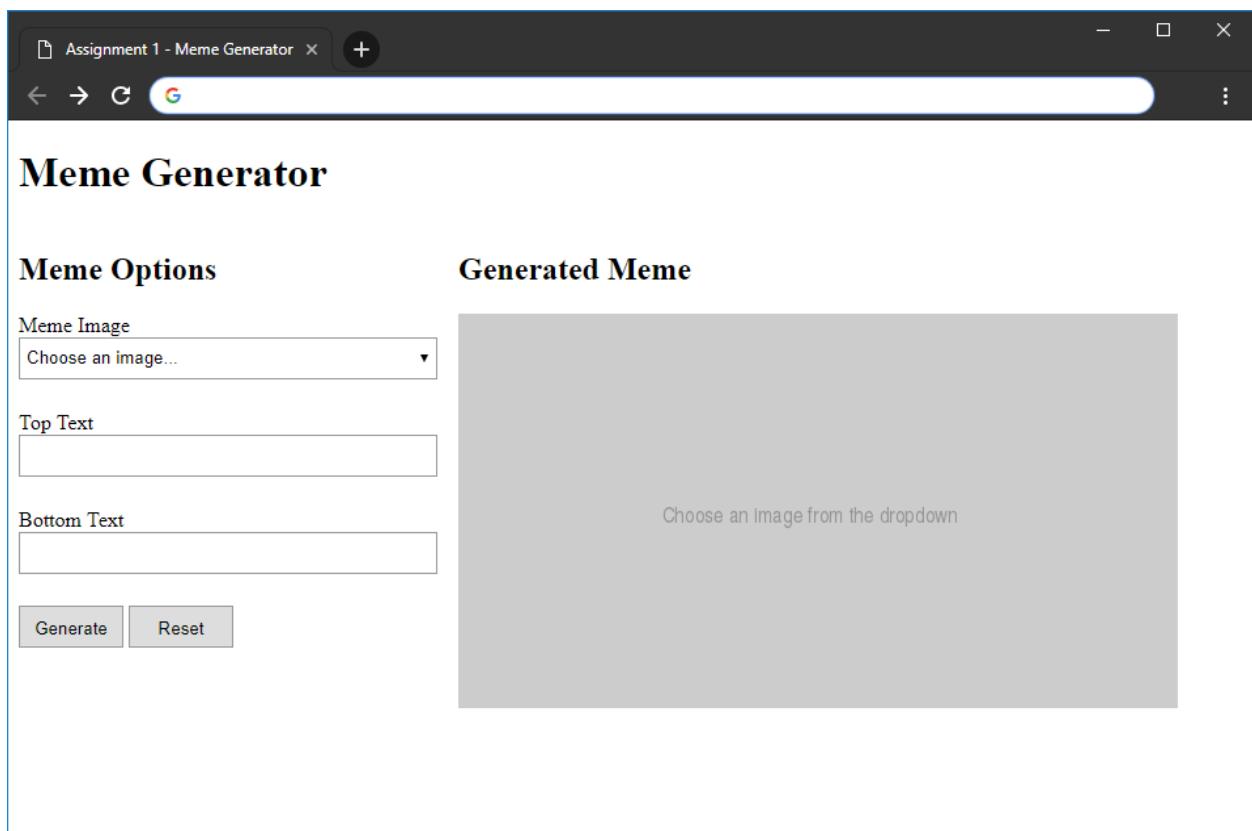
## CPSC1520 – JavaScript 1 Assignment 1: Meme Generator [10%]

### Introduction

This assignment builds on what you have already learned and demonstrated in class through the exercises and assessments to this point (i.e. up to and including event listeners and decisions). You are responsible for adding the JavaScript that will bring this simple page to life.

### Overview

The job is simple; the purpose of the page is to allow a user to create a meme from the available image options. In this case, the application contains only three images (but it should be able to contain any number of images), each of which can have unique text applied to it. There is a simple form the user will use to select the desired image and enter the text for the top and bottom of their meme.



The screenshot shows a web browser window with the title "Assignment 1 - Meme Generator". The page has a dark header bar with navigation icons. The main content area is titled "Meme Generator" in a large, bold, black font. Below the title, the interface is divided into two columns. The left column, titled "Meme Options", contains a "Meme Image" dropdown menu with the text "Choose an image..." and a downward arrow. Below this are two text input fields labeled "Top Text" and "Bottom Text". At the bottom of this column are two buttons: "Generate" and "Reset". The right column, titled "Generated Meme", features a large, light gray rectangular area. Inside this area, the text "Choose an image from the dropdown" is displayed in a light gray font, indicating that no meme has been generated yet.

Figure 1. Initial user interface for the assignment

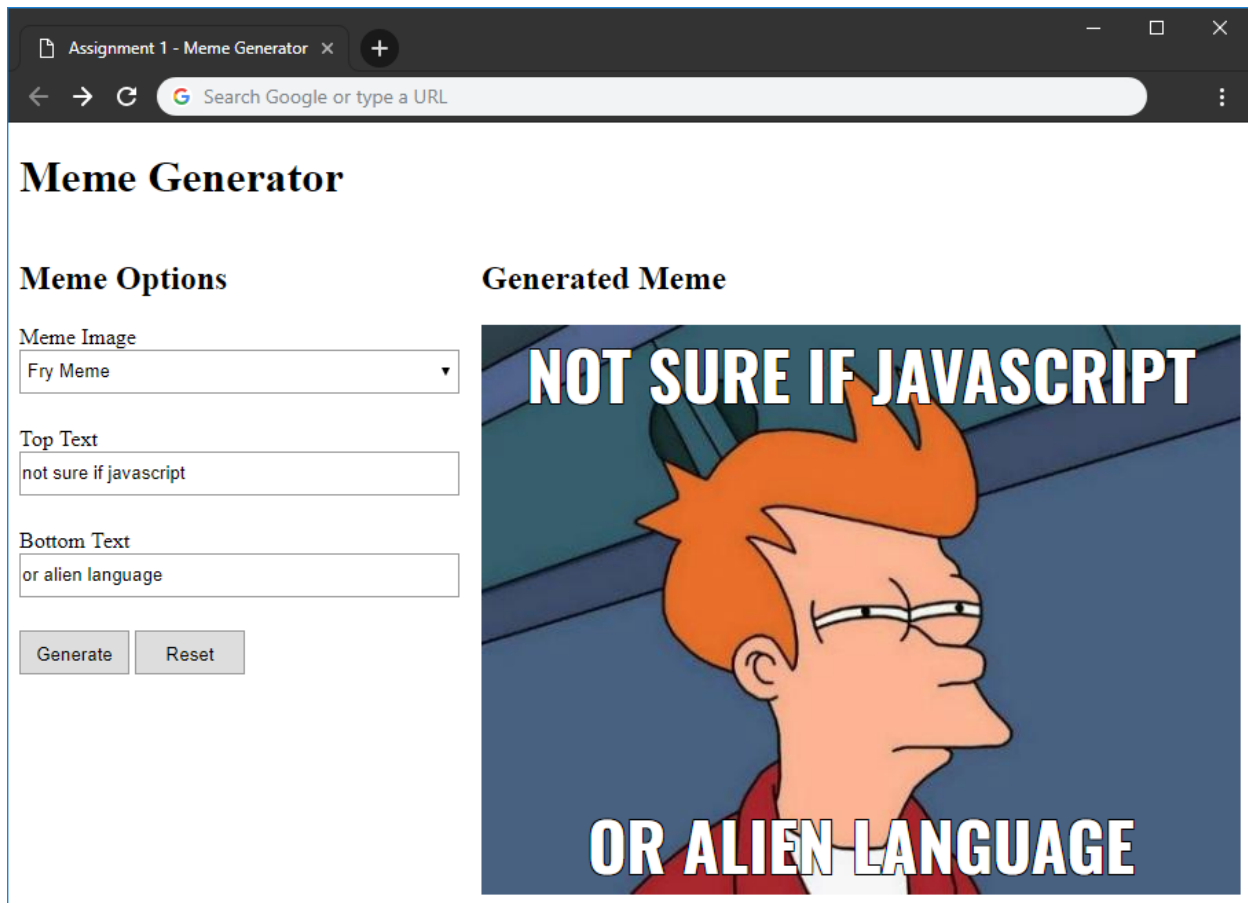


Figure 2. Display of generated meme

## Required Tasks

The following is a list of requirements for this assignment:

- Ensure that your script is properly added to the page
- An event listener is required for handling the submission of the form
  - A valid image must have been selected from the drop-down
    - display an error message in .error for the default option
  - The top and bottom text fields cannot be blank
    - display an error message in .error for empty text fields
  - If all fields are valid, set the src and alt of the .meme-display img to the value of the selected image (be sure to remove the dashes in the value) and properly display the top and bottom text in the appropriate p elements (see figure 2)
- An event listener is required for handling the reset of the form
  - In addition to clearing form (default), reset must also:
    - Clear the top and bottom p elements
    - Reset the img src to the original placeholder value
- *Bonus:* When an image is selected (i.e. the drop-down changes), the image should be loaded in the .meme-display img element (i.e. set the src and alt attributes to those of the option that was selected)

- Pay careful attention to the user experience and aesthetic (e.g. when an error message should be visible vs. when it should be reset, spaces between tags, hash sign on tags, etc.)
- **Do not alter the HTML.** Your task is simply to write the JavaScript for the assessment.
- If you are unsure of any of the requirements, **be sure to ask your instructor for clarification.**
  - It is your responsibility to seek clarification, do not make assumptions

## Marking Key

Tasks	Grade	Marks	Total
<b>Form Submit Event Listener</b> <ul style="list-style-type: none"> <li>• Event listener added; submitting form doesn't reload page</li> <li>• Proper values are set in the .meme-display to generate the meme               <ul style="list-style-type: none"> <li>○ Correct image is displayed</li> <li>○ img alt is option value with dashes removed</li> <li>○ Correct top and bottom text displayed in correct elements</li> </ul> </li> </ul>		1 5	
<b>Form Validation</b> <ul style="list-style-type: none"> <li>• Checks for blank/empty values</li> <li>• Displays proper message for invalid values</li> <li>• Remove error message(s) when appropriate</li> </ul>		1 1 1	
<b>Form Reset Event Listener</b> <ul style="list-style-type: none"> <li>• Event listener added</li> <li>• Image src and alt are properly reset</li> <li>• Top and bottom text are properly reset</li> </ul>		1 1 1	
<b>Best Practices</b> <ul style="list-style-type: none"> <li>• Proper use of let and const (where appropriate) – i.e. no var declarations</li> <li>• Commonly referenced elements are stored in variables</li> <li>• Access to form elements through the form reference</li> </ul>		3	
<b>Code Formatting and Style</b> <ul style="list-style-type: none"> <li>• Naming conventions</li> <li>• Indentation and spacing</li> <li>• Consistency</li> <li>• Etc.</li> </ul>		-3	

Comments	SCORE	GRADE
	15	

## Marking Rubric

Marks	5 Marks Criteria
5	Task was completed with the highest of proficiency adhering to best practices and followed subject matter guidelines all tasks were completed to a professional standard.
4	Task was completed well Some minor mistakes. Well above average work shows good understanding of the task and high degree of competence
3	Satisfactory work Some features missing or incorrectly implemented. Shows a moderate level of understanding in the task with room for improvement.
2	Below average work Task was poorly completed. Shows understanding of the task and the requirements to implement but implementation was poorly executed.
1	Some of the task was completed Showed a lack of understanding in the subject matter and very poorly executed
0	Not completed.

Marks	3 Marks Criteria
3	Proficient Shows a high degree of competence in completing task.
2	Capable Above average degree of competence in completing task
1	Satisfactory Shows a satisfactory degree of competence in completing task.
0	Shows a limited degree of competence in completing task.

Marks	1 Marks Criteria
1	Task Completed satisfactorily
0	Task was not completed