**Objective**

Use what you have learned about form and event handling to create a Mad Libs web application.

**Workflow**

The following outlines the web application's workflow.

1. Upon arriving on the web application the user can choose one of the two stories (see below) to complete.
2. Once chosen, the user will be presented with a form asking for a series of words of various types (noun, verb, adverb, etc.).
3. After completing and submitting the form, the user's inputs should be validated. All inputs are required and should not be left blank. The web application should alert the user when an input is not valid.
4. After ensuring all inputs are valid, use the words the user provided to display the story using the user's words.
5. After reading the completed story, the user will have the option to complete another Mad Lib.

**Stories**

The web application should incorporate the following two stories. The words in brackets ([]) indicate where the user-provided words will be placed. **When inserting the words, make sure to remove the brackets as well.**

**Lunch Room**

Make sure your lunch [CONTAINER] is filled with [ADJECTIVE 1] food. Do not go to

the [ADJECTIVE 2] food stand across the street from the school. The hamburgers

they serve are fried in [NOUN] and are made of [ANIMAL] meat. So take a sandwich

made of [VEGETABLE 1] or [VEGETABLE 2]. It's much healthier!

**Weather Report**

Early tomorrow, a [ADJECTIVE 1] front will collide with a mass of hot

[PLURAL NOUN 1] moving from the north. This means we can expect [ADJECTIVE 2]

winds and occasional [PLURAL NOUN 2] by late afternoon. Wind velocity will be

[NUMBER 1] miles an hour, and the high temperature should be around [NUMBER 2]

degrees. So, if you're going out, you had better plan on wearing your

[ARTICLE OF CLOTHING].

**Starter Files**

The starter files will include an HTML file containing three forms: one form story selection and two to request the words for the stories. The HTML file will also contain a dialog box to be used for the completed story. The HTML has already been styled using the Bootstrap framework. Students may change the content of the HTML file.

**Instructions**

1. Clone the repository created by GitHub Classroom. ***Do not create a new repository.*** Review the provided starter files.
2. Create a script.js file. Update the index.html to link to the script.js file.
3. Create the necessary JavaScript to display the appropriate story form when the user selects a story.
4. Create the necessary JavaScript and HTML to validate the form and provide feedback when the user fails to enter the correct data or enters no data at all.
5. Create the necessary JavaScript to display the completed story in the dialog modal when the user completes and submits a story form.
6. Create the necessary JavaScript to allow the user to play again. This will close the dialog modal, clear the forms, and present the user with the option to select a story.
7. All event and form handling should use event listeners.
8. At no point should the web application experience a page refresh.

**Hints**

1. Bootstrap's [display property](https://getbootstrap.com/docs/5.3/utilities/display/) includes the d-none class, which applies display: none to an element and hides it from the user.
2. All inputs are required. Only the number inputs require specific values. All others only need to check if they are not empty.
3. Bootstrap's [validation](https://getbootstrap.com/docs/5.3/forms/validation/) styling can be applied to the forms to alert the user when an input is invalid.

**Example**

***Coming soon***

**Submission**

1. Create a commit with the message "Completes the assignment"
2. Push to GitHub
3. Submit the URL to your GitHub Repository to Brightspace

Some help:

Video: <https://www.youtube.com/watch?v=BZtTdUoXduc&list=PLfow3Ls3kyIM5jFC1Douo_8SbOpyBV46h&index=25>

Lecture week 10.

<https://www.youtube.com/watch?v=FkwjeG6sDag&list=PLfow3Ls3kyIM5jFC1Douo_8SbOpyBV46h&index=21>