Kelly Kuhn
N220
August 1st, 2023

Lab 9 – Array Computation Exercises

https://rkkuhn.github.io/N220Spring2023/

Algorithm 1st Project – Bad Word Catcher

Requirements: For this exercise, bad words are: **clear, water, tires**.

Create an application that:

- Takes input from a user using a simple text input field.
- Splits the string into spaces
- Loops through the array, looking for bad words in the array
  - Adds 1 to a tally count when a bad word is found
- Outputs to the document
  - If any bad words were found (found / not found)
  - How many bad words, in total, were found
- Clears out the text field so the user can input a new string

For example, if the user input "clear water is clear", there are 3 bad words in the string.

Expected Output:

Sudo Code:

HTML

Create heading Bad word Catcher
Input type = "text" id = "textInput" placeholder = "Enter your text here"
Create onclick button name checkBadWords()

JS

Global var. badWord – clear, water tires

Create checkBadwords function
Use iinputText = document.getElementById
Split the txt entered into individual words
Let badWordCount = 0

Create a stand for statement
If (badWords.includes (wordsArray[i]))
badWordCount++ // Increases the count number when a bad word is found
*wordsArray[i] = "***" // replaces the bad words with ****

Create outputElement p document.getElementById
Reference the HTML page in two ways
Bad words found:
No bad words found

Clear the input text field

Algorithm 2nd Project – Bad Word Catcher Array

Requirements: For the purposes of this exercise, bad words are: clear, water, tires. Recreate the above exercise, but using the array.filter and array.includes methods.

Expected Output:

Sudo Code:

HTML

- Create input type with hard code of bad words
- Create onclick button command
- Create two div's
  - wordsFound
  - wordAmount

JS

- Create function badWordCatch
- Get input from document.getElementById
- Change characters entered to lower case
- Create split
- Look for the following bad words
- Do case match
  - Bad words
  - Number found

Algorithm 3rd Project – Make Divs

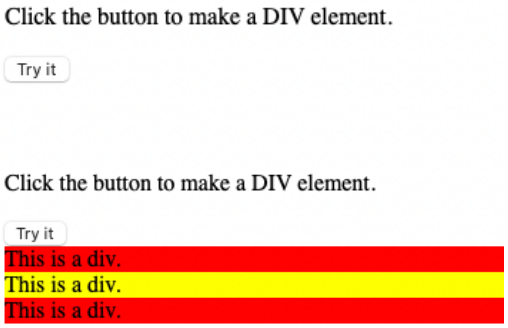
Requirements: Given the array

```
let objects = [
 { color: "#FF0000", height: 100, width: 300 },
 { color: "#FFFF00", height: 200, width: 200 },
 { color: "#ff0000", height: 300, width: 100 },
```

];

Write a loop that creates three divs based on the data in the array. You should only have one document.createElement in your code.

Expected Output:



Sudo Code:

HTML

Creat an onclick button with instructions telling the user to click it.

JS

Create the function createDiv(obj)
Set up div = document.createElement("div")
Div for width
Div for height
Div for color
Use appendChild(div)

Create myFunction
Create the array for the color, height, and width

For statement
createDiv(objects[i]

Reflection: Arrays and lists are my favorite to do. I find them easy. I guess it has to do with being forced to use them in several different classes started in computer II. This assignment I did not find too difficult. Once I had the general idea of what I wanted to do on paper,

the rest was getting it down in Visual Studio Code. I do have a question for you. I use a Mac. When I try to look at my work in Safari it is slow or does not want to pull. Yet, Firefox and Google I have no issues. Why is that? Is there a way to code that works with Safari?