**Quick Question #1**

What does the following code return?

new Set([1,1,2,2,3,4]) : [1,2,3,4]

## ****Quick Question #2****

What does the following code return?

[...new Set("referee")].join(""). : “ref”;

## ****Quick Questions #3****

What does the Map ***m*** look like after running the following code?

let m = new Map();

m.set([1,2,3], true);

m.set([1,2,3], false);

**hasDuplicate**

const hasDuplicate = arr => new Set(arr).size !== arr.length

1. Create a function called **hasDuplicate** that takes an array as its argument.
2. Inside the function, create a new **Set** object using the input array (**arr**). The **Set** object will automatically remove any duplicate elements from the array, as it only stores unique values.
3. Get the size of the **Set** object, which represents the number of unique elements in the array.
4. Compare the size of the **Set** with the length of the original array (**arr**), which is the total number of elements in the array.
5. If the size of the **Set** is not equal to the length of the array, it means there are duplicate elements present. In this case, return **true**.
6. Otherwise, if the size of the **Set** is equal to the length of the array, it means all elements are unique. In this case, return **false**

**VowelCount**

**vowelCount('awesome') // Map { 'a' => 1, 'e' => 2, 'o' => 1 }**

**vowelCount('Colt') // Map { 'o' => 1 }**

**function isVowel(char){**

**return "aeiou".includes(char);**

**}**

**function vowelCount(str){**

**const vowelMap = new Map();**

**for(let char of str){**

**let lowerCaseChar = char.toLowerCase()**

**if(isVowel(lowerCaseChar)){**

**if(vowelMap.has(lowerCaseChar)){**

**vowelMap.set(lowerCaseChar, vowelMap.get(lowerCaseChar) + 1);**

**} else {**

**vowelMap.set(lowerCaseChar, 1);**

**}**

**}**

**}**

**return vowelMap;**

**}**

function isVowel(char) {

// Checks if a character is a vowel.

return "aeiou".includes(char);

}

function vowelCount(str) {

// Returns a map of the vowels in a string, along with the number of times each vowel appears.

// Create a new map to store the vowels and their counts.

const vowelMap = new Map();

// Iterate over the characters in the string.

for (let char of str) {

// Convert the character to lowercase.

let lowerCaseChar = char.toLowerCase();

// Check if the character is a vowel.

if (isVowel(lowerCaseChar)) {

// If the character is a vowel, increment its count in the map.

if (vowelMap.has(lowerCaseChar)) {

vowelMap.set(lowerCaseChar, vowelMap.get(lowerCaseChar) + 1);

} else {

vowelMap.set(lowerCaseChar, 1);

}

}

}

// Return the map of vowels and their counts.

return vowelMap;

}