**GENERATING AN ARRAY OF RANDOM NUMBERS WITHOUT REPITITION WITH JS**

What’s an array?

Javascript provides a data structure that is used to store multiple values of the same/different datatypes in a single variable. That definition is the basic, we won’t be diving much into array. The basic and most common syntax of declaring and defining array in JS goes thus:

let arrayVariable = [“firstValue”,”secondValue”,...];

Using javascript new keyword

let arrayVariable = new Array(“firstValue”,”secondValue”,…);

I assume you are familiar with the basic JS syntax such as for loop, while loop, and methods used on array too. So let’s go straight to the main work. You can view the whole code here.

We will create a webpage that generates and display the array. Let’s create the HTML page now

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

**<meta http-equiv="X-UA-Compatible" content="ie=edge">**

**<title>Array Generator</title>**

**</head>**

**<body>**

**<main>**

**<h3 id="array"></h3>**

**<button id="generate">Generate </button>**

**</main>**

**<script src="app.js"></script>**

**</body>**

**</html>**

So Let’s write our JS code

**window.addEventListener("load", () => {**

**generate(); // this calls our generate function whenever the page is loaded**

**});**

**let buttonClicked = document.getElementById('generate'); //button that triggers generate function too**

**buttonClicked.addEventListener("click", () => { // when clicked, calls the generate function**

**generate();**

});

So the above code specifies what happens whenever the page is loaded or reloads and what happens too when the generate button is clicked

Our generate function goes thus:

**const generate = () => {**

**let arrayContainer = [];**

**let displayContainer = document.getElementById('array');**

**const genNum = Math.floor(Math.random() \* 30);**

**arrayContainer.push(genNum);**

**for (let counter = 0; counter < 10; counter++) {**

**let newGen = Math.floor(Math.random() \* 30);**

**while (arrayContainer.lastIndexOf(newGen) !== -1) {**

**newGen = Math.floor(Math.random() \* 30);**

**}**

**arrayContainer.push(newGen);**

**}**

**console.log(arrayContainer);**

**displayContainer.innerHTML = "[" + arrayContainer + "]";**

**}**

The first line of the generate block declared and initialized the arrayContainer variable as empty. It gets occupied over time.

**let arrayContainer = [];**

The second line of the generate block gets the h3 where the arrayContainer is displayed

**let displayContainer = document.getElementById('array');**

The third and the fourth line generates any random number and set it as the first element of the arrayContainer. The Math.floor() function returns the greatest integer less than or equal to its numeric argument provided by the Math.

**const genNum = Math.floor(Math.random() \* 30);**

**arrayContainer.push(genNum);**

Let’s move to the for loop block, there is something cunny about the code. Let me explain.

The condition of the for loop is dependent on the value of the newGen, you don’t want to generate 5 random non repetitive numbers between 1-3, hence my multiplication by 30. So therefore my inference is that the number used to multiply Math.random() must be greater than or equals to the number used for the condition of the for block.

Math.random() \* **30,** counter < **10, i.e 30 >= 10**

**for (let counter = 0; counter < 10; counter++) {**

**let newGen = Math.floor(Math.random() \* 30);**

**while (arrayContainer.lastIndexOf(newGen) !== -1) {**

**newGen = Math.floor(Math.random() \* 30);**

**}**

**arrayContainer.push(newGen);**

**}**

The while block makes use of lastIndexOf, the lastIndexOf method returns the index of the last occurrence of a specified value in an array, if the specified value isn’t found in the array, it returns -1.

The below code displays the arrayContainer to the web user.

**displayContainer.innerHTML = "[" + arrayContainer + "]";**

I hope this helps a lot. If you see some errors or better ways to do this, kindly share and don’t forget to hit the like button.