Working with 2D Arrays in JavaScript

This document provides a concise overview of handling 2D arrays in JavaScript, emphasizing practical, code-driven examples. A 2D array, often referred to as a matrix, is an array of arrays, where each element is itself an array. This structure is useful for applications such as game development, simulations, and data processing tasks.

Creating a 2D Array

Here's how you can initialize a 2D array and populate it with values:

```
const rows = 4;
const cols = 3;
const array2D = new Array(rows);

for (let i = 0; i < rows; i++) {
    array2D[i] = new Array(cols).fill(0);  // Fill each row with 0
}
console.log(array2D);
```

Accessing Elements

Accessing elements in a 2D array involves specifying both the row and column indices:

```
// Set value
array2D[2][1] = 5;

// Get value
const value = array2D[2][1];
console.log(value);  // Outputs: 5
```

Modifying a 2D Array

Modifying a 2D array often involves iterating through rows and columns:

```
// Increment each element by 1
for (let i = 0; i < rows; i++) {
    for (let j = 0; j < cols; j++) {
        array2D[i][j] += 1;
    }
}
console.log(array2D);
```

Example: Transposing a Matrix

Transposing a matrix means swapping its rows with its columns:

```
const matrix = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9],
];

const rows = matrix.length;
const cols = matrix[0].length;
const transposed = new Array(cols);

for (let i = 0; i < cols; i++) {
    transposed[i] = new Array(rows);
    for (let j = 0; j < rows; j++) {
        transposed[i][j] = matrix[j][i];
    }
}

console.log(transposed);
```

Example: Summing all Elements

Calculating the sum of all elements in a 2D array:

```
let sum = 0;
for (let i = 0; i < matrix.length; i++) {
    for (let j = 0; j < matrix[i].length; j++) {
        sum += matrix[i][j];
    }
}

console.log(sum);  // Outputs the sum of all numbers in originalMatrix
```

Example: Searching for an Element

Here's how to find an element in a 2D array and print its position:

```
const matrix = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9],
];
const target = 5;

// Variables to store the search results
let found = false;
let position = '';

// Iterate over each row of the matrix
for (let i = 0; i < matrix.length; i++) {
    // Iterate over each column of the current row
    for (
        let j = 0;
        j < matrix[i].length;
        j++
    ) {
        // Check if the current element matches the target
        if (matrix[i][j] === target) {
            position = `Element found at row ${i} and column ${j}`;
            found = true;
            break; // Stop searching once the element is found
        }
    }
    if (found) {
        break; // Stop searching other rows once the element is found
    }
}

// Check if the element was not found
if (!found) {
    position = 'Element not found';
}
```

```
console.log(position);
```