

## Background Story

Odessa's journey into the world of programming continues as she dives into building her own grade tracker. Being an IT student in the Philippines, Odessa often has to keep track of her quiz scores and assignments. To make this task easier, she decides to create a simple application that can help her calculate her average scores.

She starts by understanding the concept of arrays. An array is like a list that can hold multiple values in a single variable. In Odessa's case, she plans to store her quiz scores in an array so that she can perform calculations on them efficiently.

As Odessa learns about creating arrays and accessing values, she begins to see the power of such data structures. With each quiz score added to the array, she feels more confident in her ability to manage and manipulate data.

Now, Odessa faces the challenge of looping through these arrays to calculate the average score. This concept seems a bit tricky at first, but she knows that practice and determination are key to mastering it. With her newfound knowledge, Odessa is excited to apply arrays and loops to create her grade tracker.

## Theory & Lecture Content

In JavaScript, an array is a type of data structure that can store multiple values. These values can be of any data type—numbers, strings, objects, or even other arrays.

### Creating Arrays

To create an array in JavaScript, we use square brackets `[]` and separate each value with a comma.

```
// Creating an array of quiz scores
let quizScores = [80, 75, 90, 85, 88];
```

### Accessing Values

We can access individual elements in an array using their index. Remember, array indices start from 0.

```
console.log(quizScores[0]); // Output: 80
```

### Looping Through Arrays

To iterate over all elements in an array, we can use loops like `for` loop or `forEach` method.

```
// Calculating the average score
let sum = 0;
for (let i = 0; i < quizScores.length; i++) {
  sum += quizScores[i];
}
```

```
}  
let average = sum / quizScores.length;  
console.log("Average Score:", average);
```

For more information, you can visit the [Mozilla Developer Network \(MDN\) documentation on Arrays](#).

## Exercises

### Exercise 1

#### Problem Statement

Create a function `findMaxNumber` that takes an array of numbers as input and returns the maximum number in the array.

#### Starter Code

```
function findMaxNumber(numbers) {  
  // Your code here  
}
```

#### Full Solution Code

```
function findMaxNumber(numbers) {  
  return Math.max(...numbers);  
}
```

### Exercise 2

#### Problem Statement

Create a function `findVowels` that takes a string as input and returns an array of vowels present in the string.

#### Starter Code

```
function findVowels(str) {  
  // Your code here  
}
```

#### Full Solution Code

```
function findVowels(str) {  
  const vowels = ["a", "e", "i", "o", "u"];  
  return str  
    .toLowerCase()  
    .split("")  
    .filter((char) => vowels.includes(char));  
}
```

## Exercise 3

### Problem Statement

Create a function `reverseArray` that takes an array as input and returns a new array with the elements reversed.

### Starter Code

```
function reverseArray(arr) {  
  // Your code here  
}
```

### Full Solution Code

```
function reverseArray(arr) {  
  return arr.reverse();  
}
```

## Test Cases

Below are the Jest test cases to validate the above exercises.

```
test("Finding maximum number in an array", () => {  
  expect(findMaxNumber([3, 7, 12, 5, 9])).toBe(12);  
});  
  
test("Finding vowels in a string", () => {  
  expect(findVowels("Hello World")).toEqual(["e", "o", "o"]);  
});  
  
test("Reversing an array", () => {  
  expect(reverseArray([1, 2, 3, 4, 5])).toEqual([5, 4, 3, 2, 1]);  
});
```

## Closing Story

As Odessa completes her grade tracker using arrays and loops, she feels a sense of accomplishment. She can now easily manage her quiz scores and calculate averages with just a few lines of code. This small project has taught her the importance of data structures and how they can simplify complex tasks.

Next, Odessa is curious about how she can use functions to make her code more modular and reusable. Join her in the next lesson as she explores the world of functions and unlocks new possibilities in her programming journey!