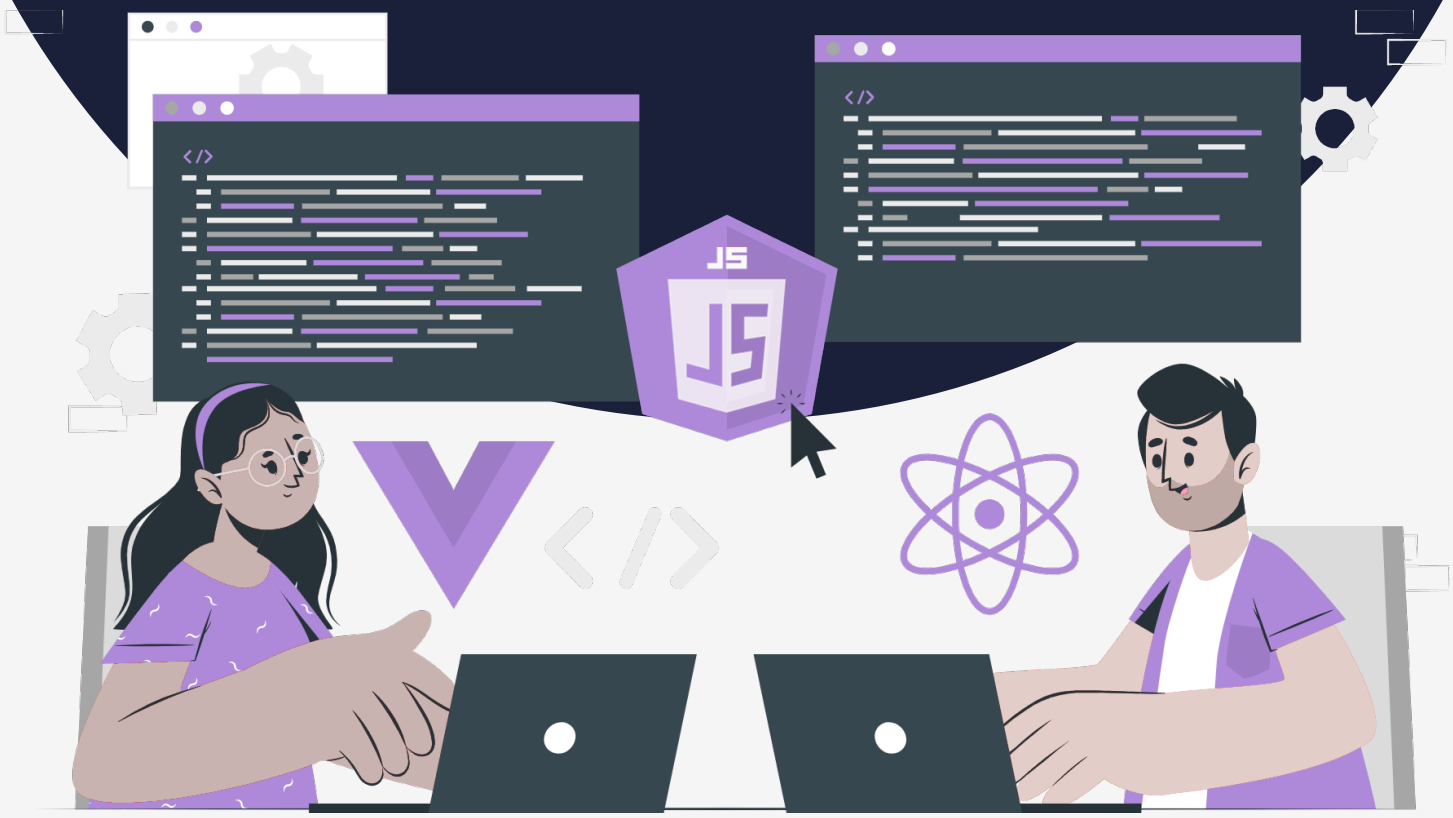
Lesson:

Arrays in JavaScript



PW SKILLS

Topics Covered:

- What is an Array?
- Why do we need arrays in JavaScript?
- Declaration of Array.
- Arra index and storing.
- Accessing elements in an array
- Changing values in an array

What is an Array?

An array in JavaScript is a data structure that stores an ordered list of elements. It can hold elements of any data type, including numbers, strings, objects, and even other arrays. Arrays are a type of object in JavaScript and have a number of built-in methods for adding, removing, and manipulating elements.

Example of an array:

```
ls et player= ["Virat Kohli", "Rohit Sharma", "Suryakumar Yadav", "KL
Rahul", "Ravindra Jadeja", "Rishabh Pant", "Shivam"];

let numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0];

let array = ["Hello", 20, true]
```

Why do we need Arrays in JavaScript?
Arrays in JavaScript are needed for several reasons:

1. **Grouping related data:** Arrays allow you to group related data together and make it easier to manipulate. For example, you could use an array to store a list of names or a list of products.

2. **Storing large amounts of data:** Arrays can store large amounts of data in a single variable, making it easier to manage and manipulate the data. This can be especially useful when working with large datasets.

3. **Improving performance:** Arrays can improve performance when working with large amounts of data. Because the elements in an array are stored in consecutive memory locations, it is faster to access elements in an array than it is to access elements in other data structures, such as objects.

4. **Ease of use:** Arrays come with a number of built-in methods for adding, removing, and manipulating elements. This makes it easy to perform operations on the data stored in an array.

5. **Better readability:** Arrays make your code more readable by grouping related data together in a single structure. This makes it easier for others to understand your code and makes it easier for you to maintain your code in the future.

Overall, arrays in JavaScript are a crucial tool for managing and manipulating data in various applications. They provide a flexible and efficient way to store and process data and make it easier to write clean, maintainable code.

Declaration of Array:

There are several ways to declare an array in JavaScript.

1. **Using square brackets [ ]:** This is the most common and recommended way to declare an array in JavaScript. You can create an array by enclosing a comma-separated list of elements in square brackets:

```
let fruits = ["Apple", "Mango", "Banana", "Kiwi"];
```

2. **Using the Array constructor:** You can also create an array using the Array constructor. You can pass in the length of the array or a list of elements:

```
let numbers = new Array(1, 2, 3, 4, 5);

let emptyArray = new Array(5); // Creates an array with 5 empty elements.
```

3. **Using an array literal:** This is a shorthand method for creating arrays that are equivalent to using square brackets:

```
let colors = Array("Black", "Red", "White", "Blue");
```

Array index and storing

Indexing and storing in arrays are related concepts in JavaScript. The index of an element in an array determines its position within the array while storing in an array refers to the act of assigning a value to an element within the array.

In JavaScript, arrays are zero-indexed, meaning that the first element has an index of 0, the second element has an index of 1, and so on. To store a value in an array, you can use the square bracket notation and the index number:

```
let players= []; // create an empty array
fruits[0] = "Virat"; // store "Virat" at index 0
fruits[1] = "Rohit"; // store "Rohit" at index 1
fruits[2] = "Suryakumar"; // store "Suryakumar" at index 2
```

Accessing elements in an array

Accessing elements in an array in JavaScript is done using the square bracket notation and an index number. The index number represents the position of the element within the array, starting from 0 for the first element, 1 for the second element, and so on.

For example, consider the following array:

```
let players = ["Virat", "Rohit", "Suryakumar"];
// To access the first element "Virat", you would use the following code.

players[0]; // returns "Virat"
```

You can access any element in the array using its index number:

```
players[1]; //returns "Rohit"
players[2]; // returns "Suryakumar"
```

It's important to keep in mind that if you try to access an index that is outside the bounds of the array, you will get **undefined**.
And also if you try to access a negative index it will give you an **undefined**.

For example,

```
players[3]; // returns undefined
players[-1]; // returns undefined
players[-3]; // returns undefined
```

In some other programming languages like Java, C, and C++ if you try to access a negative index, you will get an error which is **ArrayIndexOutOfBoundException** or **Invalid OutOfRange**.

Overall, accessing elements in an array is a basic operation in JavaScript that allows you to retrieve values stored in the array. Understanding how to access elements in an array is essential for working with arrays in JavaScript.