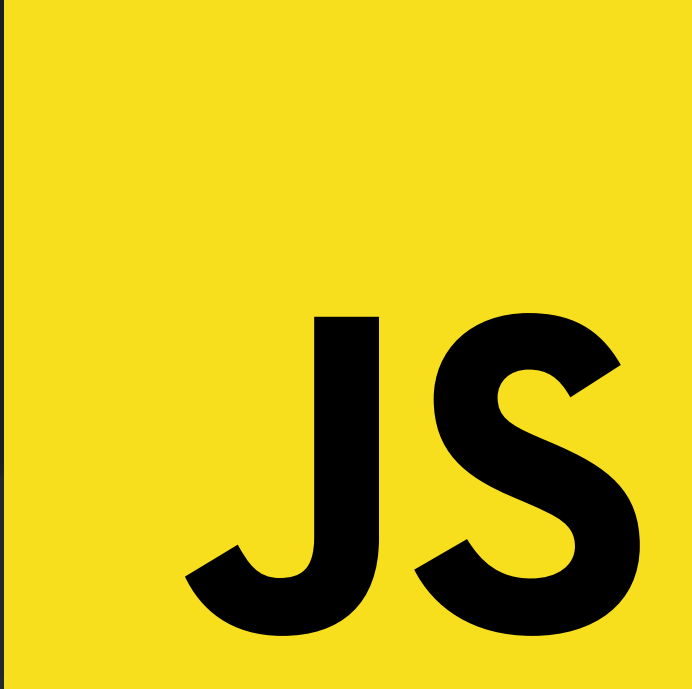


JavaScript

Destructuring

ES6 | In Detail

A yellow square containing the letters 'JS' in a bold, black, sans-serif font, representing the JavaScript logo.

JS

Mallikarjun | @CodeBustler



JS Destructuring

Allows you to extract values from **arrays** or **objects** and assign them to **variables** using a concise syntax.

It's a convenient way to extract specific data from complex data structures and simplify your code.

Destructuring is a powerful feature, It's widely used in modern JavaScript frameworks and libraries, such as React and Vue.js, to extract **props and state variables**.

Destructuring Arrays



JS index.js

```
const myArray = [1, 2, 3];
```

```
const [a, b, c] = myArray; ➡
```

```
console.log(a); // Output: 1
```

```
console.log(b); // Output: 2
```

```
console.log(c); // Output: 3
```

Here **a**, **b** & **c** are separate variables.

Destructuring Objects



JS index.js

```
const myObj = { name: "CodeBustler", age: 27 };  
  
const { name, age } = myObj; ➡  
  
console.log(name);  
// Output: 'CodeBustler'  
  
console.log(age);  
// Output: 27
```

Here **name** & **age** are separate variables.

Nested Objects



Js index.js

```
const myObj = {  
  name: "Arjun",  
  age: 27,  
  address: {  
    city: "Gulbarga",  
    state: "KA"  
  }  
};  
  
const { name, age, address: { city, state } } = myObj;  
  
console.log(name); // 'Arjun'  
console.log(age); // 27  
console.log(city); // 'Gulbarga'  
console.log(state); // 'KA'
```



Default Values



Js index.js

Array

```
const myArray = [1, 2];

const [a, b, c = 3] = myArray;
      ↑
console.log(a); // Output: 1
console.log(b); // Output: 2
console.log(c); // Output: 3
```



Js index.js

Object

```
const myObj = { name: "CodeBustler" };

const { name, age = 27 } = myObj;
              ↑
console.log(name); // Output: 'CodeBustler'
console.log(age); // Output: 27
```

Alternate names



JS index.js

Array

```
const myArray = [1, 2, 3];  
  
const [a, b, c: third] = myArray;  
                                🙋  
console.log(a); // Output: 1  
console.log(b); // Output: 2  
console.log(third); // Output: 3
```



JS index.js

Object

```
const myObj = { name: "Arjun", age: 27 };  
  
const { name: myName, age } = myObj;  
                                🙋  
console.log(myName); // Output: 'Arjun'  
  
console.log(age); // Output: 27
```

Other Posts

What Is Events ?

In JavaScript



Mallikarjun | @CodeBustler



List Of JS Events

With Categorized



Mallikarjun | @CodeBustler



To learn **HTML CSS & JavaScript** or
Front-End Projects, Visit my page all
concepts & Projects codes are
available 🔥

@CodeBustler



@CodeBustler

Mallikarjun | Web Developer

Follow For More !

And i need dopamine ⚡

