JavaScript Destructuring

Back to JS page

Table of Contents

- JavaScript Destructuring
  - Destructuring Assignment Syntax
    - Example 1
  - Object Destructuring
    - Example 2
  - String Destructuring
    - Example 3
  - Array Destructuring
    - Example 4
  - Destructuring Maps
    - Example 5
  - Swapping JavaScript Variables
    - Example 6
  - Document
  - Reference

Destructuring Assignment Syntax

The **destructuring assignment** syntax is a JavaScript expression that makes it possible to unpack values from arrays, or properties from objects, into distinct variables.

Destructuring makes it easy to extract only what is needed from data.

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Destructuring</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Destructuring Assignment Syntax</h4>
<p id="demo"></p>

<script>
// Create an Array
const vehicles = ["mustang", "f-150", "expedition"];

// Old way
const car = vehicles[0];
const truck = vehicles[1];
const suv = vehicles[2];

// New way - Destructuring
const [car2, truck2, suv2] = vehicles;

document.getElementById("demo").innerHTML =
  "Destructured: " + car2 + ", " + truck2 + ", " + suv2;
</script>

</body>
</html>
```

Example 1

Result View Example

Object Destructuring

Object destructuring allows you to extract properties from objects and assign them to variables. You can also assign **default values** and use **property aliases**.

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Destructuring</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Object Destructuring</h4>
<p id="demo"></p>

<script>
// Create an Object
const person = {
  firstName: "John",
  lastName: "Doe",
  age: 50
};

// Destructuring
let {firstName, lastName} = person;

// Default values
const {firstName: fName, country = "US"} = person;

// Property alias
const {lastName: lName} = person;

document.getElementById("demo").innerHTML =
  firstName + " " + lastName + "<br>" +
  "Alias: " + fName + " " + lName + "<br>" +
  "Country (default): " + country;
</script>

</body>
</html>
```

Example 2

Result View Example

String Destructuring

You can use destructuring to unpack characters from a string into variables.

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Destructuring</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>String Destructuring</h4>
<p id="demo"></p>

<script>
// Create a String
let name = "W3Schools";

// Destructuring
let [a1, a2, a3, a4, a5] = name;

document.getElementById("demo").innerHTML =
  a1 + " " + a2 + " " + a3 + " " + a4 + " " + a5;
</script>

</body>
</html>
```

Example 3

Result View Example

Array Destructuring

Array destructuring allows unpacking array values into separate variables. You can **skip values** using commas, use **position values**, and capture remaining elements with the **rest property** (`...`).

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Destructuring</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Array Destructuring</h4>
<p id="demo"></p>

<script>
const numbers = [10, 20, 30, 40, 50, 60, 70];

// Basic array destructuring
const [x, y] = numbers;

// Skipping array values
const [a, , , b] = numbers;

// Array position values
const {[0]:p1, [1]:p2} = numbers;

// The rest property
const [first, second, ...rest] = numbers;

document.getElementById("demo").innerHTML =
  "x = " + x + ", y = " + y + "<br>" +
  "Skipped: a = " + a + ", b = " + b + "<br>" +
  "Position: p1 = " + p1 + ", p2 = " + p2 + "<br>" +
  "Rest: first = " + first + ", second = " + second + ", rest = [" + rest + "]";
</script>

</body>
</html>
```

Example 4

Result View Example

Destructuring Maps

Destructuring can also be used with Maps to extract key-value pairs.

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Destructuring</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Destructuring Maps</h4>
<p id="demo"></p>

<script>
// Create a Map
const fruits = new Map([
  ["apples", 500],
  ["bananas", 300],
  ["oranges", 200]
]);

let text = "";

// Destructuring Maps
for (const [key, value] of fruits) {
  text += key + " is " + value + "<br>";
}

document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

Example 5

Result View Example

Swapping JavaScript Variables

Destructuring makes it easy to swap the values of two variables without needing a temporary third variable.

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Destructuring</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Swapping JavaScript Variables</h4>
<p id="demo"></p>

<script>
let firstName = "John";
let lastName = "Doe";

document.getElementById("demo").innerHTML =
  "Before: " + firstName + " " + lastName + "<br>";

// Swapping variables with destructuring
[firstName, lastName] = [lastName, firstName];

document.getElementById("demo").innerHTML +=
  "After: " + firstName + " " + lastName;
</script>

</body>
</html>
```

Example 6

Result View Example

Document

Document in project

You can Download PDF file.

Reference

- W3Schools JavaScript Destructuring