JavaScript destructuring is a feature that allows you to extract data from arrays and objects and assign them to variables. This can be a more concise and readable way to write code, especially when you need to access multiple properties from an object or multiple elements from an array.

To use destructuring, you simply place the variables you want to assign to within curly braces next to the object or array you want to extract data from. For example, the following code destructures an object with the properties **name** and **age** into two variables, **firstName** and **age**:

Explain

const { name: firstName, age } = { name: 'John Doe', age: 30 };  
  
console.log(firstName); *// 'John Doe'*  
console.log(age); *// 30*

Use code with caution.

[Learn more](https://support.google.com/legal/answer/13505487?hl=en-IN)

You can also use destructuring to extract multiple elements from an array. For example, the following code destructures an array with the elements **1**, **2**, and **3** into three variables, **a**, **b**, and **c**:

Explain

const [a, b, c] = [1, 2, 3];  
  
console.log(a); *// 1*  
console.log(b); *// 2*  
console.log(c); *// 3*

Use code with caution.

[Learn more](https://support.google.com/legal/answer/13505487?hl=en-IN)

Destructuring can also be used to extract nested properties from objects. For example, the following code destructures the nested property **address.city** from an object into a variable called **city**:

Explain

const { address: { city } } = { address: { city: 'San Francisco' } };  
  
console.log(city); *// 'San Francisco'*

Use code with caution.

[Learn more](https://support.google.com/legal/answer/13505487?hl=en-IN)

Destructuring is a powerful feature that can make your code more concise and readable. It is especially useful when you need to access multiple properties from an object or multiple elements from an array.