



# Destructuring

## in JavaScript





Destructuring is a JavaScript feature that allows you to **extract the properties** of an object or array into **individual variables**.

This can be useful for making your code more readable and concise.

Swipe to see some examples!



Swipe →



# Destructuring Arrays

Use array destructuring to extract values



```
const numbers = [1, 2, 3, 4, 5];  
  
const [first, second, ...rest] = numbers;  
  
console.log(first);  
console.log(second);  
console.log(rest);
```

output

1

2

```
// [object Array] (3)  
[3,4,5]
```



Swipe →



# Destructuring Objects

destructure objects using a similar syntax



```
const person = {  
  firstName: 'John',  
  lastName: 'Doe',  
  age: 30,  
  address: {  
    street: '123 Main St',  
    city: 'Anytown',  
    country: 'USA'  
  }  
};
```

```
const { firstName, lastName, address: { city } } = person;
```

```
console.log(firstName);  
console.log(lastName);  
console.log(city);
```

output

"John"

"Doe"

"Anytown"



Swipe →



# Default Values

You can provide default values while destructuring, which will be used if the value is undefined.



```
const { firstName, lastName, role = 'user',  
isActive = true } = person;
```

```
console.log(firstName);  
console.log(lastName);  
console.log(role);  
console.log(isActive);
```

"John"

"Doe"

"user"

true

default value used because  
**role** and **isActive** doesn't  
exist in **person**



Swipe →



# Destructuring Function Parameters

Destructuring is often used with function parameters to make the function calls cleaner and more expressive



```
function printPersonInfo({ firstName, lastName }) {  
  console.log(`Name: ${firstName} ${lastName}`);  
}
```

```
printPersonInfo(person);
```

output

```
"Name: John Doe"
```



Swipe →



# Nested Destructuring

Destructure nested arrays and objects



```
const nestedData = [1, [2, 3], [4, 5]];

const [firstValue, [secondValue, thirdValue],
      [fourthValue, fifthValue]] = nestedData;

console.log(secondValue);
console.log(fourthValue);
```

2

4

output



Swipe →





codewith**sloba**.com

Get a weekly digest of my tips and  
tutorials by subscribing now.