

JS

# Destructuring

## in JavaScript



like and  
share

Destructuring is used to  
**disassemble** entire arrays  
and objects to **merge** them  
or to **override** certain props.



# Now, how to destructure an array?

Here's an example



```
const fruits = ['apple', 'pineapple' , 'banana'];
const [x, y, z] = fruits;
```

When destructuring arrays, the order in which variables are declared is important.

Here **x** takes **apple**, **y** takes **pineapple** and **z** takes **banana**.



If you want to extract a particular value only from a particular position then leave the rest of the positions blank separated by commas.



```
const fruits = ['apple', 'pineapple' , 'banana'];  
const [x, , z] = fruits;
```

Here **x** takes apple and **z** takes banana.

Objects can also be destructured in js.



```
const vehicle = {  
    brand: 'Lamborghini',  
    model: 'Aventdor',  
    type: 'car',  
    year: 2029,  
    color: 'black'  
}  
  
car(vehicle);  
  
function car({type, color, brand, model})  
{  
    const message = 'My ' + type + ' is a '  
    + color + ' ' + brand + ' ' + model + '.';  
    console.log(message);  
}
```

"My car is a black Lamborghini Aventdor."



Swipe →

We can even destructure deeply nested objects.



```
const vehicle = {  
    brand: 'Lamborghini',  
    model: 'Aventdor',  
    type: 'car',  
    year: 2029,  
    color: 'black',  
    registration: {  
        city: 'Belgrade',  
        country: 'Serbia',  
    },  
}
```

← consider this  
nested object



By referencing the nested object and then using a colon and curly braces to again destructure the items needed from the nested object.



```
car(vehicle)
```

```
function car({ model, registration:  
{ city } }) {  
  const reg = 'My ' + model + ' is  
registered in ' + city + '.';  
  console.log(reg);  
}
```

"My Aventdor is registered in Belgrade."



Swipe →

# Do you find it helpful?

let me know down in the  
comments !



## Slobodan Gajić

Content Creator



FOLLOW FOR MORE