

You have 1 free member-only story left this month. [Sign up for Medium and get an extra one](#)

How to Destructure Nested JavaScript Objects

How to use destructuring in JavaScript to access deeply nested data



Jay Cruz

[Follow](#)



Oct 23, 2021 · 1 min read ★

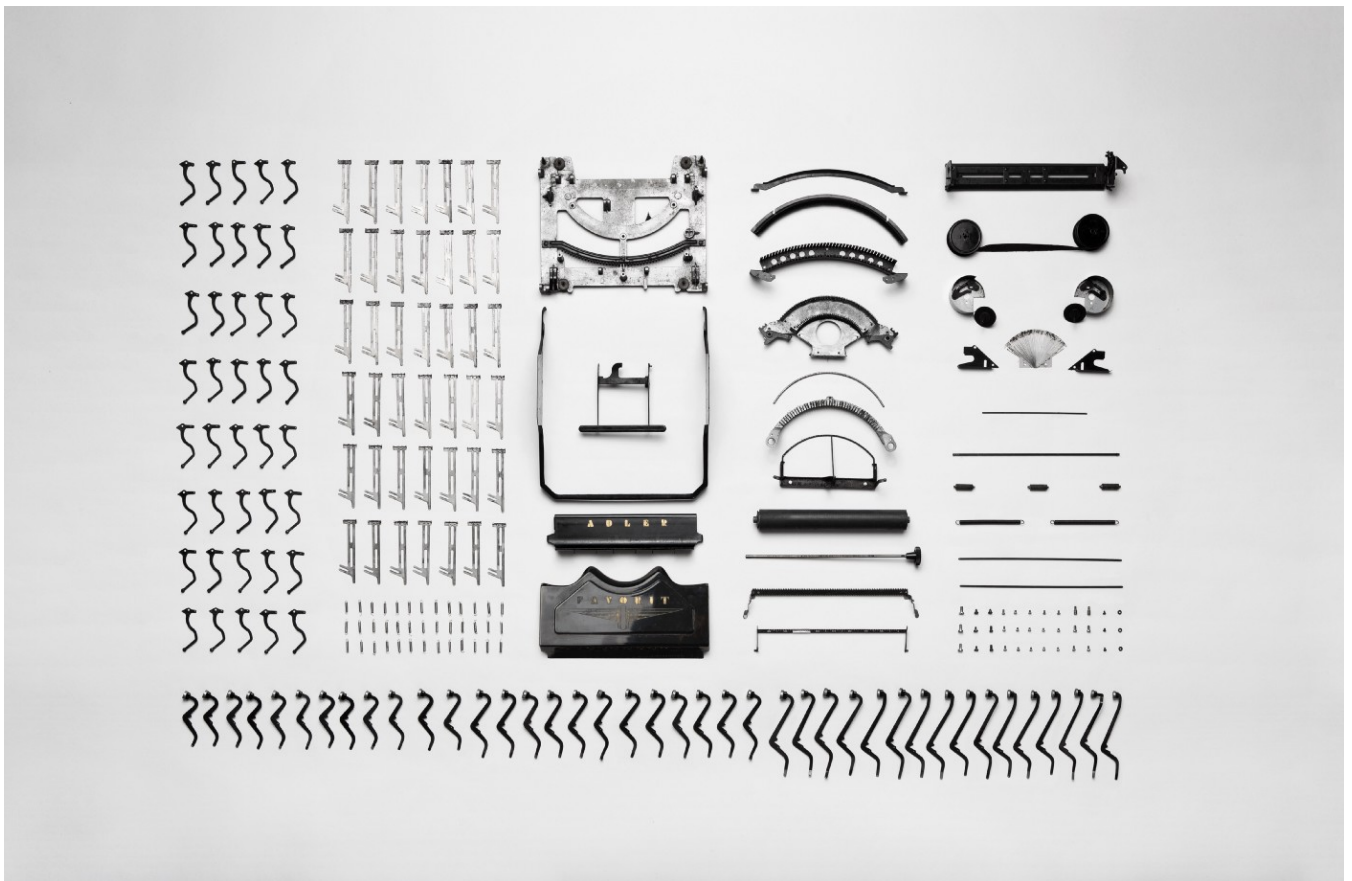


Photo by [Florian Klauer](#) on [Unsplash](#)

How to destructure objects

Let's first go over how to destructure objects in JavaScript. We can start with this simple example of an object representing an employee.

```
1  const employee = {  
2    id: 1,  
3    name: "John Doe"
```

```
3     name: "John Doe",
4     occupation: "Programmer"
5 };
6 const { id, name, occupation } = employee;
```

object_destructure.js hosted with ❤ by GitHub

[view raw](#)

This gives us access to all of the properties within the `employee` object.

```
1 console.log(id); // 1
2 console.log(name); // John Doe
3 console.log(occupation); // Programmer
```

object_destructure.js hosted with ❤ by GitHub

[view raw](#)

Pretty simple, right?

But what if we have to destructure an object that's a bit more complex? Maybe one with several levels, meaning objects within objects.

Destructuring nested objects

Now let's say we need to access data from an object representing several employees.

```
1 const employees = {
2   engineers: {
3     0: {
4       id: 1,
5       name: "John Doe",
6       occupation: "Back-end Engineer"
7     },
8     1: {
9       id: 2,
10      name: "Jane Doe",
11      occupation: "Front-end Engineer"
12    },
13  }
14 };
```

nested_object_destructure.js hosted with ❤ by GitHub

[view raw](#)

Here we have our `employees` object nested several levels deep. If we need to access an employee's info we can destructure as many levels as it takes to get to our employee object's properties.

```
1  const {  
2    engineers: {  
3      1: {  
4        id,  
5        name,  
6        occupation,  
7      },  
8    },  
9  } = employees;
```

nested_object_destructure.js hosted with ❤ by GitHub

[view raw](#)

Now we have access to all of the second employee object's properties.

```
1  console.log(id); // 2  
2  console.log(name); // Jane Doe  
3  console.log(occupation); // Front-end Engineer
```

nested_object_destructure.js hosted with ❤ by GitHub

[view raw](#)

Summary

Destructuring is a very useful feature that was added to the ES6 version of JavaScript. With destructuring, we can quickly and conveniently extract out properties or data from objects and arrays into separate variables.

This was just a brief overview of how to use destructuring to extract out and access data from nested JavaScript objects.

More content at plainenglish.io

[JavaScript](#) [Javascript Development](#) [ES6](#) [Programming](#) [Web Development](#)

[About](#) [Write](#) [Help](#) [Legal](#)

Get the Medium app



