


Day 14, Today's Topic

# Destructuring Part 1

DAILY  
JS



```
/* ===== */
/* ===== Daily JS - Day 14 ===== */
/* ===== */
```

## # Day 14: Destructuring Part 1

The concept of `Destructuring` is very useful, and quite big too. This is a really amazing feature which came with ES6, and many people think of it just as extracting variables from object, but there is much more to it.

Being a big topic, we will divide it in days instead of reading about it in a single day.


Explaining in simple words, `destructuring` allows us to extract elements from array or properties from objects.

JS Devs use destructuring quite frequently in their code, and it's quite simple to understand, plus it has many uses, and we will have a look at each of them :)

This feature (again) helps us reduce repetitive elements in our code.

Today let's look at the most basic feature of destructuring, pulling out properties from objects.

```
/* ===== madhavbahl.tech/dailyjs/day14 ===== */
/* ===== Join Discord: madhavbahl.tech/discord-c2e ===== */
```



```
/* ===== */
/* ===== Daily JS - Day 14 ===== */
/* ===== */
```

## ## 1. Pulling properties out of objects

### ### Syntax

#### **\*\*Syntax:\*\***

```
```js
var itemX = {
  costPrice: 100,
  sellingPrice: 150,
  profit: 50,
};

var { costPrice, sellingPrice, profit } = itemX;
// This way we can extract properties from an object.
console.log (costPrice, sellingPrice, profit);
```
```

```
/* ===== madhavbahl.tech/dailyjs/day14 ===== */
/* ===== Join Discord: madhavbahl.tech/discord-c2e ===== */
```



```
/* ===== */  
/* ===== Daily JS - Day 14 ===== */  
/* ===== */
```

## ## 1. Pulling properties out of objects

### ### Example

To see the importance of this case of destructuring, let's take a simple example.

**\*\*Make a greetings function which takes studentDetails object as argument and returns a greeting message\*\***

Let's solve this problem without destructuring and with destructuring, and let's compare :)

```
/* ===== madhavbahl.tech/dailyjs/day14 ===== */  
/* ≡ Join Discord: madhavbahl.tech/discord-c2e ≡ */
```

```
/* ===== */
/* ===== Daily JS - Day 14 ===== */
/* ===== */
```

## `Greetings` example without destructuring

```
```js
var studentDetails = {
  first_name: "Madhav",
  last_name: "Bahl",
  profession: "Student",
  age: 21
};

// Without Destructuring
function greetings (studentDetails) {
  var first_name = studentDetails.first_name;
  var last_name = studentDetails.last_name;
  var profession = studentDetails.profession;
  var age = studentDetails.age;

  return `Hi, I am ${first_name} ${last_name}.
    I am a ${profession}
    My age is ${age}`
}
```
```

Alright, you might be thinking, what's the problem with this?

To be honest, there is no huge problem with this syntax, but if we see it closely, we find that we are repeating `studentDetails.<...>` 4 times. What if there was a way by which we could combine those 4 lines and directly extract the object properties? What if I make a little spelling mistake in any of the 4 lines, say, I write `studentDetail` instead of `studentDetails`? Of course, it will throw an error.

Actually there is a way to condense those 4 lines into a single line, and that's what destructuring is all about. Let's have a look

```
/* ===== madhavgbahl.tech/dailyjs/day14 ===== */
/* ===== Join Discord: madhavgbahl.tech/discord-c2e ===== */
```

```
/* =====  
## ===== Daily JS - Day 14 =====  
## =====  
*/  
## `Greetings` example with destructuring
```

**\*\*Question:\*\*** Make a greetings function which takes studentDetails object as argument and returns a greeting message.

```
```js  
var studentDetails = {  
  first_name: "Madhav",  
  last_name: "Bahl",  
  profession: "Student",  
  age: 21  
};  
  
function greetings2 (studentDetails) {  
  var { first_name, last_name, profession, age } = studentDetails;  
  
  return `Hi, I am ${first_name} ${last_name}.  
    I am a ${profession}  
    My age is ${age}`  
}  
```
```

As you can see, this reduced the 4 lines where we extracted the properties into 1.

But that's not it, we can reduce it further, we can directly destructure the object in the arguments, which removes that 1 line as well.

We will see the examples of destructuring arguments in upcoming days. Stay Tuned :)

```
/* ===== madhavbahl.tech/dailyjs/day14 =====  
## ===== Join Discord: madhavbahl.tech/discord-c2e =====  
*/
```

```
/* =====  
## ===== Daily JS - Day 14 =====  
## =====  
*/  
  
var studentDetails = {  
  first_name: "Madhav",  
  last_name: "Bahl",  
  profession: "Student",  
  age: 21  
};  
  
// Without Destructuring  
function greetings1 (studentDetails) {  
  var first_name = studentDetails.first_name;  
  var last_name = studentDetails.last_name;  
  var profession = studentDetails.profession;  
  var age = studentDetails.age;  
  
  return `Hi, I am ${first_name} ${last_name}.  
    I am a ${profession}  
    My age is ${age}`  
}  
console.log ("Without destructuring: ", greetings1(studentDetails));  
  
// With Destructuring  
function greetings2 (studentDetails) {  
  var { first_name, last_name, profession, age } = studentDetails;  
  
  return `Hi, I am ${first_name} ${last_name}.  
    I am a ${profession}  
    My age is ${age}`  
}  
console.log ("Without destructuring: ", greetings2(studentDetails));  
  
/* ===== madhavbahl.tech/dailyjs/day14 =====  
## ===== Join Discord: madhavbahl.tech/discord-c2e =====  
*/
```

# Thank you!

**Feel free to reach out...**

Email: [theleanprogrammer@gmail.com](mailto:theleanprogrammer@gmail.com)

Web: [madhavbahl.tech/](http://madhavbahl.tech/)

Github: [github.com/MadhavBahlMD](https://github.com/MadhavBahlMD)

LinkedIn: [linkedin.com/in/madhavbahl/](https://linkedin.com/in/madhavbahl/)

Insta: [instagram.com/theleanprogrammer/](https://instagram.com/theleanprogrammer/)

**DAILY**  
**JS**