



Boston University  
Web Design and Development Series

# DESTRUCTURING

---

Prof. Andrew Sheehan



A scenic landscape photograph of a mountain valley. In the foreground, a calm lake reflects the surrounding scenery. The middle ground is filled with a dense forest of evergreen trees, interspersed with bright yellow wildflowers. In the background, majestic mountains with patches of snow rise against a clear blue sky.

# Destructuring

---

A technique that allows you to extract values from & objects into distinct values



# Types

---

There are 2

## Array destructuring

```
const [a, b, c] = [1, 2, 3];
```

## Object destructuring

```
const { name, age } = {name: 'Andrew', age: 21};
```





# Extraction

---

A simple and elegant approach to pulling out data that you want into variables

```
const [score, ID, URL] =  
[88.18, 'UI23456789',  
'http://bu.edu/outcomes']
```



# Nothing in your array?

---

When you destructure it, your assignment will have undefined

```
let [x] = []
```

x = undefined

```
let { age : a } = {}
```

age = undefined



# Cannot destructure null or undefined

---

When you try, you will see: “Uncaught TypeError: Cannot destructure property <property> of 'undefined' or 'null'”

```
let { value } = null
```

```
let { value } = undefined { }
```



# Selective

---

You are free to choose what you want to take on your value assignments

$\text{let } \{ x : y \} = \{ x : 3, y : 7 \}$

The value of x on the right side of this expression takes the value from y (7)



```
const contact = {  
  surname : "Andrew",  
  lastname: "Sheehan",  
  age: 21  
}  
let { surname, age } = contact;
```

## Taking the variables you want

Your object literals can be statically defined, pulled from a RESTful API call or even dynamically created on events



# Default values by destructuring

When a property doesn't have a value, you can assign a default

```
const contact = {
```

```
  first: "Andrew", last: "Sheehan", age: 21
```

```
}
```

```
let nickname = contact.nickname || "Does not have";
```



# Destructuring with function return values

```
function getAddressKeyMapping() {  
    return [  
        'e7x',  
        '2gii',  
        '9mt5'  
    ];  
}
```

```
let line1, city, postalCode;  
[line1, city, postalCode] = getAddressKeyMapping()
```



# Destructuring data

```
const wes = {  
  first: 'Wes',  
  last: 'Bos',  
  links: {  
    social: {  
      twitter: 'https://twitter.com/wesbos',  
      facebook: 'https://facebook.com/wesbos.developer',  
    },  
    web: {  
      blog: 'https://wesbos.com'  
    }  
  }  
};
```

Use this

```
const { twitter, facebook } = wes.links.social;  
console.log(twitter, facebook); // logs the 2 variables
```



# Another example

---

```
const person = {  
  first: 'Wes',  
  last: 'Bos',  
  country: 'Canada',  
  city: 'Hamilton',  
  twitter: '@wesbos'  
};  
const first = person.first;  
const last = person.last;
```

```
let { first, last } = person;
```



# Examples

Other ways to use destructuring

```
1  var a, b, rest;
2  [a, b] = [10, 20];
3  console.log(a); // 10
4  console.log(b); // 20
5
6  [a, b, ...rest] = [10, 20, 30, 40, 50];
7  console.log(a); // 10
8  console.log(b); // 20
9  console.log(rest); // [30, 40, 50]
10
11 ({ a, b } = { a: 10, b: 20 });
12 console.log(a); // 10
13 console.log(b); // 20
14
15
16 // Stage 3 proposal
17 ({a, b, ...rest} = {a: 10, b: 20, c: 30, d: 40});
18 console.log(a); // 10
19 console.log(b); // 20
20 console.log(rest); //{c: 30, d: 40}
```