ES6 Destructuring

Destructuring is a powerful feature in ES6 that allows you to extract values from arrays and objects in a concise and elegant way.

1. Array Destructuring

- **Basic Syntax:**
  JavaScript

```
const [a, b] = [10, 20];
console.log(a); // Output: 10
console.log(b); // Output: 20
```

- **Skipping Elements:**
  JavaScript

```
const [a, , b] = [10, 20, 30];
console.log(a); // Output: 10
console.log(b); // Output: 30
```

- **Rest Parameter:**
  JavaScript

```
const [a, ...rest] = [10, 20, 30, 40];
console.log(a); // Output: 10
console.log(rest); // Output: [20, 30, 40]
```

- **Default Values:**
  JavaScript

```
const [a = 1, b = 2] = [10];
console.log(a); // Output: 10
console.log(b); // Output: 2
```

2. Object Destructuring

- **Basic Syntax:**
  JavaScript

```
const person = { name: 'John', age: 30 };
const { name, age } = person;
console.log(name); // Output: "John"
console.log(age);  // Output: 30
```

- **Renaming Variables:**
  JavaScript

```
const { name: personName, age } = person;
console.log(personName); // Output: "John"
```

- **Default Values:**
  JavaScript

```
const { name, age, city = 'New York' } = person;
console.log(city); // Output: "New York"
```

- **Nested Objects:**
  JavaScript

```
const address = { street: 'Main St', city: 'Anytown' };
const { name, address: { street } } = { name: 'John', address };
console.log(street); // Output: "Main St"
```

- **Rest Parameter:**
  JavaScript

```
const { name, ...rest } = person;
console.log(name); // Output: "John"
console.log(rest); // Output: { age: 30 }
```

Benefits of Destructuring:

- **Improved Readability:** Makes code more concise and easier to read.
- **Reduced Boilerplate:** Eliminates the need for multiple assignment statements.
- **Enhanced Flexibility:** Allows for easy extraction of specific values from arrays and objects.

In Summary

Destructuring is a powerful feature that enhances the readability and maintainability of your JavaScript code. By effectively utilizing array and object destructuring, you can write more concise and elegant code.