ANSWER 8-

Destructuring in ES6 allows you to extract values from objects and arrays into separate variables. It provides a concise and convenient way to unpack data. Here's a simple explanation of how you can destructure objects and arrays:

Destructuring Objects: To destructure an object, you use curly braces {} and assign them to a variable with the same name as the object's property you want to extract.

```
const person = { name: 'John', age: 30, city: 'New York' };

 // Destructuring the object

const { name, age, city } = person;

 console.log(name); // Output: John

 console.log(age); // Output: 30

 console.log(city); // Output: New York
```

By using destructuring, you create individual variables (name, age, city) and assign them the corresponding values from the object.

Destructuring Arrays: To destructure an array, you use square brackets [] and assign them to variables in the order of the array's elements.

```
 const numbers = [1, 2, 3];

// Destructuring the array const [first, second, third] = numbers;
console.log(first); // Output: 1

console.log(second); // Output: 2

 console.log(third); // Output: 3
```

Here, the variables first, second, and third are assigned the corresponding values from the array.

Default Values: You can also provide default values during destructuring in case the object property or array element is undefined or missing.

```
const person = { name: 'John', age: 30 };

const { name, age, city = 'New York' } = person;

console.log(city); // Output: New York (default value)

 const numbers = [1, 2];

const [first, second, third = 3] = numbers;

console.log(third); // Output: 3 (default value)
```

In these examples, if the property or element is not defined, the default value is used instead. Destructuring allows you to extract values from objects and arrays with a more concise syntax. It simplifies the process of unpacking data and assigning it to individual variables, improving code readability and reducing the need for manual indexing or accessing object properties by name.