OBJECT DESTRUCTURING (AYŞENUR)

Object destructuring in JavaScript allows you to extract properties from an object and assign them to variables in a more concise way. It provides a convenient syntax for unpacking values from objects and accessing them directly.

Here's a simple explanation with examples:

1. Basic Object Destructuring:

```
const person = {
  name: 'John',
  age: 25,
  city: 'New York'
};

// Extracting properties using object destructuring
const { name, age, city } = person;

console.log(name); // Output: John
console.log(age); // Output: 25
console.log(city); // Output: New York
```

2. Renaming Variables:

```
const car = {
  brand: 'Toyota',
  model: 'Camry'
};

// Renaming variables using object destructuring
const { brand: carBrand, model: carModel } = car;

console.log(carBrand); // Output: Toyota
console.log(carModel); // Output: Camry
```

3. Default Values:

```
const person = {
```

```
  name: 'John',
  age: 25
};

// Providing default values using object destructuring
const { name, age, city = 'New York' } = person;

console.log(name); // Output: John
console.log(age); // Output: 25
console.log(city); // Output: New York
```

4. Nested Object Destructuring:

```
const student = {
  name: 'Alice',
  age: 20,
  address: {
    city: 'London',
    country: 'UK'
  }
};

// Extracting nested properties using object destructuring
const { name, address: { city, country } } = student;

console.log(name); // Output: Alice
console.log(city); // Output: London
console.log(country); // Output: UK
```

Object destructuring simplifies the process of extracting values from objects and assigning them to variables. It provides a concise and readable way to access object properties without repetitive dot notation.