

REDUCE METODU KULLANIM ÖRNEKLERİ

1. **Toplama:** Dizi içindeki tüm öğelerin toplamını bulabilirsiniz.

```
const numbers = [1, 2, 3, 4, 5];
const sum = numbers.reduce((accumulator, currentValue) => accumulator + currentValue, 0);
console.log(sum); // 15
```

2. **Çarpma:** Dizi içindeki tüm öğeleri birbiriyle çarparak bir çarpım bulabilirsiniz.

```
const numbers = [1, 2, 3, 4, 5];
const product = numbers.reduce((accumulator, currentValue) => accumulator * currentValue, 1);
console.log(product); // 120
```

3. **Filtreleme:** Belirli bir koşulu karşılayan öğeleri filtreleyebilir ve bir alt küme oluşturabilirsiniz.

```
const numbers = [1, 2, 3, 4, 5];
const evenNumbers = numbers.reduce((accumulator, currentValue) => {
  if (currentValue % 2 === 0) {
    accumulator.push(currentValue);
  }
  return accumulator;
}, []);
console.log(evenNumbers); // [2, 4] /// çift sayı filtreleme
```

4. **Nesne Oluşturma:** Bir dizi öğesinden bir nesne oluşturabilirsiniz.

```
const numbers = [1, 2, 3, 4, 5];
const numberObject = numbers.reduce((accumulator, currentValue, index) => {
  accumulator[`number_${index}`] = currentValue;
  return accumulator;
}, {});
console.log(numberObject); // { number_0: 1, number_1: 2, number_2: 3, number_3: 4, number_4: 5 }
```

5. **Gruplama:** Bir dizi öğesini belirli bir özelliğe göre gruplayabilirsiniz.

```
const data = [
  { name: 'Alice', age: 30 },
  { name: 'Bob', age: 25 },
  { name: 'Charlie', age: 30 },
];

const groupedByAge = data.reduce((accumulator, currentValue) => {
  const key = currentValue.age;
  if (!accumulator[key]) {
    accumulator[key] = [];
  }
  accumulator[key].push(currentValue);
  return accumulator;
}, {});
console.log(groupedByAge);
// {
//   25: [{ name: 'Bob', age: 25 }],
//   30: [{ name: 'Alice', age: 30 }, { name: 'Charlie', age: 30 }]
// }
```

6. **Minimum ve Maksimum Bulma:** Dizi içindeki en küçük veya en büyük değeri bulabilirsiniz.

```
const numbers = [3, 1, 7, 5, 2];
const min = numbers.reduce((accumulator, currentValue) => Math.min(accumulator, currentValue),
Infinity);
console.log(min); // 1

const max = numbers.reduce((accumulator, currentValue) => Math.max(accumulator, currentValue),
-Infinity);
console.log(max); // 7
```

7. **String Birleştirme:** Dizi içindeki string değerleri birleştirebilirsiniz.

```
const words = ['Merhaba', 'dünya', '!'];
const concatenated = words.reduce((accumulator, currentValue) => accumulator + ' ' + currentValue,
'');
console.log(concatenated); // Merhaba dünya !
```

8. **Nesne Dizisi İşlemleri:** Nesnelerden oluşan bir dizide belirli bir özelliği toplayabilir veya özelliğe göre filtreleme yapabilirsiniz.

```
const people = [
  { name: 'Alice', age: 30 },
  { name: 'Bob', age: 25 },
  { name: 'Charlie', age: 35 },
];

// Yaşların toplamını bulma
const totalAge = people.reduce((accumulator, currentValue) => accumulator + currentValue.age, 0);
console.log(totalAge); // 90

// 30 yaşından büyük kişileri filtreleme
const over30 = people.reduce((accumulator, currentValue) => {
  if (currentValue.age > 30) {
    accumulator.push(currentValue);
  }
  return accumulator;
}, []);
console.log(over30); // [{ name: 'Charlie', age: 35 }]
```

