JavaScript - 03

🔄 Transformation Methods

Map

```
const numbers = [1, 2, 3, 4];
const users = [{name: "John", age: 25}, {name: "Jane", age: 30}];

// Basic
numbers.map(n => n * 2);              // [2, 4, 6, 8]

// With objects
users.map(user => user.name);         // ["John", "Jane"]
users.map(user => ({...user, adult: user.age >= 18}));

// With index
numbers.map((n, i) => `${i}: ${n}`);  // ["0: 1", "1: 2", "2: 3", "3: 4"]
```

Filter

```
const products = [
  {name: "Laptop", price: 1000, category: "tech"},
  {name: "Book", price: 20, category: "education"},
  {name: "Phone", price: 800, category: "tech"}
];

// Basic
numbers.filter(n => n > 2);            // [3, 4]

// With objects
products.filter(p => p.price > 500);  // Laptop, Phone
products.filter(p => p.category === "tech");

// Combined
products.filter(p => p.price > 100 && p.category === "tech");
```

FlatMap - Map + Flatten

```
const sentences = ["Hello world", "JavaScript rocks"];

// Separate words
sentences.flatMap(s => s.split(" "));
// ["Hello", "world", "JavaScript", "rocks"]

// With numbers
[1, 2, 3].flatMap(n => [n, n * 2]);
// [1, 2, 2, 4, 3, 6]

// Filter and transform
users.flatMap(user => user.age > 25
? [user.name.toUpperCase()]
: []);
```

Reduce

```
const scores = [85, 92, 78, 96];

// Sum
scores.reduce((sum, score) => sum + score, 0);        // 351

// Average
scores.reduce((sum, score, i, arr) => {
  sum += score;
  return i === arr.length - 1 ? sum / arr.length : sum;
}, 0);

// Accumulator object
products.reduce((acc, product) => {
  acc[product.category] = (acc[product.category] || 0) + 1;
  return acc;
}, {}); // {tech: 2, education: 1}

// Max
scores.reduce((max, score) => score > max ? score : max);
```

Chaining

```
const data = [1, 2, 3, 4, 5, 6];

data
  .filter(n => n % 2 === 0)    // [2, 4, 6]
  .map(n => n * 3)              // [6, 12, 18]
  .reduce((sum, n) => sum + n, 0); // 36

// Complex example
products
  .filter(p => p.category === "tech")
  .map(p => ({...p, discounted: p.price * 0.9}))
  .reduce((total, p) => total + p.discounted, 0);
```