JavaScript - 02

Array Creation & Access

Creation

```
const fruits = ["apple", "banana", "orange"];
const numbers = new Array(1, 2, 3);
```

Access

```
console.log(fruits[0]); // apple
console.log(fruits.length); // 3
```

🏗️ Array Creation

```
Array.from("hello");           // ["h", "e", "l", "l", "o"]
Array.of(1, 2, 3);            // [1, 2, 3]
new Array(3).fill(0);        // [0, 0, 0]
[...Array(3)].map((_, i) => i); // [0, 1, 2]
```

🔧 Adding/Removing Elements

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

// Add
arr.push(4);           // [1, 2, 3, 4] - end
arr.unshift(0);        // [0, 1, 2, 3, 4] - start
arr.splice(2, 0, 1.5);
// [0, 1, 1.5, 2, 3, 4] - anywhere

// Remove
arr.pop();             // [0, 1, 1.5, 2, 3] - end
arr.shift();           // [1, 1.5, 2, 3] - start
arr.splice(1, 1);      // [1, 2, 3] - anywhere
```

🔧 Utility Methods

```
const colors = ["red", "green", "blue"];

colors.forEach((color, i) => console.log(`${i}: ${color}`));
colors.map((color, i) => `${i}: ${color}`);
for (let color of colors) console.log(color);
```

🔍 Searching & Finding

```
const users = [{id: 1, name: "John"}, {id: 2, name: "Jane"}];


users.find(u => u.id === 1);        // {id: 1, name: "John"}
users.findIndex(u => u.name === "Jane"); // 1
users.indexOf("John");              // -1 (not found)
users.includes({id: 1});            // false (reference)
users.some(u => u.id > 0);          // true
users.every(u => u.id > 0);         // true
```

🔄 Transformation Methods

```
const nums = [1, 2, 3, 4];

nums.map(n => n * 2);               // [2, 4, 6, 8]
nums.filter(n => n > 2);           // [3, 4]
nums.reduce((sum, n) => sum + n, 0); // 10
nums.flatMap(n => [n, n * 2]);     // [1, 2, 2, 4, 3, 6, 4, 8]
```

📋 Iteration Methods

```
const colors = ["red", "green", "blue"];

colors.forEach((color, i) => console.log(`${i}: ${color}`));
colors.map((color, i) => `${i}: ${color}`);
for (let color of colors) console.log(color);
```