

Looping Through Arrays

(Real Data Processing in JavaScript)

Subtitle: Learn how to process real-world data using loops and understand arrays as the foundation of every modern application.

Website Name: haas.dev

Website Link: <https://dev-roast-app.vercel.app>

Introduction

In real applications, loops are not used for numbers.

They are used for **data processing**.

And most data comes in one structure:

Arrays

Examples:

- users list
- products
- posts
- orders
- API responses

If you don't understand array looping properly, you cannot build real applications.

Step 1: What is Array Looping?

Array looping means:

Accessing and processing each element of an array one by one.

Basic Example:

```
let users = ["Ali", "Sara", "Ahmed"];
```

```
for (let i = 0; i < users.length; i++) {  
  console.log(users[i]);  
}
```

```
}
```

Output:

Ali

Sara

Ahmed



Step 2: Why Arrays Need Loops

Arrays store multiple values, but:

- you cannot process them all at once
- you must access each element individually

So loops act as:

a bridge between data storage and data processing



Step 3: Standard Pattern (Most Important)

```
for (let i = 0; i < array.length; i++) {
```

```
  // process array[i]
```

```
}
```

Mental Model:

Start → index 0 → process → move next → repeat → end



Step 4: Real-World Examples

1. Product List Rendering

```
let products = ["Shoes", "Laptop", "Phone"];
```

```
for (let i = 0; i < products.length; i++) {
```

```
  console.log("Product:", products[i]);
```

```
}
```

2. User System Processing

```
let users = ["Ali", "Sara", "Ahmed"];
```

```
for (let i = 0; i < users.length; i++) {  
  console.log("Welcome", users[i]);  
}
```

3. API Data Simulation

```
let data = [10, 20, 30, 40];
```

```
let sum = 0;
```

```
for (let i = 0; i < data.length; i++) {  
  sum += data[i];  
}
```

```
console.log(sum);
```



Step 5: Loop Variations for Arrays

1. for loop (classic)

Full control

2. while loop

```
let i = 0;
```

```
while (i < 3) {  
  console.log(users[i]);  
  
  i++;  
}
```

```
}
```

3. for...of loop (modern)

```
for (let user of users) {
```

```
  console.log(user);
```

```
}
```

👉 Cleaner and preferred in modern JS



Step 6: Real Engineering Thinking

Array looping is used in:

- filtering data
- transforming data
- searching data
- aggregating data

Example: Filtering Active Users

```
let users = [
```

```
  { name: "Ali", active: true },
```

```
  { name: "Sara", active: false }
```

```
];
```

```
for (let user of users) {
```

```
  if (user.active) {
```

```
    console.log(user.name);
```

```
  }
```

```
}
```



Step 7: Common Mistakes



1. Using wrong index range

```
for (let i = 0; i <= array.length; i++)
```

👉 causes undefined access

❌ 2. Modifying array while looping

can break logic

❌ 3. Not using correct loop type

- use for...of for simplicity
- use for for control

🧪 Step 8: Mini Exercises

Exercise 1

Print all numbers in array multiplied by 2.

Exercise 2

Find sum of all elements.

Exercise 3

Print only even numbers.

🧠 Step 9: Mini Quiz

1. Why do we loop arrays?
2. Difference between for and for...of?
3. What happens if index exceeds length?
4. Where is array looping used in real systems?

🚀 Step 10: Thinking Upgrade

If you understand array looping:

- you understand data processing
- you can build backend logic
- you can handle API responses

👉 This is the core of real development.

📌 Step 11: Summary

- Arrays store multiple values
- Loops process each element

- Used in real applications everywhere
- Multiple loop types exist
- Core skill for data handling