Nested Loops in JavaScript

(2D Data + Grid Processing Logic)

Subtitle: Learn how loops inside loops work and how real systems process grids, tables, and multi-layered data structures.

Website Name: haas.dev
Website Link: https://dev-roast-app.vercel.app

🧠 Introduction

So far, you've used loops to process:

- single arrays
- linear data

But real systems are not always linear.

They often contain:

- tables
- grids
- matrices
- nested data

To handle this, JavaScript uses:

Nested loops (loop inside another loop)

⚙️ Step 1: What is a Nested Loop?

A nested loop is:

A loop inside another loop.

Basic Structure:

```
for (let i = 0; i < 3; i++) {

  for (let j = 0; j < 3; j++) {

    console.log(i, j);

  }

}
```

🔄 Step 2: Execution Flow (Critical Concept)

Outer loop starts

→ Inner loop runs fully

Outer loop moves next step

→ Inner loop runs again

Output Pattern:

0 0

0 1

0 2

1 0

1 1

1 2

2 0

2 1

2 2

🧠 Step 3: Why Nested Loops Exist

Used when data has **multiple dimensions**:

- rows and columns
- categories and items
- users and posts
- grids and maps

🌍 Step 4: Real-World Use Cases

1. Table/Grid Rendering

```
for (let row = 1; row <= 3; row++) {
  for (let col = 1; col <= 3; col++) {
    console.log(`Row ${row}, Col ${col}`);
```

```
  }
}
```

2. Matrix Processing

```
let matrix = [
  [1, 2],
  [3, 4]
];

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

3. User → Posts Structure

```
let users = [
  ["Post1", "Post2"],
  ["Post3", "Post4"]
];

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

🧠 Step 5: Mental Model

Outer loop = category

Inner loop = items inside category

⚙️ Step 6: Time Complexity Reality

Nested loops increase execution cost.

Example:

1 loop → O(n)

2 loops → $O(n^2)$

👉 This becomes slow with large data.

⚠️ Step 7: Common Mistakes

❌ 1. Confusing loop levels

- outer loop controls structure
- inner loop controls details

❌ 2. Infinite inner loop

wrong update logic breaks flow

❌ 3. Using nested loops unnecessarily

not all problems need 2 loops

❌ 4. Ignoring performance impact

large datasets → slow execution

🧪 Step 8: Mini Exercises

Exercise 1

Print multiplication table using nested loops.

Exercise 2

Print a 3x3 grid pattern.

Exercise 3

Traverse 2D array and sum all values.

🧠 Step 9: Mini Quiz

1. What is a nested loop?
2. Why is inner loop executed multiple times?
3. What is time complexity of nested loops?
4. Where are nested loops used in real systems?

🚀 Step 10: Thinking Upgrade

If you understand nested loops:

- you understand multi-dimensional data
- you can process complex structures
- you can build grid-based systems

👉 This is foundation for algorithms and data structures.

📌 Step 11: Summary

- Nested loops = loop inside loop
- Used for 2D and structured data
- Inner loop runs fully per outer iteration
- Powerful but performance-sensitive
- Core concept for DSA and real systems