📘 Lecture 4: Loops in JavaScript (for, while, do-while)

🧠 What Are Loops?

Loops let us **execute a block of code repeatedly** without writing it multiple times. This is extremely useful when:

- Printing a sequence of numbers
- Iterating through arrays
- Repeating tasks until a condition is false

Think of loops as saying:
“Do this thing X times, or until Y is true.”

🔁 Types of Loops in JavaScript

1. for loop
2. while loop
3. do...while loop

Let’s break down each one.

🔹 1. for Loop

Used when you **know exactly how many times** you want to loop.

🔸 **Syntax:**

javascript
CopyEdit

```
for (initialization; condition; increment) {
  // code block to run
}
```

✅ **Example:**

javascript
CopyEdit

```
for (let i = 1; i <= 5; i++) {
  console.log("Count: " + i);
}
```

🔍 **How It Works:**

- **Initialization**: let i = 1 (start value)
- **Condition**: i <= 5 (run while true)
- **Increment**: i++ (increase i by 1 each time)

🔹 2. while Loop

Use this when you're **not sure how many times** the loop should run — only when a condition is true.

🔸 **Syntax:**

javascript
CopyEdit

```
while (condition) {
  // code block to run
}
```

✅ **Example:**

javascript
CopyEdit

```
let i = 1;

while (i <= 5) {
  console.log("While Count: " + i);
  i++;
}
```

⚠️ **Warning:**
Make sure the condition eventually becomes false, or you'll create an **infinite loop**.

🔹 3. do...while Loop

This loop runs **at least once**, even if the condition is false the first time.

🔸 **Syntax:**

javascript
CopyEdit

```
do {
  // code block to run
} while (condition);
```

✅ **Example:**

javascript
CopyEdit

```
let i = 1;

do {
  console.log("Do While Count: " + i);
  i++;
} while (i <= 5);
```

🔍 Use Case:

When you want to run something **first**, then check if it should repeat.

🧮 Real-World Analogy

- for loop: “I will do this 5 times.” (You know how many.)
- while loop: “Keep doing this **while** there are customers in the line.”
- do...while: “Do this at least once, and then see if I should do it again.”

🔄 Infinite Loops

If the condition **never becomes false**, the loop keeps running forever and may crash your program.

javascript
CopyEdit

```
// ❌ Infinite loop example
while (true) {
  console.log("This will never stop!");
}
```

🔁 Loop Control Statements

🔸 break

Stops the loop immediately.

javascript
CopyEdit

```
for (let i = 1; i <= 10; i++) {
  if (i === 5) break;
  console.log(i);
}
```

```
// Output: 1 2 3 4
```

🔸 continue

Skips the current iteration.

javascript

CopyEdit

```
for (let i = 1; i <= 5; i++) {
  if (i === 3) continue;
  console.log(i);
}
// Output: 1 2 4 5
```

🧩 Practice Tasks

1. Print numbers 1 to 10 using for loop.
2. Print even numbers from 1 to 20 using a while loop.
3. Write a do...while loop that runs at least once even if the condition is false.
4. Create a loop that breaks when a number is divisible by both 3 and 5.
5. Use continue to skip printing number 7 from 1 to 10.

🧠 Summary

Loop Type	Use When
for	You know the number of iterations
while	Repeat while a condition is true
do...while	Ensure code runs **at least once**

Loops are **fundamental** to programming and power most real-world tasks — like rendering lists, handling users, and managing games or APIs.