

□ What is a Loop?

A loop in Java is used to execute a block of code repeatedly as long as a specified condition is true. It avoids code repetition and helps in writing cleaner, optimized programs.

□ Why Loops are Important

- Reduces code redundancy
- Saves time and effort
- Essential for iterating arrays, strings, input sets
- Used heavily in logic-building and algorithm problems

□ Types of Loops in Java

Loop Type	Condition Check	Minimum Executions	Used When
for loop	Before body	Can be zero	You know how many times to run
while loop	Before body	Can be zero	You don't know how many times
do-while	After body	At least once	Must run once before checking

□ 1. for Loop – Most Common

Syntax:

java

CopyEdit

```
for(initialization; condition; update) { // body }
```

Example:

java

CopyEdit

```
for(int i = 1; i <= 5; i++) { System.out.println(i); }
```

□ How it works:

1.

Initialize $i=1$

2.

Check condition $i \leq 5$ – true

3.

Execute body

4.

Increment $i++$

5.

Repeat until condition false

□ Used when number of iterations is known.

□ 2. while Loop – Conditional Loop

Syntax:

java

CopyEdit

```
while(condition) { // body }
```

Example:

java

CopyEdit

```
int i = 1; while(i <= 5) { System.out.println(i); i++; }
```

□ Used when you don't know how many times to loop, but have a condition to check.

□ 3. do-while Loop – Post-check Loop

Syntax:

java

CopyEdit

```
do { // body } while(condition);
```

Example:

java

CopyEdit

```
int i = 1; do { System.out.println(i); i++; } while(i <= 5);
```

□ Even if condition is false, it executes at least once.

□ Loop Flowchart (Conceptual)

mathematica

CopyEdit

Initialization → [Condition Check] → True Body of Loop → [Update/Increment] → (Back to Condition) → False → Exit Loop

→ Loop Control Statements

→ break;

Used to exit the loop immediately.

Example:

java

CopyEdit


```
for(int i = 1; i <= 10; i++) { if(i == 5) break; System.out.println(i); }
```

□ Output: 1 2 3 4

□ continue;

Skips the current iteration and goes to the next.

Example:

java

CopyEdit

```
for(int i = 1; i <= 5; i++) { if(i == 3) continue; System.out.println(i); }
```

□ Output: 1 2 4 5

□ Nested Loops

A loop inside another loop.

java

CopyEdit

```
for(int i = 1; i <= 3; i++) { for(int j = 1; j <= 2; j++) { System.out.println(i + " " + j); } }
```

□ Output:

CopyEdit

1 1 1 2 2 1 2 2 3 1 3 2

□ Used for patterns, matrices, tables, etc.

□ Common Mistakes in Loops

- - Missing condition □ infinite loop
- - Not updating loop variable □ stuck
-

- Off-by-one errors (like `<=` instead of `<`)

-

- Using `=` instead of `==` in condition

-

- Declaring variables inside loop unnecessarily

□ Best Practices

-

- Use `for` when the number of iterations is known

-

- Use `while` when looping based on condition/input

-

- Use `do-while` when you need at least one run

- Avoid deeply nested loops
- Always test boundary conditions
-

1. What is a loop in Java?

A loop is a control structure that repeats a block of code multiple times as long as a specified condition is true.

2. Difference between for, while, and do-while loops?

Feature	for	while	do-while
Condition	Checked before	Checked before	Checked after
Use Case	Known count	Unknown count	At least once run
Syntax Req.	Init, cond, inc all in one	Separate lines	Post-check block

3. Which loop executes at least once regardless of the condition?

do-while loop. Because the condition is checked after executing the body.

4. When to use while loop?

When the number of iterations is not known beforehand and depends on a condition (like user input).

5. What is an infinite loop? How to create one?

A loop that never terminates.

java

CopyEdit

```
while(true) { // infinite }
```

6. What is the difference between break and continue?

- break: Immediately exits the loop
- continue: Skips current iteration, continues the loop

7. What is a nested loop?

A loop inside another loop.

Example:

java

CopyEdit

```
for(int i=0;i<3;i++) { for(int j=0;j<2;j++) { System.out.println(i + " " + j); } }
```

8. Can we use a return statement inside a loop?

Yes. It terminates the method (not just the loop).

9. Can we use multiple variables in for loop?

Yes.

java

CopyEdit

```
for(int i=0, j=10; i<j; i++, j--) { System.out.println(i + " " + j); }
```

10. What is loop variable scope?

The loop variable (e.g. `int i` in `for` loop) is only accessible within the loop body.

11. Write a program to print even numbers from 1 to 20.

(See Programs section below)

12. Can you write a loop that counts backward?

Yes.

java

CopyEdit

```
for(int i = 10; i >= 1; i--) { System.out.println(i); }
```

13. What happens if we miss update statement in a for loop?

The loop may become infinite if condition never becomes false.

14. How to skip a loop iteration?

Use continue; statement.

15. What is the output?

java

CopyEdit

```
for(int i = 1; i <= 3; i++) { for(int j = 1; j <= 2; j++) { System.out.print("* "); } System.out.println(); }
```

Output:

markdown

CopyEdit

```
* * * * *
```

16. Can a for loop run forever?

Yes.

java

CopyEdit

```
for(;;) { // infinite }
```

17. Is using loops bad for performance?

Not generally, unless the loop runs unnecessarily or is nested too deeply.

18. How is do-while used in user-driven menus?

It ensures the menu runs at least once before user input is checked to exit.

19. Can we have a loop inside a switch-case?

Yes.

20. What is loop unrolling?

It's a performance optimization technique where iterations are manually expanded.
