ScopeTreasure Courses

JAVASCRIPT COURSES · LESSON 3.6 · CODE SCOPE SERIES

Loops: for & while

Repeat work without repeating code: the for loop's three parts, while, the infinite-loop trap, break vs continue, and loops that build real menus.

Introduction

Imagine printing five lines by hand: five separate `console.log` lines, typed one by one. Now imagine printing five hundred. A **loop** lets you write the work **once, as a rule**, and JavaScript repeats it for you — as many times as the rule says.

In this lesson you will take apart the `for` loop's three parts, watch it run step by step, count backwards, meet the `while` loop, and stare at the famous danger: the **infinite loop** that freezes your browser tab. You will finish with `break` versus `continue`, and a loop that builds real menu items.

(Roman Urdu: Loop matlab aik kaam ko baar baar karna — code aik dafa likho, JavaScript khud dohrata rahega.)

Key points

- **The `for` loop has three parts:** `for (start; condition; step)` — where counting begins, the rule that must stay true to keep going, and how the counter moves each round. Example shape: `for (let i = 1; i <= 5; i++)`.
- **How a round works:** check the condition → run the block → do the step → check again. The loop **stops by itself** the moment the condition turns false. With `i` from 1 while `i <= 5`, you get exactly five rounds: 1, 2, 3, 4, 5.
  (Roman Urdu: Har round mein pehle condition check hoti hai, phir code chalta hai, phir step badhta hai.)
- **You control the counting:** start high and step down (`i--`) to count backwards, or step by 2 (`i += 2`) to take every second number.
- **`while` keeps going while the condition holds:** `while (cups < 3) { ... }` — fill cups until full. Use it when you do not know the count in advance, only the stopping rule.
- **The loop variable MUST change inside a `while`:** if nothing inside the block moves the condition toward false, the loop never ends.
  (Roman Urdu: while loop ke andar condition ko badalna lazmi hai — warna loop kabhi rukega nahi.)
- **The infinite-loop trap:** a loop that never ends **freezes the browser tab**. The fix is almost always the missing step — make sure something like `i++` actually runs every round.
- **`break` leaves the loop at once; `continue` skips only this round.** Break = walk out of the room. Continue = skip this turn and go to the next round.
- **`for` vs `while`:** you know the count → `for`. You are waiting for a condition → `while`.

Code examples

Example 1 — The for loop, running live

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

console.log("Loop finished");
```

What it does: `i` starts at `1`. Each round prints, then `i++` raises `i` by one. When `i` becomes `6`, the condition `i <= 5` is false, the loop stops by itself, and the line after the loop runs.

What you will see: six lines in the Console: `Round 1`, `Round 2`, `Round 3`, `Round 4`, `Round 5`, and then `Loop finished`.

Example 2 — while: filling cups until full, with break and continue

```
let cups = 0;

while (cups < 5) {
  cups++;

  if (cups === 3) {
    continue;
  }

  if (cups === 5) {
    break;
  }

  console.log("Filled cup " + cups);
}

console.log("Done. Cups filled: " + cups);
```

What it does: The `while` loop fills one cup per round. On cup 3, `continue` skips the print but the loop carries on. On cup 5, `break` leaves the loop immediately — before the print. The counter changes inside every round, so this loop is safe.

What you will see: three lines in the Console: `Filled cup 1`, `Filled cup 2`, `Filled cup 4` — cup 3 was skipped, cup 5 broke out — and then `Done. Cups filled: 5`.

Example 3 — A loop building a real menu

```
let drinks = ["Chai", "Coffee", "Lassi", "Green Tea"];
let menuText = "";

for (let i = 0; i < drinks.length; i++) {
  menuText = menuText + (i + 1) + ". " + drinks[i] + "\n";
}

console.log(menuText);
```

What it does: A loop walks a list of drinks (arrays get their full lesson in 3.8) and builds one numbered menu line per drink. Notice `i + 1`: the loop counts from `0`, but humans number menus from `1`.

What you will see: one block in the Console — a numbered menu on four lines: `1. Chai`, `2. Coffee`, `3. Lassi`, `4. Green Tea`.

Quick recap

- `for (start; condition; step)` — the loop checks, runs, steps, and stops when the condition turns false.
- `while` runs until a condition changes — so the change **must** happen inside, or the tab freezes forever.
- `break` leaves the loop; `continue` skips one round. Known count → `for`; waiting on a condition → `while`.

Common mistakes

1. **The infinite loop** — forgetting the step (`i++`) or never changing the `while` condition means the loop runs forever and freezes the tab. Fix: before running, point at the exact line that moves the loop toward stopping.
2. **Off-by-one with the condition** — writing `i < 5` when you wanted five rounds starting at 1 gives only four. Fix: trace the first and last values of the counter on paper before you run.
3. **Confusing break and continue** — using `break` to "skip" an item actually ends the whole loop. Fix: skip one round = `continue`; leave completely = `break`.

Practice

1. Write a `for` loop that counts backwards from 10 to 1 and prints each number, then prints `Start!` after the loop. Change the step to `i -= 2` and note which numbers appear.
2. Write a `while` loop that starts with `let water = 0;` and adds `250` each round while `water < 1000`. Print the water level every round — and before running, write down how many rounds you expect.

Practice quiz

1. for (let i = 1; i <= 5; i--) runs the wrong way — what is the fix?

- Change i-- to i++, so the counter grows
- Change <= to >=
- Start i at 0
- Remove the step

Answer: Change i-- to i++, so the counter grows

2. What does this print?

```
for (let i = 1; i <= 3; i++) {
  console.log(i);
}
```

- 1 2 3
- 0 1 2
- 1 2 3 4
- 3 2 1

Answer: 1 2 3

3. What does this print?

```
let i = 0;
while (i < 2) {
  i++;
}
console.log(i);
```

- 2
- 0
- 1
- 3

Answer: 2

Watch the video lesson: https://youtu.be/ALsto_iE8-A

Read online: https://scopetreasure.com/notes/js-loops-for-while · More free tools and notes at https://scopetreasure.com