Loops — for, while, for...of, for...in

AiCE Software Engineering Foundations — JavaScript & Node.js

Conditionals are covered. Now let's move on to loops — how a JavaScript program repeats actions. JavaScript gives several different loop forms, each suited to a different situation, and this lesson covers all of them.

1. The for Loop

Python's for loop goes directly over the items in a list. JavaScript's classic for loop instead works with a counter — you control where it starts, when it stops, and how it moves forward.

Create a file named loops.js:

```
for (let i = 0; i < 5; i++) {
    console.log(`Iteration ${i}`);
}
```

Save and run — this prints Iteration 0 through Iteration 4.

Part	What It Does
let i = 0	The starting point — runs once, before the loop begins.
i < 5	The condition — checked before every loop.
i++	The update — runs after every loop.

This form is especially useful when you need the index while looping through an array.

2. while, do...while, for...of, for...in

Let's write more code in the same file:

```
let count = 1;
while (count <= 3) {
    console.log(count);
    count++;
}

let number = 10;
do {
    console.log(number);
    number++;
} while (number < 5);
```

while behaves exactly like Python's while — it keeps looping as long as the condition is true.

> **do...while Always Runs Once**
> *Notice that the do...while block printed 10, even though number < 5 was already false before it started. This is because do...while always runs the body once before it ever checks the condition — a subtle but important difference from a regular while loop.*

Now the two loops you will actually use the most:

```
const students = [{ name: 'Adesina', score: 85 }, { name:
'Kwame', score: 90 }];
for (const student of students) {
    console.log(`${student.name}: ${student.score}`);
}

const developer = { name: 'Chiamaka', level: 'Intermediate' };
for (const key in developer) {
    console.log(`${key}: ${developer[key]}`);
}
```

Loop	Use It For
for...of	The modern, clean way to loop over an array — the closest thing JavaScript has to Python's for item in list.
for...in	Loops over the keys of an object.

> **Don't Mix These Two Up**
> *Using for...in on an array gives you the index positions instead of the actual values, which is rarely what you want. Use for...of for arrays, for...in for objects.*

3. break, continue & Nested Loops

Let's write more code:

```
const talents = ['Adesina', 'Goodnews', 'STOP', 'Kwame'];

for (const talent of talents) {
    if (talent === 'STOP') break;
    if (talent === 'Goodnews') continue;
    console.log(talent);
}
```

Save and run — this prints 'Adesina', skips 'Goodnews' entirely because of continue, and stops completely once it reaches 'STOP', because of break — it never even looks at 'Kwame'.

Keyword	What It Does
break	Exits the loop immediately.
continue	Skips the current round and moves straight to the next one.

Loops can also be placed inside other loops — this is called nesting, and it works exactly as you already know from Python:

```
for (const country of ['Nigeria', 'Ghana']) {
    for (const level of ['Beginner', 'Advanced']) {
        console.log(`${country} - ${level}`);
    }
}
```

4. Practice: A Student Progress Report

Let's build a final project practicing everything from this lesson — a program generating a short progress report for a group of students. Create a file named progress_report.js:

```
// AiCE Student Progress Report
const students = [
    { name: 'Adesina Olagunju', scores: [85, 90, 78, 92] },
    { name: 'Fatima Zahra', scores: [55, 48, 62, 58] },
    { name: 'Kofi Otieno', scores: [95, 92, 98, 96] },
];

for (const student of students) {
    let total = 0;
    for (const s of student.scores) total += s;
    const avg = (total / student.scores.length).toFixed(1);
    const status = avg >= 50 ? 'PASS' : 'FAIL';
    console.log(`${student.name}: average ${avg} -
${status}`);
}
```

Save and run — this goes through each student, uses a nested loop to add up their scores, calculates their average, and prints whether they passed or failed, bringing together everything from this unit. Push your work to GitHub.

Key Takeaways

The classic for loop gives full control with a counter, while while and do...while repeat based on a condition — do...while always running its body at least once. for...of is the modern way to loop over arrays, for...in loops over an object's keys, and mixing them up

gives the wrong result. break exits a loop entirely, continue skips just the current round, and loops can nest inside each other exactly like in Python. With every decision-making and repetition tool now covered, the next lesson moves into functions — function declarations, expressions, arrow functions, default and rest parameters, and using functions as arguments to other functions.