

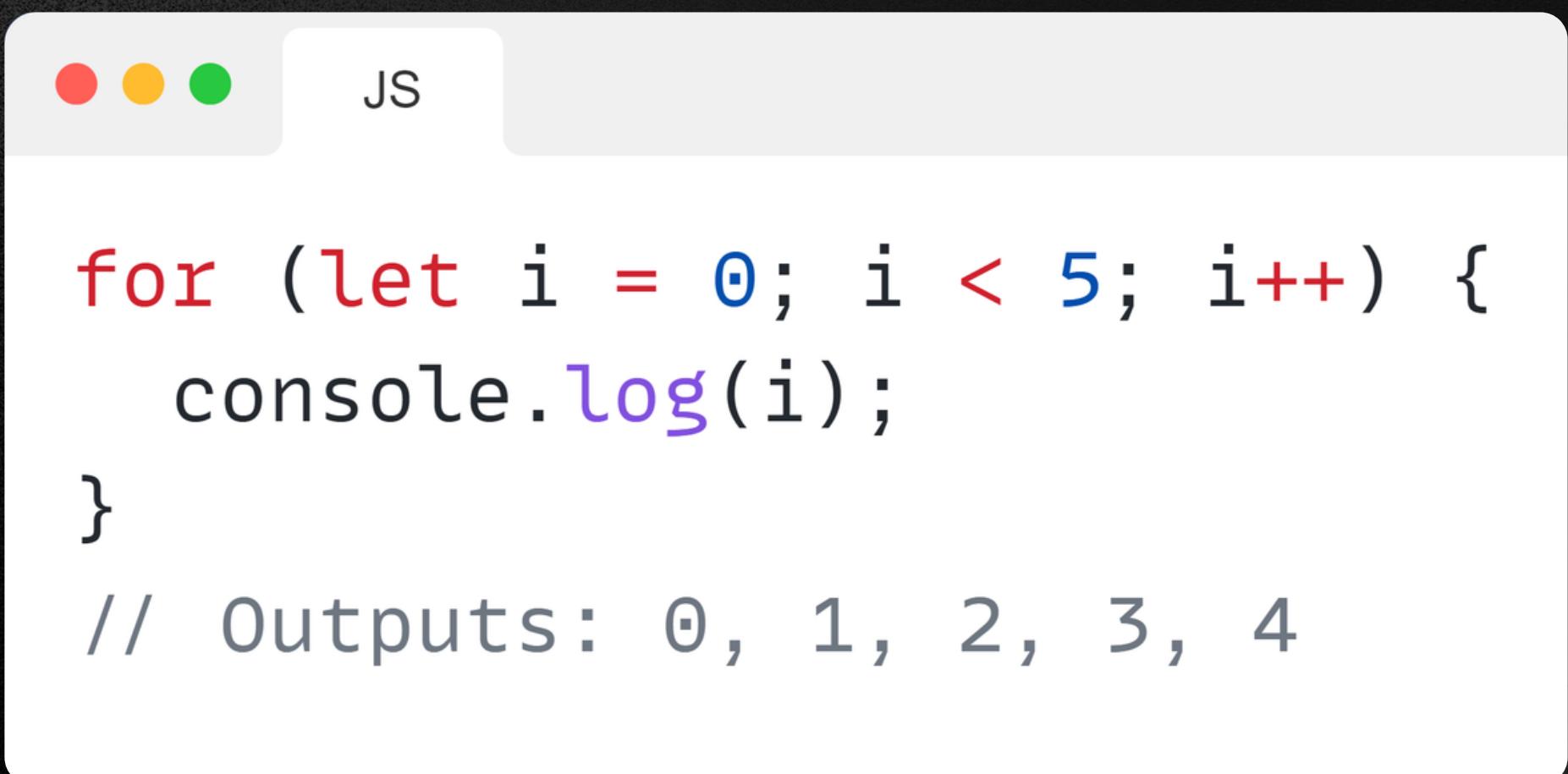


# Different Types of Loops in JavaScript



# The Classical For Loop

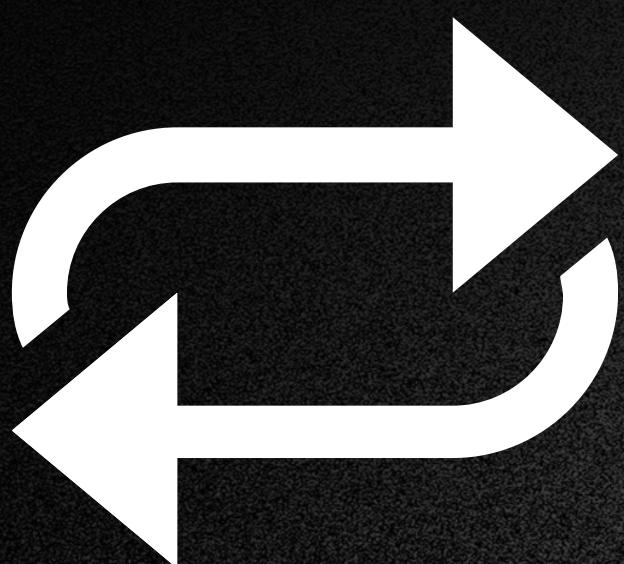
- The workhorse of loops, perfect for precise control over iteration.



A screenshot of a browser's developer tools console window. The title bar says "JS". The code in the console is:

```
for (let i = 0; i < 5; i++) {  
    console.log(i);  
}  
  
// Outputs: 0, 1, 2, 3, 4
```

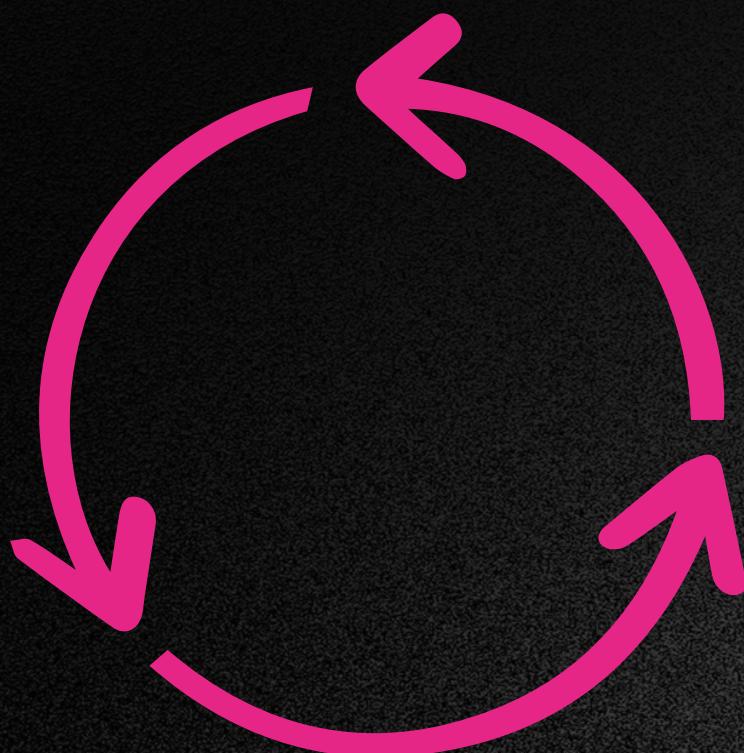
The output below the code shows the numbers 0, 1, 2, 3, 4, each on a new line, representing the iterations of the loop.



# for...of Loop

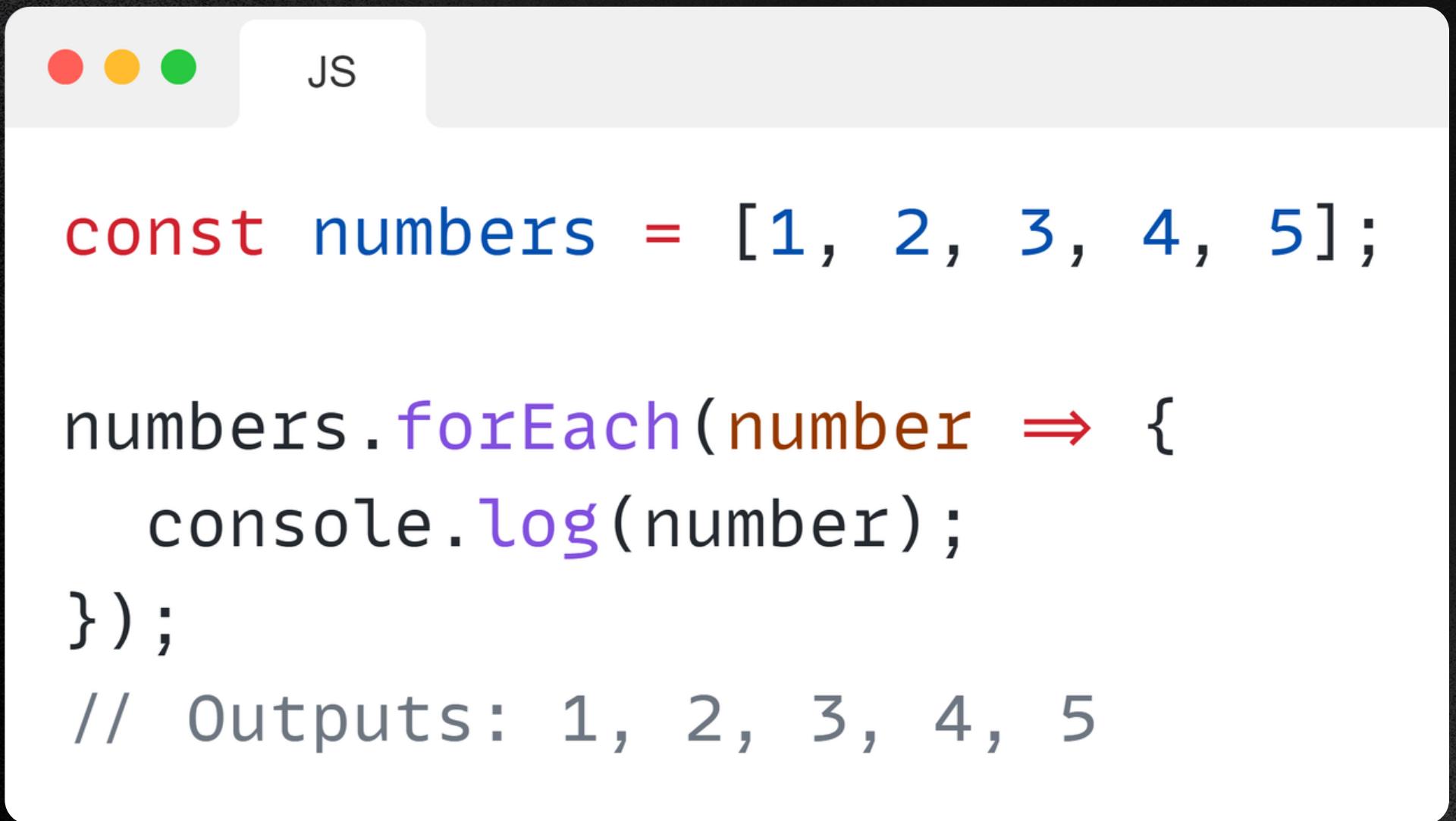
- Ideal for iterating over iterable objects like arrays or strings.

```
JS  
const techStack = ["JavaScript", "React", "Node.js"];  
  
for (const tech of techStack) {  
    console.log(tech);  
}  
// Outputs: JavaScript, React, Node.js
```



# forEach() Array Method

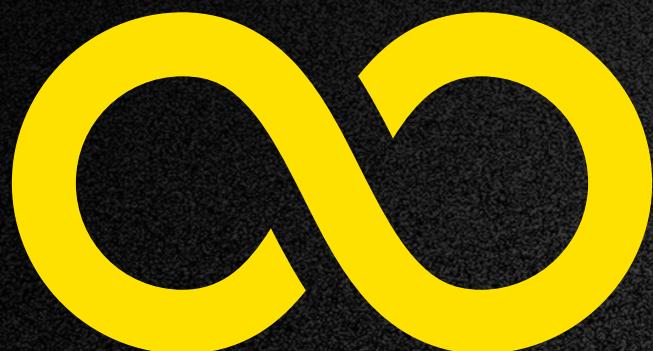
- A concise and expressive way to iterate through arrays.



The screenshot shows a browser's developer tools console window. The title bar has three colored dots (red, yellow, green) and the text "JS". The console displays the following JavaScript code:

```
const numbers = [1, 2, 3, 4, 5];  
  
numbers.forEach(number => {  
  console.log(number);  
});  
// Outputs: 1, 2, 3, 4, 5
```

The code defines an array "numbers" and uses its "forEach" method to log each number to the console. The output is shown as a comment at the bottom: "// Outputs: 1, 2, 3, 4, 5".



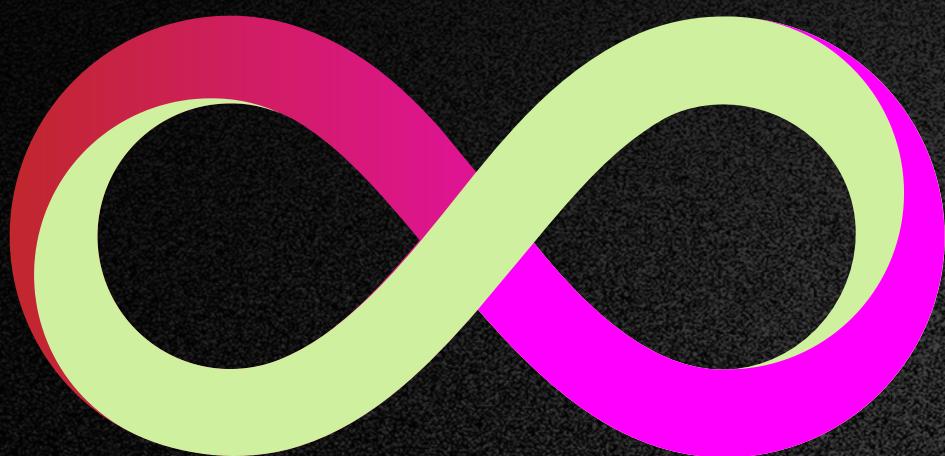
# for...in Loop

- Perfect for iterating over the properties of an object.

```
const user = { name: "TechExplorer",
               role: "Developer",
               language: "JavaScript" };

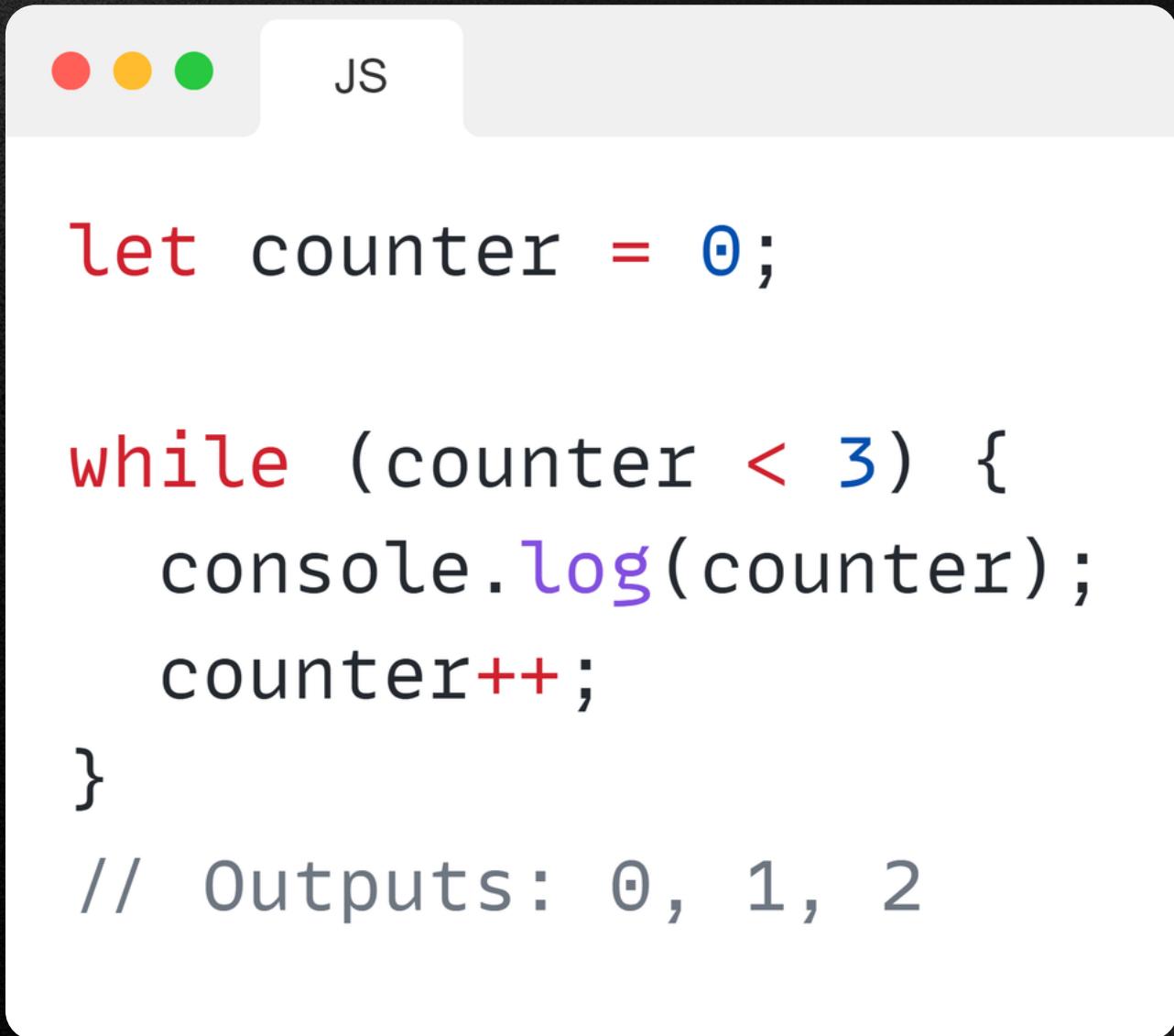
for (const key in user) {
  console.log(` ${key}: ${user[key]}`);
}

// Outputs: name: TechExplorer,
//           role: Developer,
//           language: JavaScript
```



# while Loop

- Use when the number of iterations is unknown or based on a condition.

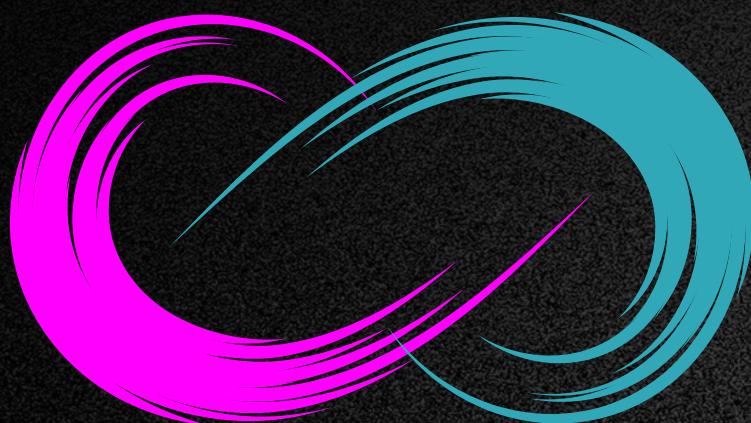


A screenshot of a code editor window titled "JS". The code inside the window is as follows:

```
let counter = 0;

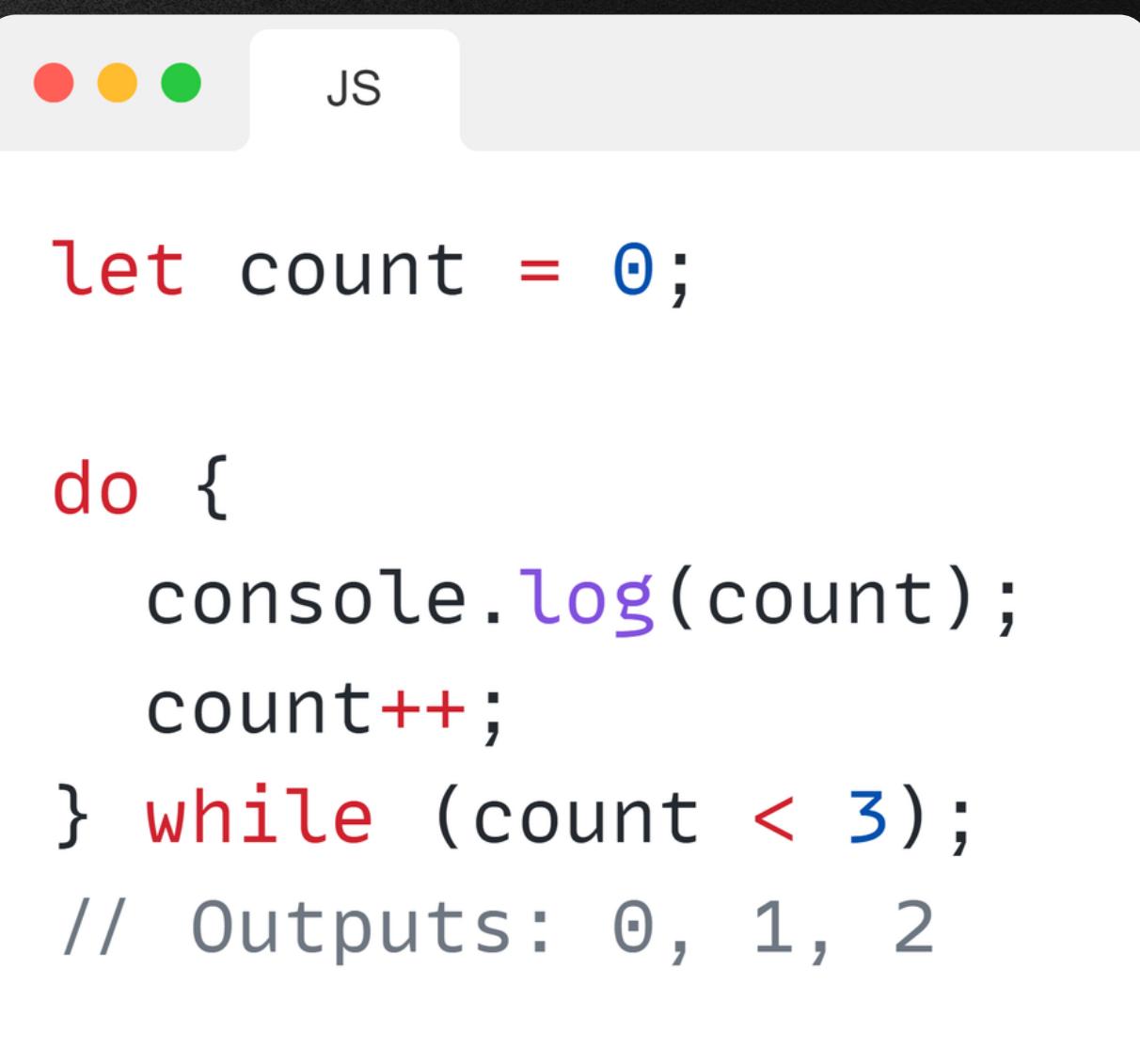
while (counter < 3) {
    console.log(counter);
    counter++;
}

// Outputs: 0, 1, 2
```



# do...while Loop

- Ensures at least one execution before checking the condition.



```
let count = 0;

do {
    console.log(count);
    count++;
} while (count < 3);
// Outputs: 0, 1, 2
```



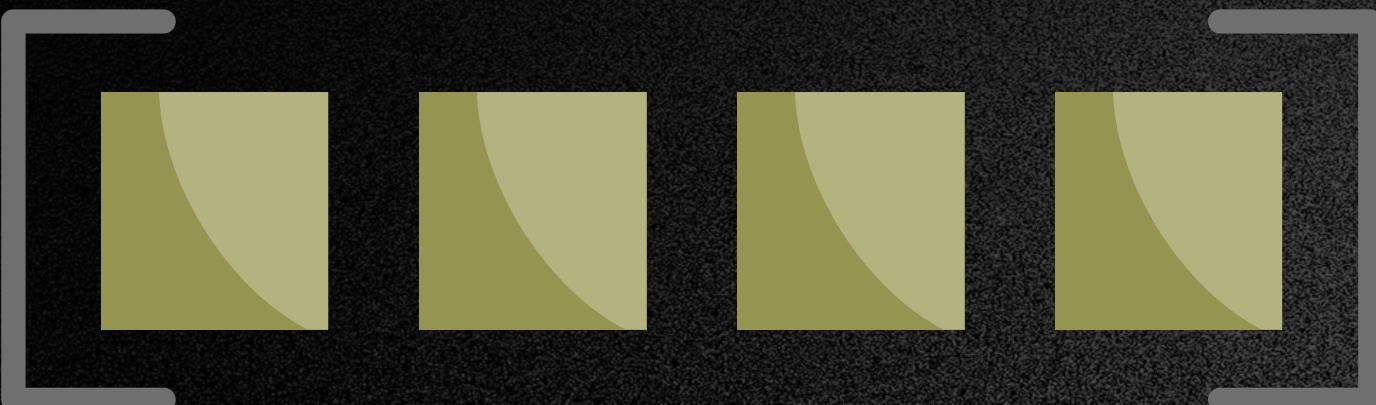
# Array Methods (map, filter, reduce)

- Elevate your array operations with these functional programming gems.

```
JS

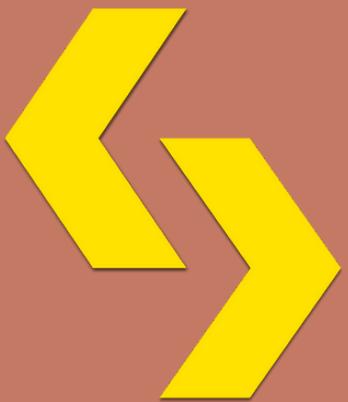
const numbers = [1, 2, 3, 4, 5];

const doubledNumbers = numbers.map(number => number * 2);
console.log(doubledNumbers);
// Outputs: [2, 4, 6, 8, 10]
```



# KEEP EXPLORING JAVASCRIPT WITH US!

Share this with a friend who needs it and make sure to practice these in scribbler.



**SCRIBBLER.LIVE**

Free and Open Interface to  
experiment JavaScript