“Nullish Coalescing” Operator

??

A Simple Guide

Introduction

What is nullish coalescing(??) operator?

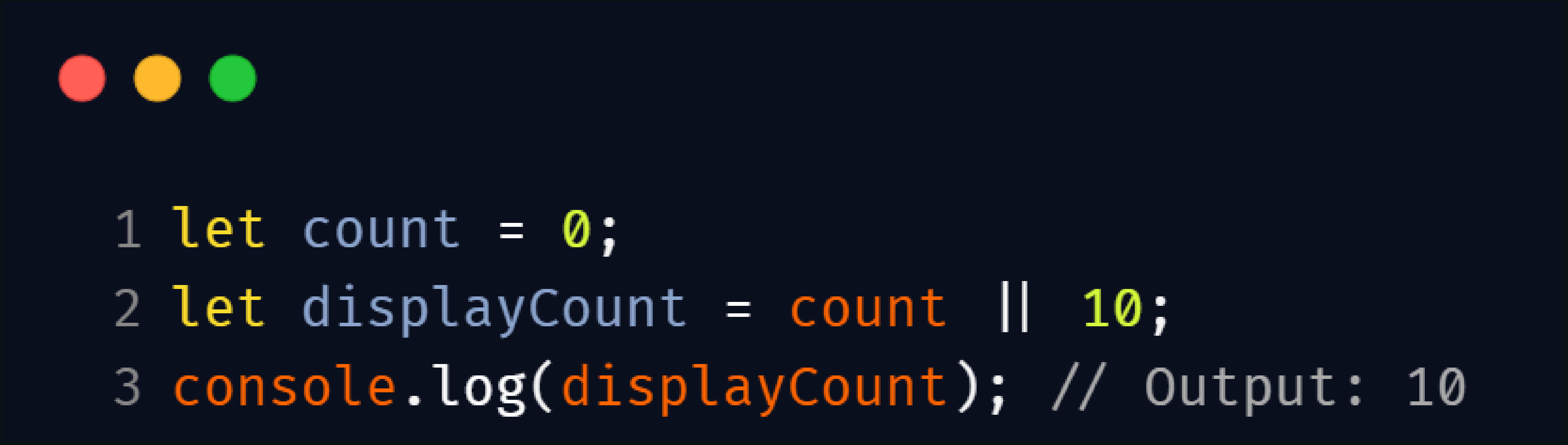
The ?? operator returns the right-hand operand when the left-hand operand is null or undefined.

This is particularly useful when you want to provide a default value but avoid false positives with 0, ' ', or false.

Although this can be achieved in a way using the “||” (or) operator, but there’s a limitation to it.

In JavaScript, the “||” operator returns the first truthy value it encounters. But this can lead to unexpected results when dealing with falsy values like 0, '', or false.

Checkout this example:



```
let count = 0;
let displayCount = count || 10;
console.log(displayCount); // Output: 10
```

In this example, count is 0 (a falsy value), so || ignores it and returns 10.

But what if you actually wanted to display 0 and not 10? 🤔

That is where the ?? operator comes useful.

The ?? operator solves this issue by returning the right-hand operand only if the left-hand operand is null or undefined.

This operator helps to handle cases where a variable might be null or undefined without unintentionally overriding other falsy values like 0, false, or an empty string "".

If the left-hand operand is any other value (even if it's a falsy value like 0, false, or ""), the operator returns the left-hand operand instead.

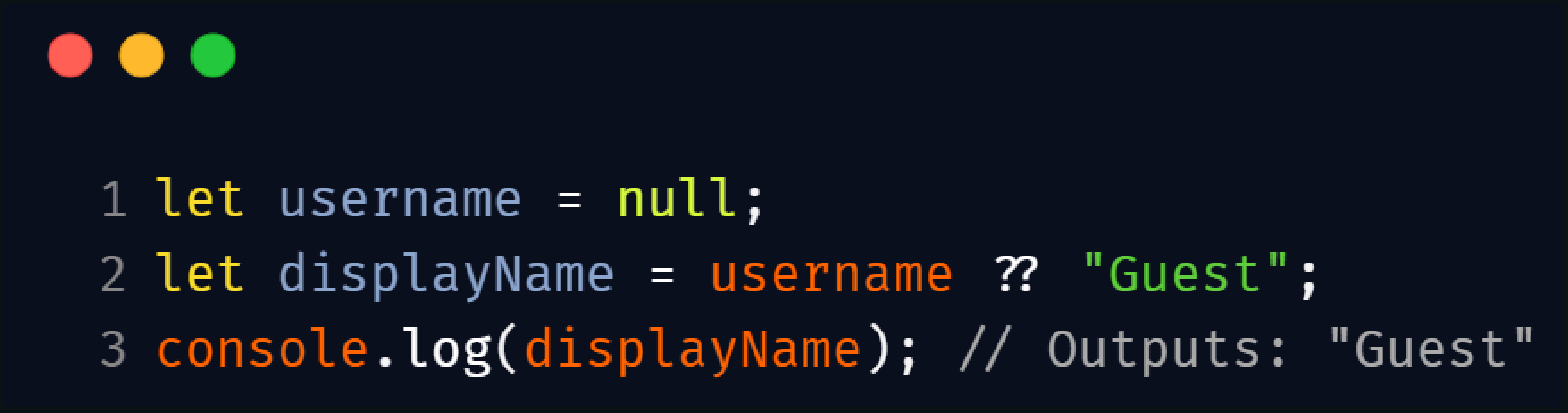
The syntax usually is:

```
let result = leftOperand ?? rightOperand
```

- leftOperand: The value to check.
- rightOperand: The value to return if leftOperand is null or undefined.

Usage of the ?? operator

1. Basic usage

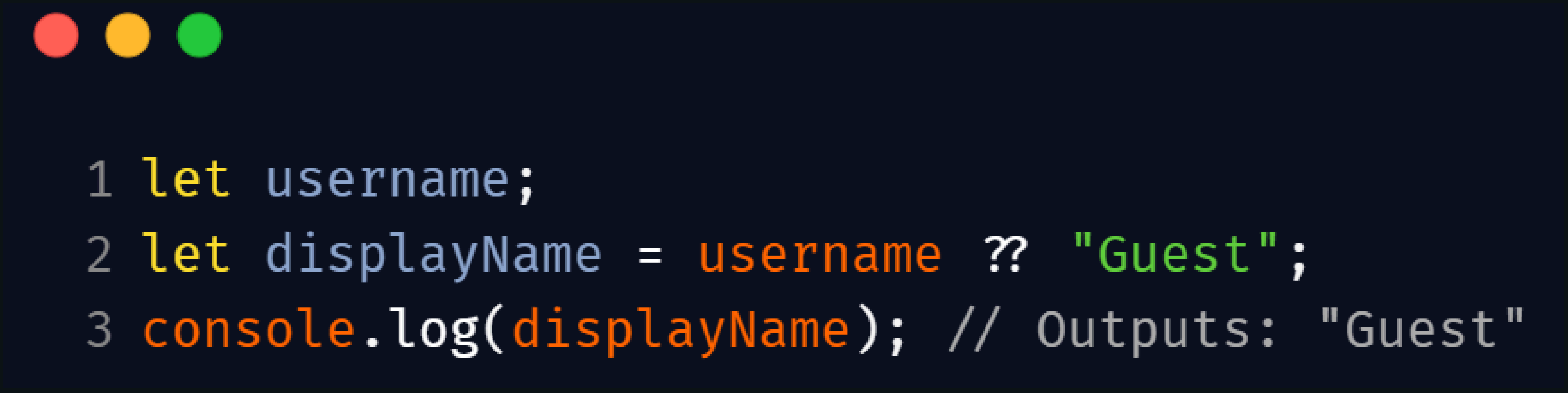


```
let username = null;
let displayName = username ?? "Guest";
console.log(displayName); // Outputs: "Guest"
```

Here, username is null, so the ?? operator returns "Guest".

kodemaven-portfolio.vercel.app

2. With Undefined

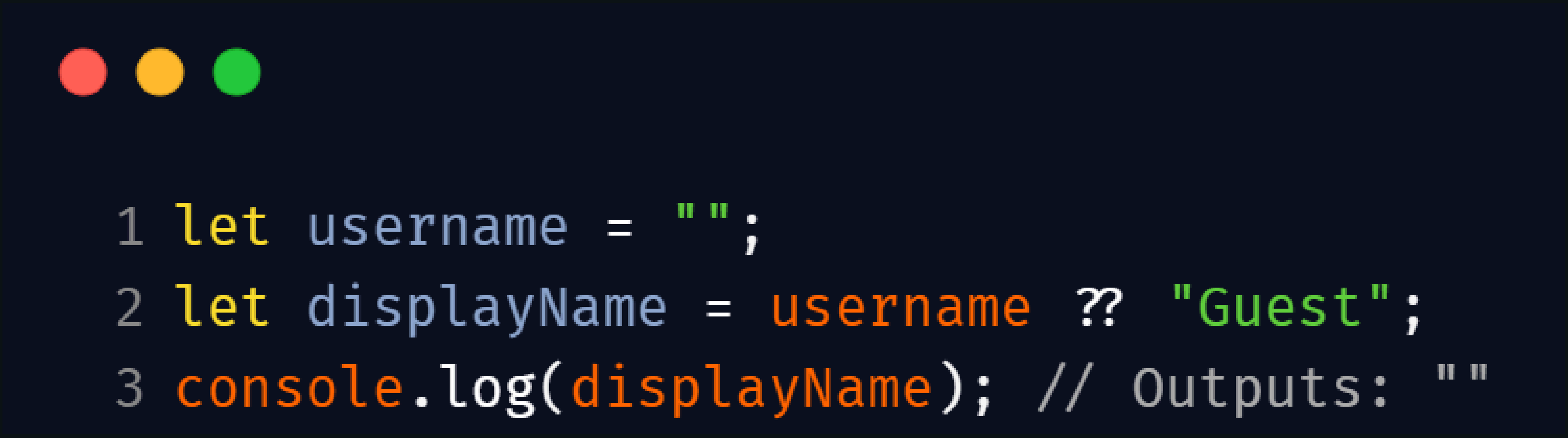


```
let username;
let displayName = username ?? "Guest";
console.log(displayName); // Outputs: "Guest"
```

Since username is undefined, the operator returns "Guest".

In this case, username is an empty string, which is falsy but not null or undefined. Therefore, the ?? operator returns the empty string.

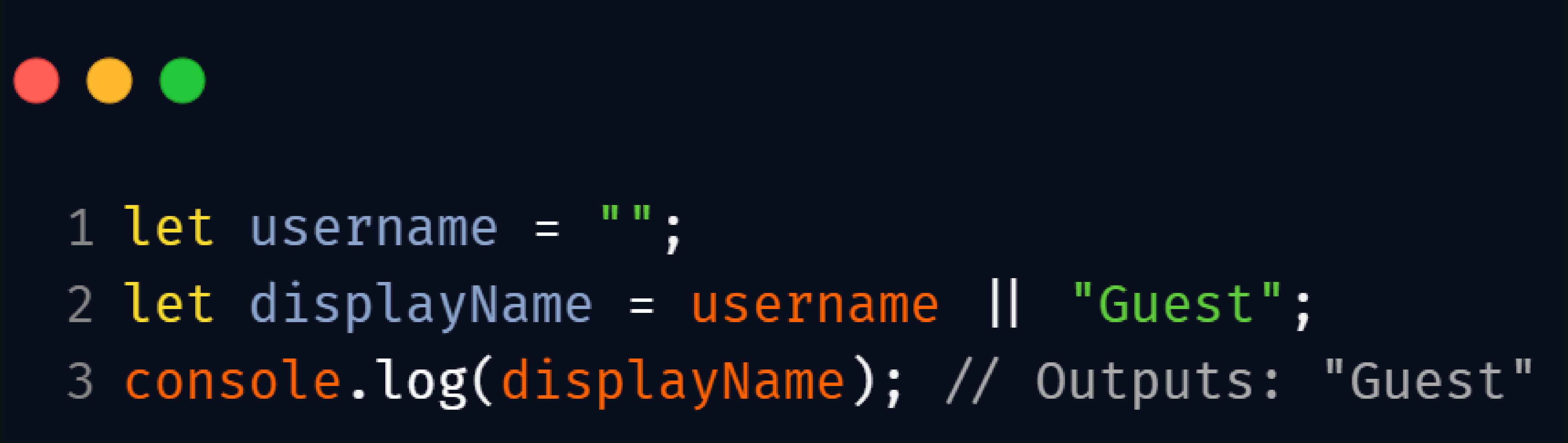
3. With Falsy Values (non-nullish):



```javascript
let username = "";
let displayName = username ?? "Guest";
console.log(displayName); // Outputs: ""
```

In this case, username is an empty string, which is falsy but not null or undefined. Therefore, the ?? operator returns the empty string.

4. Comparison with || (Logical OR) Operator:



```javascript
let username = "";
let displayName = username || "Guest";
console.log(displayName); // Outputs: "Guest"
```

The logical OR operator || returns "Guest" because it considers "" as falsy, unlike the ?? operator, which only checks for null or undefined.

Where and When to Use ??

The nullish coalescing operator is perfect for:

- Providing default values for optional parameters.
- Handling API responses where some fields might be null.
- Setting configuration defaults while allowing for 0, '', or false as valid inputs.

I hope you found this material useful and helpful.

Remember to:

Like

Save for future reference

&

Share with your network, be helpful to someone 👌

Hi There!

Thank you for reading through

Did you enjoy this knowledge?

💼 Follow my LinkedIn page for more work-life balancing and Coding tips.

🌐 LinkedIn: Oluwakemi Oluwadahunsi