04

ARRAYS

ARRAYS

ARRAYS ARE ORDERED COLLECTIONS OF VALUES THAT CAN HOLD VARIOUS DATA TYPES, MAKING THEM ESSENTIAL FOR ORGANIZING AND MANIPULATING DATA IN YOUR JAVASCRIPT PROGRAMS. HERE'S A BREAKDOWN OF KEY CONCEPTS, CODE EXAMPLES, AND BEST PRACTICES:

DECLARATION AND INITIALIZATION

```
const fruits = ["apple", "banana", "orange"];
```

DECLARING USING A CONSTRUCTOR

```
const numbers = new Array(1, 2, 3, 4, 5);
```

02

ACCESSING ELEMENTS:

```
const firstFruit = fruits[0]; // "apple"
const lastNumber = numbers[numbers.length - 1]; // 5
```

PUSH

```
fruits.push("kiwi"); // fruits = ["apple", "banana", "orange", "kiwi"]
```

02 POP

```
const removedFruit = fruits.pop();
// removedFruit = "kiwi", fruits = ["apple", "banana", "orange"]
```

FOREACH

```
fruits.forEach(function(color) {
    console.log(color);
});
```

02 POP

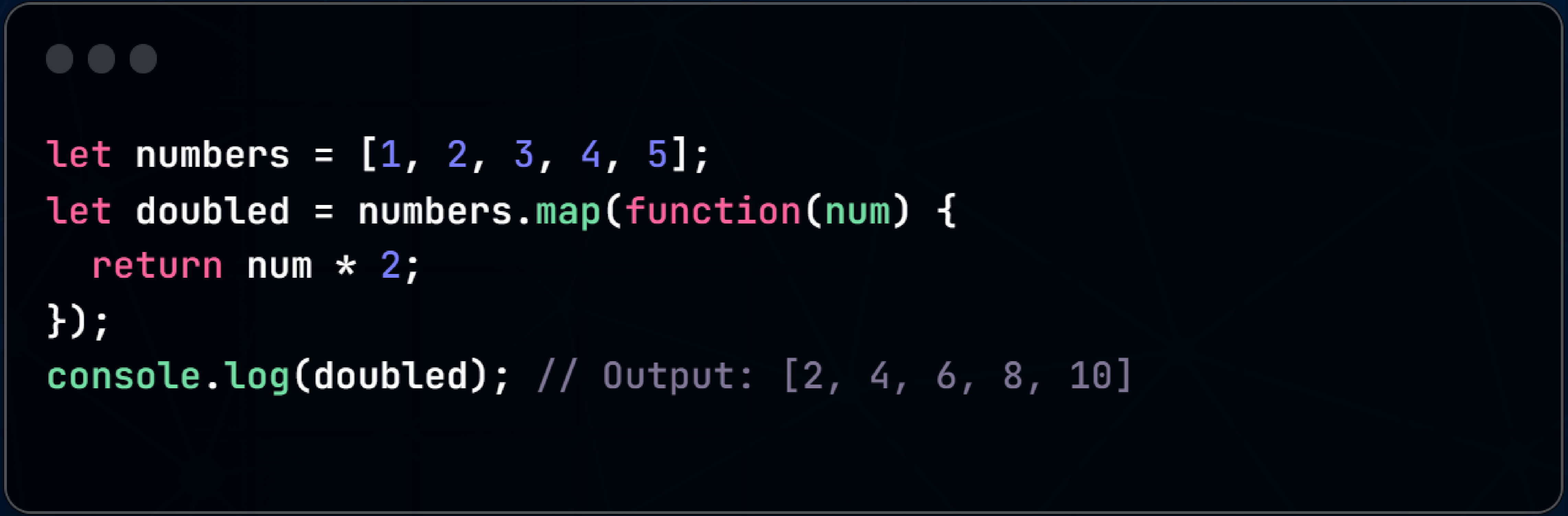
```
const removedFruit = fruits.pop();
// removedFruit = "kiwi", fruits = ["apple", "banana", "orange"]
```

FOREACH

```
fruits.forEach(function(color) {
    console.log(color);
});
```

02 MAP

The map() method creates a new array by applying a function to each element of the original array.

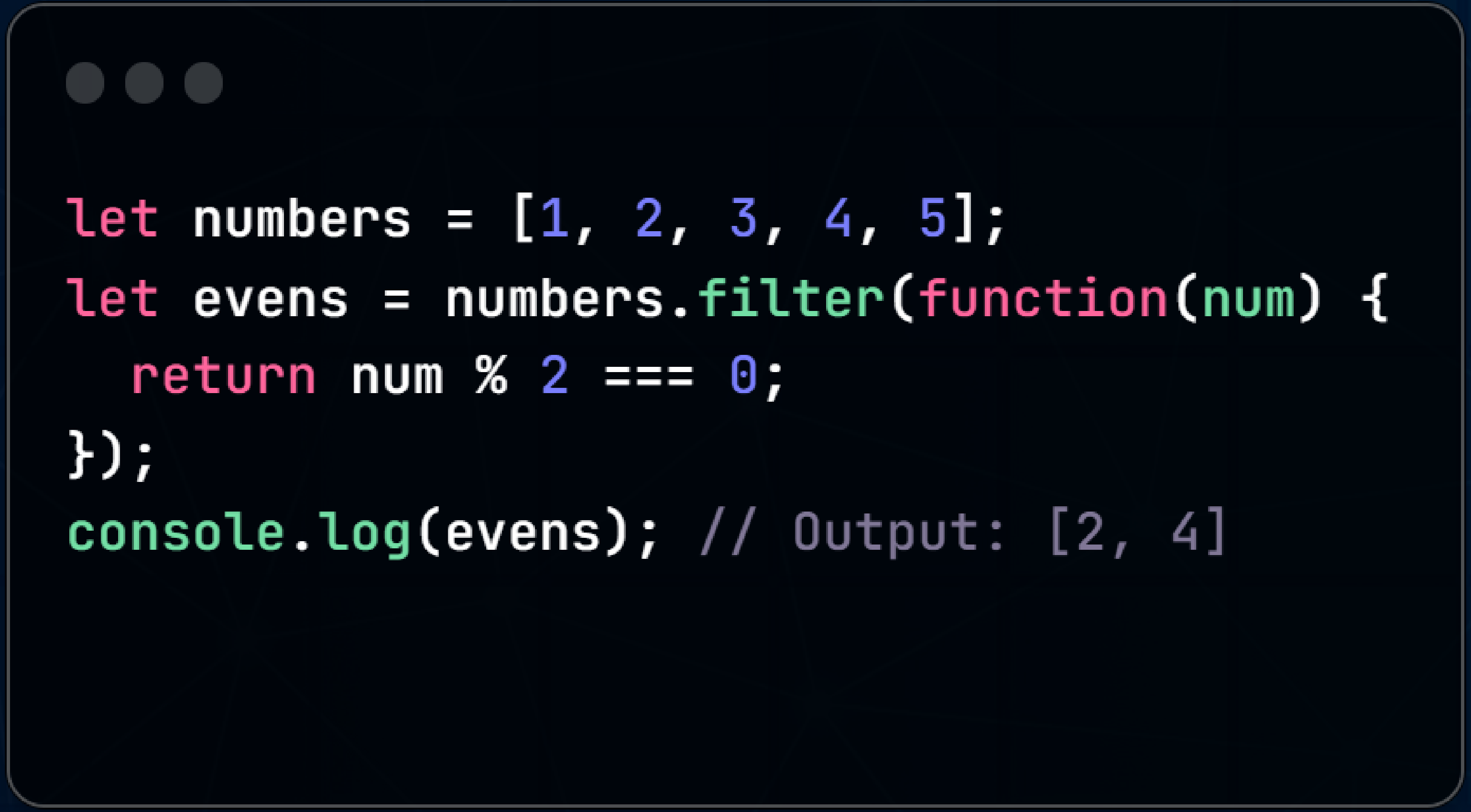


```javascript
let numbers = [1, 2, 3, 4, 5];
let doubled = numbers.map(function(num) {
  return num * 2;
});
console.log(doubled); // Output: [2, 4, 6, 8, 10]
```

02

FILTER

The filter() method creates a new array with elements that pass the test implemented by the provided function.

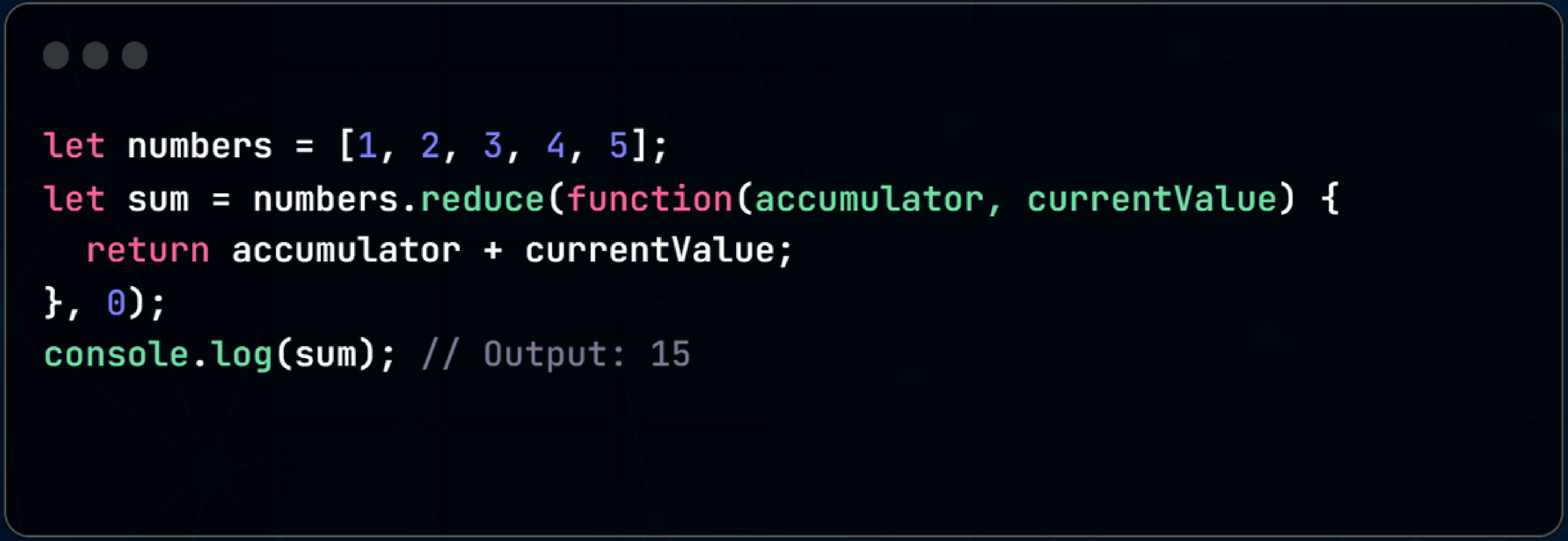


```
let numbers = [1, 2, 3, 4, 5];
let evens = numbers.filter(function(num) {
  return num % 2 === 0;
});
console.log(evens); // Output: [2, 4]
```

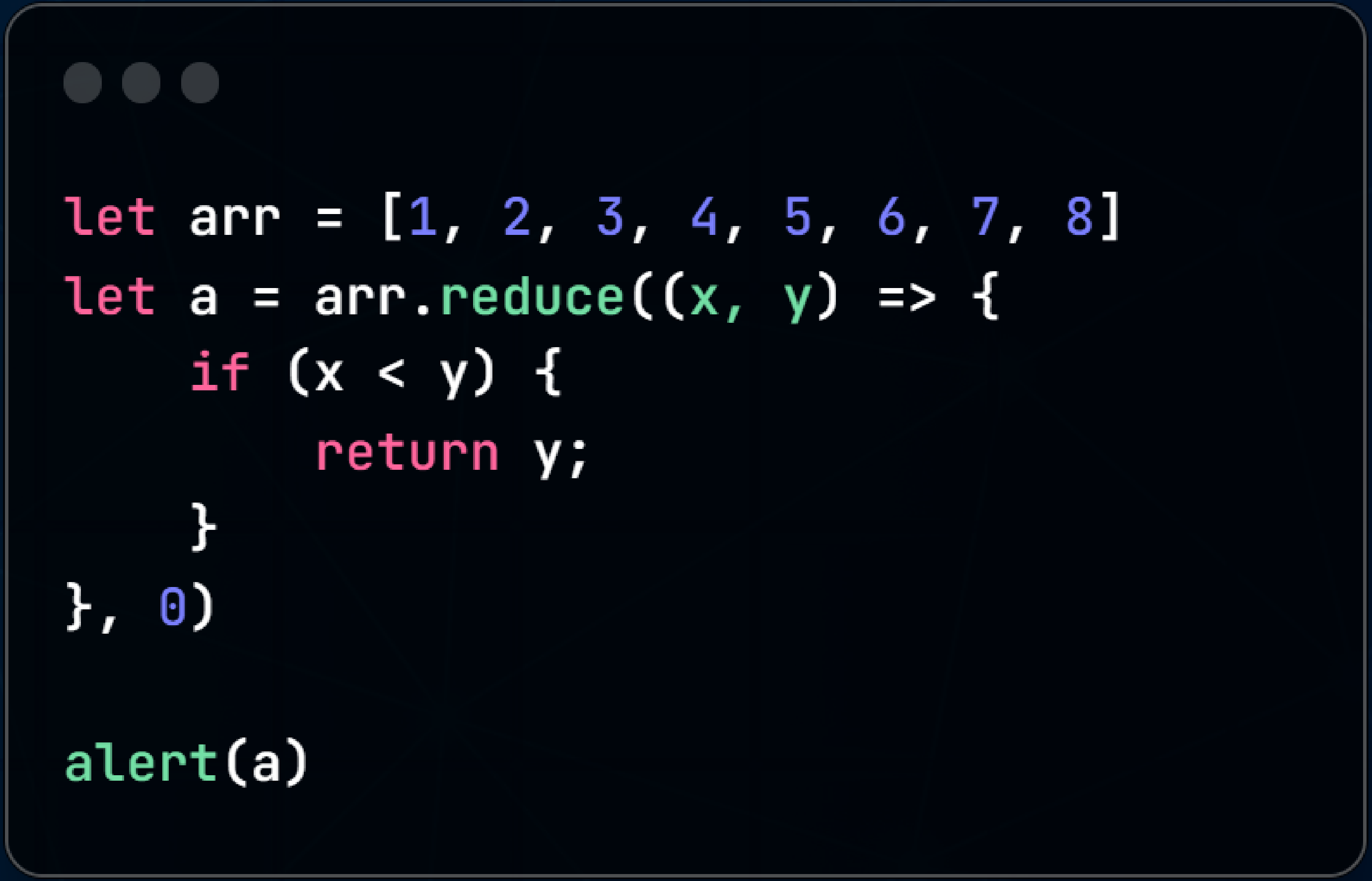
02

REDUCE

The reduce() method executes a reducer function on each element of the array, resulting in a single output value.



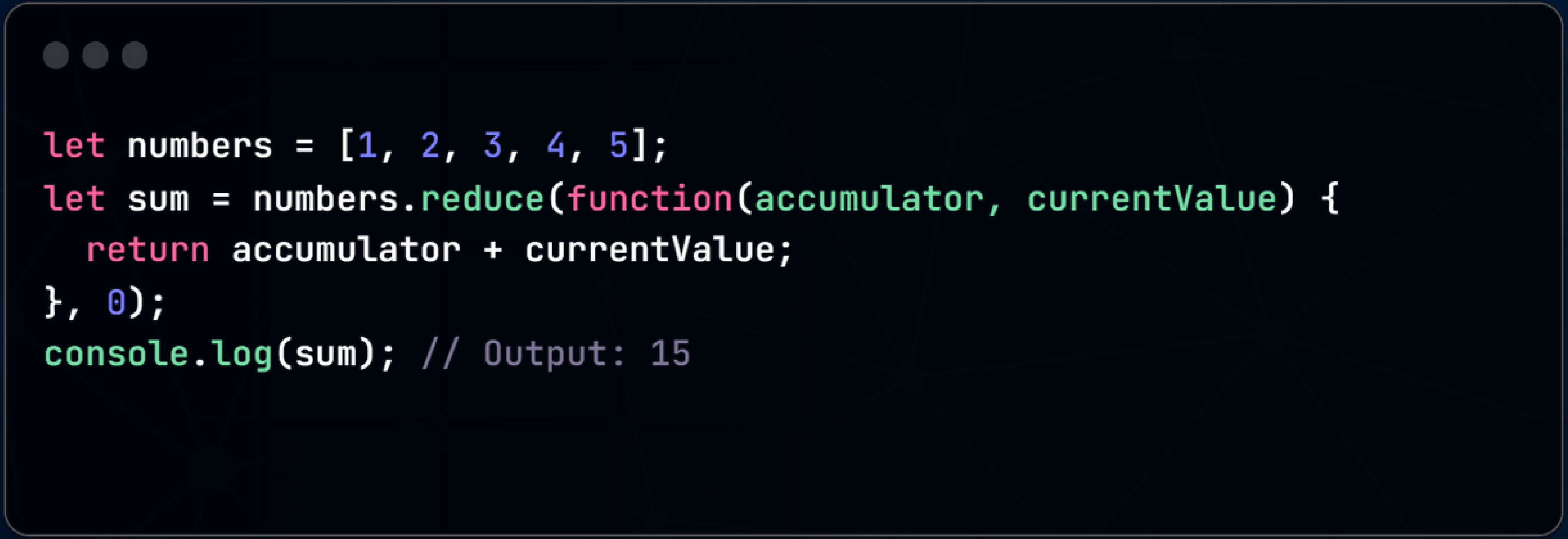
```
let numbers = [1, 2, 3, 4, 5];
let sum = numbers.reduce(function(accumulator, currentValue) {
  return accumulator + currentValue;
}, 0);
console.log(sum); // Output: 15
```



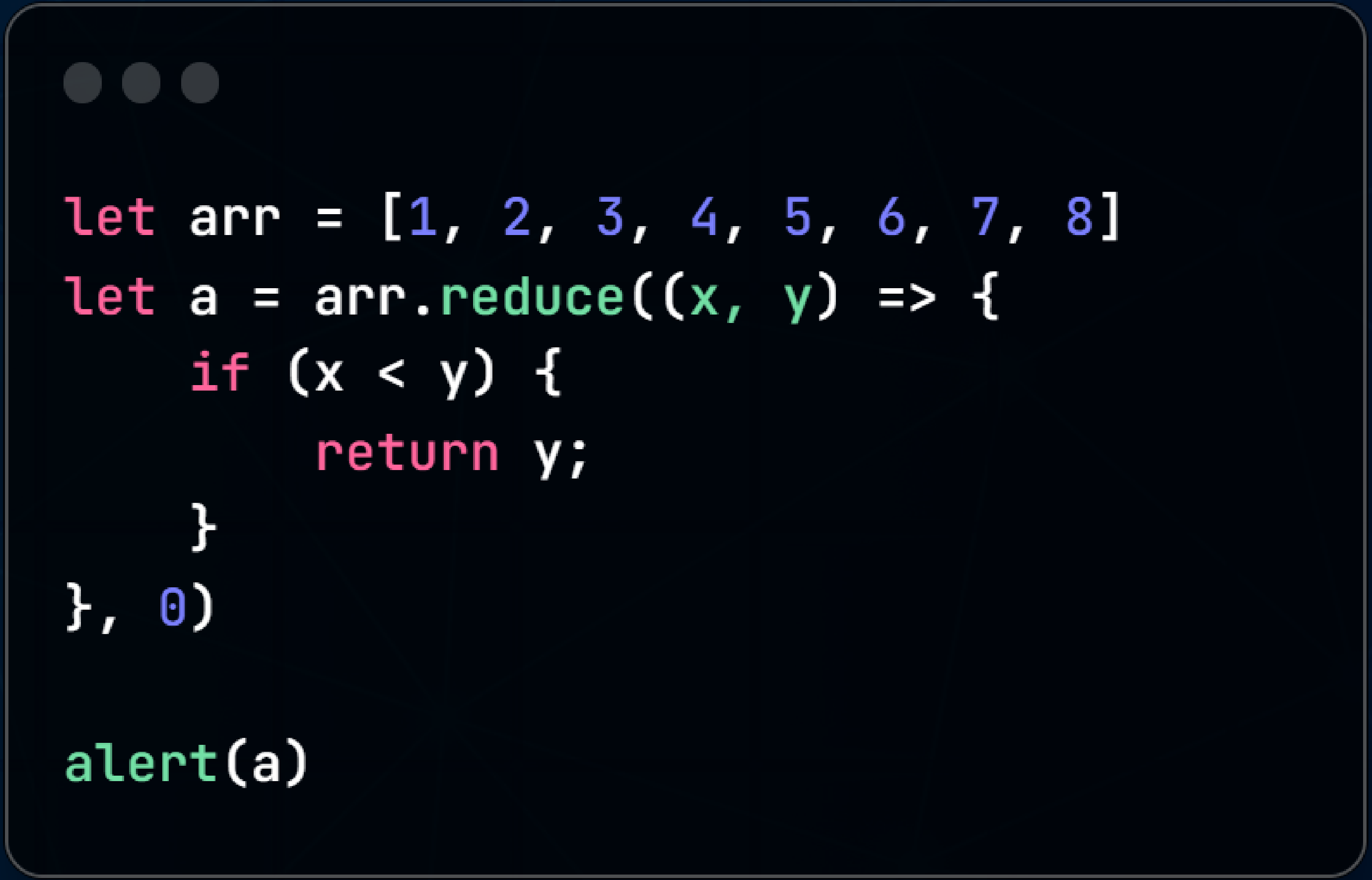
```
let arr = [1, 2, 3, 4, 5, 6, 7, 8]
let a = arr.reduce((x, y) => {
    if (x < y) {
        return y;
    }
}, 0)

alert(a)
```

02 REDUCE

The reduce() method executes a reducer function on each element of the array, resulting in a single output value.



```
let numbers = [1, 2, 3, 4, 5];
let sum = numbers.reduce(function(accumulator, currentValue) {
  return accumulator + currentValue;
}, 0);
console.log(sum); // Output: 15
```



```
let arr = [1, 2, 3, 4, 5, 6, 7, 8]
let a = arr.reduce((x, y) => {
    if (x < y) {
        return y;
    }
}, 0)

alert(a)
```