JS

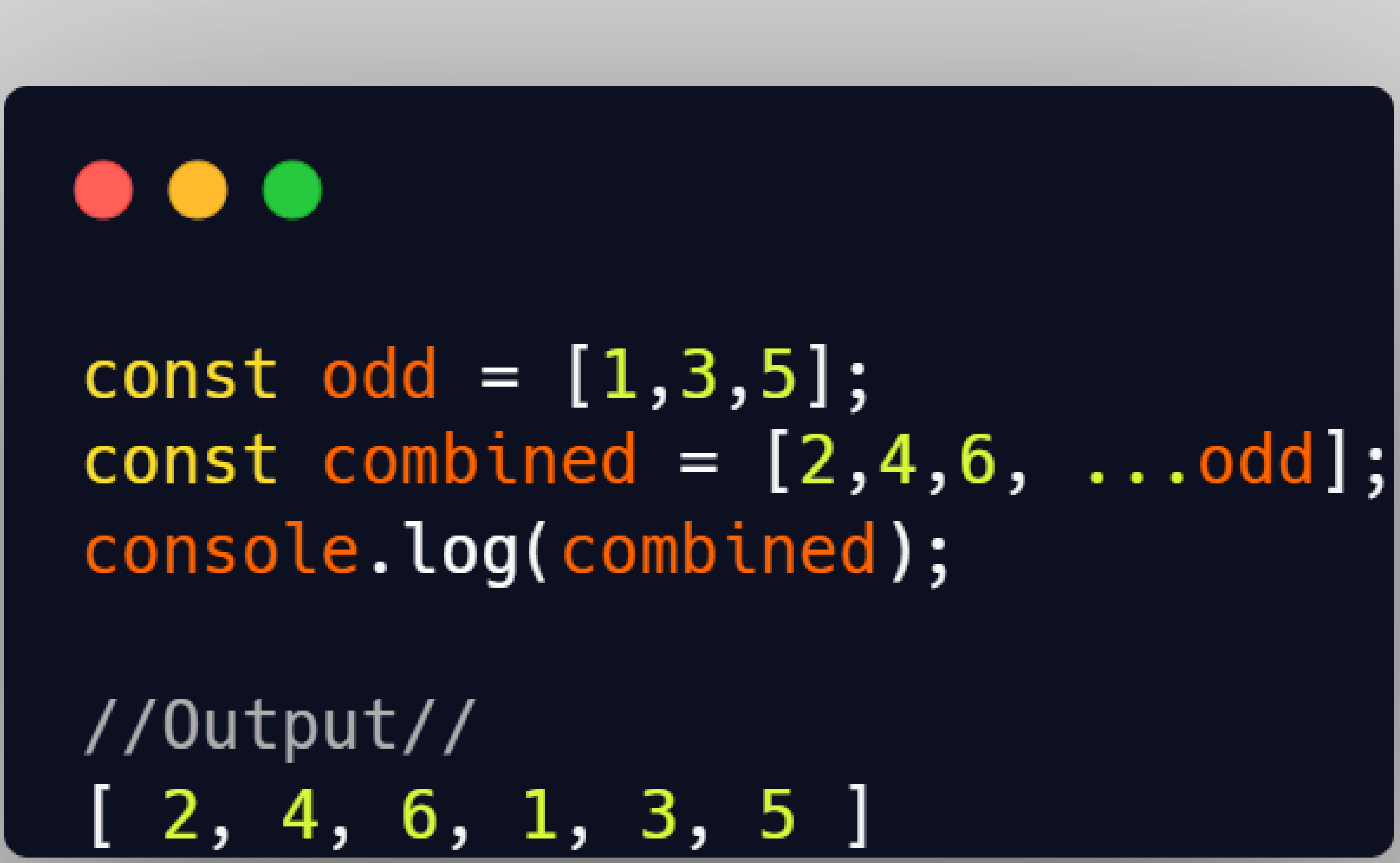
JAVASCRIPT SPREAD OPERATOR

(...)

codewithgauri

INTRODUCTION:

ES6 provides a new operator called spread operator that consists of three dots (...). The spread operator allows you to spread out elements of an iterable object such as an array, map, or set. For example:



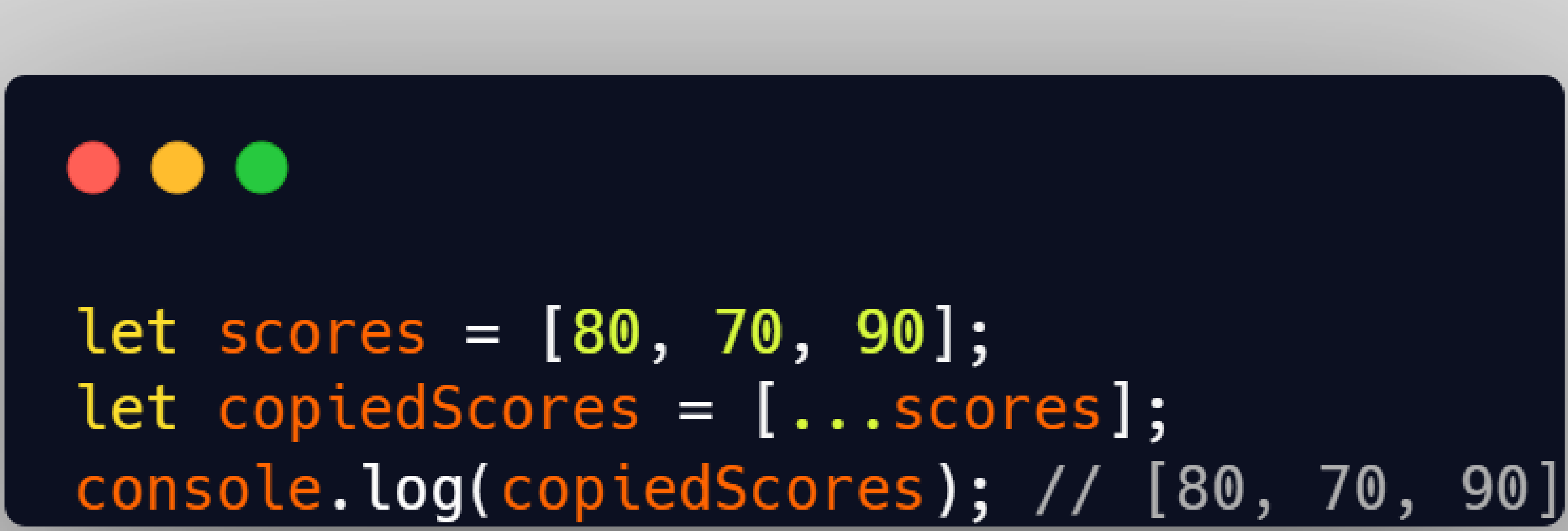
```
const odd = [1,3,5];
const combined = [2,4,6, ...odd];
console.log(combined);

//Output//
[ 2, 4, 6, 1, 3, 5 ]
```

In this example, the three dots (...) located in front of the odd array is the spread operator. The spread operator (...) unpacks the elements of the odd array.

Copying an array:

Using the ... spread operator is a convenient way to copy an array or combine arrays, and it can even add new items:

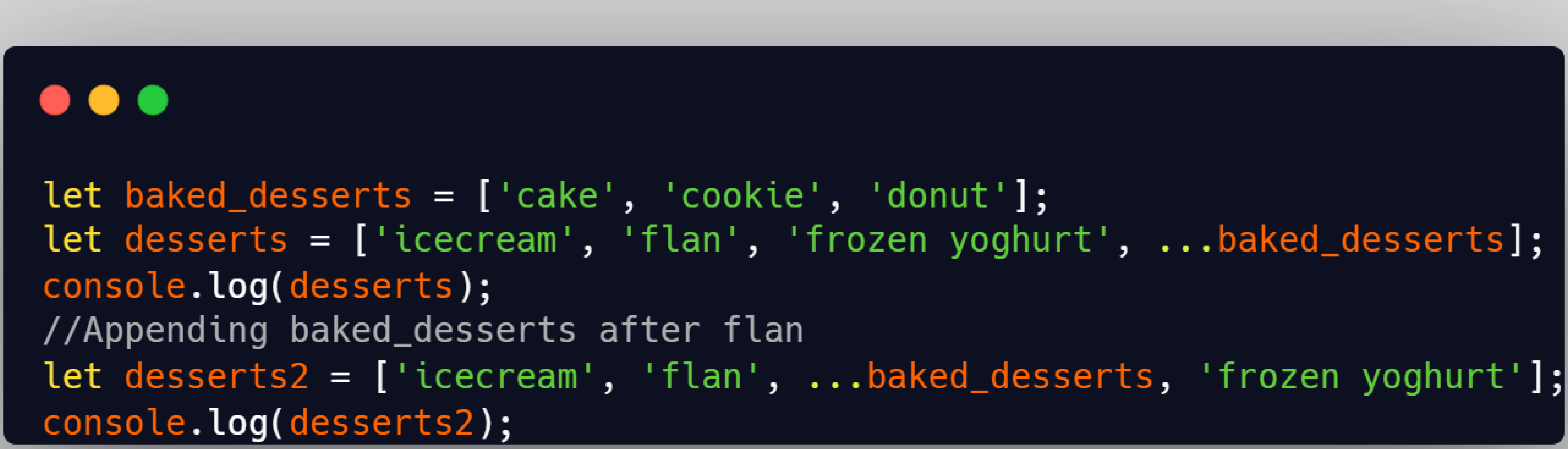


```
let scores = [80, 70, 90];
let copiedScores = [...scores];
console.log(copiedScores); // [80, 70, 90]
```

Note that the spread operator only copies the array itself to the new one, not the elements. This means that the copy is shallow, not deep.

Inserting the elements of one array into another:

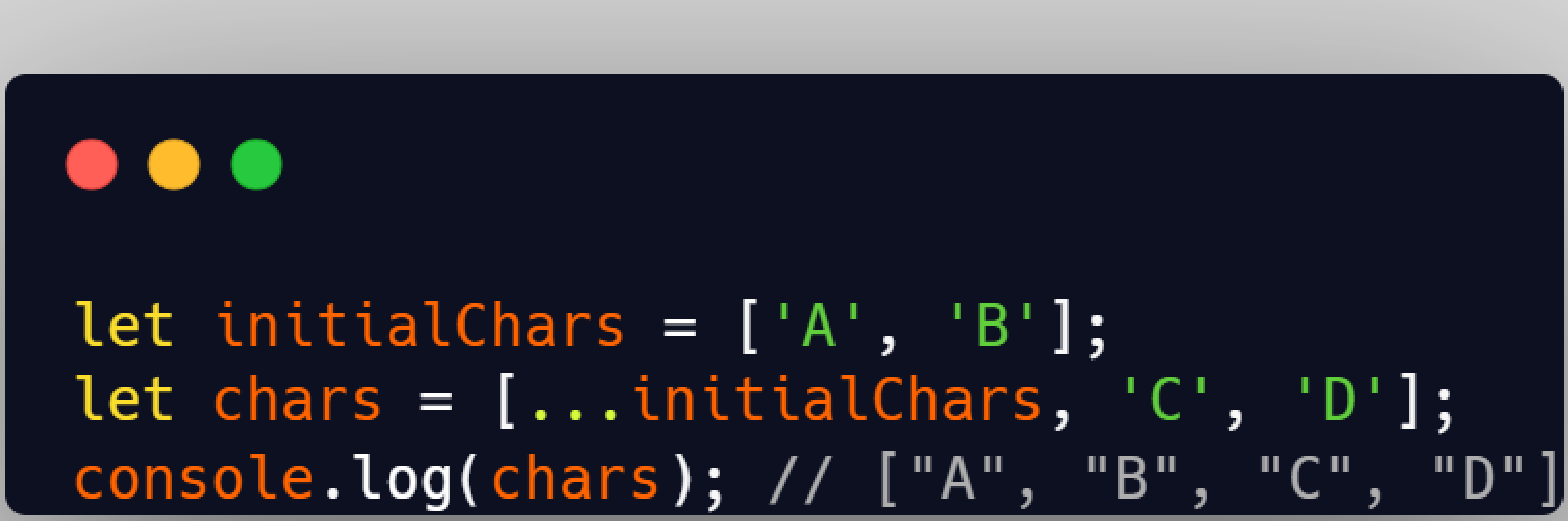
It can be seen that the spread operator can be used to append one array after any element of the second array. In other words, there is no limitation that baked_desserts can only be appended at the beginning or the end of the desserts2 array.



```
let baked_desserts = ['cake', 'cookie', 'donut'];
let desserts = ['icecream', 'flan', 'frozen yoghurt', ...baked_desserts];
console.log(desserts);
//Appending baked_desserts after flan
let desserts2 = ['icecream', 'flan', ...baked_desserts, 'frozen yoghurt'];
console.log(desserts2);
```

Constructing array literal:

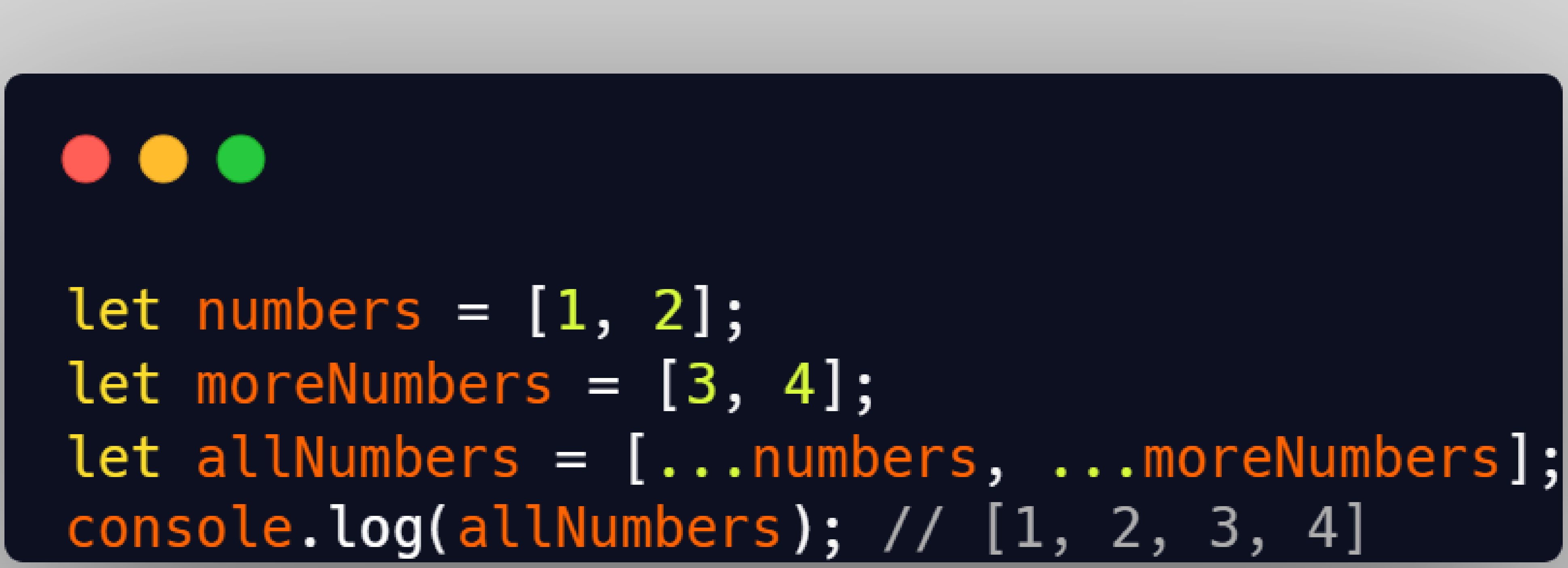
The spread operator allows you to insert another array into the initialized array when you construct an array using the literal form. See the following example:



```
let initialChars = ['A', 'B'];
let chars = [...initialChars, 'C', 'D'];
console.log(chars); // ["A", "B", "C", "D"]
```

Concatenating arrays:

Also, you can use the spread operator to concatenate two or more arrays:



```
let numbers = [1, 2];
let moreNumbers = [3, 4];
let allNumbers = [...numbers, ...moreNumbers];
console.log(allNumbers); // [1, 2, 3, 4]
```

And for amazing stuff you can follow me

Gaurav Pandey

LinkedIn :Gaurav Pandey

Twitter : @gauravcode

References:
The Practical Usages of JavaScript Spread Operator (javascripttutorial.net)
How to Use the Spread Operator (...) in JavaScript | by Dr. Derek Austin | Coding at Dawn | Medium