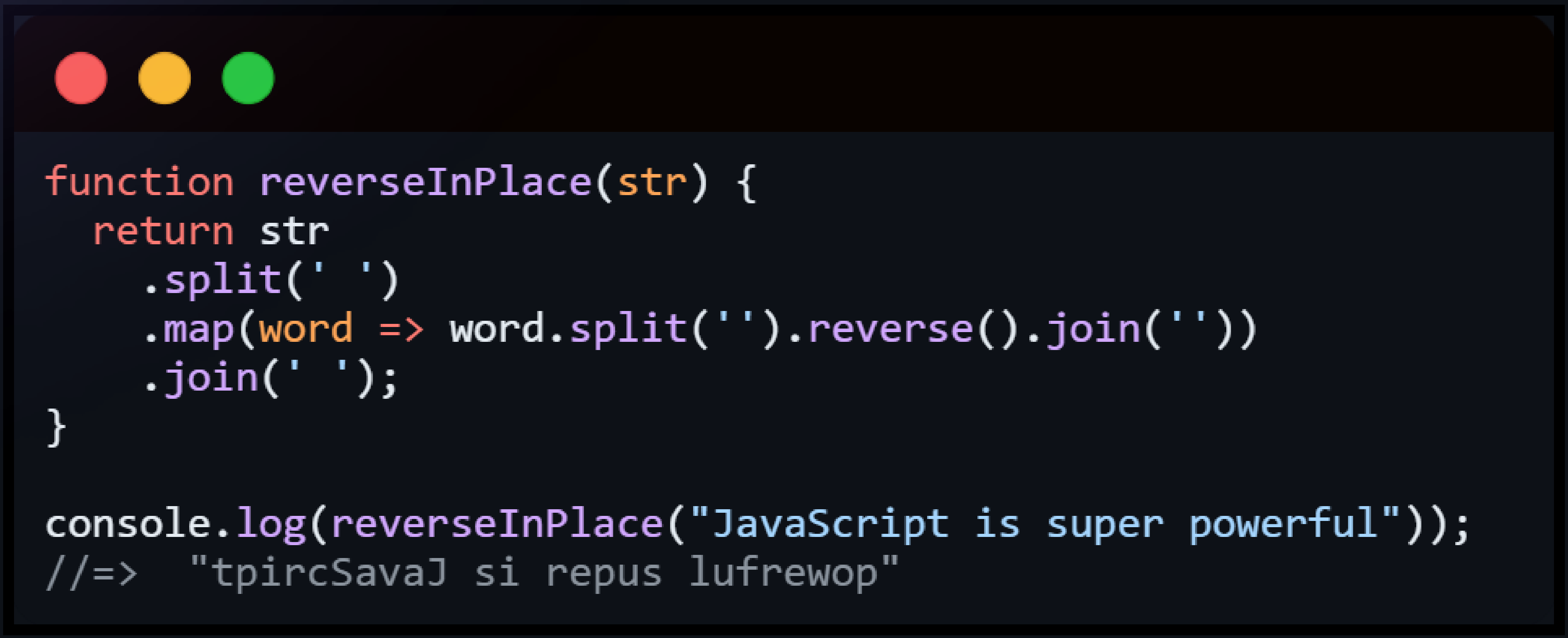
Reverse in Place

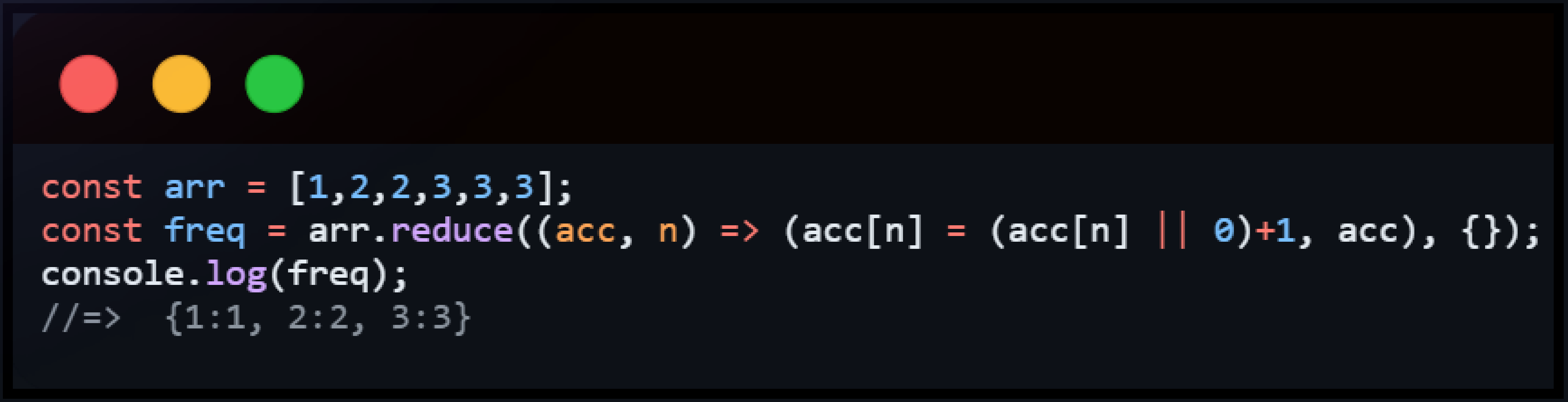


```
function reverseInPlace(str) {
  return str
    .split(' ')
    .map(word => word.split('').reverse().join(''))
    .join(' ');
}

console.log(reverseInPlace("JavaScript is super powerful"));
//=>  "tpircSavaJ si repus lufrewop"
```

Explanation :

1. Split the string into words = ["JavaScript", "is", "super", "powerful"]
2. Reverse each word using .split('').reverse().join('')
3. Join the words back into a sentence

Count Frequency of Each Element

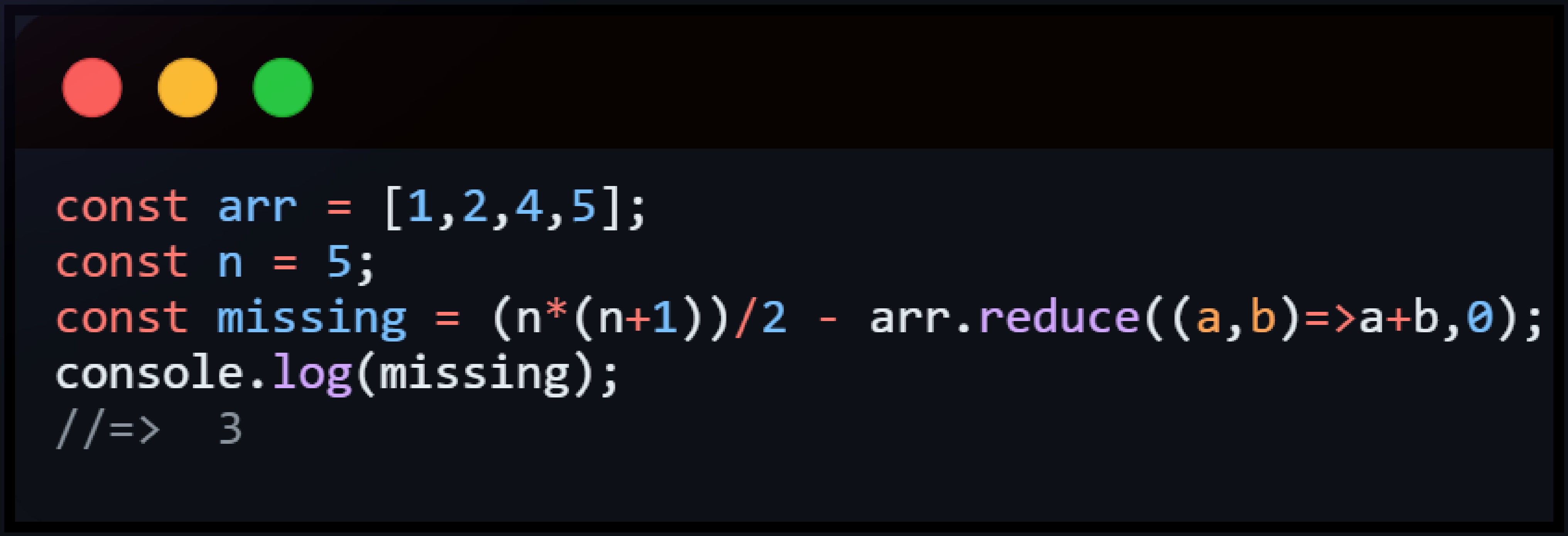


```
const arr = [1,2,2,3,3,3];
const freq = arr.reduce((acc, n) => (acc[n] = (acc[n] || 0)+1, acc), {});
console.log(freq);
//=>  {1:1, 2:2, 3:3}
```

Explanation :

1. Initialize a map/object to store element counts.
2. Iterate through the array and increment count for each element.
3. Check if element exists in the map; if not, set count to 1.

Find Missing Number in 1…N

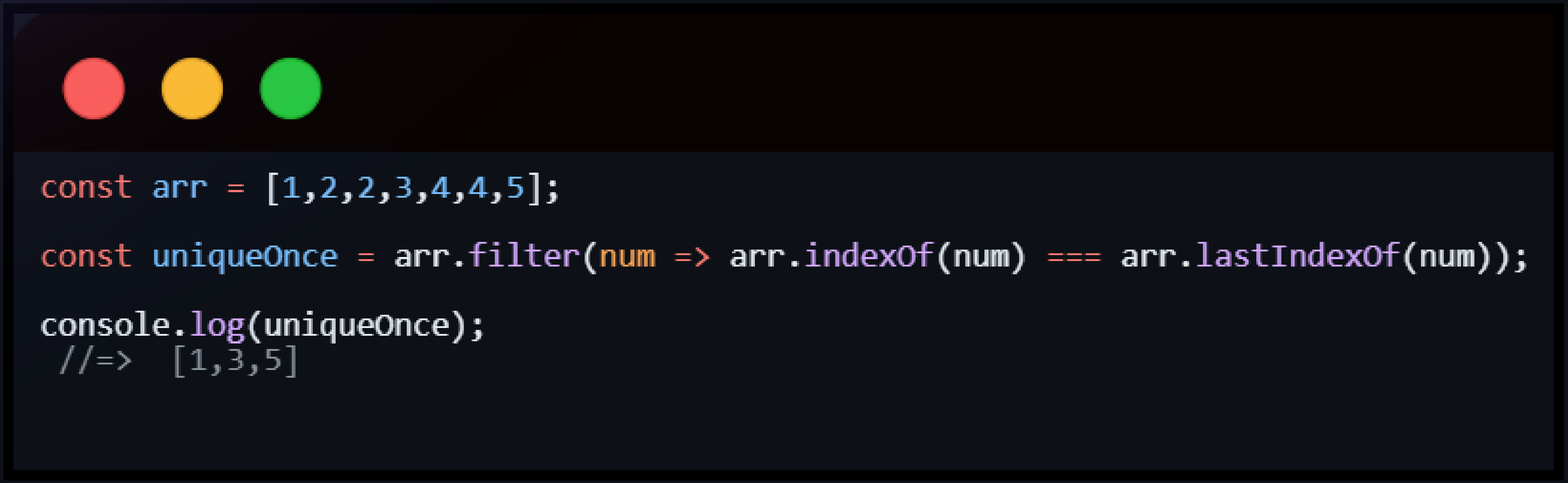


```js
const arr = [1,2,4,5];
const n = 5;
const missing = (n*(n+1))/2 - arr.reduce((a,b)=>a+b,0);
console.log(missing);
//=>  3
```

Explanation :

1. Sum of 1…N is calculated using the formula: n*(n+1)/2.
2. Sum the elements of the given array.
3. Subtract array sum from total sum to find the missing number.

Only numbers that occur once

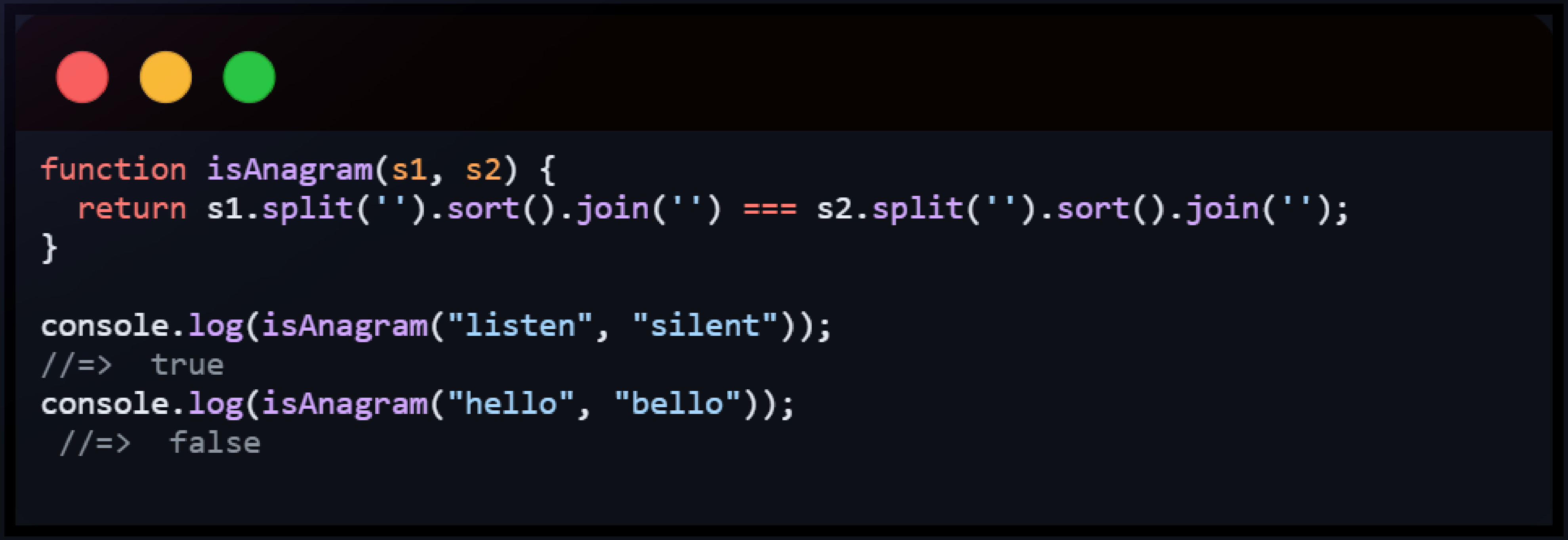


```javascript
const arr = [1,2,2,3,4,4,5];

const uniqueOnce = arr.filter(num => arr.indexOf(num) === arr.lastIndexOf(num));

console.log(uniqueOnce);
 //=>  [1,3,5]
```

Explanation :

1. arr.indexOf(num) gives the first occurrence index.
2. arr.lastIndexOf(num) gives the last occurrence index.
3. If they are equal, the number occurs only once.
4. Filter keeps only those numbers.

Check if Two Strings are Anagrams

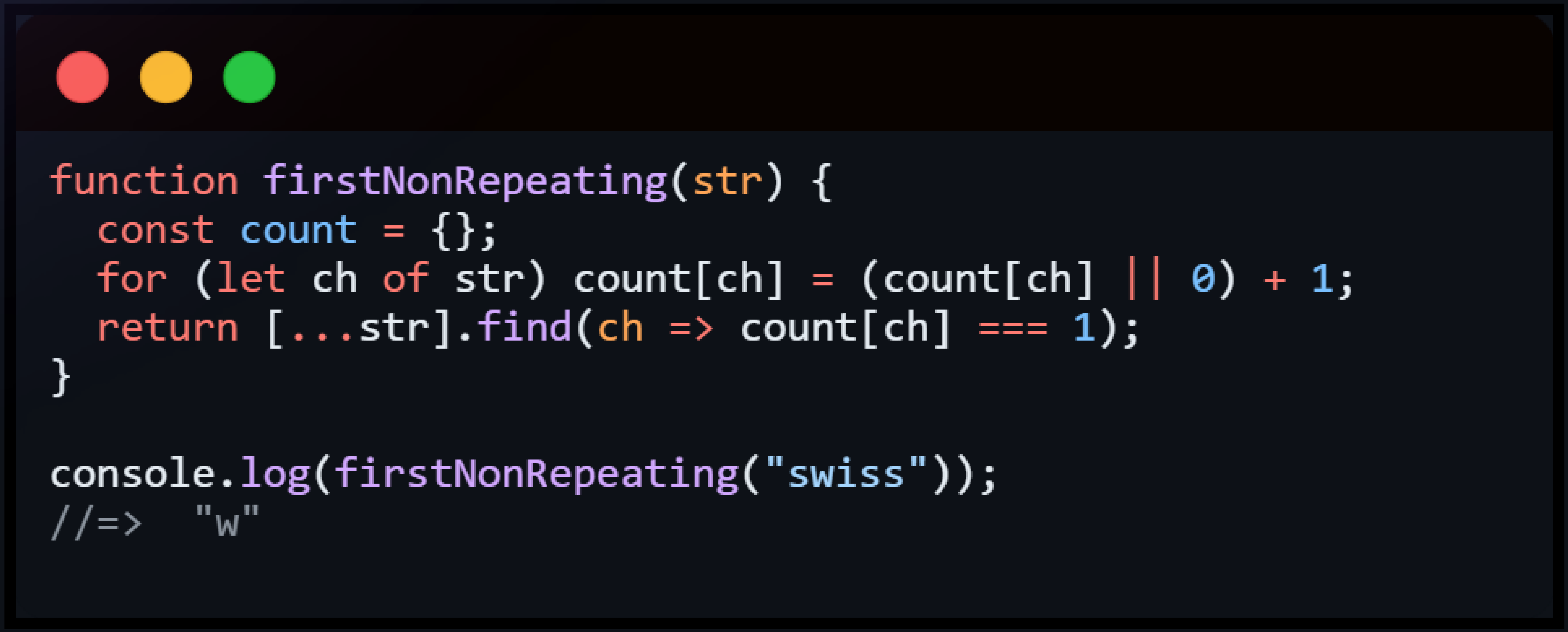


```
function isAnagram(s1, s2) {
  return s1.split('').sort().join('') === s2.split('').sort().join('');
}

console.log(isAnagram("listen", "silent"));
//=>  true
console.log(isAnagram("hello", "bello"));
 //=>  false
```

Explanation :

1. Convert both strings to arrays of characters using split('').
2. Sort the arrays alphabetically using .sort().
3. Join the arrays back into strings using .join('').
4. Compare the sorted strings.

First Non-Repeating Character

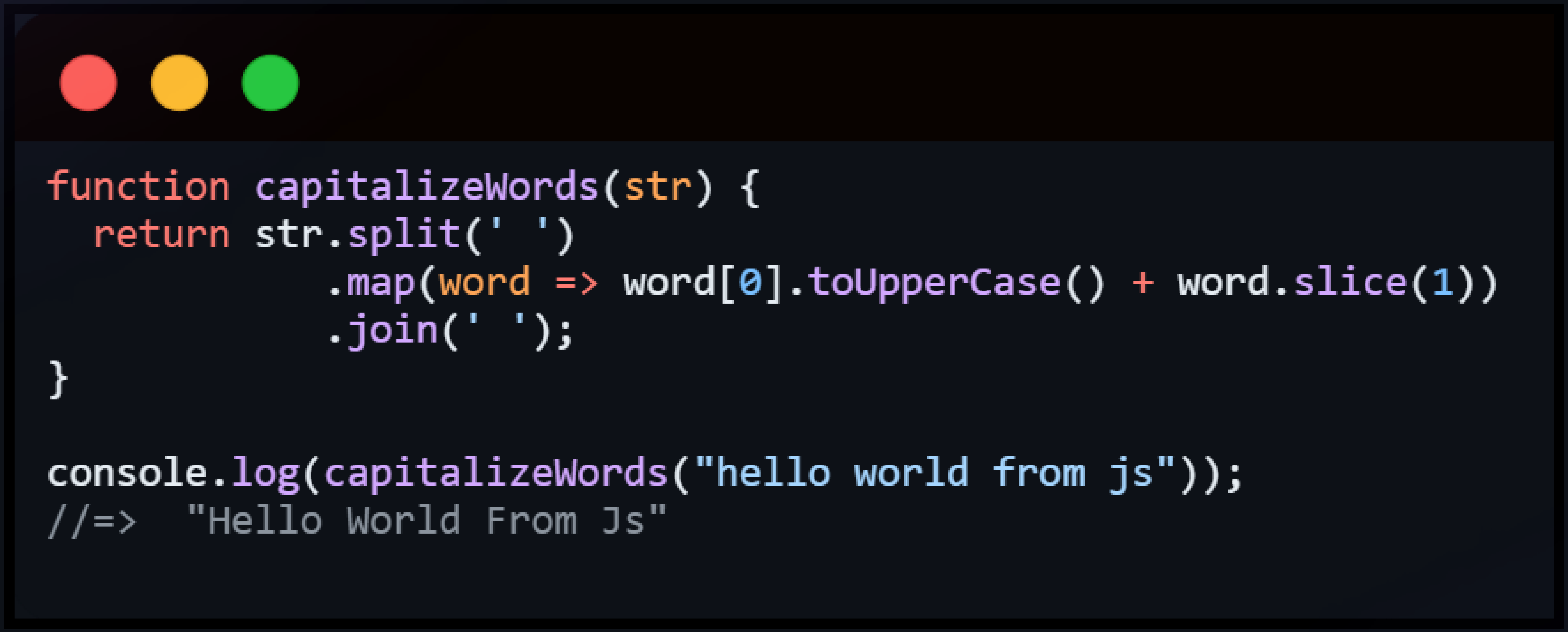


```
function firstNonRepeating(str) {
  const count = {};
  for (let ch of str) count[ch] = (count[ch] || 0) + 1;
  return [...str].find(ch => count[ch] === 1);
}

console.log(firstNonRepeating("swiss"));
//=>  "w"
```

Explanation :

1. Create a map/object to count occurrences of each character.
2. Iterate through the string and increment the count for each character.
3. Traverse the string again to find the first character with count 1.

Capitalize First Letter of Each Word

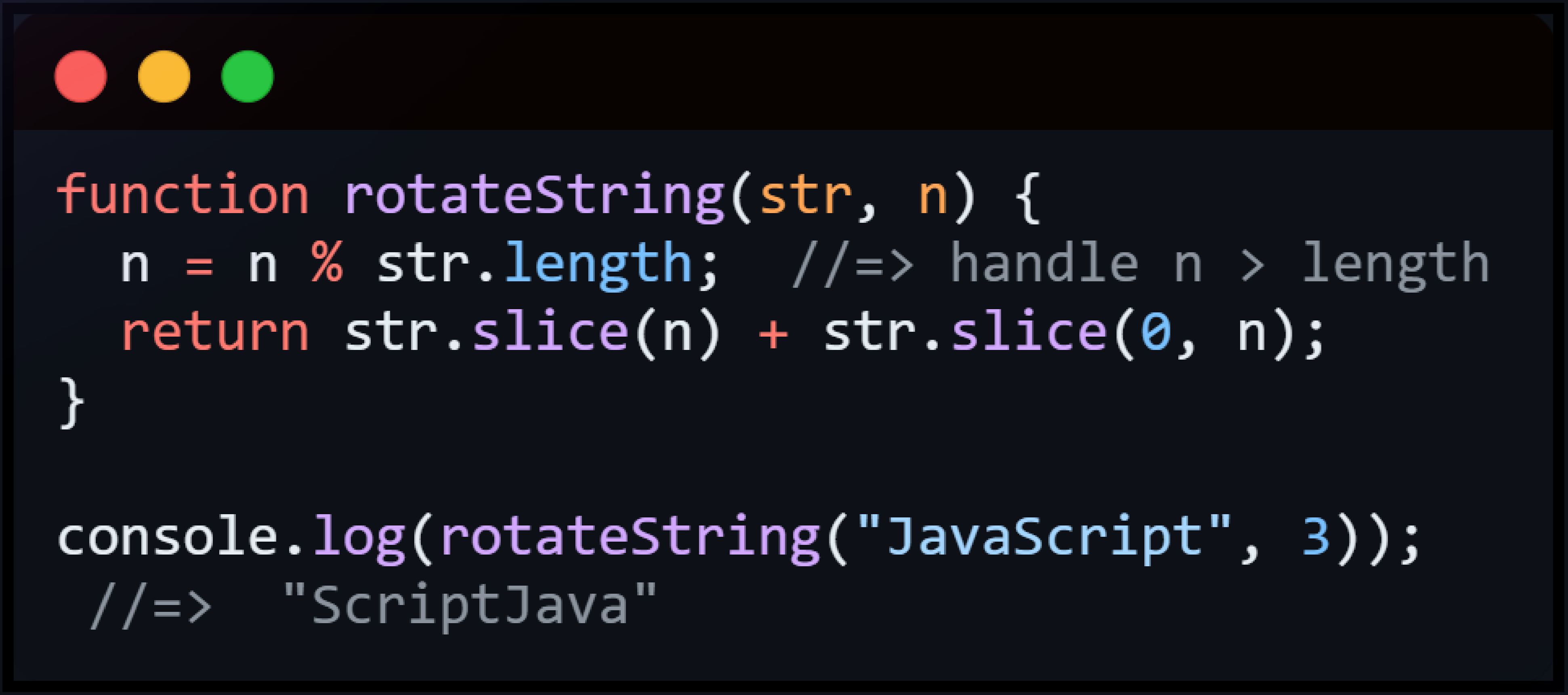


```
function capitalizeWords(str) {
  return str.split(' ')
            .map(word => word[0].toUpperCase() + word.slice(1))
            .join(' ');
}

console.log(capitalizeWords("hello world from js"));
//=>  "Hello World From Js"
```

Explanation :

1. Split sentence into words.
2. Capitalize first letter of each word.
3. Join words back into a string.

Rotate a String by N Characters

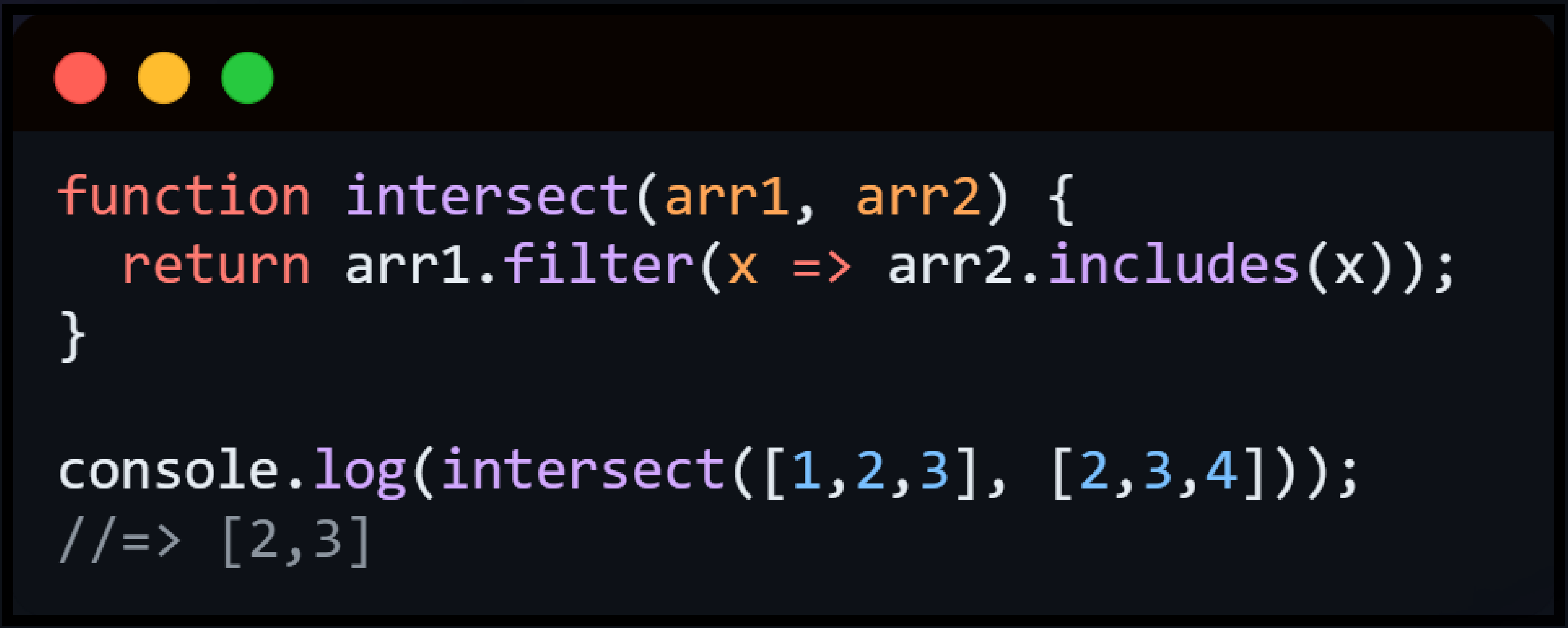


```
function rotateString(str, n) {
  n = n % str.length;  //=> handle n > length
  return str.slice(n) + str.slice(0, n);
}

console.log(rotateString("JavaScript", 3));
 //=>  "ScriptJava"
```

Explanation :

1. Handle large rotations by using modulo: n = n % str.length.
2. Slice the string into two parts at index n.
3. Swap and concatenate the two parts.

Check if Two Arrays Intersect

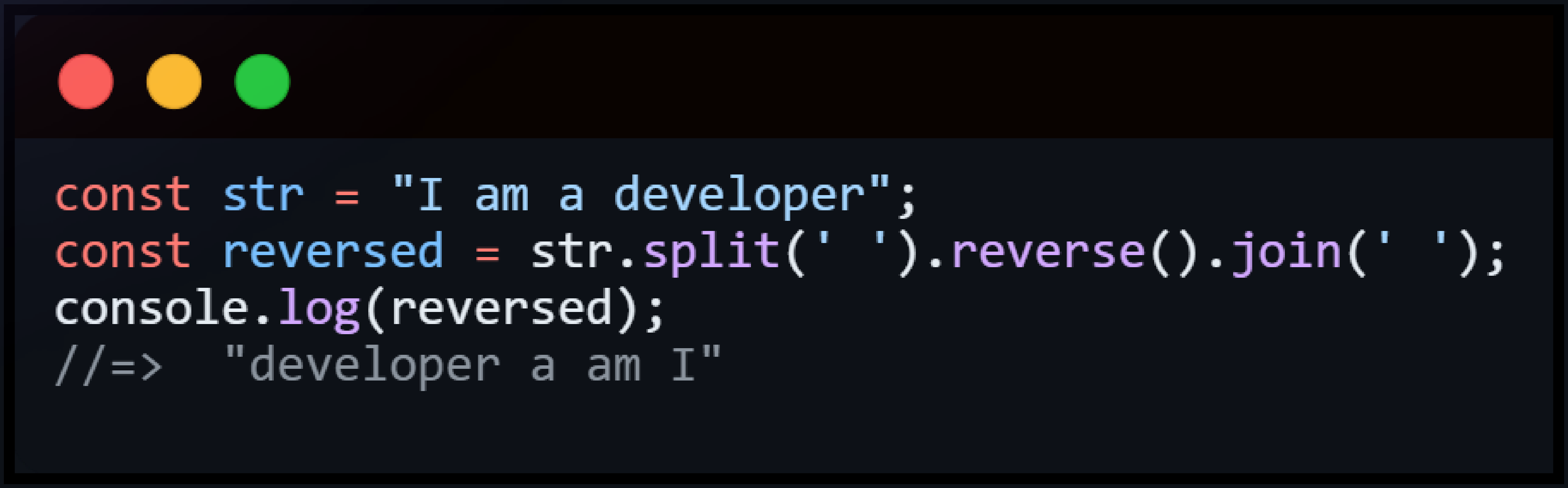


```javascript
function intersect(arr1, arr2) {
  return arr1.filter(x => arr2.includes(x));
}

console.log(intersect([1,2,3], [2,3,4]));
//=> [2,3]
```

Explanation :

1. Filter elements in first array that exist in second.
2. Returns common elements.

Reverse Words in a Sentence



```
const str = "I am a developer";
const reversed = str.split(' ').reverse().join(' ');
console.log(reversed);
//=>  "developer a am I"
```

Explanation :

1. Split the sentence into words using split(' ').
2. Reverse the array of words using .reverse().
3. Join the words back into a string with .join(' ').