map()

filter()

reduce()

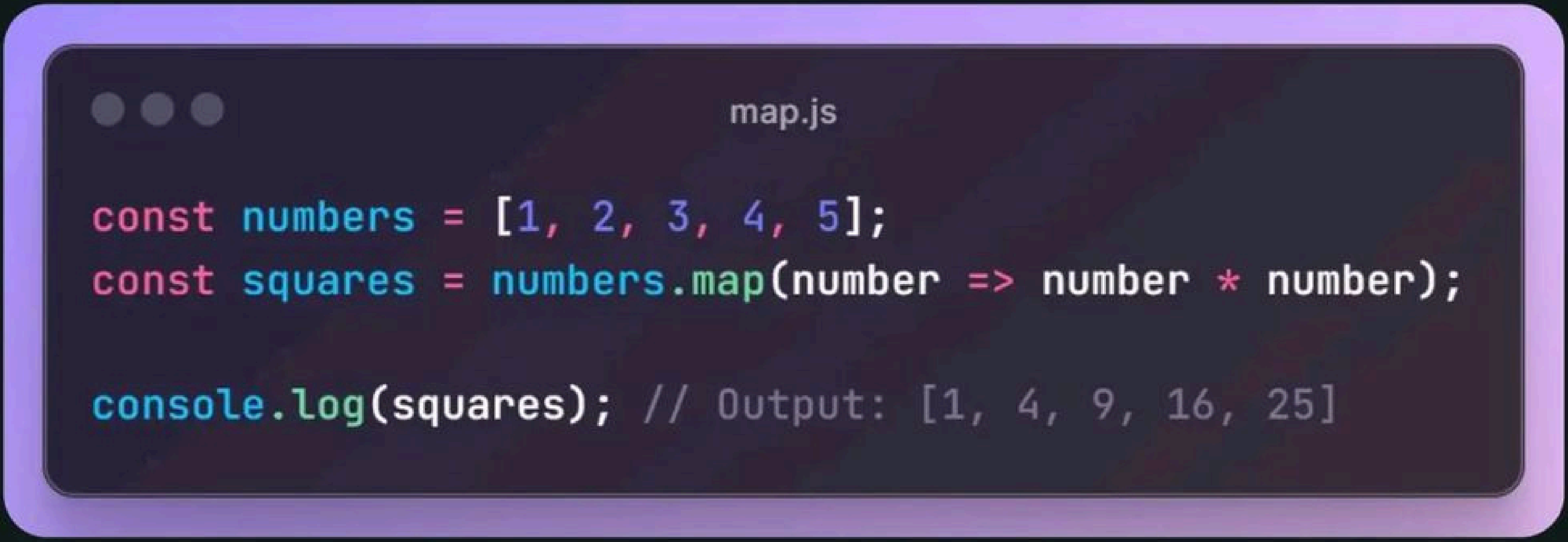
JAVASCRIPT TRIO

These methods iterate over arrays, transforming or manipulating them without creating new ones.

map()	⟶	The Transforming Magic
filter()	⟶	The Selective Power
reduce()	⟶	The Accumulation Champion

MAP()

Applies a function to each element & returns a new array with the transformed values.



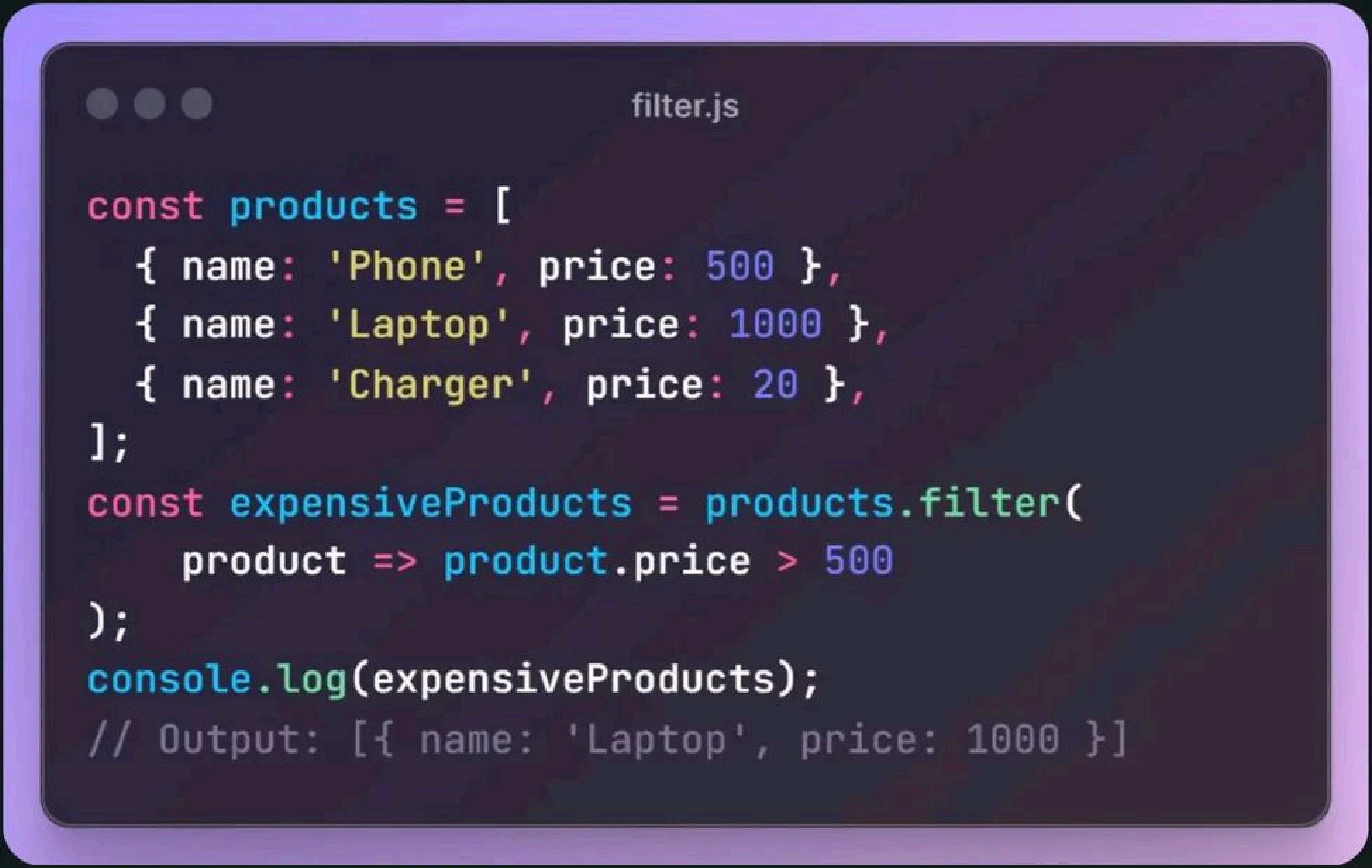
map.js

```
const numbers = [1, 2, 3, 4, 5];
const squares = numbers.map(number => number * number);

console.log(squares); // Output: [1, 4, 9, 16, 25]
```

FILTER()

Creates a new array containing only elements that pass a test implemented by the provided function.

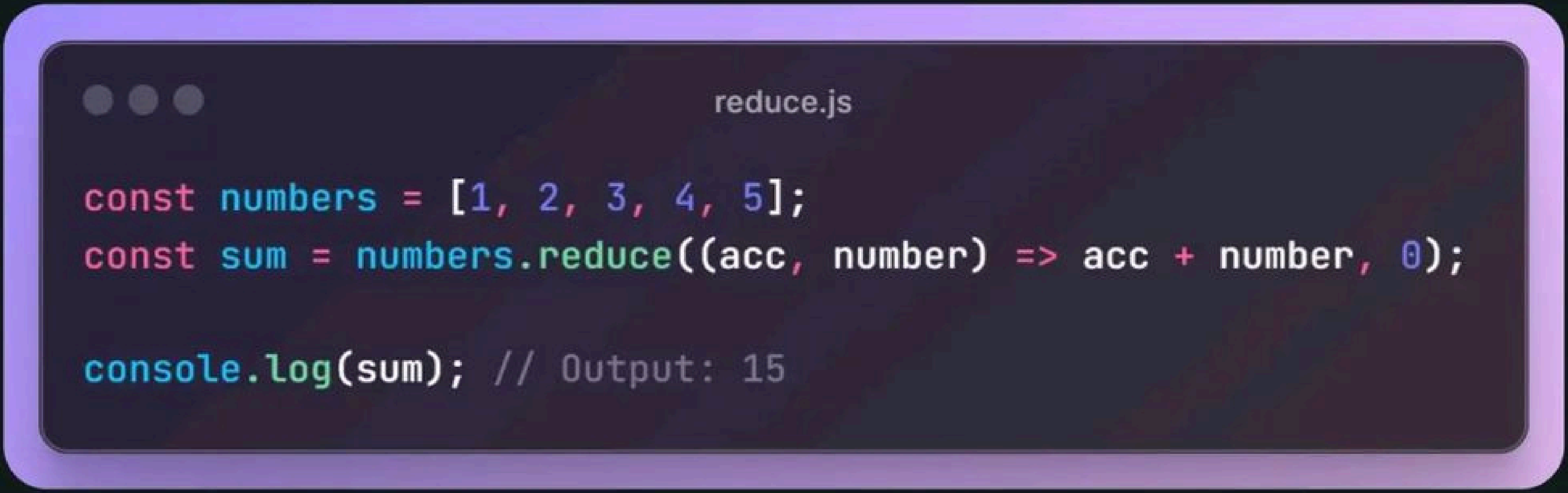


filter.js

```
const products = [
  { name: 'Phone', price: 500 },
  { name: 'Laptop', price: 1000 },
  { name: 'Charger', price: 20 },
];
const expensiveProducts = products.filter(
    product => product.price > 500
);
console.log(expensiveProducts);
// Output: [{ name: 'Laptop', price: 1000 }]
```

REDUCE()

Reduces an array to a single value by applying a function against an accumulator and each element.



reduce.js

```
const numbers = [1, 2, 3, 4, 5];
const sum = numbers.reduce((acc, number) => acc + number, 0);

console.log(sum); // Output: 15
```

WHEN TO USE EACH?

map()

↳ Create a new array with transformed elements.

filter()

↳ Create a new array with elements matching a condition.

reduce()

↳ Combine elements into a single value.

FOLLOW