Improve your productivity with these JavaScript hacks

How can I improve my productivity? 🤔

Mastering essential JavaScript hacks can boost your effectiveness and save development time. Here are some key tips for daily coding

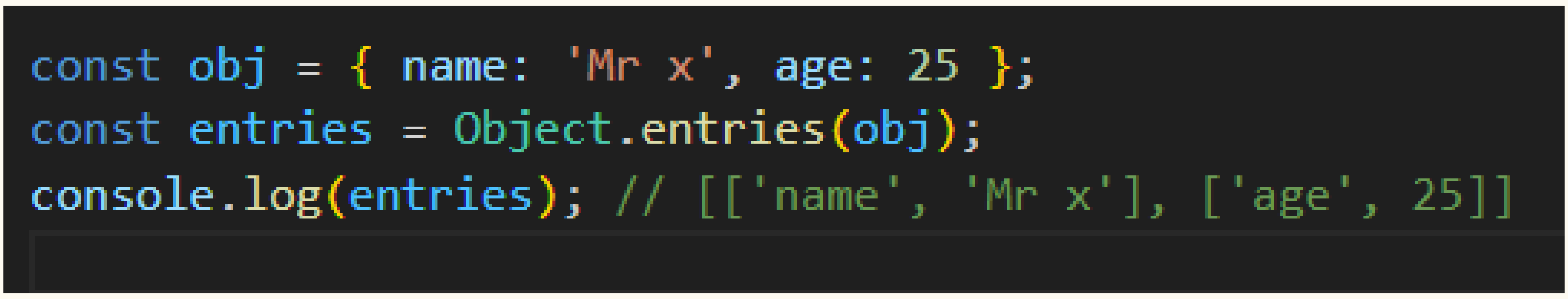
›

Scroll to next page to read more

/in/mulukenm

1. Object.entries

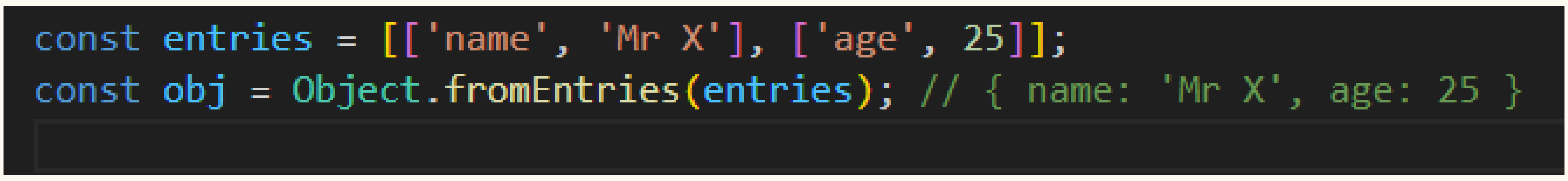
Object.entries takes an object and converts it into an array of key-value pairs. This is useful when you need to iterate over an object’s properties or transform it in some way.



```
const obj = { name: 'Mr x', age: 25 };
const entries = Object.entries(obj);
console.log(entries); // [['name', 'Mr x'], ['age', 25]]
```

2. Object.fromEntries

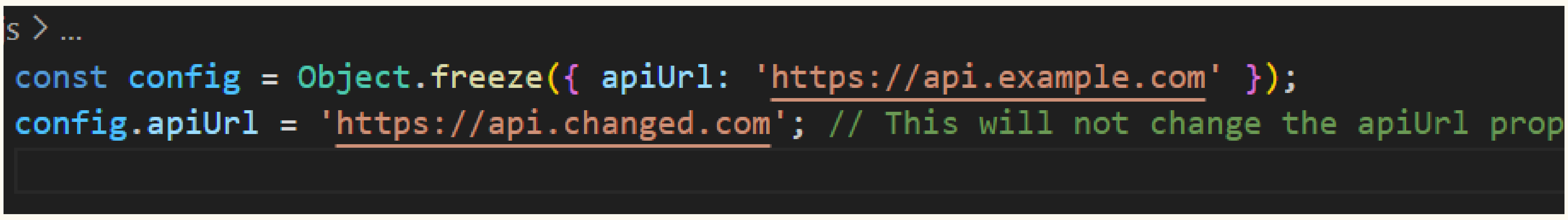
Object.fromEntries allows you to convert an array of key-value pairs back into an object. This is especially useful when working with Object.entries for transformations. Its the reverse of Object.entries



```
const entries = [['name', 'Mr X'], ['age', 25]];
const obj = Object.fromEntries(entries); // { name: 'Mr X', age: 25 }
```

3. Object.freeze

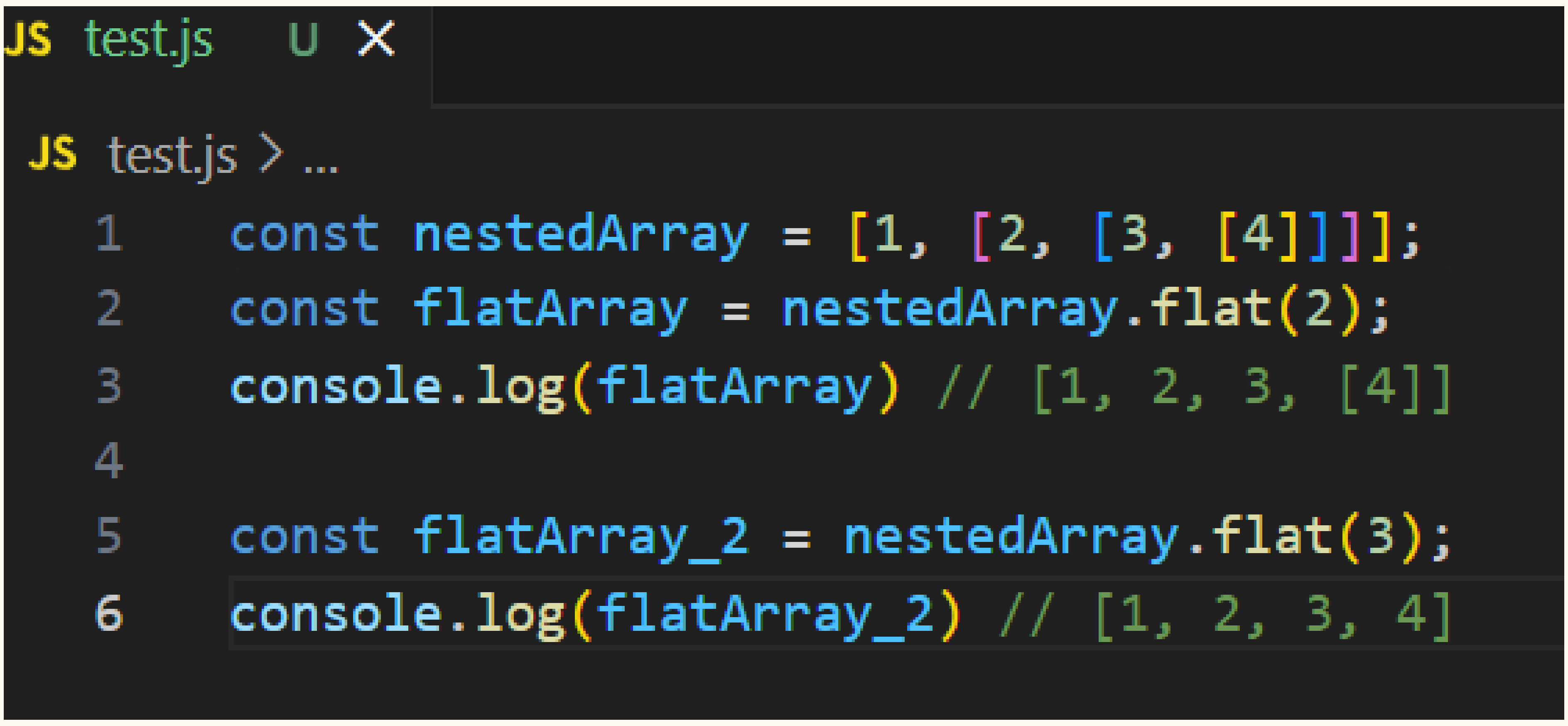
Object.freeze creates a shallow immutable object, which can help in ensuring that objects are not modified. This is great for constant values and configuration settings.



```
const config = Object.freeze({ apiUrl: 'https://api.example.com' });
config.apiUrl = 'https://api.changed.com'; // This will not change the apiUrl prop
```

4. Array.prototype.flat

Array.prototype.flat can be used to flatten multi-dimensional arrays into a single dimension, which simplifies the handling of deeply nested array structures.



```
const nestedArray = [1, [2, [3, [4]]]];
const flatArray = nestedArray.flat(2);
console.log(flatArray) // [1, 2, 3, [4]]

const flatArray_2 = nestedArray.flat(3);
console.log(flatArray_2) // [1, 2, 3, 4]
```

Thank you for visiting. See you next time!

/in/mulukenm