CODEMIND
Your Success Partner

ES 6 Features

By CODEMIND Technology

Contact us 966 5044 698
966 5044 598

advance

JavaScript ES6 (also known as ECMAScript 2015 or ECMAScript 6) is the newer version of JavaScript that was introduced in 2015.

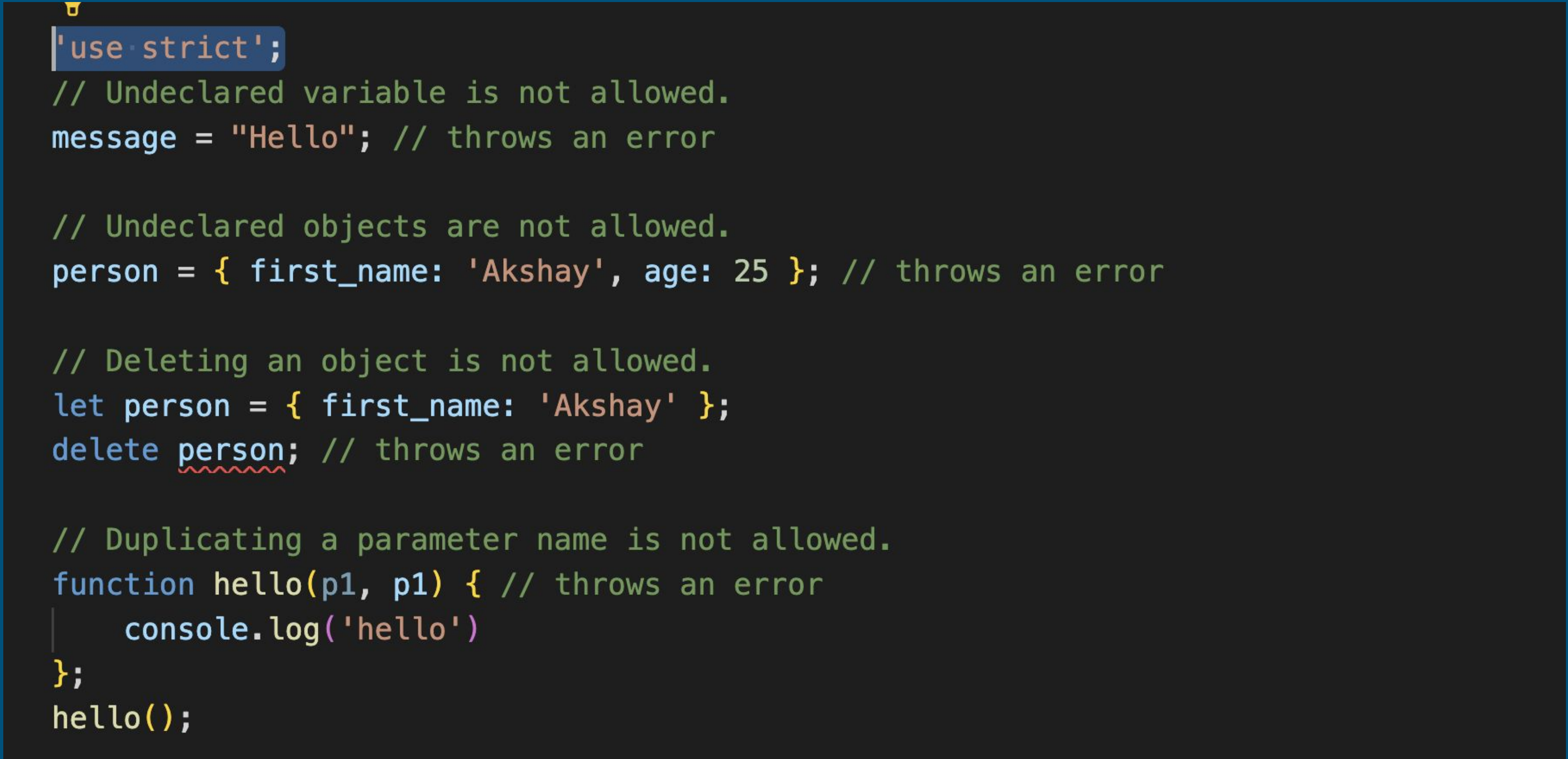
JavaScript “use strict”

In JavaScript, 'use strict'; states that the code should be executed in 'strict mode'. This makes it easier to write good and secure JS code

Benefits of Strict Mode

- Helps to write a cleaner code
- Changes previously accepted silent errors (bad syntax) into real errors and throws an error message
- Makes it easier to write "secure" JavaScript

JavaScript “use-strict”



```
'use strict';
// Undeclared variable is not allowed.
message = "Hello"; // throws an error

// Undeclared objects are not allowed.
person = { first_name: 'Akshay', age: 25 }; // throws an error

// Deleting an object is not allowed.
let person = { first_name: 'Akshay' };
delete person; // throws an error

// Duplicating a parameter name is not allowed.
function hello(p1, p1) { // throws an error
    console.log('hello')
};
hello();
```

String template - Template Literals

Use backticks (` `) characters to define string template.

Few usage of string template are:

1. In a string putting a word with double quote “ ”

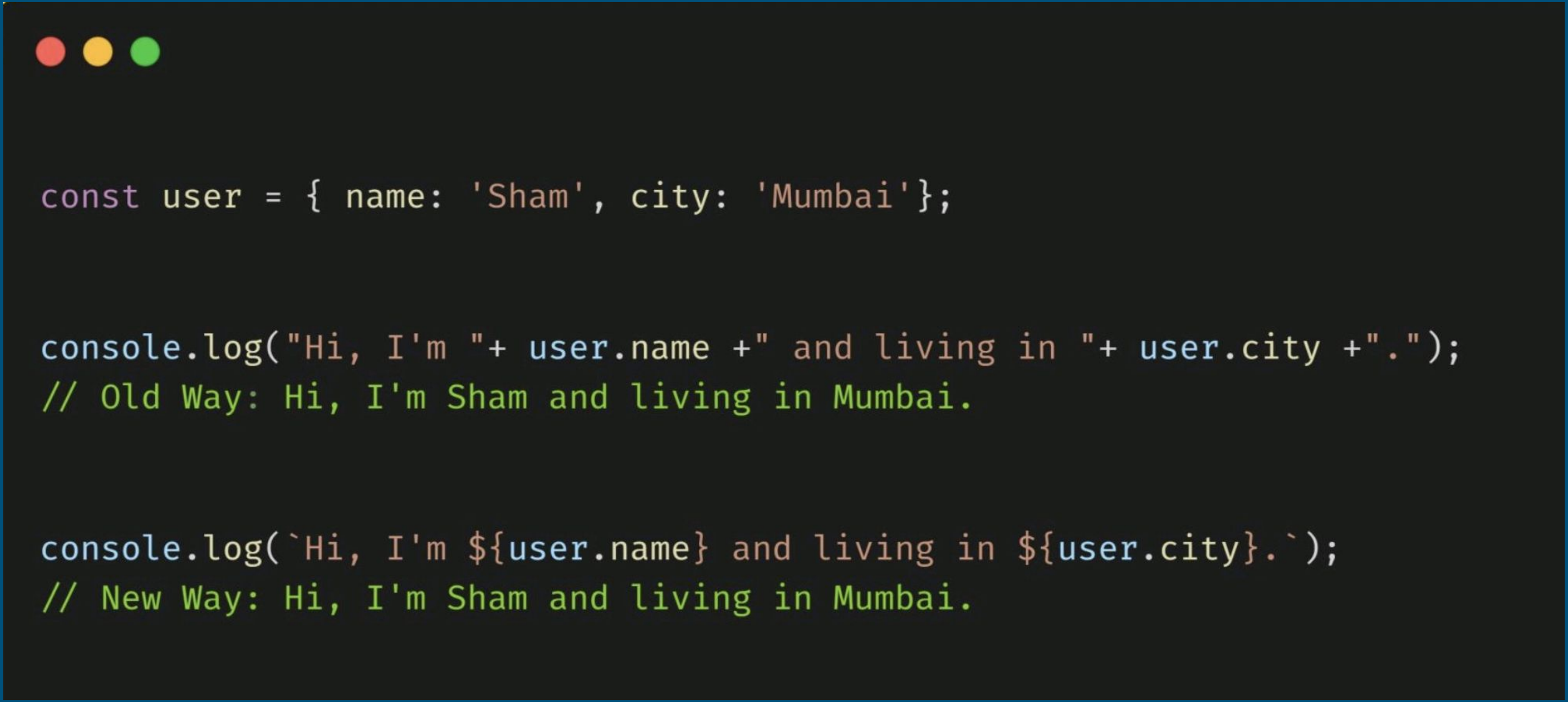
```
var message = `Hello, Very Good Morning - "Sachin" `;
console.log(message);
```

2. Variable Substitution

```
var firstName = ”Codemind”;
var lastName = “Technology”
console.log(`First Name: ${firstName} and Last Name: ${lastName}`);
```

Note: `` (backticks) is different than ‘ ’ (single quote)

String template - ES6 feature



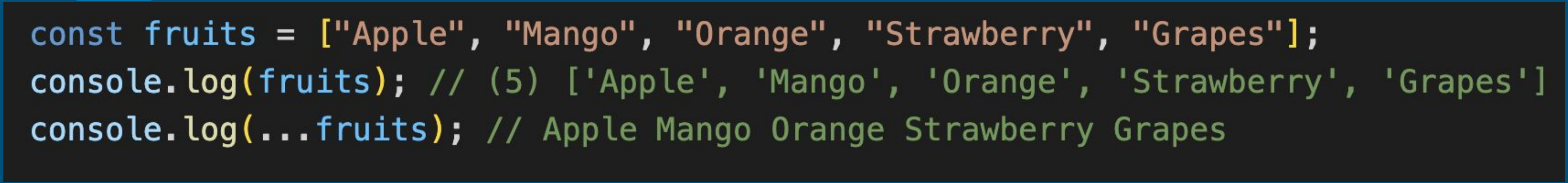
```
const user = { name: 'Sham', city: 'Mumbai'};


console.log("Hi, I'm "+ user.name +" and living in "+ user.city +".");
// Old Way: Hi, I'm Sham and living in Mumbai.


console.log(`Hi, I'm ${user.name} and living in ${user.city}.`);
// New Way: Hi, I'm Sham and living in Mumbai.
```

Spread Operator: … (3 dots)

The spread operator ... is used to expand an array or spread the elements



```
const fruits = ["Apple", "Mango", "Orange", "Strawberry", "Grapes"];
console.log(fruits); // (5) ['Apple', 'Mango', 'Orange', 'Strawberry', 'Grapes']
console.log(...fruits); // Apple Mango Orange Strawberry Grapes
```

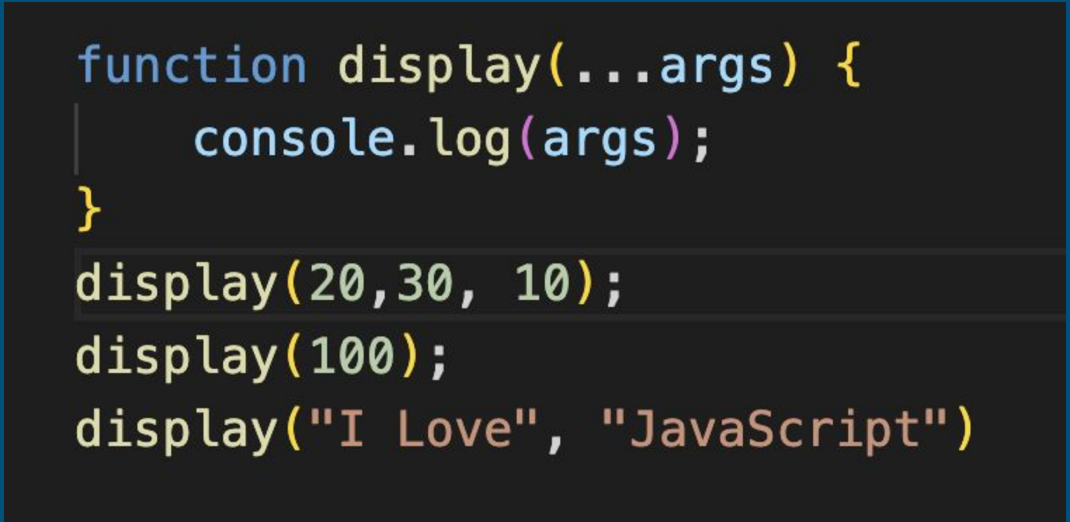
In this code: `console.log(...fruits);`

Is equivalent to: `console.log("Apple", "Mango", "Orange", "Strawberry", "Grapes");`

Rest Parameter

- When the spread operator (...) used as parameter in the function, it i/k/a Rest Parameter.
- We can accept multiple arguments in a function call using the rest parameter
- Rule: Rest parameter should be the last argument in a function

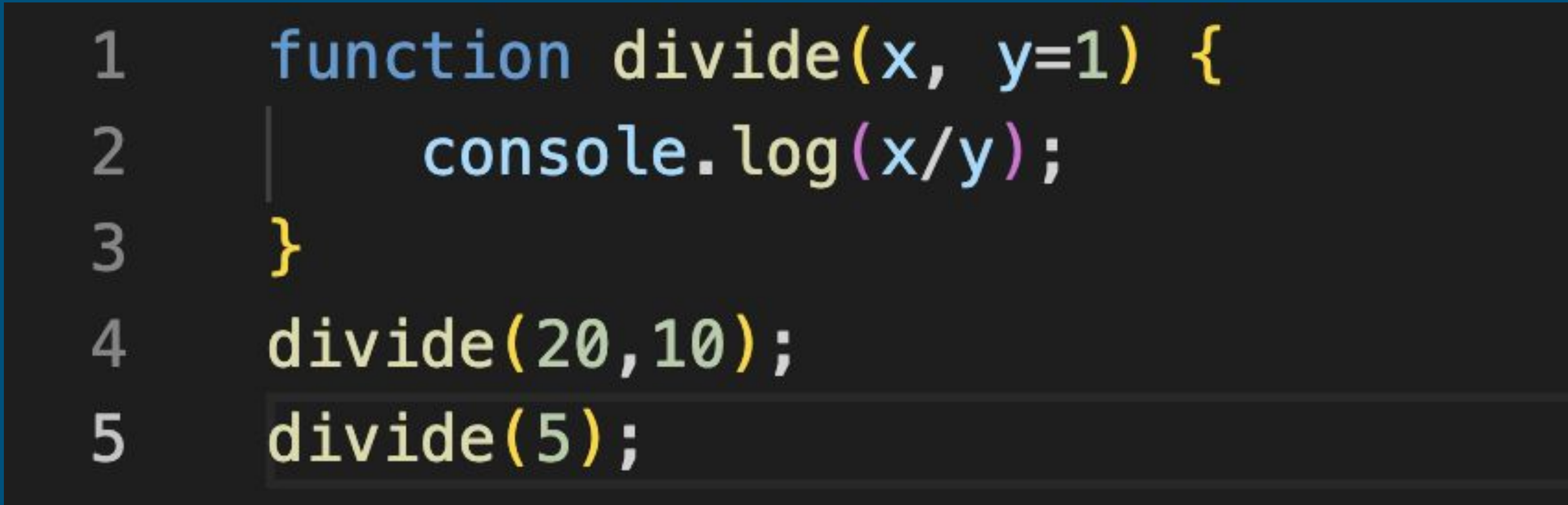
Note: Using the Rest parameter, will pass the arguments as array elements



```
function display(...args) {
    console.log(args);
}
display(20,30, 10);
display(100);
display("I Love", "JavaScript")
```

Function default parameters

- In JS, default function parameters allow you to initialize named parameters with default values if no values or undefined are passed into the function.
- As shown in below snippet. when we don’t pass the value for y parmeter, it will take 1 as default value inside divide( ) function invoked at line number → 5

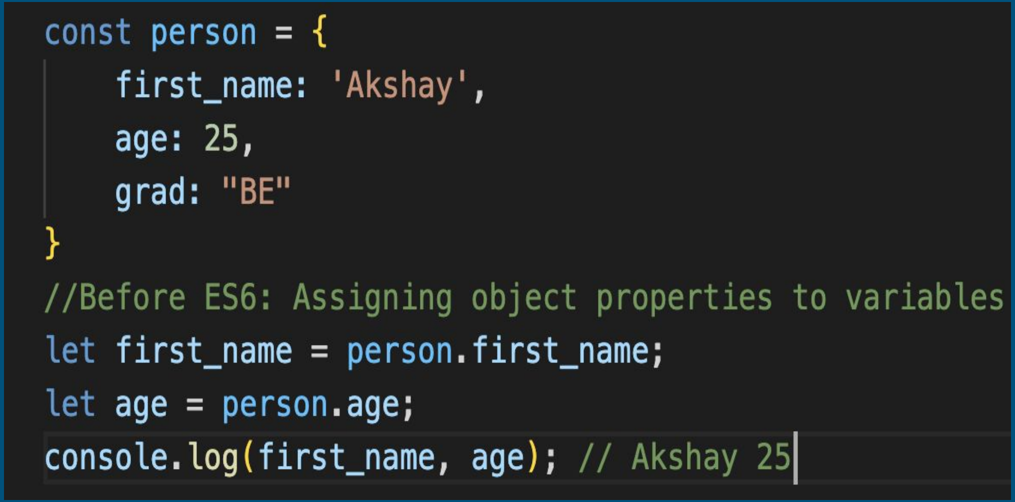


```
1    function divide(x, y=1) {
2        console.log(x/y);
3    }
4    divide(20,10);
5    divide(5);
```

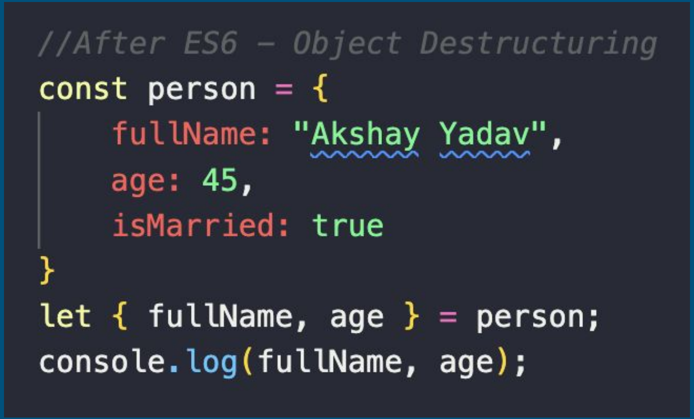
Destructuring: Object Destructuring, Array Destructuring

This destructuring ES6 features makes easy to extract properties from an object or an element from an array.

This feature will make easy to assign object properties to distinct variables.
Note: When destructuring objects, we should use the same name for the variable as the corresponding object properties.



```
const person = {
    first_name: 'Akshay',
    age: 25,
    grad: "BE"
}
//Before ES6: Assigning object properties to variables
let first_name = person.first_name;
let age = person.age;
console.log(first_name, age); // Akshay 25
```

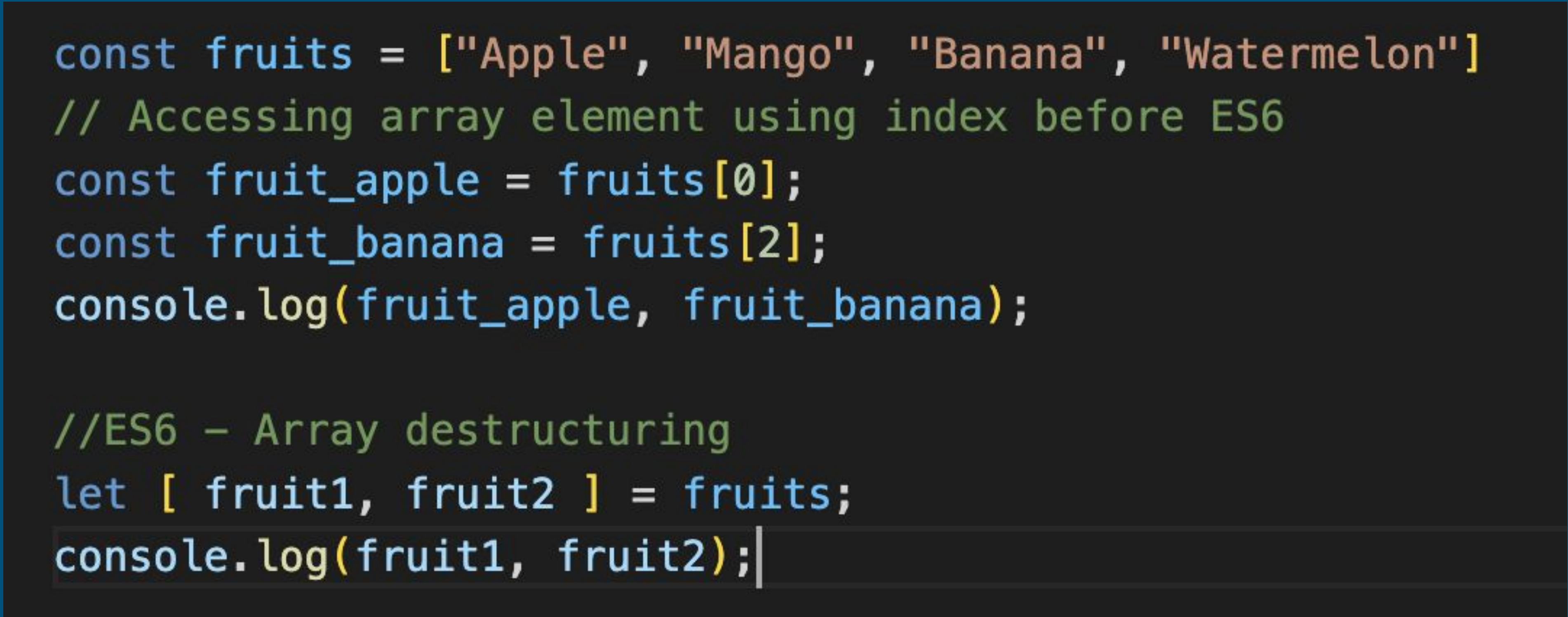


```
//After ES6 - Object Destructuring
const person = {
    fullName: "Akshay Yadav",
    age: 45,
    isMarried: true
}
let { fullName, age } = person;
console.log(fullName, age);
```

Array Destructuring

With the help of this feature we can extract array element into separate variable

Syntax → let [element1, element2, element3] = array_name

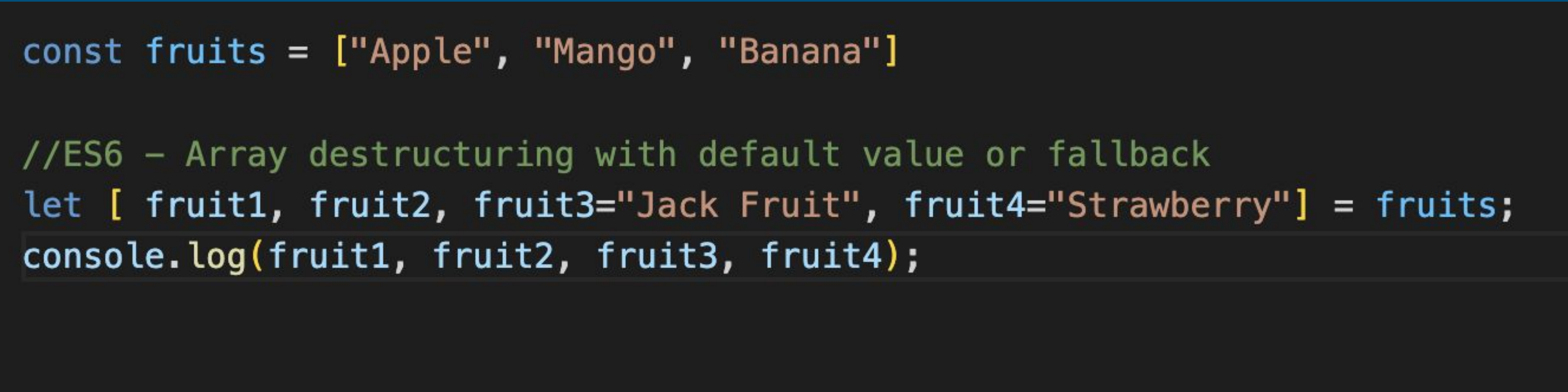


```
const fruits = ["Apple", "Mango", "Banana", "Watermelon"]
// Accessing array element using index before ES6
const fruit_apple = fruits[0];
const fruit_banana = fruits[2];
console.log(fruit_apple, fruit_banana);

//ES6 - Array destructuring
let [ fruit1, fruit2 ] = fruits;
console.log(fruit1, fruit2);
```

Array Destructuring with default values

Destructuring allows a default value to be assigned to a variable if no value or undefined is passed. It is like providing a fallback when nothing is found



```
const fruits = ["Apple", "Mango", "Banana"]

//ES6 - Array destructuring with default value or fallback
let [ fruit1, fruit2, fruit3="Jack Fruit", fruit4="Strawberry"] = fruits;
console.log(fruit1, fruit2, fruit3, fruit4);
```

Assignment:

const arrayNum = [11, 3, 4, 11, 4, 7, 3];

Remove duplicate element from array

Given String value →

const str = "How are you mate";

Expected output ⇒ "HoW ArE YoU Mate"