ES6 CONCEPTS

interview point

JS

New features in ES6

Frequently used

- Let and Const keywords
- Arrow functions
- The ... operator (spread and rest)
- For/of
- promises
- Default Parameters
- Template Literals
- Destructuring Assignment
- Multi-line Strings

1) Let and const

- The let declaration declares a block-scoped local variable

```
let x = 1;

if (x === 1) {
  let x = 2;

  console.log(x) // Expected output: 2
}

console.log(x);// Expected output: 1
```

- The const keyword allows you to declare a constant and that can't be changed for whole program

```
const x = 16

x = 23          //Type Error
```

2) Arrow functionds

- Arrow functions allows a **short syntax for writing function** expressions.
- You don't need the **function keyword**,
- for single line function : Don't need the return keyword, and the curly brackets.

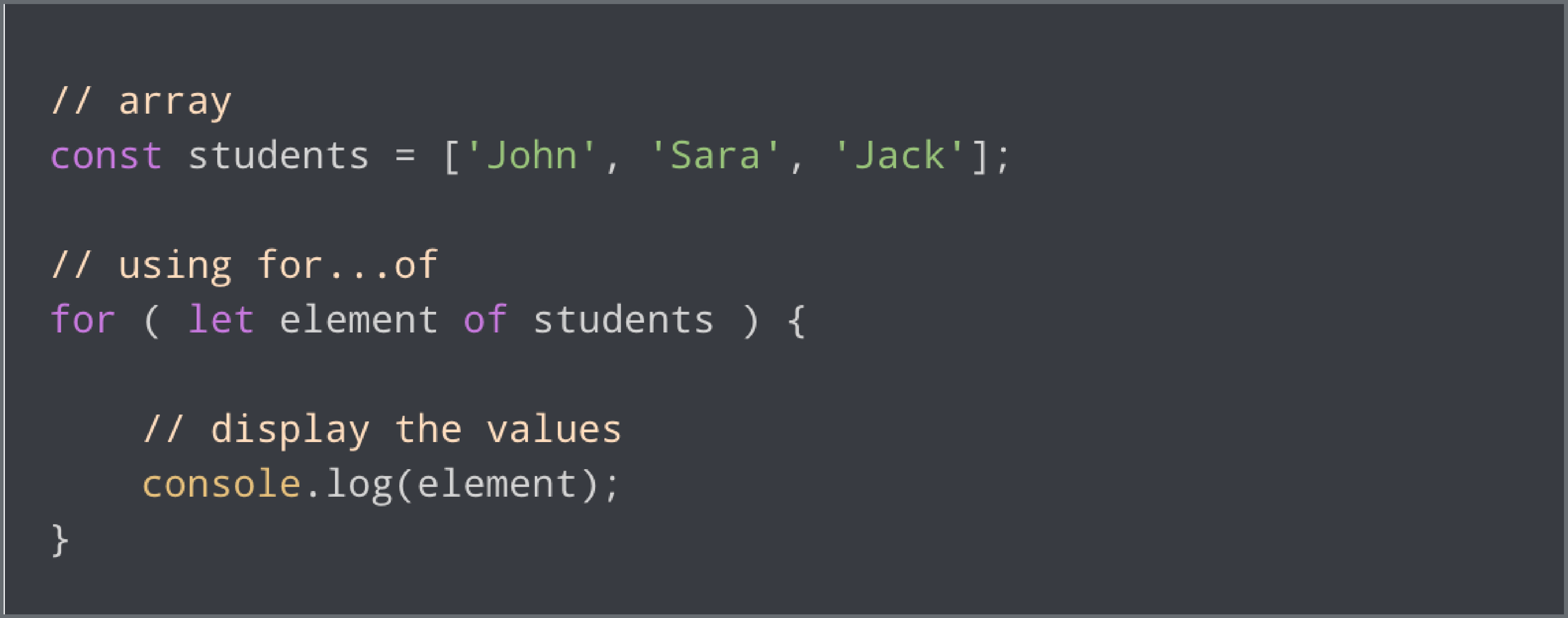
```
//ES5
var x = function(x, y) {
   return x * y;
}

// ES6
const x = (x, y) => x * y; //singleline function

// a multiline function

let sum = (a, b) => {
  let result = a + b;
  return result;      // if we use curly braces, then we need an explicit "return"
};
sum(1, 2) ;
```

3)Spread and Rest operator

The concept in explained in interviewquestion-17

4) For/of

we all know how for loop is written. and ES6 has introduced a simple way for same



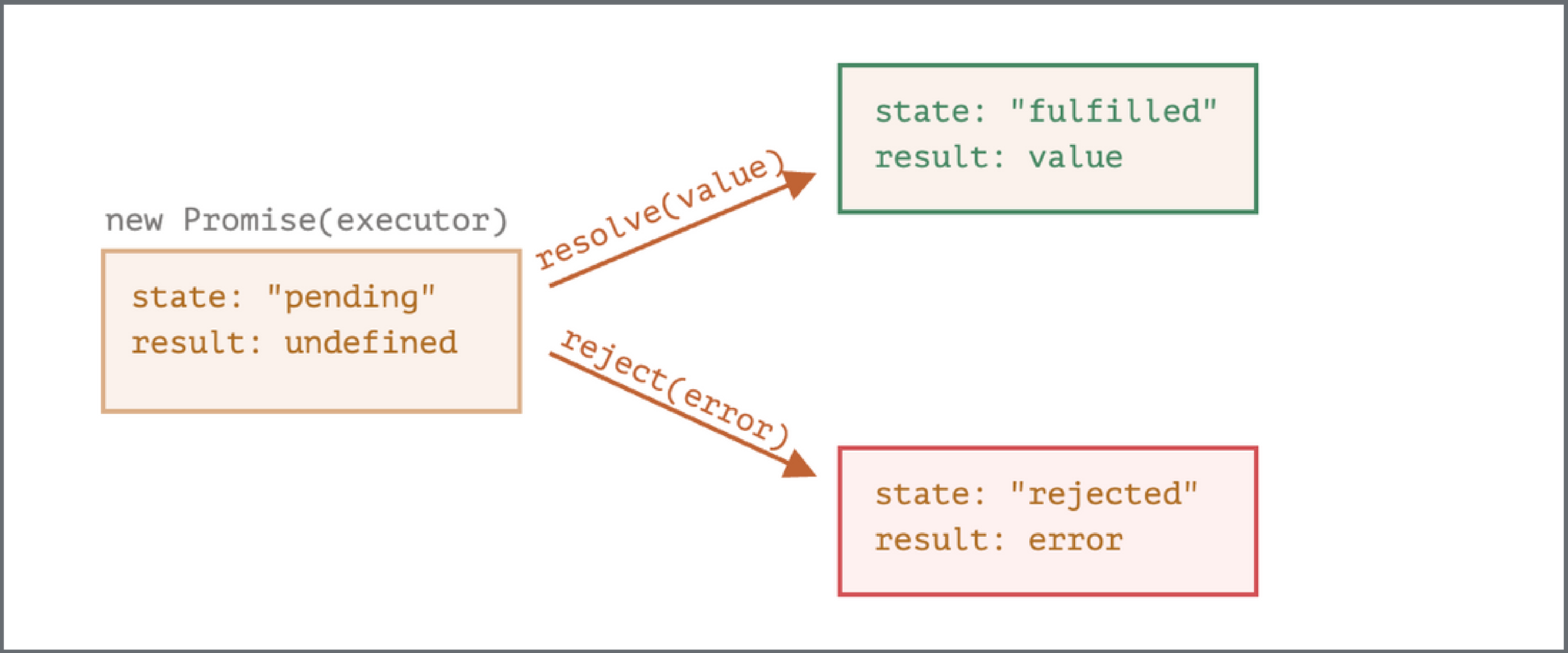
```
// array
const students = ['John', 'Sara', 'Jack'];

// using for...of
for ( let element of students ) {

    // display the values
    console.log(element);
}
```

5)Promises

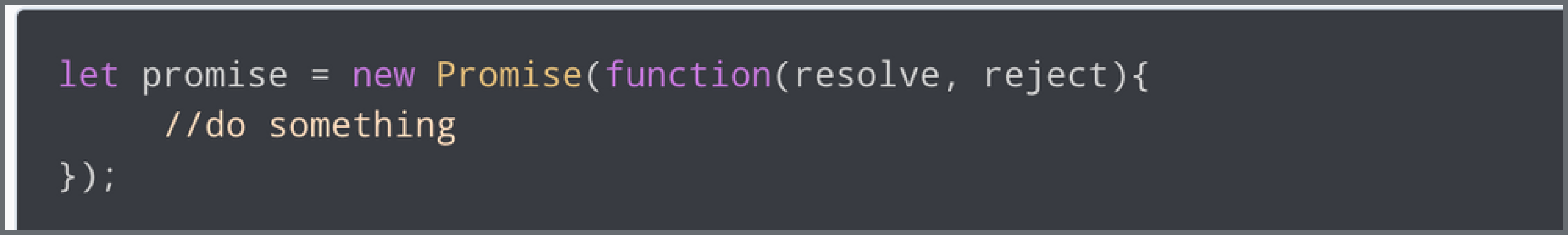
In JavaScript, **a promise is a good way to handle asynchronous operations**. It is used to find out if the asynchronous operation is successfully completed or not.

A promise may have one of three states.

- Pending
- Fulfilled
- Rejected

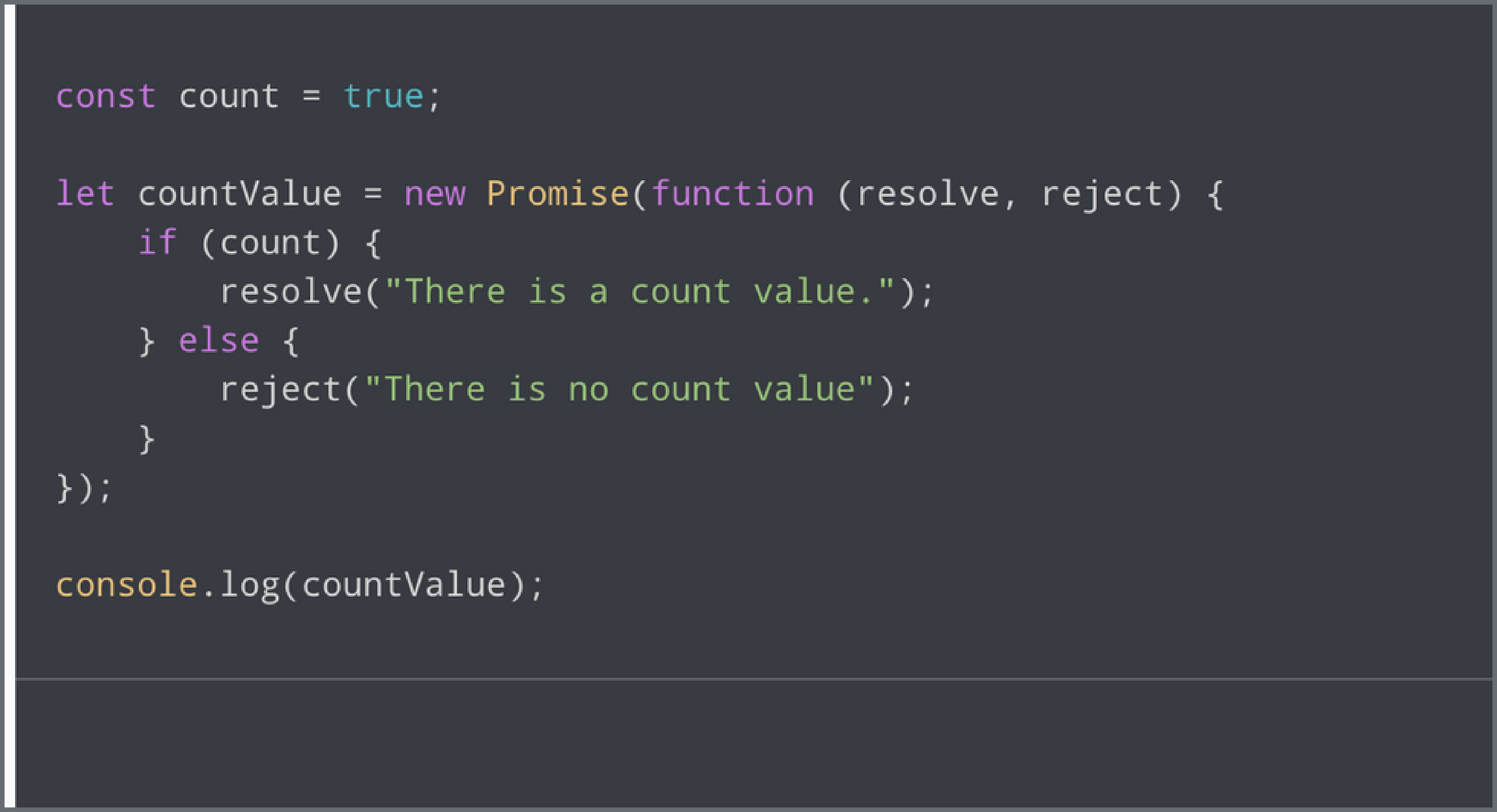


general syntax for promise



```
let promise = new Promise(function(resolve, reject){
     //do something
});
```

A program with promise



```
const count = true;

let countValue = new Promise(function (resolve, reject) {
    if (count) {
        resolve("There is a count value.");
    } else {
        reject("There is no count value");
    }
});

console.log(countValue);
```

We will learn much more into the topic when we have separate interview question on promises

6)Default Parameters

we can pass default parameters so when ever the parameters are not passed , it will use the default parameters

```
function myFunction(x, y = 10) {
  // y is 10 if not passed or undefined
  return x + y;
}
myFunction(5); // will return 15
```

7)Template Literals

In ES6, we can use a new syntax ${PARAMETER} inside of the back-ticked string.

```
var name = `Your name is ${firstName} ${lastName}.`
```

8)Multi-line Strings

In ES6, it is very simple. Just use back-ticks.

```
let poemData = `Johny Johny Yes Papa,
                    Eating sugar?  No, papa!
                     Telling lies? No, papa!
                Open your mouth Ah, ah, ah!`
```

9)Destructuring Assignment

The destructuring assignment syntax is a JavaScript expression that makes it possible to unpack values from arrays, or properties from objects, into distinct variable.

ES6

```
var o = {p: 42, q: true};
var {p, q} = o;

console.log(p); // 42
console.log(q); // true
```

ES5

```
var o = {p: 42, q: true};
var p = o.p;
var q=o.q;

console.log(p); // 42
console.log(q); // true
```