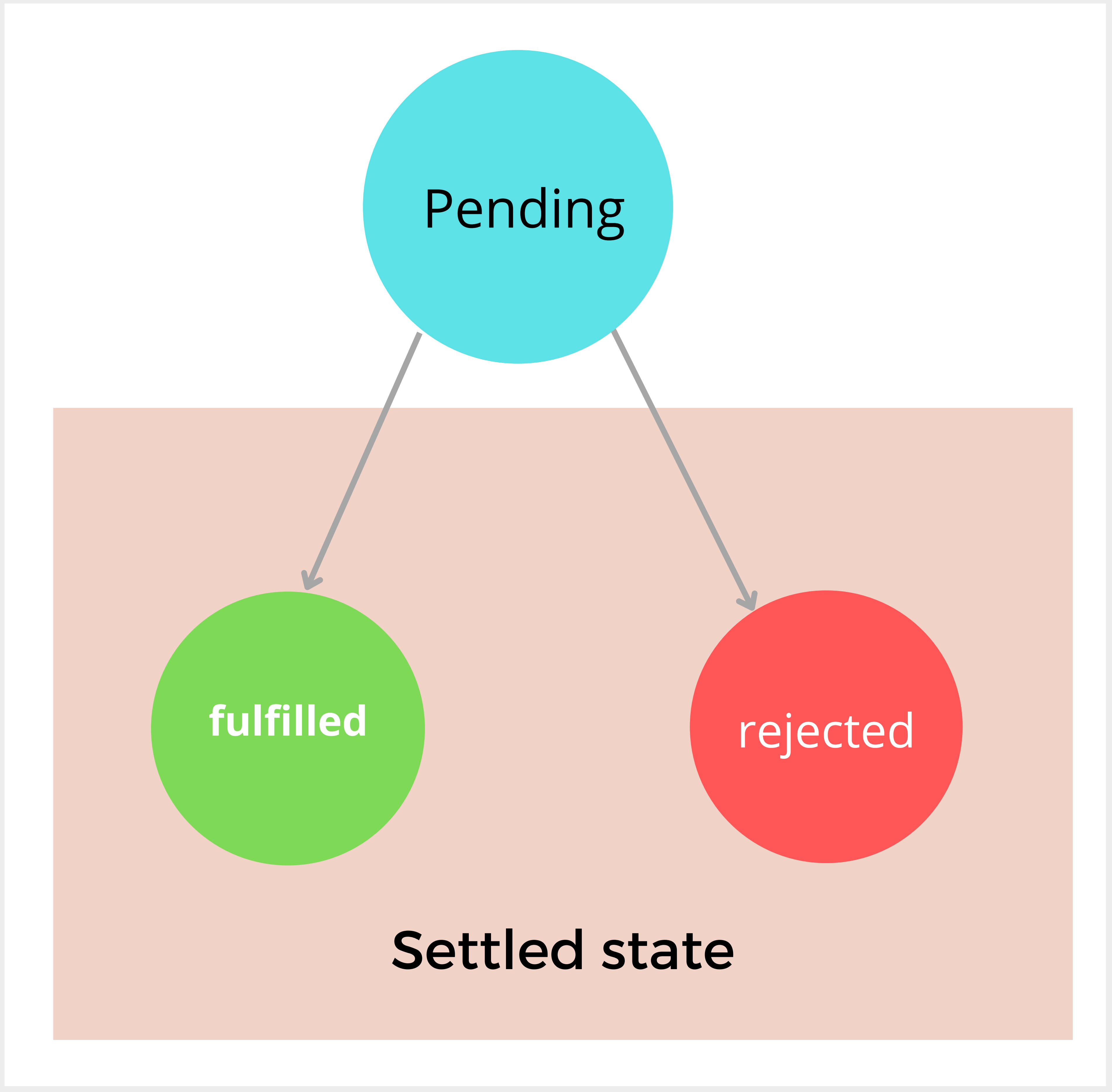
JavaScript Promises

@coder_aishya

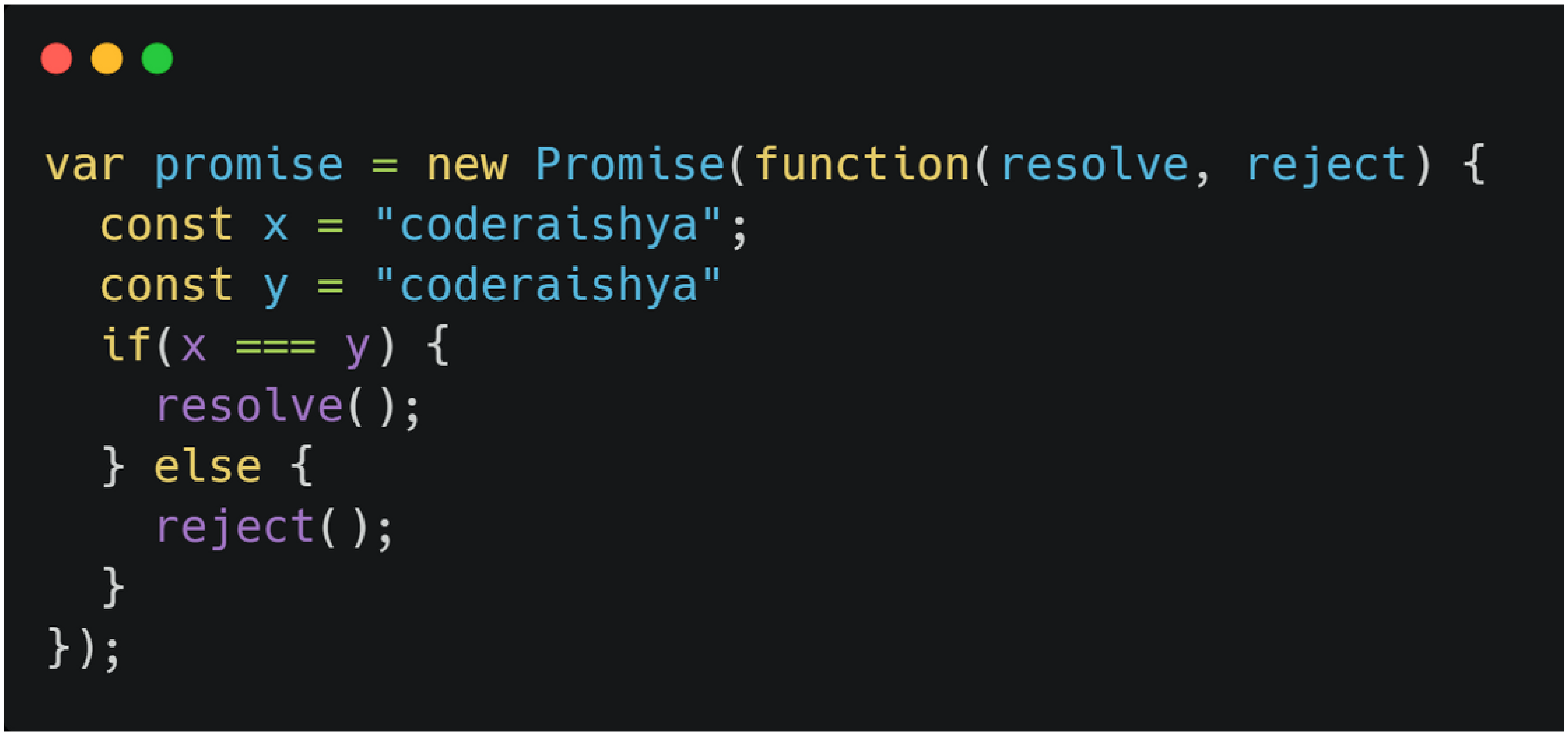
What is a Javascript Promise?

- A Promise is a **JavaScript object** which **produces a value** after an **asynchronous** (async) **operation completes** successfully, **or an error** if it does not complete successfully due to time out, network error, and so on.
- Promises where intoduced in ES6 which improves Code Readability whereas callbacks can create callback hell leading to unmanageable code.

Promise States

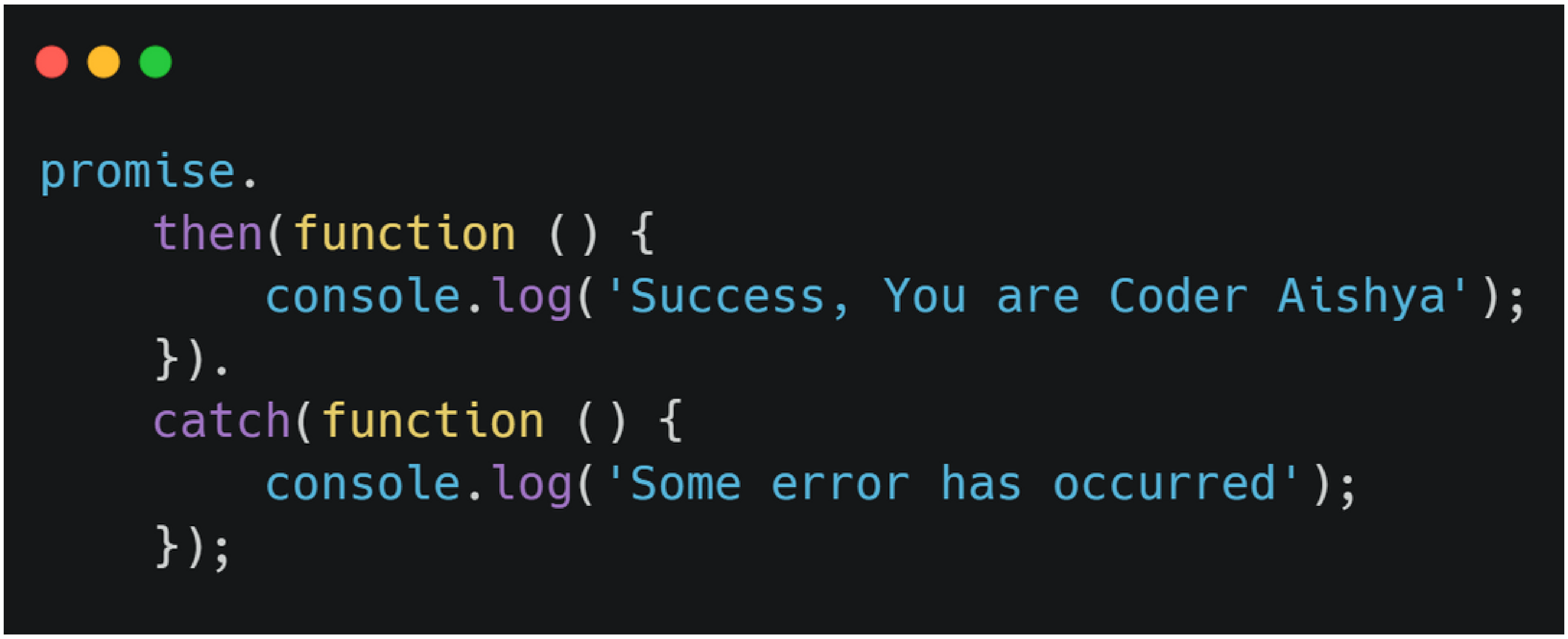


Creating a Promise



```
var promise = new Promise(function(resolve, reject) {
  const x = "coderaishya";
  const y = "coderaishya"
  if(x === y) {
    resolve();
  } else {
    reject();
  }
});
```

Calling a Promise



```
promise.
    then(function () {
        console.log('Success, You are Coder Aishya');
    }).
    catch(function () {
        console.log('Some error has occurred');
    });
```

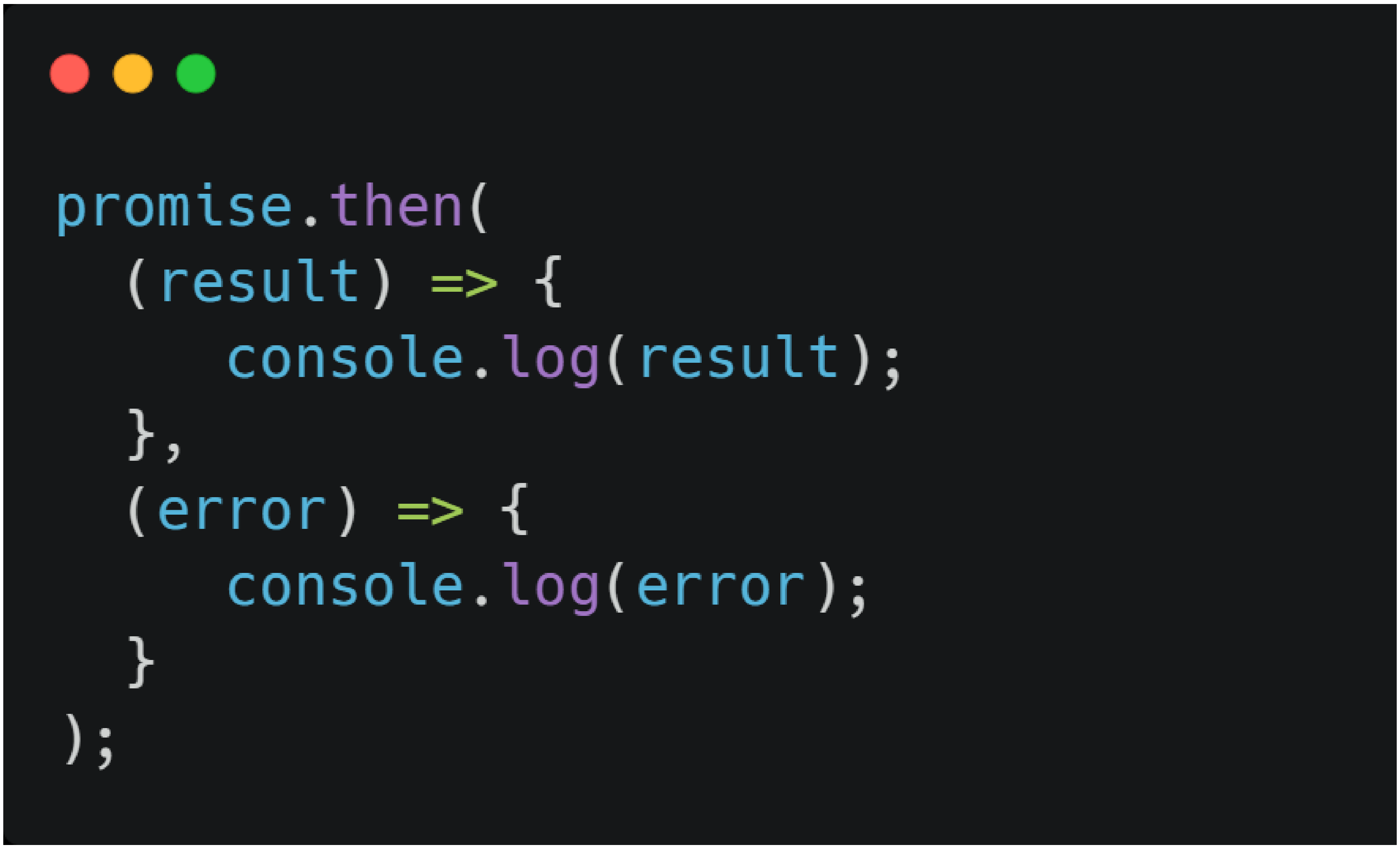
@coder_aishya

Parameters

- Promise constructor takes only one argument which is a **callback function** (and that callback function is also referred as anonymous function too).
- Callback function takes two arguments, **resolve and reject**
- Perform operations inside the callback function and **if everything went well then call resolve.**
- If desired operations **do not go well then call reject.**

The .then() Promise Handler

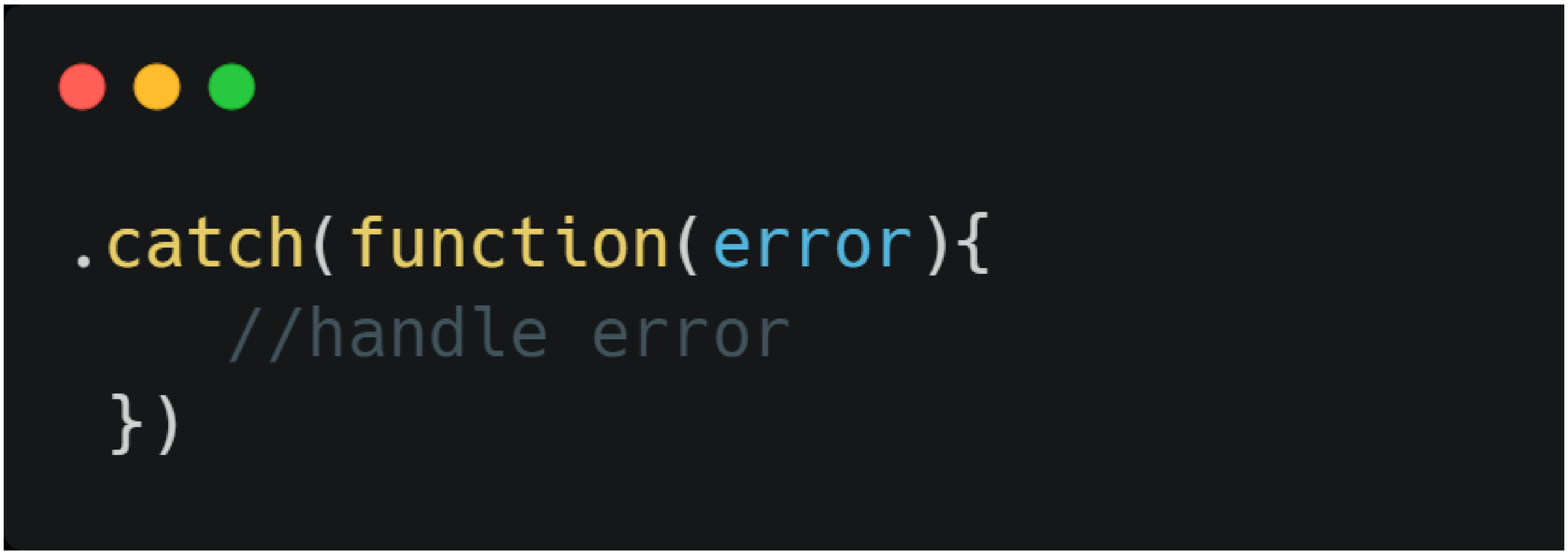
We get a .then() method from every promise. The sole purpose of this method is to let the consumer **know about the outcome of a promise.** It accepts two functions as arguments, result and error.



```
promise.then(
  (result) => {
     console.log(result);
  },
  (error) => {
     console.log(error);
  }
);
```

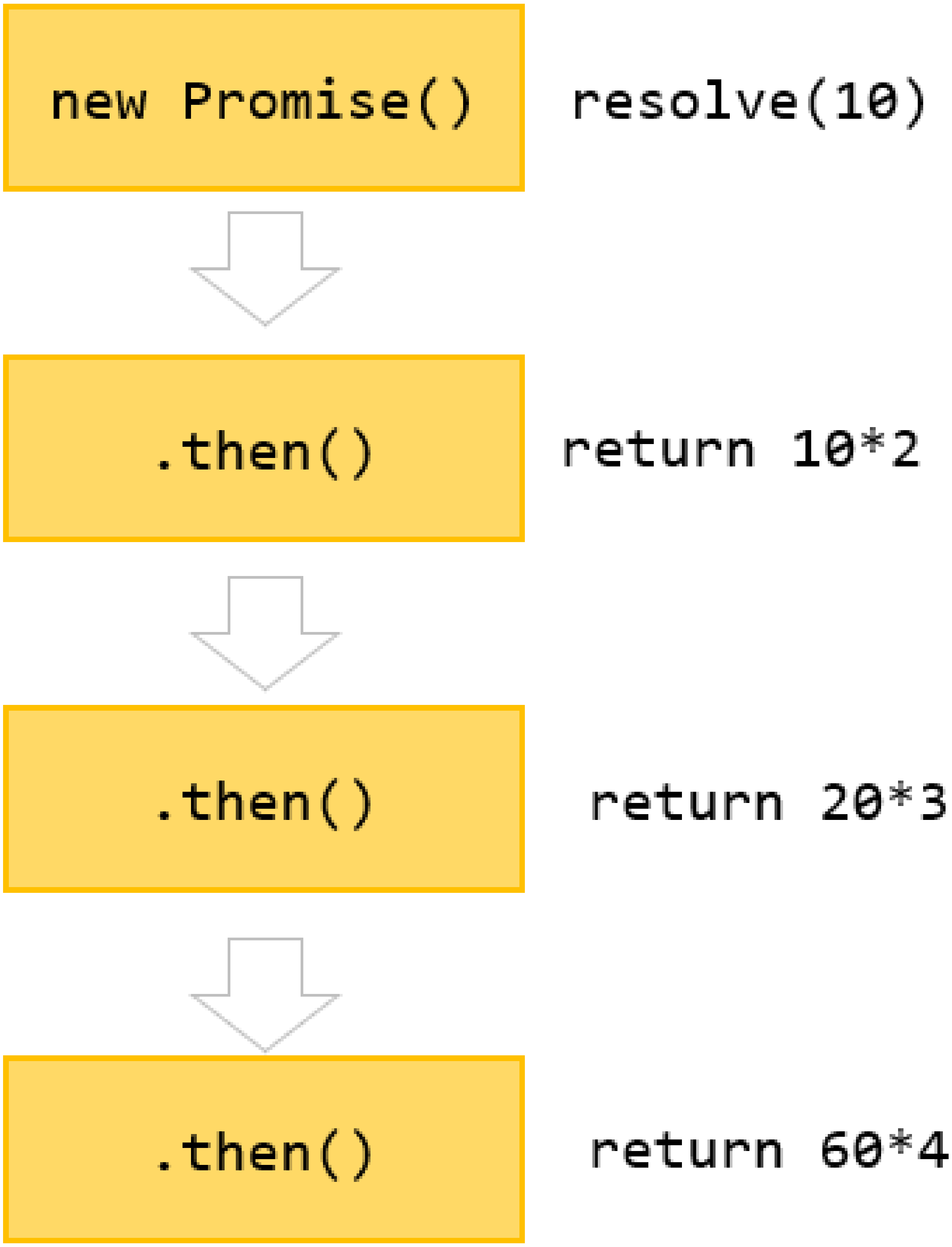
The .catch() Promise Handler

You can use this handler method to handle errors (rejections) from promises. It is a much better syntax to handle the error situation than handling it using the .then() method.



```
.catch(function(error){
     //handle error
 })
```

Promise chaining



new Promise() resolve(10)

.then() return 10*2

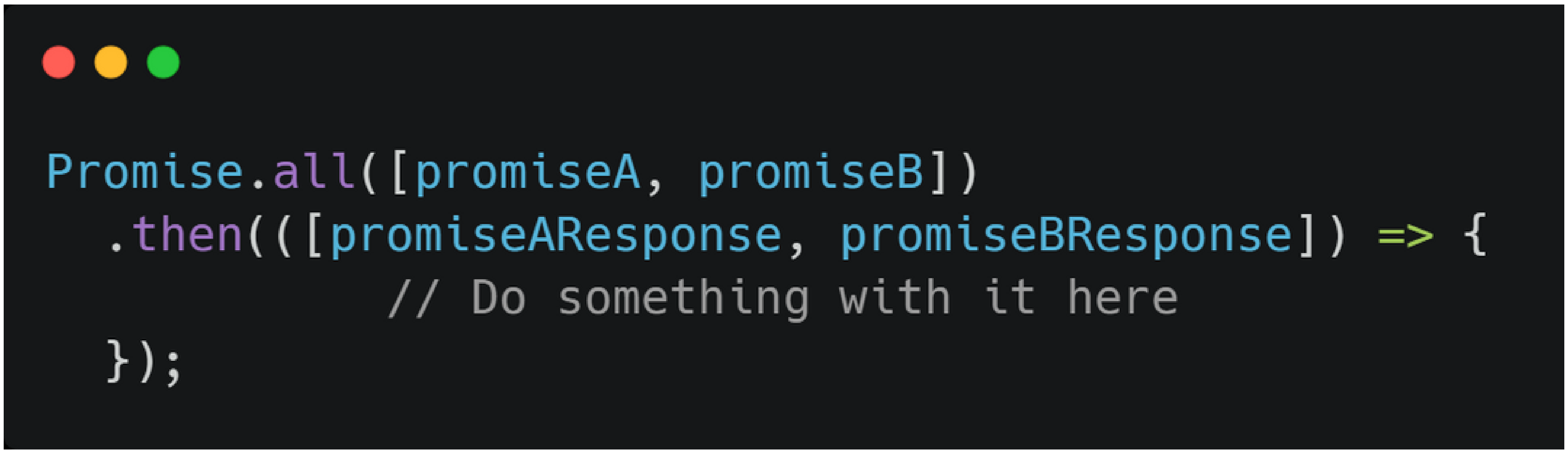
.then() return 20*3

.then() return 60*4

@coder_aishya

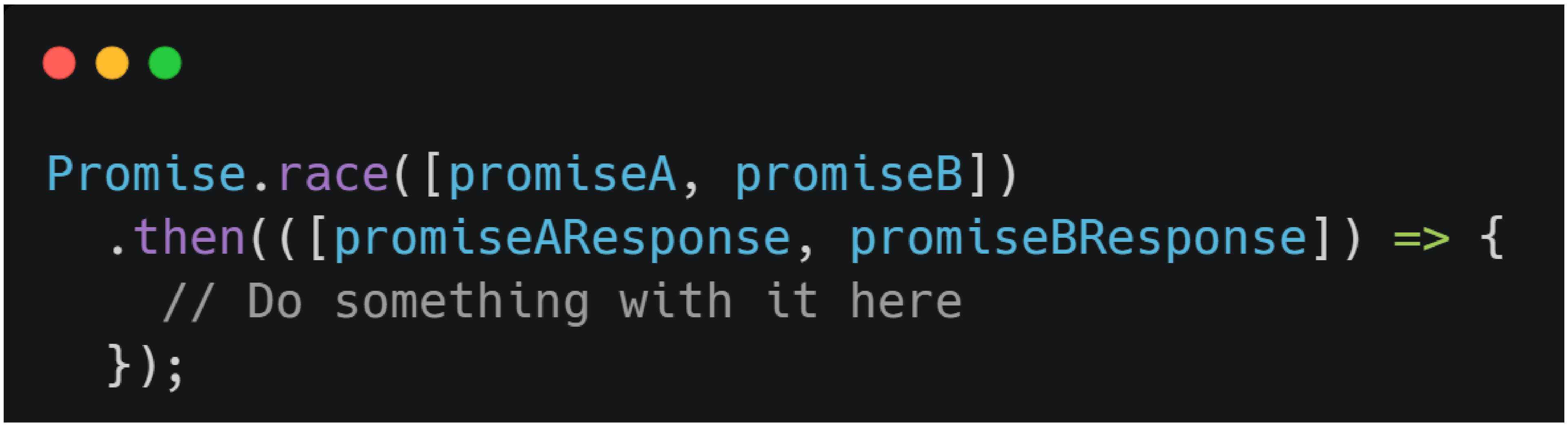
Combining Promises

Promise.all : is fulfilled if all promises that are provided to it are fulfilled or if one of the promises is rejected.



```
Promise.all([promiseA, promiseB])
  .then(([promiseAResponse, promiseBResponse]) => {
            // Do something with it here
  });
```

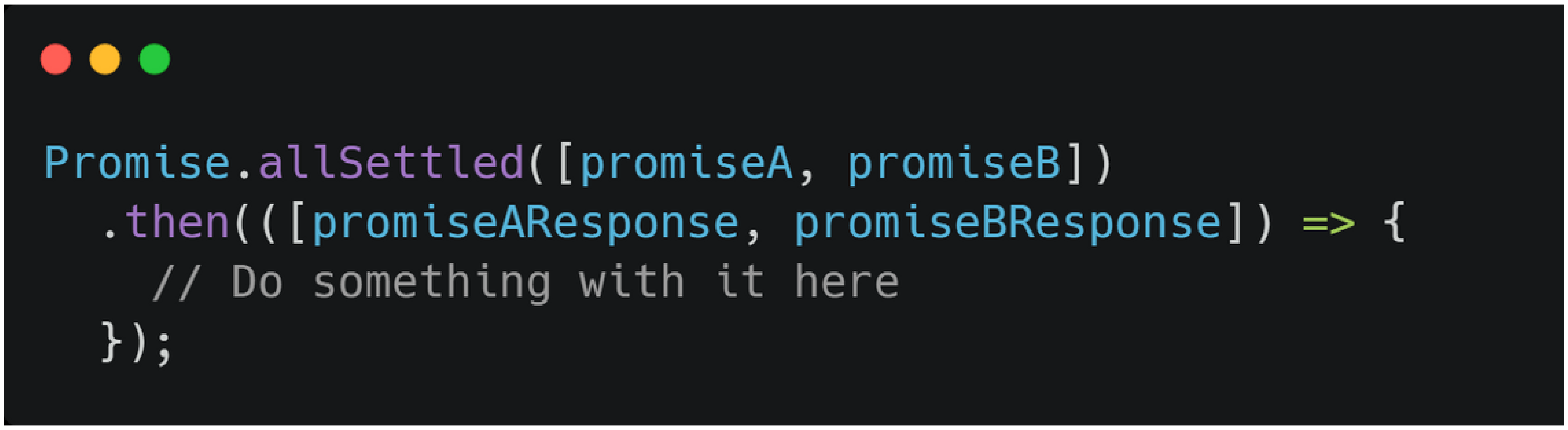
Promise.race: is fulfilled when one of the promises is settled (fulfilled or rejected)



```
Promise.race([promiseA, promiseB])
  .then(([promiseAResponse, promiseBResponse]) => {
    // Do something with it here
  });
```

Combining Promises

Promise.allSettled : Returns when all promises are settled (fulfilled or rejected).



```
Promise.allSettled([promiseA, promiseB])
  .then(([promiseAResponse, promiseBResponse]) => {
    // Do something with it here
  });
```

Follow me for more

aishwarya-dhuri

coder_aishya

Like this post!