JS

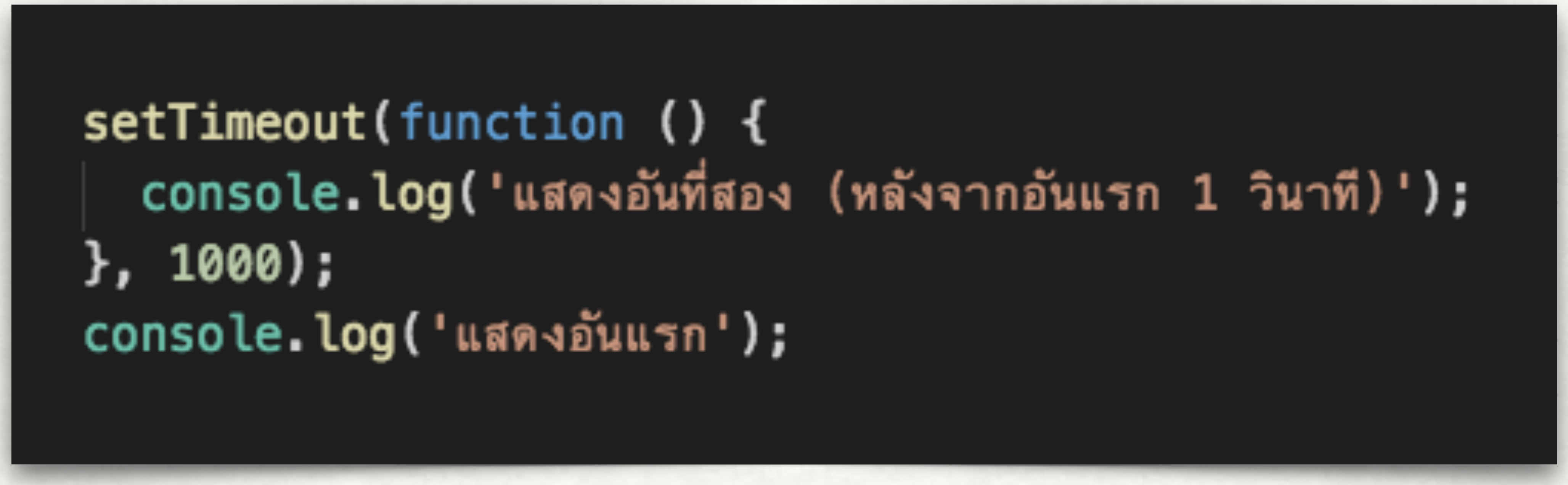
LEARNING DAY

How to use Async/Await (ES7)

มีใครเคยใช้ async/await ?

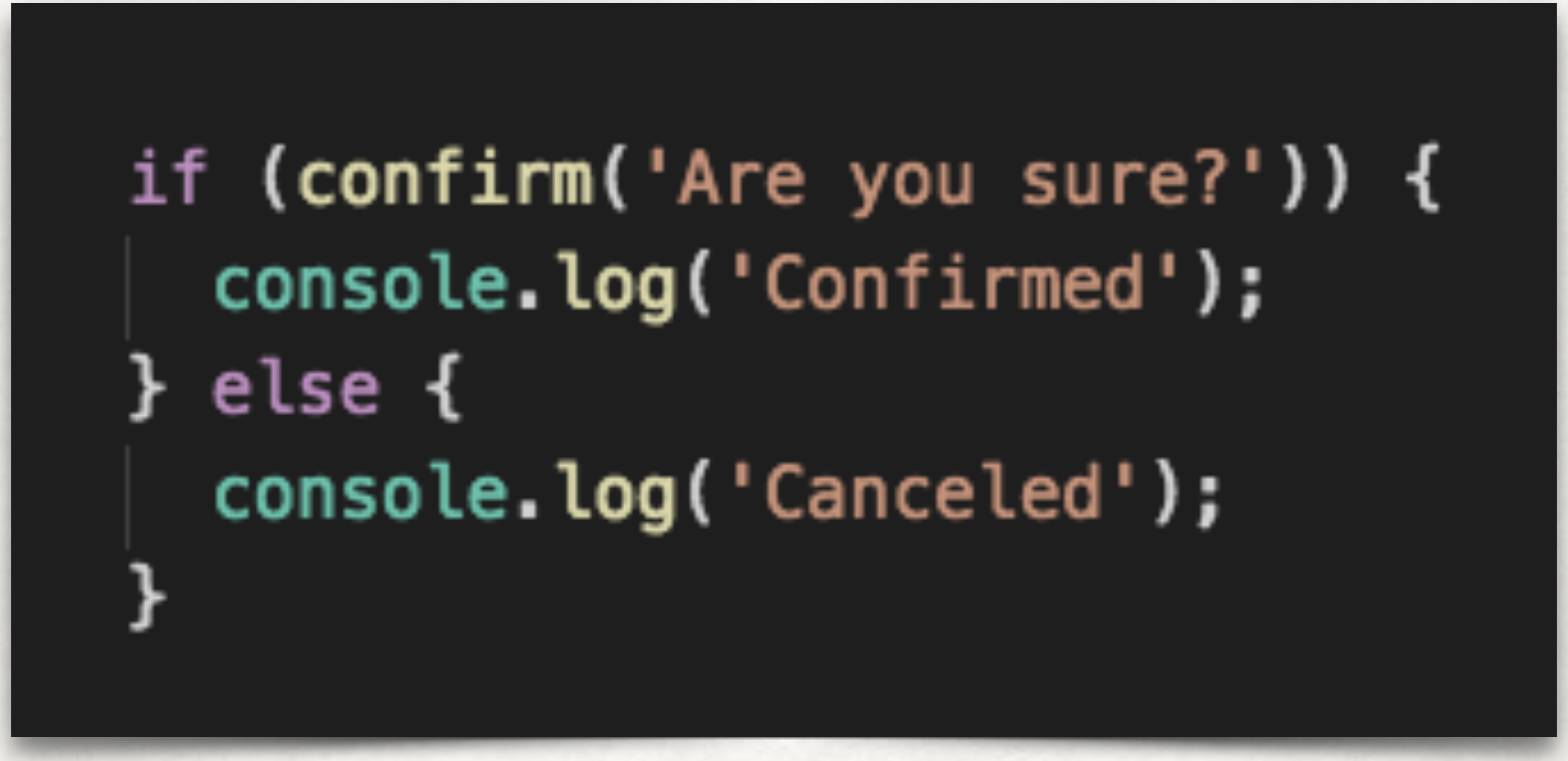
Asynchronous กับ Synchronous คืออะไร ?

ตัวอย่างคำสั่ง Asynchronous ใน JavaScript



```
setTimeout(function () {
  console.log('แสดงอันที่สอง (หลังจากอันแรก 1 วินาที)');
}, 1000);
console.log('แสดงอันแรก');
```

ตัวอย่างคำสั่ง Synchronous



```
if (confirm('Are you sure?')) {
  console.log('Confirmed');
} else {
  console.log('Canceled');
}
```

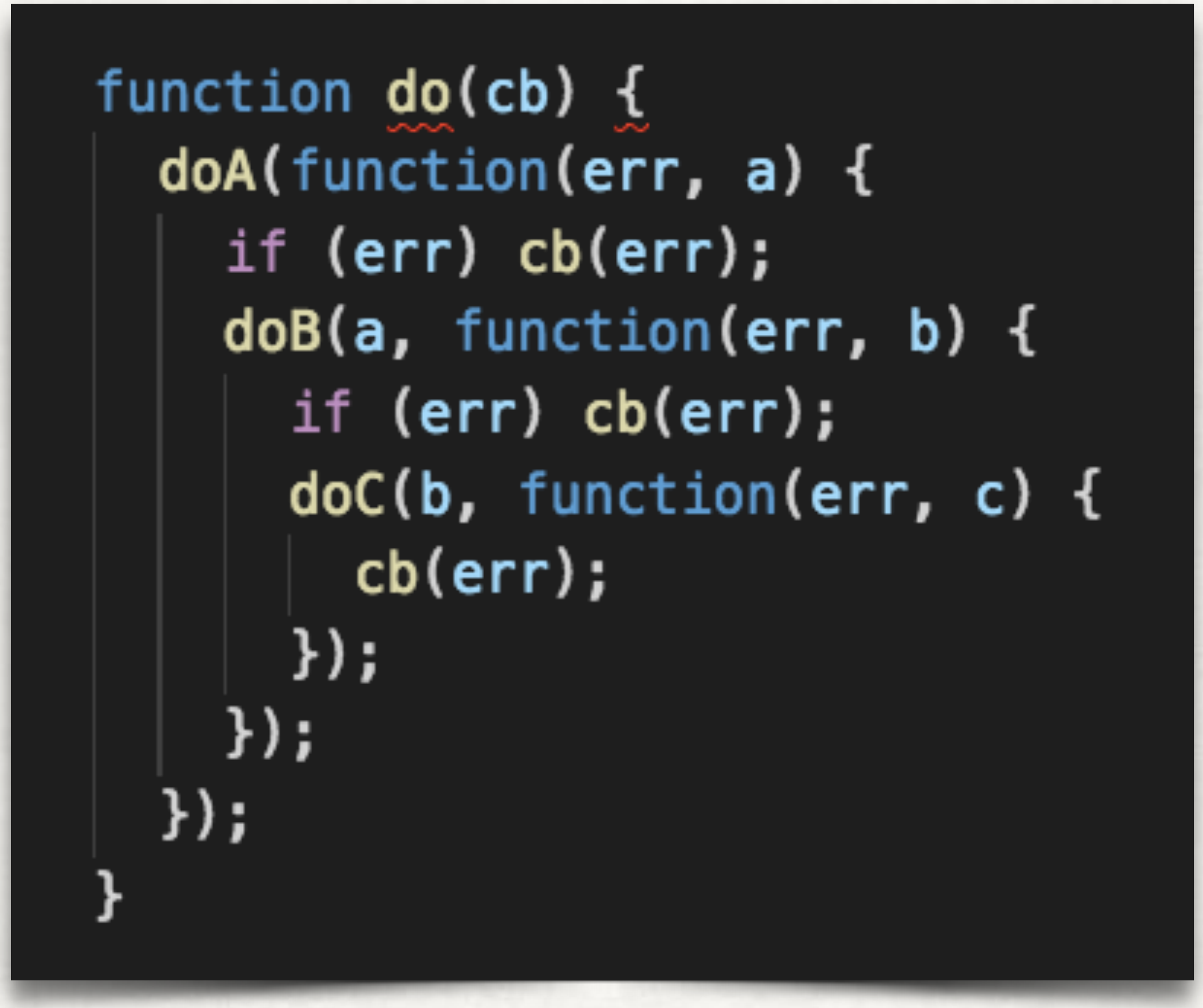
Callback

PROMISE

async / await

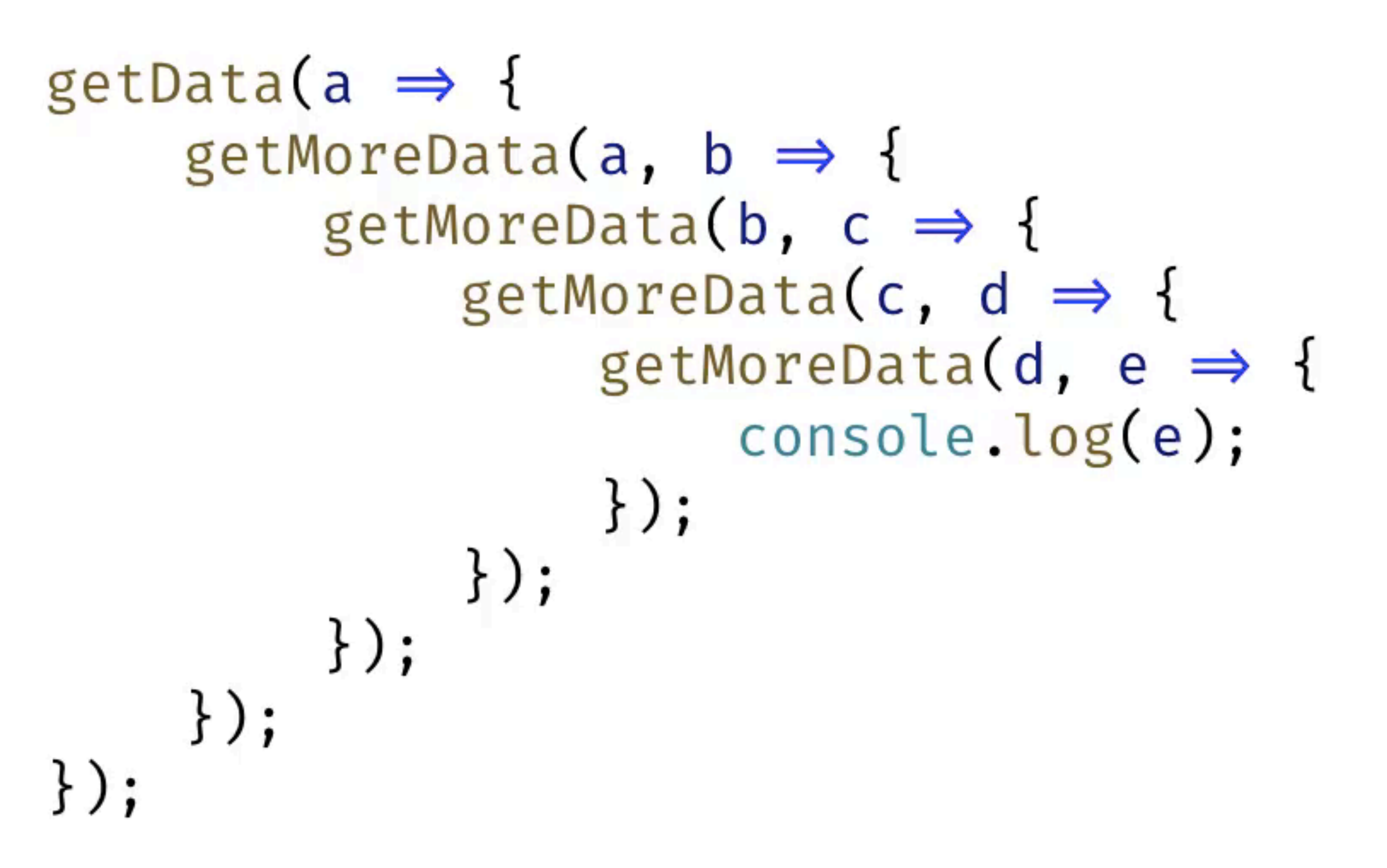
ยุคมืด: Callback

ตัวอย่างการใช้งาน Callback



```
function do(cb) {
  doA(function(err, a) {
    if (err) cb(err);
    doB(a, function(err, b) {
      if (err) cb(err);
      doC(b, function(err, c) {
        cb(err);
      });
    });
  });
}
```

Callback hell

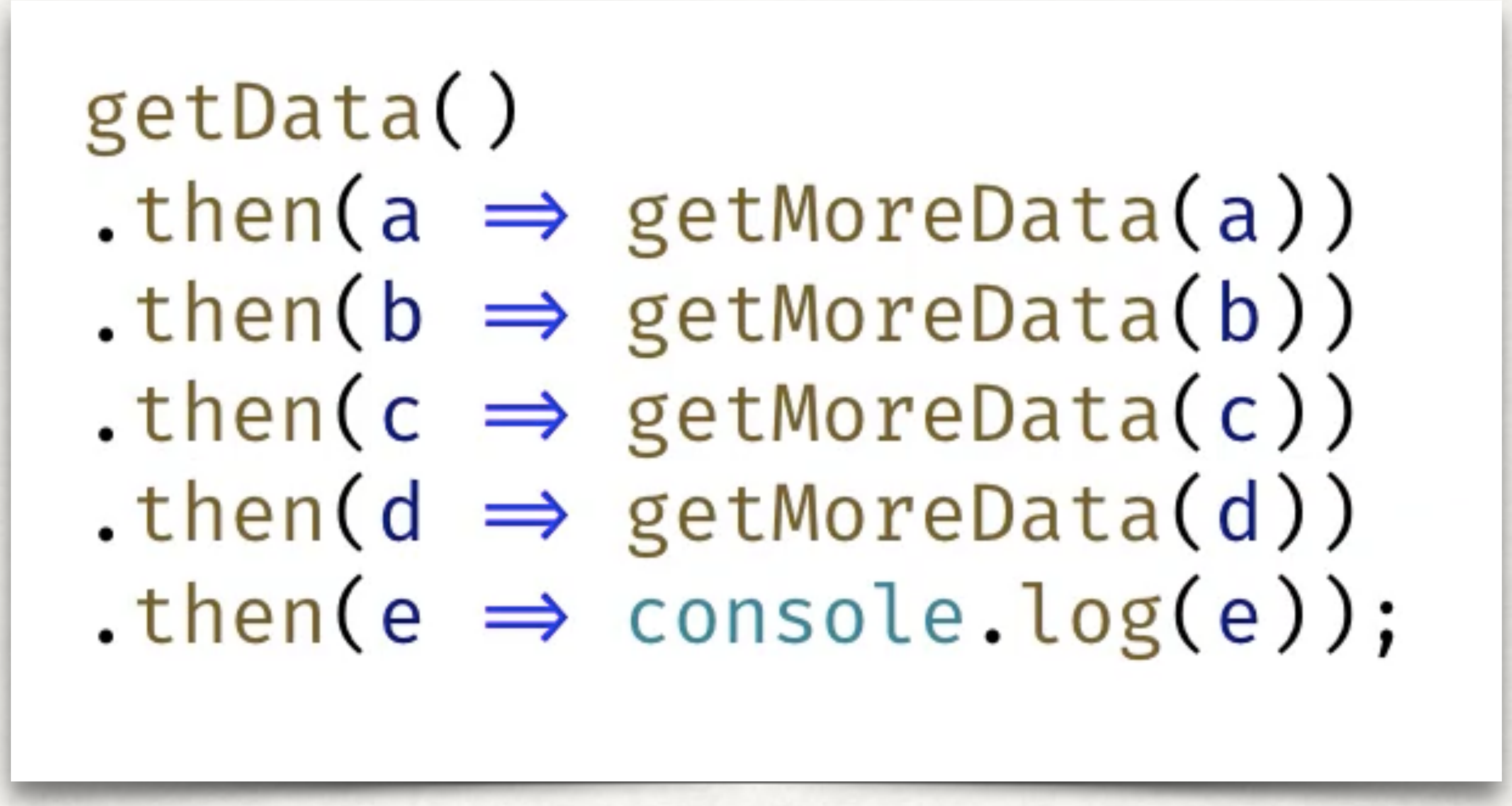


```
getData(a => {
    getMoreData(a, b => {
        getMoreData(b, c => {
            getMoreData(c, d => {
                getMoreData(d, e => {
                    console.log(e);
                });
            });
        });
    });
});
```

ยุคกลาง: Promise

ตัวอย่างการใช้งาน Promise

(pending, resolved, rejected)



```
getData()
.then(a ⇒ getMoreData(a))
.then(b ⇒ getMoreData(b))
.then(c ⇒ getMoreData(c))
.then(d ⇒ getMoreData(d))
.then(e ⇒ console.log(e));
```

ยุคใหม่: async/await

Async/await 101

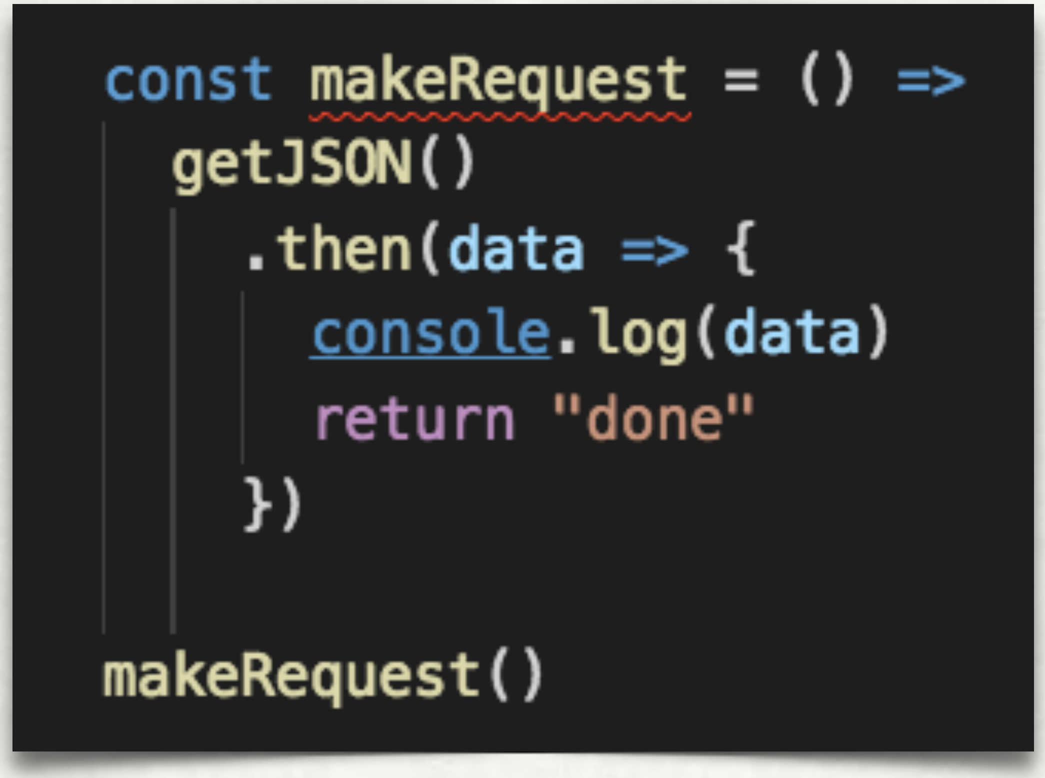
- Async/await is a new way to write asynchronous code. Previous options for asynchronous code are callbacks and promises.

- **Async/await is actually built on top of promises.** It cannot be used with plain callbacks or node callbacks.

- Async/await is, like promises, non blocking.

- Async/await makes asynchronous code look and behave a little more like synchronous code. This is where all its power lies.

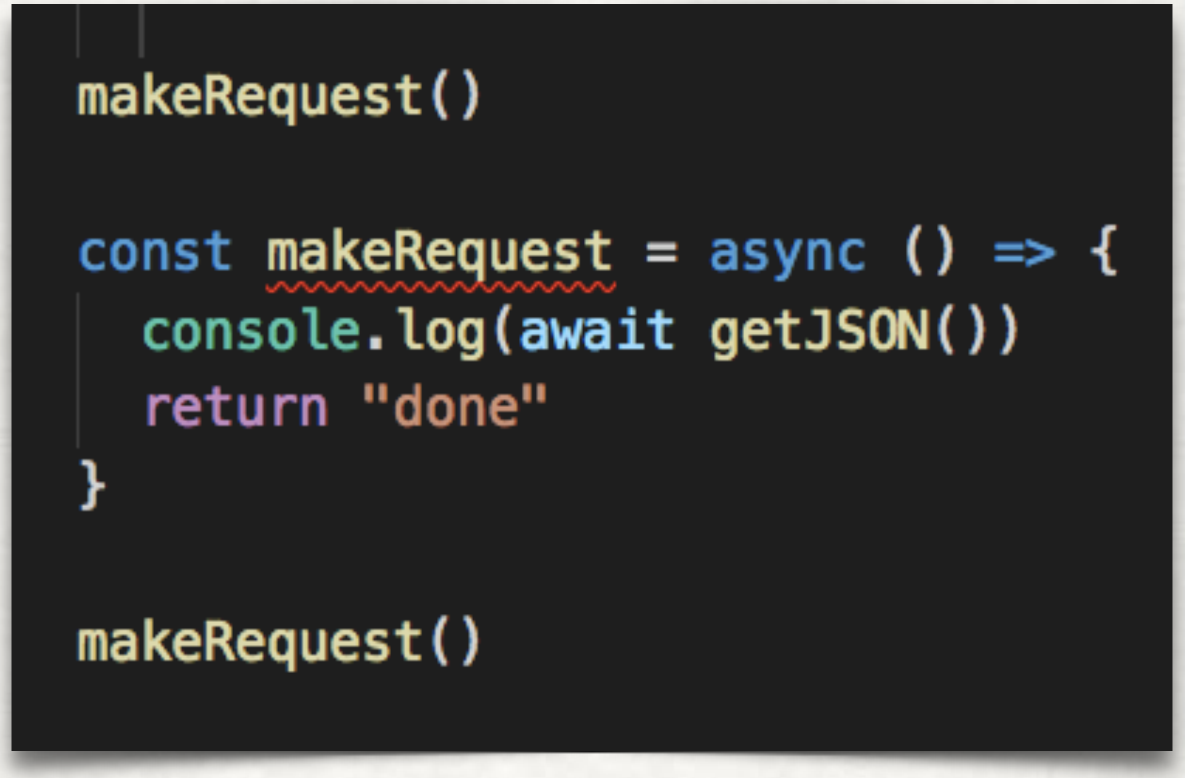
ตัวอย่างการใช้งาน Async/await



```
const makeRequest = () =>
  getJSON()
    .then(data => {
      console.log(data)
      return "done"
    })

makeRequest()
```

Promise



```
makeRequest()

const makeRequest = async () => {
  console.log(await getJSON())
  return "done"
}

makeRequest()
```

Async/await

Workshop Time!!

https://gitlab.thinknet.co.th/nattawut/workshop-js-async-await.git

1. Convert to async/await

2. Handle Errors

3. Await Multiple Promise Sequentially or **Concurrently**

4. Await Multiple Promises Concurrently with **Promise.all()**

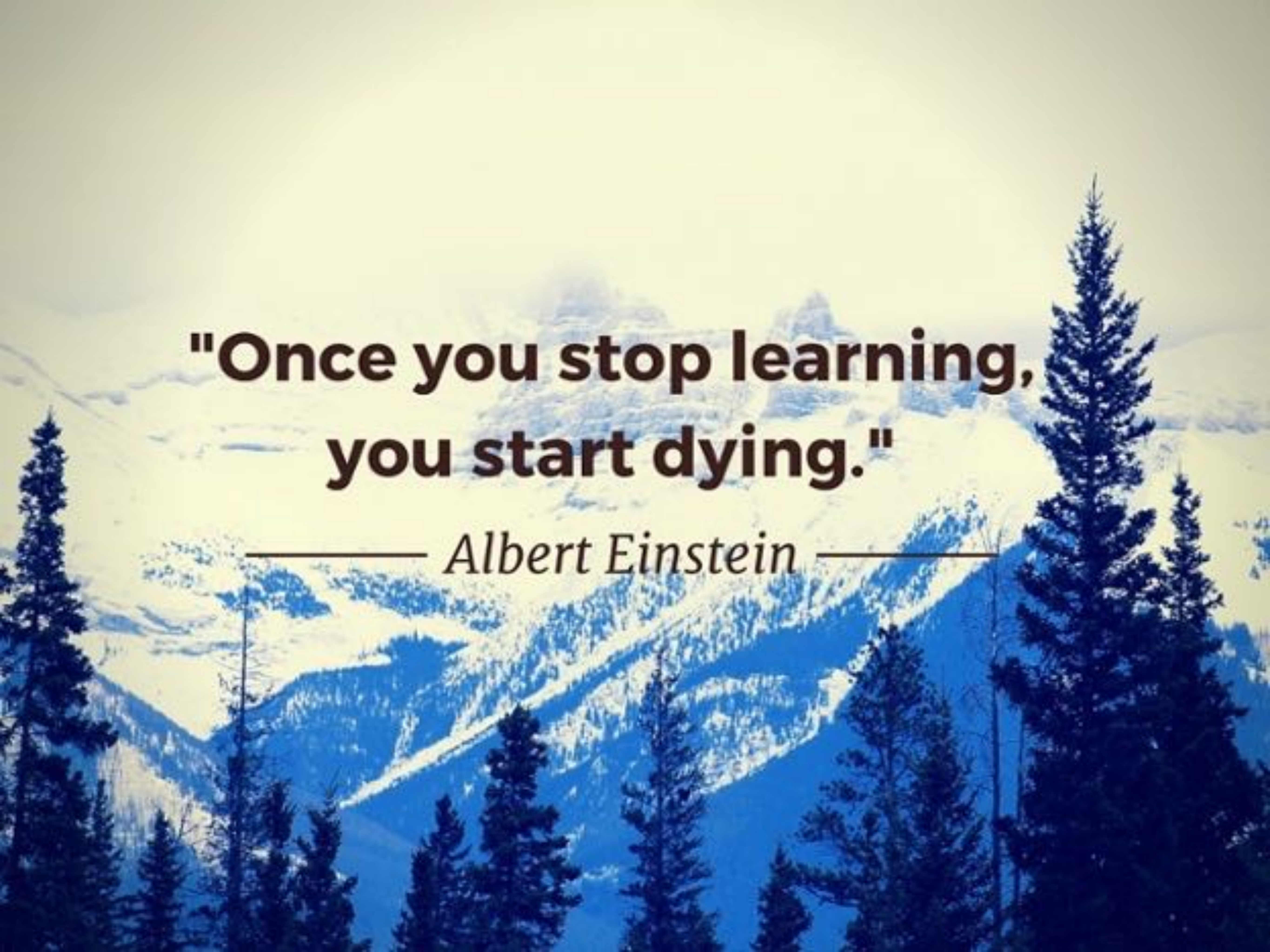
Conclusion

```
getData(a => {
    getMoreData(a, b => {
        getMoreData(b, c => {
            getMoreData(c, d => {
                getMoreData(d, e => {
                    console.log(e);
                });
            });
        });
    });
});
```

Concern

- **It makes asynchronous code less obvious**: Our eyes learned to spot asynchronous code whenever we see a callback or a .then, it will take a few weeks for your eyes to adjust to the new signs, but C# had this feature for years and people who are familiar with it know it’s worth this minor, temporary inconvenience.

- **Node 7 is not an LTS release**: Yes, but node 8 is coming next month, and migrating you codebase to the new version will most likely take little to no effort.



"Once you stop learning, you start dying."

——— *Albert Einstein* ———

Reference

https://blog.panjmp.com/async-await-เรามารู้จัก-syntax-ที่จะมาเปลี่ยนโลกของ-javascript-กัน-3f02091eca05

https://hackernoon.com/6-reasons-why-javascripts-async-await-blows-promises-away-tutorial-c7ec10518dd9