# Promise và async/await

Async/await : Được giới thiệu trong ES8, async/await là một *cơ chế* giúp bạn thực hiện các thao tác bất đồng bộ một cách *tuần tự* hơn. Async/await vẫn sử dụng Promise ở bên dưới nhưng mã nguồn của bạn (theo một cách nào đó) sẽ trong sáng và dễ theo dõi.

Để sử dụng, bạn phải khai báo hàm với từ khóa async. Khi đó bên trong hàm bạn có thể dùng await.

Luôn đua vào then() 1 hàm :

+ phương thưc then đòi hỏi tham số của nó phải là 1 hàm .

+ nếu đưa vào then 1 giá trị nó sẽ bị bỏ qua .

Promise :

1. Promise là một *cơ chế* trong JavaScript giúp bạn thực thi các tác vụ bất đồng bộ mà không rơi vào *callback hell* hay *pyramid of doom*, là tình trạng các hàm callback lồng vào nhau ở quá nhiều tầng. Các tác vụ bất đồng bộ có thể là gửi AJAX request, gọi hàm bên trong setTimeout, setInterval hoặc requestAnimationFrame, hay thao tác với WebSocket hoặc Worker... Dưới đây là một callback hell điển hình

1. Là một *cơ chế* trong JavaScript giúp bạn thực thi các tác vụ bất đồng bộ mà không rơi vào *callback hell* hay *pyramid of doom*, là tình trạng các hàm callback lồng vào nhau ở quá nhiều tầng
2. Các tác vuj bất đồng bộ có thể là gửi AJAX request , gọi hàm bên trong Settimeout , SetInterval hoặc RequestAnimationFrame , hay thao tác với WebSocket hoặc Worker...
3. Promise có 2 tham số :

- resolve : là hàm sẽ được gọi khi promise hoàn thành

-reject là hàm sẽ được gọi khi lỗi xảy ra