* Async, Await, và Promise
* Promise (ES6) là một cơ chế xử lý bất đồng bộ của Javascript được đưa vào để thay thế callback (được xây dựng trên callback) , là một lời hứa trả về kết quả (resolve là hàm được gọi khi promise hoàn thành, reject là hàm được gọi khi xảy ra lỗi)
* Async , Await là một cơ chế xử lý bất đồng bộ mới hơn Promise (ES7) được xây lên từ nền là Promise .
* Async , Await hay đi chung với nhau, nếu như có Await thì thì chương trình chờ đến khi Promise được xử lý và trả về giá trị, nếu không có Await có nghĩa là việc thực khi sẽ không bị tạm dừng để chờ Promise được xử lý và trả về giá trị
* Callback đúng như tên gọi của nó là một hàm sẽ được thực hiện sau khi một hàm khác được thực hiện , có nghĩa là gọi lại hàm đã thực thi.
* Async, Await, Promise, CallBack dùng để kiểm soát luồng chạy
* Sự khác nhau của Sync và Async:

|  |  |
| --- | --- |
| **Sync** | **Async** |
| Chạy tuần tự từng bước một(đồng bộ), bước sau phải chờ bước trước | Chạy đồng loạt, bước trên không cần chờ bước dưới (bất đồng bộ), không cần phải chờ |
| Chạy đồng bộ nên sẽ không bị các lỗi tiến trình , nếu có lỗi xảy ra thì sẽ dễ nhận biết lỗi ở đâu | Chạy bất đồng bộ nên khi có lỗi xảy ra sẽ khó nhận biết lỗi ở đâu hơn là sync |
| Chạy đồng bộ nên sẽ sinh ra trạng thái chờ và không cần thiết ở một số trường hợp => có thể gây chậm hệ thống , tràn bộ nhớ do phải lưu trữ các trạng thái chờ không mong muốn | Chạy bất đồng bộ nên có thể xử lý nhiều việc cùng một lúc không cần phải chờ => tăng số lượng việc xử lý trong cùng một lúc , nhưng dễ dàng xảy ra lỗi tiến trình nếu không kiểm soát tốt |