

Iterator và Observer Pattern (Observable)

- Có những Pattern chúng ta ngầm sử dụng hằng ngày trong Javascript
- Tìm hiểu, đào sâu, mô hình hóa một Design Pattern mới khiến chúng ta thay đổi hoàn toàn tư duy về **Asynchronous Programming**.
- Trong thế giới lập trình Web, ngoài việc sử dụng Promise và Generator, chúng ta còn tương tác với Events (DOM) rất nhiều.

```
function next(evt) {  
  console.log('evt = ', evt);  
}  
  
document.addEventListener('mousemove', next)
```

- **Iterators** và **Events** có mối liên hệ gì với nhau?

```
const arr = [10, 20, 30]; // iterable object  
const it = arr[Symbol.iterator](); // iterator object  
  
console.log(it.next());  
// > { value: 10, done: false }  
console.log(it.next());  
// > { value: 20, done: false }  
console.log(it.next());  
// > { value: 30, done: false }  
console.log(it.next());  
// > { value: undefined, done: true }
```

- Iterators bản chất cũng tuân thủ theo mô hình Producer-Consumer
 - Producer là nơi sản sinh ra dữ liệu
 - Consumer **chủ động** gửi yêu cầu lấy dữ liệu từ Producer sau mỗi lần gọi hàm `next()`
 - Producer ~ Iterable
 - Consumer ~ Iterator
 - Consumer lấy dữ liệu liên tục cho tới khi
 - Producer nói rằng hết dữ liệu để lấy rồi (`done = true`)
 - Producer nói rằng có lỗi (`error`) xảy ra

```
function next(evt) {  
  console.log('evt = ', evt);  
}  
  
// Subscribe to get data from Producer  
document.addEventListener('mousemove', next)
```

```
// Unsubscribe  
// document.removeEventListener('mousemove', next)
```

- Events Handling cũng có hơi hướng của mô hình Producer-Consumer:
 - Producer là nơi sản sinh ra dữ liệu
 - Consumer đăng ký (**subscribe**) lấy dữ liệu từ Producer
 - Consumer không chủ động gọi hàm **next()**
 - Mỗi khi có tín hiệu dữ liệu được gửi về. Producer sẽ kích hoạt hàm **next()** để gửi dữ liệu cho Consumer đã đăng ký
 - Consumer có quyền **hủy đăng ký** nếu không muốn lấy dữ liệu nữa
-

- Giống nhau:
 - Đều áp dụng theo mô hình truyền tải dữ liệu tới Consumer
- Khác nhau:
 - Ai là người quản lý việc gửi dữ liệu?
 - Iterators: Chúng ta **kiểm soát** việc lấy dữ liệu.
 - Events: Chúng ta **không kiểm soát** việc lấy dữ liệu.