

# Observer pattern

---

**Observer pattern** là một design pattern nơi mà một object (thường được gọi là subject) có nhiệm vụ duy trì một danh sách các object dựa vào nó, tự động thông báo đến các object đó nếu có bất kì sự thay đổi nào về trạng thái.

Để dễ hình dung, Observer pattern nó giống như kiểu lắng nghe sự kiện bằng `addEventListener` khi ta DOM bằng Javascript ý. Ta thêm lắng nghe `"scroll"` thì khi scroll nó sẽ chạy function mà ta đưa vào.

```
document.body.addEventListener('scroll', (event) => {  
  console.log(event)  
})
```

Bây giờ mình sẽ mô phỏng một observer pattern nhé.

```
class Subject {  
  constructor() {  
    this.observers = []  
  }  
  // Dùng để đăng ký  
  subscribe(func) {  
    this.observers.push(func)  
  }  
  // Dùng để hủy đăng ký  
  unsubscribe(func) {  
    this.observers = this.observers.filter((subscriber) => subscriber !== func)  
  }  
  // Gửi 1 thông báo đến mọi  
  fire(data) {  
    this.observers.forEach((observer) => observer(data))  
  }  
}
```

Bạn đang thắc mắc cách sử dụng như thế nào phải không, xem tiếp nhé.

```
const $gun = new Subject()  
const func = (value) => {  
  console.log(value)  
}  
$gun.subscribe(func)  
$gun.fire('boom')
```

Mình đăng ký lắng nghe sự kiện, khi mà chạy phương thức `fire` thì những object đã đăng ký sẽ nhận được thông báo ngay lập tức. Tức là `func` function sẽ được chạy.

Nếu muốn không nhận thông báo nữa chỉ cần **unsubscribe** nó là được.

```
$gun.unsubscribe(func)
```

**Angular** dùng cực nhiều pattern này