

Tại sao chúng ta cần hiểu rõ về Function?

- Xem xét một đoạn **code** sau đây:

```
function binhPhuong100() {  
  return 100 * 100  
}  
  
binhPhuong100(); // 10000
```

- Chúng ta cần tính toán bình phương 10 thì sao?

```
function binhPhuong10() {  
  return 10 * 10  
}  
  
binhPhuong10(); // 100
```

- Chúng ta có thể tổng quát hóa để **Function** có khả năng sử dụng lại cao nhất có thể

```
function binhPhuong(num) {  
  return num * num;  
}
```

- Đối với hàm đơn giản như trên, chúng ta đã sử dụng nguyên lý gì để tổng quát hóa một **Function**?
 - Sử dụng **Parameters** đại diện cho một giá trị sẽ được truyền trong tương lai khi chúng ta thực thi và gọi hàm.
 - Sau đó sẽ cung cấp một giá trị cụ thể (**Arguments**) khi hàm được gọi tại thời điểm **run-time**.
- **Higher order functions (HOF)** cũng tuân theo quy tắc này:
 - Thay vì sử dụng **Parameters** đại diện cho một giá trị thông thường (Như **number**, **string**, ...)
 - Sử dụng **Parameters** để đại diện cho một tác vụ phức tạp hơn (**Function**)
 - Chúng ta chỉ xác định tác vụ này làm việc gì tại thời điểm thực thi hàm **HOF** này mà thôi