



*FastTrack SE™*

# HÀM VÀ CÁCH SỬ DỤNG HÀM



# Mục tiêu

1. Hiểu khái niệm hàm
2. Biết cách ứng dụng hàm trong lập trình

## Nội dung

1. Định nghĩa Hàm
2. Gọi một Hàm
3. Hàm có tham số



## ❖ Khái niệm:

Một hàm là một nhóm code có thể tái sử dụng mà có thể được gọi bất cứ đâu trong chương trình.

## ❖ Ứng dụng:

- Giúp lập trình viên không cần thiết phải viết cùng một code lại nhiều lần
- Cho phép phân chia một chương trình lớn thành các phần nhỏ dễ quản lý

# Gọi một hàm

## ❖ Cú pháp:

```
function function_name (parameter-list) {  
    statements;  
}
```

## ❖ Thực hiện gọi hàm:

- Để gọi một hàm ở đâu đó sau khi trong script, bạn đơn giản sẽ cần viết tên của hàm đó

# Gọi một Hàm

## ❖ Ví dụ:

```
<script>  
function sayHello() {  
    document.write("Hello there!");  
    document.write("<br>");  
}
```

```
// Gọi sayHello  
sayHello();
```

```
// Gọi sayHello  
sayHello();
```

```
</script>
```

# Tham số của Hàm

- ❖ Hàm có thể được truyền vào 1 hoặc nhiều tham số, cách nhau bởi dấu phẩy.
- ❖ Những tham số được truyền này có thể được nắm bắt bên trong hàm đó và bất kỳ thao tác nào có thể được thực hiện trên các tham số đó.

# Tham số của Hàm

## ❖ Ví dụ:

```
<script>
function tongHaiSo(a, b) {
    var tong = a+b;
    document.write("Tong hai so " + a + " + " + b + " = " + tong);
    document.write("<br>");
}

// Tính tổng 1 và 2
tongHaiSo(1, 2);

// Tính tổng 10 và 25
tongHaiSo(10, 25);

</script>
```



## ❖ Hàm không trả về và có trả về giá trị

```
<script>
var a, b;

function tongHaiSo(soA, soB) {
    var tong = soA + soB;
    return tong;
}

// Tính tổng 1 và 2
a = 1, b = 2;
document.write("Tong hai so " + a + " + " + b + " = " + tongHaiSo(a, b));
document.write("<br>")

</script>
```

# Q&A