

[w3schools.com](https://www.w3schools.com)

Chức năng Javascript

4-5 phút

Một chức năng JavaScript là một khối mã được thiết kế để thực hiện một nhiệm vụ cụ thể.

Một hàm JavaScript được thực hiện khi "cái gì đó" gọi nó (gọi nó).

Thí dụ

```
function myFunction(p1, p2) {  
    return p1 * p2;           // The function returns the product of p1  
    and p2  
}
```

[Hãy thử nó Yourself »](#)

Javascript Function Cú pháp

Một hàm JavaScript được định nghĩa với các **chức năng** từ khóa, theo sau là một **tên** , tiếp theo là dấu ngoặc đơn **()** .

tên hàm có thể chứa các chữ cái, chữ số, dấu gạch dưới và dấu hiệu đô la (tương tự quy tắc như biến).

Các dấu ngoặc đơn có thể bao gồm tên tham số cách nhau bởi dấu phẩy:

(*argument1*, *argument2*, ...)

Mã này được thực thi, bởi chức năng, được đặt bên trong dấu ngoặc nhọn: {}

```
function name(parameter1, parameter2, parameter3) {  
    code to be executed  
}
```

Chức năng **các thông số** được liệt kê bên trong dấu ngoặc đơn () trong định nghĩa hàm.

Chức năng **lập luận** là những **giá trị** nhận được bởi hàm khi nó được gọi.

Bên trong hàm, các đối số (tham số) cư xử như các biến địa phương.

A Hàm cũng tương tự như một thủ tục hoặc một chương trình con, trong ngôn ngữ lập trình khác.

chức năng Invocation

Các mã bên trong hàm sẽ thực hiện khi "cái gì đó" **gọi** (cuộc gọi) chức năng:

- Khi một sự kiện xảy ra (khi người dùng nhấp vào một nút)
- Khi nó được gọi (gọi là) từ mã JavaScript
- Tự động (tự gọi)

Bạn sẽ học hỏi được nhiều hơn về chức năng gọi sau này trong hướng dẫn này.

chức năng Return

Khi Javascript đạt đến một **tuyên bố trở lại** , chức năng sẽ ngừng

thực hiện.

Nếu hàm được gọi từ một tuyên bố, JavaScript sẽ "trở lại" để thực thi mã sau khi tuyên bố cách gọi.

Chức năng thường tính toán một **giá trị trả về** . Giá trị trả về là "trả lại" trở lại với "người gọi":

Thí dụ

Tính tích của hai số, và trả kết quả:

```
var x = myFunction(4, 3); // Function is called, return value will  
end up in x
```

```
function myFunction(a, b) {  
    return a * b; // Function returns the product of a and b  
}
```

Kết quả trong x sẽ là:

12

[Hãy thử nó Yourself »](#)

Tại sao chức năng?

Bạn có thể sử dụng lại mã: Xác định mã một lần, và sử dụng nó nhiều lần.

Bạn có thể sử dụng mã nhiều lần cùng với các đối số khác nhau, để tạo ra kết quả khác nhau.

Thí dụ

Chuyển đổi từ Fahrenheit sang Celsius:

```
function toCelsius(fahrenheit) {  
    return (5/9) * (fahrenheit-32);  
}  
document.getElementById("demo").innerHTML = toCelsius(77);
```

[Hãy thử nó Yourself »](#)

Các () Toán tử gọi hàm

Sử dụng ví dụ trên, toCelsius đề cập đến đối tượng chức năng, và toCelsius () đề cập đến kết quả hoạt động.

Truy cập vào một chức năng mà không có () sẽ trả lại định nghĩa hàm thay vì kết quả chức năng:

Thí dụ

```
function toCelsius(fahrenheit) {  
    return (5/9) * (fahrenheit-32);  
}  
document.getElementById("demo").innerHTML = toCelsius;
```

[Hãy thử nó Yourself »](#)

Chức năng sử dụng như giá trị biến

Chức năng có thể được sử dụng theo cách tương tự như bạn sử dụng các biến, trong tất cả các loại công thức, bài tập, và tính toán.

Thí dụ

Thay vì sử dụng một biến để lưu trữ các giá trị trả về của một hàm:

```
var x = toCelsius(77);
```

```
var text = "The temperature is " + x + " Celsius";
```

Bạn có thể sử dụng chức năng trực tiếp, như một giá trị biến:

```
var text = "The temperature is " + toCelsius(77) + " Celsius";
```

[Hãy thử nó Yourself »](#)

Bạn sẽ tìm hiểu nhiều hơn về các chức năng sau này trong hướng dẫn này.

biến địa phương

Biến khai báo trong một hàm JavaScript, trở thành **ĐỊA PHƯƠNG** đến chức năng.

biến cục bộ chỉ có thể được truy cập từ bên trong hàm.

Thí dụ

```
// code here can NOT use carName
```

```
function myFunction() {  
    var carName = "Volvo";  
    // code here CAN use carName  
}
```

```
// code here can NOT use carName
```

[Hãy thử nó Yourself »](#)

Kể từ khi các biến cục bộ chỉ được công nhận trong các chức năng của họ, các biến có cùng tên có thể được sử dụng trong các chức năng khác nhau.

biến cục bộ được tạo ra khi một chức năng khởi động, và xóa khi hàm được hoàn thành.

Tự kiểm tra với bài tập!

[Bài tập 1 »](#) [Bài tập 2»](#) [Bài tập 3 »](#) [Bài tập 4»](#) [Tập thể dục 5 »](#)