

JS basic skills

Hàm - Function

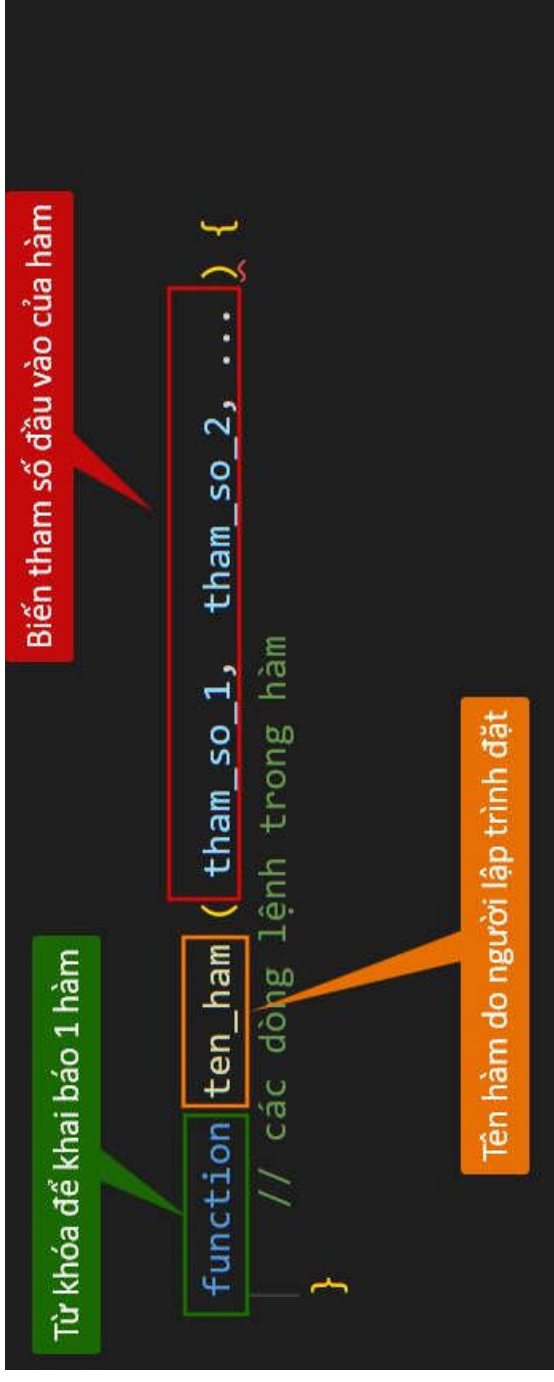
Hàm (function) là một đoạn code được viết sẵn để thực hiện một việc gì đó.

Mục tiêu:

- 01) Chia để trị: chia nhỏ code thành những phần nhỏ hơn, để viết cho dễ
- 02) Dùng lại nhiều lần: cùng 1 đoạn code, không phải viết lại nhiều lần.

HƯỚNG DẪN CƠ BẢN VỀ HÀM (FUNCTION)

Khai báo hàm trong Javascript



1/ function (hàm) là một đoạn code để thực hiện một việc cụ thể nào đó

2/ function chỉ được thực thi khi có "chỗ nào đó" gọi nó (call function)

Từ khóa **function** dùng để khai báo một hàm

Tên hàm: người lập trình tự đặt theo sở thích, ý nghĩa của hàm. Lưu ý không chứa ký tự đặc biệt.

Dấu **()** để chứa các tham số đầu vào. Nếu không có tham số đầu vào thì để trống **()**

Các biến tham số đầu vào: một hàm có thể **không có** hoặc **có nhiều** biến tham số đầu vào

{ }: bắt đầu và kết thúc phạm vi của hàm. Các dòng lệnh bên trong **{ }** sẽ chạy khi hàm được gọi

Khai báo hàm và gọi hàm

Khai báo hàm:

- **Tên hàm:** thể hiện mục đích
- **Đoạn code** sẽ chạy khi thực hiện
- Tham số **hình thức** (tham số đầu vào)
- Giá trị trả về

Khai báo hàm ví dụ 1:

Tên hàm: tinhLapPhuong

Đoạn code sẽ chạy : từ dòng 48 đến dòng 49

Tham số đầu vào: paramSoInput

Giá trị trả về: trả về vKetQua ở dòng 49

```
46 // function tính lập phương của một số
47 function tinhLapPhuong (paramSoInput) {
48     var vKetQua = paramSoInput * paramSoInput * paramSoInput;
49     return vKetQua; // trả lại kết quả
50 }
```

Gọi hàm: gọi bằng Tên hàm

Đưa: tham số thực

Thực hiện đoạn: code của hàm đó,

Nhận: Giá trị trả về như kết quả

Gọi hàm bằng **Tên:** tinhLapPhuong

Đưa tham số thực: 9

Thực hiện đoạn code: của hàm đã định nghĩa (dòng 48-49 ở trên)

Nhận: giá trị trả về của hàm được lưu vào biến vLapPhuong9

```
> var vLapPhuong9 = tinhLapPhuong(9);
   console.log( vLapPhuong9);
729
```

Gọi hàm bằng **Tên:** tinhLapPhuong

Đưa tham số thực: 12

Thực hiện đoạn code: của hàm đã định nghĩa (dòng 48-49 ở trên)

Nhận: giá trị trả về của hàm được lưu vào biến vLapPhuong12

```
> var vLapPhuong12 = tinhLapPhuong(12);
   console.log( vLapPhuong12);
1728
```

Gọi hàm chỗ nào?

1. Gọi từ console
2. Gọi từ Sự kiện của phần tử HTML
3. Gọi từ hàm số khác

Test Form - DevCamp 120

Để tiện cho test các function trên console.log !

Test Button 1

Elements

top

Filter

> var vLapPhuong12 = tinhLapPhuong(12); console.log(vLapPhuong12);

1728

< undefined

>

Console

```
6 <body onload="onPageLoading()">
7 <h1 id="hi_chao_mung"> Test Form - DevCamp 120 </h1>
8 <p id="p_muc_tieu"> Để tiện cho test các function trên console.log !</p>
9 <button onclick="onBtnTestButton1Click()"> Test Button 1</button>
10 </body>
11 <script >
12 "use strict"
13 /*
14 || Vùng 2: event handlers - function xử lý sự kiện
15 */
16 // event handler - function xử lý sự kiện
17 // gọi khi nào: click nút
18 function onBtnTestButton1Click(){
19     console.log("Ấn nút và Test code tại đây..");
20 }
21 // evant handlers - ghi thông tin phần tử H1 ra màn hình
22 // gọi khi nào: gọi khi load/f5
23 function onPageLoading(){
24     console.log("%c Load trang tại đây", "color:blue");
25 }
26
```

```
17 // gọi khi nào: click nút
18 function onBtnTestButton1Click(){
19     console.log("Ấn nút và Test code tại đây..");
20     var vLapPhuong10 = tinhLapPhuong(10);
21     console.log("Lập phương 10 = " + vLapPhuong10);
22 }
23
```

Một số câu hỏi (1/2) (Giảng viên hỏi học viên trả lời)

```
function getPersonInfo() {  
    var vPersonJohn = {  
        fullName: "John Doe",  
        age: 54,  
        email: "john.doe@gmail.com"  
    };  
    console.log("--- Thông tin Person John ---");  
    console.log(vPersonJohn.fullName);  
    console.log(vPersonJohn.email);  
}
```

1/ Ảnh này thực hiện việc gì?

- a) Gọi một function để hiển thị thông tin person
- b) Gọi một function để trả về giá trị
- c) Khai báo một function có 1 tham số là vPersonJohn
- d) Khai báo một function không có tham số đầu vào

2/ **vPersonJohn** ở trong function có ý nghĩa là gì?

- a) Tham số đầu vào của function
- b) Biến cục bộ được khai báo trong function
- c) Một biến toàn cục

Một số câu hỏi (2/2) (Giảng viên hỏi học viên trả lời)

```
/** REGION 1 - Global variables - Vùng khai báo biến, hằng số, tham số TOÀN CỤC */
/** REGION 2 - Vùng gán / thực thi hàm xử lý sự kiện cho các elements */
onPageLoading(); 1
/** REGION 3 - Event handlers - Vùng khai báo các hàm xử lý sự kiện */
function onPageLoading() 2
{
    getPersonInfo(); 3
}
/** REGION 4 - Common functions - Vùng khai báo hàm dùng chung trong toàn bộ chương trình*/
function getPersonInfo() {
    var vPersonJohn = {
        fullName: "John Doe",
        age: 54,
        email: "john.doe@gmail.com"
    };
    console.log("--- Thông tin Person John ---");
    console.log(vPersonJohn.fullName);
    console.log(vPersonJohn.email);
}
```

2/ Kết quả của đoạn code trên

- a) Bị lỗi khi chạy
- b) Có chạy nhưng không hiển thị hay trả lại gì
- c) Hiển thị fullname và email của đối tượng Person John vào Console
- d) Hiển thị toàn bộ thông tin của đối tượng Person John vào Console

1/ Xác định chỗ khai báo hàm và gọi hàm

(các điểm đánh số 1, 2, 3, 4)

Gọi hàm	Các dòng code trong hàm chỉ được thực thi khi hàm được gọi	
	Gọi hàm xong, chương trình chạy tiếp từ sau dòng gọi	link
Tham số	Có thể không có tham số hình thức, hoặc có nhiều tham số hình thức	link
	Tham số có thể là giá trị đơn giản (số, string, đúng/sai) hoặc đối tượng (object, mảng)	link
	Tham số hình thức chỉ có 1 ,tham số thực có thể có nhiều mỗi lần gọi lại có một tham số thực	link
	Với tham số là giá trị đơn giản (số, string, đúng/sai,...), Sửa tham số hình thức, không sửa tham số thực.	link
	Với tham số là Object (ví dụ mảng, hay object tự định nghĩa): Sửa thuộc tính của tham số hình thức, sẽ làm thuộc tính của tham số thực thay đổi.	link
Giá trị Trả lại	Hàm có thể không trả lại giá trị	
	Hàm có thể return ở nhiều chỗ	link
	Sau lệnh return, việc thực hiện hàm sẽ kết thúc ngay.	

PHẦN LUYỆN TẬP KỸ NĂNG VỚI HÀM (FUNCTION)

Task 712.20 Viết hàm kiểm tra chia hết cho 7

Khai báo hàm vào file HTML đi kèm (tham khảo hàm chia hết cho 3)

Sau đó, gọi hàm theo 2 cách:

- Gọi từ Console với nhiều giá trị.
- Làm nút (button) để gọi kiểm tra hàm số này (gọi từ hàm xử lý sự kiện). Viết kết quả gọi ra console.

```
> kiemTraChiaHetCho7( 10 );  
< false  
  
> kiemTraChiaHetCho7 (14);  
< true  
> |
```

Task 712.30 Tham số là mảng

Gọi thử hàm số (đã có sẵn) `tinhTongCacSoTrongArray` từ console.
Viết **nút (button)** để gọi test hàm này từ hàm xử lý sự kiện, viết kết quả ra console.

```
> tinhTongCacSoTrongArray( [10, 4, 8, 5 ] );  
< 27  
  
> tinhTongCacSoTrongArray( [5, 6, 7, 9, 4, 0, 2, -8 ] );  
< 25  
>
```

Task 712.40 Tham số là mảng, tăng giá trị phần tử mảng

Gọi thử hàm số (đã có sẵn) `tangCacSoTrongArray` từ `console`.

Viết **nút (button)** để gọi test hàm này từ hàm xử lý sự kiện, viết kết quả ra `console`.

```
> var vInputArray = [10, 15, 1];  
    tangCacSoTrongArray( vInputArray);  
    console.log(vInputArray);  
▶ (3) [11, 16, 2]
```

Task 712.45 Viết hàm tính ra số tự nhiên bất kỳ từ 1 đến 6

Hãy viết hàm tính ra số tự nhiên bất kỳ từ 1 đến 6

Tên hàm: getRandomNumber1To6

Trả lại: Số tự nhiên bất kỳ từ 1 đến 6

Viết **nút (button)** để gọi test hàm này từ hàm xử lý sự kiện, viết kết quả ra console.

Task 712.48 Viết hàm kiểm tra nhập ô input

Tính bình phương của số nhập vào:

Số hạng A:

...

Tính bình phương

Tính bình phương của số nhập vào
Lưu ý: kiểm tra dữ liệu phải **khác rỗng** và **là số**.
Gợi ý: viết 1 hàm để kiểm tra dữ liệu có phải là số không.

	Yêu cầu subtask (công việc bước nhỏ)	
1	Làm được form đủ 4 phần tử	web
2	Khi load form ghi được ra đủ id, placeholder của ô input (nhập)	console
3	Viết hàm kiểm tra xem dữ liệu truyền vào có ko rỗng và có phải là số không function laSo(paramDoanVanInput) Trả lại đúng/sai (boolean); Gọi kiểm tra hàm này trên console	console
4	Khi ấn tính Bình phương, truy vấn ô nhập liệu, và dùng hàm laSo ở trên để kiểm tra, nếu ko đúng hãy cảnh báo và dừng Nếu đúng là số, hay ghi Bình phương ra console	console

Task 712.49 Viết hàm kiểm tra nhập ô input

Tính tổng hai số sau:

Số hạng A:

...

Số hạng B:

...

Tính tổng

	Yêu cầu subtask (công việc bước nhỏ)	
1	Làm được form đủ 6 phần tử	web
2	Khi load form ghi được ra đủ id, placeholder của hai ô input (nhập)	console
3	Viết hàm kiểm tra xem dữ liệu truyền vào có ko rỗng và có phải là số không function laSo(paramDoanVanInput) Trả lại đúng/sai (boolean); Gọi kiểm tra hàm này trên console	console
4	Khi ấn tính tổng, truy vấn hai ô, và dùng hàm laSo ở trên để kiểm tra, nếu ko đúng hãy cảnh báo và dừng Nếu đúng là số, hay ghi tổng ra console	console

Tính tổng hai số hạng trên và ghi ra console.
Lưu ý: kiểm tra hai dữ liệu phải **khác rỗng** và **là số**.
Gợi ý: viết 1 hàm để kiểm tra dữ liệu có phải là số không.

Task 712.50 Viết hàm đổi màu của Paragraph

Hello Devcamp 120 !

Đổi màu green

Đổi màu red

Đổi màu purple

Làm trang web như trên. Mỗi lần ấn nút hãy đổi màu paragraph “Hello Devcamp 120”.

Gợi ý: Viết một hàm đổi màu của paragraph “Hello Devcamp 120”, tham số hình thức là màu. Gọi hàm này trong các hàm xử lý sự kiện của các nút.

Task 712.55 Viết hàm kiểm tra số nguyên và thử trên console

Test Form - DevCamp 120

Để tiện cho test các function trên console.log !

Test Button 1

Load trang tại đây

> kiemTraSoNguyen("a");
< false
> kiemTraSoNguyen("1.1");
< false
> kiemTraSoNguyen(8);
< true
> kiemTraSoNguyen(8.1);
< false
> |

Viết vào file *html* và thử trên console

Task 712.60 Đọc input ra đối tượng, kiểm tra, và ghi ra console

Dữ liệu học sinh

Họ và Tên

Tuổi

Lớp

Gửi

Làm trang web này.

- Định nghĩa đối tượng gHocSinhObject. Ví dụ có cấu trúc như sau:

```
> var gHocSinhObject = {  
  hoTen: "Trần Thị Hà",  
  tuoi: 10,  
  lop: 5  
};
```

Khi ấn nút Gửi. Bạn hãy.

- Đọc giá trị các trường vào Đối tượng HocSinh
- Kiểm tra các giá trị
- Báo lỗi hoặc Ghi Đối tượng ra console

Yêu cầu: **viết thành 03 function riêng biệt**. Gồm có: Đọc, Kiểm tra, Xử lý hiện thị
Tham số là Đối tượng học sinh.

Task 712.65 Liệt kê những phần tử là số nguyên có trong mảng

Cho mảng `var vNumbers = [10, 15, 4.5, 7, 9, 2.1, 1.11, 8]`;

Yêu cầu liệt kê những phần tử là số nguyên.

Gợi ý: Viết một **hàm** kiểm tra phần tử có phải là số nguyên hay không. Dùng vòng `for` và kiểm tra từng phần tử.

Viết vào file *html* và thử trên *console*

PHỤ LỤC 1: LƯU Ý VỀ HÀM SỐ VỚI LTV MỚI

Định nghĩa	Trong thời gian đầu, bạn không được định nghĩa hàm này bên trong hàm khác.	
Gọi hàm	Đoạn code của hàm, thực hiện khi gọi hàm . Bản thân việc định nghĩa hàm, không thực hiện được dòng code nào cả.	
Tham số	Gọi hàm xong, chương trình chạy tiếp từ sau dòng gọi .	
	Có thể không có tham số hình thức, hoặc có nhiều tham số hình thức	
	Tham số có thể là giá trị đơn giản (số, string, đúng/sai) hoặc đối tượng (object, mảng)	
	Tham số hình thức chỉ có 1 ,tham số thực có thể có nhiều mỗi lần gọi lại có một tham số thực	
	Với tham số là giá trị đơn giản (số, string, đúng/sai,...), Sửa tham số hình thức, không sửa tham số thực.	link
Giá trị Trả lại	Với tham số là Object (ví dụ mảng, hay object tự định nghĩa): Sửa thuộc tính của tham số hình thức, sẽ làm thuộc tính của tham số thực thay đổi.	link
	Có thể không trả lại giá trị	
	Có thể return ở nhiều chỗ	link
	Sau lệnh return, việc thực hiện hàm số kết thúc ngay.	

Tham số là object (cụ thể ở đây là mảng): **sửa** thuộc tham số hình thức, **tác động** vào thuộc tính tham số thực

```
65 //function tính tổng các số trong array
66 function tangCacSoTrongArray(paramArrayInput) {
67     for (var bIndex = 0; bIndex < paramArrayInput.length; bIndex ++ ) {
68         paramArrayInput[bIndex] ++;
69     }
70 }
```

```
> var vInputArray = [10, 15, 1];
    tangCacSoTrongArray( vInputArray);
    console.log(vInputArray);

▶ (3) [11, 16, 2]
```

```
> function nangTuoHocSinh (paramHocSinhObject){
    paramHocSinhObject.tuoi += 1;
}
var vHocSinhPhuong = { hoTen: "Nguyễn Phương", tuoi: 12, lop: 6 };
console.log (vHocSinhPhuong);
nangTuoHocSinh( vHocSinhPhuong);
console.log (vHocSinhPhuong);

▶ {hoTen: 'Nguyễn Phương', tuoi: 12, lop: 6}
▶ {hoTen: 'Nguyễn Phương', tuoi: 13, lop: 6}
```

< undefined

> |

Tham số là giá trị đơn giản (số , đoạn văn, đúng/sai...) sửa tham số hình thức **không** tác động vào tham số thực.

```
> function testTangThamSoHinhThuc10DonVi(paramN) {  
  paramN = paramN + 10;  
}  
var vSoA = 7;  
testTangThamSoHinhThuc10DonVi(vSoA);  
console.log(vSoA);  
7
```

Return có thể có nhiều chỗ

```
72 // function kiểm tra 1 số có phải là số nguyên không
73 function kiemTraSoNguyen( paramN ) {
74     if ( isNaN (paramN)) {
75         return false;
76     }
77     var vNumber = Number(paramN);
78     if ( Number.isInteger(vNumber) ) {
79         return true;
80     }
81     else {
82         return false;
83     }
84 }
```

Định nghĩa hàm (function) & gọi hàm

Ví dụ: Cần xếp loại nhiều học sinh, ta viết hàm xepLoaiHocSinh để sử dụng cho từng học sinh.

```
> // định nghĩa function (hàm)
function xepLoaiHocSinh(paramDiemTrungBinh) {
  var vXepLoai = "";
  if ( paramDiemTrungBinh < 5 ) {
    vXepLoai = "Yếu";
  }
  else if ( paramDiemTrungBinh <= 6.5 ) {
    vXepLoai = "Trung Bình";
  }
  else if ( paramDiemTrungBinh < 8 ) {
    vXepLoai = "Khá";
  }
  else {
    vXepLoai = "Giỏi";
  }
  return vXepLoai;
}

// gọi hàm
var vXepLoaiHocSinhPhuong = xepLoaiHocSinh(8.5); //
console.log( "Xếp loại của học sinh Phương: " + vXepLoaiHocSinhPhuong );

Xếp loại của học sinh Phương: Giỏi
```

Định nghĩa hàm

Tên: thể hiện **mục đích**

Tham số hình thức

Đoạn code định sẵn

Trả lại giá trị

Gọi hàm

Tên hàm

Tham số thực tế

Nhận lại giá trị (nếu có)