

Task specification

Task 3B.1, 3B.2, 3B.4, 3B.5, 3B.6, 3B.10

Task = công việc;

Task specification = mô tả công việc (đầu bài)

Task 3B.1 Mảng (Array) và vòng lặp For (1/4)

```
var arr = [value1, value2, ..., valueN];
```

```
var mangSo = [ 2, 5, 6, 9, 12, 19, 34 ];
```

```
var mangChu = [ "xe", "laptop", "đồng hồ", "cây cảnh" ] ;
```

Mảng là một **tập hợp các giá trị**, bắt đầu và kết thúc bằng **[]**

Các giá trị bên trong mảng gọi là: **phần tử**. Một mảng chứa nhiều phần tử

Các phần tử cách nhau bởi **dấu phẩy**

Dùng thuộc tính **length** để biết số lượng phần tử của mảng => `arr.length` `mangSo.length` `mangChu.length`

Task 3B.1 Mảng (Array) và vòng lặp For (2/4)

```
var mangSo = [ 2, 5, 6, 9, 12, 19, 34 ]
```

chỉ số	0	1	2	3	4	5	6
	2	5	6	9	12	19	34

```
var mangChu = [ "xe", "laptop", "đồng hồ", "cây cảnh", ]
```

chỉ số	0	1	2	3
	"xe"	"laptop"	"đồng hồ"	"cây cảnh"

mangSo [0] ==> 2

mangSo [3] ==> 9

mangSo [5] ==> 19

mangChu [0] ==> "xe"

mangChu [1] ==> "laptop"

mangChu [3] ==> "cây cảnh"

Sử dụng **chỉ số** (index) để **truy xuất** một phần tử của mảng: **arr[index]**

Chỉ số của mảng luôn bắt đầu từ 0 (index = 0)

Task 3B.1 Mảng (Array) và vòng lặp For (3/4)

```
for ( var chiSo = 0; chiSo < arr.length; chiSo ++ ) {  
    // việc cần lặp lại  
}
```

Vòng lặp **for** dùng để thực thi một hành động lặp đi lặp lại nhiều lần

Vòng lặp for thường được dùng để duyệt qua và thực hiện việc gì đó với từng phần tử của mảng

```
var mangSo = [ 2, 5, 6, 9, 12, 19, 34 ];
```

Vấn lặp tiếp khi chưa đến phần tử cuối cùng

```
for ( var index = 0; index < mangSo.length; index ++ ) {  
    // việc cần lặp lại: ghi ra console từng phần tử của mảng  
    console.log( mangSo[index] );  
}
```

bắt đầu từ phần tử đầu tiên (chỉ số = 0)

Sau mỗi lần lặp thì tăng chỉ số lên 1 để tới phần tử tiếp theo

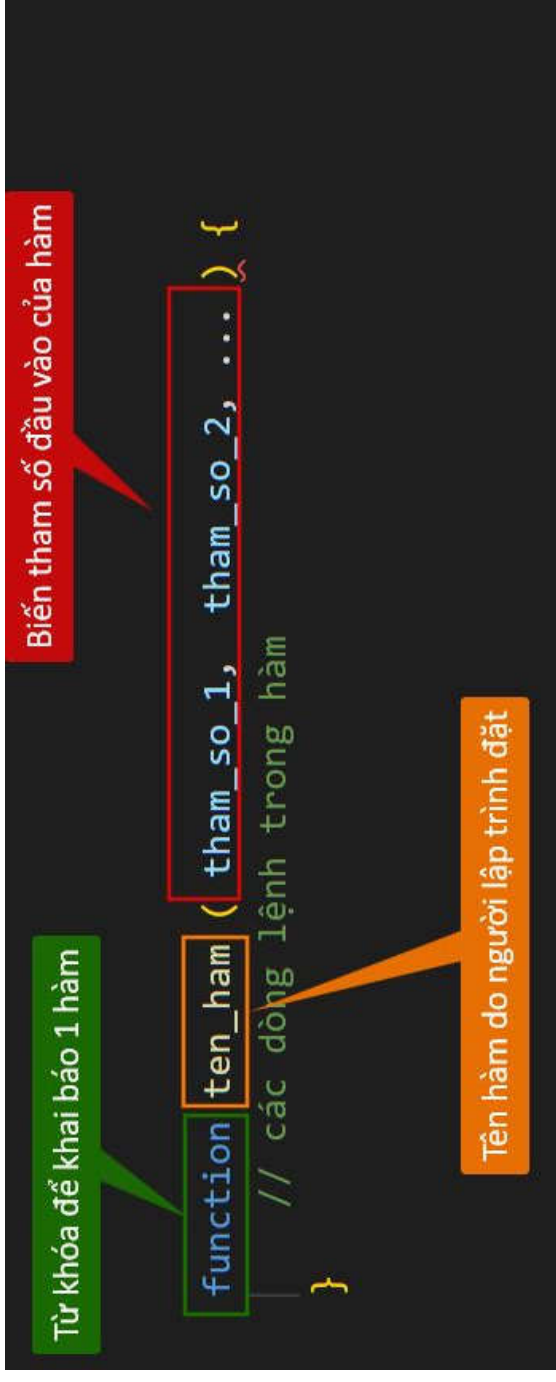
Hãy gõ lại đoạn code để xem kết quả

Task 3B.1 Mảng (Array) và vòng lặp For (4/4)

Tải file Task3b1.html về để làm các yêu cầu bên dưới. Hãy nộp mã nguồn sau khi hoàn thiện

STT	Yêu cầu
1	Khai báo một mảng các số có các phần tử: -1, 10, -10, 4, 8, 2, 9
2	Khai báo một mảng các chuỗi có các phần tử: "Cộng hòa", "Việt Nam", "Xin chào", "Hello"
3	Ghi ra console phần tử thứ 3 và phần tử thứ 7 của mảng (1)
4	Ghi ra console phần tử thứ 1 và phần tử thứ 4 của mảng (2)
5	Dùng vòng lặp for để ghi ra console tất cả các phần tử của mảng (2)
6	Dùng vòng lặp for để cộng thêm 2 cho từng phần tử ở mảng (1), rồi ghi ra console

Task 3B.2 Khai báo hàm trong Javascript (1/4)



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

Task 3B.2 Khai báo hàm và gọi hàm (2/4)

Khai báo hàm:

- **Tên hàm:** thể hiện mục đích
- **Đoạn code** sẽ chạy khi thực hiện
- **Tham số hình thức**
- **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: 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

Task 3B.2 Gọi hàm ở chỗ nào? (3/4)

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



```

1 <body onload="onPageLoading()">
2   <h1 id="h1_chao_mung"> Test Form - DevCamp 120 </h1>
3   <p id="p_muc_tieu"> Để tiện cho test các function trên console.log !</p>
4   <button onclick="onBtnTestButtonClick()"> Test Button 1</button>
5
6 </body>
7
8 <script >
9   "use strict"
10
11   /*
12   || Vùng 2: event handlers - function xử lý sự kiện
13   */
14
15   // event handler - function xử lý sự kiện
16   // gọi khi nào: click nút
17   function onBtnTestButtonClick(){
18     console.log("Ấn nút và Test code tại đây..");
19
20   }
21
22   // event handlers - ghi thông tin phần tử H1 ra màn hình
23   // gọi khi nào: gọi khi load/75.
24   function onPageLoading(){
25     console.log("%c Load trang tại đây", "color:blue");
26
27   }

```

```
// gọi khi nào: click nút
function onBtnTestButton1Click(){
    console.log("Ấn nút và Test code tại đây..");
    var vLapPhuong10 = tinhlapPhuong(10);
    console.log("Lập phương 10 = " + vLapPhuong10);
}
```


Task 3B.2 Bài tập về Hàm trong Javascript (4/4)

Tải file Task3b2.html về để làm các yêu cầu bên dưới. Hãy **nộp mã nguồn** sau khi hoàn thiện

STT	Yêu cầu
1	Bạn hãy gọi hàm kiểm tra ChiaHếtCho3 từ Console với nhiều giá trị khác nhau
2	Viết hàm tính tổng của 2 số cho trước và gọi hàm để dùng thử ở trong file html (ghi ra console)
3	<p>Viết một hàm kiểm tra Tính Tổng với các nội dung:</p> <ul style="list-style-type: none">(1) Khai báo 2 biến số, lần lượt nhận giá trị là 10 và 15(2) Khai báo 1 biến tổng.(3) Gọi hàm tính tổng đã có ở trên, truyền vào 2 biến số đã khai báo ở bước (1)(4) Kết quả của hàm tính tổng được gán cho biến tổng(5) ghi ra console giá trị của biến tổng

Task 3B.4 Mảng (Array) và vòng lặp For

Tạo một file html mới, thêm thẻ `<script></script>` để thực hiện các yêu cầu bên dưới

STT	Yêu cầu cần thực hiện
1	Khai báo một mảng các số có các phần tử: 10 3 5 9 30 8 100 45
2	Khai báo một mảng các chuỗi có các phần tử: “Nha Trang” “Hà Nội” “Vũng Tàu” “Cần Thơ” “Quảng Ninh” “Bắc Giang”
3	Ghi ra console phần tử thứ 1, thứ 4 và thứ 7 của mảng (1)
4	Ghi ra console phần tử thứ 2 và phần tử thứ 3 của mảng (2)
5	Dùng vòng lặp for để cộng thêm 10 cho từng phần tử của mảng (1) rồi ghi ra console
6	Dùng vòng lặp for thêm chữ “Xin chào ” vào trước tất cả các thành phần ở mảng (2), rồi ghi vào Console
7	Dùng code Javascript thêm phần tử “ Phú Yên ” vào trong mảng (2) Hãy search google từ khóa: add element array javascript

Task 3B.5 Đếm số phần tử trong mảng bắt đầu bằng ap

```
var openArray = ["apple", "mango", "banana", "ape", "application", "lime", "soap", "orange", "open"];
```

Hãy viết đoạn code để đếm xem: Có bao nhiêu phần tử trong mảng `openArray` bắt đầu bằng chữ "**ap**"?

Sau đó, ghi số lượng đó ra console.

Ví dụ:


apple => bắt đầu bằng chữ **ap**

application => bắt đầu bằng chữ **ap**

orange => bắt đầu bằng chữ **or**, không phải **ap**

soap => bắt đầu bằng chữ **so**, không phải **ap**

Task 3B.5 Gợi ý các bước làm (tham khảo)

Bước	Gợi ý cách làm
1	Khai báo 1 biến lưu số phần tử bắt đầu bằng ap. Ban đầu cho giá trị biến = 0
2ste 	Search xem có hàm nào xác định 01 string A có bắt đầu bằng một string B không? 1. Gợi ý từ khóa Search google: string start with other string javascript 2. Dùng thử trên console
3	Có một vòng lặp for để duyệt qua các phần tử của mảng. Trong vòng lặp, với từng phần tử, xác định xem phần tử đó có bắt đầu bằng “ap” không? Nếu có, ta phải +1 thêm vào biến đã có ở bước 1
4	Sau khi kết thúc vòng lặp, ghi giá trị của biến ra console

Ste: Search - Trial - Error: tìm kiếm, thử dùng, sai và lặp lại
Nên tìm bằng tiếng Anh, tuy nhiên nếu bạn quá khó sử dụng tiếng anh;
Hãy tìm kiếm bằng tiếng Việt: chuỗi bắt đầu bằng chuỗi khác javascript

Task 3B.6 Tạo file html và gõ lại đoạn code sau

```
1 <!DOCTYPE html>
2 <html>
3 <script>
4   function myWelcomeMsg(strWho) {
5     const messagePrefix = "Welcome ";
6     const messagePostFix = " to DevCamp120 Bootcamp.";
7     return messagePrefix + strWho + messagePostFix;
8   }
9
10  var myWelcomeStr = myWelcomeMsg("Do Nhat Quang");
11  console.log(myWelcomeStr);
12 </script>
13
14 <body>
15   <h3>File chạy thử hàm JS</h3>
16 </body>
17 </html>
```




1. Tạo ra 01 file html mới, sau đó gõ lại nội dung code ở ảnh bên

2. Mở file html vừa lưu lên trình duyệt Brave. Sau đó, mở cửa sổ Console ra để xem kết quả

3. Sửa lại tên 2 biến ở dòng 5 và 6 theo tên mà bạn muốn, có thể dùng tiếng Việt. Chú ý sửa các chỗ khác liên quan. Sau đó chạy lại để kiểm tra kết quả

4. Nộp lại file html vào task

Task 3B.10 Gợi ý các bước làm (tham khảo)

Bước	Gợi ý cách làm
1	Định nghĩa 01 biến lưu chiều dài chuỗi dài nhất, gán giá trị ban đầu bằng 0
2ste <div></div>	Tìm kiếm google xem tính chiều dài của chuỗi ở javascript bằng cách nào? 1. Gợi ý từ khóa Search : string length javascript 2. Dùng thử trên console
3	Có 01 vòng lặp chạy qua tất cả các phần tử của mảng Trong vòng lặp, lần lượt so sánh chiều dài của phần tử hiện tại với biến lưu chiều dài chuỗi dài nhất. Nếu chiều dài của phần tử hiện tại lớn hơn, thì thay giá trị của biến bằng chiều dài của phần tử đó.
4	Sau khi kết thúc vòng lặp, ghi chiều dài của chuỗi dài nhất ra console Chú ý: Đầu ra của bài này phải trả về 1 số , là chiều dài của chuỗi dài nhất

Gợi ý đoạn văn tìm kiếm **Google**: “**vấn đề lĩnh vực**”
Nếu dùng tiếng việt: “**chiều dài chuỗi javascript**”
Nếu dùng tiếng anh: “**string length javascript**”