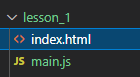
# Bài 1: Giới Thiệu

## 1.Sử dụng js với html

### -Tạo index.html và main.js



### -Code html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <h1>Javascript basic</h1>

    <!-- src muốn hỏi nguồn tài nguyên muốn cung cấp cho thằng script này là gì -->

    <script src="./main.js"></script>

</body>

</html>

## 2.Khái niệm biến và cách sử dụng

### 2.1.Code index.html

    <!DOCTYPE html>

    <html lang="en">

    <head>

        <meta charset="UTF-8">

        <meta http-equiv="X-UA-Compatible" content="IE=edge">

        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <title>Document</title>

    </head>

    <body>

        <h1>Javascript basic</h1>

        <!-- src muốn hỏi nguồn tài nguyên muốn cung cấp cho thằng script này là gì -->

        <script src="./main.js"></script>

    </body>

    </html>

### 2.2.Code main.js

// khai báo biến

// đặt biến tên fullname, sau đó đưa giá trị Tran Ca Phu vào

var fullname = 'Tran Ca Phu'; //khai báo tên

var age = '23'; //khai báo tuoi

// Gọi hàm thông báo

// alert này tại 1 thời điểm nó sẽ hiển thị một cái thôi khi mà tắt 1 cái đầu đi nó sẽ hiển thị 1 cái alert kế tiếp

alert(fullname); //thông báo tên

alert(age); //thông báo tuổi

# Bài 2: Biến, comments, build-in

-Là những hàm được xây dựng sẵn, ta chỉ cần lấy ra rồi dùng thôi.

-Một số hàm Built-in:

## 1. Alert

-Là hàm dùng để thông báo

## 2. Console

-Chứa những hàm bên trong, được gọi là phương thức có tác dụng in ra những dòng thông báo bên tab conslole trên dev tool

### 2.1.Code index.html

    <!DOCTYPE html>

    <html lang="en">

    <head>

        <meta charset="UTF-8">

        <meta http-equiv="X-UA-Compatible" content="IE=edge">

        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <title>Document</title>

    </head>

    <body>

        <h1>Javascript basic</h1>

        <!-- src muốn hỏi nguồn tài nguyên muốn cung cấp cho thằng script này là gì -->

        <script src="./main.js"></script>

    </body>

    </html>

### 2.2.Code main.js

// Ví dụ console

var fullname = 'Tran Ca Phu';

// in ra dòng chữ thông báo trên devtool

// console.log(fullname)

// in ra cảnh báo trên devtool

// console.warn(fullname)

// in ra lỗi trên devtool

console.error(fullname)

## 3.Confirm

### 3.1.Code index.html

    <!DOCTYPE html>

    <html lang="en">

    <head>

        <meta charset="UTF-8">

        <meta http-equiv="X-UA-Compatible" content="IE=edge">

        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <title>Document</title>

    </head>

    <body>

        <h1>Javascript basic</h1>

        <!-- src muốn hỏi nguồn tài nguyên muốn cung cấp cho thằng script này là gì -->

        <script src="./main.js"></script>

    </body>

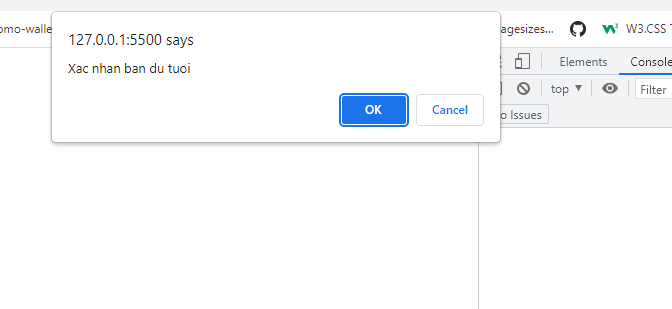
    </html>

### 3.2.Code main.js

// Ví dụ confirm

// confirm tạo ra nút cancel và oke trên màn hình

confirm('Xac nhan ban du tuoi');



## 4.Prompt

### 4.1.Code index.html

    <!DOCTYPE html>

    <html lang="en">

    <head>

        <meta charset="UTF-8">

        <meta http-equiv="X-UA-Compatible" content="IE=edge">

        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <title>Document</title>

    </head>

    <body>

        <h1>Javascript basic</h1>

        <!-- src muốn hỏi nguồn tài nguyên muốn cung cấp cho thằng script này là gì -->

        <script src="./main.js"></script>

    </body>

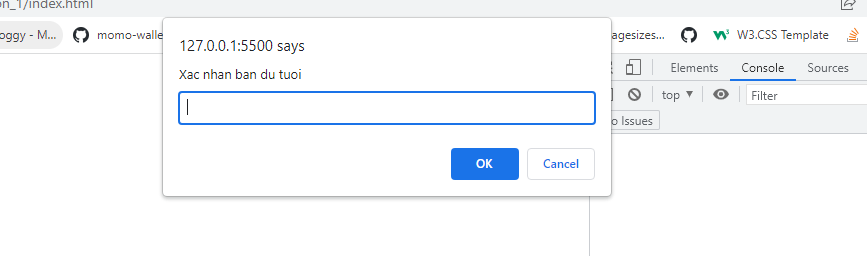
    </html>

### 4.2.Code main.js

// Ví dụ prompt

// sẽ mỡ ra hộp thoại có dòng input để ta nhập vào(còn có 2 nút cancel và ok)

prompt('Xac nhan ban du tuoi');



## 5.Set timeout

### 5.1.Code index.html

    <!DOCTYPE html>

    <html lang="en">

    <head>

        <meta charset="UTF-8">

        <meta http-equiv="X-UA-Compatible" content="IE=edge">

        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <title>Document</title>

    </head>

    <body>

        <h1>Javascript basic</h1>

        <!-- src muốn hỏi nguồn tài nguyên muốn cung cấp cho thằng script này là gì -->

        <script src="./main.js"></script>

    </body>

    </html>

### 5.2.Code main.js

// Ví dụ setTimeout

/\*\*

 \* Đối tượng truyền vào là 1 function

 \* Cho một đoạn code được chạy trong một khoảng thời gian(1000ms = 1s)

 \* Nếu mà cho thời gian lâu thì thời hiên thị sẽ lâu theo

 \*/

setTimeout(function () {

    alert('Thông báo')

}, 1000)

## 6.Set interval

### 6.1.Code index.html

    <!DOCTYPE html>

    <html lang="en">

    <head>

        <meta charset="UTF-8">

        <meta http-equiv="X-UA-Compatible" content="IE=edge">

        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <title>Document</title>

    </head>

    <body>

        <h1>Javascript basic</h1>

        <!-- src muốn hỏi nguồn tài nguyên muốn cung cấp cho thằng script này là gì -->

        <script src="./main.js"></script>

    </body>

    </html>

### 6.2.Code main.js

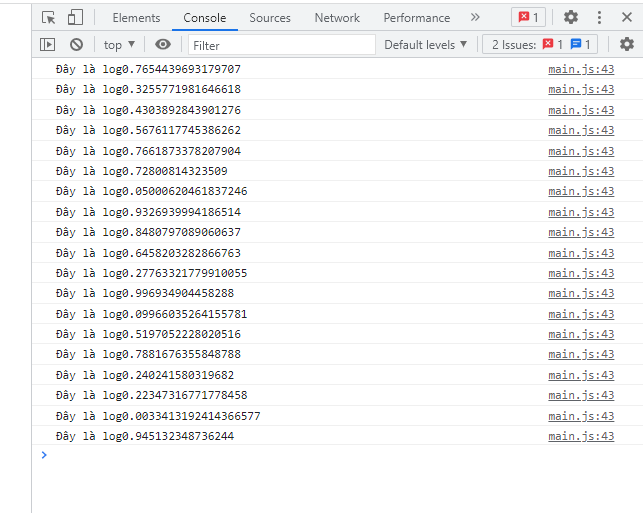
// Ví dụ setInterval

// nó cũng như setTimeout khác ở chỗ nó xuất ra 1 thông báo trong một giây và xuất ra nhiều thông báo

setInterval(function () {

    console.log('Đây là log' + Math.random())

}, 1000)



# Bài 3: Toán Tử, Kiểu Dữ Liệu

## 1.Làm quen với toán tử

### - Toán tử số học

// Toán tử số học(cộng, trừ,, nhân, chia)

var a = 1 + 2;

console.log(a);

### - Toán tử gán

// Toán tử gán(Dấu = phía dưới được gọi là toán tử gán)

var fullname = 'Tran Ca Phu'

### - Toán tử so sánh

//Toán tử so sánh(lưu ý khi so sánh nhớ dùng 2 dấu "==")

var a = 1;

var b = 2;

if (a<b){

    alert('Phu');

}

### - Toán tử logic

// Toán tử logic

var a = 1;

var b = 2;

if (a > 0 && b > 0){

    alert('a và b đều lớn hơn 0')

}

## 2.Toán tử số học

### -Cộng (+)

var a = 2;

var b = 3;

var c = a + b;

console.log(c);

### -Trừ (-)

var a = 2;

var b = 3;

var c = a - b;

console.log(c);

### -Nhân (\*)

var a = 2;

var b = 3;

var c = a \* b;

console.log(c);

### -Chia (/)

var a = 2;

var b = 3;

var c = a / b;

console.log(c);

### -Chia lấy số dư (%): dùng kiểm tra chẵn lẻ (= 0 là chẵn, = 1 là số lẻ)

var a = 8;

var b = 5;

var c = a % b;

console.log(c);

// kết quả dư 3

### -Lũy thừa (\*\*)

var c = 3 \*\* 3;

console.log(c);

### -Tăng 1 giá trị số (++)

var a = 8;

a++;

console.log(a);

// kết quả dư 9

### -Giảm 1 giá trị số (--)

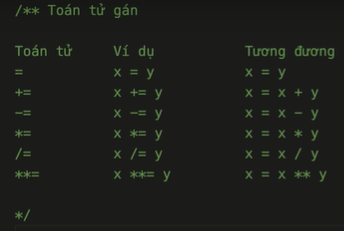
var a = 8;

a--;

console.log(a);

// kết quả dư 7

## 3.Toán tử gán



### -Toán tử =

var a = 1;

a = a + 2;

console.log(a);

### -Toán tử +=

var a = 1;

a += 2;

console.log(a);

### -Toán tử -=

var a = 1;

a -= 2;

console.log(a);

### -Toán tử \*=

var a = 2;

a \*= 2;

console.log(a);

### -Toán tử /=

var a = 2;

a /= 2;

console.log(a);

### -Toán tử \*\*=

var a = 2;

a \*\*= 2;

console.log(a);

## 4.Toán tử ++ & --

### 4.1.Toán tử ++

#### -Dùng làm hậu tố: variable++ (toán tử nằm sau biến)

var number = 1;

console.log(number++); // 1

console.log(number); // 2

console.log(number++); // 2

console.log(number); // 3

=>Toán tử ++ khi dùng là hậu tố sẽ tăng giá trị của biến lên 1 và trả về giá trị trước khi tăng. (Hiểu như là cộng sau trả về trước)

#### -Dùng làm tiền tố: ++variable (toán tử nằm trước biến)

var number = 1;

console.log(++number); // 2

console.log(number); // 2

console.log(++number); // 3

console.log(number); // 3

=>Toán tử ++ khi dùng là tiền tố sẽ tăng giá trị của biến lên 1 và trả về giá trị sau khi tăng. (Hiểu như là cộng trước trả sau)

### 4.2.Toán tử --

#### -Cách hoạt động tương tự như toán tử ++, điểm khác biệt là thay vì cộng thêm 1, thì toán tử -- sẽ trừ đi 1.

### 4.3. Nguyên lý hoạt động của ++ / --

#### Ví dụ ++ & --

var a = 6;

var output = a++;

/\*\*

 \* Việc 1: gọi a là 'a copy', a copy = 6

 \* Việc 2: +1 cho a, a = a + 1 => a = 7

 \* Việc 3: trả về a copy

 \*/

console.log('output:', output);

console.log('a:', a);

var a = 6;

var output = a++ + --a;

/\*\*

 \* kết quả output = 6 + 6 =12 (6 + --(7))

 \*/

console.log('output:', output);

var a = 6;

var output = ++a \* 2 - a-- \* 2;

/\*\*

 \* kết quả output = 7 \* 2 - 7 \* 2 =0

 \*/

console.log('output:', output);

## 5.Toán tử chuổi

### -Ví dụ 1:

var firstname = 'Phu';

var age = '24';

console.log(firstname + ' ' + age)

### -Ví dụ 2:

var firstname = 'Phu';

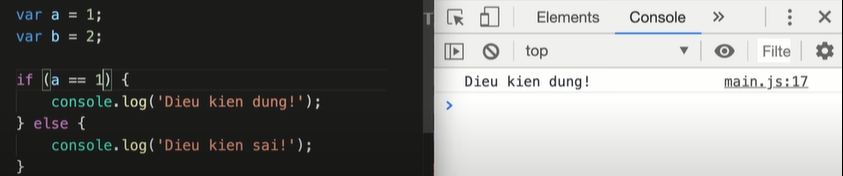
firstname += ' ' + '28';

console.log(firstname)

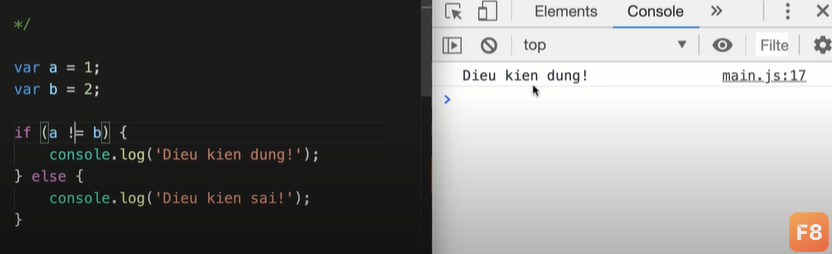
// Kết quả: Phu 28

## 6.Toán tử so sánh 1

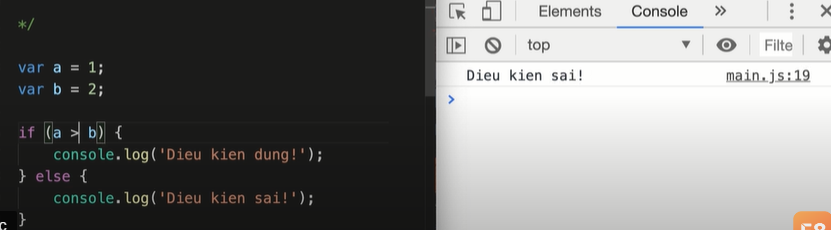
### -Bằng ==



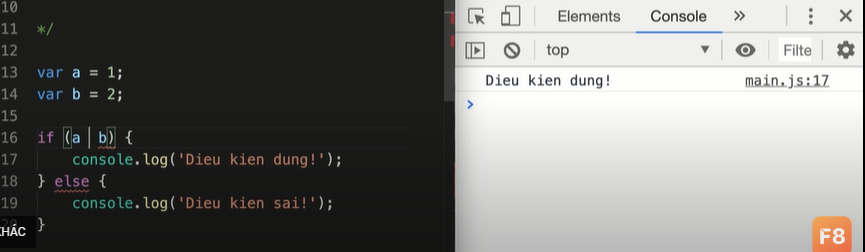
### -Không bằng !=



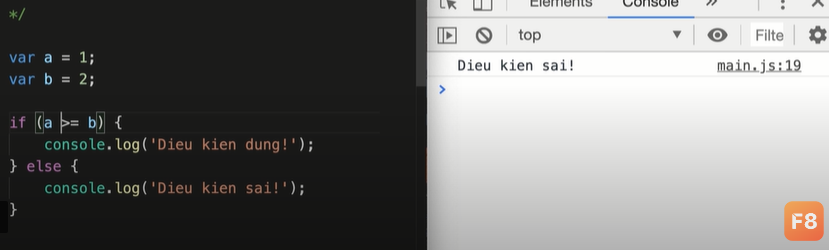
### -Lớn hơn >



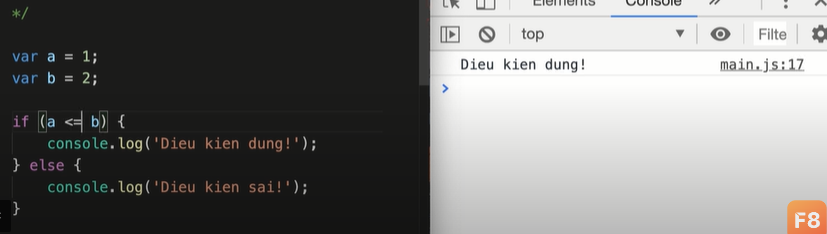
### -Nhỏ hơn <



### -Lớn hơn hoặc bằng >=



### -Nhỏ hơn hoặc bằng <=



## 7.Kiểu dữ liệu Boolean (true or false)

var a = 1;

var b = 2;

var isSuccess = a < b;

console.log(isSuccess);

// Kết quả là true

## 8.câu lệnh điều kiện if

/\*\*

 \* if (điều kiện) {

 \*      ĐÚNG

 \* } else {

 \*      SAI

 \* }

 \*/

### -Ví dụ không dùng 6 giá trị là true:

// Câu lệnh điều kiện If

// Có 6 giá trị trong js

/\*\*

 \* nếu mà sử dụng 6 giá trị này dùng cho boolean thì false

 \* ngược lại không dùng 6 giá trị cho boolean là true

 \*/

/\*\*

 \* 0

 \* false

 \* '' hay ""

 \* undefined

 \* NaN

 \* null

 \*/

// ví dụ không dùng 6 giá trị cho boolean là true

var fullname = 'ca phu';

if (fullname) {

    console.log('DIEU KIEN DUNG');

} else {

    console.log('DIEU KIEN SAI');

}

// Kết quả: là true điều kiện đúng hết

// Số chẵn là số khi chia cho 2 sẽ có phần dư là 0

// Số lẻ là số khi chia cho 2 sẽ có phần dư là 1

// Khai báo biến number, và gán trá trị cho biến number

var number = 13;

// Sử dụng toán tử % để lấy ra phần dư của number khi chia cho 2

// Khai báo biến remian ,và gán phần dư cho biến đặt tên là remain

var remain = number % 2

// Sử dụng toán tử so sánh === để thực hiện so sánh biến remain với 1

// Khai báo biến isOdd, và gán kết quả so sánh cho biến isOdd

var isOdd = remain === 1;

// In giá trị của biến isOdd ra cửa sổ console của trình duyệt

console.log(isOdd)

// Output: true

### - Ví dụ dùng 6 giá trị là false:

// Câu lệnh điều kiện If

// Có 6 giá trị trong js

/\*\*

 \* nếu mà sử dụng 6 giá trị này dùng cho boolean thì false

 \* ngược lại không dùng 6 giá trị cho boolean là true

 \*/

/\*\*

 \* 0

 \* false

 \* '' hay ""

 \* undefined

 \* NaN

 \* null

 \*/

// ví dụ không dùng 6 giá trị cho boolean là true

var fullname = false;

if (fullname) {

    console.log('DIEU KIEN DUNG');

} else {

    console.log('DIEU KIEN SAI');

}

// Kết quả: là false điều kiện sai hết

## 9.Toán tử logical

### 9.1. && là Kí hiệu của And (thì xét tất cả điều kiện, nếu đúng hết thì nó sẽ in ra đúng, nếu 1 sai thì ko thể đúng được)

var a = 1;

var b = 2;

var c = 3;

if (a > 0 && b > 0 && c > 0) {

    console.log('DIEU KIEN DUNG');

}

### 9.2. || là Kí hiệu của Or (thì xét 1 trong 2 điều kiện, nếu 1 trong 2 cái đúng thì sẽ đúng)

var a = 1;

var b = 2;

var c = 3;

if (a < 0 || b < 0 || c > 0) {

    console.log('DIEU KIEN DUNG');

}

// Kết quả: là đúng nha

### 9.3. ! là Kí hiệu của Not (Phủ định lại giá trị: đúng thì nó sẽ thành sai, sai thì nó sẽ thành đúngs)

var a = 1;

var b = 2;

var c = 3;

if (!(a < 0)) {

    console.log('DIEU KIEN DUNG');

}

// Kết quả: nếu a < 0 thì nó phủ định lại là đúng nha còn nếu a > 0 thì nó phủ định lại là sai nha

## 10.Kiểu dữ liệu

### -Typeof dùng để kiểm tra xem đó là kiểu dữ liệu gì.

### 10.1.Kiểu dữ liệu nguyên thủy

#### -Number

// Number type

var a = 1;

var b = 2;

var c = 1.5;

#### -String

// string type(nó là một chuổi kí tự bao được bao quanh dấu nháy kép "" hoặc dấu nháy đơn)

var fullname = 'ca phu';

#### -Boolean

/\*\*boolean kiểu thể hiện đúng hay sai(on hay off, thành công hay không thành công),

 \* boolean thường đặt tên có chữ is đằng trước(ví dụ: isSuccess) \*/

var isSuccess = true;

#### -Undefined

// Undefined type được hiểu là mình chỉ khai báo không gán giá trị cho nó, kiểu chỉ đặt tên

var age;

var sdt;

#### -Null

// Null được hiểu không có gì

var isNull = null;

console.log(isNull);

#### -Symbol

### 10.2.Kiểu dữ liệu phức tạp

#### -Function

// function là có thể gọi được

var myFunction = function() {

    alert('Hi, xin chào cac ban')

}

myFunction();

#### -Object

// Object có 2 kiểu thường dùng

var myObject = {

    myname: 'ca phu',

    age: 24,

    adress: 'HCM',

    myFunction: function() {

    }

};

// Array khác với object là dùng dấu ngoặc nhọn, và không dùng key(và có đánh dấu tự tăng(1,2,3,4...))

var myArray = [

    'caphu',

    '24',

    'HCM'

];

console.log(myArray);

## 11.Toán tử so sánh 2

### -3 dấu bằng (=== và !==) nó khác với 2 dấu bằng (== và !=) là nó so sánh cả kiểu dữ liệu và value luôn.

### 11.1.Toán tử bằng (===)

var a = 1;

var b = '1';

console.log(a===b)

### 11.2.Toán tử không bằng (!==)

var a = '1';

var b = 1;

console.log(a!==b)

## 12.Toán tử logical và câu lệnh điều kiện if

### 12.1. Toán tử And

/\*\*

 \* toán tử And này được hiểu xét từ trái qua phải nếu không thuộc trong 6 giá trị false(null,'',0,false,NaN,underfine),

 \* thì nó sẽ in ra giá trị bên phải cuối

 \* còn nếu mà nó thuộc một trong 6 giá trị false thì in ra giá trị đó lun(giá trị mà ta nhập thuộc 6 giá trị false)

 \*/

var result = 'A' && 'B' && 'C';

console.log(result);

if (rseult){

    console.log('Dieu kien dung');

} else{

    console.log('Dieu kien sai');

}

### 12.2. Toán tử Or

/\*\*

 \* toán tử Or này được hiểu xét từ trái qua phải nếu đúng thì nó sẽ in ra giá trị bên trái đầu tiên mà nó đọc lun

 \* còn nếu mà nó thuộc một trong 6 giá trị false(null,'',0,false,NaN,underfine),

 \* thì thằng Or nó sko quan tâm đâu, nó xét chỉ một cái đúng thì nó in ra lun.

 \*/

var result = null || 'B' || 'C';

console.log(rseult);

if (rseult){

    console.log('Dieu kien dung');

} else{

    console.log('Dieu kien sai');

}

# Bài 4: Làm Việc Với Hàm

## 1.Khái niệm hàm(funcion)

### 1.1.Hàm

-Là một khối mã.

-Làm một việc cụ thể.

### 1.2.Loại hàm

-Built-in.

-Tự định nghĩa

### 1.3.Tính chất

-Không thực thi khi định nghĩa.

-Sẽ thực thi khi được gọi.

-Có thể nhận tham số.

-Có thể trả về một giá trị.

function showDialog() {

    alert('Xin Chao Cac Ban!')

}

showDialog();

## 2.Tham số trong hàm

### 2.1.Tham số

-Định nghĩa: tham số nó là một giá trị truyền vào khi ta gọi Function

-Kiểu dữ liệu: ta truyền kiểu dữ liệu gì cũng được(string, number, ofject,…)

function showMessage(message) {

    console.log(message);

}

showMessage("Hi anh em F8!");

-Tính private

### 2.2.Truyền tham số

#### -Một Tham số

function showMessage(message) {

    console.log(message);

}

showMessage("Hi anh em F8!");

#### -Nhiều tham số

function showMessage(message, message2) {

    console.log(message);

    console.log(message2);

}

showMessage("Hi anh em F8!","Mình là Cả Phú");

### 2.3.Arguments

#### -Đối tượng arguments: là nó chỉ xuất hiện trong Function, cho nó nằm ngoài lỗi gán chịu.

function showMessage() {

    console.log(arguments);

}

showMessage("Hi anh em F8!","Mình là Cả Phú");

#### -Giới thiệu vòng for of

function showMessage() {

    var myString = '';

    for (var param of arguments){

        myString += `${param} - `;

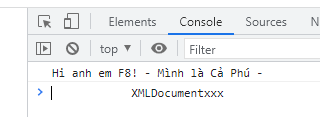
    }

    console.log(myString);

}

showMessage('Hi anh em F8!','Mình là Cả Phú');

#### -Kết quả:



## 3.Từ khóa return trong hàm

### -return có nghĩa là trả về.

var result = cong(2,8);

function cong(a,b) {

    return a + b;

}

console.log(result);

## 4.Hiểu hơn về hàm

### 4.1.Khi mà ta khai báo biến trong hàm thì chỉ được vây quanh trong hàm thôi, khi báo ngoài sẽ bị lỗi(console gọi tên chỉ được dùng trong function thôi).

-Ví dụ 1:

function showMessage() {

    var fullname = 'ca phu';

    console.log(fullname);

}

showMessage();

-Ví dụ 2:

function showMessage() {

    function showMessage2() {

        console.log('Message2');

    }

    showMessage2();

}

showMessage();

## 5.Các loại hàm(Function)

### 5.1.Declaration Function

#### -Bắt dầu bằng từ function – kế tiếp đặt tên(vậy mới gọi Declaration Function)

function showMessage() {

}

### 5.2.Expression Function

#### -Xuất hiện bằng toán tử gán

var showMessage2 = function(){

}

setTimeout(function() {

});

var myObject = {

    myFunction: function() {

    }

}

### 5.3.Arrow Function

# Bài 5: Làm Việc Với Chuỗi

## 1.Kiểu dữ liệu (string)

### 1.1.Tạo chuỗi

var fullname = 'ca phu';

console.log(fullname);

### 1.2.Một số case sử dụng backslash (\)

var fullname = 'ca phu \'is real\'';

console.log(fullname);

### 1.3.Xem độ dài của chuổi

var fullname = 'ca phu \'is real\'';

console.log(fullname.length);

### 1.4.Khi viết code String dài quá ta chỉ cần thêm dấu +

var fullname = 'ca phu'

+'1.Một số case sử dụng backslash'

+'2.Một số case sử dụng backslash'

+'3.Một số case sử dụng backslash'

+'4.Một số case sử dụng backslash';

console.log(fullname);

### 1.5.Template string ES6

var firstName = 'ca';

var lastName = 'phu';

// console.log('Tôi là: ' + firstName + ' ' + lastName);

// Thay vì viết string như trên dùng string es6

console.log(`Toi la: ${firstName} ${lastName}`);

## 2.Làm việc với chuỗi

### 2.1.Length(đo độ dài của một chuỗi)

var myString = 'Học js tai F8!';

console.log(myString.length);

function getContentLength(content) {

    return content.length;

}

console.log(getContentLength('JavaScript'));

console.log(getContentLength('Hello World'));

### 2.2.Find index(tìm vị trí kí tự đang nằm ở đâu, như nào)

/\*\*

 \* Kết quả: 4 nếu ko tìm được nó xuất ra -1

 \* nó đếm từ trái qua bắt đầu từ 0

 \* trường hợp xuất hiện nhiều từ js thì nó vẫn ở vị trí số 4

 \*/

// console.log(myString.indexOf('js'));

// vài cách tìm js:

// tìm băng cách cho nó vị trí tìm

var myString = 'Học js tai js js F8!';

// console.log(myString.indexOf('js', 6));

// cách này nó tìm từ bên phải qua

// console.log(myString.lastIndexOf('js'));

// tìm bằng lệnh search

console.log(myString.search('js'));

### 2.3.Cut string

// cut string slice(vị trí đầu, vị trí cuối)

var myString = 'Học js tai F8!';

// console.log(myString.slice(4,6));

// này là cắt từ trái sang phải Kết quả: js

console.log(myString.slice(-3,-1));

// Cắt từ phải sang trái Kết quả: F8

### 2.4.Relace (dùng ghi đè)

// replace dùng ghi đè

var myString = 'Học js js js tai F8!';

// 1 chữ js

// console.log(myString.replace('js','Javascript'));

//Nhiều chữ js

console.log(myString.replace(/js/g,'Javascript'));

### 2.5.Convert to upper case(chuyển đổi tất cả thành chữ hoa)

var myString = 'Học js tai F8!';

console.log(myString.toLocaleUpperCase())

### 2.6.Convert to lower case(chuyển đổi tất cả thành chữ thường)

var myString = 'Học js tai F8!';

console.log(myString.toLocaleLowerCase())

### 2.7.Trim(loại bỏ khoảng trắng khi nhập chuỗi)

var myString = 'Học js tai F8!';

console.log(myString.trim())

### 2.8.Split(Giúp cắt chuổi thành một array(mảng) bằng cách tìm được điểm chung của chuổi đó

var language = 'Javascript, PHP, Ruby';

console.log(language.split(', '))

var coursesStr = 'HTML & CSS, JavaScript, ReactJS';

function strToArray(str) {

    return str.split(', ');

}

// Expected results

console.log(strToArray(coursesStr))

// Output: ['HTML & CSS', 'JavaScript', 'ReactJS']

### 2.9.Get a character by index(từ một cái in dex cho trước ta có thể lấy ra một chữ cái tương ứng)

const myString2 = 'Ca Phu';

// Cách 1

console.log(myString2.charAt(0)); // Kết quả: C

// Cách 2

console.log(myString2[0]); // Kết quả: C