

TÀI LIỆU LẬP TRÌNH WEB

**JAVASCRIPT**



October 29, 2022

LÂM VŨ THÀNH TÀI

0787903125 - lamvuthanhtai14042002@gmail.com

**MENU**

[**1.** **Giới Thiệu JS** 3](#_Toc119176004)

[**2.** **Sử dụng JS trong file HTML** 3](#_Toc119176005)

[**3.** **Biến JS** 4](#_Toc119176006)

[**4.** **Comments** 4](#_Toc119176007)

[**5.** **Hàm Built-in** 5](#_Toc119176008)

[**5.1** **Alert** 5](#_Toc119176009)

[**5.2** **Console** 5](#_Toc119176010)

[**5.2.1** **console.log** 5](#_Toc119176011)

[**5.2.2** **console.warn** 6](#_Toc119176012)

[**5.2.3** **console.error** 6](#_Toc119176013)

[**5.3** **Confirm** 6](#_Toc119176014)

[**5.4** **Prompt** 6](#_Toc119176015)

[**5.5** **Set timeout** 6](#_Toc119176016)

[**5.6** **Set interval** 7](#_Toc119176017)

[**6.** **Operator (Toán Tử)** 7](#_Toc119176018)

[**6.1** **Arithmetic (Số học)** 7](#_Toc119176019)

[**6.1.1** **Prefix and Postfix (Tiền tố và hậu tố)** 8](#_Toc119176020)

[**6.2** **Assignment (Gán)** 9](#_Toc119176021)

[**6.3** **Comparison (So sánh)** 9](#_Toc119176022)

[**6.4** **Logic (logical)** 9](#_Toc119176023)

[**6.5** **String (chuỗi)** 10](#_Toc119176024)

[**7.** **Câu lệnh điều kiện If- else** 10](#_Toc119176025)

[**8.** **Kiểu dữ liệu trong JS (Data types)** 13](#_Toc119176026)

[**8.1** **Kiểu dữ liệu nguyên thủy (Primary Data)** 13](#_Toc119176027)

[**8.1.1** **Number** 13](#_Toc119176028)

[**8.1.2** **BigInt** 14](#_Toc119176029)

[**8.1.3** **String** 14](#_Toc119176030)

[**8.1.4** **Boolean** 14](#_Toc119176031)

[**8.1.5** **Undefined** 14](#_Toc119176032)

[**8.1.6** **Null** 14](#_Toc119176033)

[**8.1.7** **Symbol** 14](#_Toc119176034)

[**8.2** **Kiểu dữ liệu phức tạp (Complex Data)** 15](#_Toc119176035)

[**8.2.1** **Function** 15](#_Toc119176036)

[**8.2.2** **Object** 15](#_Toc119176037)

[**8.2.3** **Array** 15](#_Toc119176038)

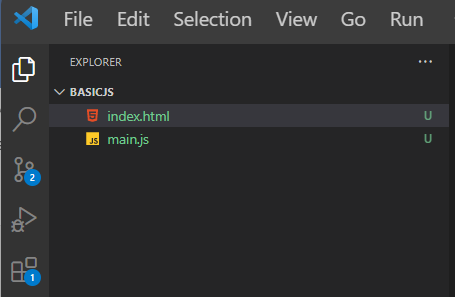
[**8.2.4** **Cách kiểm tra kiểu dữ liệu** 15](#_Toc119176039)

1. **Giới Thiệu JS**

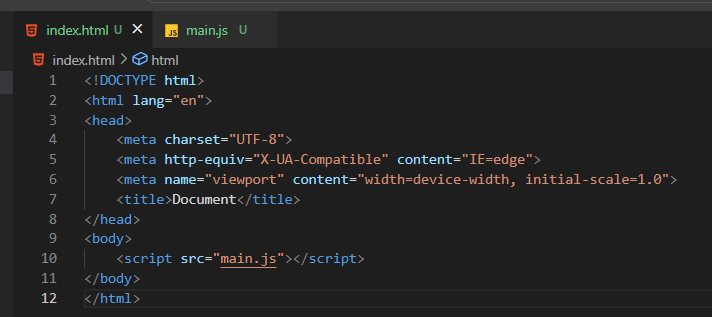
-Giúp tạo những hiệu ứng phức tạp và tạo những sự kiện, trải nghiệm người dùng tốt hơn, bên cạnh đó cũng giúp việc kết hợp giữa font end và back end dễ dàng hơn

1. **Sử dụng JS trong file HTML**

Bước 1: Tạo 1 file.js bên ngoài, có thể cùng cấp với file.html

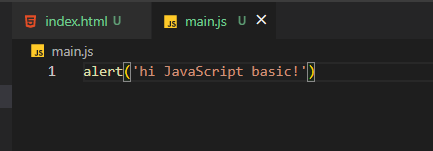


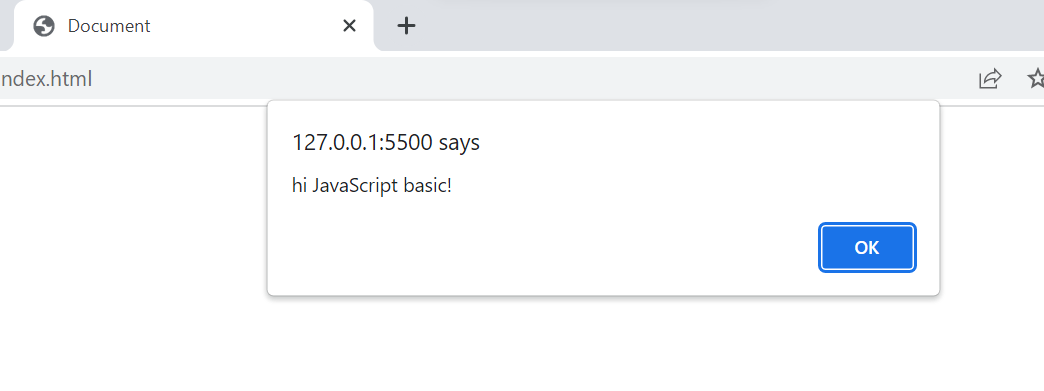
Bước 2: Trong phần body sử dụng thẻ <script></script> và link đến file.js vừa tạo



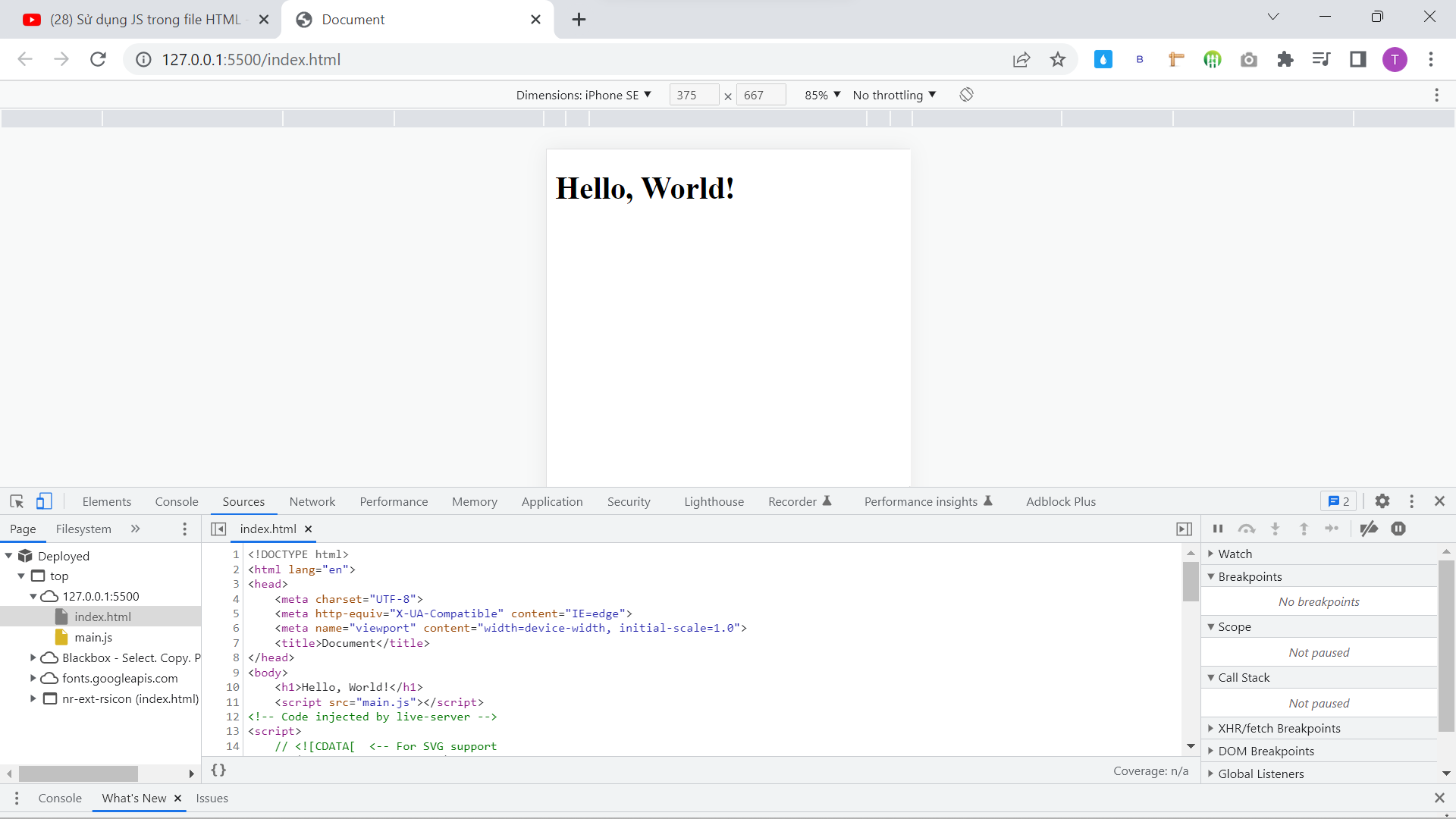
Bước 3: kiểm tra liên kết có thành công hay không

+Code vài dòng bên file.js sau đó chạy live sever check kết quả





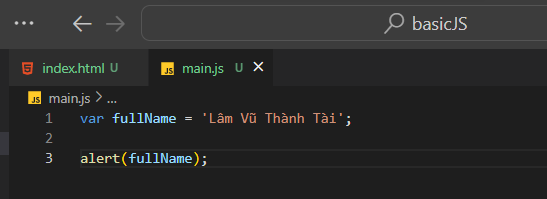
+Kiểm tra bằng Devtool



1. **Biến JS**

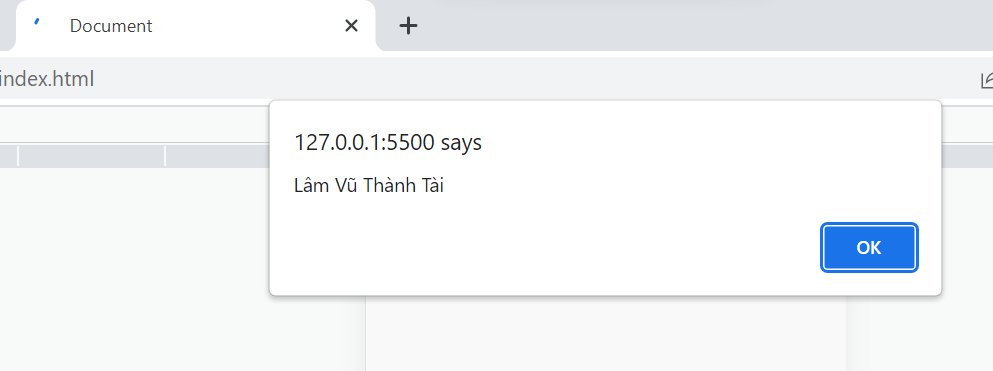
-Dùng **var** để khai báo biến:

var tênbiến = ‘Giá trị gán vào’



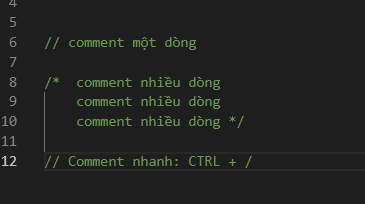
Dấu = dùng để gán giá trị cho biến

Kết quả:



1. **Comments**

-Giúp ghi chú, vô hiệu hóa hoặc giải thích code



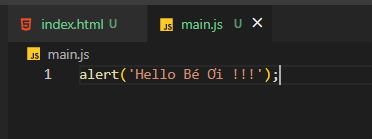
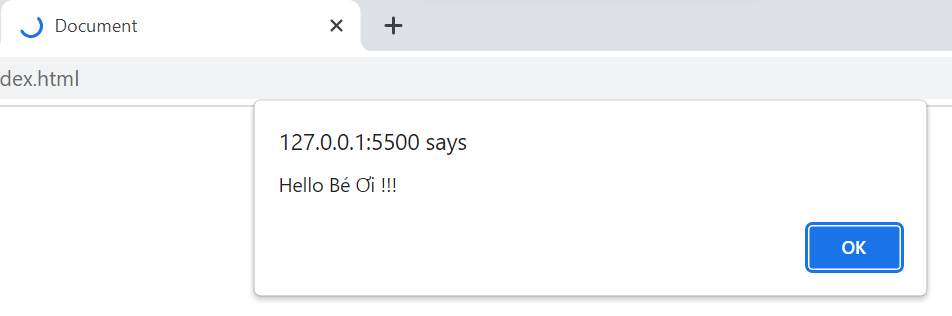
1. **Hàm Built-in**

-Là các hàm có sẵn trong JS, chỉ việc nhớ và gọi ra sài khi cần thôi !

* 1. **Alert**

-Dùng để hiển thị message (thông báo) trên web site.

-Cú pháp: alert(‘Nội dung message’);

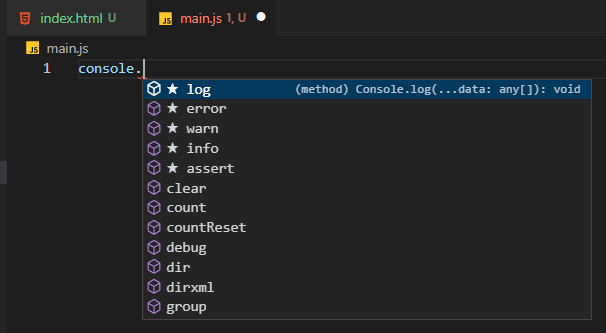
 

-Thông báo sẽ hiện cho đến khi chúng ta tắt (bấm OK).

* 1. **Console**

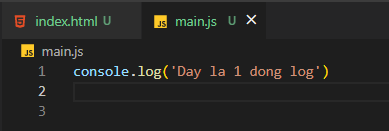
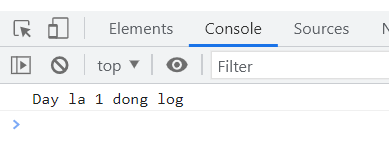
-Là một đối tượng có chứa rất nhiều các methods (phương thức).

-Cách gọi các phương thức của console ta dùng cú pháp: console + “dấu chấm”

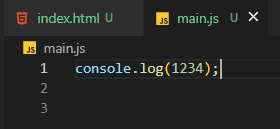
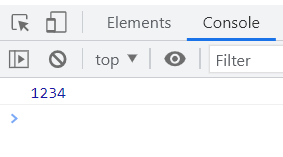


* + 1. **console.log**

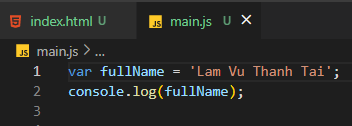
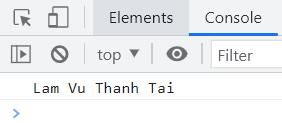
-Giúp hiển thị cảnh báo trên màn hình console (trong dev tool)

-Ngoài giá trị chuỗi ta có thể gán giá trị số

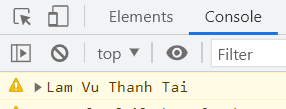
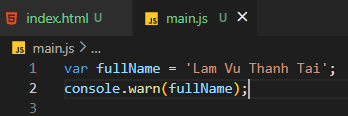
 

-Gán giá trị là biến JS

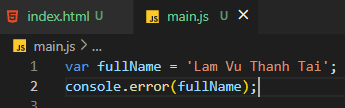
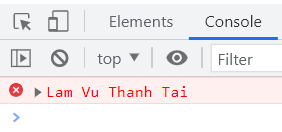
* + 1. **console.warn**

-Dùng để cảnh báo trên màng hình console



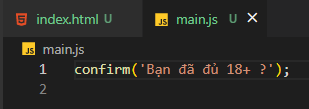
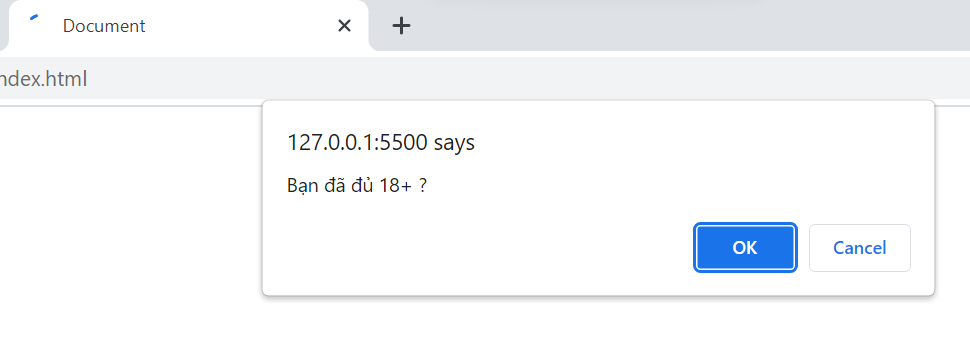
* + 1. **console.error**

-Dùng để báo lỗi trên màng hình console

* 1. **Confirm**

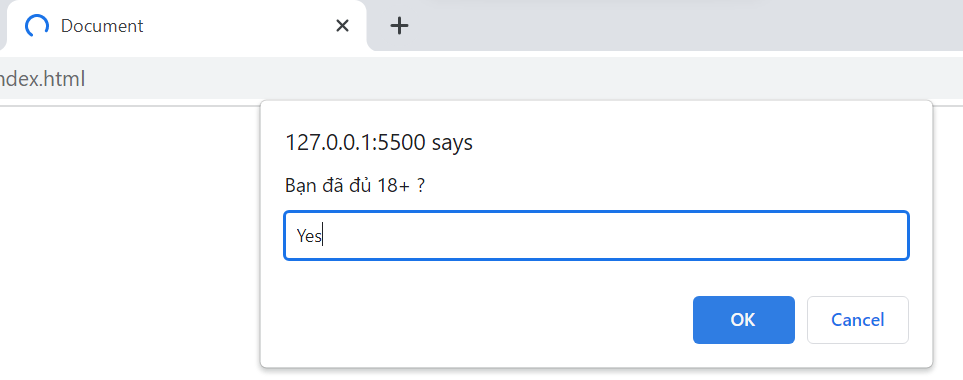
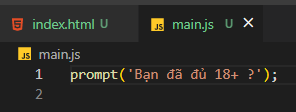
-Dùng để thông báo (message) và yêu cầu user xác nhận.

****

-Sẽ hiện hộp thông báo đến khi người dùng xác nhận.

* 1. **Prompt**

-Kết hợp Alert và Confirm, ngoài ra yêu cầu người dùng nhập input để xác nhận.



-Hộp thoại sẽ hiện đến khi user nhập input và click xác nhận.

* 1. **Set timeout**

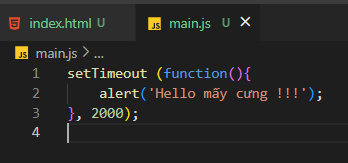
-Dùng để chạy một đoạn code trong một khoản thời gian (được tính là ms)

-Cú pháp:

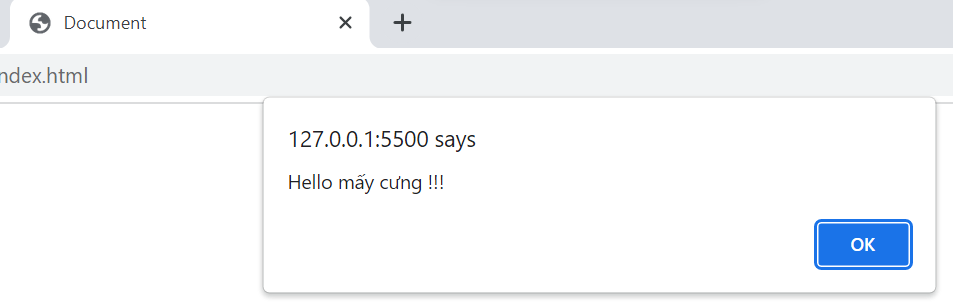
setTimeout (function(){

//Code thực thi;

}, thời gian thực hiện);



Sau 2s thì sẽ hiện lên thống báo:



* 1. **Set interval**

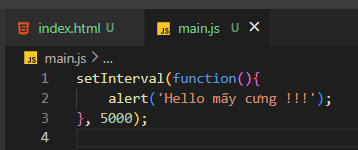
-Dùng để chạy một đoạn code trong một khoản thời gian (được tính là ms) nhưng khác với timout là nó sẽ lặp lại vô tận

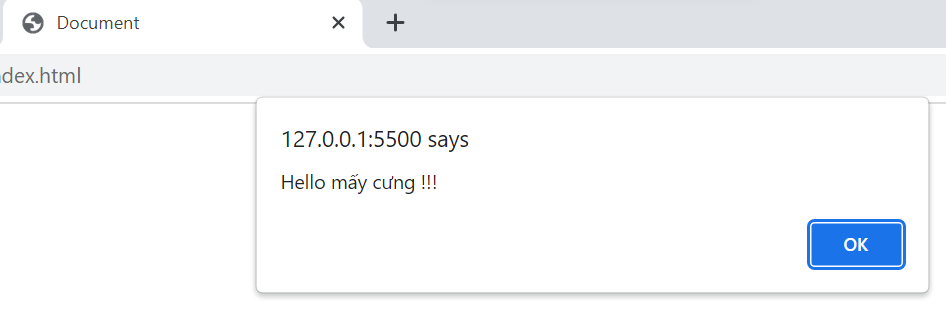
-Cú pháp:

setInterval (function(){

//Code thực thi;

}, thời gian thực hiện);

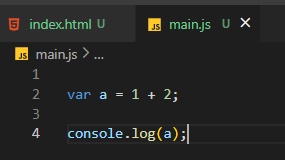
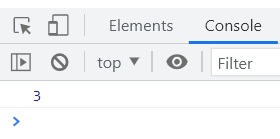
****



-Dù user có tắt thông báo thì cứ mỗi 5s sẽ hiển thị lại !

1. **Operator (Toán Tử)**
   1. **Arithmetic (Số học)**

- **Các toán tử số học** (toán tử đại số) này thực hiện trên các số - dữ liệu dạng số (cụ thể hoặc là biến). Chúng cũng gần tương tự như trong toán học thôi.

** **

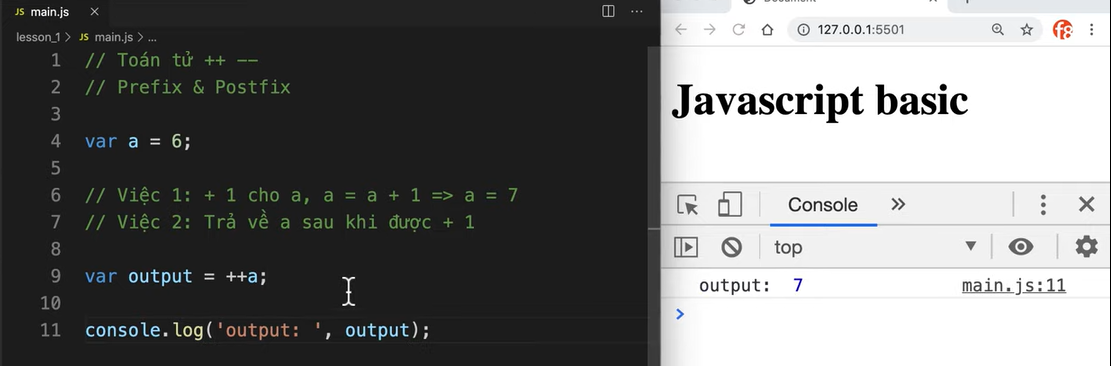
-Ngoài ra còn có các toán tử:



* + 1. **Prefix and Postfix (Tiền tố và hậu tố)**

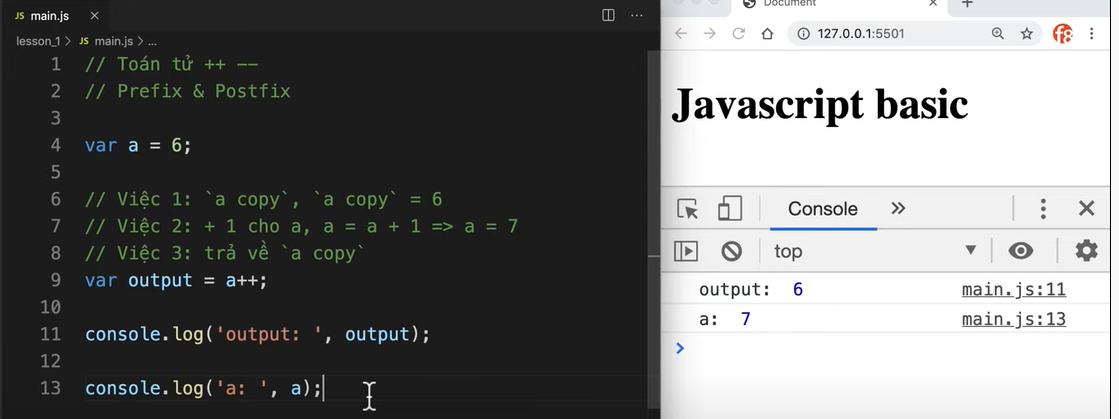
-Trong toán tử **++,** --

+Tiền tố:



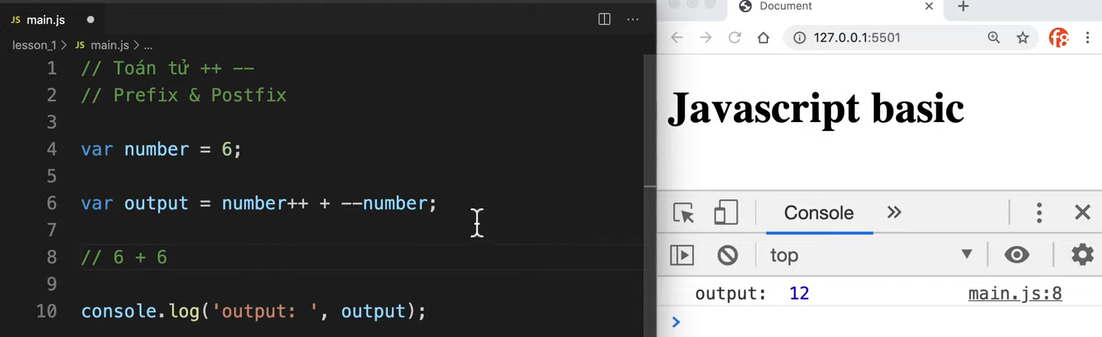
-Đối với --a; cũng giống v

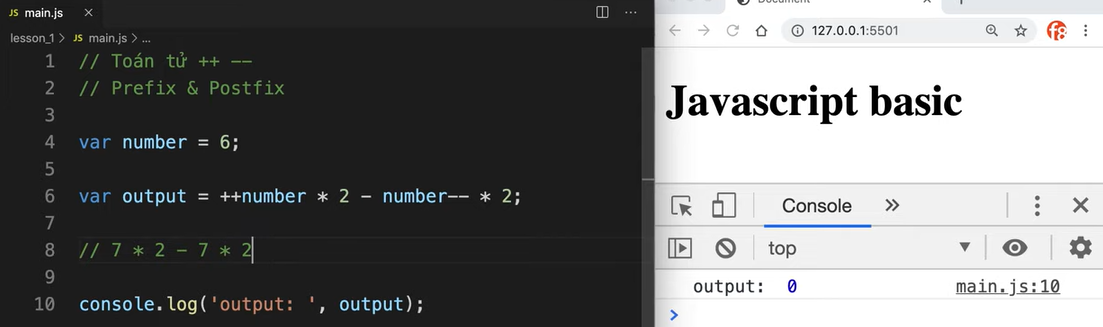
+Hậu tố:



-Đối với --a; cũng giống v

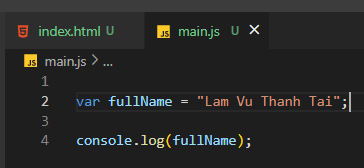
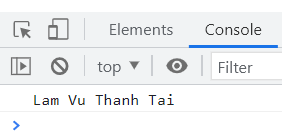
+VD:



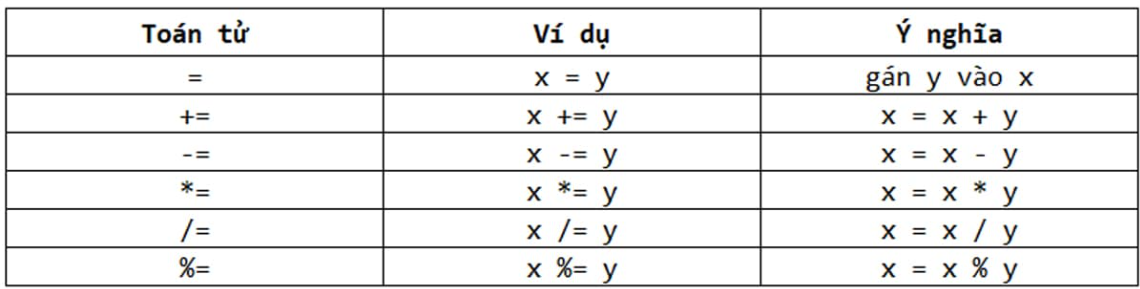


* 1. **Assignment (Gán)**

**- Toán tử gán** được dùng để **gán giá trị** ở bên phải toán tử vào biến ở bên trái toán tử.

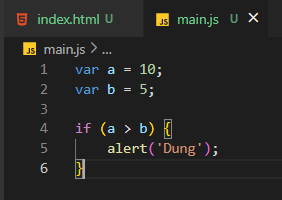
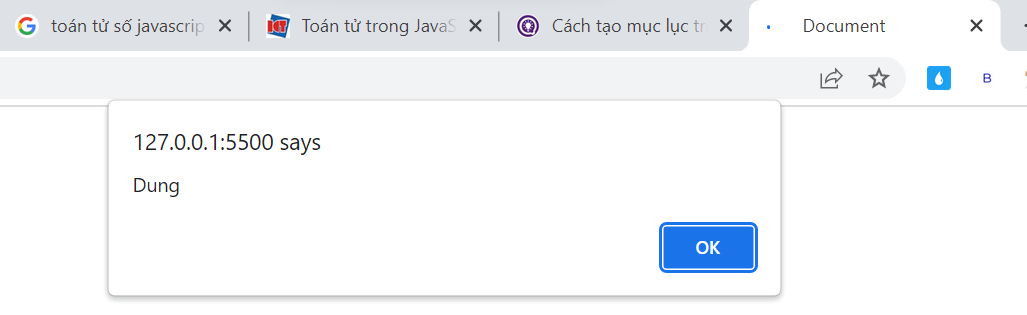
** **

-Dấu “ = ” là một trong những toán tử gán ! Ngoài ra còn:

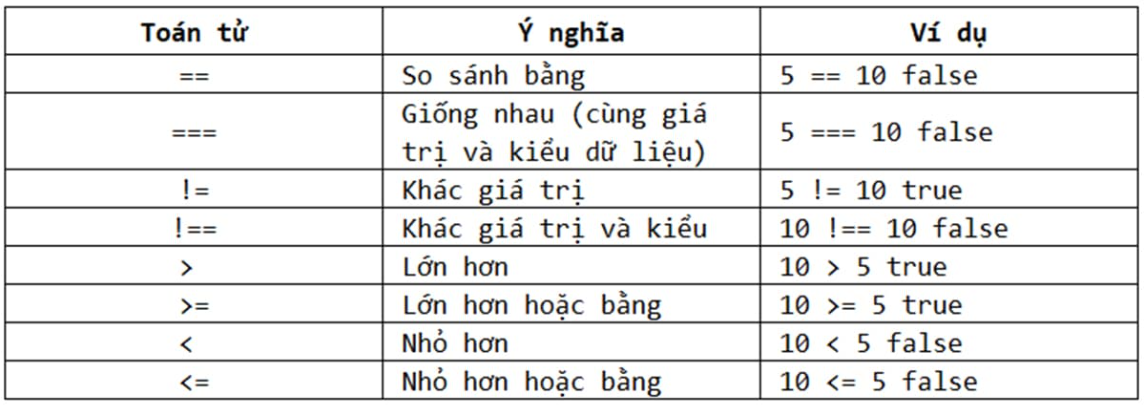


* 1. **Comparison (So sánh)**

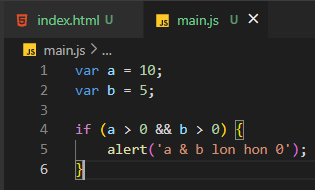
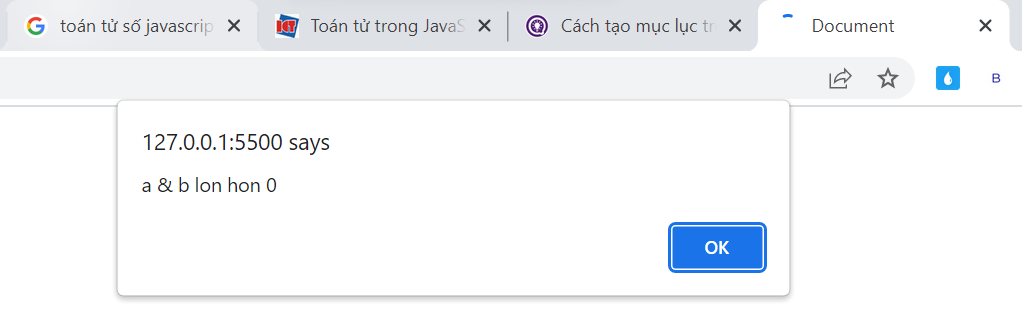
**- Toán tử so sánh trong JavaScript** sử dụng trong các biểu thức về logic để so sánh bằng, khác nhau, lớn hơn, nhỏ hơn.... Nó sẽ trả về giá trị true hoặc false.

**** ****

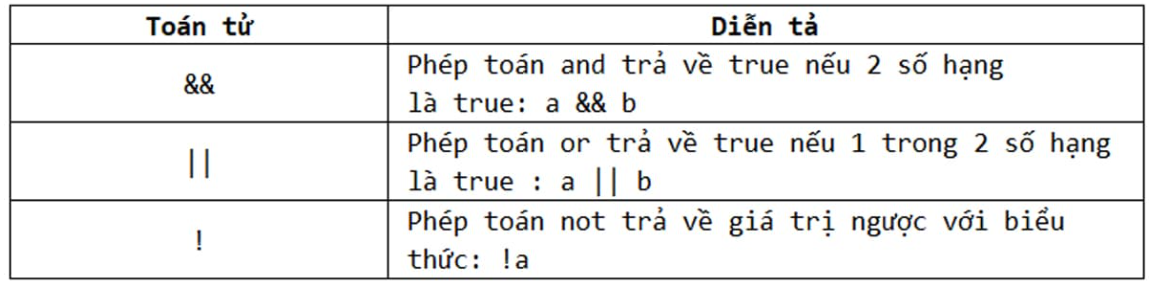
-Ngoài ra còn có các toán tử so sánh khác:

****

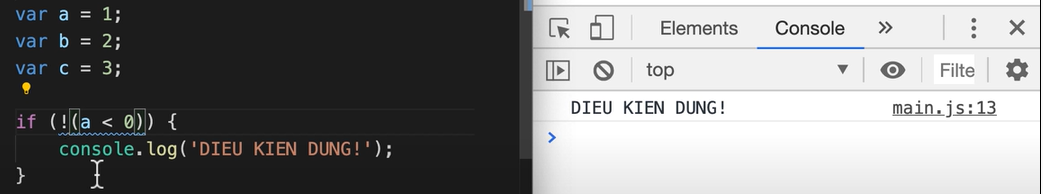
* 1. **Logic (logical)**

**** ****

**-** JavaScript cung cấp 3 loại **toán tử logic** gồm các phép toán: and, or, not.

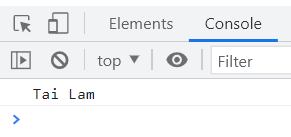
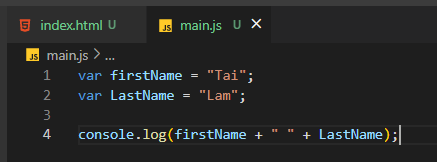
****

-Toán tử not: phủ định lại giá trị ban đầu.

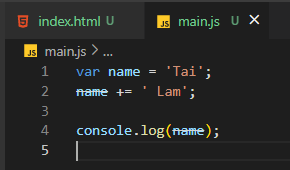
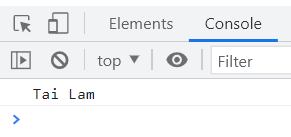
****

* 1. **String (chuỗi)**

**-**Trong JavaScript, chúng ta có thể sử dụng toán tử + để **nối hai chuỗi** lại với nhau.Nó có thể nối cả số vào chuỗi.

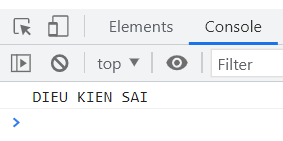
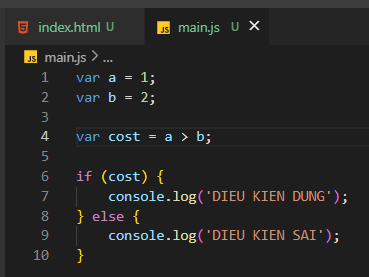
****

**-**Ngoài ra có thể kết hợp với toán tử gán.

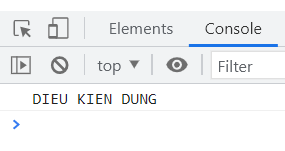
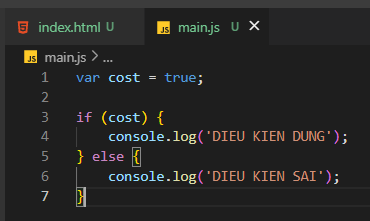
****

1. **Câu lệnh điều kiện If- else**

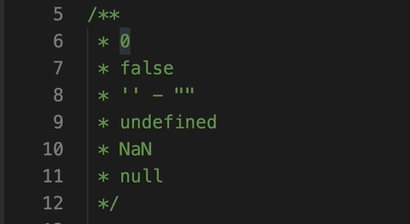
-Câu điều kiện Nếu – Thì, cấu trúc câu giống như ở các ngôn ngữ lập trình khác.

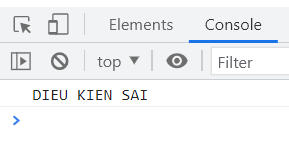
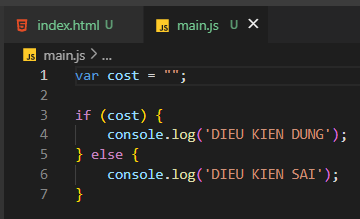
****

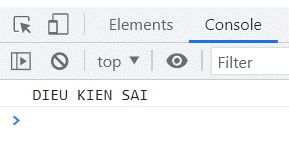
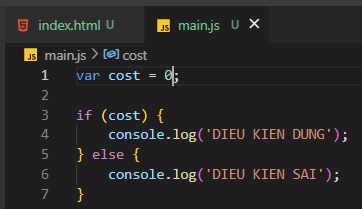
**-**Ngoài điều kiện là các toán tử ra thì còn nhận giá trị true, false.



-6 giá trị khi sử dụng JS sẽ chuyển đổi thành false.



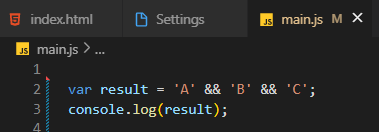
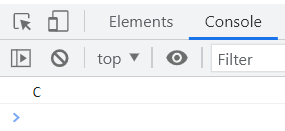


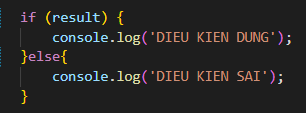
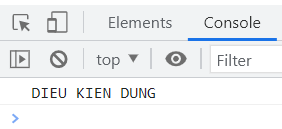


-Ngoài những giá trị này ra thì javascript sẽ convert (chuyển đổi) thành giá trị true.

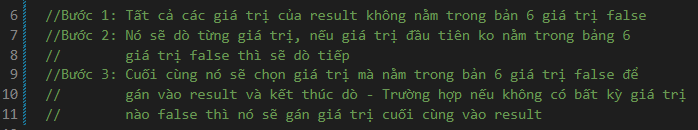
- Một số lưu ý về toán tử **And** và **OR** trong câu điều kiện if-else

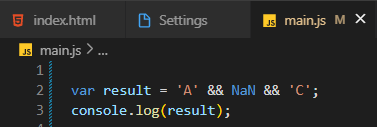
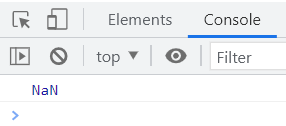
**+And**

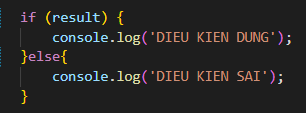
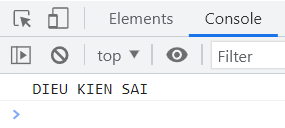
 

-Lý do in ra đáp án C là vì:



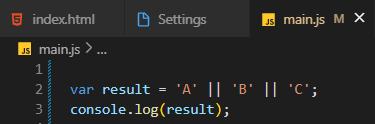
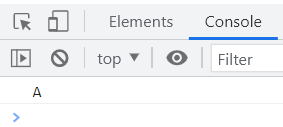
 

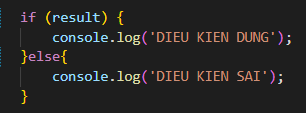
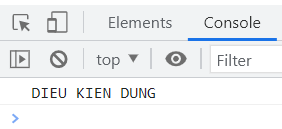
-Trường hợp có 1 giá trị nằm trong bảng 6 giá trị false thì nó sẽ gán giá trị đó cho “result” và không quan tâm vế sau.

**+OR**

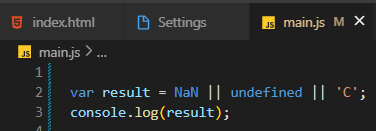
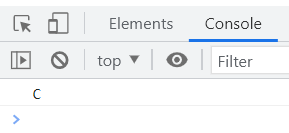
-Ngược với **And** – dò đến khi tìm giá trị false và gán vào biến

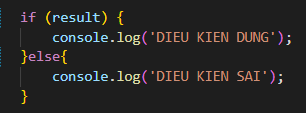
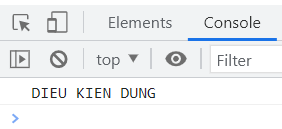
-Toán tử **OR** sẽ dò đến khi tìm được giá trị true và gán vào biến

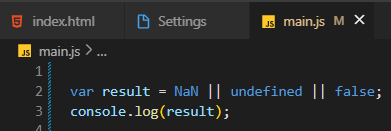
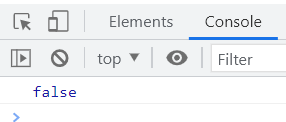
 

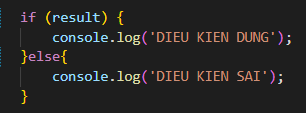
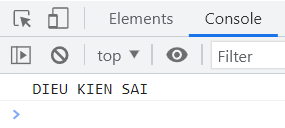
-Bởi vì A là giá trị true (không nằm trong bản 6 giá trị false) nên sẽ lấy giá trị A gán vào result và không quan tâm vế đằng sau.

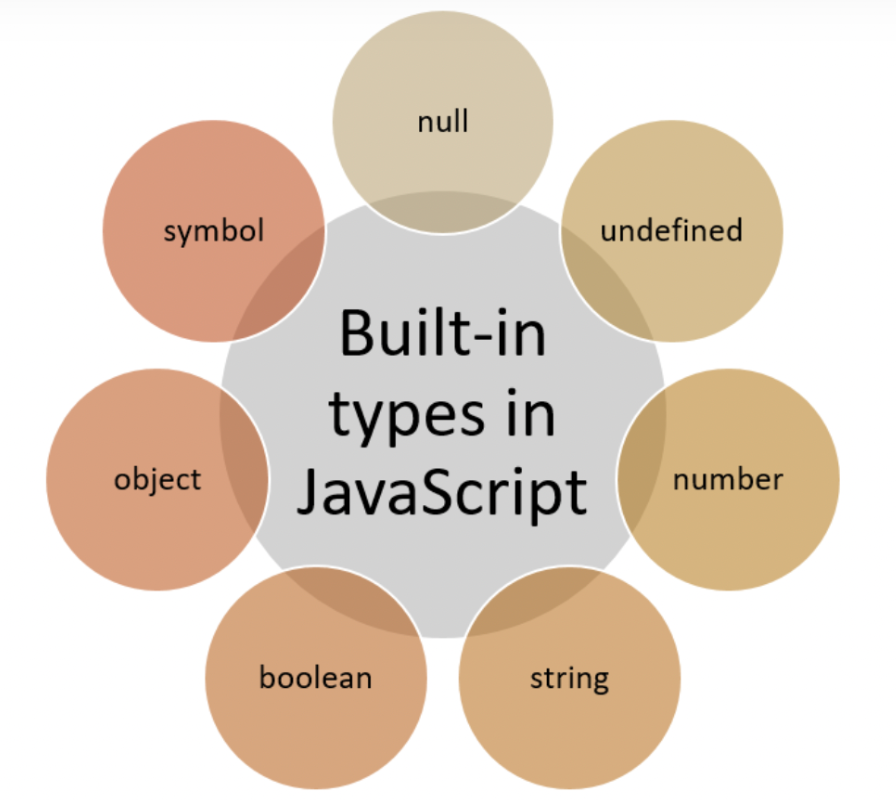
-Hai giá trị đầu là false nên sẽ được bỏ qua, đến giá trị C là giá trị true nên sẽ được gán vào result.

-Vì không có giá trị true nào nên sẽ gán false cho result

1. **Kiểu dữ liệu trong JS (Data types)**

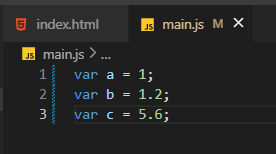
****

* 1. **Kiểu dữ liệu nguyên thủy (Primary Data)**

-Khi chúng ta gán một giá trị cho một biến thì trong RAM sẽ tạo ra một ô chứa dữ liệu của giá trị đó, nó sẽ không thể biến mất hoặc thay đổi dù chúng ta chỉnh sửa hay xóa.

* + 1. **Number**

-Gồm giá trị số nguyên, số thực.

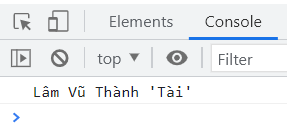
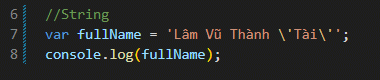
****

* + 1. **BigInt**
    2. **String**

-Là kiểu dữ liệu chứa giá trị là chuỗi, có thể điền số hoặc chữ đều đc, cách nhận biết là giá trị được bao quanh bằng dấu ngoặc đơn hoặc kép.

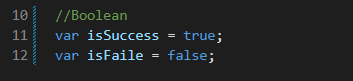
****

**-**Nếu muốn hiển thị dấu ngoặc đơn hoặc đơn hoặc kép trong một chuỗi ta sử dụng “ \ ” đặt phía trước.



* + 1. **Boolean**

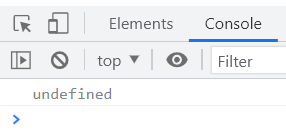
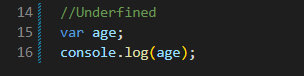
**-**Là kiểu dữ liệu chỉ biểu hiện tính “đúng” – “sai”, cách nhận biết đó là giá trị là “true” hoặc “false”.



-Thường kiểu dữ liệu Boolean người ta thường đặt biến và trước đó có chữ “is” nhằm tạo sự clean code và dễ đọc.

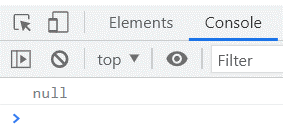
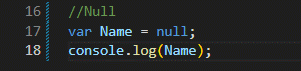
* + 1. **Undefined**

**-**Khi một biến mà không được gán bất kỳ giá trị nào, chỉ được khai báo tên biến thì ta gọi đó là undefined.



* + 1. **Null**

-Khi một biến được gán giá trị là “null” thì tức là biến đó không chứa bất kì dữ liệu gì.



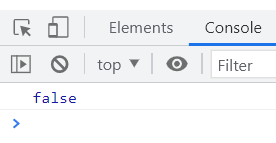
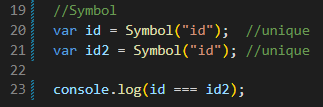
* + 1. **Symbol**

-Là kiểu dữ liệu mà dùng để mô tả một biến để biến đó là biến duy nhất

-Gán giá trị cho biến là “Symbol()”, bên trong Symbol() là phần mô tả biến đó

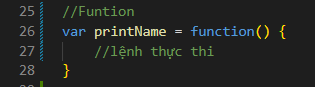


-Giá trị mô tả điền vào là duy nhất, cho dù có cùng tên biến

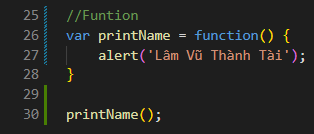
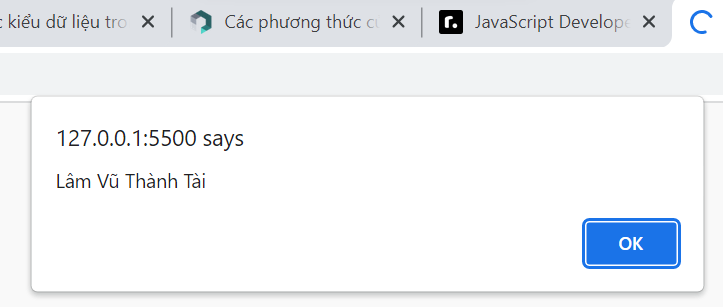


* 1. **Kiểu dữ liệu phức tạp (Complex Data)**
     1. **Function**

-Như tên gọi, là kiểu dữ liệu mà giá trị gán của biến là một hàm, bên trong hàm chứa những đoạn code thực thi một chức năng nào đó.



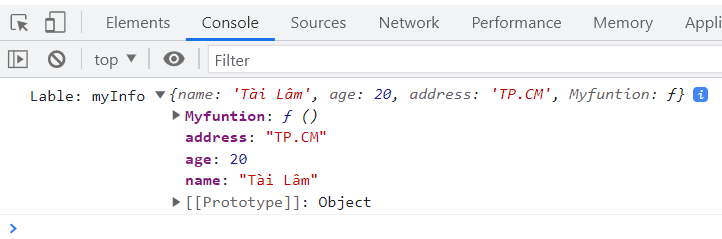
-Hàm sẽ không được chạy đến khi chúng ta gọi nó!

* + 1. **Object**

**-**Là kiểu dữ liệu mà giá trị được gán gồm “key” và “value”, trong một object thì chứa rất nhiều thứ như: giá trị (gồm nhiều kiểu dữ liệu), hàm..

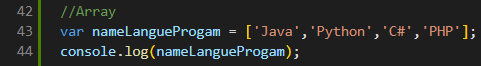
**-**Cách nhận biết đó là kiểu object là sử dụng dấu ngoặc nhọn có chấm phẩy ở cuối, ở mỗi cặp “key”-“value” cách nhau bởi dấu phẩy.

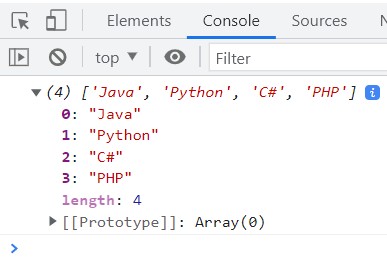
 

* + 1. **Array**

**-**Khi chúng ta cần khai báo giá trị dạng danh sách thì chúng ta sử dụng kiểu dữ liệu Array (mảng).

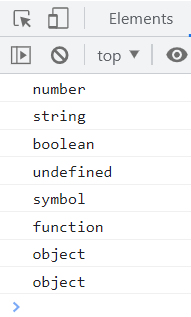
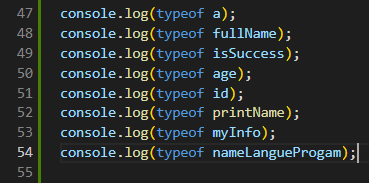
**-**Cách nhận biết kiểu Array đó dấu ngoặc vuông, bên trong ngoặc vuông có thể chứa kiểu dữ liệu số hoặc chuỗi.





* + 1. **Cách kiểm tra kiểu dữ liệu**

-Trong khi ta sử dụng hàm built-in là “console.log()” để kiểm tra thì thêm “typeof” bên trong sẽ giúp kiểm tra xem biến đó là theo kiểu dữ liệu gì



-Kiểu dữ liệu Null, Array khi hiển thị ra màn hình console thì là kiểu object (Hiện tại chưa có lời lý giải về hiện tượng này)