1. **Biến và hằng (Variable)**
   1. **Khai báo**

Có 3 cách khai báo biến: var, let, const

+ var: sử dụng cho toàn cục và có sự thay đổi liên tục

*var [dấu cách] [tên biến];*

+ let : biến mang tính chất tạm thời, được sử dụng trong phạm vi là một khối, chỉ hoạt động trong phạm vị được khai báo trong dấu {}

*let [dấu cách] [tên biến];*

+ const: là một hằng số, khi khai báo const thì phải gán giá trị cho nó luôn và không thể thay đổi giá trị của biến

*const [dấu cách] [tên biến];*

* 1. **Phạm vi (scope)**
* Trong Js có 02 phạm vi của biến
* Biến cục bộ *(Local Variable): Biến được khai báo bên trong một hàm và chỉ sử dụng được ở trong hàm đó;*
* Biến toàn cục *(Global Variable): Biến được khai báo bên ngoài một hàm và có thể được truy cập ở bất kỳ vị trí nào trong chương trình JavaScript*

1. **Kiểu dữ liệu (Data types)**
   1. **Nguyên thủy: Primitive types:**

* **Number**: dùng để biểu diễn kiểu dữ liệu số

*var numb = 100;*

* **String** : chuỗi và được khai báo trong dấu nháy đơn hoặc nháy kép

*var str = ‘ hello word’*

+ Để nối ta dùng dấu + để ghép 2 chuỗi lại với nhau

+ Ép chuỗi ta dùng cú pháp: .toString()

+ Đếm số kí tự trong chuỗi sử dụng thuộc tính length

+ Muốn tìm kiếm một chuỗi bên trong một chuỗi khác thì dùng hàm **indexOf**

* **Boolean**: có 2 giá trị true , false. Dùng để ẩn hiện trạng thái của 1 đối tượng nào đó
* **Underfile**: Biến được khai báo nhưng không có giá trị

Vd:

*Var a;*

*Console.log(a)*

*=>underfile*

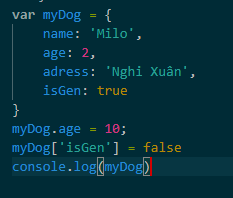
* **Null:** Biến được khai báo có giá trị null hoặc rỗng

*Vd: var a = null*

* 1. **Tham chiếu:** Non-primitive types (reference) : khi có nhiều hơn một biến được tạo để lưu trữ một object, array, function. Những biến đó được trỏ đến địa chỉ đã được phân bổ tương tự trong bộ nhớ.
* **Array**: thường được dùng để lưu trữ một list danh sách có các phần tử giống nhau

|  |  |
| --- | --- |
|  |  |

* **Object**: Dùng để mô tả một vật thể gồm nhiều thuộc tính khác nhau



* **Function**: để giải quyết, thực hiện một nhiệm vụ nào đó. Code bên trong function không được biên dịch cho tới khi được gọi đến

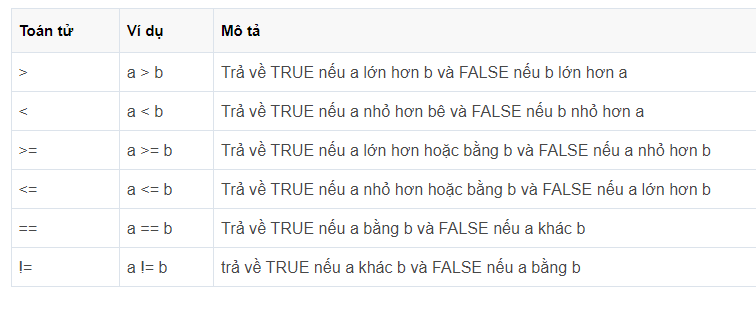
***function****doSomething(input1, input2){*

*return something*

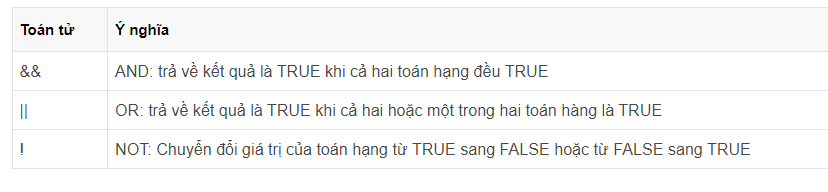
*}*

1. **Toán tử (Operators)**
   1. **Toán học**

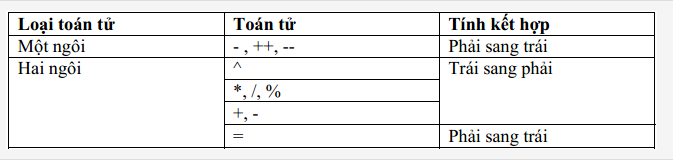
* Cộng: Phép cộng. Nếu là chuỗi thì nó sẽ thực hiện thao tác nối chuỗi, còn nêu là số thì nó sẽ cộng hai số lại.
* Trừ: Phép trừ và dùng với number
* Nhân: Phép nhân và dùng với number
* Chia: Phép chia và dùng với number
* Lấy dư: Phép chia lấy phần dư, nghĩa là khi chia hai số lại với nhau và kết quả nó sẽ lấy phần dư của phép toán
* Phép ++: Phép tăng giá trị hiện tại lên 1 đơn vị. Phép này có hai cách sử dụng đó là đặt nó trước biến và đặt nó sau biến.
* Trường hợp đứng trước biến thì nó sẽ tăng trước khi lấy giá trị.
* Trường hợp đứng sau biến thì nó sẽ lấy giá trị trước rồi tăng
* Phép --: Phép giảm giá trị hiện tại xuống 1 đơn vị. Phép này cũng có hai cách dùng đó là đặt trước biên và đặt sau biến
  1. **Gán:**
* ‘=’ :vd: x = y : Gán gía trị của biến x bằng giá trị của biến y
* ‘+=’ : vd: x+=y: tương đương x=x+y
* ‘-=’: vd: x-=y: tương đương x=x-y
* ‘\*=’ vd: x\*=y: tương đương x=x\*y
* ‘\=’ vd: x\=y: tương đương x=x\y
* ‘%=’ vd: x%=y: tương đương x=x%y
  1. **Quan hệ, so sánh:**



* 1. **Logic**: dùng để kết hợp hay phủ định biểu thức có chứa các toán tử quan hệ, kết quả của sự kết hợp sẽ có giá trị là TRUE hoặc FALSE.



* 1. **Độ ưu tiên**: Ưu tiên theo thứ tự như bảng



1. **Câu lệnh (Statements)**
   1. Phân nhánh:

* *Câu lệnh if , if-else*
* Lệnh if trong javascript

Nội dung thực hiện của lệnh if sẽ được bao quanh bởi cặp ngoặc nhọn đóng mở {}

* Lệnh if else trong javascript

Lệnh else sẽ được thực thi nếu lệnh if không được thực hiện, tức là điều kiện ở condition sẽ có giá trị là **FALSE**. Như vậy khi dùng lệnh else thì **bắt buộc** phải có một lệnh if đứng trước nó.

* Kết hợp nhiều lệnh if else trong javascript

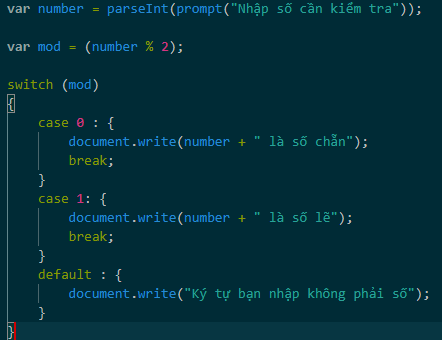
Ta có thể kết hợp nhiều câu lệnh if để xử lý bài toán, bằng cách thêm nhiều lệnh else if phía sau

* Lệnh if else lồng nhau trong javascript

Cũng giống như các ngôn ngữ lập trình khác, ta có thể lồng các câu lệnh if lại với nhau, nghĩa là bên trong câu lệnh if sẽ chứa câu lệnh if khác.

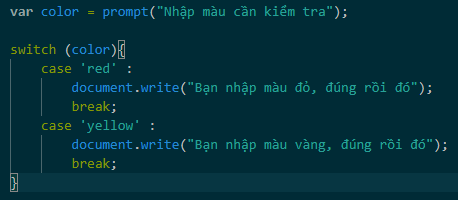
* *Lệnh switch case trong Javascript*

Lệnh switch case có công dụng giống như lệnh if else đó là đều dùng để rẻ nhánh chương trình, ứng với mỗi nhánh sẽ có một điều kiện cụ thể và điều kiện đó phải sử dụng toán tử **so sánh bằng**, còn đối với lệnh if else thì bạn có thể truyền vào một mệnh đề bất kì và sử dụng **nhiều toán tử khác nhau**.



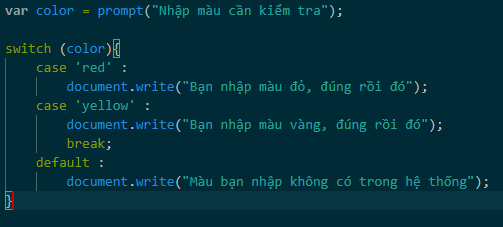
* Trường hợp không có default

Trường hợp này nếu bạn nhập một màu khác với màu đỏ (red) và vàng (yellow) thì sẽ không có thông báo gì.

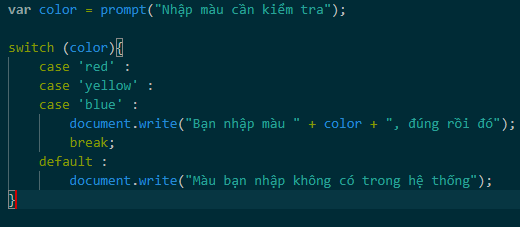


* Trường hợp không có break

Trường hợp này nếu bạn nhập vào màu đỏ (red) thì chương trình sẽ in ra cả lệnh ở case màu vàng (yellow) phía dưới, lý do là trong case màu đỏ ta không sử dụng lệnh break để thoát khỏi lệnh switch nên nó sẽ chạy thẳng xuống case phía dưới luôn mà không cần kiểm tra điều kiện.



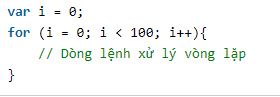
* Trường hợp gom nhóm case



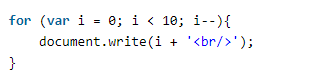
Nếu nhiều nhánh cùng xử lý chung một đoạn code thì ta sẽ bỏ đi lệnh break ở bên trong, như vậy nó sẽ chạy luôn case ở phía dưới mà không cần kiểm tra điều kiện.

* 1. **Vòng lặp:**
* Vòng lặp for

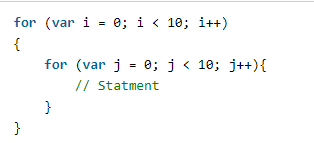
Vòng lặp for trong javascript là vòng lặp được dùng để lặp một mảng hoặc một danh sách nằm trong khoảng (min -> max). Đây là vòng lặp có thứ tự và phải biết trước tổng số lần lặp nên thường dùng trong những bài toán liệt kê và xử lý các phần tử trong danh sách mảng.



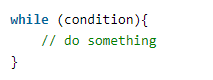
+ Vòng lặp for vô hạn



+ Vòng lặp for lồng nhau



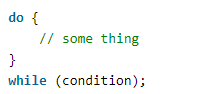
* Vòng lặp while - do while



Trong đó condition là điều kiện dừng vòng lặp, nếu condition đúng thì vòng lặp sẽ được thực thi cho tới khi condition có giá trị sai. Chính vì vậy nếu condition luôn luôn đúng thì vòng lặp sẽ dẫn tới **lặp vô hạn**

* Vòng lặp do while

**vòng lặp do while** sẽ thực hiện đoạn code bên trong lệnh **do** mới kiểm tra điều kiện



Trong đó condition là điều kiện để dừng vòng lặp

* Vòng lặp while - do while lồng nhau

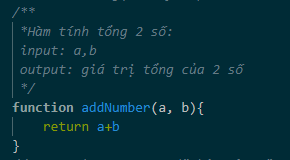
Cũng giống như trong [vòng lặp for](https://freetuts.net/vong-lap-while-trong-python-685.html), ta có thể lồng nhiều vòng lặp while hoặc do while lại với nhau để viết chương trình.

* 1. **Break và Continue trong vòng lặp**
* break: có tác dụng dừng vòng lặp cho dù điều kiện của vòng lặp **vẫn đang đúng**, hay nói cách khác là nó thoát khỏi vòng lặp một cách đột xuất và không quan tâm đến điều kiện lặp.
* continue: có tác dụng bỏ qua **một bước lặp** nào đó, nghĩa là lúc gặp lệnh continue thì tất cả những đoạn code nằm bên dưới sẽ không được thực hiện mà nó sẽ nhảy qua vòng lặp mới luôn

**5. Hàm (Function)**

- Normal function

Cách tạo hàm:



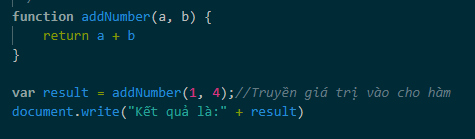
Cách gọi hàm: Để gọi hàm thì ta chỉ cần gọi đến tên hàm, sau đó truyền vào các tham số cần thiết.



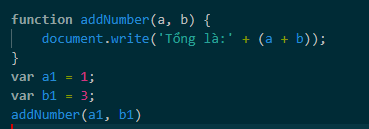
+ Hàm có return và hàm không có return

Hàm có return là hàm có sử dụng từ khóa return để đặt ở cuối hàm với mục đích trả kết quả về để sử dụng tiếp ở những đoạn code bên ngoài.

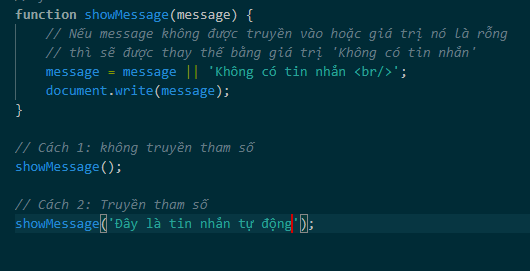
Hàm có return :



Hàm không có return:

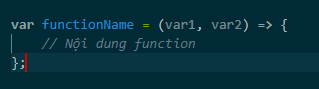


+ Giá trị mặc định của tham số: Sử dụng toán tử || để gán giá trị mặc định. Toán tử || gồm hai vế, trong đó nếu vế TRÁI có giá trị rỗng (undefined, null, false, ...) kết quả sẽ trả về vế PHẢI, ngược lại nó sẽ trả về vế TRÁI.:

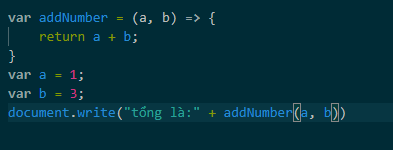


- Arrow function

Cú pháp căn bản nhất của arrow function như sau:



Vd:



- Callback: là một hàm sẽ được gọi bởi một hàm khác. Hiểu phức tạp hơn thì callback một hàm A được truyền vào hàm B thông qua các tham số của hàm B. Bên trong hàm B sẽ gọi đến hàm A để thực hiện một chức năng nào đó

Cú pháp:

**function** A() {

*// code*

}

*// Hàm B có một tham số callback*

**function** B(callback) {

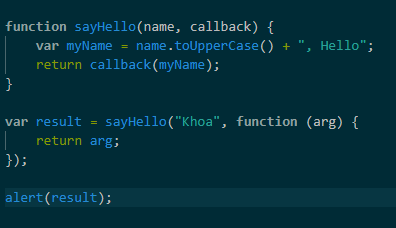
    callback();

}

*// Gọi hàm B và truyền tham số là hàm A*

B(A);

VD:

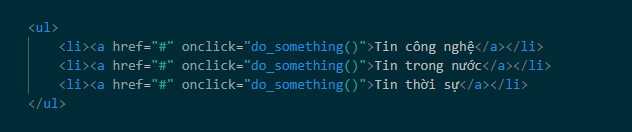


**6. Sự kiện (Event):** Sự kiện là một hành động tác động lên các đối tượng HTML, qua đó ta có thể bắt được sự kiện và yêu cầu javascript thực thi một chương trình nào đó.

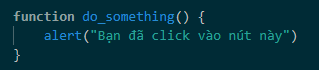
Có 2 cách để bắt sự kiện:

- Cách 1: Bắt sự kiện trực tiếp trong thẻ HTML

VD:

HTML:

Gọi hàm:

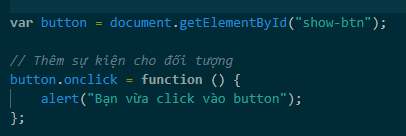


- Cách 2: Bắt sự kiện cho một thẻ HTML bằng javascript

HTML:



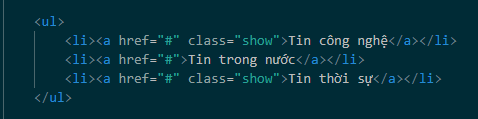
Gọi hàm:



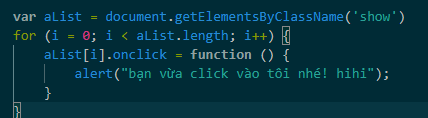
- Cách 3: Bắt sự kiện cho nhiều thẻ HTML bằng Javascript

Thêm sự kiện khi click vào tất cả các thẻ a có class="show" thì thông báo lên câu chào.

HTML:



JS:



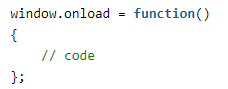
Bảng các sự kiện thường dùng trong javascript

|  |  |  |
| --- | --- | --- |
| **Tên events** | **Mô tả** | **Ví dụ** |
| onclick | Xảy ra khi click vào thẻ HTML | <button onclick="myFunction()">Click me</button>  <p id="demo"></p>  function myFunction() { document.getElementById("demo").innerHTML = "Hello World";  } |
| ondbclick | Xảy ra khi double click vào thẻ HTML | <p ondblclick="myFunction()">Double-click this paragraph to trigger a function.</p>  <p id="demo"></p>  <script>  function myFunction() { document.getElementById("demo").innerHTML = "Hello World";  }  </script> |
| onchange | Xảy ra khi giá trị (value) của thẻ HTML đổi. Thường dùng trong các đối thẻ form input |  |
| onmouseover | Xảy ra khi con trỏ chuột bắt đầu đi vào thẻ HTML |  |
| onmouseout | Xảy ra khi con trỏ chuột bắt đầu rời khỏi thẻ HTML |  |
| onmouseenter | Tương tự như onmouseover |  |
| onmouseleave | Tương tự như onmouseout |  |
| onmousemove | Xảy ra khi con chuột di chuyển bên trong thẻ HTML |  |
| onkeydown | Xảy ra khi gõ một phím bất kì vào ô input |  |
| onload | Xảy ra khi thẻ HTML bắt đầu khởi chạy |  |
| onkeyup | Xảy ra khi bạn gõ phím nhưng **lúc bạn nhã phím ra sẽ được kích hoạt** |  |
| onkeypress | Xảy ra khi bạn nhấn môt phím vào ô input |  |
| onblur | Xảy ra khi con trỏ chuột rời khỏi ô input |  |
| oncopy | Xảy ra khi bạn copy nội dung của thẻ |  |
| oncut | Xảy ra khi bạn cắt nội dung của thẻ |  |
| onpaste | Xảy ra khi bạn dán nội dung vào thẻ |  |

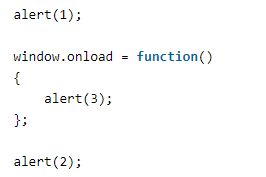
6.1. Sự kiện window.onload trong Javascript

Sự kiện window.onload có ý nghĩa rằng khi trình duyệt đã load xong mọi thứ (image, js, css) thì những đoạn code nằm bên trong đó mới được chạy. Bạn có thể sử dụng sự kiện này với các thẻ HTML liên quan đến việc tải dữ liệu như thẻ **img** hoặc thẻ **iframe**.

Cú pháp:



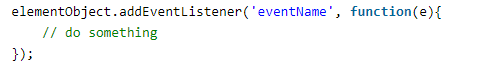
Những đoạn code nằm bên trong sự kiện onload sẽ được **chạy sau cùng**, khi mà trình biên dịch javascript đã load đủ thông tin về dom, biến, hàm .... Chính vì vậy, nếu trong sự kiện onload bạn gọi tới một hàm nào đó thì sẽ không cần quan tâm đến việc hàm đó được đặt tại vị trí nào.



6.2. addEventListener

addEventListener là một phương thức được tích hợp sẵn vào các đối tượng HTML thông qua cơ chế DOM. Khi sử dụng addEventListener thì bạn có thể bổ sung rất nhiều hành động vào sự kiện tại nhiều thời điểm khác nhau.

Cú pháp:



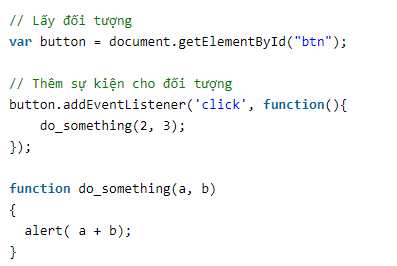
**Trong đó**:

**eventName** là tên của sự kiện bỏ đi chữ on, ví dụ click, change, ...

function ở tham số thứ hai chính là hàm sẽ được chạy khi sự kiện **eventName** được kích hoạt

-Truyền tham số vào sự kiện trong hàm addEventListener

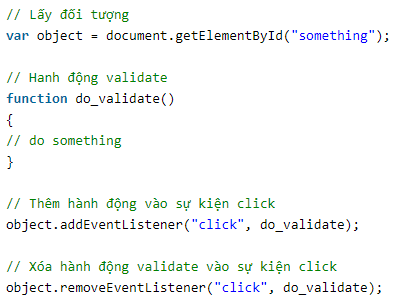
Nếu bạn muốn truyền tham số vào thì bắt buộc bạn phải tạo một hàm khác rồi gọi nó từ hàm addEventListener().



6.3. removeEventListener trong Javascript: hàm giúp bạn xóa hành động ra khỏi một sự kiện nào đó.

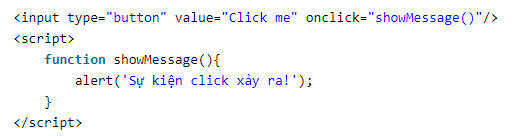
Cú pháp: object.removeEventListener("click", some\_action);

Sự kiện này cần được đặt trong một hàm để nhận diện

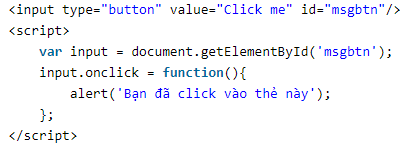


6.4. Sự kiện click: Ta cần bắt sự kiện click khi muốn xây dựng một chức năng nào đó, và chức năng này sẽ xảy ra khi người dùng click vào một đối tượng nào đó.

Có hai cách để bắt sự kiện click trong js:

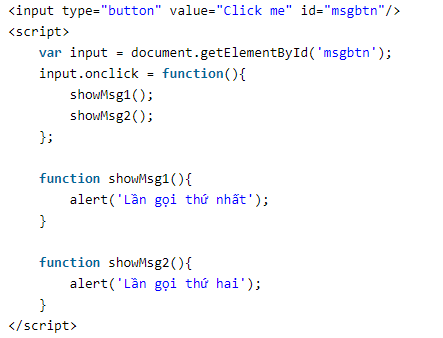
-Cách 1: Gắn trực tiếp trên thẻ HTML

- Cách 2: Gắn sự kiện click vào đối tượng DOM



Gán nhiều hành động cho sự kiện click cùng lúc

Nếu bạn muốn gọi cả hai hàm tách biệt thì tốt nhất định nghĩa hai hàm riêng, sau đó gắn chung một lần vào sự kiện click.

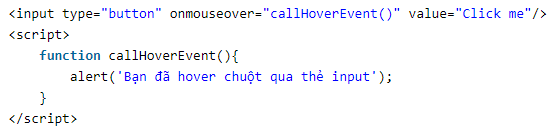


6.5. Sự kiện dbclick: tương tự sự kiện click

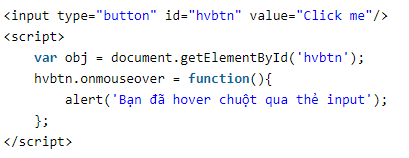
6.6. Sự kiện hover: Sự kiện hover tức là khi bạn hover chuột vào một thẻ html nào đó thì ngay lập tức sẽ xảy ra sự kiện hover.

- Có 2 cách gọi sự kiện hover:

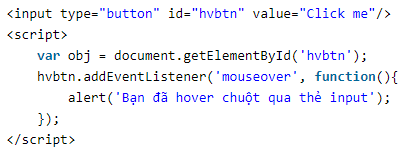
+ Gọi trực tiếp trên thẻ HTML:



+ Sử dụng DOM trong JS để gọi:



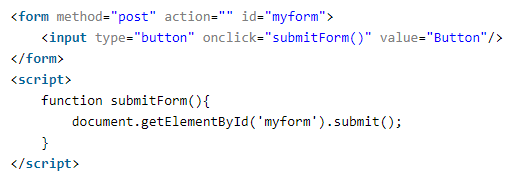
Hoặc sử dụng hàm addEventListener:



6.7. Sự kiện onmouseout, mousemove cách dùng tương tự sự kiện hover

6.8. Sự kiện onsubmit

khi click vào button đó thì gọi đến phương thức **submit()** của form để kích hoạt sự kiện



**7. DOM:** là viết tắt của chữ **D**ocument **O**bject **M**odel, dịch tạm ra là mô hình các đối tượng trong tài liệu HTML. Thông qua mô hình DOM ta có thể truy xuất đến các thẻ HTML một cách dễ dàng.

DOM có thể được: thay đổi, thêm, xóa các phần tử, thuộc tính HTML. Thay đổi CSS trang. Tương tác, tạo sự kiện với HTML trong trang

7.1. Các loại DOM

DOM document: có nhiệm vụ lưu trữ toàn bộ các thành phần trong tài liệu của website

[DOM element](https://freetuts.net/dom-element-trong-javascript-367.html): có nhiệm vụ truy xuất tới thẻ HTML nào đó thông qua các thuộc tính như tên class, id, name của thẻ HTML

[DOM HTML](https://freetuts.net/dom-html-trong-javascript-370.html): có nhiệm vụ thay đổi giá trị nội dung và giá trị thuộc tính của các thẻ HTML

[DOM CSS](https://freetuts.net/dom-css-trong-javascript-371.html): có nhiệm vụ thay đổi các định dạng CSS của thẻ HTML

[DOM Event](https://freetuts.net/su-kien-event-trong-javascript-368.html): có nhiệm vụ gán các sự kiện như onclick(), onload() vào các thẻ HTML

DOM Listener: có nhiệm vụ lắng nghe các sự kiện tác động lên thẻ HTML đó

DOM Navigation dùng để quản lý, thao tác với các thẻ HTML, thể hiện mối quan hệ cha - con của các thẻ HTML

[DOM Node](https://freetuts.net/dom-nodes-trong-javascript-383.html), Nodelist: có nhiệm vụ thao tác với HTML thông qua đối tượng (Object)

7.2. Đối tượng DOM document trong javascript

+ document.getElementById: Phương thức này giúp ta truy xuất đến thẻ HTML thông qua ID. Nếu trong tài liệu html của bạn bị trung ID thì nó sẽ lấy thẻ html có ID trùng khớp đầu tiên.

+ document.getElementsByTagName

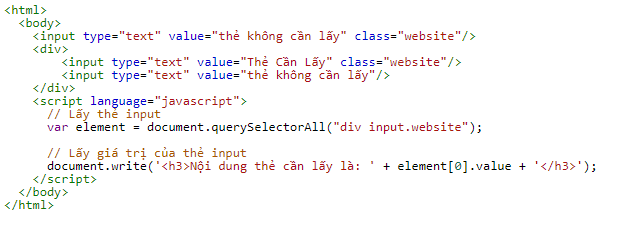
Phương thức này lấy tất cả các thẻ html có tên trùng với tham số truyền vào. Ví dụ dưới đây là mình lấy tất cả thẻ **div**.

+ document.getElementsByClassName

Phương thức này lấy tất cả thẻ html có tên class trùng với tên của tham số truyền vào.

+Sử dụng querySelectorAll để tìm theo CSS selector

Khi chọn các thẻ HTML theo class thì thường sẽ trả về hàng loạt các kết quả nên đôi khi sẽ có những kết quả mà ta không muốn lấy. Chính vì vậy javascript DOM có một phương thức kết hợp với [CSS Selector](https://freetuts.net/selector-la-gi-tim-hieu-css-selector-can-ban-345.html) để truy vấn có độ chính xác cao hơn, đó chính là hàm **querySelectorAll**



7.3. DOM HTML trong Javascript, lấy nội dung trong thẻ html

- Lấy nội dung trong thẻ HTML bằng Javascript



Trong đó **innerHTML** chính là thuộc tính của đối tượng html, công dụng của nó là dùng để lấy hoặc gán giá trị cho một thẻ html.



-Thay đổi thuộc tính thẻ html bằng Javascript

+ Để thay đổi giá trị của thuộc tính HTML thì ta sử dụng cú pháp như sau:



**+** Để lấy giá trị của thuộc tính HTML ta sử dụng cú pháp sau:



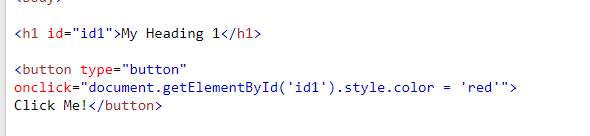
Trong đó **attributeName** là tên của thuộc tính mà bạn cần xử lý

7.4. DOM CSS: Thay đổi CSS bằng Javascript

Cú pháp:



Vd:



7.5. DOM Nodes trong Javascript

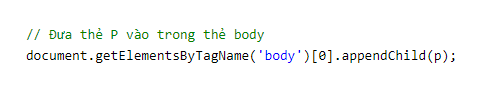
- DOM Node - document.createElement()

Tạo phần tử HTML mới vào HTML DOM, sử dụng phương thức document.createElement() với tham số truyền vào là tên của thẻ HTML cần tạo



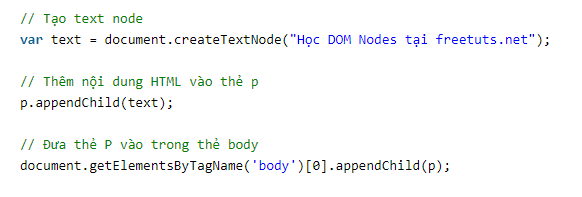


Để thêm Node này vào trang web thì chúng ta sử dụng phương thức appendChild :



-DOM Node - document.createTextNode():

Text node là một node đặc biệt, nó không phải là một thẻ HTML thông thường mà chỉ là một chuỗi (string) nên thông thường chúng ta sử dụng nó để thay thế cách gán thông thường node.innerHTML



- DOM Node - các phương thức khác

+Phương thức appendChild():Dùng để thêm (**bổ sung**) vào vị trí cuối cùng của đối tượng một thẻ HTML nào đó.

+ Phương thức insertBefore():Được dùng để thêm một Node vào đằng trước một node con nào đó. Phương thức này có hai tham số truyền vào insertBefore(node\_insert, node\_child) , trong đó:

node\_insert là node bạn muốn thêm vào

node\_child là node con mà bạn muốn thêm vào đằng trước nó.

+ Phương thức removeChild(): Được dùng để xóa một node con ra khỏi node hiện tại.

+ Phương thức replaceChild(): Dùng để replace (thay thế) một node con nào đó bằng một node khác mới hoàn toàn

**8. Các hàm sẵn có trong javascript**

- Number

|  |  |
| --- | --- |
| **Phương thức** | **Miêu tả** |
| [constructor()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/thuoc-tinh-number-constructor-trong-javascript) | Trả về hàm mà tạo sự instance của đối tượng này. Theo mặc định, nó là đối tượng Number. |
| [toExponential()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/number-toexponential-trong-javascript) | Làm một số hiển thị ở dạng số mũ, ngay cả khi số này là trong một dãy trong đó JavaScript thường sử dụng ký hiệu chuẩn. |
| [toFixed()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/number-tofixed-trong-javascript) | Định dạng một số với một số chữ số đặc trưng tới bên phải của phần thập phân. |
| [toLocaleString()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/number-tolocalestring-trong-javascript) | Trả về một phiên bản giá trị chuỗi của số hiện tại trong một định dạng mà có thể đa dạng theo thiết lập nội bộ của trình duyệt. |
| [toPrecision()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/number-toprecision-trong-javascript) | Định nghĩa bao nhiêu chữ số (bao gồm các chữ số ở bên phải và bên trái phần thập phân) để hiển thị một số. |
| [toString()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/number-tostring-trong-javascript) | Trả về sự biểu diễn chuỗi của giá trị số. |
| [valueOf()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/number-valueof-trong-javascript) | Trả về giá trị số của số. |

- String

|  |  |
| --- | --- |
| **Phương thức** | **Miêu tả** |
| [charAt()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-charat-trong-javascript) | Trả về ký tự tại chỉ mục (index) đã xác định. |
| [charCodeAt()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-charcodeat-trong-javascript) | Trả về một số chỉ dẫn giá trị Unicode của ký tự tại chỉ mục đã cho |
| [concat()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-concat-trong-javascript) | so khớp văn bản của hai chuỗi và trả về một chuỗi mới. |
| [indexOf()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-indexof-trong-javascript) | Trả về chỉ mục trong đối tượng String đang gọi của sự xảy ra đầu tiên của giá trị đã xác định, hoặc -1 nếu không tìm thấy. |
| [lastIndexOf()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-lastindexof-trong-javascript%22) | Trả về chỉ mục trong đối tượng String đang gọi của sự xảy ra cuối cùng của giá trị đã xác định, hoặc -1 nếu không tìm thấy. |
| [localeCompare()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-localecompare-trong-javascript) | Trả về một số chỉ dẫn có hay không một chuỗi tham chiếu đến trước hoặc sau hoặc là giống với chuỗi đã cho trong thứ tự phân loại |
| [length()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/thuoc-tinh-string-length-trong-javascript) | Trả về độ dài của chuỗi |
| [match()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-match-trong-javascript) | Được sử dụng để so khớp một Regular Expression với một chuỗi. |
| [replace()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-replace-trong-javascript) | Được sử dụng để tìm một so khớp giữa Regular Expression đã cho và một chuỗi, và để thay thế chuỗi phụ đã so khớp với một chuỗi phụ mới. |
| [search()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-search-trong-javascript) | Thực thi việc tìm kiếm cho một match giữa một Regular Expression và một chuỗi đã xác định. |
| [slice()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-slice-trong-javascript) | Trích (extract) một khu vực của một chuỗi và trả về một chuỗi mới. |
| [split()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/cat-chuoi-split-trong-javascript) | Chia một đối tượng String thành một mảng của các chuỗi bằng việc phân chia chuỗi thành các chuỗi phụ |
| [substr()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-substr-trong-javascript) | Trả về các ký tự trong một chuỗi bắt đầu tại vị trí đã xác định với một số ký tự đã xác định. |
| [substring()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-substring-trong-javascript) | Trả về các ký tự trong một chuỗi giữa hai chỉ mục (index) bên trong chuỗi đó. |
| [toLocaleLowerCase()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-tolocalelowercase-trong-javascript) | Các ký tự bên trong một chuỗi được biến đổi thành kiểu chữ thường đồng thời tôn trọng Locale hiện tại. |
| [toLocaleUpperCase()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-tolocaleuppercase-trong-javascript) | Các ký tự bên trong một chuỗi được biến đổi thành kiểu chữ hoa đồng thời tôn trọng Locale hiện tại. |
| [toLowerCase()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-tolowercase-trong-javascript) | Trả về giá trị chuỗi đang gọi được biến đổi thành kiểu chữ thường |
| [toString()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-tostring-trong-javascript-pkpig) | Trả về một chuỗi biểu diễn đối tượng đã xác định |
| [toUpperCase()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-touppercase-trong-javascript) | Trả về giá trị chuỗi đang gọi được biến đổi thành kiểu chữ hoa |
| [valueOf()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-valueof-trong-javascript) | Trả về giá trị gốc của đối tượng đã xác định. |

- String HTML wrapper

|  |  |
| --- | --- |
| **Phương thức** | **Miêu tả** |
| [anchor()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-anchor-trong-javascript) | Tạo một HTML Anchor mà được sử dụng như là một mục tiêu siêu văn bản (hypertext target). |
| [big()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-big-trong-javascript) | Tạo một chuỗi để được hiển thị trong một big font như thể nó trong một thẻ |
| [blink()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-blink-trong-javascript) | Tạo một chuỗi nhấp nháy như thể nó trong một thẻ |
| [bold()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-bold-trong-javascript) | Tạo một chuỗi để được hiển thị ở dạng in đậm như thể nó trong một thẻ |
| [fixed()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-fixed-trong-javascript) | Tạo một chuỗi để được hiển thị trong một font cố định như thể nó trong một thẻ |
| [fontcolor()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-fontcolor-trong-javascript) | Tạo một chuỗi để được hiển thị theo một màu cố định như khi nó trong một thẻ |
| [fontsize()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/phuong-thuc-string-fontsize-trong-javascript) | Làm một chuỗi được hiển thị trong một kích cỡ font xác định như khi nó trong một thẻ |
| [italics()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-italics-trong-javascript) | Làm một chuỗi ở dạng in nghiêng như khi nó trong một thẻ |
| [link()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-link-trong-javascript) | Tạo một HTML link siêu văn bản mà yêu cầu URL khác |
| [small()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-small-trong-javascript) | Làm một chuỗi hiển thị ở dạng Small font như khi nó trong một thẻ |
| [strike()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-strike-trong-javascript) | Làm một chuỗi hiển thị ở dạng văn bản gạch ngang như khi nó trong một thẻ |
| [sub()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-sub-trong-javascript) | Làm một chuỗi hiển thị ở dạng chỉ số dưới như khi nó trong một thẻ |
| [sup()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/string-sup-trong-javascript) | Làm một chuỗi hiển thị ở dạng chỉ số trên như khi nó trong một thẻ |

- Boolean

|  |  |
| --- | --- |
| **Phương thức** | **Miêu tả** |
| [toSource()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/boolean-tosource-trong-javascript) | Trả về một chuỗi chứa nguồn của đối tượng Boolean; bạn có thể sử dụng chuỗi này để tạo một đối tượng tương đương. |
| [toString()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/boolean-tostring-javascript) | Trả về một chuỗi hoặc "true" hoặc "false" phụ thuộc vào giá trị của đối tượng. |
| [valueOf()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/boolean-valueof-trong-javascript) | Trả về giá trị gốc của đối tượng Boolean. |

- Array

|  |  |
| --- | --- |
| **Phương thức** | **Miêu tả** |
| [concat()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/array-concat-trong-javascript) | Trả về một mảng mới bao gồm mảng này kết hợp với các mảng khác và/hoặc giá trị khác. |
| [every()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/array-every-trong-javascript) | Trả về true nếu mỗi phần tử trong mảng này thỏa mãn hàm kiểm tra đã cho. |
| [filter()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/array-filter-trong-javascript) | Tạo một mảng mới với tất cả các phần tử của mảng này, mà hàm lọc đã cho trả về true. |
| [forEach()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/array-foreach-trong-javascript) | Gọi một hàm cho mỗi phần tử trong mảng. |
| [indexOf()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/array-indexof-trong-javascript) | Trả về chỉ mục đầu tiên (thấp nhất) của một phần tử trong mảng tương đương với giá trị đã cho, hoặc -1 nếu không được tìm thấy. |
| [join()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/array-join-trong-javascript) | Kết hợp tất cả phần tử trong một mảng vào trong một chuỗi. |
| lastIndexOf() | Trả về chỉ mục cuối cùng (lớn nhất) của một phần tử trong mảng tương đương với giá trị đã cho, hoặc -1 nếu không được tìm thấy. |
| [map()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/array-map-trong-javascript) | Tạo một mảng mới với các kết quả của việc gọi một hàm đã cho trên mỗi phần tử của mảng này. |
| [pop()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/array-pop-trong-javascript) | Gỡ bỏ phần tử cuối cùng từ một mảng và trả về phần tử đó. |
| [push()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/array-push-trong-javascript) | Thêm một hoặc nhiều phần tử tới phần cuối của một mảng và trả về độ dài mới của mảng. |
| [reduce()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/array-reduce-trong-javascript) | Áp dụng một hàm đồng thời với hai giá trị của mảng (từ trái qua phải) khi để giảm nó tới một giá trị đơn. |
| [reduceRight()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/array-reduceright-trong-javascript) | Áp dụng một hàm đồng thời với hai giá trị của mảng (từ phải qua trái) khi để giảm nó tới một giá trị đơn. |
| [reverse()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/array-reverse-trong-javascript) | Đảo ngược thứ tự của các phần tử của một mảng – Đầu tiên trở thành cuối cùng và cuối cùng trở thành đầu tiên. |
| [shift()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/array-shift-trong-javascript) | Gỡ bỏ phần tử đầu tiên từ một mảng và trả về phần tử đó. |
| [slice()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/array-slice-trong-javascript) | Extract – trích một khu vực của một mảng và trả về một mảng mới. |
| [some()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/array-some-trong-javascript) | Trả về true nếu có ít nhất một phần tử trong mảng này thỏa mãn hàm kiểm tra đã cho. |
| [toSource()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/array-tosource-trong-javascript) | Biểu diễn code nguồn của một đối tượng. |
| [sort()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/array-sort-trong-javascript) | Sắp xếp phân loại các phần tử của một mảng. |
| [splice()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/array-splice-trong-javascript) | Thêm và/hoặc gỡ bỏ các phần tử từ một mảng. |
| [toString()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/array-tostring-trong-javascript) | Trả về một chuỗi biểu diễn mảng đó và các phần tử của nó. |
| [unshift()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/array-unshift-trong-javascript) | Thêm một hoặc nhiều phần tử tới phần đầu của một mảng và trả về độ dài mới của mảng. |

- Date time

|  |  |
| --- | --- |
| **Phương thức** | **Miêu tả** |
| [Date()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-trong-javascript) | Trả về ngày và thời gian của ngày hôm nay. |
| [getDate()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-getdate-trong-javascript) | Trả về ngày trong tháng cho ngày xác định theo Local time |
| [getDay()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-getday-trong-javascript) | Trả về ngày trong tuần cho ngày xác định theo Local time |
| [getFullYear()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-getfullyear-trong-javascript) | Trả về năm của ngày đã cho theo Local time |
| [getHours()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-gethours-trong-javascript) | Trả về giờ của ngày đã cho theo Local time |
| [getMilliseconds()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-getmilliseconds-trong-javascript) | Trả về mili giây của ngày đã cho theo Local time |
| [getMinutes()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-getminutes-trong-javascript) | Trả về phút của ngày đã cho theo Local time |
| [getMonth()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-getmonth-trong-javascript) | Trả về tháng của ngày đã cho theo Local time |
| [getSeconds()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-getseconds-trong-javascript) | Trả về giây của ngày đã cho theo Local time |
| [getTime()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-gettime-trong-javascript) | Trả về giá trị số của ngày đã cho khi số mili giây từ tháng 1/1/1970, 00:00:00 UTC. |
| [getTimezoneOffset()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-gettimezoneoffset-trong-javascript) | Trả về Time-zone Offset bằng phút cho Locale hiện tại. |
| [getUTCDate()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-getutcdate-trong-javascript) | Trả về ngày trong tháng của ngày đã cho theo Universal time |
| [getUTCDay()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-getutcday-trong-javascript) | Trả về ngày trong tuần của ngày đã cho theo Universal time |
| [getUTCFullYear()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-getutcfullyear-trong-javascript) | Trả về năm của ngày đã cho theo Universal time |
| [getUTCHours()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-getutchours-trong-javascript) | Trả về giờ của ngày đã cho theo Universal time |
| [getUTCMilliseconds()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-getutcmilliseconds-trong-javascript) | Trả về mili giây của ngày đã cho theo Universal time |
| [getUTCMinutes()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-getutcminutes-trong-javascript) | Trả về phút của ngày đã cho theo Universal time |
| [getUTCMonth()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-getutcmonth-trong-javascript) | Trả về tháng của ngày đã cho theo Universal time |
| [getUTCSeconds()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-getutcseconds-trong-javascript) | Trả về giây của ngày đã cho theo Universal time |
| [getYear()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-getyear-trong-javascript) | **Phương thức cũ** - Trả về năm của ngày đã cho theo Local time. Bạn sử dụng getFullYear để thay thế. |
| [setDate()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-setdate-trong-javascript) | Trả về ngày của tháng cho ngày đã xác định theo Local time. |
| [setFullYear()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-setfullyear-trong-javascript) | Thiết lập năm đầy đủ cho ngày đã cho theo Local time. |
| [setHours()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-sethours-trong-javascript) | Thiết lập giờ cho ngày đã cho theo Local time. |
| [setMilliseconds()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-setmilliseconds-trong-javascript) | Thiết lập mili giây cho ngày đã cho theo Local time. |
| [setMinutes()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-setminutes-trong-javascript) | Thiết lập phút cho ngày đã cho theo Local time. |
| [setMonth()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-setmonth-trong-javascript) | Thiết lập tháng cho ngày đã cho theo Local time. |
| [setSeconds()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-setseconds-trong-javascript) | Thiết lập giây cho ngày đã cho theo Local time. |
| [setTime()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-settime-trong-javascript) | Thiết lập thời gian đối tượng Date được biểu diễn bởi số mili giây từ 1/1/1970, 00:00:00 UTC. |
| [setUTCDate()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-setutcdate-trong-javascript) | Thiết lập ngày của tháng cho ngày đã cho theo Universal time |
| [setUTCFullYear()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-setutcfullyear-trong-javascript) | Thiết lập năm đầy đủ cho ngày đã cho theo Universal time |
| [setUTCHours()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-setutchours-trong-javascript) | Thiết lập giờ cho ngày đã cho theo Universal time |
| [setUTCMilliseconds()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-setutcmilliseconds-trong-javascript) | Thiết lập mili giây cho ngày đã cho theo Universal time |
| [setUTCMinutes()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-setutcminutes-trong-javascript) | Thiết lập phút cho ngày đã cho theo Universal time |
| [setUTCMonth()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-setutc-month-trong-javascript) | Thiết lập tháng cho ngày đã cho theo Universal time |
| [setUTCSeconds()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-setutcseconds-trong-javascript-fnzbe) | Thiết lập giây cho ngày đã cho theo Universal time |
| [setYear()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-setyear-trong-javascript-oqo0l) | **Phương thức cũ -**Thiết lập năm cho ngày đã cho theo Local time. Bạn dùng setFullYear để thay thế. |
| [toDateString()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-todatestring-trong-javascript-bzuj3) | Trả về một chuỗi ngày mà con người đọc được. |
| [toGMTString()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-togmtstring-trong-javascript) | **Phương thức cũ -**Biến đổi 1 ngày thành 1 chuỗi bởi sử dụng các qui ước Internet GMT. Bạn dùng toUTCString để thay thế. |
| [toLocaleDateString()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-tolocaledatestring-trong-javascript) | Trả về ngày dưới dạng chuỗi, sử dụng qui ước của Locale hiện tại |
| [toLocaleFormat()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-tolocaleformat-trong-javascript) | Biến đổi ngày thành chuỗi, sử dụng chuỗi định dạng. |
| [toLocaleString()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-tolocalestring-trong-javascript) | Biến đổi ngày thành chuỗi, sử dụng các qui ước của Locale hiện tại. |
| [toLocaleTimeString()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-tolocaletimestring-trong-javascript) | Trả về thời gian của một ngày ở dạng chuỗi, sử dụng qui ước của Locale hiện tại. |
| [toSource()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-tosource-trong-javascript) | Trả về một chuỗi biểu diễn nguồn cho một đối tượng Date tương đương, bạn có thể sử dụng giá trị này để tạo một đối tượng mới. |
| [toString()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-tostring-trong-javascript) | Trả về một chuỗi biểu diễn đối tượng Date đã cho. |
| [toTimeString()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-totimestring-trong-javascript) | Trả về thời gian của đối tượng Date ở dạng chuỗi con người đọc được. |
| [toUTCString()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-toutcstring-trong-javascript) | Biến đổi một ngày thành một chuỗi, sử dụng qui ước Universal time. |
| [valueOf()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-valueof-trong-javascript) | Trả về giá trị ban đầu của một đối tượng Date. |

- Math

|  |  |
| --- | --- |
| **Phương thức** | **Miêu tả** |
| [abs()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/math-abs-trong-javascript-jzdqb) | Trả về trị tuyệt đối của một số |
| [acos()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/math-acos-trong-javascript) | Trả về arccos (giá trị radians) của một số |
| [asin()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/math-asin-trong-javascript) | Trả về arcsin (giá trị radians) của một số |
| [atan()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/math-atan-trong-javascript) | Trả về arctan (giá trị radians) của một số |
| [atan2()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/math-atan2-trong-javascript) | Trả về tang lượng giác ngược |
| [ceil()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/math-ceil-trong-javascript) | Trả về số integer nhỏ nhất lớn hơn hoặc bằng một số |
| [cos()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/math-cos-trong-javascript) | Trả về cos của một số. |
| [exp()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/math-exp-trong-javascript) | Trả về EN, với N là tham số, và E là hằng số Euler |
| [floor()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/math-floor-trong-javascript) | Trả về số integer lớn nhất nhỏ hơn hoặc bằng một số |
| [log()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/math-log-trong-javascript) | Trả về logarit cơ số e của một số |
| [max()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/math-max-trong-javascript) | Trả về số lớn nhất của 0 hoặc nhiều số |
| [min()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/math-min-trong-javascript) | Trả về số nhỏ nhất của 0 hoặc nhiều số |
| [pow()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/math-pow-trong-javascript) | Trả về giá trị (cơ số)(số mũ) |
| [random()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/math-random-trong-javascript) | Trả về một số ngẫu nhiên giữa hai số 0 và 1. |
| [round()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/math-round-trong-javascript) | Làm tròn số. Trả về giá trị của một số integer gần nhất. |
| [sin()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/math-sin-trong-javascript) | Trả về sin của một số |
| [sqrt()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/math-sqrt-trong-javascript) | Trả về căn bậc hai của một số |
| [tan()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/math-tan-trong-javascript-pppmu) | Trả về tan của một số |
| [toSource()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/math-tosource-trong-javascript) | Trả về chuỗi "Math" |

* RegExp

|  |  |
| --- | --- |
| **Phương thức** | **Miêu tả** |
| [exec()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/regexp-exec-trong-javascript) | Thực thi một tìm kiếm cho một so khớp trong tham số chuỗi của nó. |
| [test()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/regexp-test-trong-javascript) | Kiểm tra một so khớp trong tham số chuỗi của nó. |
| [toSource()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/regexp-tosource-trong-javascript) | Trả về một literal đối tượng biểu diễn đối tượng đã cho; bạn có thể sử dụng giá trị này để tạo đối tượng mới. |
| [toString()](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/regexp-tostring-trong-javascript) | Trả về một chuỗi biểu diễn đối tượng đã cho. |

CÁC KIẾN THỨC ĐÃ ÁP DỤNG ĐƯỢC