**Tổng hợp kiến thức JS Cơ bản**

1. **Biến và hằng**

**Khai báo biến**

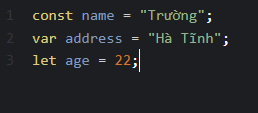
***const*** được sử dụng để khai báo 1 hằng số, và giá trị của nó không thay đổi trong suốt chương trình.

***let*** khai báo biến chỉ có thể truy cập được trong block bao quanh nó được xác định bằng cặp {}.

***var*** khai báo biến có thể truy cập ở phạm vi hàm số hoặc bên ngoài hàm số, toàn cục.  
  
Giống như ngôn ngữ khác, cách đặt tên biến của JS cũng phải tuân theo 1 số quy tắc sau :

* Tên biến phải là các chữ không dấu viết hoa hoặc viết thường, các chữ số từ 0-9 và dấu gạch dưới () và kí hiệu $.
* Tên biến bắt đầu phải là chữ hoặc dấu gạch dưới (\_), nếu bắt đầu bằng số là sai.
* Không thể sử dụng các từ dành riêng (như từ khóa JavaScript) làm tên.
* Các tên có phân biệt chữ hoa chữ thường.

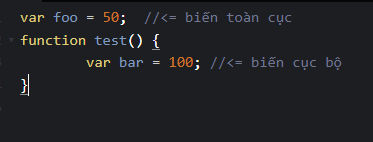
Các ví dụ về khai báo biến:



**Phạm vi (scope)**

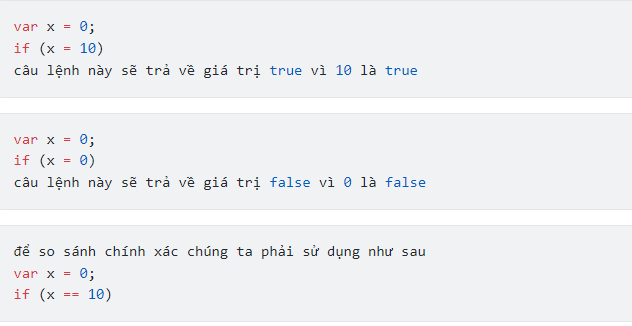
Có hai loại phạm vi trong JavaScript. biến cục bộ (local scope) và biến toàn cục (global scope). Biến toàn cục được khai báo ngoài hàm , trong khi đó biến cục bộ được lưu khai báo bên trong một hàm.(Nên với cùng một tên biến, ta có thể khai báo ở những hàm khác nhau).  
  
Một biến toàn cục có thể được truy cập từ bên ngoài của hàm cũng như bên trong của hàm nhưng, một biến cục bộ chỉ có thể được truy cập bên trong hàm. Nếu truy cập từ bên ngoài của hàm, bạn sẽ nhận được lỗi 'undefined variable' (biến không xác định).

Ví dụ:

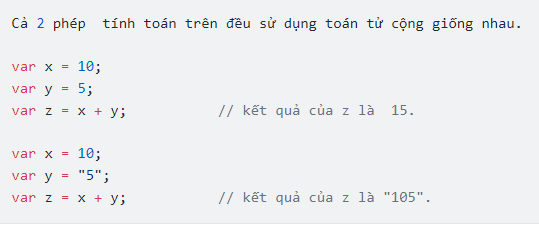


**Note: Một số lưu ý khi làm việc với biến:**

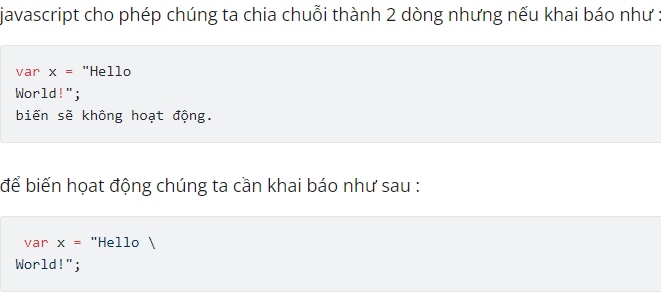
* ***Sử dụng toán tử gán (=), thay vì toán tử so sánh (==) trong câu lệnh if***

****

* **Nhầm lẫn giữa việc bổ sung và thêm số**

****

* **Phá vỡ chuỗi JavaScript**

****

1. **Kiểu dữ liệu ( Data types )**

* Nguyên thủy: Primitive types

Biến được lưu tại vùng nhớ stack, giá trị là các kiểu dữ liệu cơ sở.

* Number
* Boolean
* String
* Null
* Undefined
* Tham chiếu: Non-primitive types (bao gồm các dữ liệu còn lại như object, function, …)

Biến được lưu tại vùng nhớ stack, giá trị là địa chỉ của một đối tượng được lưu tại vùng nhớ heap.

1. **Toán tử (Operators).**

* **Toán học**

****

* **Gán**

Dùng để gán một giá trị đến một biến. Giá trị này có thể là hằng (như 6 hay “hello”), giá trị của biến khác, kết quả của một biểu thức hay hàm.

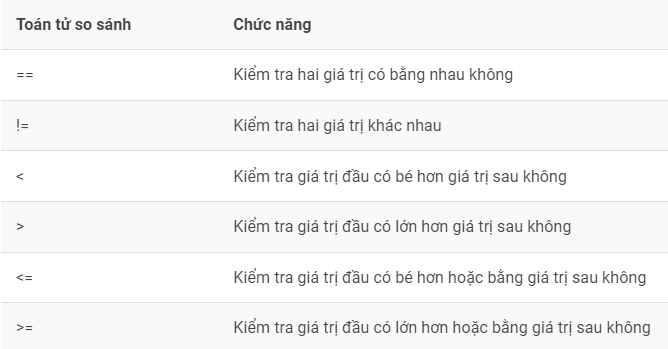
Kí hiệu: **=**

Ví dụ:



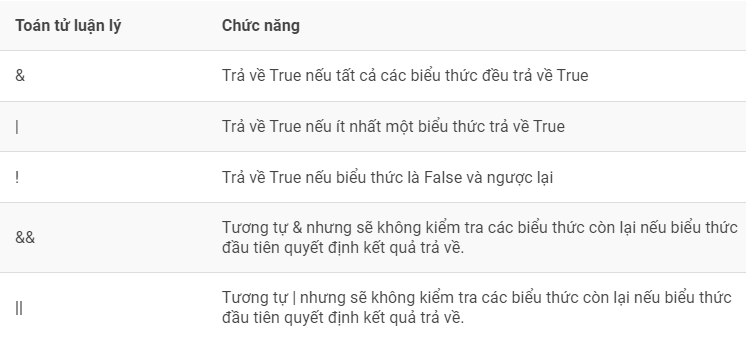
* **Quan hệ, so sánh**

Cho phép so sánh hai giá trị, thường dùng trong các biểu thức điều kiện. Kết quả trả về là **true** hay **false**. Một số toán tử so sánh phổ biến:

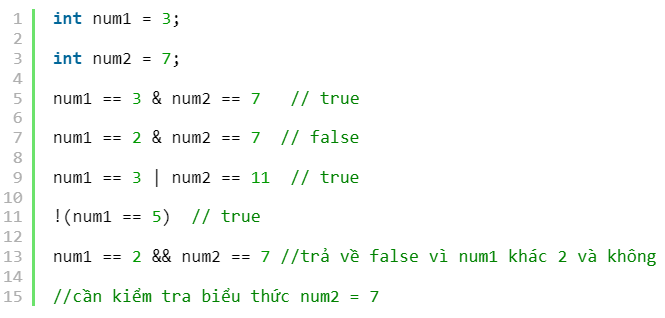
****

* **Logic**

Trả về giá trị **true** hay **false** từ việc kết hợp nhiều biểu thức. Một số toán tử.

****

Ví dụ:

****

* **Độ ưu tiên**

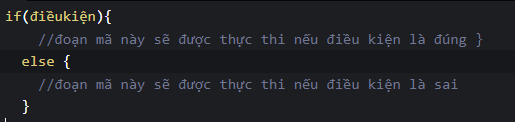
Độ ưu tiên của các toán tử trong biểu thức theo thứ tự giảm dần từ trên xuống như sau:

****

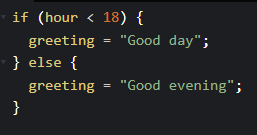
1. **Câu lệnh (Statements)**

* **Phân nhánh**
* **Điều kiện if … else**

Cú pháp:

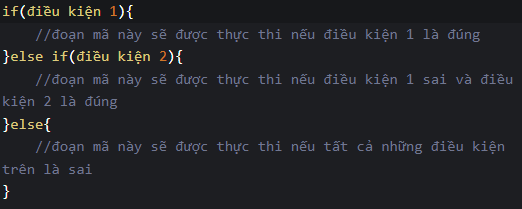


Ví dụ:

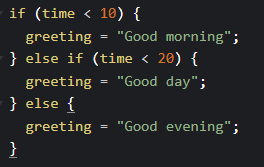


* **Điều kiện if … else if … else**

Cú pháp:

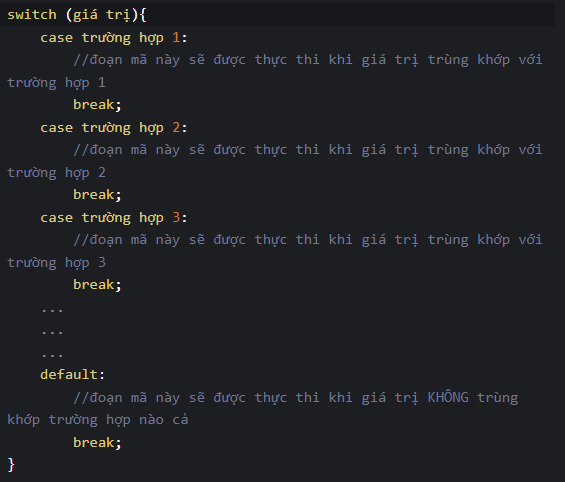


Ví dụ:

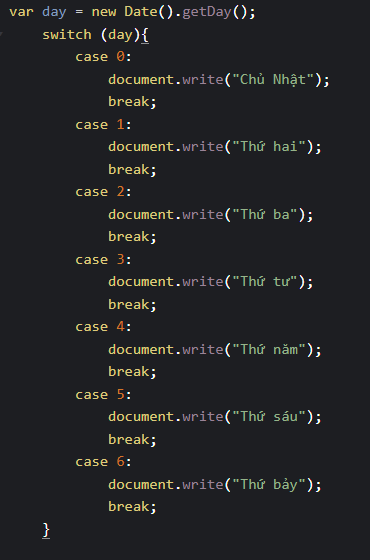


* **Switch case**

Cú pháp:

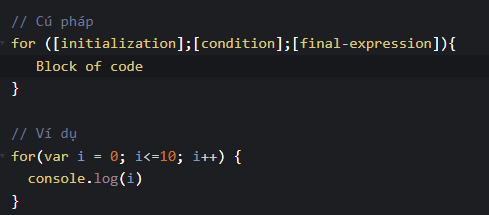


Ví dụ:



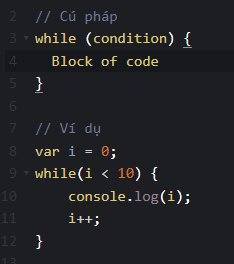
* **Vòng lặp**
* **For loop**

Với vòng lặp **for** ta sẽ khởi tạo biến đếm, kiểm tra điều kiện và tăng hoặc giảm biến được thực hiện trên cùng một dòng, do đó khá dễ dàng cho những người mới tiếp cận để debug và cũng giảm khả năng sinh ra lỗi



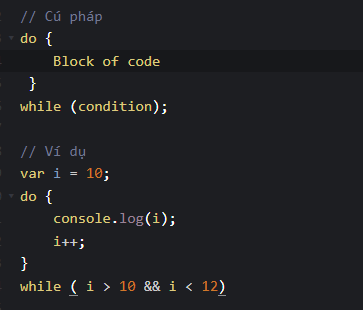
* **While loop**

Bên cạnh **for** thì **while** cũng là một trong những vòng lặp tương đối basic. Câu lệnh while tạo ra một vòng lặp thực thi một khối lệnh (block of code) cho đến khi điều kiện vẫn đúng.

****

* **Do … while**

\*\*do-while\*\* về cơ bản khá giống với \*\*while\*\*, chúng chỉ khác nhau duy nhất. Đối với \*\*Do While\*\* dù điều kiện lặp như thế nào thì đoạn code vẫn được chạy ít nhất 1 lần còn nếu điều kiện thỏa mãn thì sẽ tương tự như \*\*While\*\* : tạo ra thêm vòng lặp

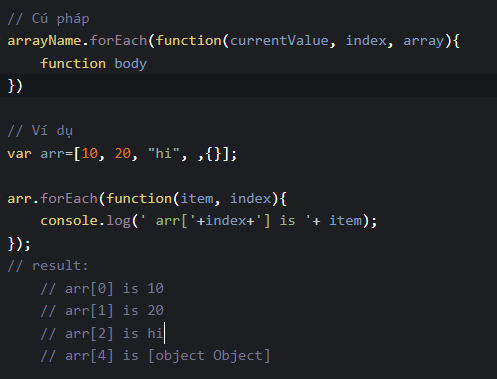


* **ForEach()**

**forEach** sẽ lặp lại từng phần tử trong mảng theo thứ tự index và thực thi **function** được truyền vào. Lưu ý rằng **forEach** sẽ không thực thi **function** đầu vào cho các phần tử không có giá trị.

Hàm được truyền vào **forEach** sẽ nhận tối đa 3 đối số đầu vào:

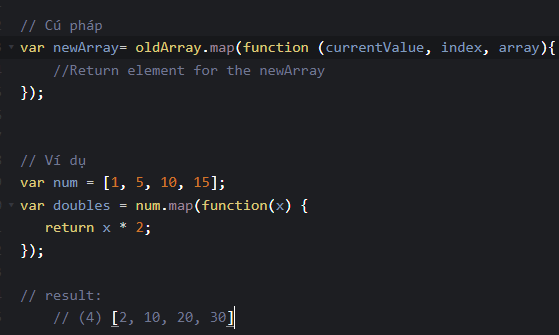
* currentValue: Giá trị đang được vòng lặp xử lý
* index: Chí số của giá trị (**currentValue**) trong mảng
* array: toàn bộ array đang gọi đến **forEach**



* **Map**

Tương tự như **forEach**, **map** cũng lấy 3 tham số đầu vào

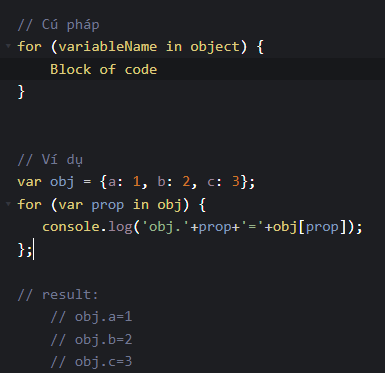
* currentValue: Giá trị đang được vòng lặp xử lý
* index: Chí số của giá trị (**currentValue**) trong mảng
* array: toàn bộ array đang gọi đến **forEach**

****

* **For … in**

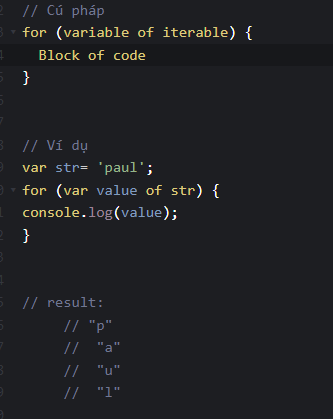
Vòng lặp này có đôi chút khác biệt với các hàm phía trên, **For ... in** mục đích chủ yếu được dùng để loop trong một **object** chứ không phải **array** . Số lượng vòng lặp sẽ tương ứng với số lượng thuộc tính của **object**

Mỗi **array** cũng là một object đặc biệt, do đó ta vẫn có thể sử đụng **for...in** cho **array**, tuy nhiên key sẽ tương ứng với giá trị **index** của từng phần tử



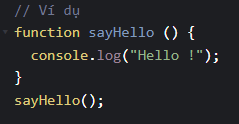
* **For … of**

Vòng lắp được ra mắt trong phiên bản ES6. Hàm này có thể sử dụng để duyệt phần lớn các đối tượng từ Array, String, Map, WeakMap, Set ,...

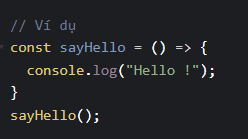
****

1. **Hàm ( Function )**

* Function (hàm, chức năng), gọi chung là subprogram (chương trình con) có thể được gọi ở bên ngoài hoặc bên trong chính nó.
* Nó bao gồm tập hợp các câu lệnh gọi là function body. Các giá trị có thể truyền đến một hàm, và một hàm có thể trả về giá trị. Bây giờ, với các ứng dụng hiện đại, các function có thể là một chương trình hoàn chỉnh.
* **Normal Function**
* Hàm là một chương trình con được thiết kế để thực hiện một công việc cụ thể.
* Chúng được thực thi khi gọi từ một hàm khác, sự kiện hoặc chính nó. Hàm trong Js có thể có không hoặc nhiều tham số



* **Arrow Function**
* Một arrow function có cú pháp ngắn hơn cú pháp function bình thường, nó có thể không có đối số, super hoặc new.target của nó. Những function này phù hợp nhất cho các non-method function và chúng không thể sử dụng như các constructor. Có điều gì khi tôi nói, function ở trên không trả về giá trị gì?
* Nếu chúng ta cố gắng để lưu trữ kết quả của function được gọi ở trên vào một biến nó sẽ nhận giá trị "undefined".



* **Callback Function**
* Callback phải là một function Callback là một function nên bạn nhất định phải truyền vào là một function, nếu bạn truyền một type khác thì bạn sẽ nhận được error notice: "Callback is function" trong console.
* Từ khóa this trong callback Như đã nói ở trên thì callback là một hàm bình thường nên khi sử dụng từ khóa this trong hàm thì nó sẽ hiểu this lúc này chính là đối tượng Window, nếu bạn dùng debuger trong hàm callback rồi vào console gõ this, thì sẽ được Window {external: Object, chrome: Object, result: undefined,...... Vì vậy cho dù bạn định nghĩa hàm callback nằm trong một object thì không thể truy cập đến dữ liệu của object thông qua từ khóa this.

1. **Sự kiện ( Event )**

Các Event trong JavaScript:

* Event Handler

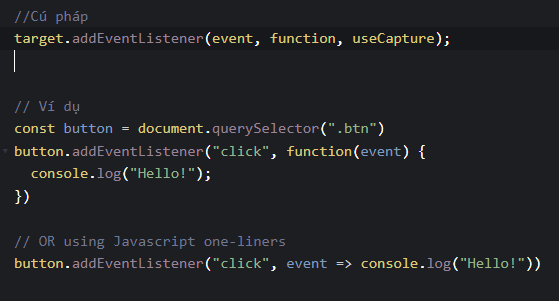
****

* Thêm Event Listener

Ngoài những thuộc tính EventHandler trên ra, chúng ta cũng có thể thêm có sự kiện vào đối tượng thông qua addEventListener.

Bằng cách thêm event listener vào một đối tượng, chúng ta có thể bắt được một loạt các sự kiện do người dùng hoặc trình duyệt tạo ra.

* target: Đây là phần tử bạn cần thêm Event Listeners vào.
* event: Là các loại sự kiện như click, mouseover, ...
* function: Tên hàm bạn cần thêm vào



* Khác biệt giữa Event Handlers và addEventListener

Nếu bạn sẽ sử dụng EventHandler, điểm khác biệt rõ ràng nhất chính là nếu bạn thêm nhiều sự kiện cho cùng một loại(ví dụ là onclick vào nút), thì sự kiện thứ 2 sẽ ghi đè lên sự kiện đầu tiên và chỉ sự kiện đó được thực thi.

1. **DOM**

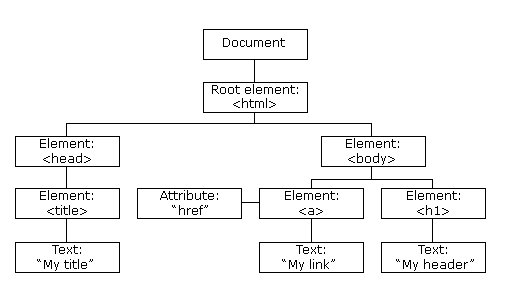
* **DOM là gì?**

DOM là viết tắt của Document Object Model, tạm dịch là mô hình các đối tượng trong tài liệu. DOM dùng để truy xuất và thao tác trên các tài liệu có cấu trúc dạng HTML hay XML bằng các ngôn ngữ lập trình thông dịch (scripting language) như JavaScript, PHP, Python.

Ngoài ra, DOM giúp thao tác dữ liệu theo mô hình hướng đối tượng. Các phần tử bên trong 1 tài liệu có cấu trúc được định nghĩa thành các đối tượng, phương thức và thuộc tính để có thể truy xuất dễ dàng mà vẫn đảm bảo tính cấu trúc: mỗi phần tử là một đối tượng, sở hữu các thuộc tính và các phương thức để làm việc với các thuộc tính đó như thêm, xóa, sửa, cập nhật. Bên cạnh đó, bạn cũng có thể thêm, bớt các phần tử tùy thích, giúp cho nội dung và cấu trúc của trang web luôn cập nhật động.

* **Cấu trúc DOM**

Đối với HTML DOM, mỗi thành phần đều được xem là 1 nút (node), được biểu diễn trên 1 cây cấu trúc dạng cây gọi là DOM Tree. Các phần tử khác nhau sẽ được phân loại nút khác nhau nhưng quan trọng nhất là 3 loại: nút gốc (document node), nút phần tử (element node), nút văn bản (text node).



* **Các loại DOM trong JavaScript**

Việc xử lý và làm việc với đối tượng HTML rất phức tạp và đa dạng. Chính vì vậy, JavaScript cung cấp cho chúng ta các phương thức, đối tượng và mỗi thành phần như vậy có những nhiệm vụ riêng biệt. Dưới đây là 8 loại DOM thường được sử dụng trong JavaScript:

* **DOM Document**: có nhiệm vụ lưu trữ toàn bộ các thành phần trong tài liệu của website.
* **DOM Element**: 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**: có nhiệm vụ thay đổi giá trị nội dung và giá trị thuộc tính của các thẻ HTML
* **DOM CSS**: có nhiệm vụ thay đổi các định dạng CSS của thẻ HTML. DOM Event: có nhiệm vụ gán các sự kiện như onclick(), onload() vào các thẻ HTML
* **DOM EventListener**: 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, Nodelist**: có nhiệm vụ thao tác với HTML thông qua đối tượng (Object).
* Đối tượng Number chỉ chứa các phương thức mặc định mà là một phần của sự định nghĩa của mỗi đối tượng.

|  |  |
| --- | --- |
| **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. |

### **Các phương thức của Boolean**

* Dưới đây là danh sách các phương thức của đối tượng Boolean và miêu tả về nó:

|  |  |
| --- | --- |
| **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. |

### **Các phương thức của đối tượng String**

* Dưới đây là danh sách miêu tả các phương thức của đối tượng 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**

* Dưới đây là danh sách các phương thức mà trả về một bản sao của chuỗi được gói bên trong một thẻ HTML thích hợp.

|  |  |
| --- | --- |
| **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 |

### **Các phương thức của đối tượng 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. |

### **Các phương thức của đối tượng Date**

* Bảng dưới liệt kê các phương thức của đối tượng Date và miêu tả về nó:

|  |  |
| --- | --- |
| **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. |

### **Các phương thức tĩnh (Static Method) của Date**

* Ngoài những phương thức được liệt kê ở trên, đối tượng Date cũng định nghĩa 2 phương thức tĩnh (Static Method). Những phương thức này được gọi thông qua chính Date() constructor.

|  |  |
| --- | --- |
| **Phương thức** | **Miêu tả** |
| [Date.parse( )](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-parse-trong-javascript) | Phân tích một biểu diễn chuỗi của một ngày và thời gian và trả về biểu diễn mili giây nội bộ của ngày đó. |
| [Date.UTC( )](https://hoclaptrinh.vn/tutorial/cac-ham-co-san-trong-javascript/date-utc-trong-javascript) | Trả về biểu diễn mili giây của ngày và thời gian UTC đã cho. |

### **Các phương thức của đối tượng Math**

* Bảng dưới liệt kê danh sách các phương thức của đối tượng Math và miêu tả về nó:

|  |  |
| --- | --- |
| **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" |

### **Các phương thức của đối tượng RegExp**

* Bảng dưới liệt kê các phương thức của đối tượng RegExp và miêu tả về nó:

|  |  |
| --- | --- |
| **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. |