# **Bài 01: Javascript là gì? Viêt ứng dụng javascript đầu tiên**

|  |
| --- |
| **Mục luc:**  [1. Javascript là gì?](http://freetuts.net/javascript-la-gi-viet-ung-dung-javascript-dau-tien-263.html#goto-h2-0)  [2. Các thư viện Javascript đình đám hiện nay](http://freetuts.net/javascript-la-gi-viet-ung-dung-javascript-dau-tien-263.html#goto-h2-1)  [3. Viết chương trình javascript đầu tiên](http://freetuts.net/javascript-la-gi-viet-ung-dung-javascript-dau-tien-263.html#goto-h2-2)  [4. Lời kết](http://freetuts.net/javascript-la-gi-viet-ung-dung-javascript-dau-tien-263.html#goto-h2-3) |

Trong loạt series này chúng ta sẽ tìm hiểu ngôn ngữ Javascript từ căn bản đến nâng cao nhằm giúp các bạn newbie hoặc những bạn đã có kiến thức Javascript rồi nhưng muốn học để bổ trợ thêm. Vì dạng viết text nên nội dung sẽ hơi khó diễn tả và khó trình bày nên các bạn cố gắng đọc từng chữ và thực hành theo demo nhé.

Trong bài này chúng ta sẽ tìm hiểu khái niệm Javascript là gì? ứng dụng của Javscript trong thực tế và viết một ứng dụng javascript đầu tay. Điều kiện tiên quyết để học ngôn ngữ này là  bạn phải biết một số ngôn ngữ khác  như PHP, C++, C nhé.

## **1. Javascript là gì?**

Javascript là một ngôn ngữ lập trình kịch bản dựa vào đối tượng phát triển có sẵn hoặc tự định nghĩa ra, javascript được sử dụng rộng rãi trong các ứng dụng Website. Javascript được hỗ trợ hầu như trên tất cả các trình duyệt như Firefox, Chrome, ... thậm chí các trình duyệt trên thiết bị di động cũng có hỗ trợ.

Nếu bạn đã biết qua về HTML5 thì bạn thấy có các khái niệm như sessionStore hay localStore, đây là hai đối tượng được tạo nên từ Javascript nên rõ ràng trong HTML5 cũng có sử dụng nó. Với những ứng dụng đó thì bạn thấy javascript không thể chết trong các ứng dụng website được.

Hay thậm chí có những ứng dụng Webgame người ta sử dụng javascript để xử lý các thao tác trên Client, nếu không có nó thì thông thường chọn Flash để xây dựng nhưng lại gặp vấn đề load chậm nên có một số người chọn Javascript để làm.

## **2. Các thư viện Javascript đình đám hiện nay**

Những ứng dụng to lớn của Javascript khiến người ta không thể quên nó được. Hiện nay có rất nhiều libraries, Framework được viêt như:

**AngularJS**: Một thư viện dùng để xây dựng ứng dụng Single Page

**NodeJS**: Một thư viện được phát triển phía Server dùng để xây dựng ứng dụng realtime

**Sencha Touch**: Một Framework  dùng để xây dựng ứng dụng Mobile

**ExtJS**: Một Framework dùng xây dựng ứng dụng quản lý (Web Applications)

**jQuery**: Một thư viện rất mạnh về hiểu ứng

...

Còn rất rất nhiều nhưng mình chỉ liệt kê bấy nhieu thôi nhé

## **3. Viết chương trình javascript đầu tiên**

Qua khái niệm javascript là gì? thì bạn thấy tất cả các trình duyệt đều hỗ trợ nên ta không cần phải download thư viện gì cả mà thông qua những đoạn mã script giúp trình duyệt nhận diện được đó là javascript.

**Cặp thẻ <script language="javascript"></scipt>**

Tất cả những đoạn mã Javascript đều phải đặt trong cặp thẻ script như trên, ví dụ:

|  |
| --- |
| <script language="javascript">       alert("Hello World!");  </script> |

**Đặt thẻ script javascript ở đâu?**

Có ba cách đặt sau đây:

**Internal - viết trong file html hiện tại**

Thông thường chúng ta sẽ viết những đoạn mã javascript trên phần head, tuy nhiên đó không phải là điều kiện bắt buộc, nghĩa là bạn có thể đặt ở đâu tùy thích miễn là những đoạn mã đó phải được bao lại bằng thẻ script.

Ví dụ đặt trong thẻ head:

|  |
| --- |
| <html>      <head>          <title></title>          <script language="javascript">              alert("Hello World!");          </script>      </head>      <body>          <!-- Body contents -->      </body>  </html> |

Ví dụ đặt trong thẻ body:

|  |
| --- |
| <html>      <head>          <title></title>      </head>      <body>          <script language="javascript">              alert("Hello World!");          </script>      </body>  </html> |

**External - viết ra một file js khác rồi import vào**

Hoặc bạn viết những đoạn mã javascript ở một file có phần mở rộng là .js, sau đó dùng thẻ script để import vào (giống CSS vậy). Ví dụ file JS của mình có tên là demo.js thì lúc này mình import vào như sau:

|  |
| --- |
| <script language="javascript" src="demo.js"></script> |

Lúc này bên trong file demo.js bạn không đặt thẻ scirpt nữa nhé vì nó là file .js rồi nên trình duyệt tự nhận diện mã JS

**Inline  - viết trực tiếp trong thẻ HTML**

Inline nghĩa là bạn sẽ viết những đoạn mã Javascript trực tiếp trong thẻ HTML luôn. ví dụ dưới đây viết inline  vì đoạn alert(1) được đặt trong sự kiện onclick.

|  |
| --- |
| <input type="button" onclick="alert(1)" value="Click Me"/> |

**Viết chương trình Hello World!**

**Bước 1**: Bây giờ chúng ta thực hành nhé, bạn hãy tạo một file index.html và lưu tại bất kì vị trí nào, desktop hay ổ D, ổ C gì đó thì tùy bạn, miễn là phần mở rộng là .html và gõ nội dung sau vào:

|  |
| --- |
| <html>      <head>          <title></title>      </head>      <body>          <input type="button" value="Click Me"/>      </body>  </html> |

Chạy lên bằng Firefox hoặc Chrome bạn thấy xuất hiện một button. Bây giờ ta sẽ viết ứng dụng khi click vào button đó thì sẽ xuất hiện một thông báo **Hello World** nhé. Nhưng có lưu ý với bạn là button đó mình có đặt id=clickme

**Bước 2**: Viết javascript khi click vào button có id="clickme" thì thông báo lên màn hình.

|  |
| --- |
| <html>      <head>          <title></title>      </head>      <body>          <input type="button" id="clickme" value="Click Me"/>          <script language="javascript">            // Lấy element có id=clickme lưu vào biến button          var button = document.getElementById('clickme');            // Khi click vào element chứa trong button thì thực hiện một function,          // bên trong function thông báo lên Hello World!          button.addEventListener('click', function(){              alert('Hello World!');          });          </script>      </body>  </html> |

Lưu ý là bạn phải đặt đoạn mã javascript bên dưới button như trong demo nhé, nếu bạn đặt ở trên là sẽ bị lỗi đấy, lý do tại sao thì trong các bài tiếp theo mình sẽ giải thích.

Vậy là xong rồi đấy, bây giờ thì bạn biết **javascript là gì?** rồi phải không nào.

## **4. Lời kết**

Bài này mình muốn giới thiệu đến các bạn khái niệm javascript là gì và đồng thời liệt kê ra những thư viện, Framework được xây dựng từ javascript để các bạn thấy quy mô của nó như thé nào. Chúng ta cũng đã làm một ví dụ nho nhỏ sử dụng hàm getElementById và addEventListener, nếu bạn không hiểu hai hàm này thì cứ từ từ nhé, các bài sau ta sẽ nói về nó nhiều hơn.

**Bài 02: Biến và khai báo biến trong javascript**

|  |
| --- |
| **Mục luc:**   * [1. Khai báo biến trong javascript](http://freetuts.net/bien-va-khai-bao-bien-trong-javascript-265.html#goto-h2-0) * [2. Gán giá trị cho biến trong javascript](http://freetuts.net/bien-va-khai-bao-bien-trong-javascript-265.html#goto-h2-1) * [3. Gán kiểu giá trị cho biến](http://freetuts.net/bien-va-khai-bao-bien-trong-javascript-265.html#goto-h2-2) * [4. In giá trị của biến ra trình duyệt](http://freetuts.net/bien-va-khai-bao-bien-trong-javascript-265.html#goto-h2-3) * [Lời kết](http://freetuts.net/bien-va-khai-bao-bien-trong-javascript-265.html#goto-h2-4) |

Trong bài này ta sẽ tìm hiểu các khai báo biến và gán giá trị cho biến trong javascript. , đồng thời học thêm cách dùng hàm document.write để in giá trị biến ra ngoài trình duyệt . Việc khai báo biến trong javascript cũng giống như [khai báo biến trong php](http://freetuts.net/khai-bao-bien-va-hang-so-trong-php-2.html) vậy, tuy cú pháp nó không giống nhau nhưng về cách gán giá trị thì y chang.

**1. Khai báo biến trong javascript**

Để khai báo một biến ta sử dụng từ khóa  var tenbien. Ví dụ:

|  |
| --- |
| var username; |

Tuy nhiên không phải muốn đặt tên biến gì cũng được mà phải tuân theo quy tắc sau đây:

* Tên biến phải là các chữ không dấu viết  hoa hoặc viết thường, các chữ số từ 0-9 và dấu gạch dưới (\_).
* Tên biến bắt đầu phải là chữ hoặc dấu gạch dưới (\_), nếu bắt đầu bằng số là sai
* Tên biến có thể đặt dài hay ngắn tùy vào lập trình viên.

**Ví dụ**:

|  |
| --- |
| // Đúng  var username;  // Đúng  var \_username;  // Đúng  var \_\_username;  // Đúng  var username90;  // SAI  var 90thehalfheart; |

Ngoài cách khai báo trên ta còn có cách khai báo biến liên tiếp cách nhau bởi dấu phẩy, ví dụ:

|  |
| --- |
| var username, \_username, \_\_username, username90; |

**2. Gán giá trị cho biến trong javascript**

Để gán giá trị cho biến ta dùng dấu bằng (=) để gán vế phải vào vế trái, ví dụ gán giá trị**thehalfheart** vào biến **username** thì ta viết là username = 'thehalfheart'.

Có hai cách gán thông dụng đó là vừa khai báo biến vừa gán giá trị và khai báo rồi mới gán giá trị:

**Ví dụ vừa khai báo vừa gán giá trị**:

|  |
| --- |
| var username = 'thehalfheart'; |

**Ví dụ khai báo xong mới gán giá trị**:

|  |
| --- |
| var username;  username = 'thehalfheart'; |

**3. Gán kiểu giá trị cho biến**

Giống như php, để xác định biến có kiểu dữ liệu là gì thì ta dựa vào giá trị mà nó đang có, có nghĩa khi bạn gán một con số không có dấu chấm động thì nó sẽ là kiểu INT, nếu gán một chuỗi thì là kiểu String.

**Ví dụ**:

|  |
| --- |
| // Biến website đag kiểu String  var website = 'thehalfehart';    // Biến website chuyển sang kiểu INT  website = 12;    // Biến website chuyển sang kieeur float  website = 12.5; |

Trong Javascript bạn sẽ có các kiểu dữ liệu thông dụng như kiểu chuỗi (*String*), số (*Number*), mảng (*Array*), đối tượng (*Object*). Chi tiết từng kiểu dữ liệu chúng ta sẽ được học trong các bài tiếp theo.

**4. In giá trị của biến ra trình duyệt**

Để in giá trị của biến hoặc một chuỗi nào đó ra trình duyệt thì ta sử dụng hàm document.write(value).

**Ví dụ**:

|  |
| --- |
| <html>      <head>          <title></title>      </head>      <body>          <script language="javascript">              var website = 'freetuts.net';              document.write(website);          </script>      </body>  </html> |

**Lời kết**

Bài này mình dừng ở đây, hy vọng qua những ví dụ trên giúp bạn hiểu được cách sử dụng biến căn bản trong javascript, tuy nói là căn bản nhưng nó là nền tảng đấy nhé các bạn, hãy xem các ví dụ và thực hành theo, kết hợp nhuần nhuyễn giữa các phần để tự đưa ra những ví dụ cho riêng mình.

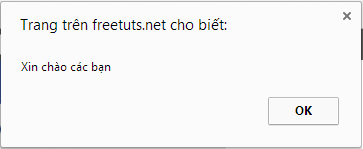
**Bài 03: Hàm alert() - confirm() - prompt() trong javascript**

|  |
| --- |
| **Mục luc:**   * [1. Hàm alert() trong javascript](http://freetuts.net/ham-alert-confirm-prompt-trong-javascript-264.html#goto-h2-0) * [2. Hàm confirm() trong javascript](http://freetuts.net/ham-alert-confirm-prompt-trong-javascript-264.html#goto-h2-1) * [3. Hàm prompt() trong javascript](http://freetuts.net/ham-alert-confirm-prompt-trong-javascript-264.html#goto-h2-2) * [Lời kết](http://freetuts.net/ham-alert-confirm-prompt-trong-javascript-264.html#goto-h2-3) |

Hôm nay chúng ta sẽ tìm hiểu đến ba hàm rất thông dụng và hữu dụng trong javascript đó là hàm alert(), confirm() prompt(), đây là ba hàm thường được sử dụng để thông báo và lấy thông tin của người dùng để xử lý. Tuy nhiên trong thực tế thời này người ta đã tự viết ra những plugin bằng javascript và HTML để thay thế cho hay hàm này vì mặc định nó thông báo nhìn hơi chuối :D.

**1. Hàm alert() trong javascript**

Hàm alert() có nhiệm vụ in một thông báo popup, nó có một tham số truyền vào và tham số này chính là nội dung ta muốn thông báo với người dùng.



**Ví dụ**: Viết chương trình khi click vào button thì xuất hiện một thông báo.

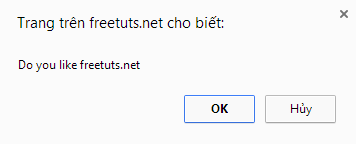
Bài này ta đã làm ở ví dụ bài tìm hiểu [javascript là gì](http://freetuts.net/javascript-la-gi-viet-ung-dung-javascript-dau-tien-263.html) rồi nên mình sẽ viết lại nó hơi khác chút xíu, đó là mình sẽ dùng sự kiện onclick thay vì dùng hàm addEventListener như ở bài trước nhé.

|  |
| --- |
| <html>    <head>    </head>    <body>      <input type="button" onclick="alert('Xin chào các bạn')" value="Click Me"/>    </body>  </html> |

Các bạn thấy mình đã code sử dụng hàm alert() trong sự kiện onclick đấy, đây là cách code inline mà chúng ta đã học ở bài trước rồi nhé.

**2. Hàm confirm() trong javascript**

Hàm confirm() cũng sẽ xuất hiện một thông báo popup nhưng nó có thêm hai sự lựa chọn là Yes và No, nếu người dùng chọn Yes thì nó trả về TRUE và ngược lại nếu chọn NO thì nó sẽ trả về FALSE. Nó cũng có một tham số truyền vào và tham số này chính là nội dung thông báo.



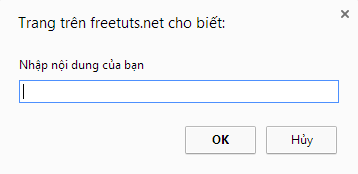
**Ví dụ**: Viết chương trình hiển thị thông báo  xác nhận và dùng hàm alert() để hiển thị kết quả người dùng chọn.

Vì chúng ta chưa học hàm nên mình sẽ không viết dài dòng mà sẽ viết chạy trực tiếp khi load trang, nghĩa là khi trang web được chạy lên thì thông báo nó hiển thị luôn chứ không cần phải click vào button nhé.

|  |
| --- |
| <html>      <head>          <title></title>          <script language="javascript">              confirm("Do you like freetuts.net");          </script>      </head>      <body>        </body>  </html> |

**3. Hàm prompt() trong javascript**

Hàm prompt() dùng  để lấy thông tin từ người dùng, gồm có hai tham số truyền vào là nội dung thông báo và giá trị ban đầu. Nếu người dùng không nhập vào thì giá trị nó sẽ trả về là NULL



**Ví dụ**: Viết chương trìn lấy thông tin tên của người dùng

Ta sẽ [khai báo biến](http://freetuts.net/bien-va-khai-bao-bien-trong-javascript-265.html) lưu trữ kết quả và kết hợp sử dụng hàm alert() để thông báo kết quả người dùng nhập vào.

|  |
| --- |
| <html>      <head>          <title></title>          <script language="javascript">              var t = prompt("Nhập tên của bạn", '');              alert(t);          </script>      </head>      <body>        </body>  </html> |

**Lời kết**

Ba hàm alert() - confirm() - prompt() trong javascript rất hay sử dụng nên các bạn hãy gắng nhớ nó nhé, vì nội dung chúng ta chưa học nhiều nên ví dụ vẫn chưa thấy hay lắm, khi nào ta học đến hàm, events hay selector trong javascript thì bạn sẽ thấy nó tuyệt vời.

**Bài 04: Các toán tử toán học và toán tử gán trong javascript**

|  |
| --- |
| **Mục luc:**   * [1. Toán tử toán học trong javascript](http://freetuts.net/cac-toan-tu-toan-hoc-va-toan-tu-gan-trong-javascript-266.html#goto-h2-0) * [2. Toán tử gán trong javascript](http://freetuts.net/cac-toan-tu-toan-hoc-va-toan-tu-gan-trong-javascript-266.html#goto-h2-1) * [3. Lời kết](http://freetuts.net/cac-toan-tu-toan-hoc-va-toan-tu-gan-trong-javascript-266.html#goto-h2-2) |

Bài này mình không trình bày các khái niệm như "toán tử là gì? biểu thức là gì" vì các bạn đã học PHP hoặc là một ngôn ngữ như C, C++ rồi, mình chỉ hướng dẫn một vài thao tác liên quan đến các phép toán thường hay sử dụng trong javascript thôi nhé.

**1. Toán tử toán học trong javascript**

Bài viết được đăng tại freetuts.net - thehalfheart@gmail.com

Thông thường chúng ta sử dụng toán tử toán học như cộng, trừ, nhận, chia, ... để thực xử lý thay đổi giá trị các biến trong javascript. Sau đây là bảng danh sách các toán tử hay dùng.

| **Toán tử** | **Mô tả** |
| --- | --- |
| **+** | 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. Ví dụ:   |  | | --- | | var a = 20;  var b = 5;  // Kết quả biến c có giá trị 25  var c = a + b; | |
| **-** | Phép trừ và dùng với number. Ví dụ:   |  | | --- | | var a = 20;  var b = 5;  // Kết quả biến c có giá trị 15  var c = a - b; | |
| **\*** | Phép nhân và dùng với number. Ví dụ:   |  | | --- | | var a = 20;  var b = 5;  // Kết quả biến c có giá trị 100  var c = a \* b; | |
| **/** | Phép chia và dùng với number. Ví dụ:   |  | | --- | | var a = 20;  var b = 5;  // Kết quả biến c có giá trị 4  var c = a / b; | |
| **%** | 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. Ví dụ:  Trường hợp chia dư 0:   |  | | --- | | var a = 20;  var b = 5;  // Kết quả biến c có giá trị 0  // Lý do là a / b dư 0  var c = a % b; |   Trường hợp chia dư khác 0:   |  | | --- | | var a = 22;  var b = 5;  // Kết quả biến c có giá trị 2  // Lý do là a / b = 4 dư 2  var c = a % b; | |
| **++** | 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, ví dụ:  |  | | --- | | var c = 12;  alert(++c); // kết quả là 13  alert(c); // kết quả là 13 |  * Trường hợp đứng sau biến thì nó sẽ lấy giá trị rồi tăng lên, ví dụ:  |  | | --- | | var c = 12;  alert(c++); // kết quả là 12  alert(c); // kết quả là 13 | |
| **--** | Phep 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.   * Trường hợp đứng trước biến thì nó sẽ giảm trước khi lấy giá trị, ví dụ:  |  | | --- | | var c = 12;  alert(--c); // kết quả là 11  alert(c); // kết quả là 11 |  * Trường hợp đứng sau biến thì nó sẽ lấy giá trị rồi giảm xuống, ví dụ:  |  | | --- | | var c = 12;  alert(c--); // kết quả là 12  alert(c); // kết quả là 11 | |

**2. Toán tử gán trong javascript**

Ở các bài trước chúng ta đã sử dụng toán tử gán rất nhiều đó là toán tử gán bằng (=), toán tử này thường hay sử dụng nhiều nhất trong các ngôn ngữ lập trình. Sau đây là danh sách các toán tử gán  mà ta hay dùng trong javascript.

| **Toán tử** | **Ví dụ** | **Mô tả** |
| --- | --- | --- |
| **=** | **x = y** | Gán gí trị của biến x bằng giá trị của biến y, ví dụ:   |  | | --- | | var x = 12;  var y = x; // y = 12 | |
| **+=** | **x += y** | Tương đương với **x = x + y**. Ví dụ:   |  | | --- | | var x = 12;  var y = 10;  // lúc này x = 22  x += y; // tương đương x = x + y | |
| **-=** | **x -= y** | Tương đương với **x = x - y**. Ví dụ:   |  | | --- | | var x = 12;  var y = 10;  // lúc này x = 2  x -= y; // tương đương x = x - y | |
| \***=** | **x \*= y** | Tương đương với **x = x \* y**. Ví dụ:   |  | | --- | | var x = 12;  var y = 10;  // lúc này x = 120  x \*= y; // tương đương x = x \* y | |
| **/=** | **x /= y** | Tương đương với **x = x / y**. Ví dụ:   |  | | --- | | var x = 12;  var y = 10;  // lúc này x = 1.2  x /= y; // tương đương x = x / y | |
| **%=** | **x %= y** | Tương đương với **x = x % y**. Ví dụ:   |  | | --- | | var x = 12;  var y = 10;  // lúc này x = 2;  x %= y; // tương đương x = x % y | |

**3. Lời kết**

Nếu bài này là một ngôn ngữ lập trình căn bản thì mình sẽ trình bày chi tiết hơn nhưng vì các bạn đã học qua ngôn ngữ lập trình khác rồi nên mình chỉ viết ở dạng liệt kê để các bạn dùng để tham khảo và tra cứu thôi nhé. Chúc các bạn học tốt với serie Javascript này.

**Bài 05: Lệnh kiểm tra điều kiện if else trong javascript**

|  |
| --- |
| **Mục luc:**   * [1. Lệnh if trong javascript](http://freetuts.net/lenh-kiem-tra-dieu-kien-if-else-trong-javascript-267.html#goto-h2-0) * [2. Lệnh else trong javascript](http://freetuts.net/lenh-kiem-tra-dieu-kien-if-else-trong-javascript-267.html#goto-h2-1) * [3. Kết hợp nhiều lệnh if else trong javascript](http://freetuts.net/lenh-kiem-tra-dieu-kien-if-else-trong-javascript-267.html#goto-h2-2) * [4. Lệnh if else lồng nhau trong javascript](http://freetuts.net/lenh-kiem-tra-dieu-kien-if-else-trong-javascript-267.html#goto-h2-3) * [5. Lời kết](http://freetuts.net/lenh-kiem-tra-dieu-kien-if-else-trong-javascript-267.html#goto-h2-4) |

Câu lệnh if else dùng để kiểm tra một mệnh đề nào đó có đúng hay không, nếu đúng thì thực thi những câu lệnh bên trong và ngược lại nếu sai thì nó sẽ bỏ qua những câu lệnh đó. Bây giờ ta sẽ tìm hiểu chi tiết nó nhé.

**1. Lệnh if trong javascript**

**Cú pháp**:

|  |
| --- |
| if (condition){      // statment  } |

Trong đó condition là mệnh đề điều kiện và luôn luôn phải có một trong hai giá trị là true/falsehoặc tương đương (1 => true, 0 => false).

**Ví dụ**:

|  |
| --- |
| var a = 12;  var b = 12;  if (a == b){      alert('a và b bằng nhau');  } |

Kết quả sẽ có thông báo popup "*a và b bằng nhau*".

Trường hợp nếu các câu lệnh bên trong là câu lệnh đơn ( chỉ có một lệnh) thì ta có thể bỏ cặp dấu ngoặc nhọn, với ví dụ trên thì ta sẽ viết lại như sau:

|  |
| --- |
| var a = 12;  var b = 12;    if (a == b)      alert('a và b bằng nhau'); |

**2. Lệnh else trong javascript**

Lệnh else sẽ được thực thi nếu lệnh if không thỏa, như vậy khi dùng lệnh else thì bắt buộc phải có một lệnh if trước nó.

**Ví dụ:**

|  |
| --- |
| var a = 12;  var b = 10;  if (a == b){      alert('a và b bằng nhau');  } else {      alert('a và b khác nhau');  } |

Kết quả sẽ chạy lênh else tại vì lệnh if không thỏa.

**3. Kết hợp nhiều lệnh if else trong javascript**

Ta có thể kết hợp nhiều câu lệnh if để xử lý, cú pháp của nó cũng không có gì khác.

**Ví dụ:**

|  |
| --- |
| var a = 12;  if (a > 12) {      alert('a > 12');  } else if (a < 12) {      alert('a < 12');  } else {      alert('a = 12');  } |

Kết quả sẽ chạy lệnh else cuối cùng tại vì các lệnh if bên trên không thỏa điều kiện. Bây giờ bạn thử đổi giá trị của a thành a = 10 thì câu lệnh else if sẽ được thực hiện.

**4. Lệnh if else lồng nhau trong javascript**

Như các ngôn ngữ 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.

**Ví dụ:**

|  |
| --- |
| var a = 13;  // Nếu a > 12  if (a > 12) {      // Khai báo biến b      var b = 20;      // Nếu a bằng b      if (a == b) {          alert(' a = b ');      } else { // ngược lại a khác b          alert(' a != b ');      }  } |

**5. Lời kết**

Quá đơn giản phải không nào, mình không giải thích nhiều lắm vì các bạn đã biết lập trình rồi và cũng thao tác với câu lệnh if else khá nhiều rồi. Còn nếu bạn chưa biết gì về lập trình thì mình nghĩ bạn nên tìm hiểu một ngôn ngữ khác trước khi quay sang đây nhé.

**Bài 06: Vòng lặp for trong javascript**

Đăng bởi: [TheHalfHeart](http://freetuts.net/author/thehalfheart)- Vào ngày: 09-01-2015- Chuyên mục: [Javascript](http://freetuts.net/hoc-javascript)- View: 9484

**Mục luc:**

* [1. Một số cách sử dụng vòng lặp for trong javascript](http://freetuts.net/vong-lap-for-trong-javascript-270.html#goto-h2-0)
* [2. Lặp vô hạn với vòng lặp for trong javascript](http://freetuts.net/vong-lap-for-trong-javascript-270.html#goto-h2-1)
* [3. Vòng lặp for lồng nhau trong javascript](http://freetuts.net/vong-lap-for-trong-javascript-270.html#goto-h2-2)
* [4. Lời kết](http://freetuts.net/vong-lap-for-trong-javascript-270.html#goto-h2-3)

Vòng lặp for trong javascript thường dùng để lặp một mảng hoặc một khoảng (min max) nào đó nhằm mục đích xử lý giải quyết vấn đề cho bài toán, vì ta đang học căn bản nên trong bài này mình chỉ làm những bài tập đơn giản liên quan đến mảng trong javascript thôi nhé. Trước tiền ta tìm hiểu cấu trúc của nó rồi đi vào vấn đề chính sau:

|  |  |
| --- | --- |
| 1  2  3  4 | var i = 0;  for (i = 0; i < 100; i++){      // Dòng lệnh xử lý vòng lặp  } |

**Trong đó**:

* var i = 0; là [khai báo biến](http://freetuts.net/bien-va-khai-bao-bien-trong-javascript-265.html) điều khiển vòng lặp i
* (i = 0) là điểm bắt đầu lặp (lặp từ 0)
* (i < 100) là điều kiện dừng vòng lặp, nghĩa là lặp nếu i < 100. Bạn có thể dùng một điều kiện bất kì thông qua các [toán tử](http://freetuts.net/cac-toan-tu-toan-hoc-va-toan-tu-gan-trong-javascript-266.html)  miễn là nó trả về **true hoặc false** như (i <= 100, i == 100)
* (i++) là tăng bước nhảy, bạn có thể dùng công thức khác như i+=2, i-=2, i--, ...

Như ví dụ trên thì ta sẽ có 100 vòng lặp từ**0 -> 99**

**1. Một số cách sử dụng vòng lặp for trong javascript**

Bài viết được đăng tại freetuts.net - thehalfheart@gmail.com

**Lặp với bước nhảy tăng 1 đơn vị**: [XEM DEMO](http://freetuts.net/editor.html?id=189)

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | var i;  // Lặp 10 lần từ 0 -> 9  // Bước nhảy là i++ nên sau mỗi lần lặp i tăng lên 1 đơn vị  for (i = 0; i < 10; i++){      document.write(i + '<br/>');  } |

**Lặp với bước nhay giảm 1 đơn vị**: [XEM DEMO](http://freetuts.net/editor.html?id=190)

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | var i;  // Lặp 10 lần từ 10 -> 1  // Bước nhảy là i-- nên sau mỗi lần lặp i sẽ giảm 1 đơn vị  for (i = 10; i > 0; i--){      document.write(i + '<br/>');  } |

**Lặp với bước nhay tăng N đơn vị**: [XEM DEMO](http://freetuts.net/editor.html?id=191)

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | var i;  var n = 2;  // Lặp 5 lần từ 0 -> 8, bước nhảy là 2  for (i = 0; i < 10; i+=n){      document.write(i + '<br/>');  } |

**Lặp với bước nhảy giảm N đơn vị**: [XEM DEMO](http://freetuts.net/editor.html?id=192)

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | var i;  var n = 2;  // Lặp 5 lần từ 10 -> 2, bước nhảy là -2  for (i = 10; i > 0; i-=n){      document.write(i + '<br/>');  } |

**Khai báo biến lặp (i) ngay trong vòng lặp**:

Ngoài cách khai báo biến (i) ở ngoài vòng lặp thì ta có thể khai báo trong vòng lặp như ví dụ sau đây:

|  |  |
| --- | --- |
| 1  2  3 | for (var i = 0; i < 10; i++){      document.write(i + '<br/>');  } |

**2. Lặp vô hạn với vòng lặp for trong javascript**

Khi sử dụng vòng lặp for khí khi lặp vô hạn hơn vòng lặp while, do while. Tuy nhiên nếu bạn dùng sai thì sẽ dẫn đến lặp vô hạn đấy

**Dưới đây là một ví dụ lặp vô hạn**:

|  |  |
| --- | --- |
| 1  2  3 | for (var i = 0; i < 10; i--){      document.write(i + '<br/>');  } |

Ta thấy bước nhảy là i-- tức là sau mỗi vòng lặp sẽ giảm - xuống 1 đơn vị trong khi điều kiện dừng là i >= 10, rõ ràng điều kiện này sẽ không bao giờ đúng nên vòng lặp dẫn đến lặp vô hạn.

**3. Vòng lặp for lồng nhau trong javascript**

Khái niệm vòng lặp lồng nhau cũng giống như câu [lệnh if else lồng nhau](http://freetuts.net/lenh-kiem-tra-dieu-kien-if-else-trong-javascript-267.html) vậy thôi, nghĩa là vòng lặp sẽ nằm trong vòng lặp. Cấu trúc của nó như sau:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | for (var i = 0; i < 10; i++)  {      for (var j = 0; j < 10; j++){          // Statment      }  } |

Có một điều lưu ý là hai biến điều khiển vòng lặp phải khác nhau nhé, trong ví dụ trên mình dùng i cho vòng lặp ở ngoài và j cho vòng lặp con phía trong.

**Ví dụ**: Viết chương trình in ra một ma trận 10x10

[XEM DEMO](http://freetuts.net/editor.html?id=193)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | for (var i = 0; i <= 9; i++)  {      for (var j = 0; j <= 9; j++){          // In ra vị trí của ma trận [i][j]          document.write("(["+i+"]["+j+"])");      }      // Xuống hàng      document.write("<br/>");  } |

**4. Lời kết**

Vòng lặp for khá là đơn giản nên các bạn dễ dàng nắm bắt phải không nào, nó cũng giống như các ngôn ngữ lập trình khác thôi nên để hiểu nó không phức tạp. Hy vọng qua bài vòng lặp for này sẽ giúp bạn học thêm được nhiều điều hơn. Chúc các bạn học tốt.

**Bài 07: Vòng lặp while - do while trong javascript**

Đăng bởi: [TheHalfHeart](http://freetuts.net/author/thehalfheart)- Vào ngày: 10-01-2015- Chuyên mục: [Javascript](http://freetuts.net/hoc-javascript)- View: 7001

**Mục luc:**

* [1. Vòng lặp while trong javascript](http://freetuts.net/vong-lap-while-do-while-trong-javascript-271.html#goto-h2-0)
* [2. Vòng lặp do while trong javascript](http://freetuts.net/vong-lap-while-do-while-trong-javascript-271.html#goto-h2-1)
* [3. Vòng lặp while - do while lồng nhau](http://freetuts.net/vong-lap-while-do-while-trong-javascript-271.html#goto-h2-2)
* [4. Lời kết](http://freetuts.net/vong-lap-while-do-while-trong-javascript-271.html#goto-h2-3)

Vòng lặp while và do while dùng để lặp với trường hợp tác không biết chính xác số lần lặp là bao nhiêu và trường hợp điều kiện dừng vòng lặp phức tạp, điều này hoàn toàn khác với [vòng lặp for](http://freetuts.net/vong-lap-for-trong-javascript-270.html). Và khi sử dụng vòng lặp while thì rất dễ bị lặp vô hạn nếu ban không xử lý đúng logic vì thế hãy cẩn thận nhé.

**1. Vòng lặp while trong javascript**

Bài viết được đăng tại freetuts.net - thehalfheart@gmail.com

**Cấu trúc vòng lặp while**

|  |  |
| --- | --- |
| 1  2  3 | while (condition){      // do something  } |

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.

**Ví dụ: Dùng vòng lặp while lặp từ 1 tới 10.**[XEM DEMO](http://freetuts.net/editor.html?id=200)

|  |  |
| --- | --- |
| 1  2  3  4  5 | var i = 1;  while (i <= 10){      document.write(i + '<br/>');      i++; // tăng i lên nếu không sẽ bị lặp vô hạn  } |

Trong ví dụ này biến i có giá trị khởi đầu là  i = 1, điều kiện dừng vòng lặp là i <= 10. Như vậy sau mỗi bước lặp nếu ta không tăng i lên thì vòng lặp sẽ dẫn tới lặp vô hạn.

**Ví dụ: Dùng vòng lặp while lặp từ 10 trở về 1.**[XEM DEMO](http://freetuts.net/editor.html?id=201)

|  |  |
| --- | --- |
| 1  2  3  4  5 | var i = 10;  while (i >= 1){      document.write(i + '<br/>');      i--; // giảm i xuống nếu không sẽ bị lặp vô hạn  } |

Trường hợp này thì lại khác sau mỗi vòng lặp thì ta phải giảm i xuống 1 đơn vị vì điều kiện lặp là i >= 1.

**Ví dụ: Lặp với điều kiện phức tạp.**[XEM DEMO](http://freetuts.net/editor.html?id=202)

Ta sẽ làm một ví dụ đơn giản đó là sử dụng hàm [prompt trong javascript](http://freetuts.net/ham-alert-confirm-prompt-trong-javascript-264.html) để lấy thông tin từ người dùng, nếu người dùng nhập vào số trong khoảng **1 -> 10** thì dừng, ngược lại thì yêu cầu họ nhập lại.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | // Biến lưu giá trị người dùng nhập vào  var value = null;    // Trong khi giá trị bé hơn 1 hoặc giá trị lớn hơn 10  // thì chạy thân vòng lặp  while (value < 1 || value > 10){      value = prompt("Nhập vào số từ 1 -> 10");  }    // Sau khi nhập đúng thì in ra màn hình  alert("Số bạn vừa nhập là " + value); |

Như vậy trường hợp này ta không hề biết là lặp bao nhiêu lần cả phải không nào :D.

**2. Vòng lặp do while trong javascript**

Khác một chút so vói vòng lặp while là vòng lặp do while sẽ thực thi trường rồi mới kiểm tra điều kiện. Ví dụ bạn chơi trò chơi bốc thăm, lần đầu tiên bạn sẽ bốc nếu cây thăm may mắn thì bốc tiếp và ai được nhiều cây thăm may mắn thì thắng, như vậy có nghĩa là bạn sẽ được bốc 1 lần rồi mới kiểm tra điều kiện cho lần bốc kế tiếp. Từ đó suy ra trong vòng lặp do while sẽ luôn luôn thực thi lặp **ít nhất 1 lần**.

**Cấu trúc vòng lặp do while**

|  |  |
| --- | --- |
| 1  2  3  4 | do {      // some thing  }  while (condition); |

Trong đó **condition** là điều kiện để dừng vòng lặp.

**Ví dụ chứng minh vòng lặp do while luôn luôn lặp ít nhất là 1 lần**

Hồi nãy mình nói có vẻ bạn không tin nên mình sẽ làm một ví dụ luôn. Như bạn biết nếu điều kiện **condition** là false thì sẽ không lặp thì bây giờ ta cho nó false luôn để xem có lặp không nhé.

[XEM DEMO](http://freetuts.net/editor.html?id=203)

|  |  |
| --- | --- |
| 1  2  3  4 | do {      alert(1);  }  while (false); |

Chạy lên bạn sẽ thấy nó có lặp 1 lần. Bây giờ ta thay bằng vòng lặp while xem có lặp không nhé.

[XEM DEMO](http://freetuts.net/editor.html?id=204)

|  |  |
| --- | --- |
| 1  2  3 | while (false){      alert(1);  } |

Im re phải không nào :D

**Ví dụ: Sử dụng vòng lặp do while để viết lại chương trình yêu cầu nhập các số từ 1 -> 10 ở ví dụ trong phần vòng lặp while.**[XEM DEMO](http://freetuts.net/editor.html?id=205)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | // Biến lưu giá trị người dùng nhập vào  var value = null;    // Thực thi  do {      value = prompt("Nhập vào số từ 1 -> 10");  }  while (value < 1 || value > 10);    // Sau khi nhập đúng thì in ra màn hình  alert("Số bạn vừa nhập là " + value); |

**3. Vòng lặp while - do while lồng nhau**

Cũng giống như trong vòng lặp for, ta có thể lồng nhiều vòng lặp while hoặc do while lại với nhau để viết chương trình.

**Ví dụ: Dùng vòng lặp while xây dựng ma trận 10x10.**[XEM DEMO](http://freetuts.net/editor.html?id=206)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | var i = 0;  // Vòng lặp ngoài  while (i <= 9)  {      // Mỗi lần lặp gan j = 0;      var j = 0;        // Lặp nếu j < <=      while (j <= 9)      {          // In ra màn hình          document.write('['+i+']['+j+'] ');            // Tăng j lên nếu không sẽ bị lặp vô hạn          j++;      }      // Xuống hàng      document.write('<br/>');        //Tăng i lên nếu không sẽ bị lặp vô hạn      i++;  } |

Tương tự cho vòng lặp do while các bạn tự làm lấy nhé.

**4. Lời kết**

Điểm lưu ý nhất của bài này là so sánh sự khác biệt giữa vòng lặp while và do while trong javascript, cách sử dụng nó và thường khi nào thì sử dụng. Đối với vòng lặp do while thì nó luôn luôn lặp ít nhất 1 lần tại vì nó lặp trường rồi mới kiểm tra điều kiện. Chỉ vậy thôi :D chúc bạn học tốt.

**Bài 08: Hàm và tạo hàm (function) trong javascript**

Đăng bởi: [TheHalfHeart](http://freetuts.net/author/thehalfheart)- Vào ngày: 10-01-2015- Chuyên mục: [Javascript](http://freetuts.net/hoc-javascript)- View: 15637

**Mục luc:**

* [1. Định nghĩa hàm (function) trong javascript](http://freetuts.net/ham-va-tao-ham-function-trong-javascript-274.html#goto-h2-0)
* [2. Hàm có return và hàm không có return](http://freetuts.net/ham-va-tao-ham-function-trong-javascript-274.html#goto-h2-1)
* [3. Một số ví dụ tạo hàm trong javascript](http://freetuts.net/ham-va-tao-ham-function-trong-javascript-274.html#goto-h2-2)
* [4. Lời kết](http://freetuts.net/ham-va-tao-ham-function-trong-javascript-274.html#goto-h2-3)

Hàm là một [phương pháp lập trình truyền thống](http://freetuts.net/phuong-phap-lap-trinh-truyen-thong-26.html) và thường được ứng dụng trong các phương pháp lập trình thủ tục, lập trình hướng module,... Với các ngôn ngữ lập trình bậc cao như C++ thì việc dùng hàm để code ứng dụng là người ta không thích tại vò khó quản lý bảo trì, nhưng đối với javascript và PHP thì ta rất hay dùng hàm.

**1. Định nghĩa hàm (function) trong javascript**

Bài viết được đăng tại freetuts.net - thehalfheart@gmail.com

Từ bài 1 tới giờ ta đang code từng đoạn code riêng lẻ và khi cần thì code lại, như vậy giả sử ta cần xử lý vấn đề đó trong 1000 trường hợp thì ta phải code lại 1000 lần nên rất tốn thời gian. Chính vì vậy người ta nghĩ ngay đến Hàm, có nghĩa là sẽ gom một số đoạn code vào một khối xử lý và khi cần thì gọi ra dùng.

Giả sử mình viết chương trình kiểm tra số chẵn hay lẻ như sau:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | var number = 2;    if (number % 2 == 0){     alert('Số chẵn');  }  else {     alert('Số lẻ');  } |

Giả sử mình cần kiểm tra cho 100 số, như vậy mình cứ viết if else cho 100 lần quá nản phải không nào :D. Để giải quyết nó ta tìm hiểu hàm đã nhé.

**Cú pháp tạo hàm trong javascript**

|  |  |
| --- | --- |
| 1  2  3  4 | function name\_of\_function(var1, var2, var3, ...)  {      // Some code  } |

**Trong đó**:

* **function**: là từ khóa của javascript nên bắt buộc phải như vậy
* **name\_of\_function**: là tên của function, thông thường chúng ta tạo những tên có ý nghĩa như find\_max, find\_min, ...
* **var1, var2 var3, ...** là các tham số truyền vào hàm. Ví dụ viết hàm kiểm tra số chẵn hay lẽ thì ta sẽ có một tham số đó là số cần kiểm tra.

**Ví dụ: Viết hàm kiểm tra một số chẵn hay lẻ.**[XEM DEMO](http://freetuts.net/editor.html?id=207)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | // Tạo hàm  function check\_number(number)  {      if (number % 2 == 0){          alert(number + ' là số chẵn');      }      else {          alert(number + 'Số lẻ');      }  }    // Sử dụng hàm kiểm tra cho 5 số  check\_number(1);  check\_number(2);  check\_number(3);  check\_number(4);  check\_number(5); |

Các bạn thấy mình đã tạo một hàm với tham số truyền vào có tên là number. Như vậy khi sử dụng nếu ta truyền số 1 vào thì lúc này biến number trong hàm sẽ có giá trị là 1, tương tự cho 2, 3,4,5.

Lưu ý với các bạn trong javascript không tồn tại khái niệm con trỏ và tham chiếu trong hàm

**2. 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.  Ví dụ ta cần viết một hàm tính tổng của hai số a và b thì hàm này phải trả về giá trị là tổng của hai số a, b. Xem ví dụ sau: [XEM DEMO](http://freetuts.net/editor.html?id=208)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | // Khai báo hàm  function tinh\_tong(a, b)  {      // trả về kết quả là a + b      return a + b;  }    // Sử dụng  var so1 = 1;  var so2 = 2;    // truyền so1 và so2 vào hàm  var ketqua = tinh\_tong(so1, so2);    alert(ketqua); |

Hàm không có return là hàm không có sử dụng từ khóa return đặt trong hàm. Ví dụ viết chương trình in ra tổng của hai số a và b. [XEM DEMO](http://freetuts.net/editor.html?id=209)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | // Khai báo hàm  function tinh\_tong(a, b)  {      document.write('Tổng là ' + (a + b));  }    // Sử dụng  var so1 = 1;  var so2 = 2;    // truyền so1 và so2 vào hàm  tinh\_tong(so1, so2); |

Như vậy tùy vào mục đích mà ta dùng có return hay không có return. Nhưng thông thường ta sử dụng return ở những trường hợp cần lấy kết quả đó để xử lý tiếp, như ở ví dụ trên đó là mình lấy kết quả để **alert**.

**3. Một số ví dụ tạo hàm trong javascript**

**Ví dụ 1: Viết chương trình kiểm tra một năm có phải là năm nhuận hay không**[XEM DEMO](http://freetuts.net/editor.html?id=210)

Năm nhuận là năm chia hết cho 4, nếu chia hết cho 100 thì nó phải chia hết cho 400. Đây là định nghĩa năm nhuận còn chính xác hay không thì mình không biết nhé :D vì có trường hợp nó sai.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | // khai báo hàm  function kiem\_tra\_nam\_nhuan(nam)  {      // nếu năm chia hết cho 100      // thì kiểm tra nó có chia hết cho 400 hay không      if (nam % 100 == 0)      {          // nêu chia hết cho 400 thì là năm nhuận          if (nam % 400 == 0){              alert(nam + ' là năm nhuận');          }          else { // ngược lại không phải năm nhuận              alert(nam + ' không phải năm nhuận');          }      }      else if (nam % 4 == 0){ // trường hợp chia hết cho 4 thì là năm nhuận          alert(nam + ' là năm nhuận');      }      else { // cuối cùng trường hợp không phải năm nhuận          alert(nam + 'không phải là năm nhuận');      }  }  // sử dụng  kiem\_tra\_nam\_nhuan(4); |

**Ví dụ 2: thực hiện lại ví dụ trên nhưng sử dụng return để trả kết quả vè, nếu true thì là năm nhuận, false thì không phải năm nhuận.**[XEM DEMO](http://freetuts.net/editor.html?id=211)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | // khai báo hàm  function kiem\_tra\_nam\_nhuan(nam)  {      // nếu năm chia hết cho 100      // thì kiểm tra nó có chia hết cho 400 hay không      if (nam % 100 == 0)      {          // nêu chia hết cho 400 thì là năm nhuận          if (nam % 400 == 0){              return true;          }          else { // ngược lại không phải năm nhuận              return false;          }      }      else if (nam % 4 == 0){ // trường hợp chia hết cho 4 thì là năm nhuận          return true;      }      else { // cuối cùng trường hợp không phải năm nhuận          return false;      }  }  // sử dụng  var flag = kiem\_tra\_nam\_nhuan(4);    if (flag){      alert('là năm nhuận');  }  else {      alert('không phải là năm nhuận');  } |

Trong ví dụ này thay vì alert trực tiếp trong hàm thì ta trả về kết quả true/false, sau đó kiểm tra kết quả này nếu true thì là năm nhuận, nếu false thì không phải là năm nhuận.

**4. Lời kết**

Qua bài này bạn phải hiểu được cách sử dụng hàm và tạo hàm trong javascript, hiểu được trong javascript không tồn tại khái niệm con trỏ và tham chiếu, hiểu được hai cách sử dụng hàm là hàm có return và hàm không có return. Chúc bạn học tốt!

**Bài 09: Một số ví dụ học javascript căn bản (Phần 1)**

Đăng bởi: [TheHalfHeart](http://freetuts.net/author/thehalfheart)- Vào ngày: 11-01-2015- Chuyên mục: [Javascript](http://freetuts.net/hoc-javascript)- View: 8680

**Mục luc:**

* [1. Bài tập tìm số lớn - nhỏ nhất với javascript](http://freetuts.net/mot-so-vi-du-hoc-javascript-can-ban-phan-1-278.html#goto-h2-0)
* [2. Bài tập giải phương trình bậc nhất với javascript](http://freetuts.net/mot-so-vi-du-hoc-javascript-can-ban-phan-1-278.html#goto-h2-1)
* [3. Bài tập kiểm tra số chính phương với javascript](http://freetuts.net/mot-so-vi-du-hoc-javascript-can-ban-phan-1-278.html#goto-h2-2)
* [Lời kết:](http://freetuts.net/mot-so-vi-du-hoc-javascript-can-ban-phan-1-278.html#goto-h2-3)

Chúng ta đã học được 8 bài rồi nên trong bài này mình sẽ làm một số ví dụ dùng javascript để viết một số chương trình căn bản để các bạn nắm rõ hơn và ôn lại kiến thức cũ. Nội dung của bài chủ yếu sử dụng kiến thức ở các bài trước nên nếu bạn chư xem thì quay lại và đọc qua nhé.

**1. Bài tập tìm số lớn - nhỏ nhất với javascript**

Bài viết được đăng tại freetuts.net - thehalfheart@gmail.com

Ta sẽ viết một chương trình cho người dùng nhập các con số vào cho tới khi không nhập nữa, sau đó sẽ xuất thông tin ra số nào lớn nhất và số nào nhỏ nhất.

Để giải quyết bài này chúng ta sẽ có ý tưởng như sau:

* Khai báo hai biến min và max để lưu giá trị nhỏ nhất và lớn nhất
* Sử dụng hàm [prompt()](http://freetuts.net/ham-alert-confirm-prompt-trong-javascript-264.html) để lấy thông tin. Hàm này nếu ta nhấn hủy thì sẽ trả về NULL
* Sử dụng vòng lặp while để yêu cầu người dùng nhập vào cho tới khi họ hủy. Vì khi click hủy sẽ có giá trị null nên điều kiện dừng của vòng lặp while là NULL
* Cuối cùng dùng lệnh document.write để in ra màn hình.

Các bạn xem code sau: [XEM DEMO](http://freetuts.net/editor.html?id=216)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38 | // Hai biến lưu min và max  var min = null;  var max = null;    // Biến lưu giá trị người dùng nhập vào  var value = '';    // trong khi người dùng chưa nhấn Hủy  // vì người dùng nhấn hủy tức là value = null,  // thì sẽ dừng vòng lặp  while (value != null)  {      // Lấy giá trị      value = prompt("Nhập số, nếu muốn dừng thì không nhập gì");        // Nếu người dùng nhấn hủy thì không thực thi những đoạn code bên trong      if (value != null)      {          // Nếu nhập lần đầu thì gán cả hai min và max bằng value          if (min == null){              min = value;              max = value;          }          else          {              // ngược lại sẽ kiểm tra gán min và max              if (min > value){                  min = value;              }              if (max < value){                  max = value;              }          }      }  }    // In ra trình duyệt  document.write("Min là " + min +", Max là " + max); |

**2. Bài tập giải phương trình bậc nhất với javascript**

Phương trình bậc nhất có dạng ax + b = 0. Như vậy ta có các trường hợp nghiệm như sau:

* Nếu a = 0 và b = 0 thì phương trình vô số nghiệm
* Nếu a # 0 và b = 0 thì nghiệm là x = 0
* Nếu a = 0 và b # 0 thì phương trình vô nghiệm
* Nếu a # 0 và b # 0 thì nghiệm là x = -b/a

Vì ta có hai số a, b là người dùng nhập vào, còn x là nghiệm nên ta sẽ viết chương trình cho người dùng nhập vào a, b rồi tính toán trên hai biến này. Các bạn xem code như sau:

[XEM DEMO](http://freetuts.net/editor.html?id=217)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | function giai\_pt\_bac\_nhat(a, b)  {      if (a == 0 && b == 0){          alert('Phương trình vô số nghiệm');      }      else if (a != 0 && b == 0){          alert('Phương trình có nghiệm x = 0');      }      else if (a == 0 && b != 0){          alert("Phương trình vô nghiệm");      }      else {          alert('Phương trình có nghiệm x = ' + (-b/a));      }  }    giai\_pt\_bac\_nhat(1,0); |

**3. Bài tập kiểm tra số chính phương với javascript**

**Định nghĩa**: Số chính phương là số có căn bậc hai là một số tự nhiên không có dấu chấm động. Ví dụ 4 là số chính phương vì căn bậc hai của 4 bằng 2. Còn 5 không phải là số chính phương vì căn bậc hai của nó là số có dư  phần dấu chấm động.

Như vậy để giải bài này ta chỉ cần kiểm tra căn bậc hai của nó có tròn hay không, nếu số tròn không dư phần số phẩy ở sau thì là số chính phương. Và để kiểm tra một số có dư hay không thì ta chia nó cho 1 là được, ví dụ 1.2 % 1 = 0.2

Các bạn xem bài giải sau: [XEM DEMO](http://freetuts.net/editor.html?id=218)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | function so\_chinh\_phuong(a)  {      // Ta dùng hàm Math.sqrt để lấy căn bậc 2      // sau đó chia lấy dư với 1, nếu bằng 0 thì là số chính phương,      // ngược lại thì không phải là số chính phương      if (Math.sqrt(a) % 1 == 0){          alert(a + " là số chính phương");      }      else {          alert(a + " không phải là số chính phương");      }  }    so\_chinh\_phuong(4);  so\_chinh\_phuong(12);  so\_chinh\_phuong(13); |

**Lời kết:**

Đêm khuya muộn rồi nên mình sẽ dừng ở đây, hy vọng qua ba bài tập javascript căn bản này sẽ giúp bạn ôn lại kiến thức javascript căn bản mà 8 bài trước chúng ta đã được học.

**Bài 10: Thao tác với mảng trong javascript**

Đăng bởi: [TheHalfHeart](http://freetuts.net/author/thehalfheart)- Vào ngày: 13-01-2015- Chuyên mục: [Javascript](http://freetuts.net/hoc-javascript)- View: 16154

**Mục luc:**

* [1. Khai báo mảng trong javascript](http://freetuts.net/thao-tac-voi-mang-trong-javascript-288.html#goto-h2-0)
  + [Khai báo với từ khóa new Arrray()](http://freetuts.net/thao-tac-voi-mang-trong-javascript-288.html#goto-h3-0)
  + [Khai báo với cặp dấu ngoặc vuông ([])](http://freetuts.net/thao-tac-voi-mang-trong-javascript-288.html#goto-h3-1)
* [2. Truy xuất các phần tử trong mảng](http://freetuts.net/thao-tac-voi-mang-trong-javascript-288.html#goto-h2-1)
* [3. In mảng ra trình duyệt và console](http://freetuts.net/thao-tac-voi-mang-trong-javascript-288.html#goto-h2-2)
* [4. Sử dụng vòng lặp để lặp mảng trong javascript](http://freetuts.net/thao-tac-voi-mang-trong-javascript-288.html#goto-h2-3)
  + [Lặp mảng với vòng lặp for](http://freetuts.net/thao-tac-voi-mang-trong-javascript-288.html#goto-h3-2)
  + [Lặp mảng với vòng lặp while](http://freetuts.net/thao-tac-voi-mang-trong-javascript-288.html#goto-h3-3)
* [5. Lời kết](http://freetuts.net/thao-tac-voi-mang-trong-javascript-288.html#goto-h2-4)

Việc xử lý mảng trong javascript cũng quan trọng như [xử lý mảng trong PHP](http://freetuts.net/cac-ham-xu-ly-mang-trong-php-21.html) vậy, bởi vì khi chúng ta dùng javascript để xử lý DOM thì kết quả mà chúng ta lấy được là một mảng các Element HTML và sẽ lặp element đó rồi xử lý. Chính vì điều này nên hôm nay mình viết một bài xử lý mảng trong javascript luôn để trình bày cho các bạn dễ nắm bắt hơn.

Trước hết chúng ta tìm hiểu khái niệm mảng là gì đã nhé. Mảng là một tập hợp các phần tử lại và mỗi phần tử sẽ được đánh dấu một vị trí trong tập hợp đó. Trong javascript nếu mảng có 10 phần tử thì các phần tử sẽ được đánh dấu từ 0 -9.

**1. Khai báo mảng trong javascript**

Bài viết được đăng tại freetuts.net - thehalfheart@gmail.com

Chúng ta có hai cách thông thường để khai báo mảng đó là sử dụng từ khóa new Array() và sử dụng cặp dấu ngoặc vuông ([]).

**Khai báo với từ khóa new Arrray()**

**Cú pháp**:

|  |  |
| --- | --- |
| 1  2  3 | var name\_array = new Array();  // Hoặc  var name\_array = new Array(1,2,3); |

Với cách hai thì ta khởi tạo giá trị lúc khai báo luôn.

**Khai báo với cặp dấu ngoặc vuông ([])**

**Cú pháp**:

|  |  |
| --- | --- |
| 1  2  3 | var name\_array = [];  // Hoặc  var name\_array = [1,2,3]; |

Tương tự cách hai là ta khai báo và khởi tạo giá trị luôn.

**2. Truy xuất các phần tử trong mảng**

Để truy xuất đến phần tử của một mảng chúng ta dùng cú pháp **tenmang[vitri]**. Ví dụ:

|  |  |
| --- | --- |
| 1  2  3  4 | var t = new Array(1,2,3);  alert(t[0]); // kết quả là 1  alert(t[1]); // kết quả là 2  alert(t[2]); // kết quả là 3 |

Như vậy phần tử đầu tiên sẽ có số chỉ mục là 0, phần tử thứ hai là 1, ... phần tử thứ n là n-1.

**3. In mảng ra trình duyệt và console**

**Hàm array.join**()

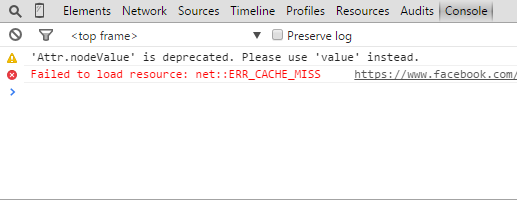
Để hiển thị các phần tử ra ngoài trình duyệt chúng ta sẽ sử dụng hàm array.join(). Ví dụ:

|  |  |
| --- | --- |
| 1  2  3 | var t = new Array(1,2,3);  document.write(t.join()); // kết quả 1,2,3  document.write(t.join('-')); // kết quả 1-2-3 |

Như vậy hàm join này được tích hợp vào đối tượng mảng trong javascript và nó có một tham số truyền vào, nếu ta không truyền gì vào thì mặc định nó lấy dấu phẩy để ngăn giữa các giá trị khi in lên trình duyệt.

**Hàm console.log()**:

Ngoài ra chúng ta có thêm một hàm rất hay dùng để debug đó là console.log().  Các bạn hãy cài đặt Firebug trên firefox hoặc dùng chế độ "kiểm tra phần tử" có sẵn trên trình duyệt (chrome tương tự), sau đó chuyển vào mục console như hình dưới đây:

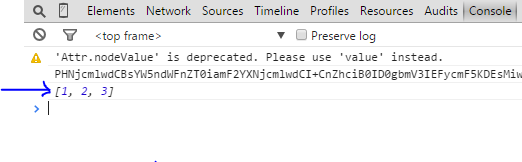


Nêu bạn code bị lỗi cú pháp hoặc một lỗi bất kì liên quan đến javascript thì nó sẽ hiển thị ở ô này nên bạn phải dùng nó để kiểm tra nếu viết mã javascript mà không thấy hoạt động nhé. Như trong hình chữ đỏ là lỗi, chữ vàng là cảnh báo.

Ví dụ:

|  |  |
| --- | --- |
| 1  2 | var t = new Array(1,2,3);  console.log(t); |

Chạy lên giao diện trong console sẽ như sau:



**4. Sử dụng vòng lặp để lặp mảng trong javascript**

Phần này là quan trọng nhất đấy nhé, chúng ta sẽ học cách sử dụng các vòng lặp đã học để lặp mảng. Nhưng trước tiên chúng ta tìm hiểu cách đếm tổng số phần tử trong mảng như sau:

Để đếm tổng số phần tử của một mảng chúng ta sẽ dùng thuộc tính length của nó. Ví dụ:

|  |  |
| --- | --- |
| 1  2 | var t = new Array(1,2,3);  alert(t.length);<br> |

**Lặp mảng với vòng lặp for**

Để lặp mảng với [vòng lặp for](http://freetuts.net/vong-lap-for-trong-javascript-270.html) thì chúng ta phải dùng thuộc tính length như trên để đếm tổng số phần tử, sau đó ở mỗi vòng lặp chúng ta sử dụng cú pháp truy xuất đến phần tử của mảng ở phần 2 để xử lý. Ví dụ:

|  |  |
| --- | --- |
| 1  2  3  4 | var name\_array = [1,2,3];  for (var i = 0; i < name\_array.length; i++){      document.write(name\_array[i]);  } |

**Lặp mảng với vòng lặp while**

Tương tự để lặp với [vòng lặp while](http://freetuts.net/vong-lap-while-do-while-trong-javascript-271.html)  chúng ta sẽ khai báo một biến index để lưu vị trí đang lặp. Ví dụ:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | var name\_array = [1,2,3];  var index = 0;  while (index < name\_array.length){      document.write(name\_array[index]);      index++;  } |

Riêng đối với vòng lặp do while thì không khuyến khích sử dụng để lặp mảng, lý do nó vòng lặp do while luôn luôn lặp ít nhất một lần nên trong trường hợp mảng cần lặp rỗng thì sẽ bị báo lỗi ngay.

**5. Lời kết**

Bài này trọng tâm là cách dùng vòng lặp for để lặp mảng vì khi làm ứng dụng thực tế đa số chúng ta sư dụng vòng lặp for để lặp. Và một lưu ý nữa là không nên dùng vòng lặp do while để lặp mảng nhé, lý do thì trong bài đã đề cập tới rồi. Bài tiếp theo chúng ta tìm hiểu một số hàm xử lý mảng trong javascript

**Bài 11: Các hàm xử lý mảng trong javascript**

Đăng bởi: [TheHalfHeart](http://freetuts.net/author/thehalfheart)- Vào ngày: 14-04-2015- Chuyên mục: [Javascript](http://freetuts.net/hoc-javascript)- View: 13555

**Mục luc:**

* [1. Danh sách các hàm xử lý mảng trong javascript](http://freetuts.net/cac-ham-xu-ly-mang-trong-javascript-289.html#goto-h2-0)
  + [Hàm array.valueOf()](http://freetuts.net/cac-ham-xu-ly-mang-trong-javascript-289.html#goto-h3-0)
  + [Hàm array.push()](http://freetuts.net/cac-ham-xu-ly-mang-trong-javascript-289.html#goto-h3-1)
  + [Hàm array.pop()](http://freetuts.net/cac-ham-xu-ly-mang-trong-javascript-289.html#goto-h3-2)
  + [Hàm array.shift()](http://freetuts.net/cac-ham-xu-ly-mang-trong-javascript-289.html#goto-h3-3)
  + [Hàm array.unshift()](http://freetuts.net/cac-ham-xu-ly-mang-trong-javascript-289.html#goto-h3-4)
  + [Hàm array.splice()](http://freetuts.net/cac-ham-xu-ly-mang-trong-javascript-289.html#goto-h3-5)
  + [Hàm array.sort()](http://freetuts.net/cac-ham-xu-ly-mang-trong-javascript-289.html#goto-h3-6)
  + [Hàm array.reverse()](http://freetuts.net/cac-ham-xu-ly-mang-trong-javascript-289.html#goto-h3-7)
  + [Hàm array.concat()](http://freetuts.net/cac-ham-xu-ly-mang-trong-javascript-289.html#goto-h3-8)
  + [Hàm array.slice()](http://freetuts.net/cac-ham-xu-ly-mang-trong-javascript-289.html#goto-h3-9)
* [2. Lời kết](http://freetuts.net/cac-ham-xu-ly-mang-trong-javascript-289.html#goto-h2-1)

Việc xử lý mảng trong javascript đóng vai trò rất quan trọng vì nó được sử dụng khá nhiều trong thực tế.  Chính vì vậy việc biết một số hàm xử lý mảng thông dụng trong javascript sẽ giúp bạn dễ dàng học cũng như tìm hiểu javascript hơn.

Mảng thực chất cũng là một Object (đối tượng) nên thực chất các hàm mà ta hay gọi chính là các phương thức (method) của đối tượng mảng, vì vậy bạn đừng nhầm lẫn giữa hai khái niệm nhé. Trong nội dung bài này mình sẽ gọi là các hàm xử lý mảng để tiện cho việc diễn giải hơn.

**1. Danh sách các hàm xử lý mảng trong javascript**

Bài viết được đăng tại freetuts.net - thehalfheart@gmail.com

Phần này chúng ta sẽ tìm hiểu một số hàm có sẵn trong javascript được tích hợp vào array object, chính vì nó chỉ dùng trong array object nên nếu bạn dùng với kiểu dữ liệu khác sẽ bị sai đấy nhé.

**Hàm array.valueOf()**

Hàm này có tác dụng tương tự như hàm array.join() mà ta đã học ở bài trước, có nghĩa là nó sẽ nối các phần tử với nhau vào một chuỗi cách nhau bởi dấu phẩy.

**Ví dụ**: [XEM DEMO](http://freetuts.net/editor.html?id=271)

|  |  |
| --- | --- |
| 1  2 | var mang = ["Học", "lập", "trình", "tại", "freetuts.net"];  document.write(mang.valueOf()); |

**Hàm array.push()**

Hàm thêm một phần tử vào cuối mảng.

**Ví dụ**: [XEM DEMO](http://freetuts.net/editor.html?id=272)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | var mang = ["Học", "lập", "trình", "tại", "freetuts.net"];    // in mảng  document.write(mang.valueOf());  document.write('<br/>');    // Thêm và in  mang.push("Miễn phí");  document.write(mang.valueOf()); |

**Hàm array.pop()**

Ngược với hàm array.push(), hàm này có tác dụng xóa đi phần tử cuối cùng trong mảng.

**Ví dụ**: [XEM DEMO](http://freetuts.net/editor.html?id=273)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | var mang = ["Học", "lập", "trình", "tại", "freetuts.net"];    // in mảng  document.write(mang.valueOf());  document.write('<br/>');    // Thêm và in  mang.pop();  document.write(mang.valueOf()); |

**Hàm array.shift()**

Hàm xóa phần tử đầu tiên của mảng, sau đó dồn các phần tử phía sau xuống một bậc.

**Ví dụ**: [XEM DEMO](http://freetuts.net/editor.html?id=274)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | var mang = ["Học", "lập", "trình", "tại", "freetuts.net"];    // in mảng  document.write(mang.valueOf());  document.write('<br/>');    // Thêm và in  mang.shift();  document.write(mang.valueOf()); |

**Hàm array.unshift()**

Thêm một phần tử vào vị trí đầu tiên của mảng, đồng thời đẩy các phẩn từ phía sau lên một bậc.

**Ví dụ**: [XEM DEMO](http://freetuts.net/editor.html?id=275)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | var mang = ["Học", "lập", "trình", "tại", "freetuts.net"];    // in mảng  document.write(mang.valueOf());  document.write('<br/>');    // Thêm và in  mang.unshift("Chào Mừng");  document.write(mang.valueOf()); |

**Hàm array.splice()**

Hàm splice() có ba tham số truyền vào như sau: splice(position\_add, num\_element\_remove, value1, value2, ...).

**Trong đó**:

* **position\_add** là vị trí sẽ thêm (vị trí đầu tiên là 0)
* **num\_element\_remove** là số phần tử sẽ xóa (bắt đầu từ position\_add)
* **value1**, **value2**, .. là danh sách các phần tử sẽ được thêm vào sau khi tại vị trí position\_add và sau khi remove num\_element\_remove phần tử.

**Ví dụ 1**: [XEM DEMO](http://freetuts.net/editor.html?id=276)

|  |  |
| --- | --- |
| 1  2  3  4 | var mang = ["Học", "lập", "trình", "tại", "freetuts.net"];    mang.splice(1, 2, 'PHP', 'căn bản ');  document.write(mang.valueOf()); |

Trong ví dụ này thì:

* Vị trí thêm là số 1 (phần tử có giá trị là "lập")
* Xóa 2 phần tử liên tiếp từ vị trí 1 (xóa phần tử "lập" và "trình")
* Thêm hai phần tử "php" và "căn bản" vào

Cuối cùng ta có được một mảng gồm ["Học", "php", "căn bản", "tại", "freetuts.net"].

**Ví dụ 2**: [XEM DEMO](http://freetuts.net/editor.html?id=277)

Trong ví dụ này ta sẽ không thêm phần tử nữa mà sẽ lợi dụng hàm này để xóa đi một số phần tử.

|  |  |
| --- | --- |
| 1  2  3  4 | var mang = ["Học", "lập", "trình", "tại", "freetuts.net"];  // xóa phần tử "lập" và "trình"  mang.splice(1, 2);  document.write(mang.valueOf()); |

**Hàm array.sort()**

Hàm này dùng để sắp xếp các phần tử trong mảng theo thứ tự chữ cái alpha.

**Ví dụ**: [XEM DEMO](http://freetuts.net/editor.html?id=278)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | var mang = ["Học", "lập", "trình", "tại", "freetuts.net"];  document.write(mang.valueOf());  document.write('<br/>');    // Sắp xếp lại  mang.sort();  document.write(mang.valueOf()); |

**Hàm array.reverse()**

Hàm đảo ngược các phẩn tử lại. Vị trí đầu sẽ được chuyển xuống cuối mảng và vị trí cuối mảng sẽ được chuyển lên đầu mảng.

**Ví dụ**: [XEM DEMO](http://freetuts.net/editor.html?id=279)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | var mang = ["Học", "lập", "trình", "tại", "freetuts.net"];  document.write(mang.valueOf());  document.write('<br/>');    // Đảo ngược vị trí của các phần tử  mang.reverse();  document.write(mang.valueOf()); |

**Hàm array.concat()**

Hàm dùng để nối hai mảng với nhau và trả về một mảng gồm tổng số phần tử của hai mảng đó.

**Ví dụ**: [XEM DEMO](http://freetuts.net/editor.html?id=280)

|  |  |
| --- | --- |
| 1 | var mang1 = ["Học", "lập", "trình"];<br>var mang2 = ["tại", "freetuts.net"];<br><br>// Nối mảng<br>var mang\_con = mang1.concat(mang2);<br>document.write(mang\_con.valueOf()); |

**Hàm array.slice()**

Hàm dùng để lấy một số phần tử con trong mảng. Có hai tham số truyền vào như sau: slice(start, end).

**Trong đó**:

* **start**: là vị trí bắt đầu
* **end**: là vị trí kết thúc

Lưu ý: Để dễ hiểu thì start sẽ phần tử đầu tiên là 0 và end sẽ tính phần tử đầu tiên là 1. Chính vì vậy bạn sẽ phải cộng thêm 1 ở end thì mới lấy đúng phần tử mong muốn

**Ví dụ**: [XEM DEMO](http://freetuts.net/editor.html?id=281)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | var mang = ["Học", "lập", "trình", "tại", "freetuts.net"];    // Lấy phần tử "tại" và "freetuts.net"  var mang\_moi = mang.slice(3, 5);    // In ra thử  document.write(mang\_moi.valueOf()); |

Trường hợp bạn muốn lấy từ vị trí nào đó đến cuối mảng thì bạn sẽ truyền một tham số thôi.

**Ví dụ**: [XEM DEMO](http://freetuts.net/editor.html?id=282)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | <!DOCTYPE html>  <html>      <body>          <script language="javascript">              var mang = ["Học", "lập", "trình", "tại", "freetuts.net"];                // Lấy phần tử "tại" và "freetuts.net"              var mang\_moi = mang.slice(3);                // In ra thử              document.write(mang\_moi.valueOf());          </script>      </body>  </html> |

**2. Lời kết**

Trên là danh sách **các hàm xử lý mảng trong javascript** mà ta hay sử dụng nhất, vẫn còn khá nhiều hàm nên mình không thể liệt kê hết được. Sau này học nâng cao lên bạn có thể tự định nghĩa thêm các hàm riêng cho mình bằng cách sử dụng prototype.

**Bài 12: DOM là gì? Các loại DOM trong Javascript**

Đăng bởi: [TheHalfHeart](http://freetuts.net/author/thehalfheart)- Vào ngày: 15-04-2015- Chuyên mục: [Javascript](http://freetuts.net/hoc-javascript)- View: 12983

**Mục luc:**

* [1. DOM là gì?](http://freetuts.net/dom-la-gi-cac-loai-dom-trong-javascript-366.html#goto-h2-0)
* [2. Các thể loại DOM trong Javascript](http://freetuts.net/dom-la-gi-cac-loai-dom-trong-javascript-366.html#goto-h2-1)
* [3. Lời kết](http://freetuts.net/dom-la-gi-cac-loai-dom-trong-javascript-366.html#goto-h2-2)

Javascript là một ngôn ngữ được sử dụng trong các trình duyệt Browser nên nó đóng một vai trò khá quan trọng trong các ứng dụng website. Và nhiệm vụ của Javascript là thao tác với các tài liệu HTML kết hợp với các cú pháp riêng của nó để tạo nên sự ảo diệu của trang web.

Để thao tác được với các thẻ HTML thì nó phải thông qua một cơ chế ta gọi là DOM và ta hay gọi là **D**ocument **O**bject **M**odel. Vậy thì DOM là gì thì chúng ta cùng tì hiểu nó nhé.

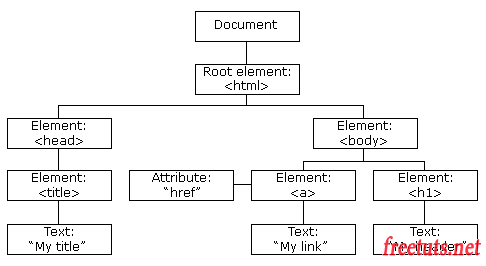
**1. DOM là gì?**

Bài viết được đăng tại freetuts.net - thehalfheart@gmail.com

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.

Như các bạn biết trong mỗi thẻ HTML sẽ có những thuộc tính (Properties) và có phân cấp cha - con với các thẻ HTML khác. Sự phân cấp và các thuộc tính của thẻ HTML này ta gọi là[selector](http://freetuts.net/selector-la-gi-tim-hieu-css-selector-can-ban-345.html) và trong DOM sẽ có nhiệm vụ xử lý các vấn đề như đổi thuộc tính của thẻ, đổi cấu trúc HTML của thẻ, ...

Bạn có thể tham khảo hình vẽ dưới đây để hiểu rõ hơn về DOM.



Trong hình thì tất cả các thẻ HTML sẽ được quản lý trong đối tượng **document**, thẻ cao nhất là thẻ **html**, tiếp theo là phân nhánh **body**và **head**. Bên trong **head**thì có những thẻ như **style**,**title**, ... và bên trong **body**thì là vô số các thẻ HTML khác.

Như vậy trong Javascript để thao tác với các thẻ HTML ta phải thông qua đối tượng document nhé các bạn, để cho nóng thì mình làm một ví dụ trước rồi tìm hiểu sau :D.

**Ví dụ**: [XEM DEMO](http://freetuts.net/editor.html?id=283)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | <html>    <body>      <h1 id="main-content"></h1>      <script language="javascript">        document.getElementById("main-content").innerHTML = "Chào mừng các bạn đến với website học lập trình online freetuts.net"      </script>    </body>  </html> |

Trong ví dụ này mình có sử dụng một đoạn code xử lý javascript như sau:

|  |  |
| --- | --- |
| 1 | document.getElementById("main-content").innerHTML = "Chào mừng các bạn đến với website học lập trình online freetuts.net" |

Và đoạn code này có ý nghĩa rằng tìm thẻ có id="main-content"và gán nội dung HTML bên trong của thẻ này là dòng chữ "*Chào mừng các bạn đến với website học lập trình online freetuts.net*".

**2. Các thể loại DOM trong Javascript**

Việc xử lý và làm việc với đối tượng HTML rất phức tạp và đa dạng, chính vì vậy javascript có cung cấp cho chúng ta nhiều phương thức, đối tượng và mỗi thành phần như vậy sẽ có những nhiệm vụ riêng biệt. Sau đây mình sẽ liệt kê danh sách chia nhóm và trong những bài học tiếp theo chúng ta sẽ cùng tìm hiểu nó nhé.

**Danh sách chia nhóm DOM**:

* 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 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, Nodelist: có nhiệm vụ thao tác với HTML thông qua đối tượng (Object)

Như vậy ta có tổng cộng 8 loại thường được sử dụng như trên, và trong bài này mình không đề cập chi tiết vào nhé các bạn.

**3. Lời kết**

Việc xử lý các thẻ HTML rất là ảo diệu và tuyệt vời cho nên website nào cũng phải sử dụng nó. Trong thực tế các Libraries như jQuery vẫn sử dụng các thể loại DOM này , ví dụ sự kiện click(), hover() thì bản chất nó chính là thuộc nhóm Event + Listener.

**Bài 13: DOM Element trong javascript**

Đăng bởi: [TheHalfHeart](http://freetuts.net/author/thehalfheart)- Vào ngày: 15-04-2015- Chuyên mục: [Javascript](http://freetuts.net/hoc-javascript)- View: 10011

**Mục luc:**

* [1. Tìm thẻ HTML theo ID](http://freetuts.net/dom-element-trong-javascript-367.html#goto-h2-0)
* [2. Tìm thẻ HTML theo tên của thẻ HTML](http://freetuts.net/dom-element-trong-javascript-367.html#goto-h2-1)
* [3. Tìm thẻ HTML theo tên class](http://freetuts.net/dom-element-trong-javascript-367.html#goto-h2-2)
* [4. Tìm thẻ HTML theo cú pháp của Selector CSS](http://freetuts.net/dom-element-trong-javascript-367.html#goto-h2-3)
* [5. Lời kết](http://freetuts.net/dom-element-trong-javascript-367.html#goto-h2-4)

Trong bài này chúng ta sẽ tìm hiểu đến **DOM Element** và sẽ học cách truy xuất đến một thẻ HTML bất kì thông qua các thuộc tính như id, class, name hay thậm chí là tên của thẻ HTML và cú pháp CSS Selector .

Nội dung của bài này chúng ta tìm hiểu các vấn đề sau:

* Tìm thẻ HTML theo ID
* Tìm thẻ HTML theo class
* Tìm thẻ HTML theo tên thẻ
* Tìm thẻ HTML theo selector

**1. Tìm thẻ HTML theo ID**

Bài viết được đăng tại freetuts.net - thehalfheart@gmail.com

Để truy xuất tới một thẻ HTML theo ID ta sử dụng cú pháp sau:

|  |  |
| --- | --- |
| 1 | var element = document.getElementById('idname'); |

**Ví dụ**: [XEM DEMO](http://freetuts.net/editor.html?id=284)

|  |  |
| --- | --- |
| 1  2  3  4  5 | // Lấy thẻ input  var element = document.getElementById('website');    // Lấy giá trị của thẻ input  document.write(element.value); |

**2. Tìm thẻ HTML theo tên của thẻ HTML**

Tên thẻ HTML chính là tên các thẻ như p, a, div, ... Và ta sẽ truy xuất tới nó bằng cú pháp sau:

|  |  |
| --- | --- |
| 1 | var element = document.getElementsByTagName('tagname'); |

**Ví dụ**: [XEM DEMO](http://freetuts.net/editor.html?id=285)

|  |  |
| --- | --- |
| 1  2  3  4  5 | // Lấy thẻ input  var element = document.getElementsByTagName('input');    // Lấy giá trị của thẻ input  document.write(element[0].value); |

Trong ví dụ này có sự khác biệt đó là đoạn code element[0].value. Tại sao lại như vậy? Như bạn biết trong một trang web có thể có nhiều thẻ HTML giống nhau (ví dụ có hai thẻ div) nên hàm getElementsByTagName() sẽ **trả về một mảng các object** (xem bài [mảng trong javascript](http://freetuts.net/thao-tac-voi-mang-trong-javascript-288.html)) chứ không phải là một object nữa, chính vì vậy ta sẽ lấy input thứ nhất nên truyền số 0 vào.

**3. Tìm thẻ HTML theo tên class**

Để tìm các thẻ HTML có class nào đó thì ta dùng cú pháp sau:

|  |  |
| --- | --- |
| 1 | var element = document.getElementsByClassName('input'); |

Tương tự như tìm theo tên thẻ HTML thì tìm theo tên class sẽ trả về một mảng các object nên bạn sẽ phải sử dụng cú pháp truy xuất mảng để chọn đúng đối tượng muốn lấy.

**Ví dụ**: XEM DEMO

|  |  |
| --- | --- |
| 1  2  3  4  5 | // Lấy thẻ input  var element = document.getElementsByClassName('website');    // Lấy giá trị của thẻ input  document.write(element[0].value); |

**4. Tìm thẻ HTML theo cú pháp của Selector CSS**

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

Và đây là cú pháp sử dụng:

|  |  |
| --- | --- |
| 1 | var element = document.querySelectorAll("selector.css"); |

**Ví dụ**: Tôi có một đoạn mã HTML như sau:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | <html>    <body>      <input type="text" value="thẻ không cần lấy" class="website"/>      <div>          <input type="text" value="Thẻ Cần Lấy" class="website"/>          <input type="text" value="thẻ không cần lấy"/>      </div>    </body>  </html> |

Câu hỏi đặt ra là làm thế nào có thể chọn đúng một thẻ input nằm trong thẻ div và có class="website"?

Trước tiên ta quay lại chút với CSS Selector đã nhé. Trong CSS để chọn thẻ input nằm trong thẻdiv và có class="website" thì cú pháp là:

|  |  |
| --- | --- |
| 1 | div input.website |

Như vậy bài giải sẽ như sau: [XEM DEMO](http://freetuts.net/editor.html?id=287)

|  |  |
| --- | --- |
| 1 | var element = document.querySelectorAll("div input.website"); |

Và có một lưu ý tương tự là kết quả sẽ trả về một mảng các Objects.

**5. Lời kết**

Trên là 4 cách thông thường chúng ta hay sử dụng trong javascript để truy xuất tới một thẻ HTML bất kì. Nếu sau này làm việc với jQuery thì bạn không cần phải nhọc nhằn như thế nữa mà nó có những phương thức ngắn gọn hơn rất nhiều.