câu lệnh trong javascript

### [Giáo trình JavaScript: Các lệnh trong JavaScript - VOER](https://www.google.com.vn/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwic3YWGof3SAhUDO48KHUyWDUkQFggZMAA&url=https%3A%2F%2Fvoer.edu.vn%2Fc%2Fcac-lenh-trong-javascript%2F9aef1b7a%2F3ff66a56&usg=AFQjCNGJJReRsBmcZNMdGf7syAZDpRRXrw&sig2=BXEFxYSlGnf47MQe3wHePA)

https://voer.edu.vn/c/cac-lenh-trong-javascript/9aef1b7a/3ff66a56

**Câu lệnh** điều kiện cho phép chương trình ra quyết định và thực hiện công việc nào đấy dựa trên kết quả của quyết định. **Trong JavaScript**, **câu lệnh** điều kiện là ...

Bạn đã truy cập trang này 2 lần. Lần truy cập cuối: 05/03/2017

### [Hàm trong JavaScript | 208 bài học Javascript miễn phí hay nhất](http://vietjack.com/javascript/ham_trong_javascript.jsp)

vietjack.com/javascript/ham\_trong\_javascript.jsp

**Hàm trong JavaScript** - Học Javascript cơ bản và nâng cao cho người mới học từ Cú pháp Đối tượng Syntax Objects Form Validations Cookies Regular ...

Bạn đã truy cập trang này.

### **[PDF]**[Cú pháp và câu lệnh trong JavaScript](http://www.poly.edu.vn/book/4_Java%20Chap_3.pdf)

www.poly.edu.vn/book/4\_Java%20Chap\_3.pdf

Cú pháp và **câu lệnh trong JavaScript**. Sau khi đọc xong chương này, bạn có thể: □. Nắm được các nguyên tắc cơ bản khi sử dụng ngôn ngữ lập trình ...

Bạn đã truy cập trang này vào ngày 05/03/2017.

### [Thủ thuật javacript : Nhúng JavaScript vào file HTML - Thủ Thuật Web](https://www.thuthuatweb.net/javacript/thu-thuat-javacript-nhung-javascript-vao-file-html.html)

https://www.thuthuatweb.net › WordPress › BlogSpot › HTML5 › Photoshop

1 thg 6, 2012 - Bạn có thể nhúng **JavaScript** vào một file HTML theo một **trong** các cách sau đây: • Sử dụng các **câu lệnh** và các hàm **trong** cặp thẻ <SCRIPT>

Bạn đã truy cập trang này 2 lần. Lần truy cập cuối: 19/03/2017

### [Cùng Học Web 123: Các lệnh cơ bản trong JavaScript [JavaScript]](https://www.google.com.vn/url?sa=t&rct=j&q=&esrc=s&source=web&cd=5&cad=rja&uact=8&ved=0ahUKEwic3YWGof3SAhUDO48KHUyWDUkQFgg1MAQ&url=http%3A%2F%2Fcunghocweb123.blogspot.com%2F2013%2F07%2Fcac-lenh-co-ban-trong-javascript.html&usg=AFQjCNG-tUZ5fn3QvQEi7WYxiN-xutj9fA&sig2=1VZ96_AMB7lp4-Pe7wWUog)

cunghocweb123.blogspot.com/2013/07/cac-lenh-co-ban-trong-javascript.html

18 thg 7, 2013 - Thông thường mã **lệnh JavaScript** bắt đầu bằng thẻ <script ... Tuy nhiên, chúng ta vẫn có thể nhúng mã **JavaScript** bên **trong** thẻ <body> Ví dụ:

### [Hàm và tạo hàm (function) trong javascript - Freetuts](http://freetuts.net/ham-va-tao-ham-function-trong-javascript-274.html)

freetuts.net › Javascript › Javascript căn bản

10 thg 1, 2015 - Tìm hiểu **Hàm** và tạo **hàm** (function) **trong javascript**, ví dụ demo xây dựng **hàm trong javascript** và hai cách ... **Câu** hỏi và bài tập thực hành.

### [Lệnh kiểm tra điều kiện if else trong javascript - Freetuts](http://freetuts.net/lenh-kiem-tra-dieu-kien-if-else-trong-javascript-267.html)

freetuts.net › Javascript › Javascript căn bản

6 thg 1, 2015 - **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 ...

### [Hàm alert() - confirm() - prompt() trong javascript - Freetuts](http://freetuts.net/ham-alert-confirm-prompt-trong-javascript-264.html)

freetuts.net › Javascript › Javascript căn bản

6 thg 1, 2015 - Tìm hiểu các **Hàm** alert() - confirm() - prompt() **trong javascript**,thông ... **Hàm** prompt()**trong javascript**; Lời kết .... **Câu** hỏi và bài tập thực hành.

### [Bài tập cơ bản về JavaScript - Forum](http://k709dhcl.forums-free.com/bai-tap-co-ban-ve-javascript-t596.html)

k709dhcl.forums-free.com/bai-tap-co-ban-ve-javascript-t596.html

Hướng dẫn: Dùng thẻ để tạo nút nhấn và thêm thuộc tính onClick = "<**Câu lệnh JavaScript**>;" (**Trong**đó <**Câu lệnh JavaScript**> có thể là một lệnh **JavaScript** bất ...

### [Câu lệnh trong JavaScript - Đức Anh Plus](http://www.ducanhplus.com/2015/01/cau-lenh-trong-javascript/)

www.ducanhplus.com/2015/01/cau-lenh-trong-javascript/

23 thg 1, 2015 - **Câu lệnh** dưới đây nói cho trình duyệt hãy ghi dòng “Blog Đức Anh Plus” vào phần tử HTML có id=”website”. Ví dụ: document.

### [Javascript cơ bản - Hành trình của một Anh Hùng - Thạch Phạm](https://thachpham.com/web-development/javascript/javascript-co-ban-hanh-trinh-cua-mot-anh-hung.html)

https://thachpham.com/.../javascript/javascript-co-ban-hanh-trinh-cua-mot-anh-hung....

Chuyển đến [6.Quest 6: Tạo và gọi **Hàm**](https://thachpham.com/web-development/javascript/javascript-co-ban-hanh-trinh-cua-mot-anh-hung.html#ftoc-heading-8) - Hàm có thể nhận tham số từ biến và trả kết quả về để gán ... hàm ngay lập tức, nên nếu có **câu lệnh** nào phía sau, ... Một lưu ý khi sử dụng hàm là đừng ôm đồm quá nhiều xử lý **trong** một hàm.

Bạn đã truy cập trang này 3 lần. Lần truy cập cuối: 19/03/2017

### [Câu lệnh javascript - Logo](http://congdongcoder.com/cau-lenh-javascript/)

congdongcoder.com/cau-lenh-javascript/

Bài tiếp theo : Hiển thị dữ liệu **trong Javascript** ». Mỗi **câu lệnh** javascript được đọc và thực thi bởi trình duyệt web. Một **câu lệnh** javascript có thể là. Lời khai báo ...

### [Hiển thị dữ liệu trong Javascript - Logo](http://congdongcoder.com/hien-thi-trong-javascript/)

congdongcoder.com/hien-thi-trong-javascript/

Hiển thị dữ liệu **trong Javascript**. ... Bài trước : **Câu lệnh** javascript ... Hiển thị trong thông báo, sử dụng window.alert(); Hiển thị trong đầu ra của HTML, sử dụng ...

### [Cách hiển thị dữ liệu ra màn hình trong JavaScript - Web cơ bản](http://webcoban.vn/javascript/cach-hien-thi-du-lieu-ra-man-hinh-trong-javascript.html)

webcoban.vn/javascript/cach-hien-thi-du-lieu-ra-man-hinh-trong-javascript.html

**JavaScript** không hỗ trợ hiển thị dữ liệu trực tiếp giống như **trong** HTML. Nếu muốn ... **Câu lệnh**alert() dùng để hiển thị dữ liệu bên **trong** một hộp thông báo.

### **[PDF]**[6. Javascript Basic.pdf - Khoa Công Nghệ Thông Tin | Đại học Tôn ...](http://it.tdt.edu.vn/~pnnam/BG_503023/BG_503023/6.%20Javascript%20Basic.pdf)

it.tdt.edu.vn/~pnnam/BG\_503023/BG\_503023/6.%20Javascript%20Basic.pdf

Cách 1 : Nhúng đoạn **javascript** vào **trong** file HTML. – Cách 2 ... Biến cục bộ : được khai báo **trong hàm**. .... **Hàm** eval(“Chuỗi”) : **hàm** trả về kết quả thực thi **câu**.

### [Bài 6 - Câu lệnh trong JavaScript - YouTube](https://www.youtube.com/watch?v=H4MgrifsVLw)

[**▶ 4:17**](https://www.youtube.com/watch?v=H4MgrifsVLw)

https://www.youtube.com/watch?v=H4MgrifsVLw

2 thg 8, 2016 - Tải lên bởi Tin học Hữu Hoàng

**Câu lệnh trong JavaScript** - Trong HTML, câu lệnh JavaScript là chỉ dẫn được thực hiện bởi trình duyệt web - Các câu lệnh ...

### [JavaScript – Wikipedia tiếng Việt](https://vi.wikipedia.org/wiki/JavaScript)

https://vi.wikipedia.org/wiki/JavaScript

Chuyển đến [**Hàm**](https://vi.wikipedia.org/wiki/JavaScript#H.C3.A0m) - Hàm là một khối các **câu lệnh** với một danh sách một hoặc nhiều đối số (có thể không có đối số) và thường có tên (mặc dù **trong** ...

### [Tu hoc javascript - SlideShare](https://www.slideshare.net/zingoncmu2/tu-hoc-javascript)

https://www.slideshare.net/zingoncmu2/tu-hoc-javascript

15 thg 1, 2015 - 3. Sử dụng **Javascript** như thế nào? Để sử dụng JS rất đơn giản, chúng ta chỉ cần đưa các **câu lệnh** của JS vào **trong** thẻ <script> của HTML Ví ...

### [Câu lệnh điều kiện if else trong JavaScript - Đức Anh Plus](http://www.ducanhplus.com/2016/07/cau-lenh-if-else-trong-javascript/)

www.ducanhplus.com/2016/07/cau-lenh-if-else-trong-javascript/

9 thg 7, 2016 - Bạn có thể sử dụng **câu lệnh** điều kiện **trong** đoạn code của bạn để làm ... Sử dụng**câu lệnh** if để chỉ định khối code **JavaScript** cần thực thi ...

### [Javascript - Những khó hiểu trong câu lệnh điều kiện if và phép toán ...](https://kipalog.com/posts/Javascript---Nhung-kho-hieu-trong-cau-lenh-dieu-kien-if-va-phep-toan-so-sanh)

https://kipalog.com/.../Javascript---Nhung-kho-hieu-trong-cau-lenh-dieu-kien-if-va-p...

4 thg 3, 2016 - Một số vấn đề Theo cá nhân tôi, **Javascript** có lẽ là một **trong** những ngôn ngữ dễ học, dễ viết nhất. Một web developer mới bắt đầu có lẽ chỉ ...

# **GIÁO TRÌNH JAVASCRIPT**

SCIENCE AND TECHNOLOGY

## **Các lệnh trong JavaScript**

Có thể chia các lệnh của JavaScript thành ba nhóm sau:

* Lệnh điều kiện.
* Lệnh lặp.
* Lệnh tháo tác trên đối tượng.

# **Câu lệnh điều kiện**

Câu lệnh điều kiện cho phép chương trình ra quyết định và thực hiện công việc nào đấy dựa trên kết quả của quyết định. Trong JavaScript, câu lệnh điều kiện là **if...else**

if...else

Câu lệnh này cho phép bạn kiểm tra điều kiện và thực hiện một nhóm lệnh nào đấy dựa trên kết quả của điều kiện vừa kiểm tra. Nhóm lệnh sau **else** không bắt buộc phải có, nó cho phép chỉ ra nhóm lệnh phải thực hiện nếu điều kiện là sai.

**Cú pháp**

Cú pháp  
if ( <điều kiện> )  
{  
 //Các câu lệnh với điều kiện đúng  
}  
else  
{  
 //Các câu lệnh với điều kiện sai  
}

if (x==10){  
document.write("x bằng 10, đặt lại x bằng 0.");  
x = 0;}  
else  
document.write("x không bằng 10.");

Ký tự { và } được sử dụng để tách các khối mã.

# **Câu lệnh lặp**

Câu lệnh lặp thể hiện việc lặp đi lặp lại một đoạn mã cho đến khi biểu thức điều kiện được đánh giá là đúng. JavaScipt cung cấp hai kiểu câu lệnh lặp:

* for loop
* while loop

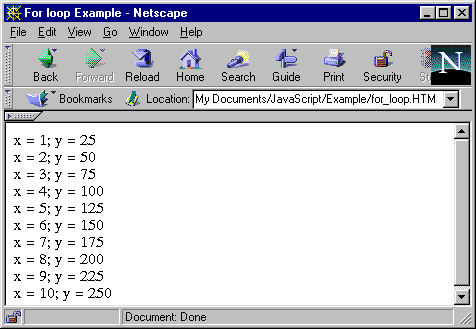
## **VÒNG LẶP FOR**

Vòng lặp for thiết lập một biểu thức khởi đầu - *initExpr*, sau đó lặp một đoạn mã cho đến khi biểu thức *<điều kiện>* được đánh giá là đúng. Sau khi kết thúc mỗi vòng lặp, biểu thức *incrExpr* được đánh giá lại.

**Cú pháp:**

for (initExpr; <điều kiện> ; incrExpr){  
 //Các lệnh được thực hiện trong khi lặp  
}

<HTML> <HEAD>  
<TITLE>For loop Example </TITLE>  
<SCRIPT LANGUAGE= "JavaScript">  
for (x=1; x<=10 ; x++) {  
 y=x\*25;  
 document.write("x ="+ x +";y= "+ y + "<BR>");  
}  
</SCRIPT>  
</HEAD>  
<BODY></BODY>  
</HTML>

Ví dụ này lưu vào file *for\_loop.Html.*

Vòng lặp này sẽ thực hiện khối mã lệnh cho đến khi x>10.

## **WHILE**

Vòng lặp while lặp khối lệnh chừng nào *<điều kiện>* cònđược đánh giá là đúng

**Cú pháp:**

while (<điều kiện>)  
{  
 //Các câu lệnh thực hiện trong khi lặp  
}

x=1;  
while (x<=10){  
 y=x\*25;  
 document.write("x="+x +"; y = "+ y + "<BR>");  
 x++;  
}

Kết quả của ví dụ này giống như ví dụ trước.

## **BREAK**

Câu lệnh **break** dùng để kết thúc việc thực hiện của vòng lặp **for** hay **while**. Chương trình được tiếp tục thực hiện tại câu lệnh ngay sau chỗ kết thúc của vòng lặp.

### **Cú pháp**

break;

Đoạn mã sau lặp cho đến khi x lớn hơn hoặc bằng 100. Tuy nhiên nếu giá trị x đưa vào vòng lặp nhỏ hơn 50, vòng lặp sẽ kết thúc

while (x<100)  
{  
 if (x<50) break;  
 x++;  
}

## **CONTINUE**

Lệnh **continue** giống lệnh **break** nhưng khác ở chỗ việc lặp được kết thúc và bắt đầu từ đầu vòng lặp. Đối với vòng lặp **while**, lệnh **continue** điều khiển quay lại <điều kiện>; với **for,** lệnh **continue** điều khiển quay lại incrExpr.

### **Cú pháp**

continue;

Đoạn mã sau tăng x từ 0 lên 5, nhảy lên 8 và tiếp tục tăng lên 10

x=0;  
while (x<=10)  
{  
 document.write(“Giá trị của x là:”+ x+”<BR>”);  
 if (x=5)  
 {  
 x=8;  
 continue;  
 }  
 x++;  
}

# **Các câu lệnh thao tác trên đối tượng**

JavaScript là một ngôn ngữ dựa trên đối tượng, do đó nó có một số câu lệnh làm việc với các đối tượng.

## **FOR...IN**

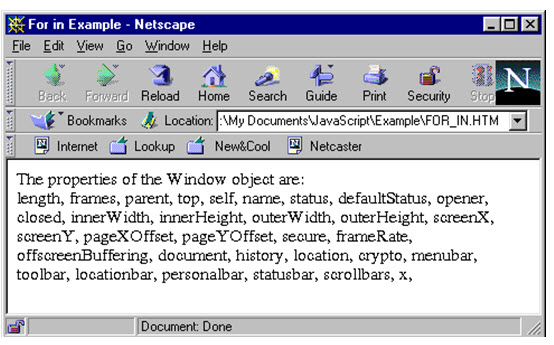
Câu lệnh này được sử dụng để lặp tất cả các thuộc tính (properties) của một đối tượng. Tên biến có thể là một giá trị bất kỳ, chỉ cần thiết khi bạn sử dụng các thuộc tính trong vòng lặp. Ví dụ sau sẽ minh hoạ điều này

**Cú pháp**

for (<variable> in <object>)  
{  
 //Các câu lệnh  
}

Ví dụ sau sẽ lấy ra tất cả các thuộc tính của đối tượng Window và in ra tên của mỗi thuộc tính. Kết quả được minh hoạ trên hình 2

<HTML>  
<HEAD>  
<TITLE>For in Example </TITLE>  
<SCRIPT LANGUAGE= "JavaScript">  
document.write("The properties of the Window object are: <BR>");  
for (var x in window)  
 document.write(" "+ x + ", ");  
</SCRIPT>  
</HEAD>  
<BODY>   
</BODY>  
</HTML>

*Kết quả của lệnh For..in*

## **NEW**

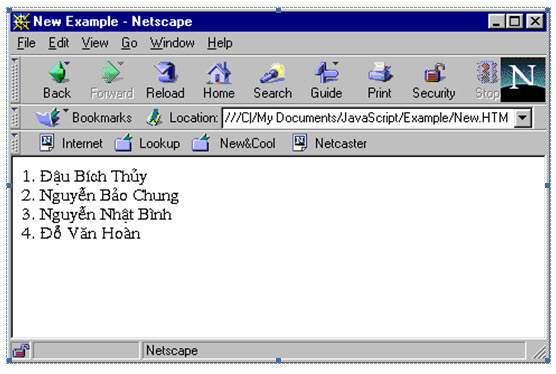
Biến **new** được thực hiện để tạo ra một thể hiện mới của một đối tượng

**Cú pháp:**

objectvar = new object\_type ( param1 [,param2]... [,paramN])

Ví dụ sau tạo đối tượng **person** có các thuộc tính *firstname, lastname, age, sex*. Chú ý rằng từ khoá **this** được sử dụng để chỉ đối tượng trong hàm **person**. Sau đó ba thể hiện của đối tượng **person** được tạo ra bằng lệnh **new**

<HTML>  
<HEAD>  
<TITLE>New Example </TITLE>  
<SCRIPT LANGUAGE= "JavaScript">  
function person(first\_name, last\_name, age, sex){  
 this.first\_name=first\_name;  
 this.last\_name=last\_name;  
 this.age=age;  
 this.sex=sex;  
}  
person1= new person("Thuy", "Dau Bich", "23", "Female");  
person2= new person("Chung", "Nguyen Bao", "24", "Male");  
person3= new person("Binh", "Nguyen Nhat", "24", "Male");  
person4= new person("Hoàn", "Đỗ Văn", "24", "Male");  
document.write ("1. "+person1.last\_name+" " + person1.first\_name + "<BR>" );  
document.write("2. "+person2.last\_name +" "+ person2.first\_name + "<BR>");  
document.write("3. "+ person3.last\_name +" "+ person3.first\_name + "<BR>");  
document.write("4. "+ person4.last\_name +" "+ person4.first\_name+"<BR>");  
</SCRIPT>  
</HEAD>  
<BODY>  
</BODY>  
 </HTML>

*Kết quả của ví dụ lệnh New*

## **THIS**

Từ khoá **this** được sử dụng để chỉ đối tượng hiện thời. Đối tượng được gọi thường là đối tượng hiện thời trong phương thức hoặc trong hàm.

**Cú pháp:**

this [.property]

Có thể xem ví dụ của lệnh new.

## **WITH**

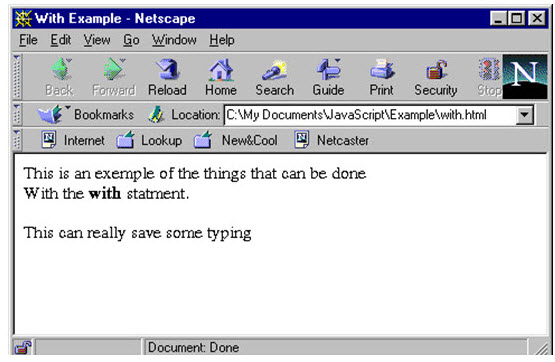
Lệnh này được sử dụng để thiết lập đối tượng ngầm định cho một nhóm các lệnh, bạn có thể sử dụng các thuộc tính mà không đề cập đến đối tượng.

**Cú pháp:**

with (object)  
{  
 // statement  
}

Ví dụ sau chỉ ra cách sử dụng lệnh with để thiết lập đối tượng ngầm định là **document** và có thể sử dụng phương thức**write** mà không cần đề cập đến đối tượng document

<HTML>   
<HEAD>  
<TITLE>With Example </TITLE>  
<SCRIPT LANGUAGE= "JavaScript">  
with (document){  
write(“This is an exemple of the things that can be done <BR>”);  
write(“With the <B>with<B> statment. <P>”);  
write(“This can really save some typing”);  
}  
</SCRIPT>  
</HEAD>  
<BODY>   
</HTML>

*Kết quả của ví dụ lệnh With*

# **Các hàm (Functions)**

JavaScript cũg cho phép sử dụng các hàm. Mặc dù không nhất thiết phải có, song các hàm có thể có một hay nhiều tham số truyền vào và một giá trị trả về. Bởi vì JavaScript là ngôn ngữ có tính định kiểu thấp nên không cần định nghĩa kiểu tham số và giá trị trả về của hàm. Hàm có thể là thuộc tính của một đối tượng, trong trường hợp này nó được xem như là phương thức của đối tượng đó.

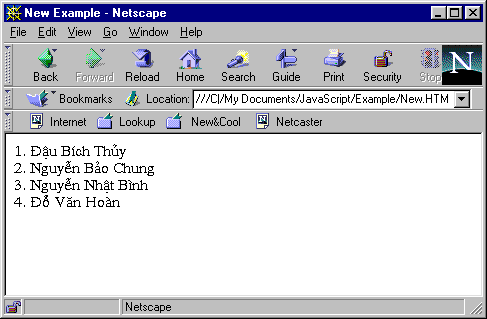
Lệnh **function**được sử dụng để tạo ra hàm trong JavaScript.

**Cú pháp:**

function fnName([param1],[param2],...,[paramN])  
{  
//function statement  
}

Ví dụ sau minh hoạ cách thức tạo ra và sử dụng hàm như là thành viên của một đối tượng. Hàm *printStats* được tạo ra là phương thức của đối tượng *person*

<HTML> <HEAD>  
<TITLE>Function Example </TITLE>  
<SCRIPT LANGUAGE= "JavaScript">  
function person(first\_name, last\_name, age, sex)  
{  
 this.first\_name=first\_name;  
 this.last\_name=last\_name;  
 this.age=age;  
 this.sex=sex;  
 this.printStats=printStats;  
}  
function printStats() {  
 with (document) {  
write (" Name :" + this.last\_name + " " + this.first\_name + "<BR>" );  
 write("Age :"+this.age+"<BR>");  
 write("Sex :"+this.sex+"<BR>");  
 }  
}  
person1= new person("Thuy", "Dau Bich", "23", "Female");  
person2= new person("Chung", "Nguyen Bao", "24", "Male");  
person3= new person("Binh", "Nguyen Nhat", "24", "Male");  
person4= new person("Hoan", "Do Van", "23", "Male");  
person1.printStats();  
person2.printStats();  
person3.printStats();  
person4.printStats();  
</SCRIPT>  
</HEAD>  
<BODY> </BODY>  
</HTML>

*Ví dụ về hàm*

# **Các hàm có sẵn**

JavaScript có một số hàm có sẵn, gắn trực tiếp vào chính ngôn ngữ và không nằm trong một đối tượng nào:

* eval
* parseInt
* parseFloat

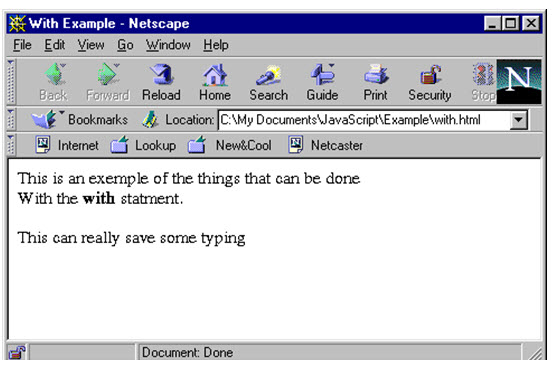
## **EVAL**

Hàm này được sử dụng để đánh giá các biểu thức hay lệnh. Biểu thức, lệnh hay các đối tượng của thuộc tính đều có thể được đánh giá. Đặc biệt hết sức hữu ích khi đánh giá các biểu thức do người dùng đưa vào (ngược lại có thể đánh giá trực tiếp).

**Cú pháp:**

returnval=eval(bất kỳ biểu thức hay lệnh hợp lệ trong Java)

<HTML>  
 <HEAD>  
<TITLE>Eval Example </TITLE>  
<SCRIPT LANGUAGE= "JavaScript">  
 var string="10+ Math.sqrt(64)";  
 document.write(string+ "="+ eval(string));  
</SCRIPT>  
</HEAD>  
<BODY>   
</BODY>  
</HTML>

*Ví dụ về hàm Eval*

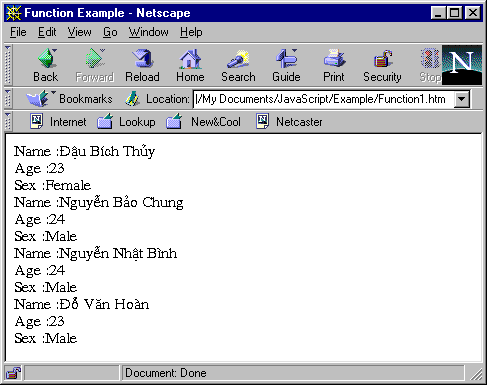
## **PARSEINT**

Hàm này chuyển một chuỗi số thành số nguyên với cơ số là tham số thứ hai (tham số này không bắt buộc). Hàm này thường được sử dụng để chuyển các số nguyên sang cơ số 10 và đảm bảo rằng các dữ liệu đưọc nhập dưới dạng ký tự được chuyển thành số trước khi tính toán. Trong trường hợp dữ liệu vào không hợp lệ, hàm parseInt sẽ đọc và chuyển dạng chuỗi đến vị trí nó tìm thấy ký tự không phải là số. Ngoài ra hàm này còn cắt dấu phẩy động.

**Cú pháp**

parseInt (string, [, radix])

<HTML>  
<HEAD>  
<TITLE> perseInt Exemple </TITLE>  
<SCRIPT LANGUAGE= "JavaScript">  
document.write("Converting 0xC hex to base-10: " + parseInt(0xC,10) + "<BR>");  
document.write("Converting 1100 binary to base-10: " + parseInt(1100,2) + "<BR>");  
</SCRIPT>  
</HEAD>  
<BODY>  
</BODY>  
</HTML>

*Ví dụ về Parseint*

## **PARSEFLOAT**

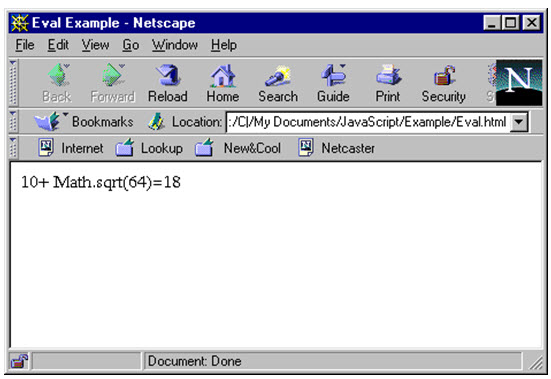
Hàm này giống hàm parseInt nhưng nó chuyển chuỗi thành số biểu diễn dưới dạng dấu phẩy động.

**Cú pháp**

parseFloat (string)

Ví dụ sau minh hoạ cách thức xử lý của parseFloat với các kiểu chuỗi khác nhau. Hình 5.8 minh họa kết quả

<HTML> <HEAD>  
<TITLE> perseFload Exemple </TITLE>  
<SCRIPT LANGUAGE= "JavaScript">  
document.write("This script will show how diffrent strings are ");  
document.write("Converted using parseFloat<BR>");  
document.write("137= " + parseFloat("137") + "<BR>");  
document.write("137abc= " + parseFloat("137abc") + "<BR>");  
document.write("abc137= " + parseFloat("abc137") + "<BR>");  
document.write("1abc37= " + parseFloat("1abc37") + "<BR>");  
</SCRIPT>  
</HEAD>  
<BODY> </BODY>  
</HTML>

*Kết quả của ví dụ hàm parsefloat*

# **Mảng (Array**

Mặc dù JavaScript không hỗ trợ cấu trúc dữ liệu mảng nhưng Netscape tạo ra phương thức cho phép bạn tự tạo ra các hàm khởi tạo mảng như sau:

function InitArray(NumElements){  
 this.length = numElements;  
 for (var x=1; x<=numElements; x++){  
 this[x]=0  
 }  
 return this;  
 }

Nó tạo ra một mảng với kích thước xác định trước và điền các giá trị 0. Chú ý rằng thành phần đầu tiên trong mảng là độ dài mảng và không được sử dụng.

Để tạo ra một mảng, khai báo như sau:

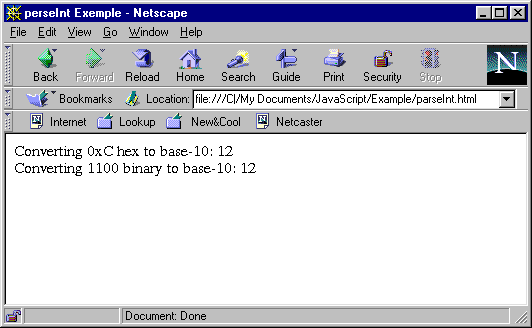
myArray = new InitArray (10)

Nó tạo ra các thành phần từ myArray[1] đến myArray[10] với giá trị là 0. Giá trị các thành phần trong mảng có thể được thay đổi như sau:

myArray[1] = "Nghệ An"  
myArray[2] = "Lào"

Sau đây là ví dụ đầy đủ:

<HTML> <HEAD>  
<TITLE> Array Exemple </TITLE>  
<SCRIPT LANGUAGE= "JavaScript">  
function InitArray(numElements) {  
 this.length = numElements;  
 for (var x=1; x<=numElements; x++){  
 this[x]=0  
 }  
 return this;  
}  
myArray = new InitArray(10);  
myArray[1] = "Nghệ An";  
myArray[2] = "Hà Nội";  
document.write(myArray[1] + "<BR>");  
document.write(myArray[2] + "<BR>");  
</SCRIPT>  
</HEAD>  
<BODY> </BODY>  
</HTML>

*Ví dụ về mảng*

# **Sự kiện**

JavaScript là ngôn ngữ định hướng sự kiện, nghĩa là sẽ phản ứng trước các sự kiện xác định trước như kích chuột hay tải một văn bản. Một sự kiện có thể gây ra việc thực hiện một đoạn mã lệnh (gọi là các chương triình xử lý sự kiện) giúp cho chương trình có thể phản ứng một cách thích hợp.

**Chương trình xử lý sự kiện (Event handler):**Một đoạn mã hay một hàm được thực hiện để phản ứng trước một sự kiện gọi là chương trình xử lý sự kiện. Chương trình xử lý sự kiện được xác định là một thuộc tính của một thẻ HTML:

<tagName eventHandler = "JavaScript Code or Function">

Ví dụ sau gọi hàm **CheckAge()** mỗi khi giá trị của trường văn bản thay đổi:

<INPUT TYPE=TEXT NAME="AGE" onChange="CheckAge()">

Đoạn mã của chương trình xử lý sự kiện không là một hàm; nó là các lệnh của JavaScript cách nhau bằng dấu chấm phẩy. Tuy nhiên cho mục đích viết thành các module nên viết dưới dạng các hàm.

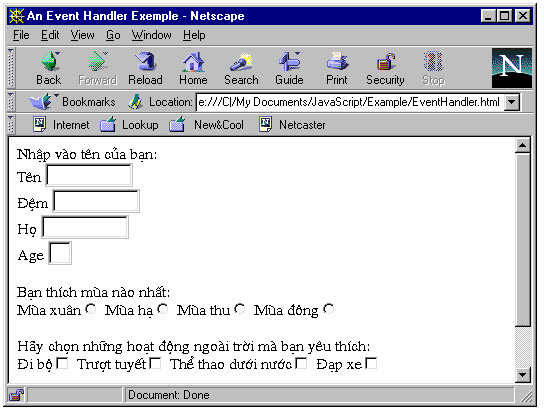
|  |  |
| --- | --- |
| onBlur | Xảy ra khi input focus bị xoá từ thành phần form |
| onClick | Xảy ra khi người dùng kích vào các thành phần hay liên kết của form. |
| onChange | Xảy ra khi giá trị của thành phần được chọn thay đổi |
| onFocus | Xảy ra khi thành phần của form được focus(làm nổi lên). |
| onLoad | Xảy ra trang Web được tải. |
| onMouseOver | Xảy ra khi di chuyển chuột qua kết nối hay anchor. |
| onSelect | Xảy ra khi người sử dụng lựa chọn một trường nhập dữ liệu trên form. |
| onSubmit | Xảy ra khi người dùng đưa ra một form. |
| onUnLoad | Xảy ra khi người dùng đóng một trang |

Sau đây là bảng các chương trình xử lý sự kiện có sẵn của một số đối tượng. Các đối tượng này sẽ được trình bày kỹ hơn trong phần sau.

|  |  |
| --- | --- |
| Đối tượng | Chương trình xử lý sự kiện có sẵn |
| Selection list | onBlur, onChange, onFocus |
| Text | onBlur, onChange, onFocus, onSelect |
| Textarea | onBlur, onChange, onFocus, onSelect |
| Button | onClick |
| Checkbox | onClick |
| Radio button | onClick |
| Hypertext link | onClick, onMouseOver, onMouseOut |
| Clickable Imagemap area | onMouseOver, onMouseOut |
| Reset button | onClick |
| Submit button | onClick |
| Document | onLoad, onUnload, onError |
| Window | onLoad, onUnload, onBlur, onFocus |
| Framesets | onBlur, onFocus |
| Form | onSubmit, onReset |
| Image | onLoad, onError, onAbort |

Ví dụ sau là một đoạn mã script đơn giản của chương trình xử lý sự kiện thẩm định giá trị đưa vào trong trường text. Tuổi của người sử dụng được nhập vào trong trường này và chương trình xử lý sự kiện sẽ thẩm định tính hợp lệ của dữ liệu đưa vào. Nếu không hợp lệ sẽ xuất hiện một thông báo yêu cầu nhập lại. Chương trình xử lý sự kiện được gọi mỗi khi trường AGE bị thay đổi và focus chuyển sang trường khác. Hình 5.10 minh hoạ kết quả của ví dụ này

<HTML>  
<HEAD>  
<TITLE> An Event Handler Exemple </TITLE>  
<SCRIPT LANGUAGE= "JavaScript">  
function CheckAge(form) {  
if ( (form.AGE.value<0)||(form.AGE.value>120) )  
 {  
alert("Tuổi nhập vào không hợp lệ! Mời bạn nhập lại");  
 form.AGE.value=0;  
 }  
}  
</SCRIPT>  
</HEAD>  
<BODY>  
<FORM NAME="PHIEU\_DIEU\_TRA">  
  
Nhập vào tên của bạn:<BR>  
Tên <INPUT TYPE=TEXT NAME="TEN" MAXLENGTH=10 SIZE=10><BR>  
Đệm <INPUT TYPE=TEXT NAME="DEM" MAXLENGTH=15 SIZE=10><BR>  
Họ <INPUT TYPE=TEXT NAME="HO" MAXLENGTH=10 SIZE=10><BR>  
Age <INPUT TYPE=TEXT NAME="AGE" MAXLENGTH=3 SIZE=2 onChange="CheckAge(PHIEU\_DIEU\_TRA)"><BR>  
<P>  
  
Bạn thích mùa nào nhất:<BR>  
Mùa xuân<INPUT TYPE=RADIO NAME="MUA" VALUE="Mua xuan">  
Mùa hạ<INPUT TYPE=RADIO NAME="MUA" VALUE="Mua ha">  
Mùa thu<INPUT TYPE=RADIO NAME="MUA" VALUE="Mua thu">  
Mùa đông<INPUT TYPE=RADIO NAME="MUA" VALUE="Mua dong">  
<P>  
  
Hãy chọn những hoạt động ngoài trời mà bạn yêu thích:<BR>  
Đi bộ<INPUT TYPE=CHECKBOX NAME="HOAT\_DONG" VALUE="Di bo">  
Trượt tuyết<INPUT TYPE=CHECKBOX NAME="HOAT\_DONG" VALUE="Truot tuyet">  
Thể thao dưới nước<INPUT TYPE=CHECKBOX NAME="HOAT\_DONG" VALUE="Duoi nuoc">  
Đạp xe<INPUT TYPE=CHECKBOX NAME="HOAT\_DONG" VALUE="Dap xe">  
<P>  
<INPUT TYPE=SUBMIT>  
<INPUT TYPE=RESET>  
</FORM>  
</BODY>  
</HTML>

*Minh hoạ cho ví dụ Event Handler*

# **Bài tập**

1. Viết một đoạn lệnh JavaScript sử dụng cách thức confirm() và câu lệnh if...then để thực hiện:

Hỏi người sử dụng có muốn nhận được một lời chào không

Nếu có thì hiện ảnh wellcome.jpg và một lời chào.

Nếu không thì viết ra một lời thông báo

Sử dụng cách thức confirm() và cấu trúc if...then:

<HTML>  
<HEAD>  
<TITLE>Execise 5.1</TITLE>  
<HEAD>  
<BODY>  
<P>  
<SCRIPT LANGUAGE="JavaScript">  
 var conf=confirm("Click OK to see a wellcome message!")  
 if (conf){  
 document.write("<IMG SRC='wellcome.jpg'>");  
 document.write("<BR>Wellcome you come to CSE's class");  
 }  
 else  
 document.write("What a pity! You have just click Cancel button");  
</SCRIPT>  
</P>  
</BODY>  
</HTML>

2. Viết một đoạn lệnh JavaScript để thực hiện:

Hỏi người sử dụng: "10+10 bằng bao nhiêu?"

Nếu đúng thì hỏi tiếp: Có muốn trả lời câu thứ hai không?"

Nếu muốn thì hỏi: "10\*10 bằng bao nhiêu?"

Đánh giá kết quả bằng biểu thức logic sau đó viết ra màn hình:Đúng: "CORRECT"; Sai: "INCORRECT"

Gợi ý: Sử dụng biến toàn cục lưu kết quả đúng để so sánh với kết qủa đưa vào.

Thực hiện hỏi người sử dụng:

Thực hiện hỏi người sử dụng:  
<HTML>  
<HEAD>  
<TITLE>Exercise 3.3</TITLE>  
<SCRIPT LANGUAGE="JavaScript">  
<!-- HIDE FROM OTHER BROWSERS  
// DEFINE VARIABLES FOR REST OF SCRIPT  
var question="What is 10+10?";  
var answer=20;  
var correct='CORRECT';  
var incorrect='INCORRECT';  
// ASK THE QUESTION  
var response = prompt(question,"0");  
// chECK THE ANSWER THE FIRST TIME  
if (response != answer) {  
// THE ANSWER WAS WRONG: OFFER A SECOND chAncE  
if (confirm("Wrong! Press OK for a second chance."))  
 response = prompt(question,"0");  
 } else {  
 // THE ANSWER WAS RIGHT: OFFER A SECOND QUESTION  
 if (confirm("Correct! Press OK for a second question.")) {  
 question = "What is 10\*10?";  
 answer = 100;  
 response = prompt (question,"0");  
 }  
 }  
 // chECK THE ANSWER  
 var output = (response == answer) ? correct : incorrect;  
 // STOP HIDING FROM OTHER BROWSERS -->  
 </SCRIPT>  
 </HEAD>  
 <BODY>  
 <SCRIPT LANGUAGE="JavaScript">  
 <!-- HIDE FROM OTHER BROWSERS  
 // OUTPUT RESULT  
 document.write(output);  
 // STOP HIDING FROM OTHER BROWSERS -->  
 </SCRIPT>  
 </BODY>  
 </HTML>

3. Câu lệnh nào trong các câu sau đây sử dụng sai thẻ sự kiện

a. <BODY onClick="doSomething();">

b. <INPUT TYPE=text onFocus="doSomething();">

c. <INPUT TYPE=textarea onLoad="doSomething();">

d. <BODY onUnload="doSomething();">

e. <FORM onLoad="doSomething();">

f. <FORM onSubmit="doSomething();">

Các câu sai: a, c, e. Các câu đúng: b, d, f

4. Điều gì xảy ra khi thực hiện script sau:

<HTML>  
 <HEAD>  
 <TITLE>Exercise 5.4</TITLE>  
 <SCRIPT LANGUAGE="JavaScript">  
 <!-- HIDE FROM OTHER BROWSERS  
 var name = "";  
 function welcome() {  
 name = prompt("Welcome to my page! What's Your Name?","name");  
 }  
 function farewell() {  
 alert("Goodbye " + name + ". Thanks for visiting my page.");  
 }  
 // STOP HIDING FROM OTHER BROWSERS -->  
 </SCRIPT>  
 </HEAD>  
 <BODY onLoad="welcome();" onUnload="farewell();";>   
 This is my page!  
</BODY>  
</HTML>

Khi chương trình được chạy (load), hàm wellcome sẽ thực hiện hỏi tên người sử dụng, lưu tên đó vào biến toàn cục name. Khi người sử dụng sang một địa chỉ URL khác, hàm farewell() sẽ thực hiện gửi một lời cảm ơn tới người sử dụng.

5. Sử dụng vòng lặp while để mô phỏng các vòng lặp for sau:

a. for (j = 4; j > 0; j --) {

document.writeln(j + "<BR>");

}

b. for (k = 1; k <= 99; k = k\*2) {

k = k/1.5;

}

c. for (num = 0; num <= 10; num ++) {

if (num == 8)

break;

}

5. Sử dụng vòng lặp while như sau:

a. j = 5;

while (--j > 0) {

document.writeln(j + "<BR>");

b. k = 1;

while (k <= 99) {

k = k \* 2 / 1.5;

}

c. num = 0;

while (num <= 10) {

if (num++ == 8)

break;

}

## **Thứ Năm, 18 tháng 7, 2013**

### Các lệnh cơ bản trong JavaScript [JavaScript]

Thông thường mã lệnh JavaScript bắt đầu bằng thẻ **<script type="text/javascript" >** và kết thúc bằng thẻ **</script>**

**Ví dụ:**

<script type="text/javascript" >  
 alert("Hello World");  
</script>

Đoạn mã JavaScript sẽ được đặt giữa hai thẻ *<head>* và *</head>*. Tuy nhiên, chúng ta vẫn có thể nhúng mã JavaScript bên trong thẻ *<body>*

**Ví dụ:**

<html>  
 <head></head>  
 <body>  
 <script>  
 alert("Hello World");  
 </script>  
 </body>  
</html>

Bạn cũng có thể dùng một thuộc tính **src** để bao gộp một tập tin chứa mã JavaScript ở bên ngoài vào trong trang web. Ví dụ bạn có một tập tin *Hello.js* chứa mã lệnh JavaScript như sau: *document.write(“Hello World!”)*

Khi đó bạn có thể bao gộp tập tin này vào trang web:

<script type="text/javascript" src=”hello.js”></script>

**Chú ý:** tập tin chứa mã lệnh bên ngoài chỉ là một tập tin văn bản bình thường với phần mở rộng là *.js*. Dĩ nhiên bạn có thể dùng bất cứ một trình soạn thảo văn bản nào để tạo những tập tin như vậy. Ngoài ra, mã JavaScript nằm trong tập tin này không được chứa thẻ < script type... > và </script>.

**1. Phương thức write() và writeln()**

Hàm *write()* dùng để xuất nội dung ra thiết bị xuất chuẩn (ví dụ như màn hình), hàm *writeln()* cũng có tác dụng tương tự ngoại trừ sau khi xuất xong thì con trỏ sẽ tự động chuyển xuống đầu dòng tiếp theo

**Ví dụ:**

<HTML>  
 <HEAD>  
 <TITLE> Welcome to my site </TITLE></HEAD>  
 <BODY>  
 <SCRIPT TYPE="TEXT/JAVASCRIPT" >  
 **document.write(“Welcome to my site!”);**  
 </SCRIPT>  
 </BODY>  
</HTML>

Đối tượng *Document* là một trong những đối tượng rất quan trọng của JavaScript. Nó ám chỉ đến tài liệu mà chúng ta đang thao tác trên tài liệu đó hay nói cách khác là nó chính là trang web chúng ta đang viết trên trang web đó

**Ví dụ:**

document.write(“Hi there.”)

Trong đoạn mã trên, *document* là một đối tượng, *write* là một phương thức của đối tượng này

**2. Thuộc tính lastModified**

Bạn có thể dùng phương thức này để hiển thị ngày cuối cùng tài liệu của bạn được cập nhật trong trang web bằng đoạn mã sau:

<script language=”JavaScript”>  
 document.write(“This page created by John N. Last update:” + document.lastModified);  
</script>

**3. Thuộc tính bgColor và fgColor**

Thuộc tính bgColor và fgColor dùng để xác định màu nền và màu chữ của tài liệu

**Ví dụ:**

<script>  
document.bgColor=”black” document.fgColor=”#336699"  
</script>

**Các hộp thông báo (message box)**

**1. Hộp alert (cảnh báo)**

Dùng để đưa ra lời cảnh báo cho người dùng đang xem trang web:

<script>  
window.alert(“Welcome to my site!”)  
</script>

**2. Hộp confirm (xác nhận)**

window.confirm(“Are you sure you want to quit?”)

**3. Hộp prompt (nhắc nhở)**

Hộp prompt cho phép người dùng nhập vào thông tin như được yêu cầu

window.prompt(“please enter user name”)

**Sử dụng biến và điều kiện**

Hãy xem xét ví dụ sau:

<script>  
 var x=window.confirm(“Are you sure you want to quit?”)  
 if (x)  
 window.alert(“Thank you!”)  
 else  
 window.alert(“Good choice!”)  
</script>

Có nhiều khái niệm chúng ta cần phải hiểu rõ trong đoạn mã trên. Trước tiên, *var x=* là một khai báo biến. Nếu bạn muốn tạo một biến, bạn phải khai báo biến đó dùng phát biểu *var*. Trong ví dụ trên, *x* sẽ nhận trị là *true* (đúng) hay *false* (sai). Sau đó chúng ta dùng phát biểu điều kiện *if else* để có thể lựa chọn một trong hai khả năng dựa trên kết quả của biến x. Nếu là *true* (nghĩa là người dùng click Ok) thì sẽ hiển thị hộp alert có nội dung là “Thank you”. Ngược lại, nếu kết quả là *false* (người dùng click Cancel) hộp alert “Good choice sẽ được hiển thị.

Một ví dụ phức tạp hơn:

<html>  
 <head>  
 <script>  
 var x=confirm(“Are you sure you want to quit?”)  
 if (!x)  
 window.location=”http:// www.yahoo.com”  
 </script>  
 </head>  
 <body>  
 Welcome to my website!.  
 </body></html>

**Hàm**

Hàm là một nhóm mã lệnh. Trước tiên hãy thử tạo một hàm đơn giản:

function test()  
 {  
 document.write(“Hello can you see me?”)  
 }

Tuy nhiên nếu chỉ đặt đoạn mã trên vào giữa hai thẻ <script> và </script> thì bạn sẽ... không thấy gì trên màn hình cả bởi vì hàm bạn viết chưa được gọi. Bạn thêm lời gọi hàm test() vào đoạn mã trên, như sau:

function test()  
 {  
 document.write(“Hello can you see me?”)  
 }  
test()

Nguồn: Sưu tầm Internet