

*Giáo trình
Javascript*

Mục Lục

Giới Thiệu Về Hocvui.net	3
MỞ ĐẦU Javascript	3
Mở đầu về Javascript	3
Cú pháp Javascript.....	5
Các kiểu giá trị	6
Tiếp theo.....	7
Phép toán, câu lệnh gán.....	10
javascript Cách đặt tên biến	11
javascript Function - hàm	12
thủ tục vào, ra đơn giản.....	12
javascript - event.....	13
Các hàm sẽ dùng trong các ví dụ	14
Một số hàm hay dùng trong JavaScript	14
JavaScript Comments - chú thích	15
CẤU TRÚC RỄ NHÁNH	16
cấu trúc rẽ nhánh - câu lệnh if	16
else if	17
CẤU TRÚC LẶP	19
Cấu trúc lặp.....	19
break, dừng vòng lặp.....	20
MẢNG	21
Mảng trong javascript	21
Tại sao lại dùng đến mảng	22
Array function - các hàm làm việc với mảng	24
Mảng hai chiều javascript.....	26
Mảng dựng sẵn.....	27
JS FUNCTION - HÀM.....	29
Tự soạn thảo các hàm javascript.....	29
Hàm và các tham số.....	30
Biến toàn cục và biến cục bộ	32
javascript - function return.....	34
FORM	37
Giải đáp trước thắc mắc về những dấu chấm.....	37
javascript làm việc với các form.....	37
Nhận + thiết lập thông tin từ các biểu mẫu.....	41
Kiểm tra các trường text,password,textarea.....	43
kiểm tra trường select.....	44

Kiểm tra các nút radio	45
Xử lý chuỗi.	46
WINDOW.....	48
Làm việc với các cửa sổ.....	48
Đóng cửa sổ đã mở.....	49
Di chuyển cửa sổ.....	50
Địa chỉ cửa sổ.....	51
FRAMES.....	52
Javascript và frame.....	52
Ví dụ về frame và js.....	53
HẸN GIỜ.....	66
Định giờ cho các sự kiện.....	66
clearTimeout - Ngưng hẹn giờ.....	67
Tạo một đồng hồ bấm giờ.....	68
LINH TINH KHÁC.....	69
Làm việc với ngày giờ.....	69
JavaScript getElementById.....	70
JavaScript innerHTML.....	71
JavaScript Void 0.....	71
Vấn đề trình duyệt Navigator.....	72
COOKIES.....	73
Cookies dùng để làm gì.....	73
Thiết lập cookies.....	75
Đọc cookies.....	76

Giới Thiệu Về Hocvui.net

Rõ ràng, Internet đang thay đổi thế giới, và dĩ nhiên thay đổi cả nhiều người trong chúng ta. Chúng ta có thêm nhiều cách để giao dịch, nhiều cách để trò chuyện, nhiều thứ để vui chơi, **và có lẽ trên hết là nhiều kiến thức hơn...**

Các tiện ích từ những Website từ một thứ thú vị đã trở thành tất yếu, tìm kiếm thông tin với Google, một góc riêng của mình với Yahoo Blog, xem video trên Youtube có lẽ đã trở thành thói quen của nhiều người trong chúng ta.

Một điều chắc chắn là ứng dụng Web sẽ càng ngày đi sâu và phát triển hơn nữa vào cuộc sống của con người. Vì thế hocvui.net được tạo ra để đem lại cho các bạn những hiểu biết nền tảng về mảng kiến thức này.

Lượng thông tin qua mạng Internet giờ đây nhiều hông kể xiết, hocvui.net chẳng qua là một Website được xây dựng bởi những học sinh, chắc chắn sẽ không thể nào đem tới cho bạn những kiến thức chuyên sâu, cũng như không hề dám khẳng định rằng hocvui.net không có thiếu sót. Mọi điều hocvui.net muốn đem lại chia sẻ cho các bạn những gì chúng tôi biết và ngược lại từ phía bạn. Chúng ta hãy cùng học hỏi và cùng chia sẻ cho nhau những kiến thức đó!

Các bài viết hướng dẫn của hocvui.net đều được viết ra trong quá trình tự học, sau từ những kinh nghiệm thực tế, mình đã trình bày lại, sưu tầm thêm, cũng như thêm vào những lưu ý của chính bản thân mình.

Bài viết của mình đã được mình sắp xếp theo thứ tự từ đơn giản tới phức tạp, hệ thống kiến thức lại một cách cô đọng nhất có thể. Kể cả thứ tự các ngôn ngữ mình cũng đã sắp xếp theo thứ tự từ cơ bản nhất tới nâng cao dần, mong các bạn hãy xem qua các bài viết của mình với thứ tự đó. **HTML>>JAVASCRIPT/VBSCRIPT>>PHP/ASP>>SQL**, mỗi bài đều xem từ trên xuống dưới, đừng nên bỏ qua thứ gì, đó là kinh nghiệm mà mình đúc kết được.

Và điều quan trọng nhất:

Mình luôn mong muốn các bạn có thể giúp mình hoàn thiện nội dung kiến thức bằng cách viết các comment cho bài viết, thẳng thắn đưa ra ý kiến, nêu ra những khuyết điểm để chúng ta cùng sửa chữa.

Chân thành cảm ơn các bạn!

MỞ ĐẦU Javascript

Mở đầu về Javascript

Tháng Sáu 02, 2009, 03:11:00 PM gửi bởi [nvcnvn](#)

Javascript là ngôn ngữ thứ hai mình muốn giới thiệu với các bạn vừa mới bước chân vào con đường webmaster, rất nhiều các cao thủ thiết kế web sử dụng ngôn ngữ này và khiến trang web của họ trở nên sống động, có thể nói javascript đã trở thành một ngôn ngữ không thể không nhắc tới khi nói tới lĩnh vực thiết kế website.

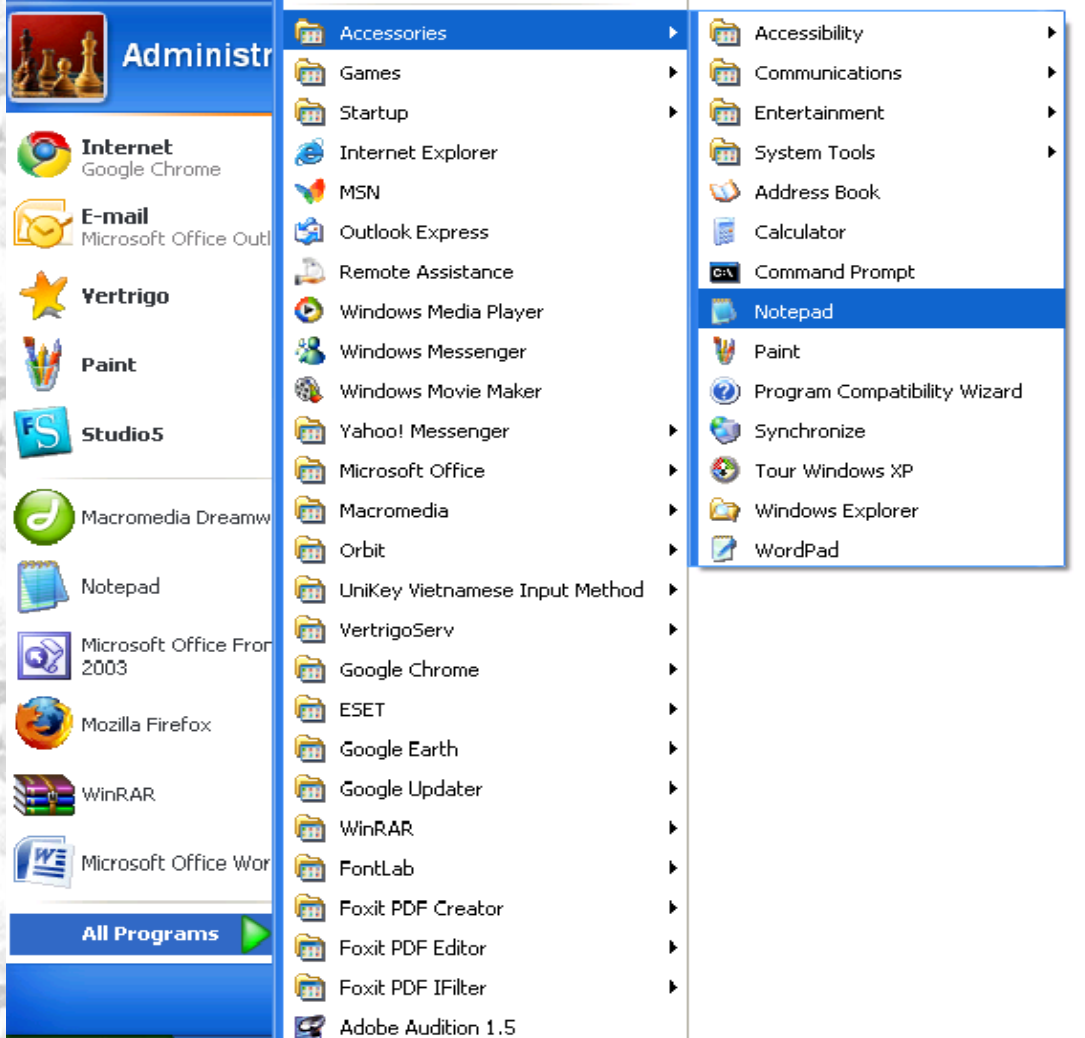
Sau đây là một số thông tin mà mình sưu tầm để chúng ta còn đi ba hoa

JavaScript, theo phiên bản hiện hành, là một ngôn ngữ lập trình kịch bản dựa trên đối tượng được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ này được dùng rộng rãi cho các trang web, nhưng cũng được dùng để tạo khả năng viết script sử dụng các đối tượng nằm sẵn trong các ứng dụng. Nó vốn được phát triển bởi

Brendan Eich tại Hãng truyền thông Netscape với cái tên đầu tiên Mocha, rồi sau đó đổi tên thành LiveScript, và cuối cùng thành JavaScript. Giống Java, JavaScript có cú pháp tương tự C, nhưng nó gần với Self hơn Java. **.js** (sau này mình gọi tắt javascript là js luôn cho nó ngắn) là phần mở rộng thường được dùng cho tập tin mã nguồn JavaScript.

Phiên bản mới nhất của JavaScript là phiên bản 1.5, tương ứng với ECMA-262 bản 3. ECMAScript là phiên bản chuẩn hóa của JavaScript. Trình duyệt Mozilla phiên bản 1.8 beta 1 có hỗ trợ không đầy đủ cho E4X - phần mở rộng cho JavaScript hỗ trợ làm việc với XML, được chuẩn hóa trong ECMA-357.

Điều kiện cần để học **javascript**: Kiến thức cơ bản của HTML và người bạn Notepad



Và để có thể đi đâu cũng có thể noi rằng: Tui biết chút ít Javascript, ta hãy viết đoạn javascript đầu tiên của mình, bằng cách mở bất kì một file html nào của mình ra, chèn đoạn code nà vào bất cứ đâu:

Code:

```
<script language="javascript">
alert('javascript');
</script>
```

Ví dụ:

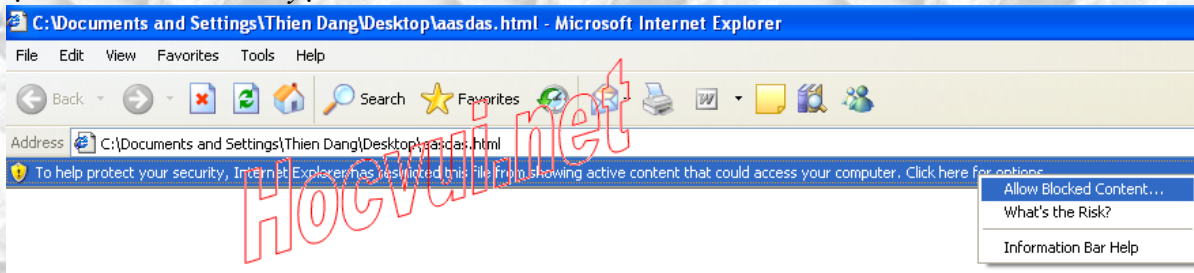
Code:

```
<html>
<head>
<title>Test Script</title>
</head>
<body>
<script language="javascript">
alert('javascript');
</script>
```

</body>

</html>

Chú ý: Nếu bạn dùng trình duyệt IE, rất có thể bạn sẽ không chạy được script, bạn sẽ thấy một thông điệp tại lề trên của trình duyệt:



Click vào cái thanh báo chọn *Allow blocked content*..

Thì lúc đó javascript mới có thể chạy!...

Cú pháp Javascript

Tháng Sáu 05, 2009, 07:03:39 AM gửi bởi [nvcnvn](#)

Javascript thường được hèn vào một trang web, nếu không muốn nói là sinh ra chỉ để chèn vào trang web. Ta có thể chèn Javascript vào html rất đơn giản với cái thẻ **script**, sau đó chèn code javascript vào bên trong nội dung thẻ script.

Code:

```
<script language="javascript">
document.write("Hello World!");
</script>
```

kết quả:

Hello World!

điều đặc biệt là js có thể xuất ra html.

Đây dĩ nhiên là điều không thể thiếu để lập trình web với một ngôn ngữ lập trình 🎉!

Code:

```
<script language="javascript">
document.write("<b>Hello, what's your name!?!</b>");
document.write("<b>My name:</b> <input type='text' />");
</script>
```

Bạn hãy tự thử xem nhé!

Câu hỏi đặt ra là: tại sao lại không viết mọi thứ một cách bình thường! hay bạn thắc mắc về bất cứ điều gì!

Câu trả lời sẽ nằm ở các phần sau!

Bây giờ chú ý tới mặt cú pháp!

bạn để ý **trước**, mỗi lần mình viết **document.write()** thì đó là một câu lệnh, mỗi câu lệnh:

Code:

Câu lệnh là đơn vị cơ bản của một ngôn ngữ lập trình. Trong trường hợp đặc biệt, nó có thể cũng trở thành một đơn vị thao tác của máy tính điện tử hay còn gọi là một chỉ thị.

Vì mức độ phức tạp, việc dùng các chỉ thị để trực tiếp điều khiển máy tính sẽ rất ít thông dụng. Thay vào đó,

người ta ghép một số tổ hợp của các chỉ thị để cho máy thi hành được một động tác lớn hơn gọi là câu lệnh. Như vậy mỗi câu lệnh bao gồm một hay một số mệnh lệnh máy tính được sắp xếp theo trình tự xác định và nhằm mục đích ra lệnh cho CPU tiến hành một thao tác cố định có ý nghĩa.

Tùy theo ngôn ngữ lập trình, các câu lệnh sẽ có cấu trúc khác nhau và có trật tự sắp xếp nhất định. Trật tự này thường không đổi và được gọi là cú pháp (syntax).

Câu lệnh có thể hiểu như là mệnh đề cơ bản có thể được cấu trúc thông qua việc sử dụng các từ khóa (đã được định nghĩa từ trước bởi ngôn ngữ lập trình) hoặc là có thể tạo bởi các chỉ thị từ các cấu trúc ngữ pháp hay cú pháp đã được định nghĩa sẵn. Các câu lệnh của một chương trình dùng để chỉ thị cho máy tính biết làm gì, xử lý như thế nào với các dữ liệu và từ đó tiến hành các phép tính toán hay biến đổi dữ liệu để đạt được kết quả.

Và kết thúc mỗi câu lệnh, ta phải kết thúc với dấu ;

Chèn javascript vào HTML

Các ví dụ ở trên đã nêu ra cách chèn javascript vào HTML bình thường với cặp thẻ <script>

Nếu bạn có một đoạn javascript, và muốn dùng nó cho nhiều trang web, bạn không cần phải viết lại. Save nội dung javascript (không có hai thẻ <script> nhé) lại với một file có phần mở rộng là **.js** (giống như với html là .html thôi)

Đặt thêm thuộc tính src vào thẻ script mở:

VD:

Code:

```
<script language="javascript" src="http://hocvui.net/javascript/ads.js"></script>
```

Bên trong thẻ script không cần chứa bất cứ gì nhưng nội dung của file chứa javascript đã được thêm vào trang html.

Sau này khi các bạn có đã có thể viết những hàm cho riêng mình, đặt biệt là với những hàm dài, công dụng trên nhiều site, cách làm này rất hữu ích.

Các kiểu giá trị

Tháng Sáu 09, 2009, 11:24:53 AM gửi bởi [nvcnvn](#)

Javascript thuộc loại ngôn ngữ mà kiểu của biến được đoán, một số ngôn ngữ cần được khai báo trước kiểu biến như pascal chẳng hạn! Một số kiểu giá trị:

Boolean

Boolean là giá trị logic có hai giá trị là **TRUE** hoặc **FALSE**. (TRUE có nghĩa là đúng, FALSE là sai)

Số

là loại giá trị dùng để tính toán đó, đừng nói bạn không biết số là gì nha!

Số thì có số động, số nguyên.....

Chuỗi

Chuỗi là các kí tự bằng chữ cái + kí tự đặc biệt + số, nói chung là bất kì cái gì mà ta viết ra trên bàn phím được, thì là chuỗi!

Phân biệt kiểu chuỗi và số

ta gán biến a và b với các giá trị như sau:

```
a=1;
```

```
b="1";
```

a sẽ mang giá trị số, còn b mang giá trị chuỗi ==> đặt bên trong dấu nháy là chuỗi!

Tiếp theo

Tháng Sáu 08, 2009, 03:37:51 PM gửi bởi [nvcnvn](#)

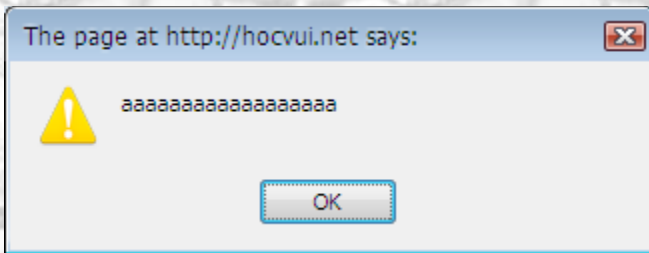
các bạn đã làm quen với hai khái niệm đầu tiên là **hàm** và **biến**, ở đây nếu bạn đã từng học qua bất kì một ngôn ngữ lập trình nào thì mọi việc sẽ được đơn giản hoá, nhưng còn với những người không chuyên như tui mình, mọi việc sẽ hơi rắc rối phải dành riêng một bài để nói cho rõ!

Ví dụ mình có đoạn code

1

Code:

```
<script language="javascript">
alert("aaaaaaaaaaaaaaaa");
</script>
```

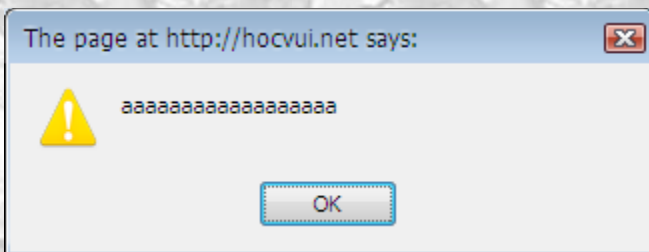


Với đoạn code tương tự nhưng:

2

Code:

```
<script language="javascript">
thongbao="aaaaaaaaaaaaaaaa";
alert(thongbao);
</script>
```

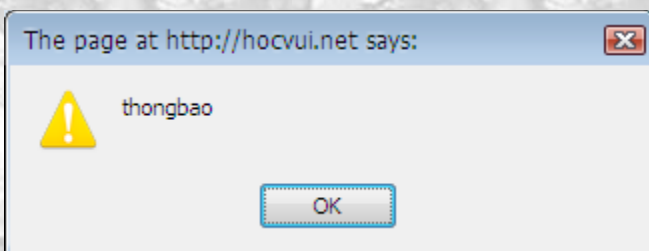


và đoạn code thứ hai:

3

Code:

```
<script language="javascript">
thongbao="aaaaaaaaaaaaaaaa";
alert("thongbao");
</script>
```

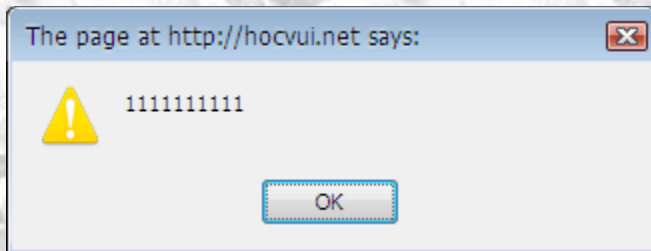


Cũng một loạt ví dụ tương tự nhưng với nội dung là một con số:

4

Code:

```
<script language="javascript">  
alert("111111111");  
</script>
```

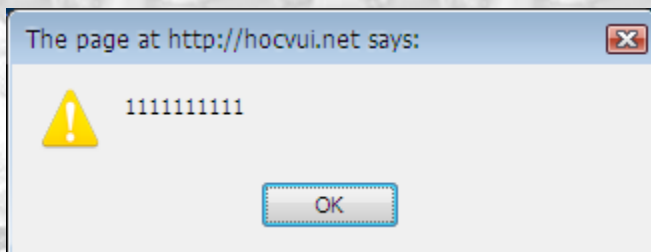


và:

5

Code:

```
<script language="javascript">  
alert(111111111);  
</script>
```

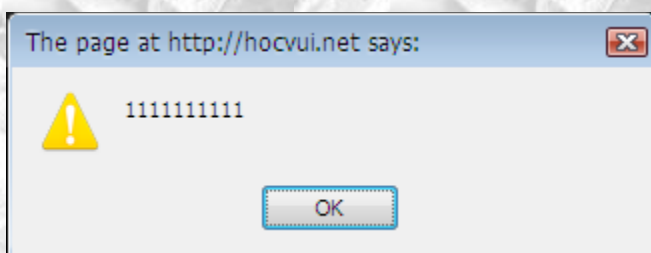


Bây giờ ta gán cho một biến giá trị 11111111111

6

Code:

```
<script language="javascript">  
thongbao=1111111111;  
alert(thongbao);  
</script>
```

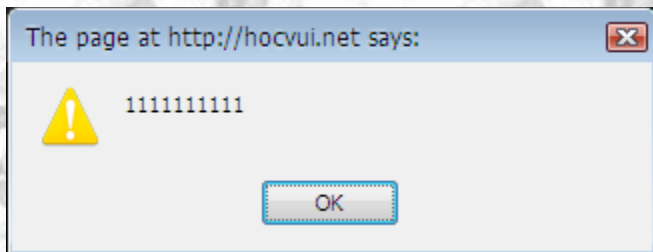


và:

7

Code:

```
<script language="javascript">  
thongbao="1111111111";  
alert(thongbao);  
</script>
```



Chắc các bạn cũng nhận ra, sự đặc biệt mình muốn nhấn mạnh chính là các dấu ("), từ trong phép gán giá trị tới nội dung hàm.

Ở ví dụ 2,3 chúng ta nhận ra ngay sự khác biệt! giữ hai kết quả *aaaaaaaa* và *thongbao*, thật sự là như thế này:

thongbao="aaaaaaaaaaaaa"; dòng này có nghĩa là gán giá trị là **chuỗi aaaaaaaaa** cho biến *thongbao*.
alert(thongbao); lệnh `alert` sẽ lấy giá trị mà biến *thongbao* mang và in ra, trong lúc này *thongbao* có giá trị là *aaaaaaaa*. Nhưng với dòng lệnh:

alert("thongbao"); dòng này có nghĩa là in ra **chuỗi thongbao**.

Như vậy, ta thấy được rằng giá trị chuỗi sẽ được đặt trong cặp dấu "

Còn với các ví dụ 4,5,6,7 các bạn biết tại sao kết quả lúc nào cũng là những con số một tuy rõ ràng là có sự khác biệt giữa các dùng các dấu "

Thật ra là vì :

alert("111111111"); các con số một ở đây hiểu là chuỗi dạng số.

alert(111111111); còn đây là con số 111111111.

Kí tự chữ cái lúc nào cũng là chuỗi, trừ phi là tên riêng của cái gì đó, số mà đặt bên trong dấu ngoặc kép kép thì trở thành chuỗi.

Đề dùng giá trị của biến thì biến không được đặt trong dấu ngoặc kép giống như ở ví dụ 3, ý của người lập trình là hàm này sẽ in ra giá trị của biến *thongbao*, nhưng lại nhầm khiến giá trị in ra là chuỗi *thongbao*.

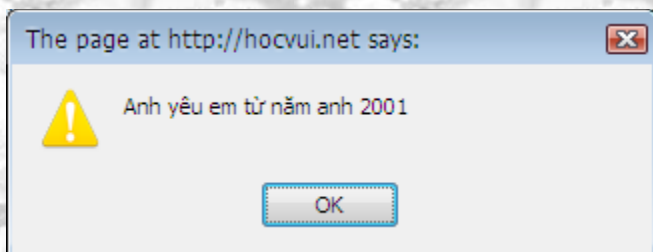
Đây cũng là lí do mà tên biến bắt buộc phải bắt đầu với kí tự chữ cái, thử nghĩ xem, một anh chàng quyết định đặt tên biến là **123** và giá trị của nó là **456** rắc rối sẽ nảy sinh ra ở đây!

Còn đây là ví dụ cuối cùng của bài này:

8

Code:

```
<script language="javascript">
a=2001;
alert("Anh yêu em từ năm anh "+a);
</script>
```

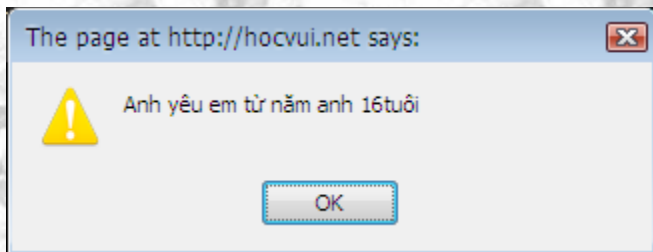


Hoặc là với ví dụ này

9

Code:

```
<script language="javascript">
b="Anh yêu em từ năm anh ";
a=15;
alert(b+(a+1)+"tuổi");
</script>
```



Làm ơn đừng chú ý tới cái dấu cộng, chỉ hãy hiểu là dấu + dùng để nối hai thứ đó lại.

Ví dụ này vừa thể hiện rõ mối quan hệ giữa biến, chuỗi và số, tất nhiên là tại sao cần dùng tới biến. ví như ví dụ 8, bạn nói bạn yêu một cô từ năm 2001, lỡ vài năm sau chia tay, bạn muốn dùng lại trang web này tặng cho một cô khác, lúc này bạn thấy mọi việc đơn giản là đổi năm quen biết lại, nhưng hãy đặt trường hợp cả chương trình không chỉ có một dòng có năm **2001** thì để sửa lại mọi chuyện sẽ đơn giản hơn nhiều nếu ta dùng tới biến.

Còn ở ví dụ 9, các cặp dấu () cũng thể hiện thứ tự ưu tiên cho các bước thực hiện như trong môn toán, mình chỉ viết ví dụ này cho các bạn hiểu thêm.

Phép toán, câu lệnh gán.

Tháng Sáu 11, 2009, 11:19:25 AM gửi bởi [nvcnvn](#)

Câu lệnh gán:

Lệnh gán là 1 trong những lệnh cơ bản nhất của ngôn ngữ lập trình, trong javascript có dạng:

<tên biến>=<biểu thức>;

nếu gán biến cho giá trị chuỗi:

<tên biến>="<biểu thức>";

cần có thêm cặp dấu nháy.

Các bạn sẽ tìm hiểu sâu hơn cách đặt tên biến ở bài cách đặt tên biến

1.Toán tử số học (+[cộng], -[trừ], *[nhân], /[chia], %[chia lấy dư], ++[tự tăng 1],--[tự giảm 1])

+ : Phép cộng.(khỏi giả thích)

- : Phép trừ.(khỏi giả thích)

* : Phép nhân.(khỏi giả thích)

/: Phép chia.(khỏi giả thích)

% : Phép chia lấy phần dư.VD: 5/3 được 1 dư 2 thì **5%3=2**,

++: Phép tăng một đơn vị.VD: 1++=2; 5++=6

--: Phép giảm một đơn vị.VD:1--=0; 9--=8

2.Toán tử so sánh

== : So sánh bằng.

> : So sánh lớn hơn.

< : Nhỏ hơn.

>= : So sánh lớn hơn hoặc bằng.

<= : Nhỏ hơn hoặc bằng.

!= : So sánh khác

Chú ý: chỉ có <= (bé hơn hoặc bằng) chứ không có =<(bằng hoặc bé hơn, cũng như thế đối với >=, không có chuyện =>)

3.Toán tử logic

&& (and): Giá trị đúng khi cả hai cùng đúng. VD: nếu (anh yêu em) và (em yêu anh) thì cưới ngay 😊. ta chỉ cưới ngay khi hai điều kiện trong ngoặc cùng đúng.

|| (or) : Giá trị đúng nếu một trong 2 đều đúng. VD: nếu (em đau khổ) hoặc (anh đau khổ) thì chia tay



Chỉ cần một trong hai điều kiện đúng là chia tay

Đối với chuỗi

Toán tử + tượng trưng cho việc nối hai chuỗi lại.

VD:

a="I am";

b="Nguyen";

c=a+b;

thì c có giá trị là chuỗi "I am Nguyen"

javascript Cách đặt tên biến

Tháng Sáu 05, 2009, 11:29:25 AM gửi bởi [nvcnvn](#)

trước tiên, dành cho các bạn mới bước chân vào lĩnh vực lập trình:

Biến là đại lượng được đặt tên, dùng để lưu giữ giá trị và giá trị có thể thay đổi trong quá trình thực hiện chương trình.

SGK 11/12

Quả thật khái niệm được coi là cơ bản này khá là mơ hồ cho những người mới bắt đầu như chúng ta, nhưng các bạn sẽ từ từ hiểu ở mọi vấn đề khi tiếp tục đi sâu vào!

Một ví dụ đời thường:

Bạn có một bao thuốc lá, bạn lấy cây viết đặt tên nó là a, bạn nhét 10,000 VND vào bao thuốc lá, thì lúc này bao thuốc lá có giá trị là 10,000 VND (tất nhiên là không tính tiền cái bao 😊) rồi một hồi sau, bạn lấy 10,000 ra, bỏ vào tờ 5,000 thì nó có giá trị 5,000....

hay là x,y,z trong một bài toán, ta cho x=5, y=6, z=x+y thì Z=11

Còn trong lập trình javascript

Biến lại chia làm 2 loại, biến toàn cục và biến cục bộ, đây là hai khái niệm mà nói thiệt, hơi khó là phân biệt ngay lúc này, các chiều khó từ từ rồi ta sẽ quay lại, bây giờ chúng ta chỉ nhận biết sơ qua!

Biến toàn cục: ta khai báo biến toàn cục rất đơn giản, ví dụ ta đặt tên biến là a và giá trị =1.

Code:

a=1;

Biến toàn cục có giá trị trong toàn bộ văn bản

biến cục bộ: ta cần thêm từ khóa var vào trước.

Code:

var a=1;

biến cục bộ tất nhiên khác với biến toàn cục là nó chỉ có hiệu lực trong cục bộ 🤖, tất nhiên cục bộ ấy là cái gì thì ta sẽ tìm hiểu sau ha!

Quy tắc đặt tên biến

Có 3 quy tắc:

- Ký tự bắt đầu phải là một chữ cái, còn các ký tự tiếp theo có thể là chữ số, gạch dưới, chữ cái. Ngoài mấy thứ kể trên ra, bạn không nên thêm bất cứ thứ gì khác vào tên biến!
- Không được có khoảng trắng
- Các biến phân biệt chữ hoa chữ thường, vì thế **ANH** sẽ khác với **Anh** cũng như **anh**, để tránh rắc rối, ta nên dùng chữ thường cho tất cả tên biến và dùng dấu `_` để phân cách thay cho khoảng trắng.

javascript Function - hàm

Thứ Sáu 08, 2009, 11:30:21 AM gửi bởi [nvcnvn](#)

Hàm.à..ò, hàm không có gì là xa lạ với các bạn, trong các ví dụ của mình ở bài đầu tiên cũng có dùng một hàm, đó là hàm **alert**. Nói đơn giản :

Hàm là thứ mà khi ta gọi (gọi có nghĩa là viết ra và cho dòng lệnh chạy) thì nó sẽ thực hiện một số thao tác nhất định.

Hàm thường có dạng: `functionname()`

bên trong dấu ngoặc () đôi khi chả có gì, đôi khi là một con số hoặc trong trường hợp hàm **alert** thì là một chuỗi, từ từ ta sẽ giải thích thêm!

Ví dụ khi ta dùng hàm **alert("thông báo cái gì đó")** thì công việc của nó sẽ là: mở ra cái khung, cái khung có cái viền, cái viền màu ..., nền màu..., trong nền có dòng chữ thông báo cái gì đó.

Hàm **alert** là một trong những hàm dựng sẵn trong javascript, có rất nhiều các hàm như thế, các bài sau sẽ nói rõ hơn về từng hàm. bạn cũng có thể viết một hàm co riêng mình.

nói thêm về cách tự viết hàm:

Mình sẽ không nói về các viết mà sẽ nói cho bạn một ví dụ:

Bạn làm một trang web cho người yêu, bạn muốn ít nhất 10 lần sự kiện A suất hiện, Sự kiện A bao gồm:

Viết ra dòng chữ *anh yêu em*

Chuyển dòng chữ thành màu đỏ

chuyển dòng chữ thành màu xanh

viết ra dòng chữ *em yêu anh không thì tuy*



Ví dụ này dễ hiểu hơn ha!

thủ tục vào, ra đơn giản

Thứ Sáu 09, 2009, 04:14:41 PM gửi bởi [nvcnvn](#)

Trước tiên ta cũng phải biết một, hai cách để nhận dữ liệu và xuất dữ liệu ra màn hình để còn biết đường làm ví dụ!

hai thủ tục mà mình dùng để nhập và xuất dữ liệu là hàm **Prompt** và hàm **alert**.

Ví dụ như ta làm một cái máy tính, tính bình thương của một số nào đó, thì tất nhiên phải có một chỗ nào đó cho người dùng điền số vào.

hàm **Prompt** yêu cầu người dùng nhập vào một số thông tin nào đó và sau khi người dùng nhập và nhấn OK, hàm sẽ gán những gì người dùng nhập vào cho một biến.

VD:

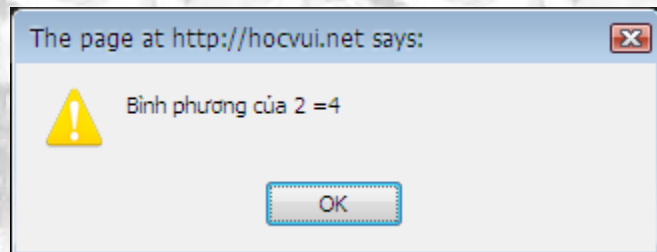
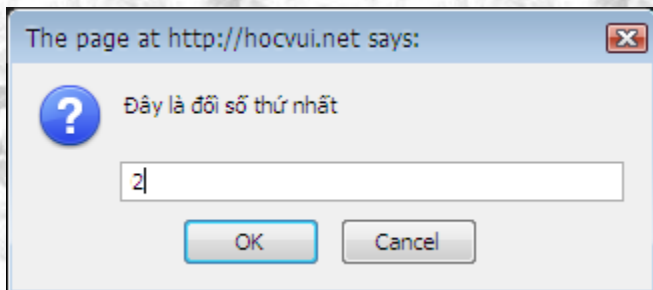
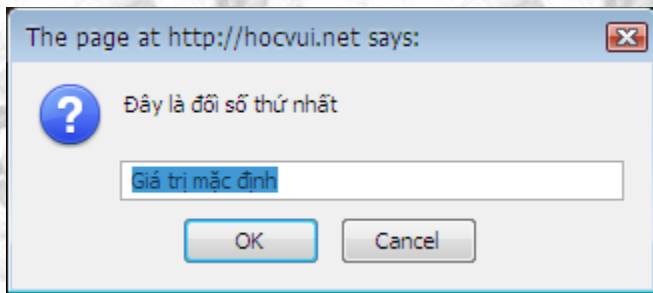
Code:

```
a=prompt("Đây là đối số thứ nhất","Giá trị mặc định");
```

```
b=a*a;
```

```
alert("Bình phương của "+a+"="+b);
```

[Xem ví dụ](#)



Ta phân tích `a=prompt("Số thứ nhất","");` ở đây ta thấy a được gán giá trị là hàm prompt, trong hàm prompt yêu cầu hai đối số (đối số chằng qua là cái gì hàm yêu cầu, hai đối số phân cách nhau bởi dấu phẩy)

đối số thứ nhất là thông báo sẽ hiển thị trong ô prompt, thường là câu hỏi, hay yêu cầu nhập giá trị, đối số thứ hai là giá trị mặc định bên trong trường thông tin, nếu không muốn có bất kỳ giá trị nào bên trong đây, ta đặt ngay sao dấu phẩy cặp dấu nháy.

Hàm **alert** thì quá quen thuộc chắc không cần nói nữa!

javascript - event

Tháng Sáu 08, 2009, 05:09:39 PM gửi bởi [nvcnvn](#)

Bạn còn nhớ bài [Event của HTML](#) không, chính là nó đó!

Các bạn nên xem qua trước khi đi vào. Khó mà hiểu được bài này nếu các bạn không xem qua nó!

Event dịch ra nghĩa Tiếng Việt chắc là Sự kiện.

ví dụ như sự kiện **onclick** có nghĩa là khi click chuột. Khi click chuột sẽ có một cái gì đó hiện ra.

Trong javascript, giả sử bạn có một đoạn code như thế này (đã chèn vào file html nhé):

Code:

```
alert("lần thứ nhất");
```

```
alert("lần thứ hai");
```

```
alert("lần thứ ba");
```

Viết ra như thế thì sẽ lần lượt hiện ra ba bảng không báo, nhưng nếu bạn muốn tùy theo hành động của người dùng, script sẽ chạy thì phải dùng tới event, dĩ nhiên tính tương tác của website phụ thuộc vào vấn đề này!

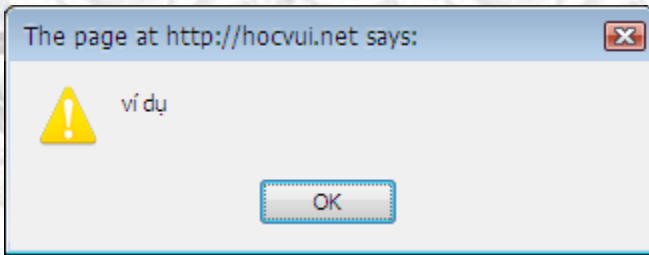
một điều chú ý:

bạn còn nhớ trong các quy tắc XHTML, ta gán giá trị của các sự kiện phải đặt trong cặp dấu "", nếu đoạn script cũng có chứa dấu nháy kép, mọi thứ sẽ bị nhầm lẫn, ta cần đổi dấu nháy kép của script thành dấu nháy đơn.

VD:

Code:

```
<a href="#" onclick="alert('ví dụ');">Click</a>
```



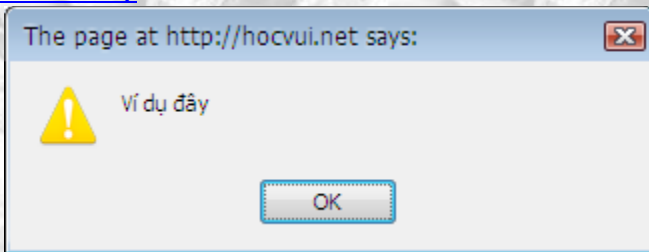
Vấn đề này nói tới đây thì đã rõ, các bạn chỉ việc xem kỹ lại bài HTML Event.

Các hàm sẽ dùng trong các ví dụ

Tháng Sáu 17, 2009, 04:10:09 PM gửi bởi [nvcnvn](#)

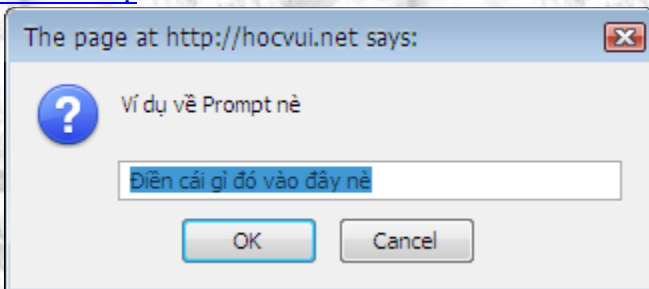
- `alert()`: Mở một hộp thông báo với nội dung đặt bên trong dấu ngoặc kép

[Xem ví dụ](#)



- `prompt()`: Mở một hộp thoại cho phép người sử dụng hông tin vào

[Xem ví dụ](#)



- `document.write()`: in ra màn hình với nội dung đặt bên trong dấu ngoặc kép

...

Một số hàm hay dùng trong JavaScript

Tháng Hai 12, 2009, 12:07:38 PM gửi bởi [nvcnvn](#)

* `parseInt("chuỗi")`: biến chuỗi thành số nguyên

* `parseFloat("chuỗi")`: biến chuỗi thành số thực

* `Number(<đối tượng>)`: chuyển đổi một "đối tượng" sang dạng số. Nếu thất bại trả về NaN (not a number)

* `isNaN(<giá trị>)`: kiểm tra giá trị. Nếu không phải số trả về true, ngược lại nếu là số trả về false

- * eval("chuỗi"): biến chuỗi thành biểu thức tính toán được, hoặc biến chuỗi thành lệnh thi hành được như là mã lệnh của JS.
- * Math.PI : hằng số $PI = 3.14$
- * Math.sqrt(a) : căn bậc 2 của a.
- * Math.pow(x,y) : tính x^y
- * Math.random() : tạo số ngẫu nhiên >0 và <1

JavaScript Comments - chú thích

Tháng Sáu 19, 2009, 02:11:02 PM gửi bởi [handsome](#)

Với những đoạn mã dài, chắc hẳn sẽ rất khó nhớ ngay cả ý nghĩ của bạn sao nhiều thán. Cái mà bạn cần lúc viết code là một vài dòng chú thích.

Chú thích sẽ không có ý nghĩa lập trình, mỗi lần bắt gặp chú thích javascript sẽ bỏ qua nó.

Chú thích bắt đầu với // Ở đầu một hàng bạn hãy viết hai dấu này, và tất cả những gì nằm cùng hàng với hai dấu này sẽ được xem là chú thích.

VD:

Code:

```
<script language="javascript">
//Dòng chú thích a
alert("Code javascript");
//dòng chú thích b
alert("Code javascript");
</script>
```

Nhiều chú thích cùng lúc

Khi mà bạn có một chú thích quá dài và việc viết trên cùng một hàng trở nên quá khó khăn, hãy dùng hai kí tự /* để bắt đầu những dòng chú thích, và đánh dấu kết thúc bằng hai kí tự */

VD:

Code:

```
/*
Những dòng chú thích
không nhất thiết rằng
phải nằm cùng hàng
*/
alert("Code javascript");
```


CẤU TRÚC RỄ NHÁNH

cấu trúc rẽ nhánh - câu lệnh if

Tháng Sáu 11, 2009, 04:29:50 PM gửi bởi [nvcnvn](#)

Đọc bài này các bạn cần xem lại: [các kiểu giá trị](#), [Phép toán](#), [câu lệnh gán](#).

Trong cuộc sống, có nhiều việc ta chỉ làm khi thoả mãn một số điều kiện cụ thể.

Ví dụ:

Một ông bố hứa với con trai:

_ Nếu con **đỗ đại học** thì bố sẽ **thưởng cho con một chiếc xe**.

Chừng nào mà điều kiện **con đỗ đại học** thì ông bố mới lo tới chuyện **thưởng cho con một chiếc xe**. Tới một ngày kia, ông bố lại nói với cậu con trai.

_ Nếu con **đỗ đại học** thì bố sẽ **thưởng cho con một chiếc xe**, nếu không thì **đi nghĩa vụ quân sự**.

Ở câu nói thứ nhất của ông bố, không nói rõ thì rõ thì có chuyện gì, còn ở câu thứ hai có nói rõ.

Trong lập trình tuy hình thức từng ngôn ngữ hơi khác nhau nhưng cũng có hai dạng cơ bản của câu điều kiện, ta chỉ bàn tới câu lệnh này trong phạm vi javascript

if-then

if(<điều kiện>)

```
{  
  câu lệnh 1;  
  câu lệnh 2;  
  .....  
}
```

Nếu chỉ có một câu lệnh sau về điều kiện thì ta không cần cặp dấu ngoặc {}, nhưng luôn có có hiện diện của cặp dấu {} là một thói quen tốt, giúp tránh nhầm lẫn!

if-then-else

if-then

if(<điều kiện>)

```
{  
  //các câu lệnh sẽ thực hiện nếu điều kiện thoả mãn  
  câu lệnh 1;  
  câu lệnh 2;  
  .....  
}
```

else{

```
  //các câu lệnh thực hiện khi điều kiện không thoả mãn  
  câu lệnh 1;  
  câu lệnh 2;  
  .....  
}
```

Cách xác định đúng hay sai

Ví dụ ta có đoạn script:

Code:

```
a=3;  
b=5;  
if(a<b)  
{  
  alert("Đúng");  
}else{
```

```
alert("Sai");
}
```

Trường hợp này thì kết quả sẽ là bản thông báo với nội dung **Đúng**
Nhưng hãy xét thêm trường hợp:

```
Code:
a=3;
b=5;
if(a<b==false)
{
  alert("Đúng");
}else{
  alert("Sai");
}
```

Thì kết quả sẽ là một hộp thông báo với nội dung là **Sai**
??? 😞 ???

Vì:
bản thân biểu thức **a<b** với a=3 và b=5 mang giá trị đúng(true) mà đúng(true)==sai(flase) thì tất nhiên là sai.

Sử dụng toán tử **or(||)** , **and(&&)**

Bây giờ chúng ta sẽ viết một đoạn script trong đó nếu người ta nhập vào một ký tự số, chương trình sẽ thông báo là người dùng đã nhập một giá trị số, còn bất kỳ ký tự nào khác nó sẽ báo lỗi!

```
Code:
<script language="javascript">
//prompt để nhập dữ liệu gán cho biến c
c=prompt("Nhập một ký tự số từ 0--->9:", "");
//vì một con số 1 chữ sẽ có giá trị từ 0-->9
if((c>=0)&&(c<=9))
{
  alert("Bạn vừa nhập vào một giá trị số");
}
else
{
  alert("Bạn đã nhập sai yêu cầu");
}
</script>
Tự bản thân toán tử <= hay >= đã là một toán tử hoặc
```

else if

Tháng Sáu 13, 2009, 05:02:03 PM gửi bởi [nvcnvn](#)

ELSE IF cũng tương tự như **IF**, nhưng thật ra là một câu lệnh **if** ngay sau về **else**.

tại sao lại dùng tới ư, xem ví dụ nhé:

Nếu điểm Trung Bình năng của con từ 9 trở lên thì quà của con sẽ là một chiếc Novol, từ 7 tới dưới 9 là một chiếc Martin, còn mà dưới 7 là có chuyện đó! 🙄 .

để thể hiện thông điệp đó trong javascript:

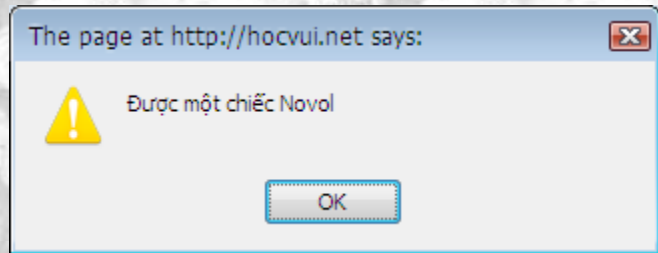
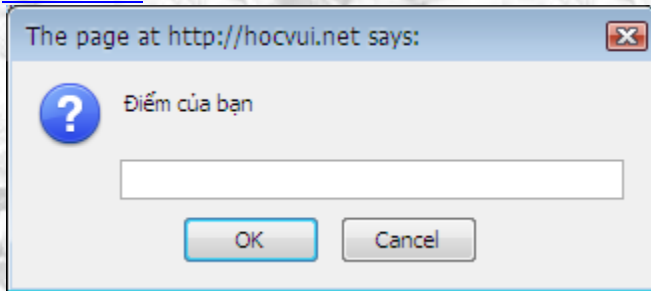
```
Code:
diem=prompt("Điểm của bạn", "");
if(diem>=9)
{
  alert("Được một chiếc Novol");
}else if(diem>=7)
{
```

```

alert("Được một chiếc Martin");
}else
{
alert("Tiêu rồi");
}

```

[Click Xem](#)



bạn thấy cấu trúc else if xuất hiện ở phần giữa chương trình không hãy xem thử ví dụ với điểm của bạn từ 9 trở lên, để thấy được khác biệt, ta xem ví dụ này:

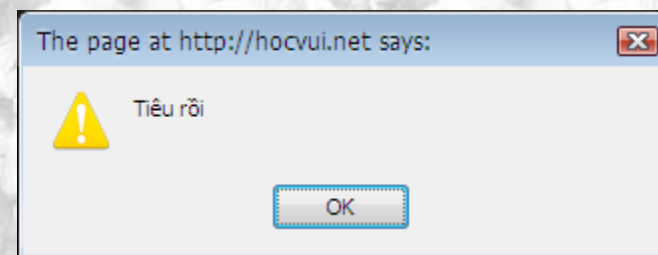
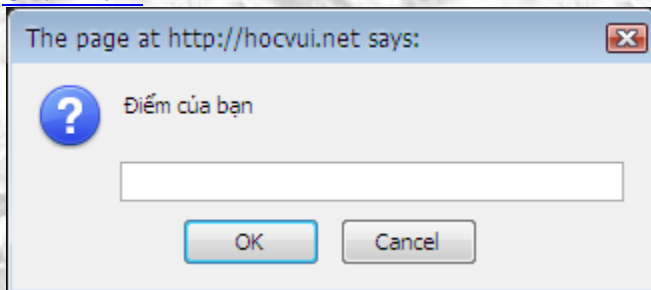
Code:

```

diem=prompt("Điểm của bạn", "");
if(diem>=9)
{
alert("Được một chiếc Novol");
}
if(diem>=7)
{
alert("Được một chiếc Martin");
}else
{
alert("Tiêu rồi");
}

```

[Click Xem](#)



Ta thay cấu trúc else if với một câu if, điều gì sẽ xảy ra khi ta nhập một điểm từ 9 trở lên. Vâng, nó sẽ hiện ra tới hai hộp thông báo, một chiếc Novol và một chiếc Martin, lời nhỉ.

Sở dĩ là vì ở đoạn con dùng cấu trúc else if có nghĩ là khi trường hợp điểm từ 9 trở lên đã thỏa mãn lần if đầu tiên, lần else if chỉ diễn ra khi lần if đầu tiên không thỏa mãn, nói một cách khác, đó là 2 vế của 1 câu lệnh.

Còn ở đoạn code thứ hai, đó là hai câu lệnh riêng biệt, nếu điểm từ chính trở lên, thỏa điều kiện cho câu lệnh 1, thì tất nhiên cũng lớn hơn 7 và thỏa luôn điều kiện câu lệnh 2.

Nếu bạn không dùng tới cấu trúc else if mà muốn dùng toàn câu if thì ta phải thêm điều kiện vào câu lệnh, ví như với đoạn code 2, ta cần thêm vào điều kiện như thế này:

Code:

```

diem=prompt("Điểm của bạn", "");
if(diem>=9)
{
alert("Được một chiếc Novol");
}
if(((diem>=7)&&(diem<9))

```

```
{
alert("Được một chiếc Martin");
}
if(diem<7)
{
alert("Tiêu rồi");
}
```

Nhưng quả thật sẽ rất khó khăn nếu có nhiều mức thưởng nhé!

Cấu trúc else if này rất thuận lợi để bạn giải quyết những vấn đề với nhiều điều kiện và nhiều hành động.

CẤU TRÚC LẶP

Cấu trúc lặp

Thứ Sáu 14, 2009, 04:50:34 PM gửi bởi [nvcnvn](#)

Bạn muốn bày tỏ tình yêu với người ấy, và muốn cho người ấy biết bạn yêu người ấy đến mức nào, và ý tưởng của bạn sẽ là một trang web với hàng ngàn dòng chữ : **I LOVE YOU**

Có hai cách để làm chuyện này:

_Đầu tiên bạn hãy viết ra một câu: **I LOVE YOU**, sau đó copy & past bằng tổ hợp phím **Ctrl + V** cho nó lẹ, được cỡ 5,6 dòng, tô đen hết lại past tiếp, được cỡ một ganf, tô đen rồi past cho nó nhiều, chả mấy chốc cũng sẽ được một ngàn dòng.

_Cách thứ hai là làm mọi thứ với khoảng 5 dòng với javascript, quãng luôn đoạn code để mọi người xem rồi ta cùng đi vào tìm hiểu sau:

Code:

```
<script language="javascript">
var i=0;
while(i<1001)
{
document.write(" <b><font color='\#FF00FF'\>I LOVE YOU</font></b><br />");
i=i+1;
}
</script>
```

Không tính hai thẻ đóng và mở của tag <script> thì ta có 6 dòng, đoạn js này dài 6 dòng nhưng sẽ in ra được 1000 câu anh yêu em với định dạng như trên, bắt đầu phân tích nào:

dòng **var i=0;** khai báo biến đếm (vì nó là để đếm xem bao nhiêu lần rồi nên gọi là biến đếm thôi) i và gán cho nó giá trị =0;

dòng **while(i<1001)** dòng này là linh hồn của cả đoạn, chữ **while** trong tiếng anh có nghĩa là trong khi, còn với cú pháp javascript trên thì dịch cả câu là : **khi điều kiện còn đúng thì làm đi, làm lại cái trong cặp dấu ngoặc móc**, với câu lệnh trên của chúng ta thì điều kiện là **i<1001**.

Hàm **document.write** có vài điểm hơi lạ, các bạn chắc hẳn còn nhớ là để tránh sự nhầm lẫn giữa các dấu nháy, nếu bạn muốn thêm vào trong hàm các phần tử html với các giá trị bao quanh dấu nháy thì phải thay các dấu nháy đôi bằng

dấu nháy đơn, **nhưng** tại sao mình lại dùng được các dấu này kép được, dĩ nhiên các bạn sẽ đoán được là nhờ các dấu \ ngay phía trước các dấu nháy kép, đây là một số lưu ý để sao này các bạn chắc hẳn sẽ cần dùng tới.

Dòng **i=i+1**; đây là dòng tăng giá trị của i lên, phải có dòng này thì tới một lúc nào đó vòng lặp mới ngưng chứ!

Một dạng vòng lặp khác là **vòng lặp for**.

Bạn hãy xem ví dụ với cùng chức năng với đoạn code trên:

Code:

```
for(i=0;i<1001;i++)
{
document.write(" <b><font color=\"#FF00FF\">I LOVE YOU</font></b><BR />");
}
```

tất cả vòng lặp for đều như thế này:

for(phần khởi tạo;phần điều kiện;tăng hay giảm giá trị)

Phần khởi tạo: bạn khai báo biến đếm và gán giá trị ban đầu cho nó luôn.

Phần điều kiện: bạn đặt ra điều kiện.

Phần tăng/giảm giá trị: đây là điểm khác biệt nhất với vòng lặp while, trong khi vòng lặp while thì cần một câu lệnh để tăng/giảm giá trị cho biến đếm, thì bản chất vòng lặp for đã có sẵn điều này.

Mình trong ví dụ trên, điều kiện mình đã đạt theo kiểu toán tự rút gọn (++ có nghĩa là tự tăng lên 1, bạn nên xem lại bài toán tử)

Chú ý: mọi con đường đều về La Mã, ở các ví dụ trên mình chỉ cho tăng biến đếm lên, trong khi vẫn có thể làm như thế với các giảm biến đếm xuống, chỉ cần linh hoạt một chút:

Code:

```
for(i=1001;i>1;i--)
{
document.write(" <b><font color=\"#FF00FF\">I LOVE YOU</font></b><BR />");
}
```

hãy linh hoạt và vận dụng thích hợp cho mục đích của bạn, đó là việc mà chỉ con người mới làm được.

break, dừng vòng lặp

Tháng Sáu 14, 2009, 07:34:45 PM gửi bởi [nvcnvn](#)

Thử đặt ra trường hợp ta có một chương trình học toán cho trẻ, loại học bảng cửu chương ấy(ví dụ bảng cửu chương 2), chương trình như thế này:

- _ Mỗi lần chương trình sẽ in ra màn hình câu: 2x1=?
- _ Một prompt sẽ xuất hiện, yêu cầu trẻ nhập số vào.
- _ Nếu đúng thì chương trình tiếp tục và sẽ kết thúc cho tới khi tới 2x10.
- _ Nếu sai thì chương trình sẽ dừng, và bảo trẻ nên học lại bài.

Giải pháp:

_ Chạy một vòng lặp từ một tới 10, dùng cấu trúc if để xác định đúng sai. **Nhưng làm sao ta có thể dừng vòng lặp lại khi kết quả sai.**

_ Lệnh **break** sẽ là giải pháp, nó sẽ bẻ gãy vòng lặp ngay khi chương trình phát hiện nó.

Code:

```
<script language="javascript">
for(i=1;i<11;i++)
{
    a=prompt("2 lần "+i+" là?", "")
    if(a==(i*2))
    {
        alert("Đúng rồi 2 lần "+i+" là "+a);
    }else{
        alert("Sai rồi, em nên học lại bài cho kĩ nhé")
        break;
    }
}
</script>
```

MẢNG

Mảng trong javascript

Tháng Sáu 09, 2009, 11:40:10 AM gửi bởi [nvcnvn](#)

Nếu bạn đã từng học qua bất cứ ngôn ngữ lập trình nào, thì chắc hẳn sẽ không xa lạ gì với mảng!
Còn nếu bạn là người mới vào nghề như mình, thì đây là một khái niệm mới, cũng khá khó đấy!

Mảng gồm có hai loại, mảng một chiều và mảng nhiều chiều, trong mảng nhiều chiều thông dụng nhất ta dùng tới mảng hai chiều, vì thế bài này chỉ nói về mảng một chiều và mảng hai chiều.

mảng một chiều là dãy hữu hạn các phần tử cùng kiểu. Mảng được đặt tên và mỗi phần tử có một chỉ số. Đề mô tả mảng một chiều cần xác định kiểu của phần tử và cách đánh số các phần tử của nó.

Nhớ lại ví dụ về biến mà mình mô tả với bao thuốc lá ớ, bây ta ví dụ trường hợp ta có **10** bao thuốc là và bạn quyết định đặt tên cho chúng là **bao1, bao2, ..., bao8, bao9, bao10**, với mục đích là mỗi bao chứa một cái gì đó, tới khi nào cần cái gì thì lấy ra mà dùng.

Còn mảng, mảng chính là cả cây thuốc lá! 😊, đơn giản chỉ là một cây thuốc là, bên trong có những bao thuốc lá! giải thích cho dài dòng, **mảng 1 chiều đơn thuần là một tập hợp các biến**, và các biến có thứ tự thay vì tên!

tạo một mảng:

Ví dụ toạ một mảng đơn giản

Code:

```
mang=Array("a","b","c");
alert(mang[0]);
Thử in ra phần tử thứ nhất của mảng(kết quả sẽ là a)
```

Đây cũng là một các để khai báo mảng, các phần tử của mảng được đặt trong dấu nháy kép và chúng được ngăn cách nhau giữa dấu (,)

Còn đây là cách thứ hai, khá dễ nhìn!

Bạn hãy khai báo mảng, không cần khai báo nội dung bên trong:

Code:

```
mang=Array();  
sau đó, khai báo từng phần tử bằng cách này:
```

Code:

```
mang[0]="a";  
mang[1]="b";  
mang[2]="c";  
mang[3]="d";
```

Chú ý: Trong javascript phần tử đầu tiên của mảng có thứ tự là 0.

Một số hàm làm việc với mảng:

`<tên mảng>.length`: Xác định số phần tử của mảng.

...

Tại sao lại dùng đến mảng

Tháng Sáu 14, 2009, 08:19:13 PM gửi bởi [nvcnvn](#)

tại sao lại dùng đến mảng

Câu trả lời: mảng là cách tuyệt vời để quản lý dữ liệu. Người ta xây dựng khá nhiều các hàm để làm việc với mảng

Xét bài toán:

Tính tổng số tiền thu được trong tuần, tính và in ra màn hình số ngày có thu nhập cao hơn trung bình, thấy ngay nếu dùng biến đơn giản, ta cần tới 7 biến để lưu trữ thu nhập của mỗi ngày, một biến b để lưu trữ số ngày có thu nhập cao hơn TB, biến b lưu trữ giá trị Tb để so sánh.

Để người ta nhập vào 7 biến này bằng prompt.

Sau đó tính trung bình.

Cho một dãy 7 câu lệnh if nếu ngày nào có thu nhập cao hơn TB thì biến b tăng lên một đơn vị.

Code:

```
var a=0;  
var b=0;  
t1=prompt("Thu nhập ngày thứ nhất", "");  
t2=prompt("Thu nhập ngày thứ 2", "");  
t3=prompt("Thu nhập ngày thứ 3", "");  
t4=prompt("Thu nhập ngày thứ 4", "");  
t5=prompt("Thu nhập ngày thứ 5", "");  
t6=prompt("Thu nhập ngày thứ 6", "");  
t7=prompt("Thu nhập ngày thứ 7", "");  
b=(eval(t1)+eval(t2)+eval(t3)+eval(t4)+eval(t5)+eval(t6)+eval(t7))/7;  
alert(b);  
if(t1>b)
```

```
{
a=a+1
}
if(t2>b)
{
a=a+1
}
if(t3>b)
{
a=a+1
}
if(t4>b)
{
a=a+1
}
if(t5>b)
{
a=a+1
}
if(t6>b)
{
a=a+1
}
if(t7>b)
{
a=a+1
}
alert("Số ngày có thu nhập cao hơn Trung bình tuần là "+a);
```

Phù, 7 ngày đã xong, vậy thì bữa sau hết tháng, tính cả tháng! Cẩn lười quá 😊

Mãng sẽ giúp ta rút ngắn quá trình trên, **nhưng** tất nhiên là không phải chỉ dùng riêng mình nó.

Xem lại một chút bài vòng lặp, ta có thể dùng vòng lặp để lướt qua cả mảng, gán giá trị cho cả mảng. Nói chung là khá lẹ!

_ Hướng đi là cho vòng lặp duyệt qua mảng, vừa duyệt vừa gán giá trị

Code:

```
<script language="javascript">
//Khai báo mảng là phần tử mảng
mang=Array();
//Số ngày yêu cầu, giả sử là 30
songay=5;
//biến tổng lưu giữ giá trị tổng số tiền thu được
tong=0;
//biến tb lưu giữ giá trị trung bình
tb=0;
//biến num lưu giữ số ngày có thu nhập cao hơn
num=0;
//bắt đầu vòng lặp
for(i=0;i<songay;i++)
{
```



```

    mang[i]=prompt("Thu nhập của ngày "+(i+1), "");
    //mỗi lần vòng lặp chạy, biến tổng được tăng lên
    tong=eval(tong)+eval(mang[i]);
}
tb=tong/songay;
for(i=0;i<mang.length;i++)
{
    if(mang[i]>tb)
    {
        num=num+1;
    }
}
alert("Số ngày trong "+songay+" có thu nhập cao hơn trung bình "+tb+" là "+num);
</script>

```

Ví dụ trên hơi rắc rối nhưng nếu xem xét kỹ, việc một mảng duy nhất thay thế công dụng của 7 biến quả là lời nhỉ!

Array function - các hàm làm việc với mảng

Tháng Sáu 17, 2009, 04:24:38 PM gửi bởi [nvcnvn](#)

Trước tiên mình tạo một mảng và nó sẽ được dùng trong suốt bài này:

Code:

```

mang=Array();
mang[0]="anh";
mang[1]="yêu";
mang[2]="em";

```

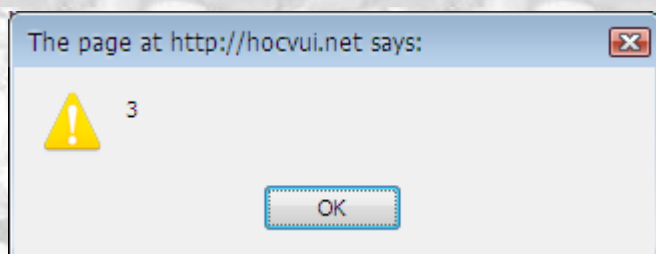
Các **thuộc tính** và **Method** để làm việc với đối tượng đều có dạng:

<tên đối tượng>.**xxx**

Trong đó **xxx** chính là các **thuộc tính** và **Method** (nhớ là có dấu chấm ha)

length Trả về số phần tử của mảng

[>Xem](#)

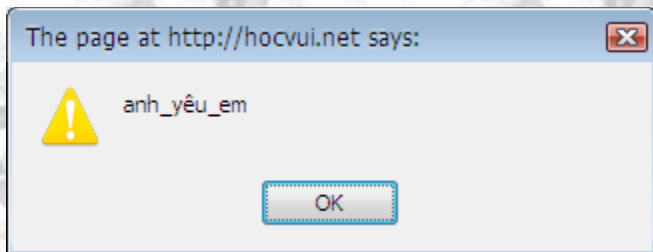


join tạo một chuỗi từ một mảng, các phần tử chuỗi sẽ các nhau kí tự bên trong cặp ()

Code:

```
mang.join("_")
```

[Xem](#)

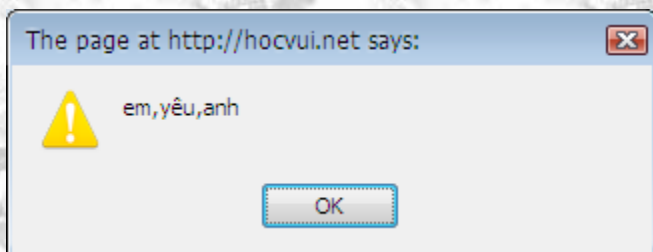


reverse Trả lại một mảng với thứ tự các phần tử bị đảo ngược

Code:

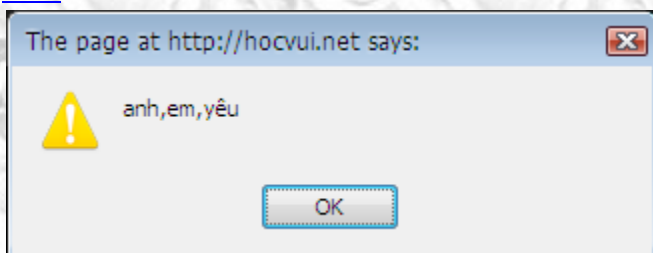
```
mang.reverse()
```

[Xem](#)



sort Trả lại mảng mà các phần tử được sắp xếp theo bảng chữ cái

[Xem](#)

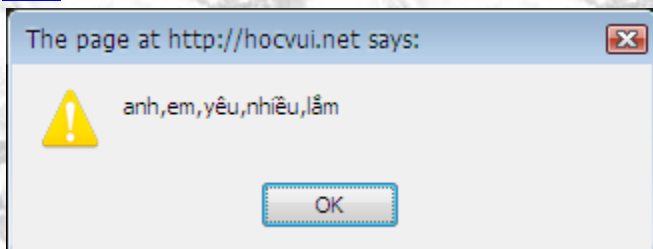


concat Nối nhiều phần tử vào một mảng

Code:

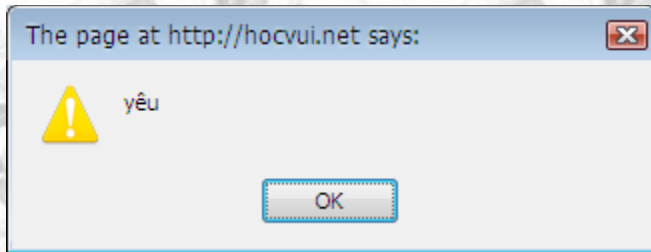
```
mang.concat("nhiều", "lắm")
```

[Xem](#)



pop trả lại phần tử cuối cùng và xoá nó ra khỏi mảng

[Xem](#)

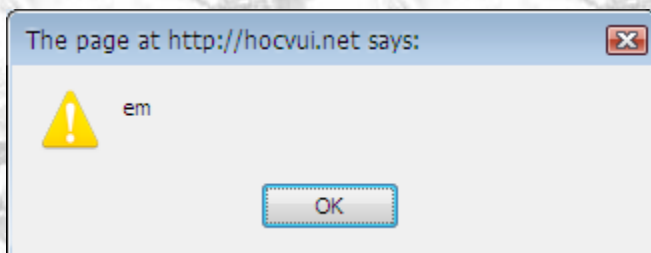


splice chia nhỏ mảng ra, hàm này nhận vào 2 đối số, đối số thứ nhất là vị trí của phần tử bắt đầu, đối số thứ hai là vị trí kết thúc.

Code:

```
mang.splice(1,2)
```

[Xem](#)



Mảng hai chiều javascript

Tháng Sáu 17, 2009, 10:50:43 AM gửi bởi [nvcnvn](#)

Một khi đã hiểu được mảng một chiều, thì sẽ rất đơn giản để hiểu về mảng hai chiều.

Mảng hai chiều thật chất chỉ là một mảng, mà mỗi phần tử lại chứa một mảng khác. Ví dụ thực tế có thể là cái bàn cờ hay hệ trục tọa độ.

Khai báo cũng khá đơn giản

```
mang2c=Array(Array(1,2,3),Array("a","b","c"))
```

Có thể khai báo một cách đơn giản hơn bằng cách:

Code:

```
<script language="javascript">  
var mang1=Array(1,2,3);  
var mang2=Array("a","b","c")  
var mang2c=Array(mang1,mang2);  
//chạy và xem thử kết quả nhé  
alert(mang2c[0][0]);  
</script>
```

Duyệt qua mảng 2 chiều với hai vòng lặp

Ta dùng hai vòng lặp để duyệt qua mảng, vòng lặp thứ nhất chứa vòng lặp thứ hai, khi vòng lặp thứ nhất chạy một lần duyệt phần tử thứ nhất của mảng lớn (chính là mang1 trong ví dụ) thì vòng lặp thứ hai sẽ duyệt hết các phần tử trong mang1, tiếp theo vòng lặp lớn sẽ chuyển tới phần tử thứ 2, vòng lặp nhỏ.....

Mảng dựng sẵn

Tháng Sáu 23, 2009, 09:09:37 AM gửi bởi [nvcnvn](#)

Khi trình duyệt đọc mã HTML, nó tự động tạo ra các mảng, các mảng đó chính là các HTML Element, và ta có thể dùng chúng như một cách tham chiếu tới đối tượng ta muốn.

Ví dụ như ta có một loạt các hình ảnh trong trang với các tag image, trình duyệt sẽ tạo ra một mảng để lưu trữ các image này. Để tham chiếu tới một image nào đó ta:

document.images[<thứ tự ảnh>]

trình duyệt đọc code theo thứ tự từ trên xuống dưới, từ trái qua phải, nó phát hiện cái nào trước thì thứ tự nhỏ nhất.

VD: ta có đoạn code của trang html như sau:

Code:

```
<html>
<head>
<title>Hocvui.net - Trang ví dụ</title>
</head>
<body>
  <form name="form_01">
    <input type="text" name="text_01" />
    <input type="text" name="text_02" />
  </form>
  <form name="form_02">
    <input type="text" name="text_03" />
    <input type="text" name="text_04" />
  </form>
</body>
</html>
```

Ở đây, ngoài cách tham chiếu tới bất cứ phần tử nào trong form, để đặt giá trị chặn hạn, ví dụ như là **text_03** chẳng hạn. ta có thể viết một đoạn code bình thường như sau:

Code:

```
<script language="javascript">
document.form_02.text_03.value="Một cách tham chiếu";
</script>
hay theo những thông tin mới về những mảng dựng sẵn
```

Code:

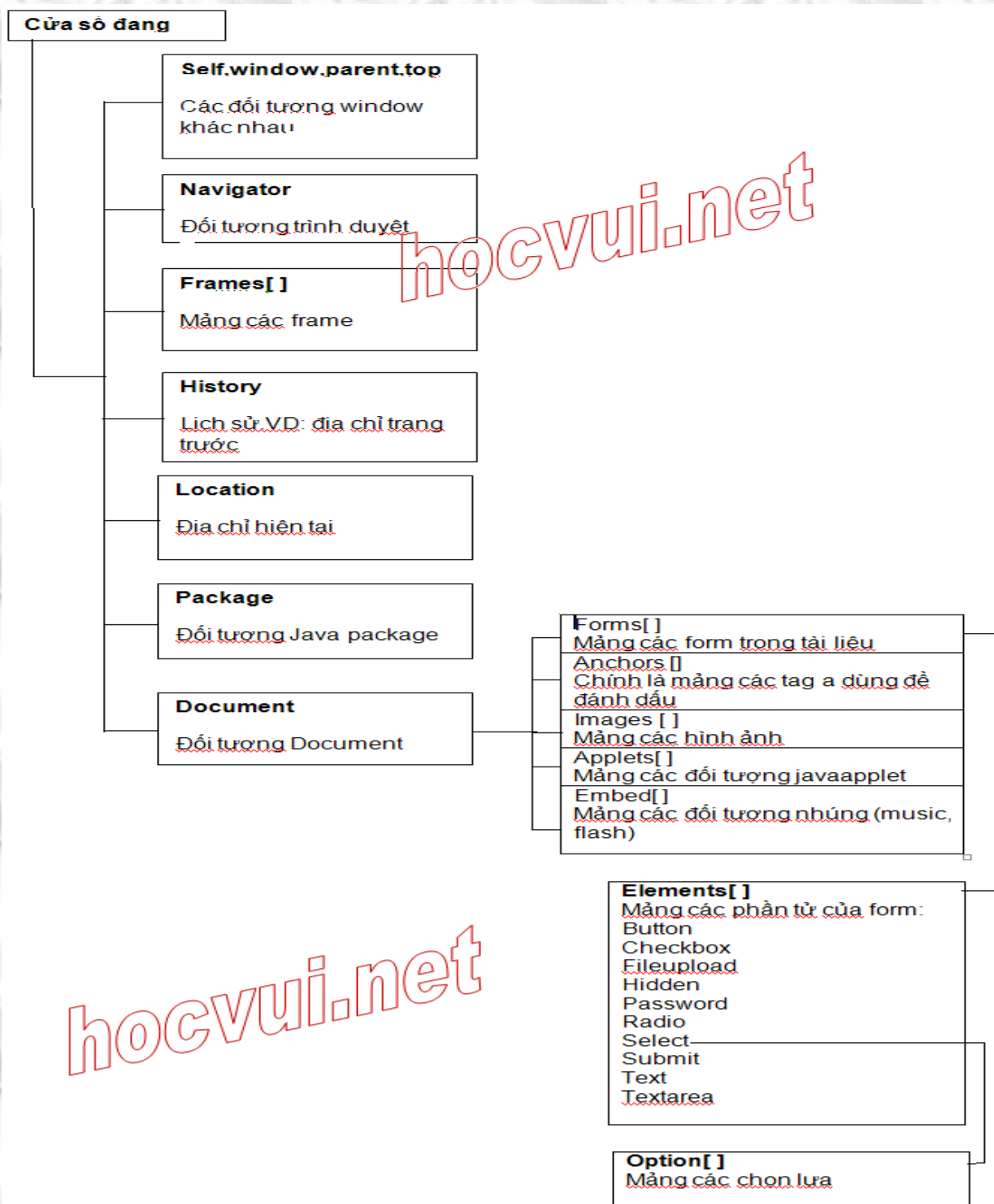
```
<script language="javascript">
// đoạn này dịch là: tìm trong văn bản, form thứ hai, phần tử thứ nhất của form, đặt giá trị
document.forms[1].elements[0].value="Một cách tham chiếu";
</script>
```

Chú ý: nếu các bạn có ý định chạy hai đoạn javascript kia trên đoạn mã html mình lấy ra ví dụ, các bạn cần đặt đoạn javascript bên dưới cả hai form, điều này là vì **trình duyệt đọc code theo thứ tự từ trên xuống dưới, từ trái qua phải** khi nó bắt gặp đoạn js, nó sẽ chạy mã khi, trong khi đó nó chưa tìm thấy những đối tượng mà js sẽ làm việc!

Đây chỉ là một vài trong số rất nhiều mảng dựng sẵn, và thật ra các bài hướng dẫn javascript của hocvui.net cũng chỉ mới nói tới những gì mà mình nghĩ là gần gũi và cần thiết nhất đối với các bạn.

Một câu hỏi trong lúc mà mình học javascript là làm sao biết những mảng nào được dựng sẵn, làm sao biết những đối tượng nào có những thuộc tính nào. Để giải thích cho điều đó ta hãy tìm câu hỏi với google và từ khóa HTML DOM. Bảo đảm với một chút vốn tiếng Anh, bạn sẽ giải đáp được hết những thắc mắc trên. và tất nhiên, nếu có khả năng, mình sẽ tìm, tổng hợp lại các tài liệu đó và dịch ra tiếng Việt để phục vụ cho những người Việt Nam ham học hỏi.

Dưới đây mình có một mô hình thiểu sót mô tả DOM (Document Object Model)



Các bạn hãy cố gắng tìm hiểu thêm nha.

JS FUNCTION - HÀM

Tự soạn thảo các hàm javascript

Tháng Sáu 18, 2009, 03:10:34 PM gửi bởi [nvcnvn](#)

Một số hàm wen thuộc với các bạn cho tới lúc này chính là các hàm **alert,prompt**, các hàm này là những hàm đã có sẵn trong javascript, và bạn cũng có thể tự soạn thảo cho mình một hàm.

Hàm - function các bạn tự viết giống như là một dạng viết tắt, qua nhiều ví dụ, ví dụ viết ra hàng ngàn câu **I Love You**, qua công cụ là các ngôn ngữ lập trình ta đã rút ngắn mọi việc chỉ còn vài hàng, nhưng sẽ không hay nếu bạn lặp đi lặp lại nhiều lần dù chỉ một đoạn code đó trên trang web.

Xét ví dụ

Bạn hãy chắc chắn biết cách viết ra 1000 dòng **I Love You** với vòng lặp while. Lần này bạn muốn tô điểm thêm cho trang web bằng cách, nói một nghìn lần **I Love You**, sau đó là một bức ảnh trái tim, rồi lại nói 1000 lần **I Love You**, lại ảnh trái tim, 1000 lần **I Love You**, một đóa hoa,.....

Nói chung là phải vài lần 100 lần cách nhau, và nói cách riêng :) là ta phải lặp đi lặp lại đoạn code:

Code:

```
for(i=0;i<1001;i++)
{
document.write(" <b><font color=\"#FF00FF\">I LOVE YOU</font></b><BR />");
}
```

Ở nhiều phần của site, điều này sẽ làm cho ta khó hiệu chỉnh, giả như khi tình yêu bớt mặn nồng, bạn muốn giảm số lần nói anh yêu em xuống còn 500, vậy là phải moi cho bằng hết ra mà sửa.

Và nói mọi thứ cho dài dòng thì câu hàm là một giải pháp chắc các bạn cũng đoán ra

hàm - function cú pháp khai báo có dạng như thế này:

Code:

```
function <tên hàm>()
{
//Nội dung các câu lệnh sẽ nằm trong hai dấu ngoặc nhọn này
}
```

Ví dụ mình muốn tạo hàm **say_Iloveyou**(quy tắc đặt tên hàm giống như quy tắc đặt tên biến) với công dụng là viết ra 1000 câu **I Love You** thì code sẽ là:

Code:

```
function say_Iloveyou()
{
for(i=0;i<1001;i++)
{
document.write(" <b><font color=\"#FF00FF\">I LOVE YOU</font></b><BR />");
}
}
```

Và để chạy hàm này, bạn hãy đặt dòng:

Code:

```
say_Iloveyou();
```

ở bất cứ đâu các bạn muốn in ra 1000 câu tình cảm đó. và khi muốn sửa lại chỉ còn 500 hay 5 câu gì đó, thì bạn chỉ việc sửa lại code bên trong function.

Hàm của chúng ta giống như các hàm khác, sẽ rất hữu dụng khi chúng ta kết hợp với các sự kiện.

Hàm **say_Iloveyou()** không giống những hàm mà ta đã biết như là alert hay prompt, nó không hề nhận vào một tham số nào trong cặp dấu ngoặc (). Trong bài sau ta sẽ tìm hiểu thêm vấn đề này.

Hàm và các tham số

Tháng Sáu 19, 2009, 09:16:08 AM gửi bởi [nvcnvn](#)

Giống như hàm **alert** với việc nhận vào một tham số đặt bên trong cặp dấu (), ta có thể dễ dàng và linh hoạt thay đổi nội dung xuất hiện bên trong hộp thoại.

Quay lại ví dụ về hàm viết 1000 câu **I Love You**. Hãy đặt ra trường hợp tương tự như ở ví dụ trước, bạn muốn in ra nhiều dòng chữ **I Love You**, nhưng số lượng không đều nhau (ví dụ: lần đầu bạn in 100 câu, lần hai in 390 câu, lần 3 in 5 câu chẵn hạn). Dĩ nhiên đoạn code cũ của bạn:

Code:

```
function say_Iloveyou()
{
  for(i=0;i<1001;i++)
  {
    document.write("<b><font color='#FF00FF'>I LOVE YOU</font></b><BR />");
  }
}
```

sẽ không thể nào tùy chỉnh như ý muốn của bạn và công việc của chúng ta sẽ không còn tối ưu nữa.

Thật may, có một cách để cho các hàm của bạn linh động hơn với việc dùng những tham số. Hãy nhìn vào ví dụ:

Code:

```
function say_Iloveyou(num)
{
  for(i=0;i<num;i++)
  {
    document.write("<b><font color='#FF00FF'>I LOVE YOU</font></b><BR />");
  }
}
```

Hãy chú ý dòng đầu tiên, cái bạn cần nhìn là từ đặt trong cặp dấu (), từ đó là một biến lưu giữ giá trị mà bạn cần dùng trong hàm.

Sau đó hãy để mắt nhìn qua dòng bắt đầu vòng lặp (hàng thứ 3), bạn có nhận thấy điều gì khác với đoạn code cũ của chúng ta chứ. Đó chính là thay vì con số **1001** mình thay bằng từ **num** là từ nằm trong cặp dấu ngoặc kép trên phần khai báo tên hàm. Điều đó có nghĩa là gì?

Code:

```
for(i=0;i<1001;i++)
```

Nếu bạn đã xem qua kĩ phần vòng lặp, bạn sẽ hiểu ý nghĩa của cả phần này , ta hãy chú ý tới đoạn in đậm nhé:

for(i=0;i<1001;i++), phần in đậm chính là phần điều kiện (như ta đã biết) điều kiện ở đây là *i<1001*, khi mình thay giá trị 1001 vào bằng biến **num**, thì điều kiện sẽ là cho tới khi nào còn bé hơn **num**

Và như vậy, khi gọi hàm ta có thể tùy chỉnh có bao nhiêu câu xuất hiện với một con số đặt trong dấu ngoặc, tương ứng với biến num.

Copy hết ví dụ này, xem kĩ, rồi chạy thử nhé!

Code:

```
<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Javascript Funtion</title>
</head>

<body>
<script language="javascript">
function say_Iloveyou(num)
{
  for(i=0;i<num;i++)
  {
    document.write(" <b><font color='\#FF00FF'\>I LOVE YOU</font></b><br />");
  }
}
</script>
Nói 9 lần I Love You<br />
<script language="javascript">
say_Iloveyou(9);
</script>
Nói 99 lần I Love You<br />
<script language="javascript">
say_Iloveyou(99);
</script>
Nói 999 lần I Love You<br />
<script language="javascript">
say_Iloveyou(999);
</script>
</body>

</html>
```

Đưa nhiều tham số vào hàm

Giống như hàm prompt nhận vào tới hai tham số, ta tất nhiên có thể đưa vào thêm các tham số trong hàm để hàm trở nên linh động hơn.

Tất cả hàm trên của chúng ta đều in ra câu **I Love You**, với định dạng in đậm và màu hồng, ta có thể thêm tham số vào để thay đổi định dạng và màu chữ, hoặc là thay đổi luôn cả câu chữ. Mọi việc cần làm là đặt tham số đúng chỗ. Sau đây là ví dụ thay đổi luôn cả nội dung câu:

Code:

```
<script language="javascript">
function say_Iloveyou(num,cau)
```



```
{
  for(i=0;i<num;i++)
  {
    document.write(cau);
  }
}
```

Chỉ cần thêm một tham số **cau** vào, tham số này cung lưu trữ một giá trị mà bạn cần làm việc, nếu mình muốn in ra bảy lần câu "Anh ghét em" chẳng hạn, thì mình sẽ gọi hàm như thế này:

Code:

```
say_Iloveyou(7,"anh ghét em <br />");
Chú ý là nếu giá trị gán vào cho tham số là chuỗi thì nó cần đặt trong cặp dấu nháy ""
```

Trên đây chỉ là những hàm đơn giản nhất, với những code đơn giản nhất, chỉ với những kiến thức thông thường như thế này thôi, nhưng với bộ óc con người, chúng ta sẽ phát triển nó lên một tầm cao mới.

Biến toàn cục và biến cục bộ

Tháng Sáu 19, 2009, 09:56:47 AM gửi bởi [nvcnvn](#)

Vấn đề của hai loại biến toàn cục và cục bộ mình đã trình bày ngay từ đầu! Nhưng chỉ giải thích sơ qua là khác nhau ở cái tên và không đi sâu thêm, bởi muốn hiểu vấn đề này, chúng ta cần phải đi qua phần hàm - function để có thể hiểu rõ nhất.

Nhắc lại chút:

Biến cục bộ(ngay từ từ cục bộ cũng cho biết nó chỉ hoạt động ở một bộ phận nhất định)

Biến cục bộ khi khai báo bắt đầu với từ khoá **var**

Biến toàn cục đương nhiên khác với biến toàn cục ở chỗ nó có giá trị ở toan bộ phận.

Biến toàn cục khi khai báo không có từ khoá **var**.

Trong các ví dụ ít khi nào mình dùng tới từ khoá var, bởi vì đơn giản mình muốn các ví dụ đơn giản nhất có thể, dĩ nhiên, các ví dụ đó cũng không hề bị ảnh hưởng bởi việc dùng loại biến nào.

Xét lại ví dụ của loạt bài mạng nói về công dụng : Tính tổng số tiền thu được trong tuần, tính và in ra màn hình số ngày có thu nhập cao hơn trung bình.

Đoạn code của chúng ta đây:

Code:

```
<script language="javascript">
//Khai báo mảng là phần tử mảng
mang=Array();
//Số ngày yêu cầu, giả sử là 30
songay=5;
//biến tổng lưu giữ giá trị tong so tien thu duoc
tong=0;
//biến tb lưu giữ giá trị trung bình
tb=0;
```

```
//biến num lưu giữ số ngày có thu nhập cao hơn
num=0;
//bắt đầu vòng lặp
for(i=0;i<songay;i++)
{
    mang[i]=prompt("Thu nhập của ngày "+(i+1), "");
    //mỗi lần vòng lặp chạy, biến tổng được tăng lên
    tong=eval(tong)+eval(mang[i]);
}
tb=tong/songay;
for(i=0;i<mang.length;i++)
{
    if(mang[i]>tb)
    {
        num=num+1;
    }
}
alert("Số ngày trong "+songay+" có thu nhập cao hơn trung bình "+tb+" là "+num);
</script>
```

Đó là đoạn code cũ, bây giờ mình đã biết về hàm, điều đầu tiên mình nghĩ tới là tùy biến để đoạn code không chỉ làm việc cho 7 ngày mà có thể là 1 tháng hay 2 tuần tùy ý! code lúc này là:

Code:

```
<script language="javascript">
function tinhtoan(numngay)
{
//Khai báo mang là phần tử mảng
mang=Array();
//Số ngày yêu cầu, lúc này được gán với giá trị numngay là tham số duy nhất của hàm tintoan
songay=numngay;
//biến tổng lưu giữ giá trị tong so tien thu đươc
tong=0;
//biến tb lưu giữ giá trị trung bình
tb=0;
//biến num lưu giữ số ngày có thu nhập cao hơn
num=0;
//bắt đầu vòng lặp
for(i=0;i<songay;i++)
{
    mang[i]=prompt("Thu nhập của ngày "+(i+1), "");
    //mỗi lần vòng lặp chạy, biến tổng được tăng lên
    tong=eval(tong)+eval(mang[i]);
}
tb=tong/songay;
for(i=0;i<mang.length;i++)
{
    if(mang[i]>tb)
    {
        num=num+1;
    }
}
alert("Số ngày trong "+songay+" có thu nhập cao hơn trung bình "+tb+" là "+num);
}
</script>
```

bây giờ bạn hãy bắt đầu gọi hàm này ở bất cứ đâu trên trang web. Nhưng hãy thêm một số dòng bên dưới như mình đây:

Code:

```
<script language="javascript">
// Ở đây mình gọi hàm để tính toán trong 9 ngày
tinhtoan(9);
// Đưa ra số ngày đã tính toán
alert("Số ngày tính toán: "+num);
// thông báo riêng tổng thu nhập
alert("tổng thu nhập: "+tong);
// và mức thu nhập trung bình
alert("thu nhập trung bình: "+tb);
</script>
```

Sau khi bạn nhập đầy đủ thông tin, cả thầy sẽ có 4 hộp thông báo, 1 vốn dĩ của hàm, 1 Đưa ra số ngày đã tính toán, 1 thông báo riêng tổng thu nhập, 1 mức thu nhập trung bình.

Nhưng mọi việc thường hay chuyển hướng với từ **nhưng**....

Ta thay đổi một số dòng ở các đoạn khai báo function, cụ thể là thêm từ khóa **var** vào trước dòng khai báo hai biến, **tong** và **tb**

Code:

```
//Khai báo mang là phần tử mảng
mang=Array();
//Số ngày yêu cầu, giả sử là 30
songay=numngay;
//biến tong lưu giữ giá trị tong so tien thu duoc
var tong=0;
//biến tb lưu giữ giá trị trung bình
var tb=0;
//biến num lưu giữ số ngày có thu nhập cao hơn
num=0;
```

rồi hãy gọi hàm và thêm một số dòng như trường hợp trên, mọi thứ sẽ không như bình thường nữa. Đầu tiên là một hộp thông báo của hàm, thứ hai là hộp thông báo Đưa ra số ngày đã tính toán, còn hai hộp thông báo kia sẽ "vấn bóng"

Đơn giản vì khi khai báo từ khóa **var** trước hai biến **tong** và **tb**, lúc này chúng trở thành hai biến cục bộ, và như đã nói, biến cục bộ chỉ có giá trị trong một bộ phận(cụ thể trường hợp này chính là hàm *tinhtoan*), khi ta gọi hai biến này ngoài cấu trúc hàm, javascript sẽ cho rằng những hàm này chưa tồn tại, chưa được khai báo(vốn dĩ ta đã khai báo trong hàm) và lỗi sẽ xảy ra khiến hai hợp thoại không thể xuất hiện.

Lại nhấn mạnh một lần nữa, tùy theo mục đích của trương trình, chính lập trình viên sẽ quyết định xem dùng những gì cho thích hợp nhất với chương trình. Và đó là một yếu tố không thể thiếu, yếu tố con người!

javascript - function return

Tháng Sáu 19, 2009, 02:54:29 PM gửi bởi [handsome](#)

Bạn có nhớ những hàm tính toán không, giống như **Math.sqrt()** nên trong bài **Một số hàm hay dùng trong JavaScript**

Bản thân các hàm này mang luôn một giá trị. Có nghĩa là ta có thể lấy ngay:

```
5 + Math.sqrt(4) //bằng mấy nhỉ
```

Các hàm từ trước tới giờ ta viết chỉ mới thực hiện hành động mà ta muốn, chứ chưa hề mang giá trị nào. Để có thể gán cho hàm một giá trị ta dùng lệnh **return**.

Xét hàm đơn giản tính tổng và đưa ra màn hình hai số.

Code:

```
<script language="javascript">
function tinhong(a,b)
{
    alert(a+b)
}
</script>
<script language="javascript">
function tinhong(a,b)
{
    alert(a+b)
}
</script>
```

```
<a href="javascript: tinhong(2,3)">Xem</a><br />
```

Dùng return cho hàm này:

Code:

```
function tinhong(a,b)
{
    var kq=a+b;
    return kq;
}
</script>
```

Lúc này nếu bạn gọi hàm này một cách bình thường thì sẽ chả có gì xảy ra, nhưng hãy thử:

Code:

```
alert(tinhong(2,3));
```

Hoặc:

Code:

```
alert(tinhong(2,3)+tinhong(3,7));
```

Bạn sẽ thấy lúc này của hàm đã được gán giá trị. Ngoài ra ta cũng có thể gán cho nó những giá trị boolean, và dùng hàm làm điều kiện.

Xét đoạn code so sánh ba số người dùng nhập vào, nếu cả ba số bằng nhau thì thông báo "Đúng" ngược lại "Sai".

Đầu tiên sẽ là đoạn không dùng tới funtion

Code:

```
<script language="javascript">
a=prompt("Số a", "");
b=prompt("Số b", "");
```

```
c=prompt("Số c", "");
if((a==b)&&(a==c)&&(c==b))
{
    alert("Đúng");
}else{
    alert("Sai");
}
```

</script>

Cũng công dụng tương tự, mình viết một chương trình dùng tới funtion:

Code:

```
function vd2()
{
a=prompt("Số a", "");
b=prompt("Số b", "");
c=prompt("Số c", "");
if((a==b)&&(a==c)&&(c==b))
{
    return true;
}else{
    return false;
}
}
```

Rồi ở bất kì đâu trên trang trình, bạn dùng một cấu trúc if:

Code:

```
if(vd2())
{
    alert("Đúng");
}else{
    alert("Sai");
}
```

Lúc này hàm vd2() đã tính toán và gán cho chính nó một giá trị hoặc *true* hoặc *false* và có thể dùng làm điều kiện.

Thực tế việc return giá trị của một hàm về được ứng dụng rất nhiều, một trong số đó sẽ được nêu ra ở loạt bài về làm việc với form.

FORM

Giải đáp thắc mắc về những dấu chấm

Tháng Sáu 17, 2009, 05:18:33 PM gửi bởi [nvcnvn](#)

Bắt đầu từ bài sau, các bạn sẽ làm việc với những html forms, cái mà nắm vị trí chủ đạo trong việc tương tác với người dùng.

Bạn sẽ thấy có rất nhiều dấu chấm trong khi ta làm việc, những dấu chấm này không xa lạ, ví dụ như:

```
document.write() //viết ra màn hình cái gì đó  
array.sort() //Sắp xếp các phần tử trong mảng
```

Đó là cách mà ta tham chiếu đến cái gì đó và làm những cái gì đó, nếu bạn đã xem qua toàn bộ các bài viết html tại <http://hocvui.net/html> thì chắc bạn sẽ nhớ cái mình gọi là **tính lồng ghép**, phần tử này chứa trong phần tử kia, và các phần tử chứa trong đối tượng document và đối tượng lớn nhất chính là window.

Các dấu chấm trong một dòng js sẽ phân tách các đối tượng, khi javascript bắt gặp một chuỗi các đối tượng được ngăn cách bởi các dấu chấm, nó sẽ đi dần xuống đối tượng cuối cùng trong chuỗi đó.

Mọi việc sẽ dần giã rỏ ra khi các bạn xem các bài sau! Hãy học cùng hocvui.net nhé.

javascript làm việc với các form

Tháng Sáu 17, 2009, 07:21:09 PM gửi bởi [nvcnvn](#)

Để học qua phần này, các bạn bắt buộc phải tìm hiểu html form qua các bài viết này:

http://hocvui.net/html/?id_post=1436

html - forms - biểu mẫu

Tháng Năm 31, 2009, 03:39:04 PM gửi bởi [nvcnvn](#)

Cho đến lúc này, mình tin là nếu các bạn đã thử dùng vài công cụ hỗ trợ thiết kế web như MS FrontPage hay Dreamweaver thì các bạn đã có thể thể hiện trang web theo ý muốn của mình, tuy là chỉ dùng html thì sẽ không thể nào tối ưu hoá về mặt kĩ thuật.

Có 2 điều làm cho trang web trở nên hấp dẫn: **giao diện, tính tương tác với khách hàng.**

Vấn đề giao diện thì mặc dù hơi khó nhưng các bạn cũng đã phần nào giải quyết được, tùy theo thị hiếu khách hàng mà tùy ta lựa chọn (vd: khách hàng là tầng lớp trung niên thì tông màu nhẹ nhàng, tuổi teen thì màu mè lòe loẹt). Nhưng vấn đề giao tiếp với khách hàng thì có thể khác nhau về hình thức nhưng cơ bản thì luôn giống nhau, và lại một lần nhìn vào thực tế **bạn chưa thể nào giao tiếp với khách hàng** cho tới khi bạn biết được mọi **server-script**(php,asp....) hay ít nhất một **client-script**(javascript,vbscript....)

Tuy đã nói là không biết một trong hai loại script trên thì không thể giao tiếp với khách hàng, nhưng mà html lại như là cầu nối, thiếu html thì cũng hơi khó mà mang được cái gì!

Hình thức giao tiếp già cô nhất là cái không trông cho ta điền thông tin vào rồi nhấn nút, đợi một hồi thì cái gì mình điền vào sẽ được gửi đi, hay là mấy cái ô rồi mình click chuột vào check, đó chính là các phần tử con bên trong phần tử **form**.

Đề thông tin của các phần tử trên được gửi đi, chúng cần được đặt trong cặp thẻ **<form>**, trong thẻ form này cũng có một vài thuộc tính.

Code:

```
<form action="index.html" method="post" name="the_form">
```

```
</form>
```

action: đó là link mà sau khi form được gửi đi, trang web sẽ được chuyển tới, trang này có thể là một trang bất kì hay chính trang chứa form, trang này chứa code xử lý thông tin form theo cách mà bạn muốn. Code đó có thể là 1 client-script hay sever-script.

method: là hình cách thức gửi form đi, có hai giá trị là *post, get*, giá trị *get* là mặc định, bạn hãy tự viết một trang chứa form sau bài này để nhận thấy sự khác biệt giữa *post* và *get*, để xem kĩ năng quan sát của bạn như thế nào.

name: cái này thì các bạn biết rồi, trong html đơn thuần cái này chả giúp được gì, nhưng mà form là để phối hợp với các ngôn ngữ khác, những ngôn ngữ này dùng tới cái *name* này khá nhiều.

Thẻ **form** không trực tiếp show cái gì ra màn hình cả, và cũng chả có giá trị gì nếu thiếu các phần tử con bên trong. Một số phần tử của form ở các bài sau.

...

http://hocvui.net/html/?id_post=1441

html from input elements

Tháng Năm 31, 2009, 08:19:46 PM gửi bởi [nvcvn](#)

<input>

cái thẻ input này hiển thị ra được nhiều thứ lắm, nhờ giá thuộc tính **type**, vừa ví dụ vừa kê luôn một số loại phổ biến:

text

Code:

```
<input type="text" name="type_text" />
```

Có thể đặt sẵn chữ vào trong cái ô đó bằng cách thêm thuộc tính **value** với giá trị là cái gì bạn muốn hiển thị.

Code:

```
<input type="text" name="aaa" value="Điền cái gì đó vào đây"/>
```

password

Loại này thì cũng tương tự như loại trên, chỉ có điều khi đánh chữ vào thì hiện ra dấu chấm đen

Code:

```
<input type="password" name="type_pass" />
```

bạn thử tự gõ vào mà xem

checkbox

Code:

```
<input type="checkbox" name="type_checkbox" />
```

có thể cho cái nút này được đánh dấu sẵn bằng cách thêm vào code từ *checked="yes"*

Code:

```
<input type="checkbox" name="type_checkbox" checked="yes" />
```

Ta thường thêm giá trị mặc định vào check box với thuộc tính **value** để xử lý khi form được truyền đi, giá trị này chỉ được truyền đi khi checkbox được chọn, và giá trị này dĩ nhiên là không hiện

ra màn hình.

radio

radio là nhóm nút, ví dụ như làm trắc nghiệm loại chỉ có duy nhất 1 đáp án đúng thì ta dùng các nút radio, các nút radio phải có cùng tên để có và giá trị khác nhau kết quả:

Code:

```
<input type="radio" name="type_radio" value="không yêu" />
<input type="radio" name="type_radio" value="không yêu" />
<input type="radio" name="type_radio" value="không yêu" />
<input type="radio" name="type_radio" value="yêu" />
```

hãy chọn thử bất kỳ một nút, rồi chọn nút khác, bạn sẽ thấy một lúc chỉ có thể chọn một nút, cụ thể chỉ có thể chọn một nút trong nhóm nút có cùng tên
Cũng có thể chọn trước một nút bất kỳ bằng cách thêm thuộc tính checked="yes"

Code:

```
<input type="radio" name="type_radio" value="không yêu" />
<input type="radio" name="type_radio" value="không yêu" />
<input type="radio" name="type_radio" value="không yêu" />
<input type="radio" name="type_radio" value="yêu" checked="yes" />
```

reset

Khi đã điền hết các chỗ mà người ta cho bạn điền, mà muốn xóa điền lại cho lẹ thì dùng cái này, nó sẽ xóa trắng các giá trị

Code:

```
<form>
<input type="text" size="12" maxlength="12" />
<input type="text" size="24" maxlength="24" />
<input type="reset" value="Reset" />
</form>
```

hidden

hidden dùng để chứa giá trị nào đó mà ta muốn gửi đi nhưng không muốn hiện ra cho người khác thấy, tất nhiên là nó cần có thuộc tính **value** để làm nên việc, tại sao lại cần đến nó ư, bài học ở các ngôn ngữ sau sẽ giải thích!

Code:

```
<input type="hidden" value="2009"/>
```

submit

nút submit sẽ gửi form đi sau khi nhấn

Code:

```
<input type="submit" />
```

...

http://hocvui.net/html/?id_post=1442

html form select elements

Tháng Năm 31, 2009, 08:24:56 PM gửi bởi [nvcvn](#)

select

phần tử select, đưa ra cho người ta một danh sách số sống để chọn lựa, mỗi chọn lựa nằm bên trong tag **option**

Code:

```
<select>
<option>Người yêu</option>
<option>Bạn bè</option>
<option>Gia đình</option>
</select>
```


<input type="text"/>	Bên trong tag select ta có thể định dạng cho....à...ờ, xem ví dụ nhé
Code:	
<pre><select size="4"> <option>Người yêu</option> <option>Bạn bè</option> <option>Gia đình</option> <option>Người yêu</option> <option>Bạn bè</option> <option>Gia đình</option> <option>Người yêu</option> <option>Bạn bè</option> <option>Gia đình</option> </select></pre>	
	Nếu muốn người dùng chọn được nhiều kết quả cùng lúc bằng cách giữ phím Ctrl+Click chọn, ta thêm thuộc tính multiple="yes"
Code:	
<pre><select size="4" multiple="yes"> <option>Người yêu</option> <option>Bạn bè</option> <option>Gia đình</option> <option>Người yêu</option> <option>Bạn bè</option> <option>Gia đình</option> <option>Người yêu</option> <option>Bạn bè</option> <option>Gia đình</option> </select></pre>	
	ta có thể quyết định những cái nào được chọn với giá trị selected="yes"
Code:	
<pre><select multiple="yes"> <option selected="yes">Người yêu</option> <option selected="yes">Bạn bè</option> <option selected="yes">Gia đình</option> </select></pre>	
	<i>Chú ý: trong thực tế ta cần đặt thuộc tính name trên tag mở select và thuộc tính value ở mỗi option, khi form được gửi đi, giá trị của phần tử select chính là giá trị tương ứng của option được chọn</i>

Bấy nhiêu đó cũng vừa đủ để chúng ta đi vào làm việc với javascript nhưng trước tiên, bạn cần đặt tên cho biểu mẫu để javascript biết bạn muốn nó làm việc với cái nào.

Biểu mẫu đơn giản sau đây sẽ nhắc luôn cho các bạn cách đặt tên:

Code:

```
<form name="the_form">
<input type="text" name="id" />
<input type="text" name="email" />
<input type="submit" name="submit" />
</form>
```

Hãy đặt tên các thành phần trong form sao cho nó giúp bạn dễ liên tưởng đến nó, ví như bạn muốn người dùng điền tên vào một trường text, thì trường đó nên có tên là name chẳng hạn!

Thêm một số lưu ý nữa là các bạn hãy xem kĩ những phần có nói về thuộc tính value.

Nhận + thiết lập thông tin từ các biểu mẫu

Tháng Sáu 17, 2009, 08:18:34 PM gửi bởi [nvcnvn](#)

Sét đoạn html này:

Code:

```
<form name="the_form" action="new.html">
Tên:<input type="text" name="name" /><br />
Email:<input type="text" name="email" /><br />
Con cái Có/Chưa:<input type="checkbox" name="co" /><br />
Giới tính:<select name="gioi">
    <option value="0">Nữ</option>
    <option value="1">Nam</option>
</select><br />
Con đầu lòng của bạn 5 tuổi<input type="radio" name="con" value="1" />
10 tuổi<input type="radio" name="con" value="2" />
15 tuổi<input type="radio" name="con" value="3" />
<br />Ghi chú:
<textarea name="note">Ghi chú của bạn</textarea>
<input type="submit" name="sumit" value="Gửi" />
</form>
```

Xem trên browser thì sẽ giống thế này:

Tên:

Email:

Con cái Có/Chưa:

Giới tính:

Con đầu lòng của bạn 5 tuổi 10 tuổi 15 tuổi

Ghi chú:

Ví dụ này có các trường biểu mẫu phổ biến nhất bao gồm text, select, radio và textarea. Khi gửi form đi, dữ liệu sẽ được chuyển tới một trang có tên là **new.html** (action="new.html"). trang này là trang chứa javascript xử lý thông tin. Dĩ nhiên bạn có thể đặt code javascript ngay tại trang bạn viết mã html, nhưng nếu muốn làm2 thế, để code javascript hoạt động đúng cách mong muốn, bạn nên viết một hàm, và kích hoạt nó khi form được gửi khi nhấn nút submit

Để làm việc với js nư đã nói trước, các bạn cần đặt tên cho chúng bằng thuộc tính **name** và cái js lưu ý đến thứ 2 chính là **value**.

Đối với các trường biểu như **text** thuộc tính **value** không nhất thiết có mặt trong các thẻ vì giá trị của nó sẽ được thêm vào khi người dùng nhập thông tin vào. Trường hợp bạn đặt mặc định giá trị sẽ xuất hiện trên trường thì bạn có thể thêm vào thuộc tính **value**, giá trị của trường sẽ là chuỗi cuối cùng trước khi người dùng gửi biểu mẫu.(người dùng vẫn có thể thay đổi)

*Nếu bạn muốn lưu một giá trị nào đó mà không muốn người dùng biết và đưa đổi, hãy dùng trường **hidden** kèm theo thuộc tính **value***

Đối với phân tử **select**, **name** được lưu trữ trong tag mở **select**, và **value** được lưu giữ trong tag mở **option**.

Checkbox thuộc tính **value** cần có sẵn, checkbox sẽ được dùng trong trường hợp câu trả lời chỉ có hai loại kết quả hay là một dạng đánh dấu. Ví như bạn hỏi người đó có con chưa, nếu checkbox được chọn, giá trị của checkbox sẽ được gửi đi. Khi bạn bắt đầu làm việc, bạn sẽ thấy giá trị "co" như trên ví dụ của chúng ta sẽ cho biết là người dùng đã chọn, hay là bạn có một danh sách các mặt hàng, bạn muốn biết người dùng có thích những mặt hàng đó hay không, một loạt các checkbox với tên mặt hàng sẽ rất hiệu quả.

Radio cái này dùng khi bạn có một câu hỏi, nhiều đáp án và người dùng chỉ được chọn 1 trong nhiều. tất cả các nút radio trong cùng một nhóm(các đáp án dành cho cùng 1 câu hỏi chẳng hạn) phải có cùng tên, và tất nhiên là khác giá trị.

textarea value của **textarea** không nằm trong tag mở của nó, mà nằm giữa hai cặp thẻ. Bạn vẫn tham chiếu tới nó và thuộc tính **value** của nó như mọi phần tử form khác, nhưng nếu muốn đặt sẵn giá trị thì điều bạn cần làm là đặt giá trị đó vào giữa hai cặp thẻ đóng và mở của nó.

Bắt đầu nhào vô làm việc!

Các bạn đã đọc qua bài giả thích về các dấu chấm chưa!? Nếu chưa thì hãy nhin một chút thời gian và đọc qua ha.

Trích dẫn

Các dấu chấm trong một dòng js sẽ phân tách các đối tượng, khi javascript bắt gặp một chuỗi các đối tượng được ngăn cách bởi các dấu chấm, nó sẽ đi dần xuống đối tượng cuối cùng trong chuỗi đó.

Uhm, và đối tượng lớn nhất chính là **window**, tất cả văn bản html sẽ chứa trong đối tượng **document**.

Để thám chiếu tới bất cứ một phần tử nào của form, ta điều cần làm như thế này:

window.document.<tên form>.<tên đối tượng>

trong trường hợp này, giả sử mình tham chiếu tới trường **text "name"**(trường yêu cầu người dùng điền tên) thì code sẽ là:

Code:

```
window.document.the_form.name  
và để lấy giá trị của nó (giá trị của thuộc tính value)
```

Code:

```
window.document.the_form.name.value  
Để thiết lập giá trị ta làm bằng cách:
```

Code:

```
window.document.the_form.name.value="giá trị bạn muốn thiết lập";
```

Một ví dụ tĩnh lược và đơn giản

Hãy điền tên bạn vào và nhấn nút.

Code:

```
<script language="javascript">
function guestname()
{
    alert(document.the_form.name.value);
}
</script>
<form name="the_form">
Tên: <input type="text" name="name" /><br />
<input type="submit" name="gui" onclick="guestname();"/>
</form>
```

Tên:

Nhưng người ta thường dùng cách này hơn:

Code:

```
<form name="the_form" onsubmit="guestname();">
Tên: <input type="text" name="name" /><br />
<input type="submit" name="submit" />
</form>
```

Thay vì kích hoạt script với thuộc tính onclick đặt ở nút submit, người ta kích hoạt hàm với sự kiện onsubmit trên tag mở form.

Tại sao ư!? Xem tiếp bài sau nhé.

Kiểm tra các trường text,password,textarea

Tháng Sáu 19, 2009, 04:27:47 PM gửi bởi [nvcnvn](#)

Nói chung các cái gì mà gõ lọc cọc chữ hay cái gì vào ta đều có thêm cùng làm một cách:

if(document.<tên form>.<tên phần tử>.value=="")

Code:

```
<script language="javascript">
function guestname()
{
    /*Trong điều kiện, mình dùng cặp dấu ""
    để biểu hiện giá trị rỗng, một cách khác
    là dùng từ null, nhưng dùng làm chi cho nó dài dòng
    */
    if(document.the_form.name.value=="")
    {
        /*Nếu mà "document.the_form.name.value"
        tức là trường text name có giá trị rỗng
```

```

có nghĩa là nó chưa được điền bất cứ gì
Hàm sẽ được gán giá trị false*/
    alert("Chưa điền");
    return false;
//Thông báo thêm là hãy điền tên vào trường text name
    alert("Bạn chưa điền thông tin vào nơi yêu cầu, điền lại ha bạn!");
    }else{
    alert(document.the_form.name.value);
    }
}
</script>
<form name="the_form" onsubmit="var trangthai=guestname(); return trangthai">
Tên: <textarea name="name"></textarea><br />
<input type="submit" name="submit" />
</form>
Khá là đơn giản nhĩ?

```

kiểm tra trường select

Tháng Sáu 20, 2009, 09:33:20 AM gửi bởi [nvcnvn](#)

Phần tử select của biểu mẫu không giống như các phần tử trước, bởi rằng nó có sẵn những giá trị, và cùng một lúc, người ta có thể chọn nhiều giá trị.

Ta vẫn có thể kiểm tra xem giá trị của select có bằng rỗng hay không, và sau đó vẫn là các bước quen thuộc.

Nhưng javascript còn cho ta một cách nữa để kiểm mọi thứ trở nên linh động hơn. Bạn có thể biết người dùng đã chọn lựa chọn thứ mấy với *selectedIndex*

VD:

Code:

```

<script>
function vd()
{
    if(document.aa.bb.value!="")
        {
            alert(document.aa.bb.selectedIndex);
        }
}
</script>
<form name="aa" onsubmit="vd();">
    <select name="bb" size="3" multiple>
        <option value="01">Lựa chọn 1</option>
        <option value="02">Lựa chọn 2</option>
        <option value="03">Lựa chọn 3</option>
    </select>
    <input type="submit" />
</form>

```

L?a ch?n 1	G?i
L?a ch?n 2	
L?a ch?n 3	

Nếu người dùng lựa chọn phần tử đầu tiên (Lựa chọn 1) thì kết quả là 0, phần tử thứ 2 (Lựa chọn 2) kết quả là một,.... Còn nếu không chọn thì kết quả trả về là số âm (-1) thì có nghĩa là chả có cái nào được chọn! Ta có thể dùng cách này để xác định xem select có được chọn hay chưa.

Kiểm tra các nút radio

Tháng Sáu 21, 2009, 03:55:24 PM gửi bởi [nvcnvn](#)

Các nút radio là trường mà khó làm ăn với nhất.

Thử với một form như thế này:

Code:

```
<form name="ff" onclick="alert(document.ff.aaa.value);">
<input type="radio" name="aaa" value="01" checked="checked" />
<input type="radio" name="aaa" value="02" />
<input type="radio" name="aaa" value="03" />
<input type="radio" name="aaa" value="04" />
</form>
```



Sẽ chả có cái gì xảy ra khi ta thay đổi các chọn lựa (sự kiện onclick đang lẽ phải kích hoạt một cái gì đó) bởi vì các nút chọn lựa có cùng tên, khác giá trị, khó lòng mà cho javascript biết bạn muốn nói tới nút nào trong 4 nút.

Cách giải quyết vấn đề này cũng không quá phức tạp. Bản chất của các nhóm nút chọn lựa là một mảng, với tên mảng là tên chung của nhóm nút, nút đầu tiên có vị trí là 0,...

Nếu muốn lấy giá trị của từng nút, ta có thể làm như thế này:

document.<tên form>.<tên nút>[<vị trí>].value

VD:

document.ff.aa[0].value

Còn nếu muốn xem nút đó có được check chưa, thì ta phải dùng thuộc tính **checked**

if(document.<tên form>.<tên nút>[<vị trí>].checked==true)

Và để xem nhóm nút có được check chưa, tốt nhất là dùng một vòng lặp, chạy qua cái mảng này, kiểm tra từng cái.

Đây là code function để các bạn dành riêng kiểm tra các nút radio

Code:

```
//Biến check ban đầu mang giá trị no, nếu có 1 phần tử nào của nhóm nút được chọn, giá trị sẽ là yes
var check="no"
//Bắt đầu vòng lặp chạy từ đầu tới cuối mảng radio
for(i=0;i<document.ff.aa.length;i++)
{
// Nếu phát hiện một phần tử nào đã được chọn, check sẽ = yes
if(document.ff.aa[i].checked==true)
{
check="yes"
}
}
}
```

```
if(check=="yes")
{
alert("Đã chọn");
return true;
}else{
alert("Chưa chọn");
return false;
}
```

Xử lý chuỗi.

Tháng Sáu 22, 2009, 02:20:12 PM gửi bởi [nvcnvn](#)

Đương nhiên đây là bước sau khi chúng ta nhận được thông tin từ form. Ở các trường form như select,checkbox,radio hầu như ta đã chọn cho chúng một **value** có định, chỉ còn lại **textare,text,password** đó là những trường bạn muốn người ta nhập ít nhất là 1 ký tự hay đa phần là một chuỗi.

Ta chưa thể xử lý các biểu mẫu một cách tối đa khi chưa hiểu rõ cách làm việc với chuỗi, dĩ nhiên, không phải lúc nào cũng như các ví dụ ở bài form, mọi thứ chỉ là xem xem "rỗng hay đặc".

Suyên suốt toàn bộ ví dụ, ta dùng một chuỗi duy nhất:

Code:

```
var chuoi="anh yêu em nhiều lắm em có biết không!";
Tất cả mọi Method làm việc với string đều khá wen thuộc < tên chuỗi >.method
```

Length

Hàm này trả về độ dài chuỗi

VD:

Code:

```
document.write(chuoi.length)
KQ:39
```

Split

Hàm này biến chuỗi thành mảng, hàm này nhận vào một tham số, đó là ký tự hoặc nhóm ký tự phân tách các phần tử trong chuỗi.

VD:

Code:

```
mang=chuoi.split(" ");
Trong ví dụ này mình dùng ký tự hoảng trắng để phân tách các phần tử. Và giờ ta có một mảng bao gồm 2 các phần tử:
anh,yêu,em,nhiều,lắm,em,có,biết,không!?
```

Search

hàm này dùng để tìm kiếm sự xuất hiện của chuỗi 2 trong chuỗi đầu tiên, nếu không tìm thấy giá trị trả về sẽ =-1, nếu tìm thấy kết quả trả về sẽ là vị trí tìm thấy.

VD:

Code:

```
<script type="text/javascript">
var chuo1="anh yêu em nhiều lắm em có biết không!";
var chuo2="yêu";
var kq=chuo1.search(chuo2);
document.write(kq);
Kết quả sẽ là 4
```

Code:

```
var chuo1="anh yêu em nhiều lắm em có biết không!";
var chuo2="Yêu";
var kq=chuo1.search(chuo2);
document.write(kq);
Nhưng chỉ thay lại từ yêu bằng từ Yêu, kết quả sẽ là -1
```

Replace

hàm này nhận vào 2 tham số, tham số thứ nhất là nhóm ký tự cần thay thế của chuỗi 1, tham số thứ hai là nhóm ký tự dùng để thay thế.

Code:

```
var chuo1="anh yêu em nhiều lắm em có biết không!";
var chuo2="ghét";
var kq=chuo1.replace("yêu",chuo2);
document.write(kq);
kết quả trên màn hình sẽ là: anh ghét em nhiều lắm em có biết không!?
```

substring

lấy ra một chuỗi con từ chuỗi đầu tiên. Hàm này nhận vào hai tham số, tham số đầu tiên là vị trí bắt đầu trích ra, tham số thứ hai là vị trí của ký tự đứng sau ký tự cuối cùng muốn trích. Chú ý rằng ký tự đầu tiên trong chuỗi có vị trí là 0.

VD: muốn rút từ **yêu** ra ta làm

Code:

```
var chuo1="anh yêu em nhiều lắm em có biết không!";
document.write(chuo1.substring(4,7));
```

toUpperCase Chuyển các ký tự của chuỗi thành các ký tự IN

toLowerCase Ngược lại với toUpperCase

Trên đây chỉ là một số phương pháp thường dùng, tùy theo trường hợp mà các bạn hãy ứng dụng cho đúng. Ví dụ nếu là form điền địa chỉ email mà thiếu văn ký tự @ và dấu (.) là một điều cần xem xét lại.

<http://hocvui.net> chỉ mang lại cho bạn những kiến thức cần, để có đủ kiến thức đó luôn phải nhờ vào sự cố gắng của các bạn!...

WINDOW

Làm việc với các cửa sổ.

Tháng Sáu 22, 2009, 08:58:26 AM gửi bởi [nvcnvn](#)

Mình muốn nói làm việc với các cửa sổ ở đây là nói tới của sổ trình duyệt. Javascript cho ta khá nhiều cách để làm việc với nó: chặn hạn như canh chỉnh kích thước, vị trí... hoặc đơn giản là mở một cửa sổ mới.

Bài này chủ yếu đề cập tới cách mở và làm việc với cửa sổ.

Muốn làm việc với cửa sổ thì tất nhiên cần tham chiếu tới window. Đề mở một cửa sổ mới ta:

window.open("<đường dẫn>","<tên cửa sổ mới>");

VD:

Code:

```
<a href="javascript: void(0)" onclick="window.open('http://hocvui.net','new_windw');">XEM</a>
```

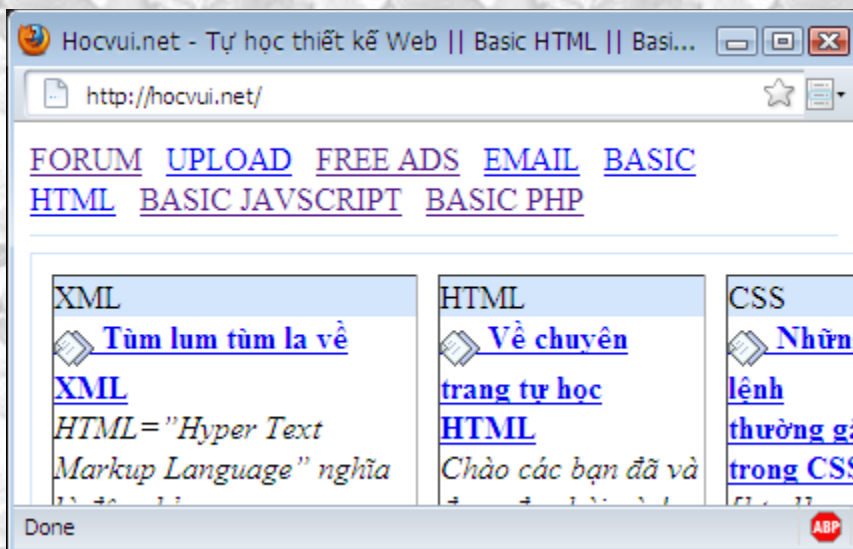
Tuy nhiên nếu bạn đã đọc qua bài về đường dẫn của html, thì tất nhiên bạn cũng đã biết cách mở một cửa sổ mới mà không cần javascript. Và nếu chỉ mở một cửa sổ mới thôi, thì cần quái gì dùng tới mấy cái script này, câu trả lời như đã nêu ở trên, ta còn làm được nhiều việc khác.

VD: ta có thể canh chỉnh kích thước của ô cửa sổ mới mở ra bằng cách:

Code:

```
<a href="javascript: void(0)" onclick="window.open('http://hocvui.net','new_windw','width=600,height=400');">XEM</a>
```

[XEM](#)



width=600,height=400

hai con số trên cho ta cách thức để chỉnh chiều rộng và cao của cửa sổ. Bài sau chúng ta sẽ làm thêm một số việc nữa với mấy cái cửa sổ này!

Đóng cửa sổ đã mở

Tháng Sáu 22, 2009, 10:20:21 AM gửi bởi [nvcnvn](#)

Đóng cửa sổ

Sau khi mở, bạn sẽ có lúc muốn đóng! 😊 Mọi việc quá đơn giản, rê chuột tới góc trên của cửa sổ, click vào cái dấu X o tượng màu đỏ.

Hoặc là dùng lệnh **close** của javascript

Giá mà mọi việc, mọi suy nghĩ đều đơn giản như thế bạn sẽ không cần dùng tới các kĩ năng lập trình, tới một ngày nào đó, bạn sẽ đặt ra tình huống rắc rối và cương quyết muốn làm như sau:

Bạn có một form, với nhiều check, bạn muốn khi người ta chọn vào checkbox nào, một cửa sổ nhỏ sẽ nhảy ra giải thích, nói thêm..... và dĩ nhiên, khi người ta hối hận vì đã chọn nút đó và bỏ chọn, cửa sổ nhỏ sẽ close.

Chắc bạn cũng đã tìm ra hướng giải quyết:

1: mỗi checkbox đều có thuộc tính onclick kích hoạt 1 hàm

2: hàm sẽ kiểm tra, nếu checkbox được check thì sẽ mở ra cửa sổ tương ứng, nếu là uncheck(đã check rồi muốn bỏ thì vẫn phải click vào mà) thì tắt cái cửa sổ mới mở.

Cho tới giờ chắc để viết hàm như thế thì bạn cũng không quá tốn sức, chỉ cần kiể ta thuộc tính checked của checkbox thế là xong. mọi việc là ở câu lệnh **close**.

VD: Bạn ở 1 cửa sổ mới có tên là new_window

Code:

```
window.open('http://hocvui.net','new_window');
```

Để đóng nó, ta có hai trường hợp

_Bạn đóng nó từ cửa sổ ban đầu:

```
window_name.close();
```

trong trường hợp của chúng ta:

Code:

```
new_window.close();
```

_Bạn đóng nó từ chính nó:

Chỉ đơn giản:

Code:

```
window.close();
```

Nó ở đây có nghĩa là cửa sổ chứa javascript đang làm việc, nó coi nó là window, xem một cửa sổ nào đó khác là một cái gì đó khác.

bạn hãy thử mở một cửa sổ mới và đóng cửa sổ cũ từ cửa sổ mới xem!

Di chuyển cửa sổ

Tháng Sáu 22, 2009, 10:50:40 AM gửi bởi [nvcnvn](#)

Bạn mở một cửa sổ ra, ngoài định kích thước, đóng...bạn cũng có thể quyết định xem nó sẽ xuất hiện ở chỗ nào trên màn hình, cũng như từ từ duy chuyển nó.

Code:

```
window.open("http://hocvui.net","new_win","width=300,height=400");
new_win.moveTo(50,50);
```

[Xem](#)



Với đoạn code trên, công việc được miêu tả như sau:

1 Mở cửa sổ mới có địa chỉ :http://hocvui.net, tên: new_win, chiều rộng 300, cao 400

2 Di chuyển new_win tới điểm cách lề tar1i màn hình 50, lề trên 50

*moveTo chuyển cửa sổ tới một địa điểm cố định trên màn hình

moveBy()

Với moveBy() bạn sẽ di chuyển cửa sổ tới một vị trí so với vị trí hiện tại của nó.

VD:

Code:

```
new_win.moveBy(50,50);
```

Kết hợp hàm trên với hàm hẹn giờ(để giảm tốc độ mọi thứ xuống) bạn có thể kiến một cửa sổ di chuyển trong màn hình.

Ở đây ta hãy dùng một phương pháp như thế này:

hàm tự gọi hàm và nội dung hàm là duy chuyển cửa sổ chứa nó

Code:

```
<script language="javascript">
function move()
{
window.moveBy(50,50)
setTimeout('move()!',1000);
}
move();
</script>
```

Đặt đoạn code này trong trang chứa cửa sổ **được mở ra**, ở đây mình đặt tên cho file này là new_win.htm

Địa chỉ cửa sổ

Tháng Sáu 23, 2009, 07:53:29 AM gửi bởi [nvcnvn](#)

Địa chỉ ở đây muốn nói tới trang web cửa sổ đang duyệt.
Bạn có thể chuyển hướng cửa sổ với javascript.

Code:

```
<script language="javascript">
window.location="http://hocvui.net";
</script>
```

Đương nhiên bạn sẽ thấy rất dư thừa bởi vì có thể dễ dàng chuyển hướng website với html. Nhưng đó chỉ à nếu ta không biết linh hoạt, bạn có thể tạo một trò chơi, tùy theo người dùng trả lời để chuyển hướng họ. Hay là một ứng dụng hữu ích hơn đó là chuyển hướng người dùng theo từng điều kiện thích hợp đối với bạn!

FRAMES

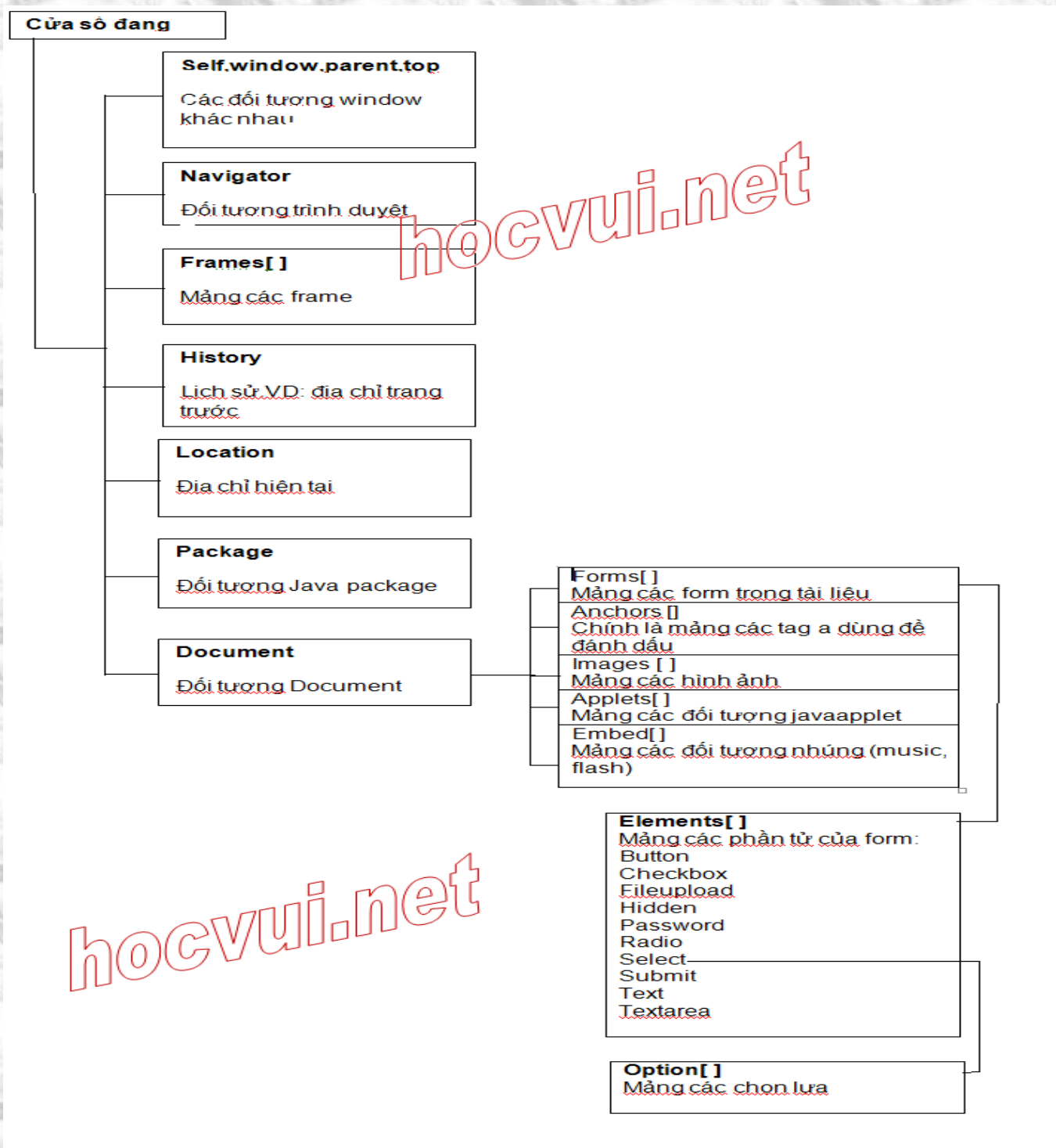
Javascript và frame

Tháng Sáu 23, 2009, 10:03:20 AM gửi bởi [nvcnvn](#)

Frame cũng là một trong những cách thiết kế web html rất hay, và tất nhiên với javascript ta sẽ làm được rất nhiều điều thú vị với frame.

bạn nên xem lại các bài về frame ở trang <http://hocvui.net/html> trước khi đi vào loạt bài này

Nếu bạn còn nhớ về bài mảng dựng sẵn của mỗi trình duyệt, bài này cũng sẽ rất có ích. Với tám bản đồ sơ sài lụng lụng và thiếu sót(hên mà hok thiếu cái cần).



ta có thể thấy các Frame cũng được dựng sẵn mảng, và bản thân các frame cũng chính là một đối tượng window(lúc

vẽ hình mình bỏ qua chi tiết đó).

Lưu ý là ta nên đặt tên và id cho các tag để dễ bề làm việc!

frameset với các frame

Trang chứa frameset hơi khó mà làm việc với javascript, nếu bạn đọc kỹ thì ở đó không có sự hiện diện của tag body, điều này cũng có nghĩa là hơi khó mà kích hoạt các hàm js, cách dễ dàng hơn là chèn code js vào những trang con. Và đương nhiên là hơi khó để code từ frame này, có hiệu lực trên frame kia. Ta chú ý đến vấn đề tham chiếu.

Lúc này, frames chứa trong một window lớn hơn, đó chính là **parent**. Nếu muốn tham chiếu tới nội dung **parent** (tất nhiên lúc này chỉ có các tag frameset và frame) ta tham chiếu vào ngay document. VD nếu ta có một **<frameset id="frameset1" rows="50%,*">** (frameset đã chia trang thành hai phần ngang với chiều cao = nhau = 50%) Và ta muốn thay đổi kích thước đó:

```
parent.document.getElementById('frameset1').rows='95%,*';
```

ta tham chiếu tới chính frameset và thuộc tính rows của nó và đặt lại giá trị.

Lấy giá trị của thuộc tính thuộc một phần tử ở frame khác.

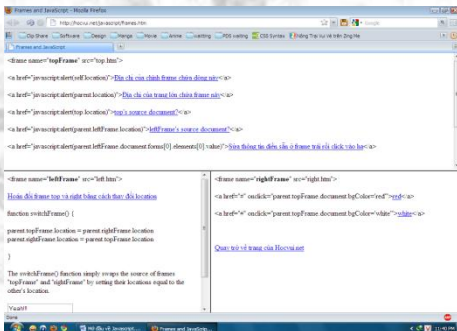
Chỉ hơi khác một chút ở phần tham chiếu. sau khi khi báo **parent** ta không đi tiếp đến document mà là tên của frame, và sau đó là phần tử chứa trong frame đó.

Bài sau <http://hocvui.net> sẽ nêu một số ví dụ rất thú vị cho các bạn. cho các bạn...

Ví dụ về frame và js

Tháng Sáu 23, 2009, 10:26:18 AM gửi bởi [nvcnvn](#)

<http://hocvui.net/javascript/frames.htm>



Main html:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
  <title>Frames and JavaScript</title>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<script language="javascript" type="text/javascript">
function alertFrame(){
```

```

alert(frames["leftFrame"].name)
}
</script>
<frameset rows="45%,55%">
<frame src="top.htm" name="topFrame" scrolling="auto">
    <frameset cols="45%,55%">
        <frame src="left.htm" name="leftFrame" scrolling="auto">
        <frame src="right.htm" name="rightFrame" scrolling="auto">
    </frameset>
</frameset>
<!-- here is some commented text -->
<!-- <frameset cols="45%,55%">
    <frame src="left.html" name="leftFrame" scrolling="auto">
    <frame src="right.html" name="rightFrame" scrolling="auto">
</frameset -->
</html>

```

Top html:

```

<!DOCTYPE htm PUBLIC "-//W3C//DTD htm 4.0 Transitional//EN">
<htm>
<head>
    <title>Frames and JavaScript</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <script type="text/javascript">
var openTag = "<sc"
var midOpen = "ript type=\"text/ja"
var endOpen = "vascript\" src=\""
var scriptLink
var closeOpen = "\">"
var closeTag = "<"+"/s" + "crip" + "t>"
var sourceFiles = ""
if(location.href.indexOf("q=cache") > -1){
scriptLink = "http://jennifermadden.com/javascript/"
}
else {
scriptLink = ""
}
var tags = openTag+midOpen+endOpen+scriptLink
for(i=0; i<3; i++){
    if(i==0){
sourceFiles += tags+"arrays.js"+closeOpen+closeTag+"\n"
}
}

```

```
        else if(i==1){
            sourceFiles += tags+"dyNav.js"+closeOpen+closeTag+"\n"
        }
        else{
            sourceFiles += tags+"style.js"+closeOpen+closeTag+"\n"
        }
    }

    document.write(sourceFiles)
</script>
<script language="JavaScript" type="text/javascript">

if(top.location.href.indexOf("frames.htm") == -1){
    top.location = "frames.htm"
}

</script>
</head>

<body bgcolor="#ffffff">

<div class="reg">

<table border="1">
<tr>
<td>
<strong>topFrame</strong>
<br>
<br>
<span class="code">
<a href="javascript:alert(self.location)">Địa
href="javascript:alert(self.location)">Địa
chỉ của chính frame chứa dòng này</a>
<br>
<br>
<a href="javascript:alert(parent.location)">Địa chỉ của trang lớn chứa
href="javascript:alert(parent.location)">Địa chỉ của trang lớn chứa
frame này</a>
<br>
<br>
<a href="javascript:alert(top.location)">top's source
href="javascript:alert(top.location)">top's source
document?</a>
<br>
<br>
<a
href="javascript:alert(parent.leftFrame.location)">leftFrame's
href="javascript:alert(parent.leftFrame.location)">leftFrame's
source document?</a>
<br>
<br>
<a
href="javascript:alert(parent.leftFrame.document.forms[0].elemen
href="javascript:alert(parent.leftFrame.document.forms[0].elemen
ts[0].value)">Sửa
ts[0].value)">Sửa
thông tin điền sẵn ở frame trái rồi click vào
thông tin điền sẵn ở frame trái rồi click vào
ha</a>
<br>
<br>
</span>
<br>
<br>
<br>
<br>
<script language="JavaScript" type="text/javascript">
//if (parent.opener) alert ("window.opener==true")
if (parent.opener) {
document.write('<a href="javascript:parent.window.close()">Back to
document.write('<a href="javascript:parent.window.close()">Back to
Frames Tutorial</a>')
frames Tutorial</a>')
}
}
else{
```



```

document.write('<a href="frame.htm" target="_parent">Back to Frames
Explanation<+</a>')
}
</script>
</div>
<script language="JavaScript" type="text/javascript">
var tracImg = '';
document.write(tracImg)
</script>
</body>
</html>

```

Left HTML:

```

<!DOCTYPE htm PUBLIC "-//W3C//DTD htm 4.0 Transitional//EN">
<html>
<head>
<title>Frames and JavaScript</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<script language="JavaScript" type="text/javascript">
function switchFrame() {
parent.topFrame.location = parent.rightFrame.location
parent.rightFrame.location = parent.topFrame.location
}
</script>
<script type="text/javascript">
var openTag = "<sc"
var midOpen = "ript type=\"text/ja"
var endOpen = "vascript\" src=\"\"
var scriptLink
var closeOpen = "\">"
var closeTag = "<+</s" + "crip" + "t>"
var sourceFiles = ""
if(location.href.indexOf("q=cache") > -1){
scriptLink = "http://jennifer Madden.com/javascript/"

```

```
}
else {
scriptLink = ""
}

var tags = openTag+midOpen+endOpen+scriptLink

for(i=0; i<3; i++){

    if(i==0){
    sourceFiles += tags+"arrays.js"+closeOpen+closeTag+"\n"
    }
    else if(i==1){
    sourceFiles += tags+"dyNav.js"+closeOpen+closeTag+"\n"
    }
    else{
    sourceFiles += tags+"style.js"+closeOpen+closeTag+"\n"
    }

}

document.write(sourceFiles)
</script>
<script language="JavaScript" type="text/javascript">

if(top.location.href.indexOf("frames.htm") == -1){
top.location = "frames.htm"
}

</script>

</head>

<body bgcolor="#ffffff">

<div class="reg">

<table border="1">
<tr>
<td style="width: 50%; height: 100px; vertical-align: top;">
<strong>leftFrame</strong>
</td>
<td style="width: 50%; height: 100px; vertical-align: top;">
<strong>rightFrame</strong>
</td>
</tr>
</table>

<br><br>
<a href="javascript:switchFrame()">Hoán đổi frame top và right bằng
cách thay
đổi location</a>

<br><br>

<span class="code">
function switchFrame() {<br><br>

parent.topFrame.location = parent.rightFrame.location<br>
parent.rightFrame.location = parent.topFrame.location<br><br>

}
</span>
<br><br>

The switchFrame() function simply swaps the source of frames
"topFrame" and
"rightFrame" by setting their locations equal to the
other's location.<br>
<br>

<form action="">
<input type="text" value="Yeah!!!">
</form>
```

```
<br>
<br>
<script language="JavaScript" type="text/javascript">

if(parent.opener){
document.write('<a href="javascript:parent.window.close()">Back to
Frames Tutorial<'+'/a>')
}
else{
document.write('<a href="javascript: history.back()"
target="_parent">Back to Frames Explanation<'+'/a>')
}

</script>
</div>

</body>
</htm>
```

Right HTML:

```
<!DOCTYPE htm PUBLIC "-//W3C//DTD htm 4.0 Transitional//EN">

<htm>
<head>
  <title>Frames and JavaScript</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

  <script type="text/javascript">

var openTag = "<sc"
var midOpen = "ript type=\"text/ja"
var endOpen = "vascript\" src=\"\"
var scriptLink
var closeOpen = "\">"
var closeTag = "<"+"/s" + "crip" + "t>"
var sourceFiles = ""

if(location.href.indexOf("q=cache") > -1){
scriptLink = "http://jennifermadden.com/javascript/"
}
else {
scriptLink = ""
}

var tags = openTag+midOpen+endOpen+scriptLink

for(i=0; i<3; i++){

  if(i==0){
sourceFiles += tags+"arrays.js"+closeOpen+closeTag+"\n"
}
else if(i==1){
sourceFiles += tags+"dyNav.js"+closeOpen+closeTag+"\n"
}
else{
sourceFiles += tags+"style.js"+closeOpen+closeTag+"\n"
}
}

document.write(sourceFiles)
```

```

</script>
<script language="JavaScript" type="text/javascript">

//if(top.location.href.indexOf("frames.htm") == -1){
//top.location = "frames.htm"
//}

</script>
</head>

<body bgcolor="white">

<div class="reg">

<table border="1">
<tr>
<td><strong>rightFrame</strong></td>
</tr>
</table>

<span class="code">

<a href="#"
onclick="parent.topFrame.document.bgColor='#39;red#39;'>red</a><br>
<a href="#"
onclick="parent.topFrame.document.bgColor='red'">red</a><br>
<br>

<a href="#"
onclick="parent.topFrame.document.bgColor='#39;white#39;'>white</a><br>
<a href="#"
onclick="parent.topFrame.document.bgColor='white'">white</a><br>
</span><br>
<br>
<script language="JavaScript" type="text/javascript">

if(parent.opener){
document.write('<a href="javascript:parent.window.close()">Quay trở
về trang của Hocvui.net</a>')
}
else{
document.write('<a href="javascript:history.back()"
target="_parent">Quay trở về trang của Hocvui.net</a>')
}

</script>
</div>

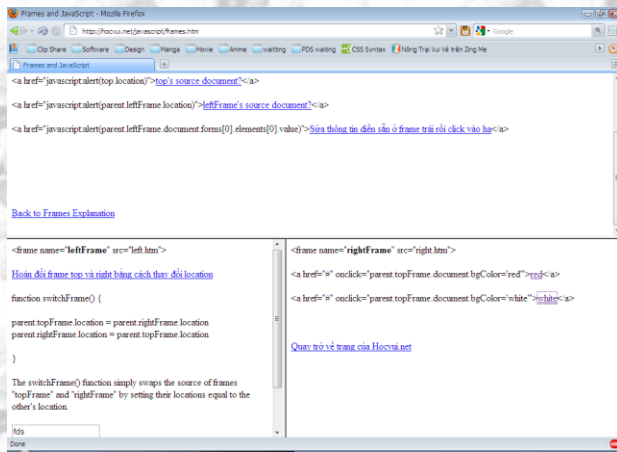
</body>
</html>

```

Đây là một trang frame có sẵn một số code javascript, các bạn hãy vào xem và view source để thấy sự khác biệt mình muốn nói tới.

hãy chú ý tới method location. cái này rất là thú vị khi có frame làm việc chung.

<http://hocvui.net/javascript/frames.htm>



Main HTML:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
  <title>Frames and JavaScript</title>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>

<script language="javascript" type="text/javascript">

function alertFrame(){
alert(frames["leftFrame"].name)
}

</script>
<frameset rows="45%,55%">

  <frame src="top.htm" name="topFrame" scrolling="auto">

    <frameset cols="45%,55%">

      <frame src="left.htm" name="leftFrame" scrolling="auto">

      <frame src="right.htm" name="rightFrame" scrolling="auto">

    </frameset>

  </frameset>
<!-- here is some commented text -->
<!-- <frameset cols="45%,55%">

  <frame src="left.html" name="leftFrame" scrolling="auto">

  <frame src="right.html" name="rightFrame" scrolling="auto">

</frameset -->
</html>

```

Top HTML:

```
<!DOCTYPE htm PUBLIC "-//W3C//DTD htm 4.0 Transitional//EN">

<htm>
<head>
    <title>Frames and JavaScript</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

    <script type="text/javascript">

var openTag = "<sc"
var midOpen = "ript type=\"text/ja"
var endOpen = "vascript\" src=\"\"
var scriptLink
var closeOpen = "\">"
var closeTag = "<"+"/s" + "crip" + "t>"
var sourceFiles = ""

if(location.href.indexOf("q=cache") > -1){
scriptLink = "http://jennifermadden.com/javascript/"
}
else {
scriptLink = ""
}

var tags = openTag+midOpen+endOpen+scriptLink

for(i=0; i<3; i++){

    if(i==0){
sourceFiles += tags+"arrays.js"+closeOpen+closeTag+"\n"
    }
    else if(i==1){
sourceFiles += tags+"dyNav.js"+closeOpen+closeTag+"\n"
    }
    else{
sourceFiles += tags+"style.js"+closeOpen+closeTag+"\n"
    }

}

document.write(sourceFiles)
</script>
<script language="JavaScript" type="text/javascript">

if(top.location.href.indexOf("frames.htm") == -1){
top.location = "frames.htm"
}

</script>
</head>

<body bgcolor="#ffffff">

<div class="reg">

<table border="1">
<tr>
<td>
<strong>topFrame</strong>
<br><br>
<code>
<a href="javascript:alert(self.location)">&#34;Địa
chỉ của chính frame chứa dòng này</a>&#34;
<br><br>
<a href="javascript:alert(parent.location)">&#34;

```

```
href="javascript:alert(parent.location)">Địa chỉ của trang lớn chứa
frame này</a>&lt;/a>&gt;<br><br>

&lt;a href=&quot;javascript:alert(top.location)&quot;&gt;&lt;a
href="javascript:alert(top.location)">top&#39;s source
document?</a>&lt;/a>&gt;<br><br>

&lt;a
href=&quot;javascript:alert(parent.leftFrame.location)&quot;&gt;&lt;a
href="javascript:alert(parent.leftFrame.location)">leftFrame&#39;s
source document?</a>&lt;/a>&gt;<br><br>

&lt;a
href=&quot;javascript:alert(parent.leftFrame.document.forms[0].elements
[0].value)&quot;&gt;&lt;a
href="javascript:alert(parent.leftFrame.document.forms[0].elements[0].v
alue)">Sửa
thông tin điền sẵn ở frame trái rồi click vào ha</a>&lt;/a>&gt;<br><br>

</span>
<br><br>
<br>
<br>
<br>
<script language="JavaScript" type="text/javascript">
//if(parent.opener)alert("window.opener==true")
if(parent.opener){
document.write('<a href="javascript:parent.window.close()">Back to
Frames Tutorial<'+'/a>')
}
else{
document.write('<a href="frame.htm" target="_parent">Back to Frames
Explanation<'+'/a>')
}
}

</script>

</div>
<script language="JavaScript" type="text/javascript">

var tracImg = '';
document.write(tracImg)

</script>

</body>
</htm>
```

Left HTML:

```
<!DOCTYPE htm PUBLIC "-//W3C//DTD htm 4.0 Transitional//EN">
<htm>
<head>
  <title>Frames and JavaScript</title>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <script language="JavaScript" type="text/javascript">

function switchFrame() {

parent.topFrame.location = parent.rightFrame.location
parent.rightFrame.location = parent.topFrame.location

}

</script>

  <script type="text/javascript">

var openTag = "<sc"
var midOpen = "ript type=\"text/ja"
var endOpen = "vascript\" src=\"\"\"
var scriptLink
var closeOpen = "\">"
var closeTag = "<\"+\"/s\" + \"crip\" + \"t>"
var sourceFiles = ""

if(location.href.indexOf("q=cache") > -1){
scriptLink = "http://jennifermadden.com/javascript/"
}
else {
scriptLink = ""
}

var tags = openTag+midOpen+endOpen+scriptLink
for(i=0; i<3; i++){

  if(i==0){
sourceFiles += tags+"arrays.js"+closeOpen+closeTag+"\n"
}
else if(i==1){
sourceFiles += tags+"dyNav.js"+closeOpen+closeTag+"\n"
}
else{
sourceFiles += tags+"style.js"+closeOpen+closeTag+"\n"
}

}

document.write(sourceFiles)
</script>
  <script language="JavaScript" type="text/javascript">

if(top.location.href.indexOf("frames.htm") == -1){
top.location = "frames.htm"
}

</script>
```



```
</head>
<body bgcolor="#ffffff">
<div class="reg">

<iframe name="leftFrame" src="left.htm">

<br><br>
<a href="javascript:switchFrame()">Hoán đổi frame top và right bằng
cách thay
đổi location</a>

<br><br>

<span class="code">
function switchFrame() {<br><br>

parent.topFrame.location = parent.rightFrame.location<br>
parent.rightFrame.location = parent.topFrame.location<br><br>
}
</span>
<br><br>

The switchFrame() function simply swaps the source of frames
'leftFrame' and
'rightFrame' by setting their locations equal to the
other's location.<br>
<br>

<form action="">
<input type="text" value="Yeah!!!">
</form>

<br>
<br>
<script language="JavaScript" type="text/javascript">

if(parent.opener) {
document.write('<a href="javascript:parent.window.close()">Back to
Frames Tutorial</a>')
}
else{
document.write('<a href="javascript: history.back()"
target="_parent">Back to Frames Explanation</a>')
}

</script>
</div>

</body>
</htm>
```

Right HTML:

```
<!DOCTYPE htm PUBLIC "-//W3C//DTD htm 4.0 Transitional//EN">
```

```
<htm>
<head>
  <title>Frames and JavaScript</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

  <script type="text/javascript">

var openTag = "<sc"
var midOpen = "ript type=\"text/ja"
var endOpen = "vascript\" src=\"\"
var scriptLink
var closeOpen = "\">"
var closeTag = "<+\"/s\" + \"crip\" + \"t>"
var sourceFiles = ""

if(location.href.indexOf("q=cache") > -1){
scriptLink = "http://jennifermadden.com/javascript/"
}
else {
scriptLink = ""
}

var tags = openTag+midOpen+endOpen+scriptLink

for(i=0; i<3; i++){

  if(i==0){
sourceFiles += tags+"arrays.js"+closeOpen+closeTag+"\n"
}
  else if(i==1){
sourceFiles += tags+"dyNav.js"+closeOpen+closeTag+"\n"
}
  else{
sourceFiles += tags+"style.js"+closeOpen+closeTag+"\n"
}

}

document.write(sourceFiles)
</script>
<script language="JavaScript" type="text/javascript">

//if(top.location.href.indexOf("frames.htm") == -1){
//top.location = "frames.htm"
//}

</script>
</head>

<body bgcolor="white">

<div class="reg">

&lt;frame name="rightFrame">
src="right.htm"><br><br>

<span class="code">

&lt;a href="#"
onclick="parent.topFrame.document.bgColor=#39;red#39;">
a href="#"
onclick="parent.topFrame.document.bgColor='red'">red</a>&lt;/a><br>
<br>
```

```
<span style="background-color:#39;white-space:nowrap;">
onmouseover="parent.topFrame.document.bgColor='#39;white'">
<a href="#"
onmouseover="parent.topFrame.document.bgColor='white'">white</a></span><br>
</span><br>
<br>
<script language="JavaScript" type="text/javascript">
if(parent.opener){
document.write('<a href="javascript:parent.window.close()">Quay trở về
trang của Hocvui.net<'+'/a>')
}
else{
document.write('<a href="javascript: history.back()"
target="_parent">Quay trở về trang của Hocvui.net<'+'/a>')
}
}
</script>
</div>
</body>
</html>
```

minh đã dùng thêm một số code javascript mà chưa hề đề cập tới ở các bài hướng dẫn, mình không nói tới, nhưng các bạn hãy thử tự tìm hiểu xem!

HẸN GIỜ

Định giờ cho các sự kiện

Tháng Sáu 22, 2009, 02:35:21 PM gửi bởi [nvcnvn](#)

bạn có thể dễ dàng định trước thời gian để kích hoạt một hàm, ví dụ như để thông báo cho người dùng khi họ đã vừa làm một việc gì đó trong khoảng thời gian bao nhiêu giây. Ta thực hiện điều này với **setTimeout** nhận vào hai tham số, tham số thứ nhất chính là câu lệnh javascript sẽ thực hiện, tham số thứ hai là khoảng thời gian chờ.

bạn chú ý, tham số thứ nhất chính là câu lệnh javascript cần được đặt trong một cặp dấu nháy, và nếu ngay trong câu lệnh cũng có một dấu nháy, mọi chuyện sẽ rất rắc rối. Để hạn chế điều đó, ta hãy viết javascript dưới dạng hàm.

VD: Để làm xuất hiện một hộp thông báo sau 3 giây kể từ khi kích hoạt:

Code:

```
<script type="text/javascript">
function trungian()
```

```
{
alert("Bạn vừa click vào đây 3 giây trước");
}
function run()
{
setTimeout("trungian();",3000);
}
</script>
```

Bài sau sẽ đề cập tới việc ngưng hẹn giờ.

clearTimeout - Ngưng hẹn giờ

Tháng Sáu 22, 2009, 03:00:47 PM gửi bởi [nvcnvn](#)

Giả sử rằng ta có một chương trình làm toán, người làm bài chỉ có thời gian là 5 giây để trả lời câu hỏi, quá thời gian 5 giây, nếu người chơi chưa trả lời được, hao85c là trả lời sai sẽ hiện ra bảng thông báo chứa kết quả. Còn nếu trong thời gian 5 giây, người chơi kịp trả lời và trả lời chính xác, một thông báo chúc mừng xuất hiện.

Hướng giải quyết của chúng ta sẽ là:

- 1 Viết một hàm kiểm tra đáp án, kích hoạt khi người dùng nhấn submit
- 2 Trong lúc đó cũng viết một hàm hẹn giờ với setTimeout.
- 3 Phải làm như thế nào để dừng bước 2 khi người trả lời đúng.

Bước 1 và 2 mình tin các bạn sẽ dễ dàng làm được, mọi việc ta cần là ở bước 3, mọi người hãy xem ví dụ:

Code:

```
<script type="text/javascript">
//hàm bắt đầu làm toán
function batdau()
{
alert("Kết quả của biểu thức: 3+5=?");
dapan=8;
/*bắt đầu tính thời gian, trong trường hợp này ta
gan một biến cho setTimeout để dễ hao tác hơn*/
hengio=setTimeout("sai();",5000);
}
//hàm thông báo khi sai
function sai()
{
alert("Đáp án đúng là "+dapan);
}
//hàm kiểm tra
function check()
{
    if(document.the_form.kq.value==dapan)
    {
        alert("Chúc mừng bạn đã trả lời đúng");
        //Xóa bỏ hẹn giờ với clearTimeout
    }
}
```

```

clearTimeout(hengio);
} else {
sai();
clearTimeout(hengio);
}
}
</script>

```

Hàm bắt đầu sẽ in ra câu hỏi, sau đó bắt đầu hẹn giờ, hết thời gian, setTimeout sẽ gọi hàm sai() thông báo đáp án. Hàm check sẽ được gọi khi gửi thông tin, nếu đáp án đúng thì gửi câu chúc mừng, sai thì gọi hàm sai, cả hai hành động đều dùng thêm một hàm **clearTimeout** để tắt stTimeout.

*Để làm việc với **clearTimeout** ta cần gán setTimeout với một giá trị*

bài sao sẽ hướng dẫn cách để chúng ta làm một đồng hồ bấm giờ với clearTimeout và setTimeout

Tạo một đồng hồ bấm giờ

Tháng Sáu 22, 2009, 03:26:41 PM gửi bởi [nvcnvn](#)

Nếu ta muốn làm một đồng hồ bấm giờ, khi bấm chat thì bắt đầu đếm, khi bấm ngưng thì ngừng lại thì làm thế nào nhỉ.

Dĩ nhiên là cần tới setTimeout và clearTimeout vì đang nằm trong loạt5 bài về chúng mà 🤖.

Cụ thể thì làm thế nào nhỉ, đồng hồ bấm giờ cứ mỗi giây tăng lên 1, vậy thì có nghĩ là phải gọi một hàm setTimeout thực hiện một hàm tăng lên 1, sau đó là phải làm sao cho hàm setTimeout chạy liên tục cứ sau mỗi 1 giây.

Trong đầu bạn chắc hẳn sẽ hiện ra ý tưởng về một vòng lặp, đó cũng là nhưng gì mình nghĩ tới đầu tiên. Đáng buồn thay:

Code:

```

<script type="text/javascript">
function thongbao()
{
alert("Thêm 3 giây nữa đã trôi qua");
}
i=0;
while(i<10)
{
setTimeout("thongbao();",3000);
i=i+1;
}
</script>

```

Mình viết đoạn code này với suy nghĩ, cứ mỗi lần các vòng lặp chạy, nó sẽ gọi hàm setTimeout, đợi 3 giây rồi chạy tiếp. Mọi thứ tan vỡ 🤖 khi mình chạy thử, sau khi đợi 3 giây thì cả 10 hộp cacnh3 báo hiện ra 🤖, câu trả lời là vì: Vòng lặp chỉ gọi hàm setTimeout chứ không đợi nó, đối với tốc độ khinh hồn của vòng lặp, 10 lần sẽ trôi qua ngay như cùng một lúc, và bạn chỉ vừa tắt hộp cacnh3 báo thứ nhất thì 9 hộp kia đã xuất hiện. Không thể nào giảm tốc độ của một vòng lặp được!

Vậy hãy thử nhìn vào hướng khác, nhìn rộng và xa ra, châu Mỹ chắn hạn :)) OAO, một hàm tự gọi lại chính nó trong một khoảng thời gian. Đó là cách giải quyết:

Code:

```
function thongbao()
{
alert("Thêm 3 giây nữa đã trôi qua");
setTimeout("thongbao();",3000);
}
```

Lần đầu tiên thì hơi bị trục trặc, xuất hiện ngay bảng thông báo nếu bạn chạy hàm, nhưng những lần su thì tron tru, đúng kết hoạch. Ta có thể sửa lỗi này bằng cách dùng một lần hàm setTimeout bên ngoài hàm, ngoài hàm này, sau lần đầu tiên hàm setTmeouor đó sẽ ngưng hoạt động, và hàm bắt đầu gọi hàm setTimeout của riêng nó.

và đây là đồng hồ bấm giờ của chúng ta:

Code:

```
<script type="text/javascript">
function baogio()
{
num=document.the_form.kq.value;
num=eval(num)+1;
document.the_form.kq.value=num;
hengio=setTimeout("baogio();",1000);
}
</script>
<form name="the_form">
<input type="text" name="kq" value="0" /><br />
<input type="button" value="Bắt đầu" onclick="baogio();" />
<input type="button" value="Dừng" onclick="clearTimeout(hengio);" />
</form>
```

Hãy thử chạy code nhé 😊

LINH TINH KHÁC

Làm việc với ngày giờ

Tháng Sáu 22, 2009, 04:00:50 PM gửi bởi [nvcnvn](#)

Javascript có khả năng lấy ra ngày giờ từ hệ thống của bạn và đem nó lên màn hình. Đầu tiên bạn cần khai báo một đối tượng kiểu thời gian:

Code:

```
noe=new Date();
Và dùng nó với một số Method theo kiểu : now.method, một số method
gateDate(): ngày dưới dạng số.
15
getDay(): Ngày trong tuần dưới dạng số với 0 là Chủ nhật, 1 là thứ hai...
2
```

getHours(): Giờ trong ngày với dạng số nguyên, 0 -->23

22

getMinutes(): phút 0-->59

35

getMonth(): Tháng dưới dạng số 0 là tháng 1, 1 là tháng 2....

8

getSeconds():Số giây, 0-->59

getTime(): giờ hiện tại tính theo mili giây

1253028911165

getFullYear(): Năm, nhưng mà cái này rắc rối lắm, hiển thị khác nhau trên từng loại trình duyệt

109

Dưới mỗi ví dụ mình đều để chèn kết quả vào, hãy thử xem lại bài những lúc khác nhau và trên những trình duyệt khác nhau, các bạn sẽ thấy!...

JavaScript getElementById

Tháng Sáu 22, 2009, 04:12:26 PM gửi bởi [nvcnvn](#)

getElementById đơn giản là một cách thay thế cách tham chiếu tới đối tượng một cách bình thường:

VD: Để tham chiếu tới giá trị của trường text có tên name, bên trong form the_form:

Code:

```
<form name="the_form">  
<input type="text" name="name" />  
</form>
```

ta dùng code:

Code:

```
document.the_form.name.value  
Nhưng nếu ta thêm thuộc tính id vào text name:
```

Code:

```
<form name="the_form">  
<input type="text" name="name" id="name" />  
</form>
```

ta có thể tham chiếu tới giá trị nó như sau:

Code:

```
document.getElementById("name").value
```

Với **getElementById** ta có thể dễ dàng tham chiếu tới mọi đối tượng, các thẻ div, các tag <p>.....chỉ cần thêm thuộc tính id vào để xác định các thẻ ta muốn làm việc.

JavaScript innerHTML

Tháng Sáu 22, 2009, 04:33:36 PM gửi bởi [nvcnvn](#)

innerHTML là phương thức lấy ra cái gì được bọc bởi cặp thẻ nào đó. Đây là một phương thức rất thú vị để làm việc với HTML hoàn toàn dùng javascript (bạn chỉ dùng javascript sinh ra mã html)

Ví dụ ta có :

Code:

```
<div id="1"><b id="2"><i>Xem dòng text mẫu</i></b></div>
<script>
document.write("Lần thứ hai vẫn thế vì chỉ bỏ cặp thẻ div
"+document.getElementById("1").innerHTML+"<br />");
document.write("nhưng lần sau thì chỉ còn in nghiêng "+document.getElementById("2").innerHTML+"<br
/>");
</script>
```

Kết quả của đoạn mã trên:

Xem dòng text mẫu

Lần thứ hai vẫn thế vì chỉ bỏ cặp thẻ div *Xem dòng text mẫu* nhưng lần sau thì chỉ còn in nghiêng *Xem dòng text mẫu*

Thay đổi định dạng dùng innerHTML.

Đơn giản là ta lấy nó ra, đặt lại nó, rồi lại chèn ngược vào. Đó là tất cả những gì trong function changeText()

Code:

```
<script type="text/javascript">
function changeText(){
    var oldHTML = document.getElementById('para').innerHTML;
    var newHTML = "<font color='\"#FF0000\"'>" + oldHTML + "</font>";
    document.getElementById('para').innerHTML = newHTML;
}
</script>
<p id='para'>Chào mừng các bạn đến với <b id='boldStuff3'>hocvui.net</b> </p>
<input type='button' onclick='changeText()' value='Change Text' />
```

Chào mừng các bạn đến với [hocvui.net](#)

JavaScript Void 0

Tháng Sáu 22, 2009, 10:34:52 AM gửi bởi [nvcnvn](#)

JavaScript Void 0 là một cách để click một link mà chả đi tới đâu. Bạn đừng nhầm với cách đặt giá trị của thuộc tính href="#", cái này có nghĩa là đi tới một phần nào trong trang. Còn cách mình đang nói tới là click vào mà chả làm gì cả.

hãy thử một link như thế này:

Code:

```
<a href="javascript: void(0)">Link chả đi tới đâu</a>
```


[Link chả đi tới đâu](#) [không có phản ứng khi click vào link này]

Đây là một cách hay để chạy một hàm, kết hợp với thuộc tính onclick, nhưng mọi thứ không chỉ có thế.

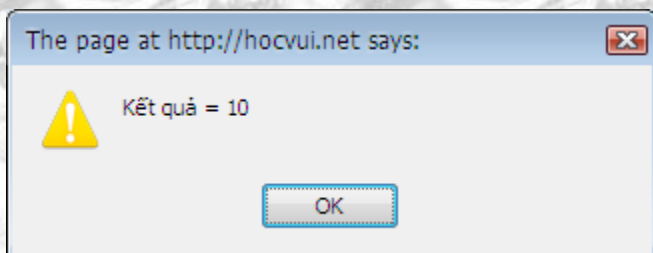
*Mình không muốn đi sâu vào vấn đề này, bởi vì phần lớn sự hữu dụng mà mình thấy đều đã nằm trên kia, phần dưới này theo mình là nâng cao hơn, tuy cũng ít ứng dụng hơn, nhưng kha là hay, chỉ có một ví dụ, các bạn hãy tự tìm hiểu thêm nhé:

Code:

```
<a href="javascript: void(num=10);alert("Kết quả = '+num)'">
```

```
Click xem nhé!</a>
```

[Click xem nhé!.....](#)



Vấn đề trình duyệt Navigator

Tháng Sáu 23, 2009, 10:49:30 AM gửi bởi [nvcnvn](#)

Có một điều rất "cực" các bạn khi thiết kế web dùng các client-script (script chạy trên máy khách) đó là tùy theo trình duyệt sẽ có một cách hoạt động khác nhau.

Như chúng ta đã biết sự khác nhau ở các trình duyệt thậm chí xuất hiện ở cả html, các màu và sự khác nhau không chỉ giữa các loại trình duyệt và ở cá phiên bản....

Đối với các script, một số function ở một số trình duyệt khác nhau sẽ có công dụng, cú pháp hơi khác nhau. Hay là cùng để làm một công việc mà tùy loại trình duyệt lại có những cú pháp nhất định.

Và thậm chí một số trình duyệt tự phát hành những method chỉ dành riêng cho mình,. Các bạn có nhớ những bài có đề cập tới DOM không, những đối tượng đó cũng thay đổi theo từng trình duyệt.

Thật may là không phải mọi thứ đều quá khác xa nhau, chủ yếu là chỉ có anh IE **rất** (Internet Explorer) là hơi khác các anh còn lại, những anh khác thì cũng có cái này cái kia...

Để tránh tình trạng lỗi xảy ra giữa các trình duyệt, ta cần hiểu rõ chúng và các hàm. Xem từng hàm hoạt động như thế nào, từng đối tượng có sự khác biệt như thế nào ở các trình duyệt. Chúng có là chuẩn của W3C hay không (Nếu là chuẩn của W3C thì sẽ ít có khác biệt giữa các trình duyệt)

Javascript cho phép ta phần nào làm được những điều đó bằng cách tìm ra tên và phiên bản của trình duyệt. Đó là đối tượng **Navigator**.

Để xác định tên trình duyệt:

Navigator.appName

Để xác định phiên bản trình duyệt:

Navigator.appVersion

Ví Dụ:

Code:

```
<script language="javascript">
document.write("Trình duyệt của bạn là: <b>"+navigator.appName+"</b> phiên bản:
"+navigator.appVersion);
</script>
```

Trình duyệt của bạn là: **Netscape** phiên bản: 5.0 (Windows; en-US)

Một cách mình hay dùng cho vấn đề này là dùng một lượt nhiều loại trình duyệt thông dụng như: IE, FF, Opera, Google Chrome, sau đó chạy thử trang web trên tất cả những trình duyệt đó.

Nếu có lỗi xảy ra do sự khác biệt và không có cách nào thay thế, ta có thể tạo những trang dành riêng cho những trình duyệt, tùy theo người dùng sử dụng trình duyệt nào mà ta sẽ chuyển họ tới (dùng cấu trúc if là đối tượng location).

Việc càng khó khăn, ở đó càng bộc lộ sự tin cậy của con người, đó chính là ưu điểm mà cho tới bây giờ, máy móc không bao giờ bì lại so với chúng ta.

COOKIES

Cookies dùng để làm gì

Thứ Bảy 04, 2009, 02:48:11 PM gửi bởi [nvcnvn](#)

Đối với những người mới vào nghề như chúng ta, các khái niệm mới luôn là điều khó hiểu. Đó là lời đầu tiên mình muốn nói, và lời thứ hai là: cookies không phải là bánh quy đâu ha 🍪🍪

Cookie là các thông tin lưu trong máy tính thường được dùng để nhận ra người dùng khi viếng thăm một trang web. Khi người dùng truy cập đến một trang web có sử dụng cookie, trang web đó sẽ tự động gửi cookie đến máy tính của người dùng. Những cookie này tự động được tổ chức trong hệ thống máy tính. Khi truy cập đến các trang web sử dụng được cookie đã lưu, Website sẽ đọc thông tin từ cookies đó, để làm gì đó theo yêu cầu của người lập trình. Đa số các trình duyệt hiện tại đều cho phép bạn tùy chỉnh việc lưu cookies, hoặc hoàn toàn không cho phép việc sử dụng cookies bởi một số lý do tế nhị.

Một ví dụ thực tế à khi bạn đăng nhập vào một website, đa phần ở mục đăng nhập của các website đều có chức năng lưu mật khẩu, đó chính là một trong những công dụng của cookies.

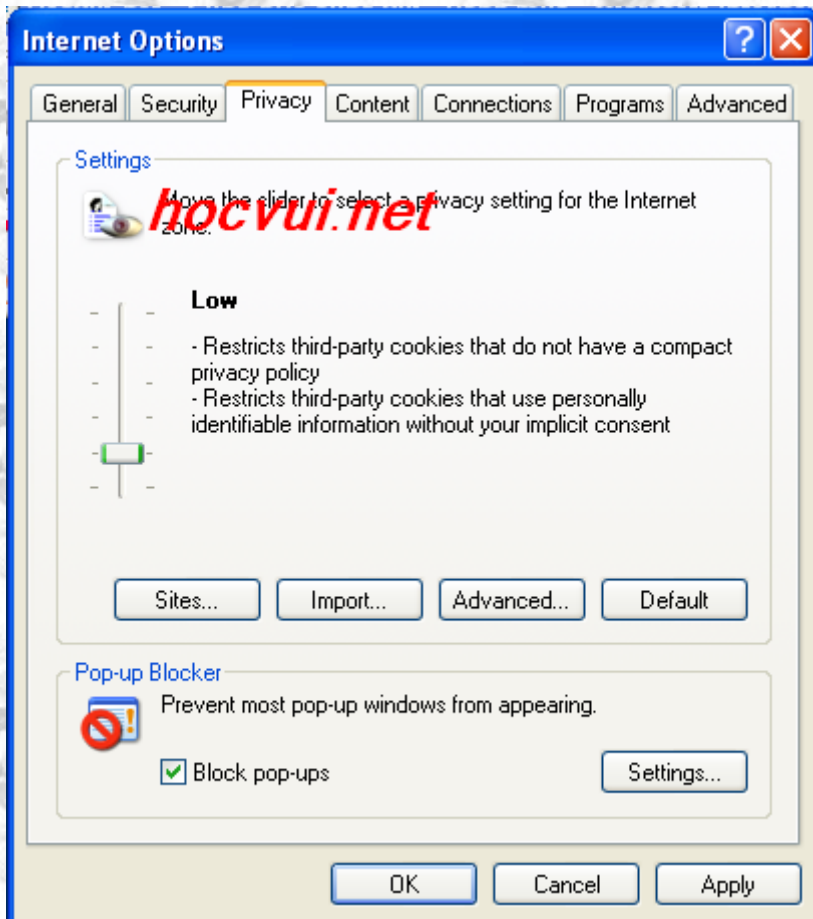
Đa số các ngôn ngữ dùng cho lập trình web đều có những công cụ làm việc với cookies, các bài sau sẽ hướng dẫn các bạn ghi và sử dụng cookies bằng javascript.

Bạn hãy tìm thử các file cookies được lưu ở thư mục có đường dẫn(dành cho các bạn sử dụng Window):
C:\Documents and Settings\Administrator\Cookies
với Administrator chính là user mà bạn dùng.

Để chắc chắn trình duyệt của bạn cho phép đọc và ghi các cookies, bạn làm theo các hướng dẫn sau:

Đối với Internet Explorer:

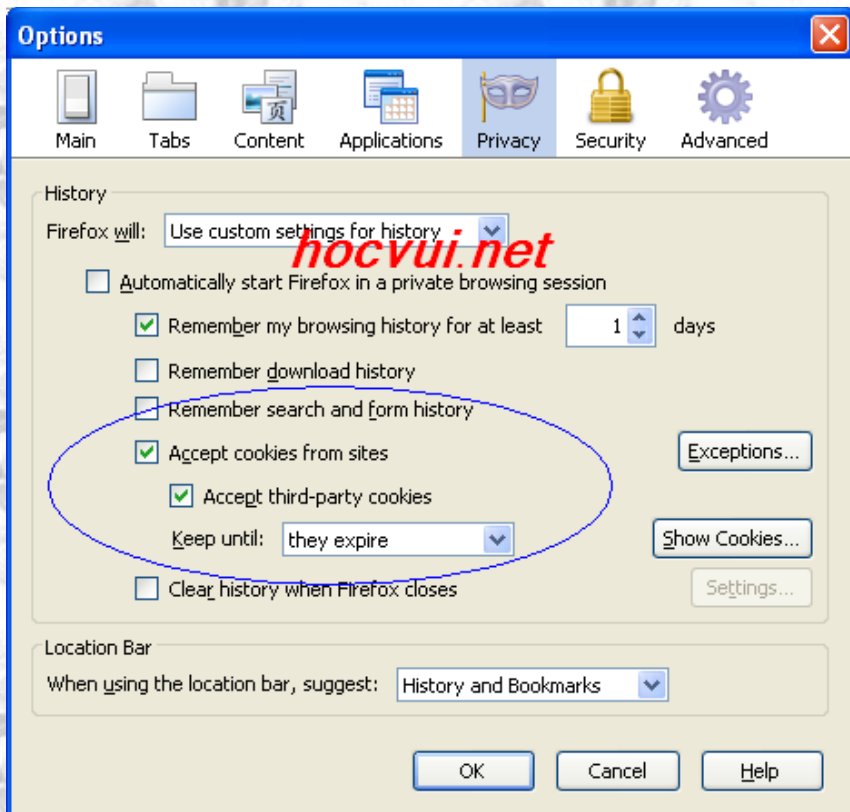
Bạn vào **Tools/Internet Option**, ở tag **Privacy** các bạn sẽ thấy:



Rê thanh kéo tới mức **Low** là OK

Đối với Fire Fox

Bạn vào **Tools/Option**, cũng ở tag **Privacy** các bạn sẽ thấy:



Kiểm tra lại xem những mục mình khoanh tròn có giống như vậy chưa, cứ đánh dấu như mình là OK. Rồi, chuẩn bị cho các bài sau là vừa!

Thiết lập cookies

Tháng Bảy 08, 2009, 02:23:23 PM gửi bởi [nvcnvn](#)

Tất cả các **Cookie** đều được lưu giữ trong đối tượng **document.cookie**, điều này có nghĩa là nó nằm trong một đối tượng mảng. Và tất cả mọi chuyển phải mảng chính là "móc" cái ta cần ra khỏi đống lộn xộn ấy!

Nhưng mà đầu tiên hãy đi vào thiết lập cookies cái đã!

Việc thiết lập cookie cũng khá đơn giản, mọi việc đơn giản chỉ là:

```
document.cookie="<tên cookies>=<nội dung>";
```

Nói thẳng ra là ta nhét một chuỗi mà đằng trước ký tự "=" được xem là tên, và sau đó là nội dung.

Còn một điều nên lưu ý nữa là một domain (tức 1 website) chỉ có thể tạo 20 cookies trên một khách và chắc chắn rằng vào thời buổi này bạn không bao giờ muốn lãng phí thứ gì! Cho nên nếu không cần thiết thì các cookies phải được xóa bỏ.

Thiết lập thời gian tồn tại của cookies

Cookies sẽ chỉ tồn tại cho tới thời điểm mà bạn quy định, ví dụ như bạn quy định cho nó tồn tại tới ngày 1 tháng 1 chẳng hạn, nhưng lưu ý một điều, thời gian phải theo định dạng **GMT** và hơi bị nản để làm theo cách này.

Một giải pháp khác chính là cho cookies tồn tại trong một khoảng thời gian từ khi thiết lập.

```
var exdate=new Date();
exdate.setDate(exdate.getDate()+365);
```

Với đoạn code trên, biến **exdate** đã lưu thời điểm trong nó là 365 ngày sau kể từ thời điểm cookies được thiết lập. Để thiết lập thời gian tồn tại cho cookies, bạn thêm **expire=<ngày hết hạn>** trong chuỗi nội dung của cookies, được

phân tách bằng dấu (;).

Hãy thử nối tất cả lại và tạo một đoạn code js tạo một cookies, trong đó lưu trữ một chuỗi mà bạn nhập vào

Code:

```
<script language="javascript">
var exdate=new Date();
var your_name=prompt("What's your name!?", "");
exdate.setDate(exdate.getDate()+365);
document.cookie=escape("last_time_visit="+your_name+";expire="+exdate.setDate(exdate.getDate()+365))
;
</script>
```

Hàm escape có tác dụng chuyển đổi những ký tự đặc biệt, ví dụ %20 sẽ thay thế cho khoảng trắng

bài sau chúng ta sẽ đi vào tìm hiểu cách đọc những gì mà ta đã viết...

Đọc cookies

Tháng Bảy 25, 2009, 09:36:50 AM gửi bởi [nvcnvn](#)

Code:

```
function getCookie(c_name)
{
if (document.cookie.length>0)
{
c_start=document.cookie.indexOf(c_name + "=");
if (c_start!=-1)
{
c_start=c_start + c_name.length+1;
c_end=document.cookie.indexOf(";",c_start);
if (c_end==-1) c_end=document.cookie.length;
return unescape(document.cookie.substring(c_start,c_end));
}
}
return "";
}
```

Ở đây mình có một hàm, hàm này sẽ đọc cookies theo tên, hàm này nhận vào một tham số **c_name** chính là tên cookies ta tìm, ta bắt đầu xem xét từng dòng:

if (document.cookie.length>0) dòng này xác định xem trong đối tượng cookies có lưu thông tin gì không, nếu trong đối tượng cookies có lưu thông tin, ta mới bắt đầu lục cookies của mình ra.

c_start=document.cookie.indexOf(c_name + "=");

Chắc hẳn các bạn còn nhớ bài nói về việc kiểm tra phần tử select của form. Việc này cũng tương tự như thế, tìm trong chuỗi cookies xem có tên cookies của ta hay không. dòng **if (c_start!=-1)** sẽ xem xét điều kiện trên.

c_start=c_start + c_name.length+1;

c_end=document.cookie.indexOf(";",c_start);

if (c_end===-1) c_end=document.cookie.length;

Ba dòng này xác định vị trí đầu tiên và cuối cùng của cookies, chắc các bạn đã hiểu cách nó tìm chứ!? 😊

return unescape(document.cookie.substring(c_start,c_end));

Dòng này trả giá trị của hàm trở về cookies được **unescape** (thực hiện hành động ngược lại với hàm **escape** ở bài trước)

ta có thể in ra và sử dụng cookies với đoạn javascript sau:

Code:

```
<script type="text/javascript">
username=getCookie('username');
if (username!=null && username!="")
{
  alert('Welcome again '+username+!');
}
else
{
  username=prompt('Please enter your name:', "");
  if (username!=null && username!="")
  {
    setCookie('username',username,365);
  }
}
</script>
```

Sau khi đoạn code này chạy, nó sẽ khởi động hàm check cookies, nếu giá trị của cookies trả về khác rỗng thì in ra, ngược lại thì chạy hàm **setCookie** để thiết lập cookies. Hàm này mình viết hơi khác với ví dụ trong bài trước để thuận tiện hơn.

Code:

```
function setCookie(c_name,value,expiredays)
{
  var exdate=new Date();
  exdate.setDate(exdate.getDate()+expiredays);
  document.cookie=c_name+ "=" +escape(value)+
  ((expiredays==null) ? "" : ";expires="+exdate.toGMTString());
}
```

Tin hẳn các bạn đã đủ mọi điều để hiểu hàm trên, chỉ có lưu ý một chút:

((expiredays==null) ? "" : ";expires="+exdate.toGMTString());

Dòng này là dạng viết tĩnh lược của câu lệnh **if** gồm có vế bên trong cặp dấu ngoặc (phía trước dấu ?) là điều kiện.

Phía sau dấu ? là hai dòng lệnh cho trường hợp đúng và sai, phân cách bởi dấu (:).

Mình nhắm vào sự trong sáng của mã, và nói thật, viết mã như trên đối với mình không dễ dàng gì, mình luôn thích dùng câu lệnh **if** cơ bản, và dùng các dấu ngoặc móc giới hạn các dòng lệnh, viết thể tuy rườm rà, không "chuyên nghiệp" theo một số người, nhưng lại rất dễ nhìn dễ hiểu, đúng theo nghĩa cơ bản 😊

Sở dĩ mình dùng nó trong ví dụ này là vì mình muốn giới thiệu thêm cho các bạn, chứ như lúc mình tự học, nhìn thấy mấy dòng này trong code người ta viết thì cũng phải nhăn mặt tư duy dữ lắm!

<http://hocvui.net/javascript/cookies.htm>

```
<html>
<head>
<script type="text/javascript">
```

```
function getCookie(c_name)
{
if (document.cookie.length>0)
{
c_start=document.cookie.indexOf(c_name + "=");
if (c_start!=-1)
{
c_start=c_start + c_name.length+1;
c_end=document.cookie.indexOf(";",c_start);
if (c_end==-1) c_end=document.cookie.length;
return unescape(document.cookie.substring(c_start,c_end));
}
}
return "";
}

function setCookie(c_name,value,expiredays)
{
var exdate=new Date();
exdate.setDate(exdate.getDate()+expiredays);
document.cookie=c_name+ "=" +escape(value)+
((expiredays==null) ? "" : ";expires="+exdate.toGMTString());
}

</script>
</head>

<body>
<p align="center"><a href="http://hocvui.net/javascript">
http://hocvui.net/javascript</a></p>
<script type="text/javascript">

username=getCookie('username');
if (username!=null && username!="")
{
alert('Welcome again '+username+!');
}
else
{
username=prompt('Please enter your name:','');
if (username!=null && username!="")
{
setCookie('username',username,365);
}
}

</script>
</body>
</html>
```

Đây là trang ví dụ bao gồm tất cả những gì ta đã đề cùng làm việc ở đây, mời các bạn cùng xem qua nhé!

...