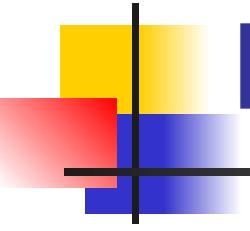




Bộ môn Công nghệ Phần mềm
Viện CNTT & TT
Trường Đại học Bách Khoa Hà Nội

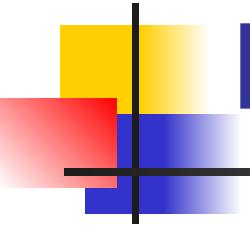
CÔNG NGHỆ WEB TIÊN TIẾN Javascript



Mục đích

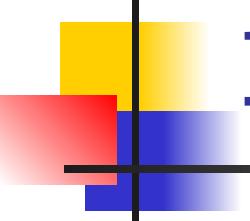
- Ví dụ về Javascript
- Cung cấp cho sinh viên những kiến thức về ngôn ngữ JavaScript để tạo ra sự tương tác của trang Web





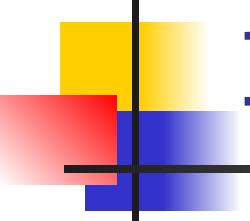
Nội dung

- 1. Giới thiệu về JavaScript
- 2. Vai trò của JavaScript
- 3. Chèn mã lệnh JavaScript
- 4. Các thành phần cơ bản
- 5. Cấu trúc điều khiển
- 6. Xử lý sự kiện với JavaScript
- 7. Một số đối tượng trong JavaScript
- 8. Trắc nghiệm
- 9. Thực hành



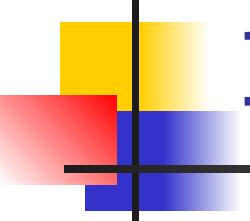
1. Giới thiệu về JavaScript

- Lý do ra đời
 - HTML, XHTML là các định dạng tài liệu, có thể được sử dụng để trao đổi dữ liệu trên mạng Internet
 - Các Web page ngày càng yêu cầu tính sáng tạo, độc đáo để hấp dẫn người sử dụng
 - Yêu cầu: tính tương tác của các web page
=> JavaScript và Visual Basic Script



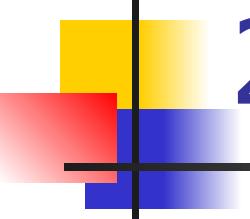
1. Giới thiệu về JavaScript

- JavaScript = Java + Script?
- Script?
 - Ngôn ngữ kịch bản
 - Là một ngôn ngữ lập trình cho phép tạo ra sự điều khiển trong một ứng dụng khác.
 - Các ngôn ngữ kịch bản thường được thông dịch



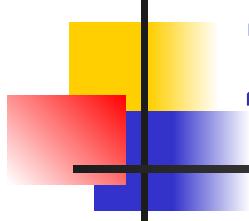
1. Giới thiệu về JavaScript

- Mục tiêu của JavaScript
 - Cung cấp cho người phát triển Web khả năng tạo ra sự tương tác trong một trang Web
 - Mã JavaScript có thể nhúng trong tài liệu HTML để điều khiển nội dung trang Web và hợp lệ dữ liệu người sử dụng nhập vào
- Kết luận
 - JavaScript là một ngôn ngữ kịch bản hướng đối tượng cho việc phát triển các ứng dụng Internet phía Client và Server.



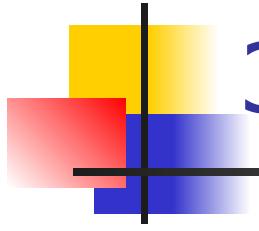
2. Vai trò của JavaScript

- Cung cấp sự tương tác với người dùng
 - Sự kiện do người dùng tạo ra: di chuột, click chuột...
 - Sự kiện do hệ thống tạo ra: thay đổi kích thước trang, tải trang...
- Thay đổi động nội dung
 - Thay đổi nội dung và vị trí các thành phần trên trang Web theo sự tương tác của người dùng



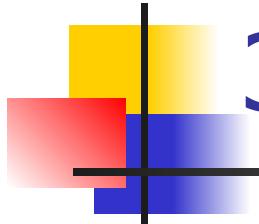
2. Vai trò của JavaScript

- Hợp lệ hóa dữ liệu
 - Kiểm tra xem dữ liệu có hợp lệ không trước khi nó được gửi (submit) đến Web Server để xử lý



3. Chèn mã JavaScript vào Web Page

- Viết trực tiếp trong tài liệu, sử dụng thẻ Script
- Liên kết đến một file JavaScript

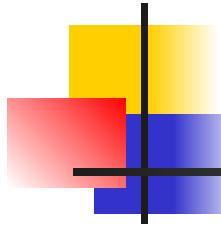


3.1. Viết trực tiếp trong tài liệu

- Cú pháp

```
<script language="JavaScript">  
  <!--  
    các lệnh JavaScript  
  -->  
</script>
```

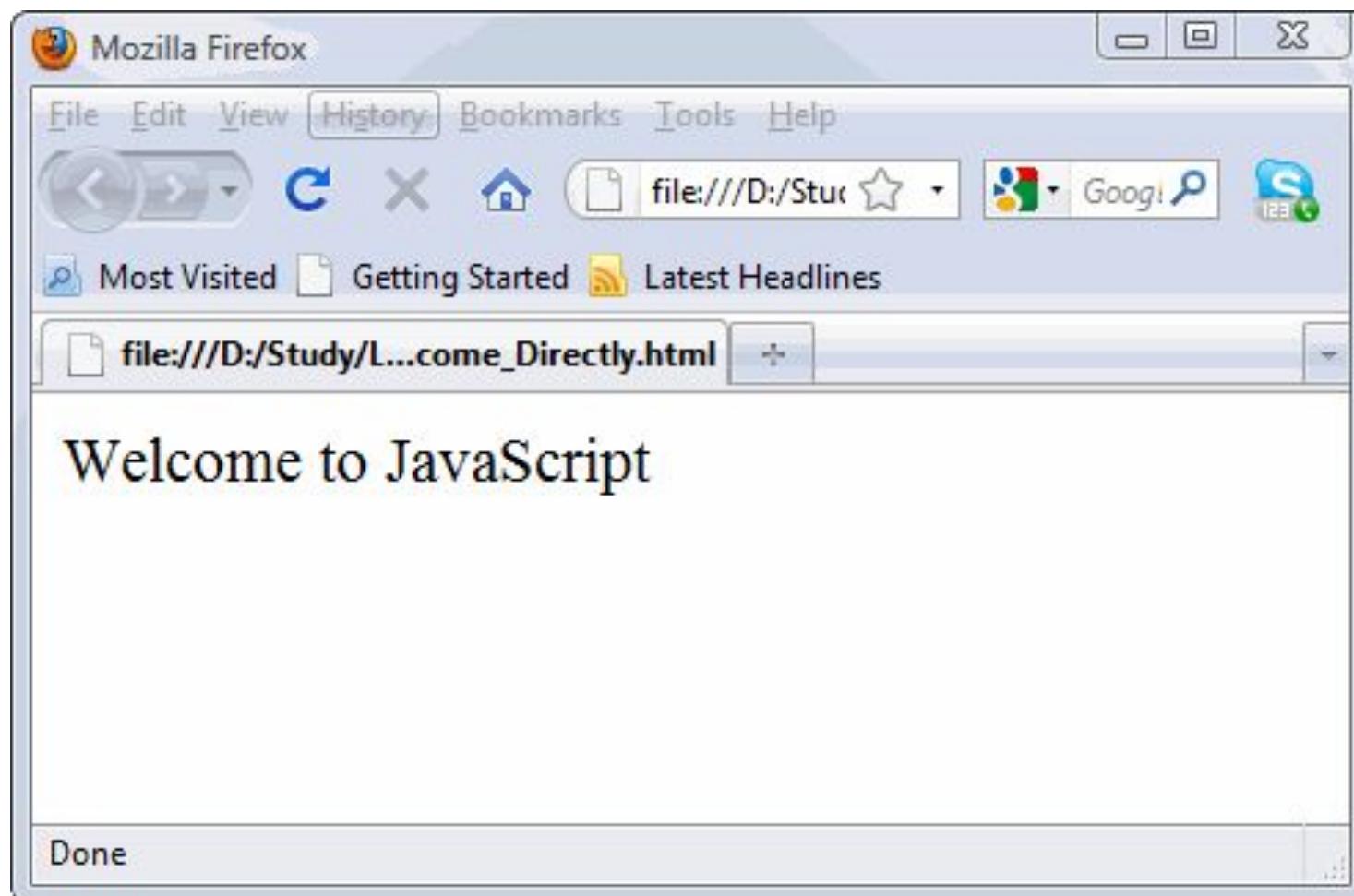
- Có thể xác định cụ thể phiên bản JavaScript
- Mỗi lệnh kết thúc bằng dấu ;

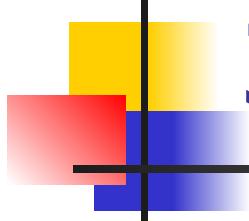


3.1. Viết trực tiếp trong tài liệu

```
<html>
  <head>
    <script language="javascript">
      document.write("Welcome to
                    JavaScript");
    </script>
  </head>
  <body>
    </body>
</html>
```

3.1. Viết trực tiếp trong tài liệu





3.2. Sử dụng file bên ngoài

- Cú pháp

```
<script language="JavaScript"  
        src="filename.js">  
    </script>
```

- File JavaScript (.js): chỉ bao gồm mã JavaScript, không chứa thẻ HTML

3.2. Sử dụng file bên ngoài

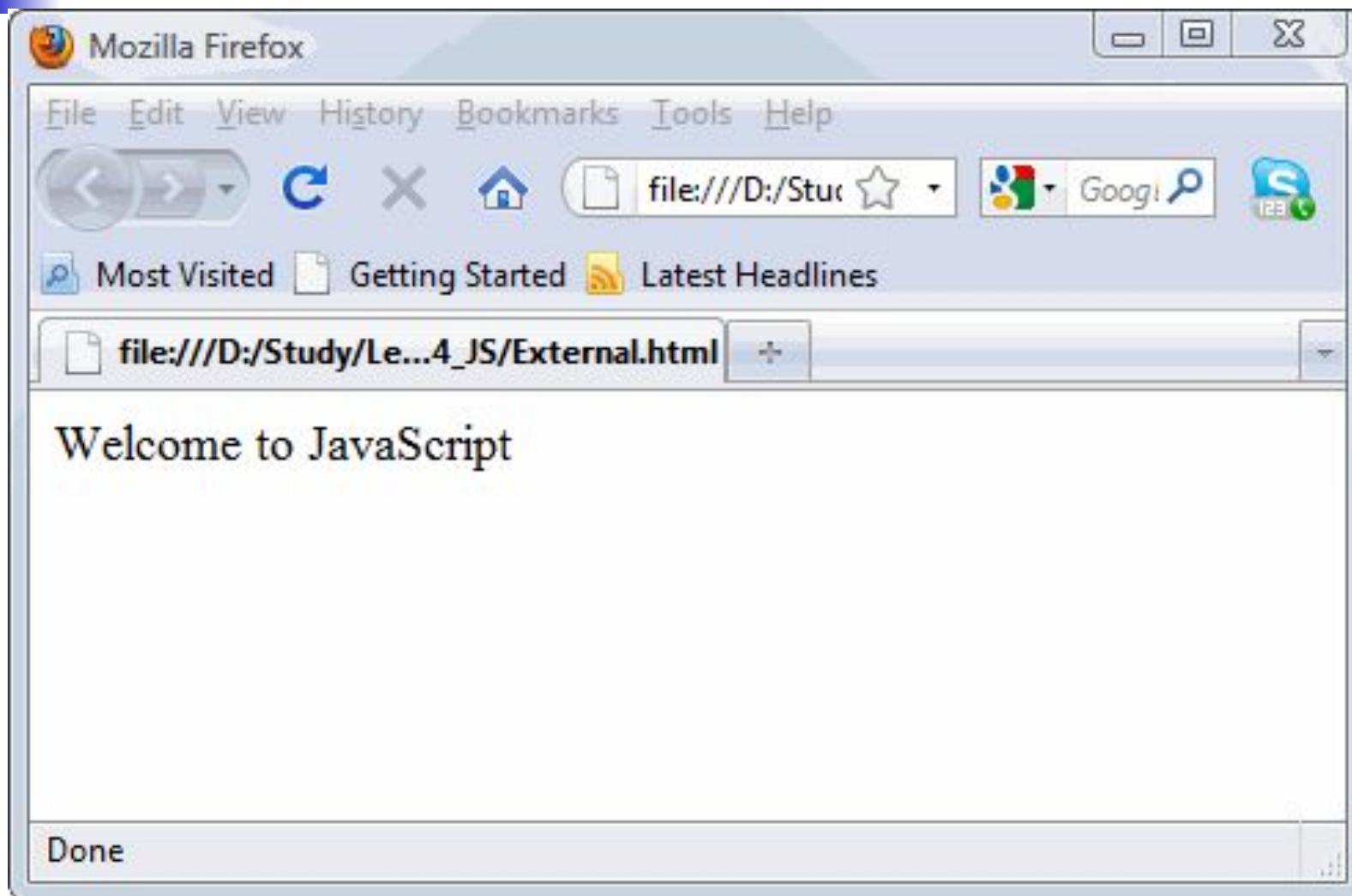
HTML file: external.html

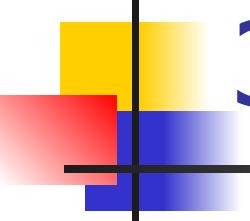
```
<html>
    <head>
        <script language="javascript"
            src="test.js">
        </script>
    </head>
    <body>
        </body>
</html>
```

JavaScript file: test.js

```
document.write("Welcome to JavaScript");
```

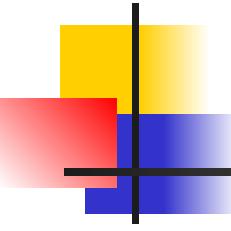
3.2. Sử dụng file bên ngoài





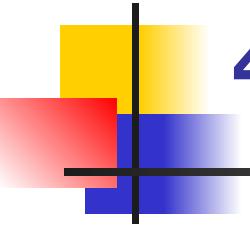
3.2. Sử dụng file bên ngoài

- So sánh
 - Mã JavaScript được chèn trực tiếp có thể đặt ở bất kì nơi đâu.
 - Đặt ở Head
 - Đặt ở Body
 - File JavaScript bên ngoài cho phép sử dụng lại nhiều lần và chỉnh sửa dễ dàng



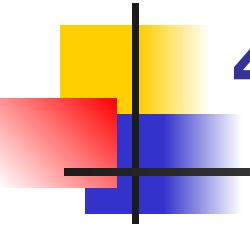
4. Các thành phần cơ bản của JavaScript

- 4.1. Biến
- 4.2. Kiểu dữ liệu
- 4.3. Phép toán
- 4.4. Biểu thức
- 4.5. Hộp thông báo
- 4.6. Chú thích
- 4.7. Các kí tự đặc biệt
- 4.8. Hàm



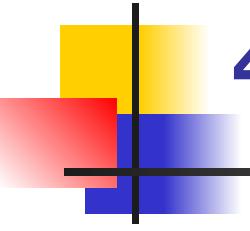
4.1. Biến

- Là đại lượng có thể thay đổi giá trị trong chương trình
- Quy tắc đặt tên
 - Tên phải bắt đầu bằng kí tự hoặc dấu gạch dưới
 - Tên có thể chứa số
 - JavaScript phân biệt chữ hoa và chữ thường



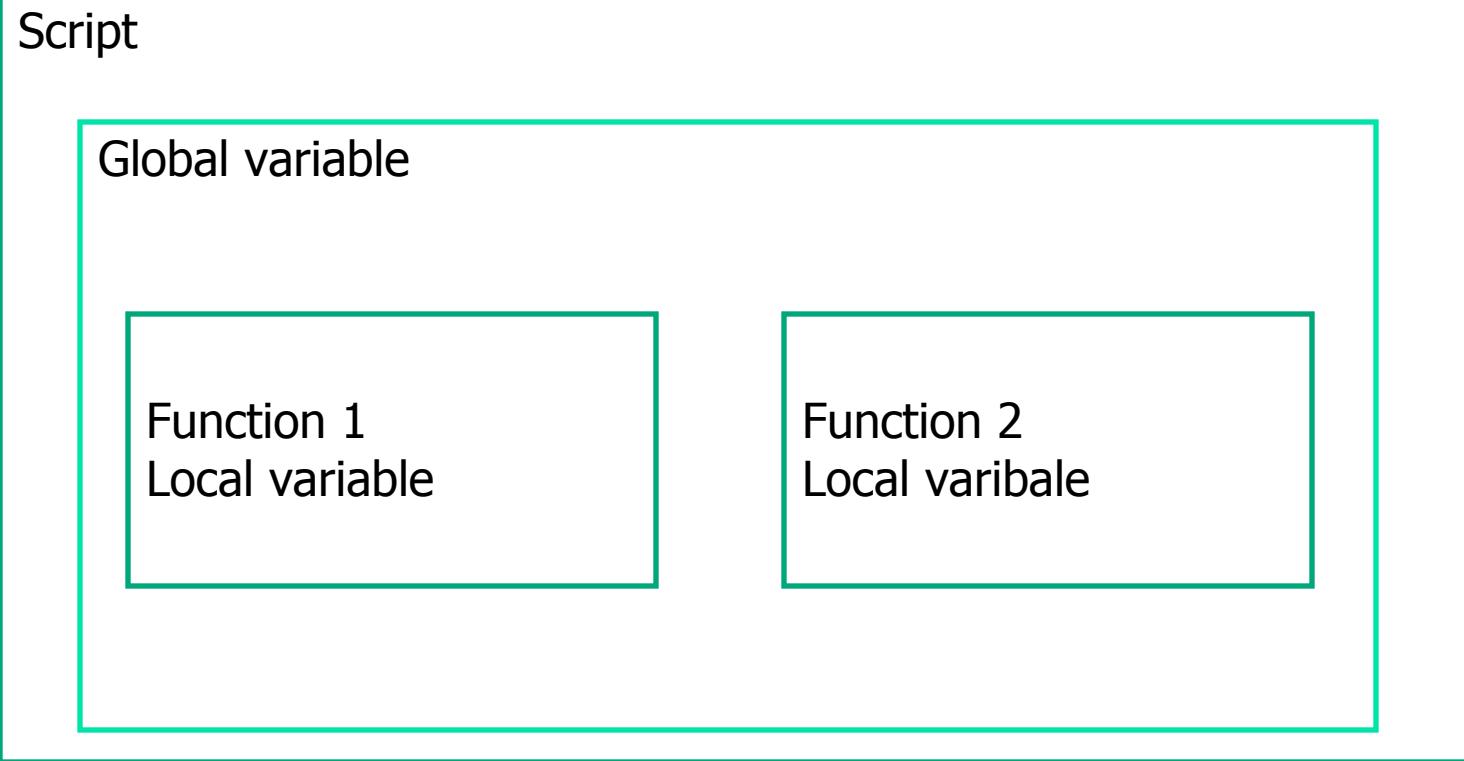
4.1. Biến

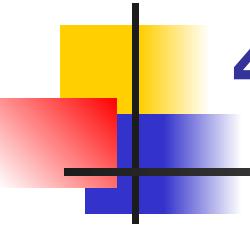
- Khai báo biến
 - Cú pháp: `var tên_biến;`
 - Khởi tạo giá trị cho biến cùng khai báo hoặc khởi tạo giá trị sau
 - `var A=10;`
 - hoặc
 - `var A;`
 - `A =10;`
 - Khai báo nhiều biến cùng dòng



4.1. Biến

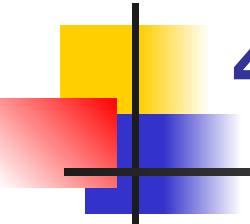
- Phạm vi của biến
 - Được xác định bởi nơi biến được khai báo





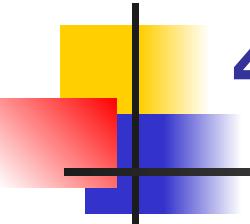
4.1. Biến

- Biến toàn cục
 - Khai báo bên ngoài hàm
 - Được truy cập từ mọi nơi trong kịch bản
 - Khai báo không cần từ khóa var
- Biến cục bộ
 - Khai báo bên trong hàm
 - Phạm vi trong hàm khai báo
 - Bắt buộc phải có từ khóa var



4.2. Kiểu dữ liệu

- Numbers
 - Số nguyên hoặc số thực như 12 hay 12.4
- Logical or Boolean
 - true hoặc false
- Strings
 - Xâu kí tự như "Thiet ke Web"
- null
 - Để cập đến giá trị null



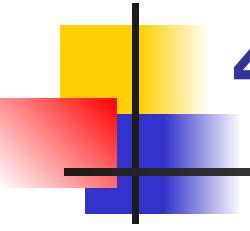
4.2. Kiểu dữ liệu

- Dữ liệu luôn luôn thuộc về một loại nào đó
- Biến sẽ thuộc về một kiểu dữ liệu khi gán giá trị
- Kiểu thay đổi khi dữ liệu thay đổi

```
var x = 10;
```

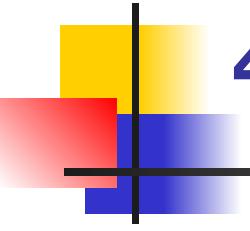
```
x = true;
```

```
x = "Hello World";
```



4.2. Kiểu dữ liệu

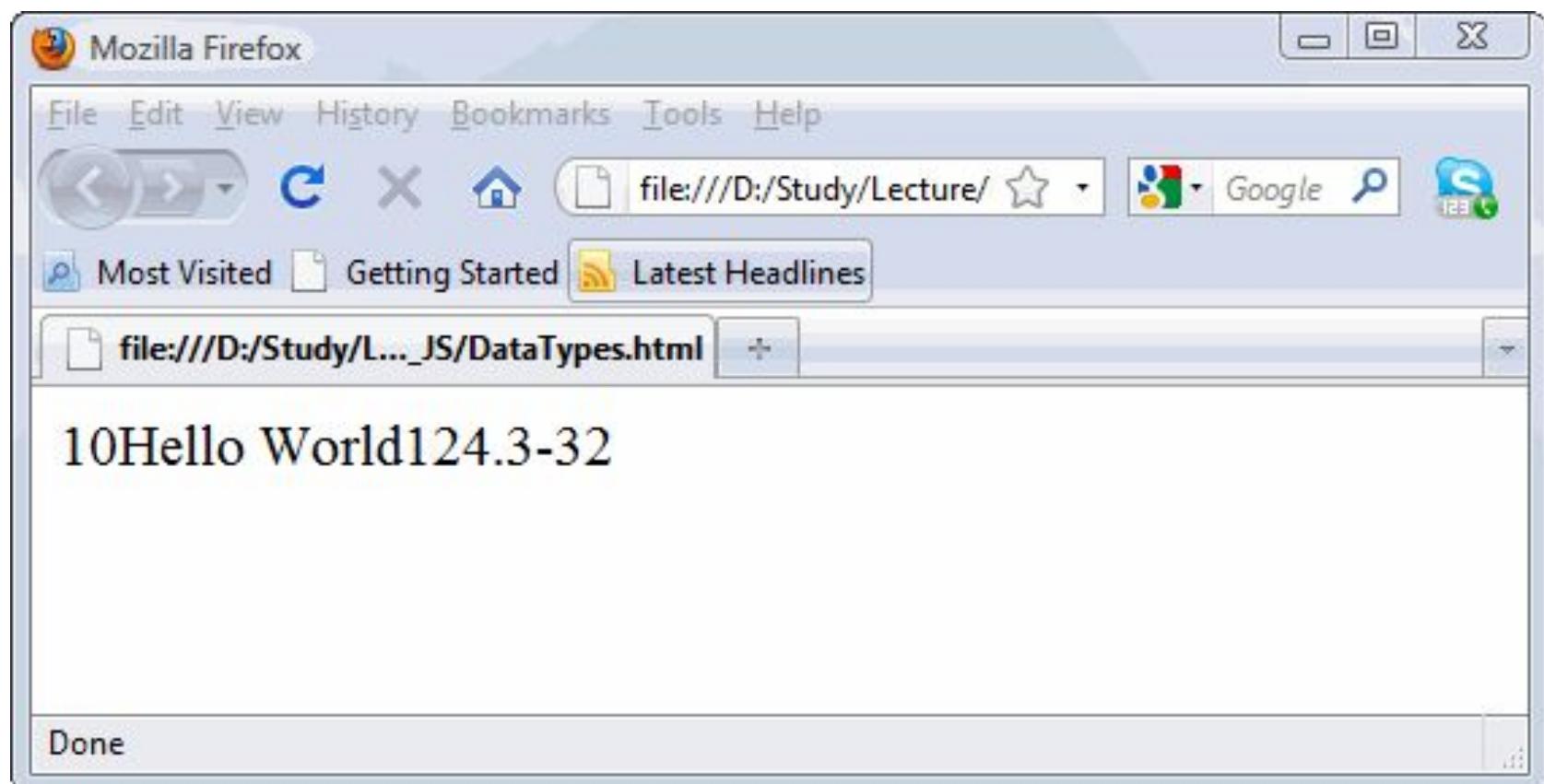
- Kết hợp hai biến khác kiểu: xâu và số...
var A = "Ha Noi " + 36 + "pho phuong";
- Hàm
 - parseInt(): chuyển xâu thành số nguyên
 - parseFloat(): chuyển xâu thành số thực
 - Chuyển không thành công: "NaN"

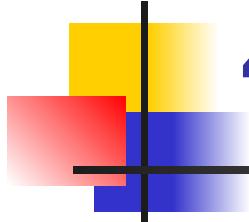


4.2. Kiểu dữ liệu

```
<script language="javascript">
    var x = 10;
    document.write(x);
    x=true;
    x = "Hello World";
    document.write(x);
    x = "12"+4.3;
    document.write(x);
    x = parseInt("-32");
    document.write(x);
</script>
```

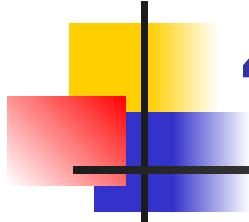
4.2. Kiểu dữ liệu





4.3. Phép toán

- 4.3.1. Phép toán số học
- 4.3.2. Phép toán so sánh
- 4.3.3. Phép toán logic
- 4.3.4. Phép toán định lượng

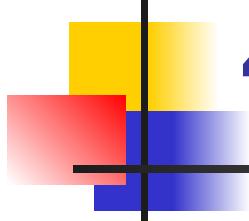


4.3.1. Phép toán số học

Phép toán	Mô tả	Ví dụ
+	Cộng	$A = 5 + 8$
-	Trừ	$A = 8 - 5$
/	Chia	$A = 20 / 5$
%	Số dư	$10 \% 3 = 1$
++	Tăng 1	++ x và x ++
--	Giảm 1	-- x và x --
-	Đảo dấu	nếu $A = 5$ thì $-A = -5$

4.3.2. Phép toán so sánh

Phép toán	Mô tả so sánh	Ví dụ ($x=5$)	Kết quả
$= =$	Bằng giá trị	$x == 5$	true
$= = =$	Bằng giá trị và kiểu	$x == = "5"$	false
$! =$	Không bằng	$x != 4$	true
$>$	Lớn hơn	$x > 5$	false
$>=$	Lớn hơn hoặc bằng	$x >= 5$	true
$<$	Nhỏ hơn	$x < 4$	false
$<=$	Nhỏ hơn hoặc bằng	$x <= 5$	true



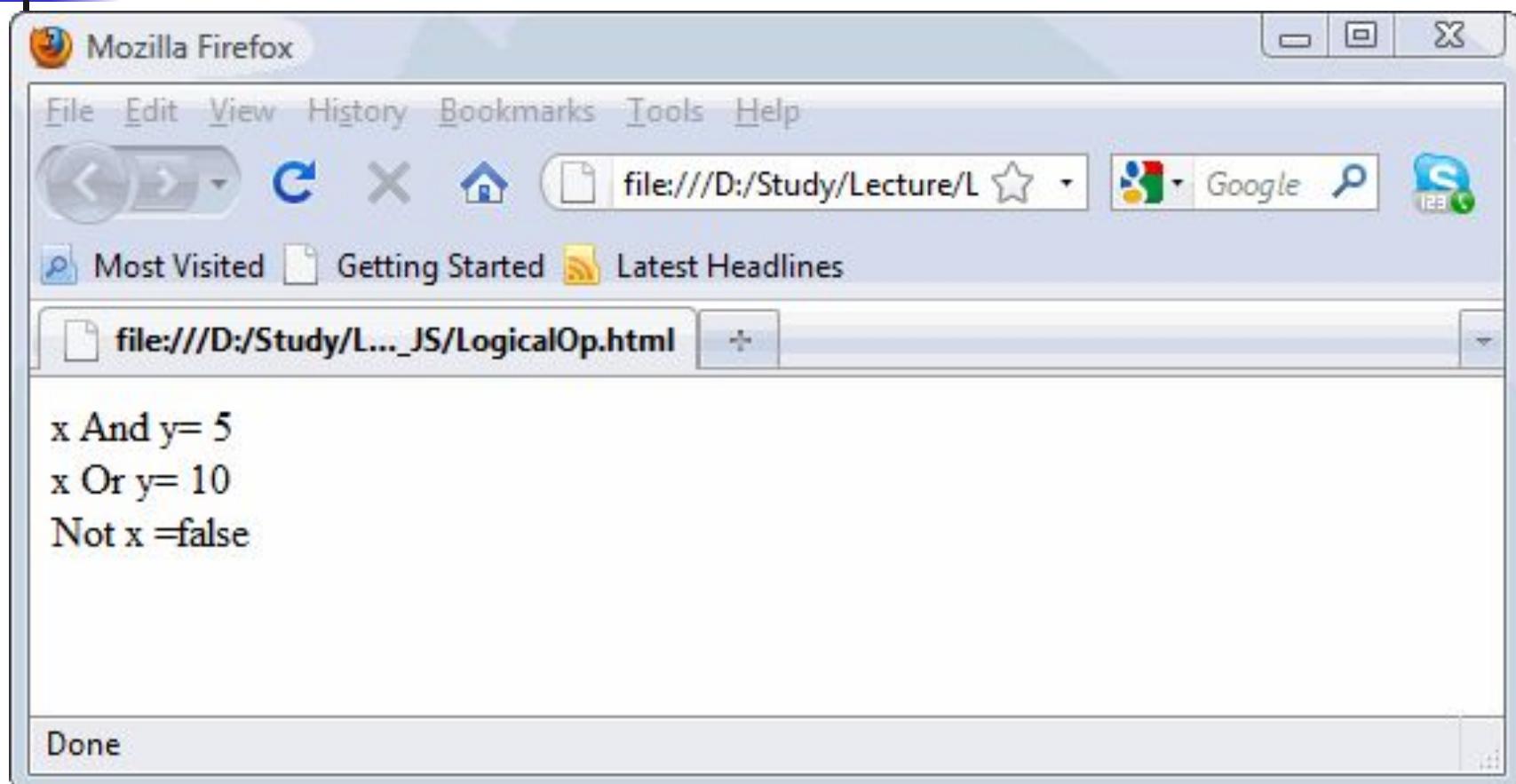
4.3.3. Phép toán logic

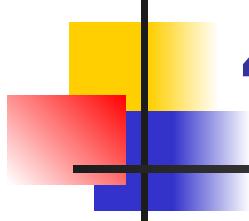
Phép toán	Mô tả	Ví dụ (x=6, y=3)	Kết quả
&&	And	$(x < 10) \&\& (y > 1)$	true
	Or	$(x < 1) (y > 5)$	false
!	Not	$! (x == y)$	true

4.3.3. Phép toán logic

```
<script language="javascript">
    var x=10;
    var y=5;
    document.write("x And y= "+(x&&y));
    document.write("<br/>");
    document.write("x Or y= "+(x||y));
    document.write("<br/>");
    document.write("Not x =" +( !x));
</script>
```

4.3.3. Phép toán logic





4.3.4. Phép toán định lượng

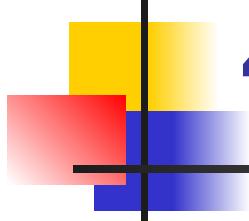
- Phép toán điều kiện

Cú pháp

tên biến = điều kiện ? giá trị1 :
giá trị2

Ví dụ:

status = (diem >= 5) ? "đạt": "thi lại"



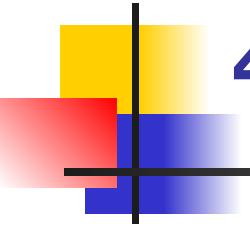
4.3.4. Phép toán định lượng

- **typeof**

- Trả về xâu xác định kiểu dữ liệu của biến

Ví dụ:

```
var x = 5;  
document.write(typeof(x));  
  
=> number
```



4.4. Biểu thức

- Là sự ghép nối các phép toán và toán hạng theo một quy tắc xác định
- Các toán hạng: biến, hằng, hàm hoặc biểu thức khác
- Thường thể hiện công thức tính toán một giá trị nào đó
- 3 loại
 - Biểu thức số học
 - Biểu thức logic
 - Biểu thức xâu

4.5. Hộp thông báo

- Sử dụng các phương thức: confirm, alert, prompt
- Phương thức alert()
Cú pháp

```
alert ("thông_báo");
```

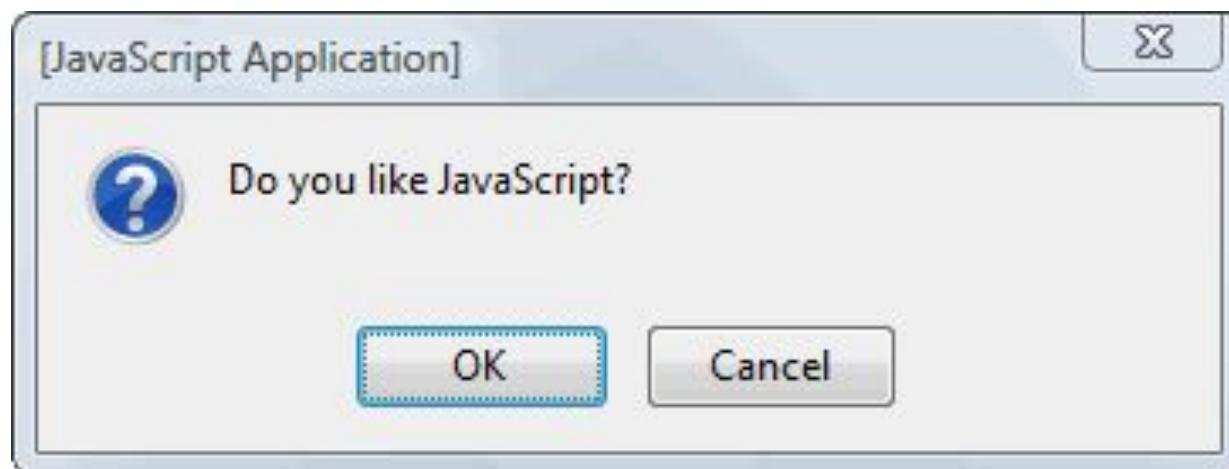


4.5. Hộp thông báo

- Phương thức confirm()

Cú pháp

```
ret = confirm ("thông_báo");
```

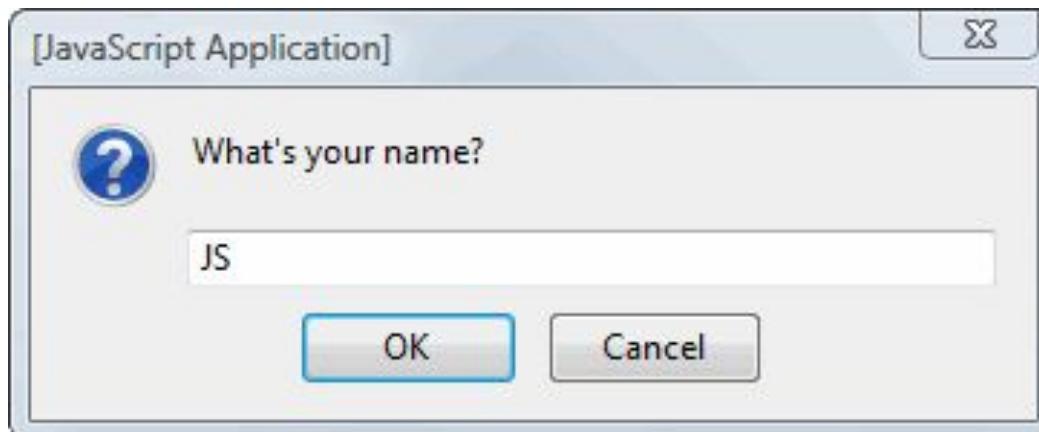


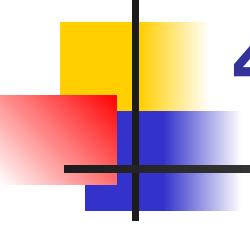
4.5. Hộp thông báo

- Phương thức prompt()

Cú pháp

```
value = prompt ("thông_báo",
giá_trị_mặc định);
```

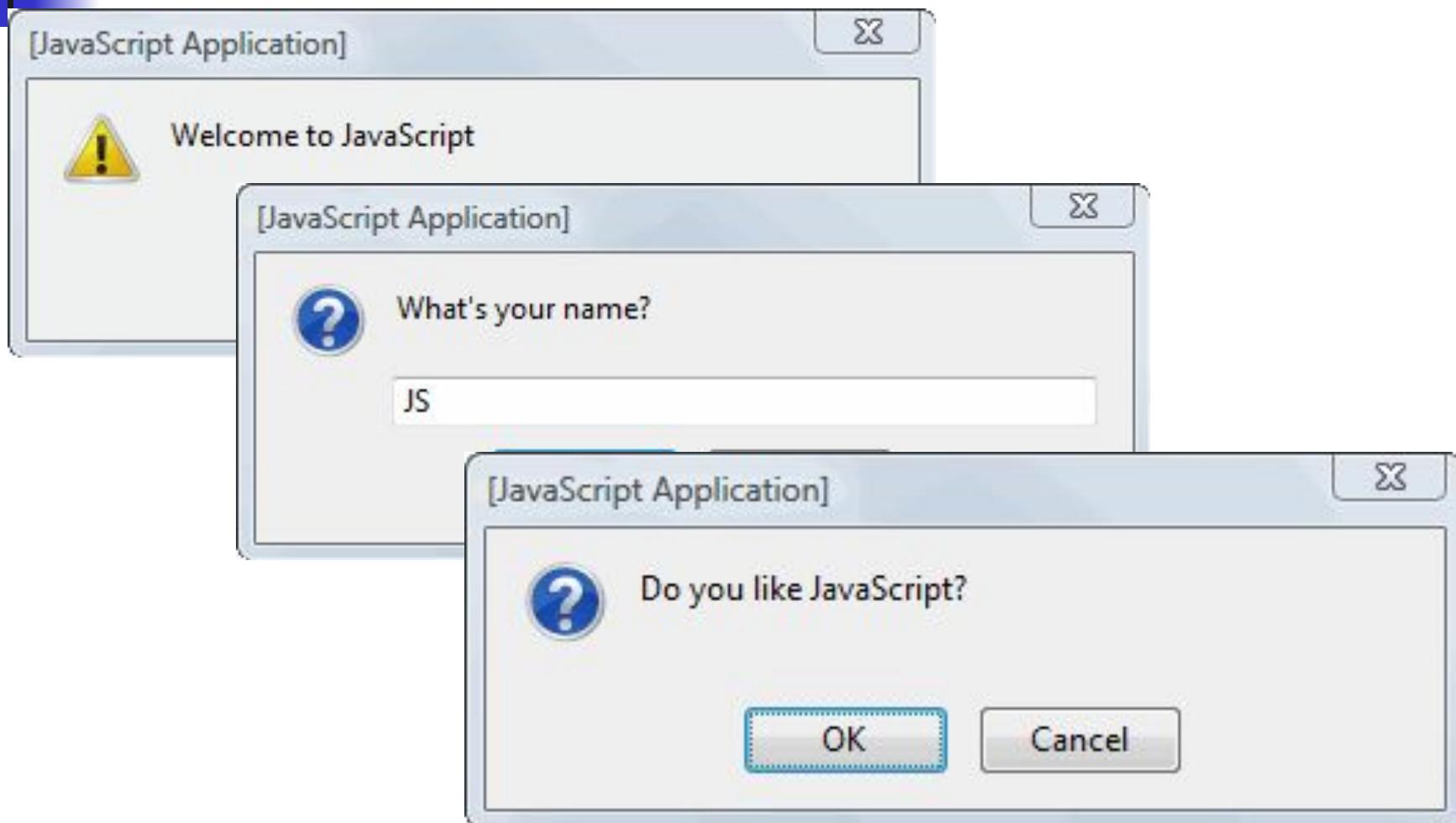


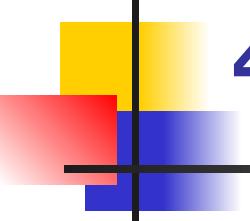


4.5. Hộp thông báo

```
<script language="javascript">
    alert("Welcome to JavaScript");
    var value = prompt("What's your
        name?", "JS");
    alert("Hello "+value);
    var ret = confirm("Do you like
        JavaScript?");
    alert("You pressed "+ret);
</script>
```

4.5. Hộp thông báo

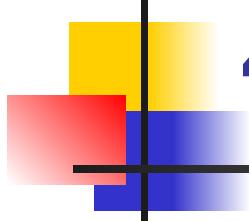




4.6. Viết chú thích

- Viết chú thích trên một dòng: //
- Viết chú thích trên nhiều dòng: /* và */

```
/*Todo: Example in MessageBox
include: alert(), confirm(),
prompt()
*/
var value = prompt("What's your
name?", "JS"); //get your name
```



4.7. Các kí tự đặc biệt

- **Vấn đề:**

```
var txt="He is very "intelligence"";
```

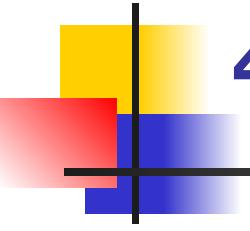
- **Các kí tự đặc biệt trong JavaScript:**

"' & \n r t b f

- **Để sử dụng:** thêm dấu \ phía trước

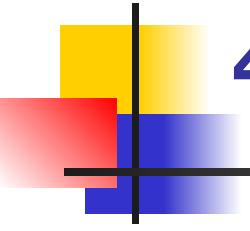
```
var txt=
```

```
"He is very \"intelligence\"";
```



4.8. Hàm

- Giới thiệu
 - Giúp trình duyệt không tự động thực hiện các kịch bản
 - Hàm chứa các lệnh sẽ được thực hiện khi có lời gọi hàm
 - Có thể được khai báo trong thẻ <head> hoặc <body>



4.8. Hàm

- Cú pháp

```
function tên_hàm(arg1, arg2,...) {  
    lệnh;  
}
```

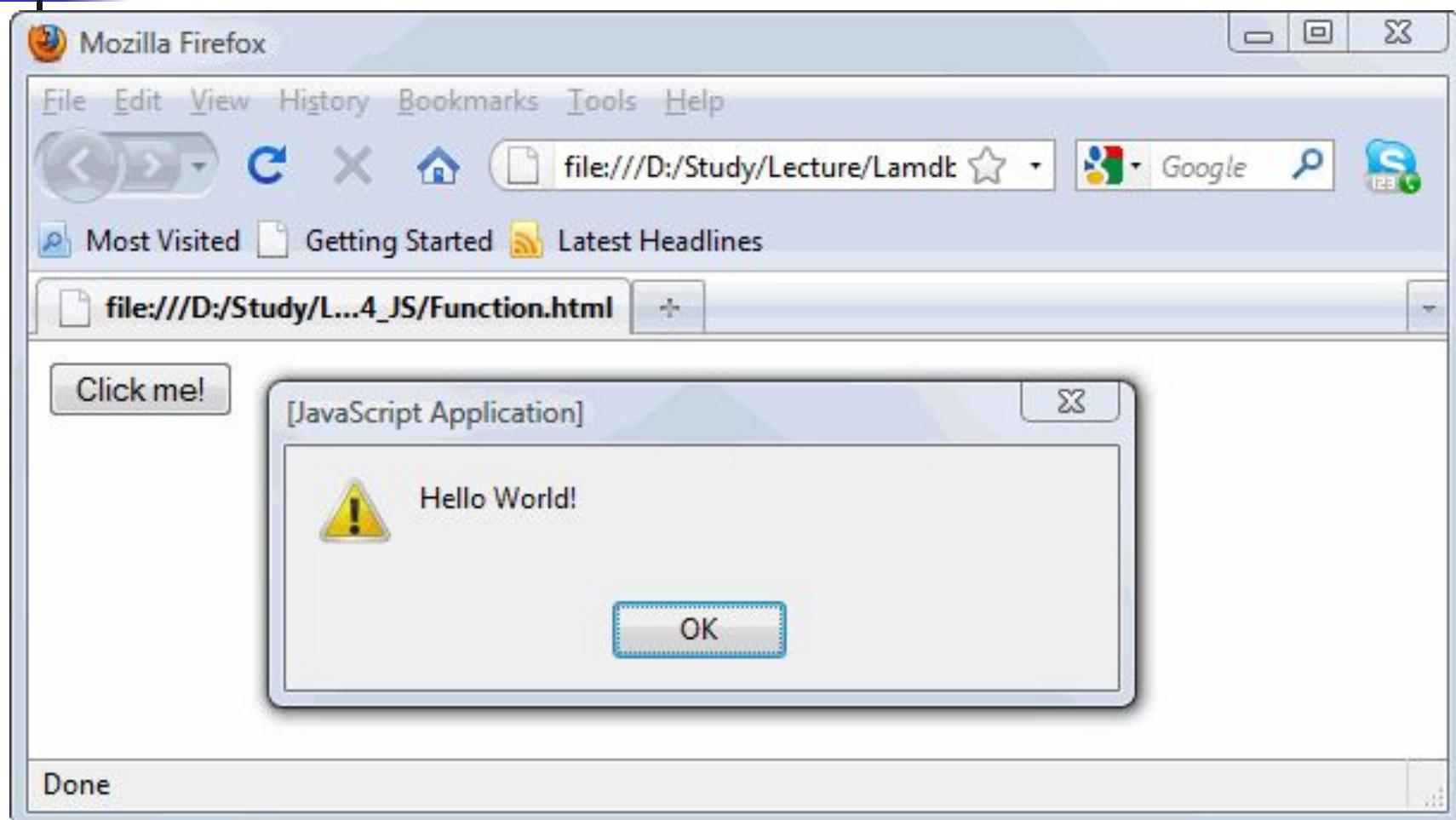
- Lời gọi hàm

```
tên_hàm(thamso1, thamso2,...);
```

4.8. Hàm

```
<html>
  <head>
    <script language="javascript">
      function displaymessage() {
        alert("Hello World!");
      }
    </script>
  </head>
  <body>
    <form>
      <input type="button" value="Click
me!" onclick="displaymessage(); " />
    </form>
  </body>
</html>
```

4.8. Hàm

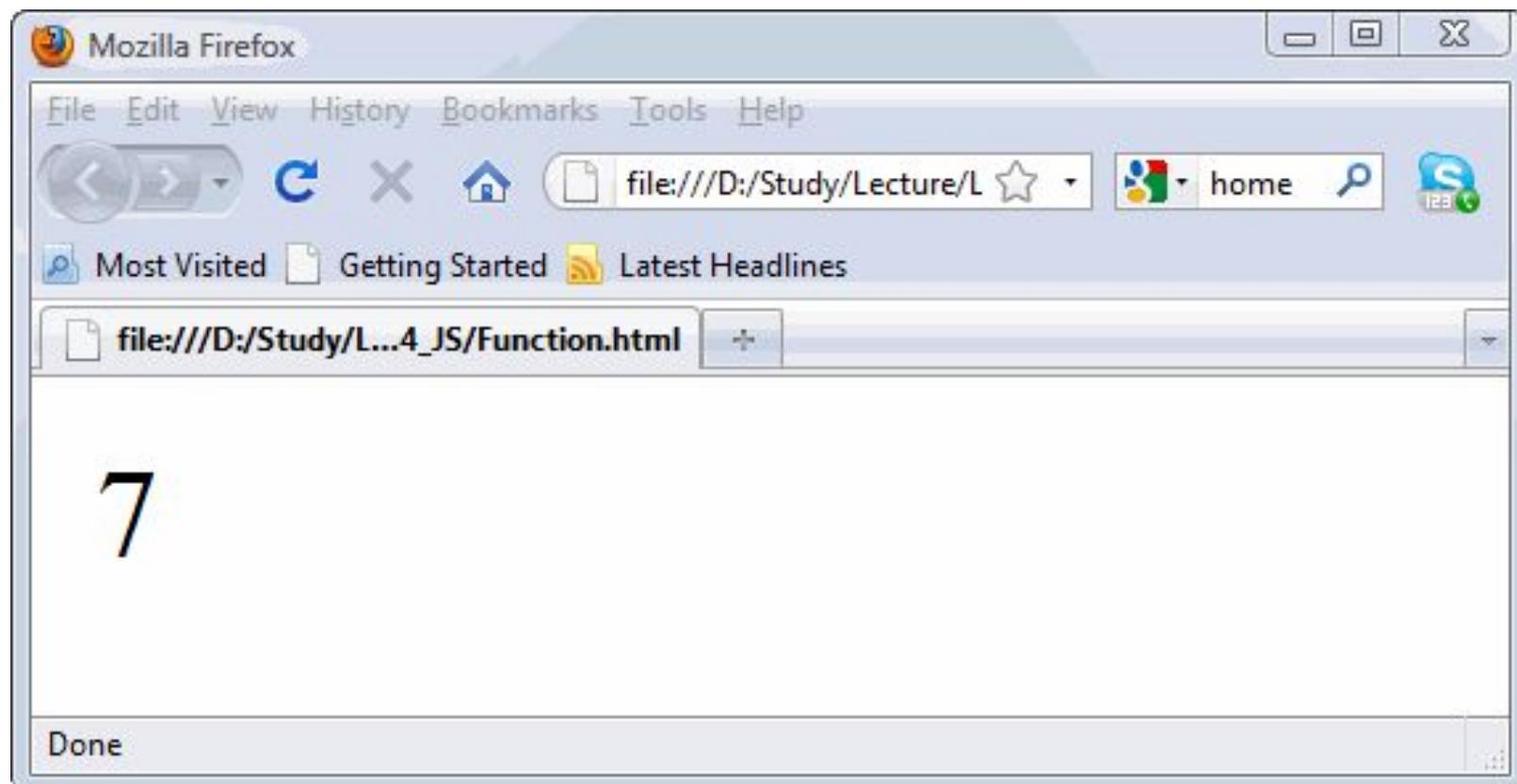


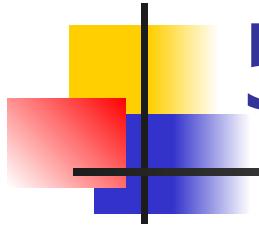
4.8. Hàm

- Lệnh return được sử dụng để xác định giá trị trả về từ hàm

```
<html>
  <head>
    <script language="javascript">
      function sum(a, b) {
        return (a+b);
      }
    </script>
  </head>
  <body>
    <script language="javascript">
      document.write(sum(4,3));
    </script>
  </body>
</html>
```

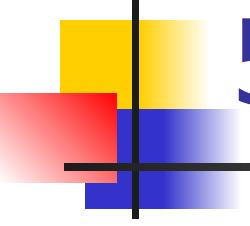
4.8. Hàm





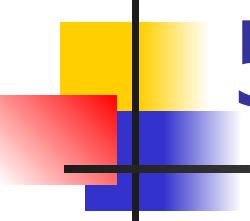
5. Cấu trúc điều khiển

- 5.1. Cấu trúc if, if...else
- 5.2. Cấu trúc switch
- 5.3. Cấu trúc for
- 5.4. Cấu trúc do while
- 5.5. Cấu trúc while
- 5.6. Câu lệnh break, continue
- 5.7. Cấu trúc for...in và with



5. Cấu trúc điều khiển

- **Đặc điểm cấu trúc lựa chọn**
 - Lựa chọn đoạn chương trình thực hiện dựa vào điều kiện chọn
- **Đặc điểm cấu trúc lặp**
 - Thực hiện lặp lại một đoạn chương trình
 - Lặp cho đến khi điều kiện lặp là sai (false)
 - Điều kiện lặp cần phải có sự thay đổi để kết thúc lặp



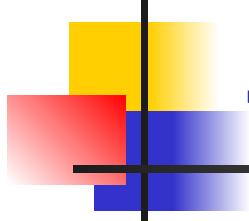
5.1. Cấu trúc if, if..else

- Câu trúc if

```
if (điều kiện)
    lệnh;
```

- Câu trúc if...else

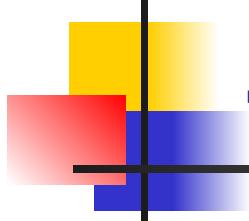
```
if (điều kiện)
    lệnh 1;
else
    lệnh 2;
```



5.2. Cấu trúc switch

- Cú pháp

```
switch (biểu_thức) {  
    case giá_trí_1: lệnh 1; break;  
    case giá_trí_2: lệnh 2; break;  
    ...  
    case giá_trí_n: lệnh n; break;  
    default: lệnh n+1; break;  
}
```



5.3. Cấu trúc for

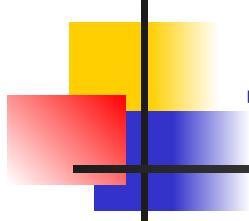
- Cú pháp

```
for ([biểu_thức_1];  
      [biểu_thức_2]; [biểu_thức_3])  
    lệnh;
```

- Trong đó

- biểu_thức_1: thường là câu lệnh khởi tạo
- biểu_thức_2: biểu thức điều kiện
- biểu_thức_3: thay đổi giá trị

- Thường dùng với số lần lặp xác định



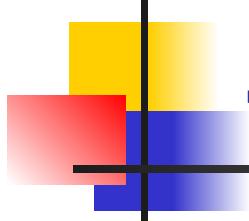
5.4. Cấu trúc do...while

- Cú pháp

```
do {  
    lệnh;  
} while (điều_kiện)
```

- Chú ý

- điều_kiện được kiểm tra *sau* khi thực hiện lệnh



5.5. Cấu trúc while

- Cú pháp

```
while (điều_kiện) {  
    lệnh;  
}
```

- Chú ý

- điều_kiện được kiểm tra *trước* khi thực hiện lệnh

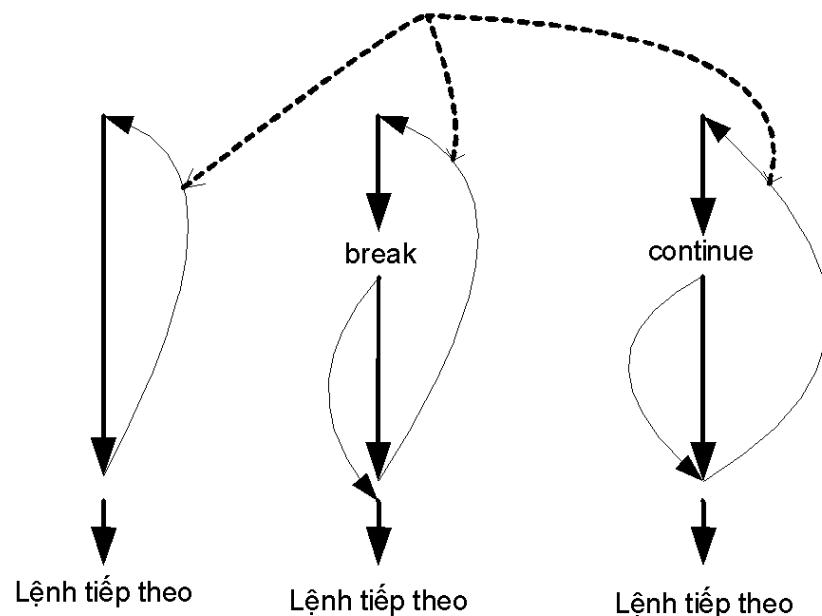
5.6. Câu lệnh break, continue

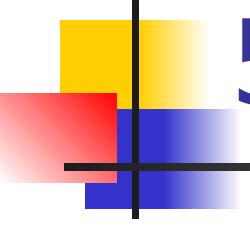
Mục đích:

- Thoát khỏi vòng lặp ngay cả khi điều kiện lặp còn thỏa mãn
- Chuyển sang vòng lặp mới

Vòng lặp với các lệnh **break** và **continue**

Điều kiện lặp còn thỏa mãn

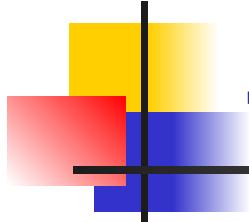




5.6. Câu lệnh break, continue

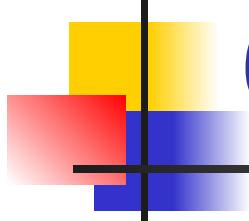
- Ví dụ:

```
for(i=0; i<10; i++) {  
    if(i%2==0) continue;  
    if(i==7) break;  
    document.write(" "+ i);  
}
```



5.7. Cấu trúc for...in và with

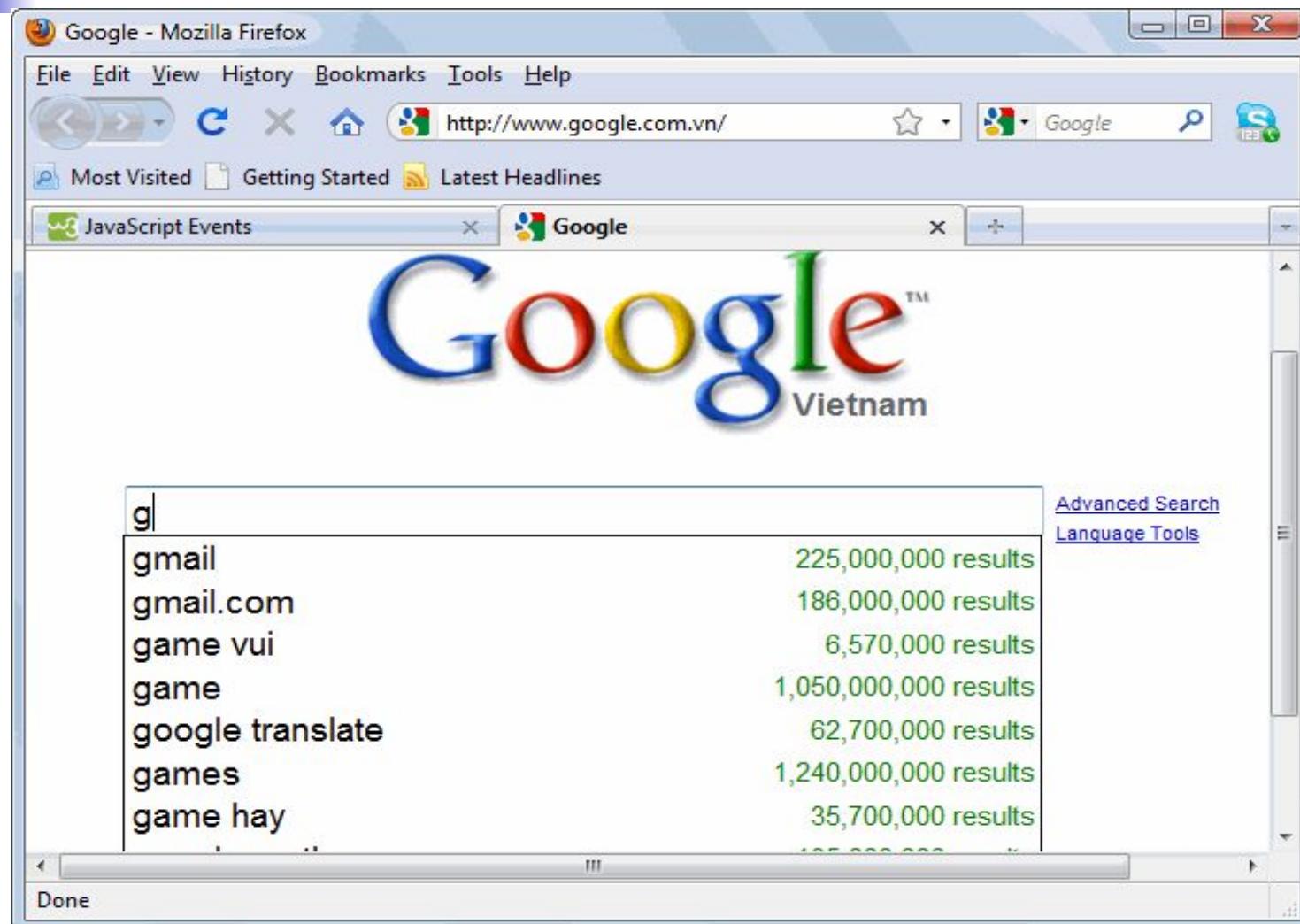
- Xem trong phần Đối tượng



6. Xử lý sự kiện với JavaScript

- 6.1. Giới thiệu về sự kiện
- 6.2. Xử lý sự kiện tải, đóng trang
- 6.3. Xử lý sự kiện chuột
- 6.4. Xử lý sự kiện nhập dữ liệu

6. 1. Giới thiệu về sự kiện



6. 1. Giới thiệu về sự kiện

2. Select an ID and password

Yahoo! ID and Email @ Password Strength

Password

Re-type Password

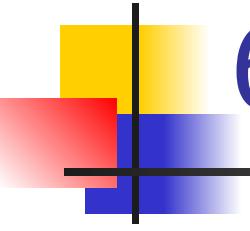
2. Select an ID and password

Yahoo! ID and Email @ Password Strength

Capitalization matters. Use 6 to 32 characters, no spaces, and don't use your name or Yahoo! ID.

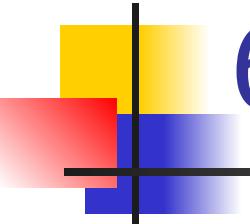
Password

Re-type Password



6. 1. Giới thiệu về sự kiện

- Sự kiện (event)
 - Là các hành động có thể được thể hiện bằng JavaScript
 - Mỗi thành phần trên trang Web đều có những sự kiện xác định, có thể kích hoạt một script
- Ví dụ
 - Sự kiện click chuột
 - Trang web hoặc ảnh được tải
 - Nhập dữ liệu vào một trường trên form



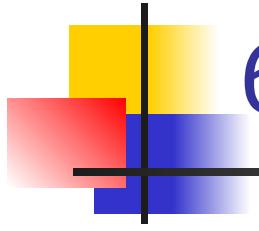
6. 1. Giới thiệu về sự kiện

- Chú ý:
 - Sự kiện thường được kết hợp với hàm. Hàm sẽ không thực thi chừng nào sự kiện chưa xuất hiện
 - Cú pháp

```
sự_kiện = "tên_hàm () ;"
```

hoặc

```
sự_kiện = "tên_hàm (ds_tham_số) ;"
```



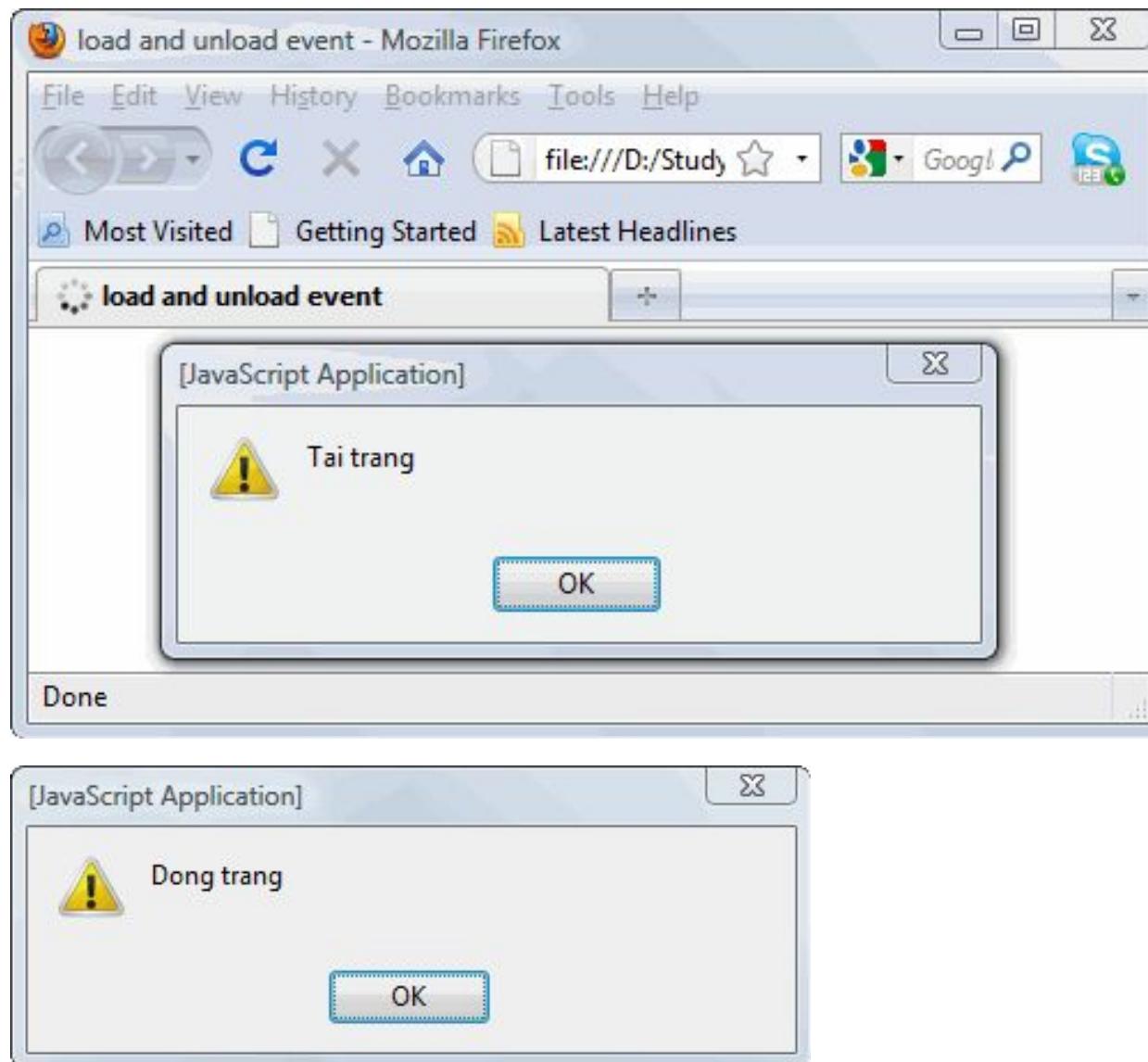
6.2. Xử lý sự kiện tải và đóng trang

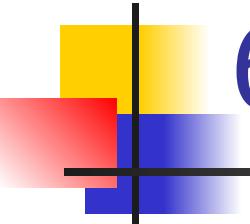
- Tải một trang
Sự kiện: onload;
- Đóng một trang
Sự kiện: onunload;

6.2. Xử lý sự kiện tải và đóng trang

```
<html>
    <head>
        <script language="javascript">
            function load() {
                alert("Tai trang");
            }
            function unload() {
                alert("Dong trang");
            }
        </script>
    </head>
    <body onload="load(); " onunload="unload(); ">
    </body>
</html>
```

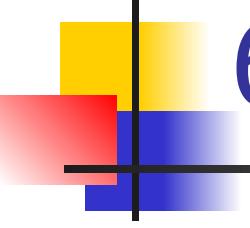
6.2. Xử lý sự kiện tải và đóng trang





6.3. Xử lý sự kiện chuột

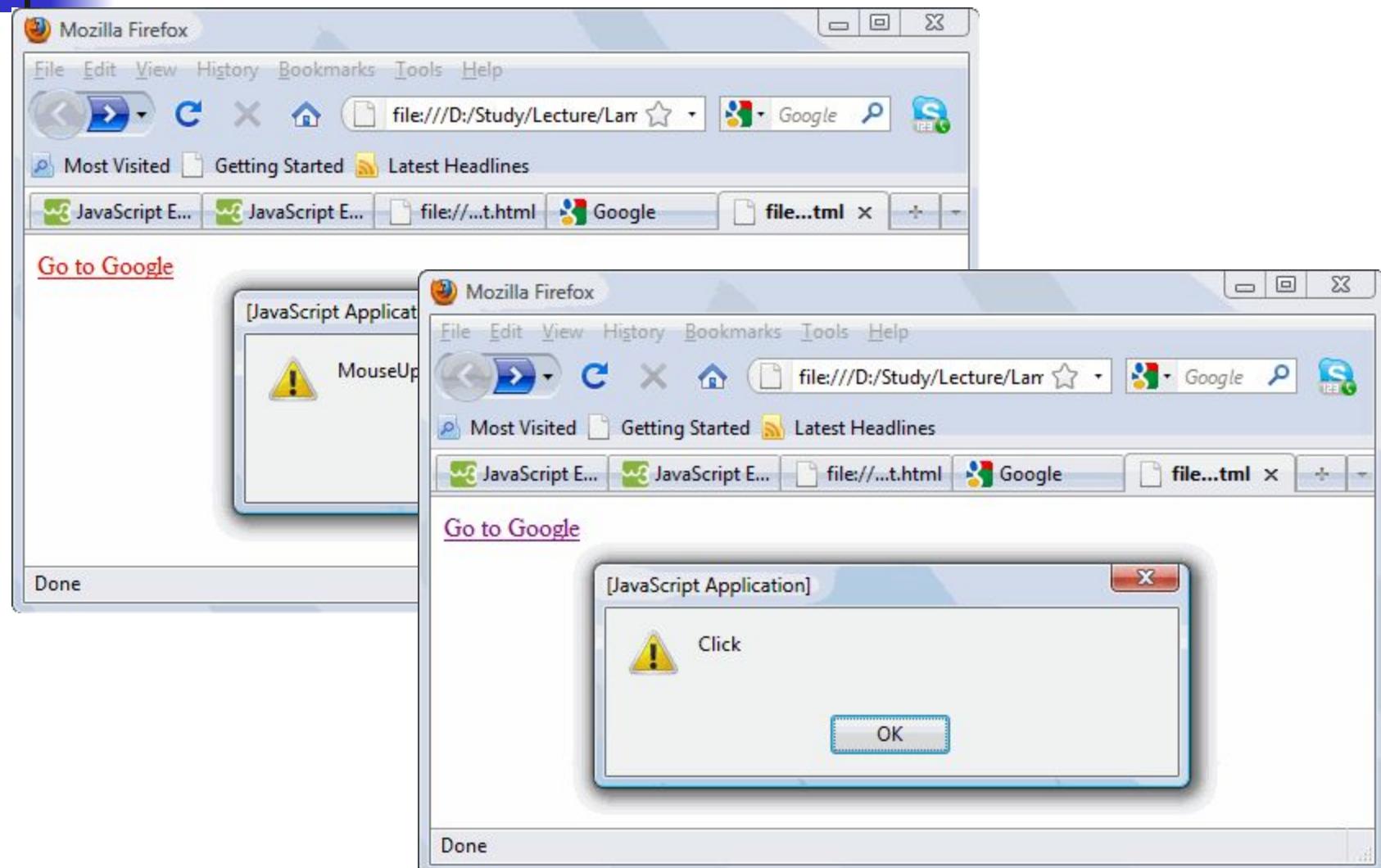
Sự kiện	Ý nghĩa
onMouseUp	nút chuột được nhả
onMouseDown	ấn chuột
onMouseOver	chuột di chuyển qua vùng
onMouseOut	chuột di chuyển khỏi vùng
onMouseMove	chuột được di chuyển
onclick	click chuột
ondbclick	click đúp chuột

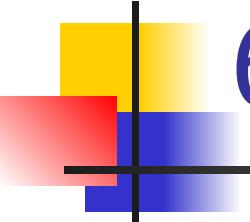


6.3. Xử lý sự kiện chuột

```
<html>
  <body>
    <a href="http://google.com.vn"
      onClick="alert ('Su kien Click');"
      onMouseUp="alert ('Su kien
      MouseUp');">
      Go to Google
    </a>
  </body>
</html>
```

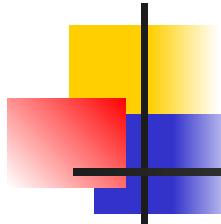
6.3. Xử lý sự kiện chuột





6.4. Xử lý sự kiện nhập dữ liệu

Sự kiện	Ý nghĩa
onFocus	Đối tượng được sử dụng (tập trung)
onBlur	Đối tượng không còn được sử dụng
onChange	Đối tượng được thay đổi (nhập dữ liệu)
onSubmit	Xác nhận tất cả dữ liệu trên form



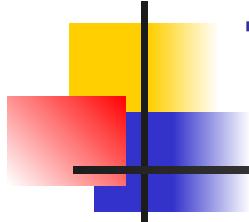
7. Một số đối tượng trong JavaScript

- 7.1. Giới thiệu về đối tượng
- 7.2. Đối tượng của Browser
- 7.3. Đối tượng của JavaScript
- 7.4. Đối tượng của HTML

7.1. Giới thiệu về đối tượng

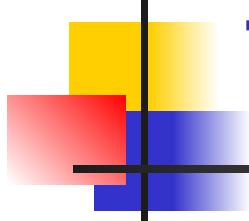
- Đối tượng là một thực thể trong thế giới thực được mô hình hóa
- Đối tượng = Dữ liệu + Phương thức



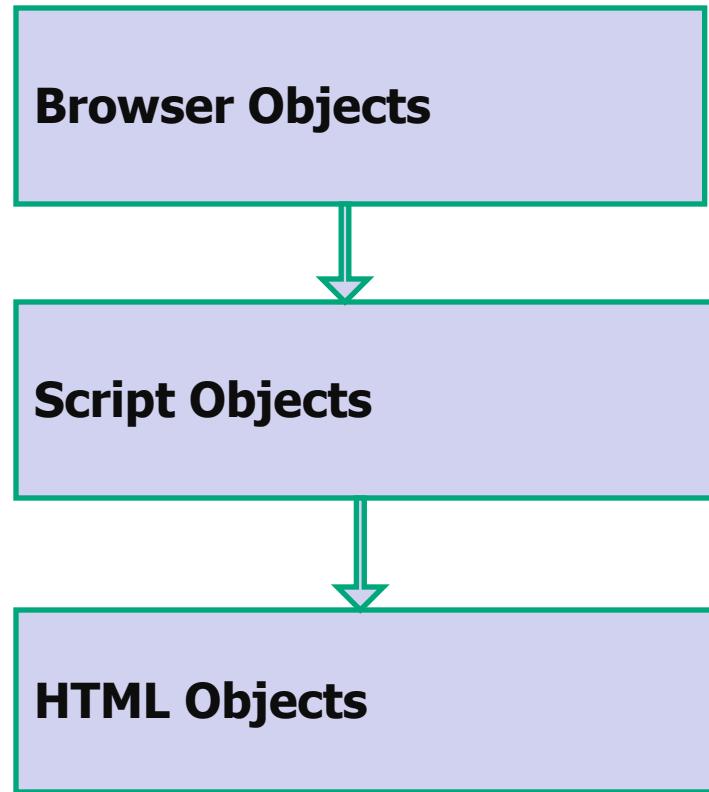


7.1. Giới thiệu về đối tượng

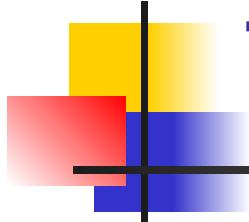
- JavaScript là ngôn ngữ lập trình hướng đối tượng
- Khi tạo trang Web, chúng ta có thể sử dụng các đối tượng được cung cấp bởi browser, JavaScript, HTML
- Truy nhập
 - Thuộc tính: *tên_đối_tượng . tên_thuộc_tính*
 - Phương thức: *tên_đối_tượng . tên_phương_thức()*



7.1. Giới thiệu về đối tượng



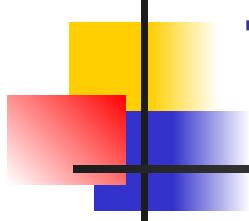
Phân tầng các đối tượng



7.1. Giới thiệu về đối tượng

- Khởi tạo đối tượng

```
tên đối tượng = new  
đối tượng(danh_sách_tham_số)
```



7.1. Giới thiệu về đối tượng

- Cấu trúc for..in và with
- Cú pháp

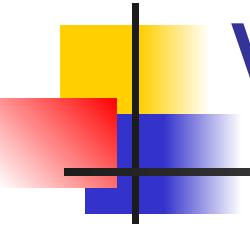
for (variable in object)

lệnh;

và

with (object)

lệnh;



Ví dụ

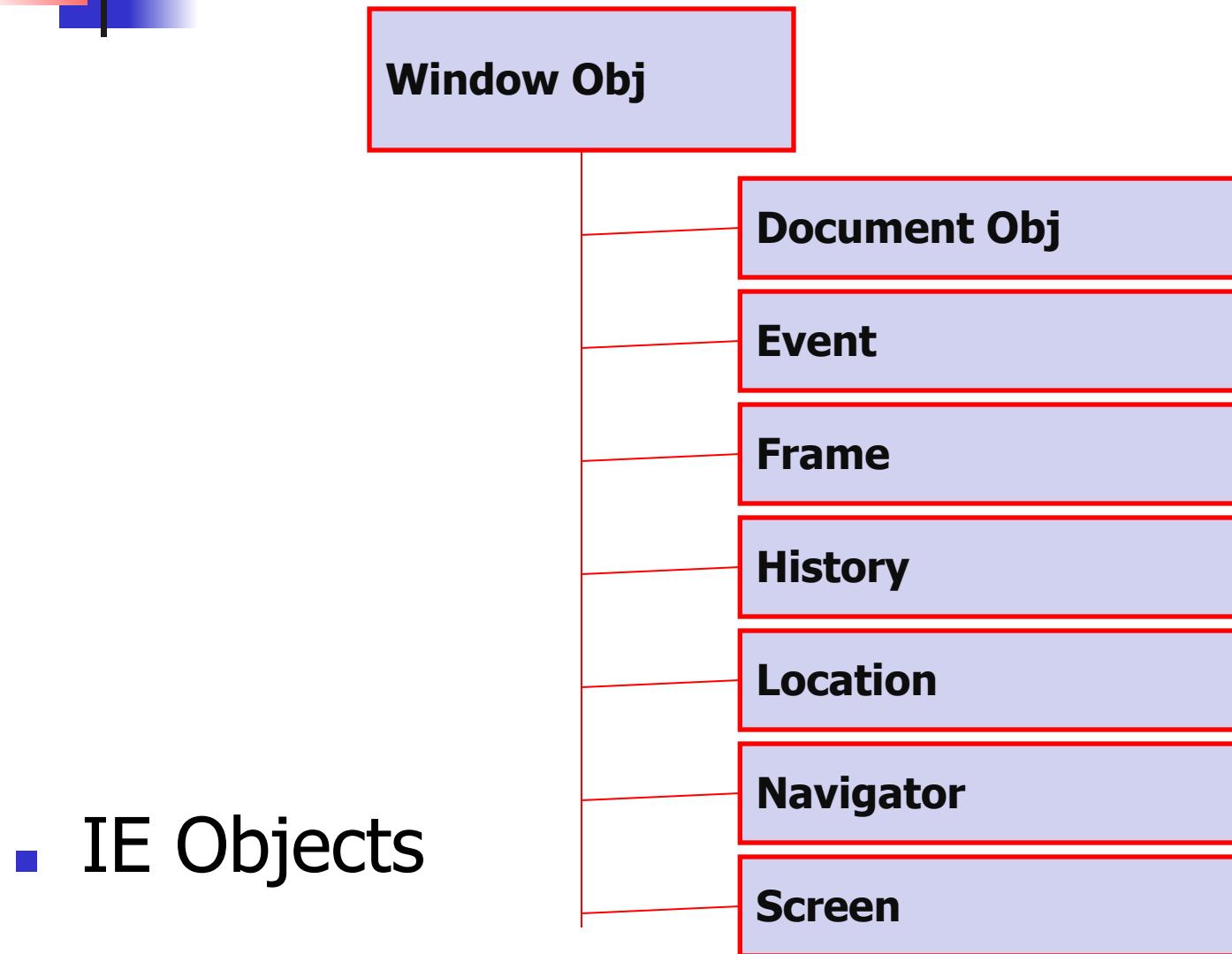
```
<html>
<body>

<script type="text/javascript">
var x;
var mycars = new Array();
mycars[0] = "Saab";
mycars[1] = "Volvo";
mycars[2] = "BMW";

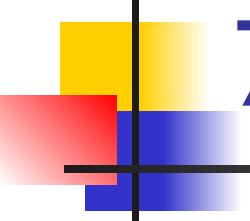
for (x in mycars)
{
    document.write(mycars[x] + "<br />");
}
</script>

</body>
</html>
```

7.2. Đối tượng của browser



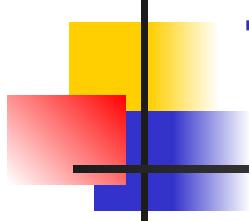
- IE Objects



7.2. Đối tượng của browser

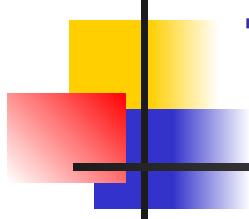
- **Đối tượng Window**

- **Thể hiện cửa sổ của trình duyệt**
- **Được sử dụng để**
 - nhận các thông tin về trạng thái của cửa sổ
 - hiển thị các đối tượng khác
 - truy cập các sự kiện



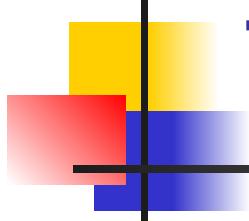
7.2. Đối tượng của browser

- document Obj: biểu diễn tài liệu HTML
- event: biểu diễn trạng thái của sự kiện
- frame: biểu diễn các frame (child windows)
- history: thông tin về URLs đã mở
- location: thông tin về URL hiện tại
- navigator: thông tin về Web browser
- screen: thông tin về màn hình client, khả năng render



7.2. Đối tượng Window

- document Obj: biểu diễn tài liệu HTML
- event: biểu diễn trạng thái của sự kiện
- frame: biểu diễn các frame (child windows)
- history: thông tin về URLs đã mở
- location: thông tin về URL hiện tại
- navigator: thông tin về Web browser
- screen: thông tin về màn hình client, khả năng render



7.2. Đối tượng Window

Ví dụ:

```
window.defaultStatus="String";
```

```
window.status="String";
```

```
window.location="URL";
```

Ta cũng có thể mở một trang web mới bằng lệnh:

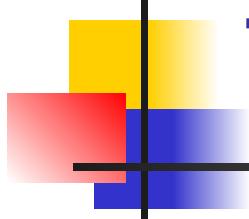
```
window.location.href= "URL";
```

7.2. Đối tượng Window

Phương Thức	Mô tả
alert("msg")	Hiển Thị hộp thoại thông báo
blur()	Di chuyển con trỏ đến cửa sổ hiện hành
clearinterval(ID)	Hủy thời gian đã thiết lập bằng setinterval()
clearTimeout(ID)	Hủy thời gian đã thiết lập bằng setTimeout()
close()	Đóng cửa sổ hiện hành
confirm("msg")	Hiển thị hộp thoại xác nhận với hai nút Cancel và OK
focus()	Đưa con trỏ về cửa sổ hiện hành
MoveBy(x,y)	Di chuyển cửa sổ đến một vị trí mới một đoạn pixel so với cửa sổ hiện hành
MoveTo(x,y)	Di chuyển cửa sổ qua trái và lên trên một đoạn pixel cụ thể so với cửa sổ hiện hành.

7.2. Đối tượng Window

open(URL,"windowname","FeatureList")	URL : đc trang web nạp vào cửa sổ. WindowName: là tên cửa sổ . FeatureList : danh sách các thuộc tính của cửa sổ: toolbars, menu, status
print()	in nội dung trong cửa sổ hiện hành.
prompt("msg","reply")	Hiển thị hộp thoại nhập liệu
setTimeout(func,millisec)	Thiết lập thời gian mili giây để gọi 1 hàm
stop()	Hủy việc download một cửa sổ. Tương tự như việc đóng một cửa sổ trình duyệt.
resizeBy(dx,dy)	Thay đổi kích thước cửa sổ sang phải dx, dưới dy pixel
resizeTo(x,y)	Thay đổi kích thước x, y pixel
scrollBy(dx,dy)	cuộn nội dung sang phải dx, xuống dưới dy pixel
scrollTo(x,y)	cuộn nội dung trên trang đến vị trí x,y



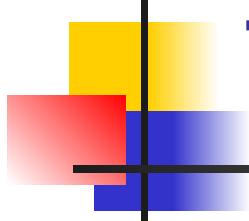
7.2. Đối tượng Window

Ví dụ: Objwindow.close()

- ❖ **Từ khoá self:** trong trường hợp muốn thao tác trên cửa sổ hiện hành ta dùng từ khoá self thay thế cho đối tượng window

Ví dụ : đóng cửa sổ hiện hành:

Self.close() hoặc window.close()



7.2.1. Đối tượng Document

■ Thuộc tính

- Title: title của tài liệu
- bgColor: màu nền của tài liệu
- linkColor: màu liên kết
- vlinkColor: màu liên kết đã mở

■ Phương thức

- clear: xóa nội dung
- write: viết biểu thức HTML

7.2.1. Đối tượng Document

Thuộc tính

Thuộc tính	Mô tả
alinkcolor	Thiết lập hoặc trả về giá trị màu của liên kết đang xem của tài liệu
bgcolor	Thiết lập hoặc trả về giá trị màu nền của tài liệu
cookie	Chứa giá trị các cookies dành cho tài liệu hiện hành
domain	Trả về giá trị tên miền máy chủ chứa document
fgcolor	Thiết lập hoặc trả về giá trị màu chữ của tài liệu
lastmodified	Trả về giá trị ngày giờ cuối cùng mà tài liệu được cập nhật

7.2.1. Đối tượng Document

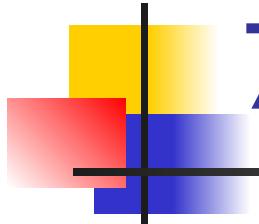
Thuộc tính

linkcolor	Thiết lập hoặc trả về giá trị màu của liên kết trong tài liệu
location	mở một trang web mới
referrer	Returns the URL of the document that loaded the current document
title	Trả về giá trị của tựa đề của tài liệu
url	Trả về đường dẫn củ tài liệu hiện hành
vlinkcolor	Thiết lập hoặc trả về giá trị màu của liên kết đã xem của tài liệu

7.2.1. Đối tượng Document

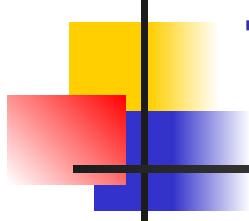
▪ Phương thức

Phương thức	Mô tả
clear()	Xóa tài liệu
close()	Đóng một tài liệu
focus()	Đưa trỏ về một đối tượng trong trang
open(["mimetype"])	Mở một stream để thu thập dữ liệu của các phương thức write và writeln.
write("str")	viết một chuỗi vào một tài liệu
writeln("str")	viết một chuỗi vào một tài liệu và xuống dòng



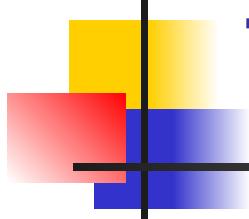
7.2.2. Đối tượng History

- Cung cấp các URLs được mở gần nhất
- Phương thức
 - back: quay lại URL ngay trước
 - go(number): quay lại vị trí xác định trong tài liệu
 - forward: mở URL tiếp theo



7.3. Đối tượng JavaScript

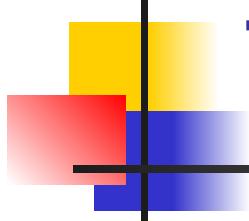
- 7.3.1. Đối tượng String
- 7.3.2. Đối tượng Date
- 7.3.3. Đối tượng Math
- 7.3.4. Đối tượng Array



7.3.1. Đối tượng String

- Mỗi xâu trong JavaScript là một đối tượng, gồm các thuộc tính và phương thức
- Được sử dụng để làm việc với xâu văn bản
- Khởi tạo

```
var tên_xâu = "nội_dung_xâu";
tên_xâu = "nội_dung_xâu";
var tên_xâu = String("nội_dung_xâu");
```



7.3.1. Đối tượng String

- Thuộc tính
 - length: độ dài xâu
 - Các ký tự trong chuỗi được đánh chỉ số từ 0 đến length-1.
 - Cách tham chiếu đến thuộc tính length của đối tượng String:
 - `StringLength=stringVariable.length`
 - `StringLength="This is a string".length`

7.3.1. Đối tượng String

Phương thức

Phương thức	Mô tả	Ví dụ
anchor("name")	Tham số name là thuộc tính NAME của tag <A>.	str.anchor("name")
big()	Trả về một chuỗi đặt trong cặp thẻ <big>	str.big()
Bold()	Trả về một chuỗi in đậm	str.bold()
charAt(index)	Trả về ký tự thứ index trong chuỗi. index từ 0 đến str.length-1	str.charAt(0)
concat()	Nối chuỗi	
fontcolor()	Trả về một chuỗi với màu đã được xác lập.	str.fontcolor("red")
fontsize()	kích thước font	str.fontsize("5")

7.3.1. Đối tượng String

Phương thức

indexOf("str", [fromindex])	Trả về vị trí của đầu tiên được tìm thấy của chuỗi “str” bắt đầu tìm từ vị trí fromindex . Nếu không có fromindex thì tìm từ vị trí 0. Nếu không tìm thấy thì hàm trả về giá trị -1
italics()	Trả về một chuỗi in nghiêng
lastIndexOf("str")	Trả về vị trí cuối cùng được tìm thấy chuỗi “str”, tìm từ trái qua phải . Nếu không tìm thấy thì hàm trả về giá trị -1
link()	Trả về một chuỗi liên kết
match()	Tương tự như hàm indexOf và lastIndexOf, nhưng phương thức này trả về một chuỗi cụ thể nếu không tìm thấy thì trả về giá trị "null".
replace()	Thay thế một chuỗi bằng một chuỗi mới

7.3.1. Đối tượng String

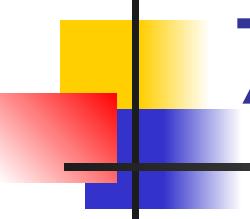
Phương thức

search('str')	Trả về giá trị là vị trí chuỗi con được tìm thấy trong chuỗi cha, nếu không tìm thấy thì trả về giá trị -1	
slice(index)	Trả về một chuỗi con được cắt từ chuỗi mẹ tại vị trí index	
small()	Trả về một chuỗi có kích thước nhỏ hơn	
strike()	Trả về một chuỗi được gạnh ngang qua thân chuỗi	
sub()	Trả về một chuỗi kiểu subscript	Str.sub()

7.3.1. Đối tượng String

Phương thức

substr(start,length)	Trả về chuỗi con bắt đầu từ vị trí start và có chiều dài length. nếu không có start xem như start=0	Str.substr(0,2) ="Th"
substring(Start,end)	Tách ra một chuỗi con từ một chuỗi. Bắt đầu từ chỉ số start đến end. Nếu Start<end, chuỗi trả về từ start đến end-1 Nếu end<start, chuỗi trả về từ end đến start Nếu start=end chuỗi trả về là null.	
sup()	Trả về chuỗi kiểu superscript	
toLowerCase()	Chuyển chuỗi thành chữ thường	
toUpperCase()	Chuyển chuỗi thành chữ hoa	



7.3.2. Đối tượng Date

- Chứa các thông tin về ngày, giờ
- Lưu trữ thời điểm thông qua số lượng ms từ ngày 1/1/1970 tại 00:00:00
- Khởi tạo

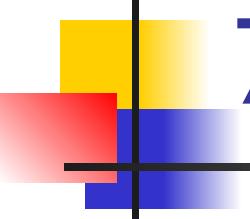
`tên_đối_tượng = new Date(tham_số);`

`var variableName= new Date("month, day, year, hours : minutes : seconds");`

- Ví dụ

`today = new Date();`

`dateobj = new Date("Octorber 13,2009,10:10:10");`



7.3.2. Đối tượng Date

- Phương thức chia làm:

- get: lấy giá trị ngày và giờ

- set: thiết lập giá trị ngày và giờ

- Bao gồm: Hours, Minutes, Seconds, Time, Year, Month, Date

- Giá trị

- seconds và minutes: 0-59
 - hours: 0-23
 - day: 0-6
 - date: 0-31
 - months: 0-11

7.3.2. Đối tượng Date

Phương thức

Phương thức	Mô tả
Date()	trả về đối tượng date
getDate()	Trả về giá trị ngày (số nguyên từ 1-31) trong tháng
getDay()	Trả về giá trị ngày trong tuần (số nguyên từ 0-6 Sunday=0)
getMonth()	Trả về tháng trong năm (from 0-11. 0=January, 1=February)
getFullYear()	Trả về giá trị năm (bốn số)
getYear()	Trả về giá trị năm (hai số)
getHours()	Trả về giờ của hệ thống (từ 0-23)

7.3.2. Đối tượng Date

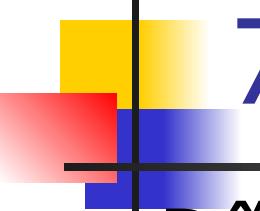
Phương thức

getMinutes()	Trả về phút của hệ thống (từ 0-59)
getSeconds()	Trả về giây của hệ thống (từ 0-59)
getMilliseconds()	Trả về giá trị millisecond from 0-999)
setFullYear()	Thiết lập lại năm cho ngày hệ thống (4 số)
setHours()	Thiết lập lại giờ cho hệ thống (từ 0-24)
setMinutes()	Thiết lập lại phút cho hệ thống (từ 0-59)
setMonth()	Thiết lập lại tháng cho hệ thống (từ 0-11)
setSeconds()	Thiết lập lại giây cho hệ thống (from 0-59)

7.3.2. Đối tượng Date

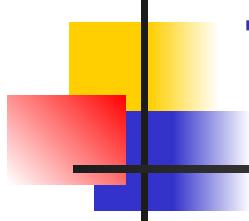
Ví dụ:

```
<html>
<head>
<script language="javascript">
function Ngay()
{
    var day=new Date();
    var w,m,d,y;
    w=day.getDay(); d=day.getDate();
    m=day.getMonth()+1; y=day.getFullYear();
    document.write("Hom nay:" + " ngay " + d + " thang " + m + " nam
" +y);
}
</script></head>
<body>
<script language="Javascript">Ngay()</script>
</body></html>
```



7.3.3. Đối tượng Math

- Đối tượng Math() cung cấp các hàm và các phương thức cần thiết để thực hiện các phép toán số học.
- Không cần phải tạo đối tượng Math() mà chúng ta có thể sử dụng trực tiếp đối tượng này
- Cú pháp chung:
Math.method([value])



7.3.3. Đối tượng Math

- Thuộc tính
 - PI: số Pi, xấp xỉ 3.1415
 - E: số e, xấp xỉ 2.718

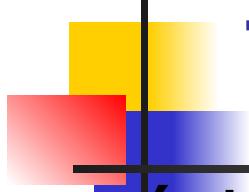
7.3.3. Đối tượng Math

Phương thức

Phương thức	Mô tả
abs(x)	Trả về giá trị tuyệt đối của biến x
acos(x)	Trả về giá trị arccosine của x
ceil(x)	Trả về số nguyên lớn hơn hoặc bằng x
floor(x)	Trả về số nguyên nhỏ hơn hoặc bằng x
log(x)	Trả về giá trị log của x
max(x,y)	Trả về giá trị lớn nhất trong hai số x và y
min(x,y)	Trả về giá trị nhỏ nhất trong hai số x và y
pow(x,y)	Trả về giá trị x lũy thừa y
random()	Trả về giá trị một số ngẫu nhiên từ 0 đến 1
round(x)	Làm tròn số x
sqrt(x)	Trả về giá trị căn bậc 2 của x

7.3.4. Đối tượng Array

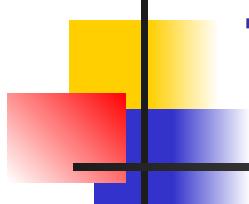
- **Đối tượng Array()**: Dùng để lưu trữ nhiều giá trị với cùng một tên gọi.
- Trong Javascript đối tượng mảng có thể chứa các thành phần mang kiểu giá trị khác nhau.
- Một mảng có n phần tử được đánh chỉ số từ 0 đến n-1.
- Dùng từ khóa **new** để khởi tạo một mảng
var Variable = new Array(size)



7.3.4. Đối tượng Array

Ví dụ:

```
<script>
    var arr= new Array()
    arr[0]= "thu hai";
    arr[1]= "Thu ba";
    arr[2]= "Thu tu";
    arr[3]= "Thu nam";
    arr[4]= "Thu sau";
    arr[5]= "Thu bay";
    for(i=0; i<=5;i++)
        document.write(arr[i]+ "<br>")
</script>
```



7.3.4. Đối tượng Array

- Khởi tạo

```
tên_mảng = new Array(kích_thước_mảng);
```

hoặc

```
tên_mảng = new Array(ds_phần_tử);
```

Ví dụ: mybook = new Array(2);

```
mybook[0] = "Access 2000";
```

```
mybook[1] = "Programming Techniques";
```

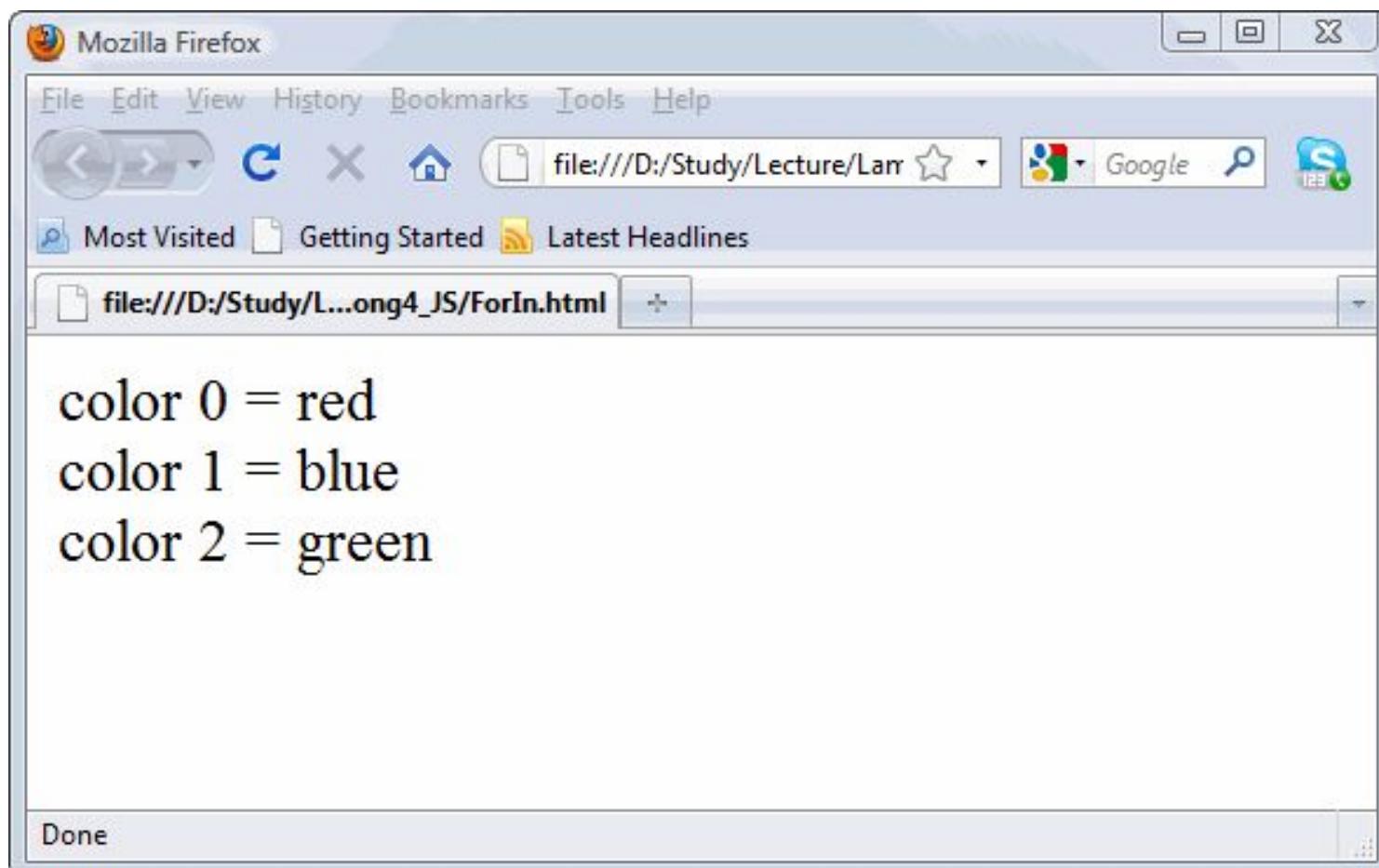
- Phương thức

- reverse: đảo ngược các phần tử trong mảng
- pop: trả về phần tử cuối cùng của mảng, rồi xóa nó
- push: thêm một phần tử vào cuối mảng

7.3.3. Đối tượng Array

```
<head>
    <script language="javascript">
color = new Array("red", "blue", "green");
var scolor;
for(var elem in color) {
    scolor = "color " + elem + " = " +
        color[elem] + "<br>";
    document.write(scolor);
}
</script>
</head>
```

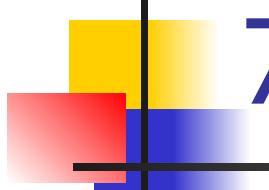
7.3.3. Đối tượng Array



7.3.4. Đối tượng Array

- Thuộc tính:
 - **length**: để xác định số phần tử trong mảng
- Phương thức:

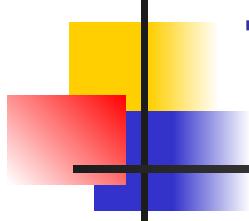
Phương thức	Ý nghĩa	Ví dụ
concat()	Dùng để nối 2 mảng	a=a.concat(b)
join(separator)	để ghép các phần tử trong mảng lại với nhau cách nhau bởi dấu separator	a=a.join(“+”)
slice(start,end)	Dùng tách một mảng bắt đầu từ vtrí start đến vtrí end-1.	str=a.slice(i,j)
reverse()	Dùng để đảo ngược mảng	a.reverse()
valueOf()	Dùng để lấy tất cả các đối tượng trong mảng	a.valueOf()
pop()	Lấy phần tử cuối của mảng	
push()	Thêm 1 hoặc nhiều phần tử vào cuối mảng	
Shift()	lấy phần tử và trả về phần tử đầu tiên của mảng	
Sort()	sắp xếp các phần tử của mảng	



7.3.4. Đối tượng Array

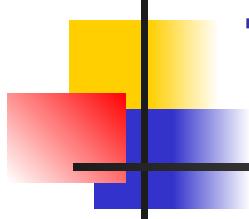
- Ví dụ

```
<script type="text/javascript">
    var arrName = new Array(3)
    arrName [0] = "Jani"
    arrName [1] = "Tove"
    arrName [2] = "Hege"
    document.write(arrName.length + "<br>")
    document.write(arrName.join(".") + "<br>")
    document.write(arrName.reverse() + "<br>")
    document.write(arrName.sort() + "<br>")
    document.write(arrName.push("Ola","Jon") + "<br>")
    document.write(arrName.pop() + "<br>")
    document.write(arrName.shift() + "<br>")
</script>
```



7.3. Đối tượng JavaScript

- 7.3.1. Đối tượng String
- 7.3.2. Đối tượng Date
- 7.3.3. Đối tượng Math
- 7.3.4. Đối tượng Array



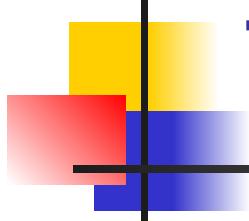
7.3.1. Đối tượng String

- Được sử dụng để làm việc với xâu văn bản
- Khởi tạo

```
var tên_xâu = "nội_dung_xâu";
```

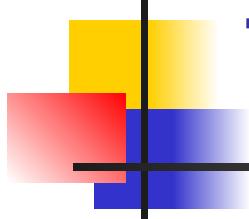
```
тên_xâu="nội_dung_xâu";
```

```
var tên_xâu = String("nội_dung_xâu");
```



7.3.1. Đối tượng String

- Thuộc tính
 - length: độ dài xâu
- Phương thức
 - big: tăng kích thước xâu
 - small: giảm kích thước xâu
 - toLowerCase: chuyển thành chữ thường
 - toUpperCase: chuyển thành chữ hoa
 - fontcolor: xác định màu chữ
 - charAt: vị trí kí tự trong xâu



7.3.2. Đối tượng Date

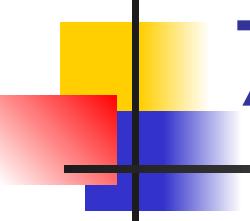
- Chứa các thông tin về ngày, giờ
- Lưu trữ thời điểm thông qua số lượng ms từ ngày 1/1/1900 tại 00:00:00
- Khởi tạo

tên_đối_tượng = new Date(tham_số);

- Ví dụ

```
today = new Date();
```

```
dateobj = new Date("Octorber  
13,2009,10:10:10");
```



7.3.2. Đối tượng Date

- Phương thức chia làm:

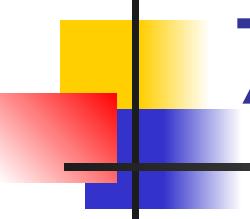
get: lấy giá trị ngày và giờ

set: thiết lập giá trị ngày và giờ

Bao gồm: Hours, Minutes, Seconds, Time, Year, Month, Date

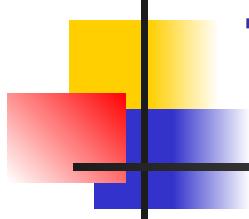
- Giá trị

- seconds và minutes: 0-59
- hours: 0-23
- day: 0-6
- date: 0-31
- months: 0-11



7.3.3. Đối tượng Math

- Thuộc tính
 - PI: số Pi, xấp xỉ 3.1415
 - E: số e, xấp xỉ 2.718
- Phương thức
 - abs: lấy giá trị tuyệt đối
 - sin, cos, tan
 - min (number1, number2)
 - max (number1, number2)
 - sqrt
 - pow



7.3.4. Đối tượng Array

- Khởi tạo

```
tên_mảng = new Array(kích_thước_mảng);
```

hoặc

```
tên_mảng = new Array(ds_phần_tử);
```

Ví dụ: mybook = new Array(2);

```
mybook[0] = "Access 2000";
```

```
mybook[1] = "Programming Techniques";
```

- Phương thức

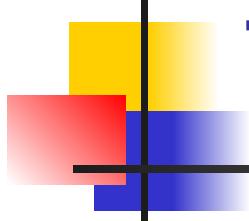
- reverse: đảo ngược các phần tử trong mảng
- pop: trả về phần tử cuối cùng của mảng, rồi xóa nó
- push: thêm một phần tử vào cuối mảng

7.3.4. Đối tượng Array

```
<head>
  <script language="javascript">
    color = new Array("red", "blue", "green");
    var scolor;
    for(var elem in color) {
      scolor = "color " + elem + " = " +
        color[elem] + "<br>";
      document.write(scolor);
    }
  </script>
</head>
```

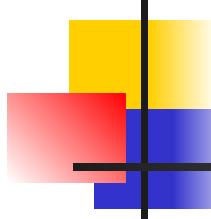
7.3.4. Đối tượng Array





7.4. Đối tượng HTML

- 7.4.1. Đối tượng Form
- 7.4.2. Đối tượng TextField
- 7.4.3. Đối tượng Checkbox



Đối tượng Document - DOM

- **getElementById (id1)**

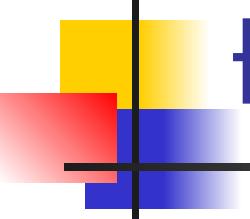
Trả về node có giá trị thuộc tính id = id1

Ví dụ:

```
//<p id="id1" >  
// some text  
//</p>
```

Text Node

```
var node = document.getElementById("id1");  
var nodeName = node.nodeName; // p  
var nodeType = node.nodeType; // 1  
var nodeValue = node.nodeValue; // null  
var text = node.innerText; // some text
```



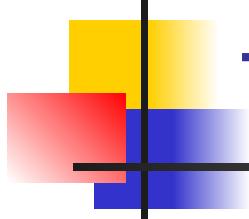
Đối tượng Document - DOM

- createElement (nodeName)

Cho phép tạo ra 1 node HTML mới tùy theo đối số nodeName đầu vào

Ví dụ:

```
var imgNode = document.createElement("img");  
imgNode.src = "images/test.gif";  
  
// 
```

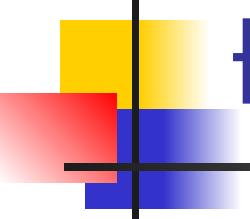


Đối tượng Document - DOM

- **createTextNode (content)**

Ví dụ:

```
var textNode = document.createTextNode("New  
text");           var pNode =  
document.createElement("p");  
pNode.appendChild(textNode);  
  
// <p>New text</p>
```



Đối tượng Document - DOM

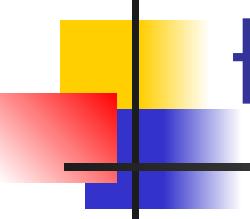
- **appendChild (newNode)**

Chèn node mới **newNode** vào cuối danh sách các node con của một node.

Ví dụ:

```
//<p id="id1" >  
// some text  
//</p>  
  
var pNode = document.getElementById("id1");  
var imgNode = document.createElement("img");  
imgNode.src = "images/test.gif";  
pNode.appendChild(imgNode);
```

```
//<p id="id1" >  
// some text  
//</p>
```



Đối tượng Document - DOM

■ innerHTML

Chỉ định nội dung HTML bên trong một node.

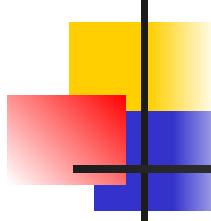
ví dụ:

```
//<p id="para1" >  
// some text  
//</p>
```

```
var theElement = document.getElementById("para1");  
theElement.innerHTML = "Some <b> new </b> text";
```

// Kết quả :

```
// <p id="para1" >  
// Some <b> new </b> text  
// </p>
```



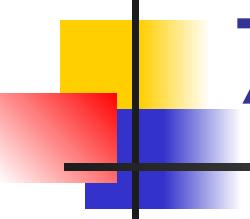
Đối tượng Document - DOM

■ innerText

Tương tự innerHTML, tuy nhiên bất kỳ nội dung nào đưa vào cũng được xem như là text hơn là các thẻ HTML.

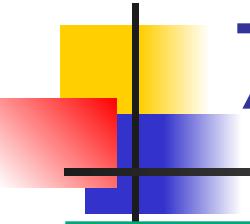
ví dụ:

```
var theElement = document.getElementById("para1");
theElement.innerText = "Some <b> new </b> text";
// Kết quả hiển thị trên trình duyệt
// bên trong thẻ p: "Some <b> new </b> text"
```



7.4.1. Đối tượng Form

- Form được sử dụng để tương tác với người sử dụng
- Thuộc tính
 - ACTION: xác định nơi xử lý dữ liệu khi kết thúc quá trình nhập dữ liệu (submit)
 - METHOD: cách thức gửi dữ liệu đến Server (POST/GET)
 - POST: gửi thông tin đến server
 - GET: thông tin truyền đi sẽ hiển thị trên địa chỉ trình duyệt.



7.4.1. Đối tượng Form

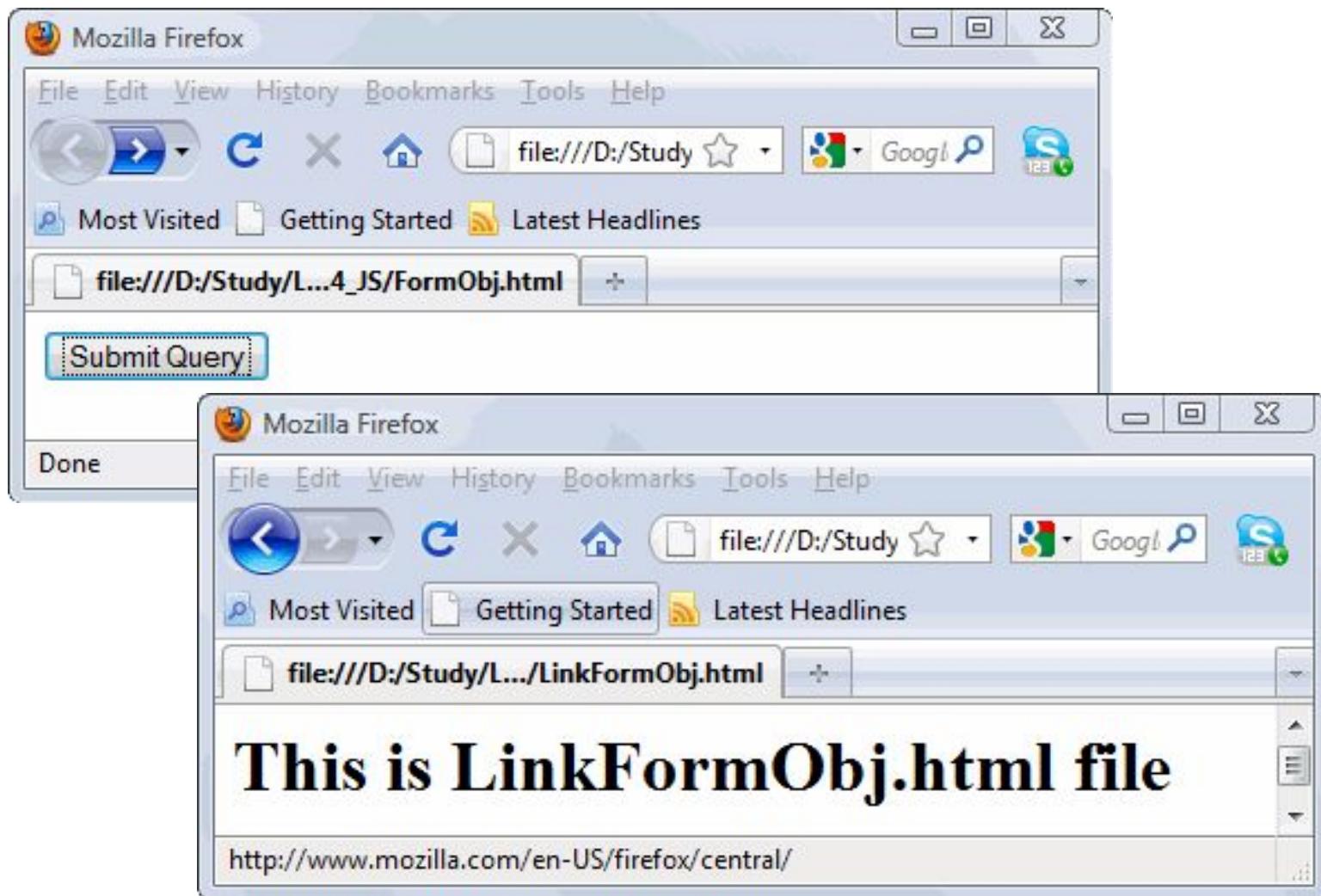
File: LinkFormObj.html

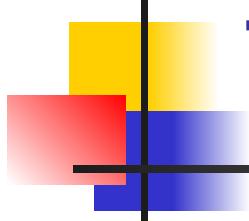
```
<html>
  <body>
    <H1>This is LinkFormObj.html file</H1>
  </body>
</html>
```

File: FormObj.html

```
<html>
  <body>
    <form action="LinkFormObj.html">
      <input type="submit">
    </form>
  </body>
</html>
```

7.4.1. Đối tượng Form





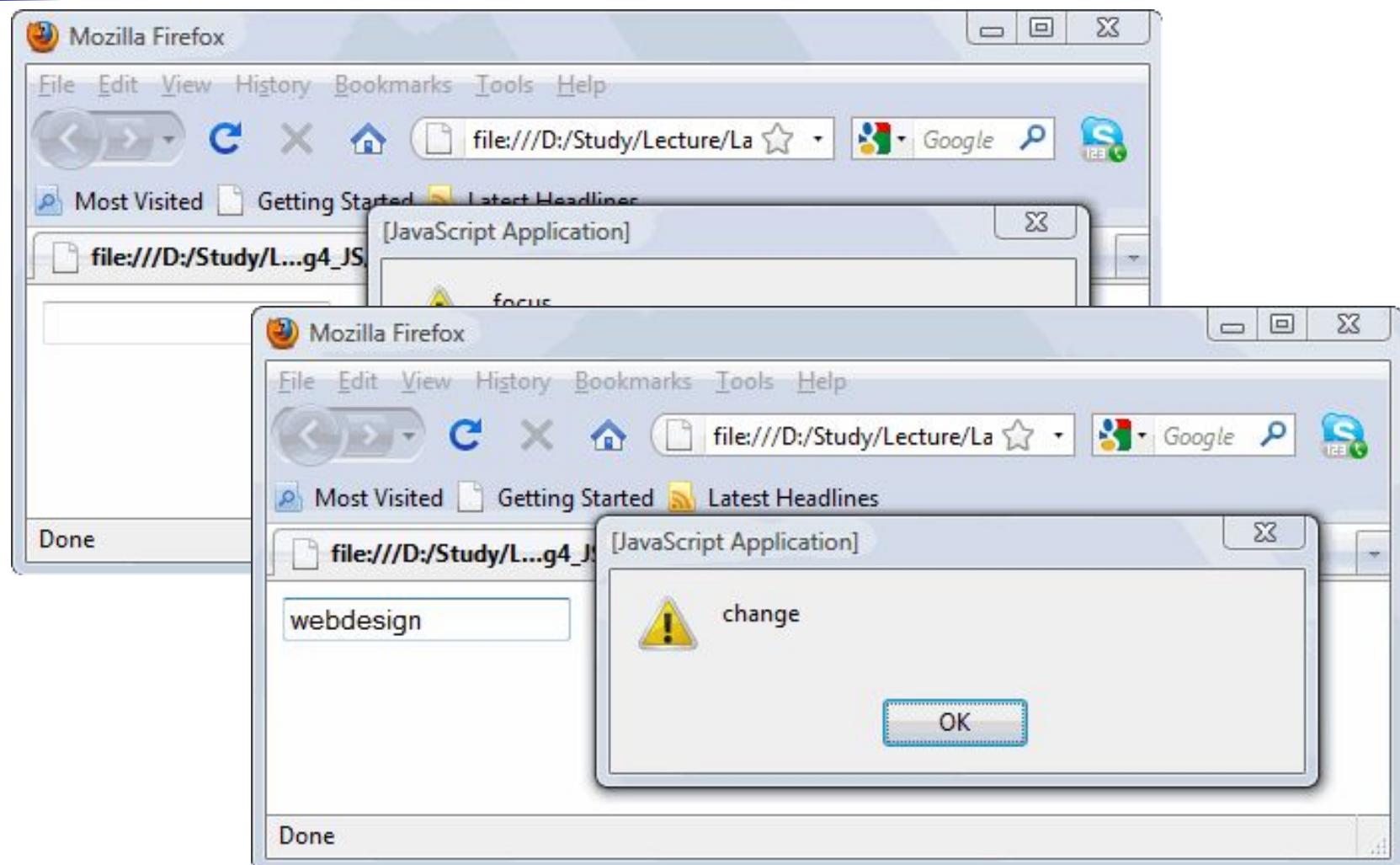
7.4.2. Đối tượng TextField

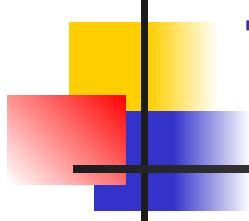
- Các sự kiện liên quan
 - onFocus: khi người sử dụng click chuột vào bên trong đối tượng
 - onBlur: khi người sử dụng click chuột ra bên ngoài hoặc sử dụng tab
 - onChange: khi người sử dụng thay đổi nội dung textfield và di chuyển ra ngoài.

7.4.2. Đối tượng TextField

```
<html>
    <head>
        <script language="javascript">
            function show(content) {
                alert(content);
            }
        </script>
    </head>
    <body>
        <form name="myfrm">
<input type="text onFocus="show ('focus') ; "
       onChange="show ('change') ; ">
        </form>
    </body>
</html>
```

7.4.2. Đối tượng TextField



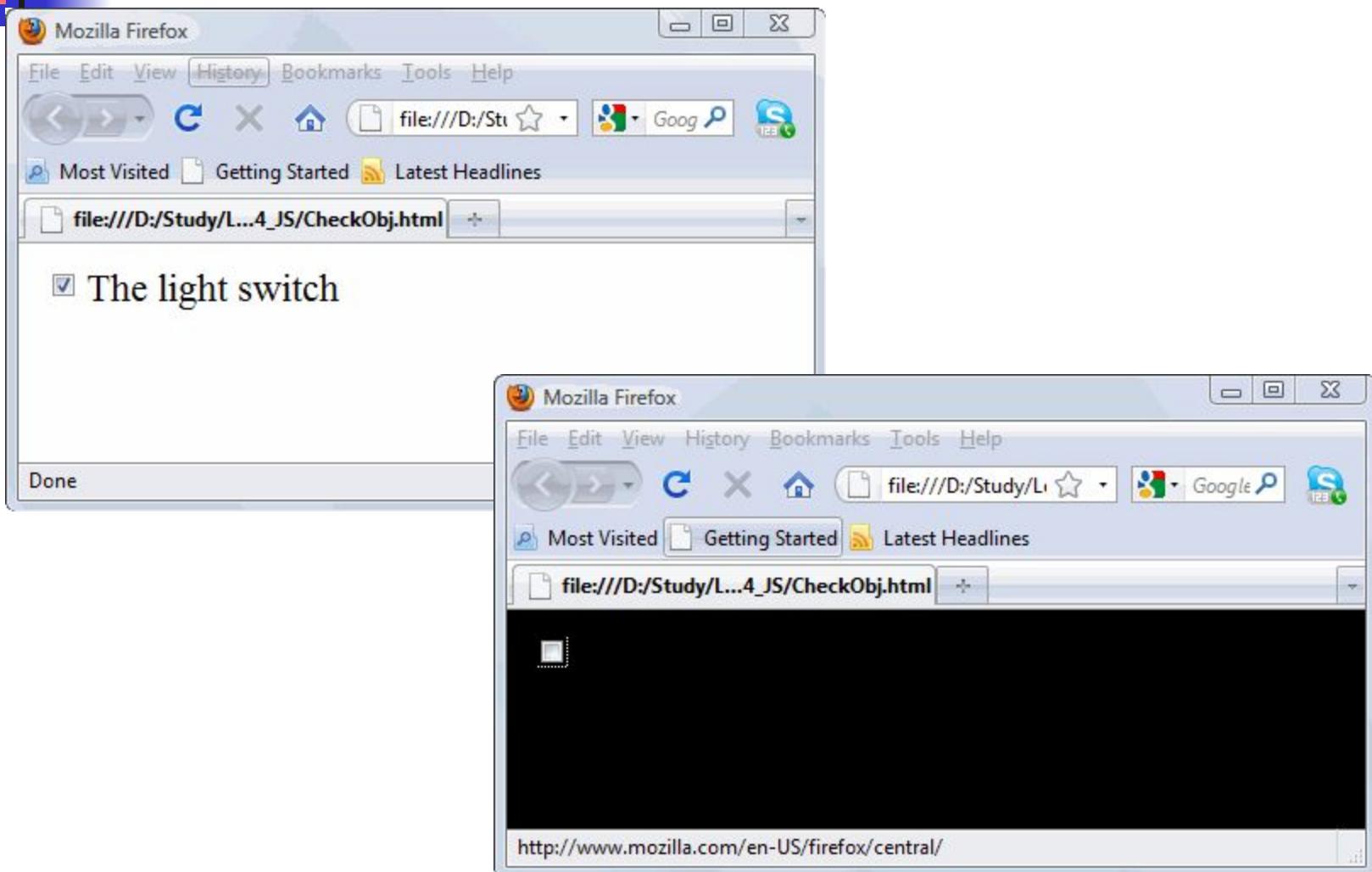


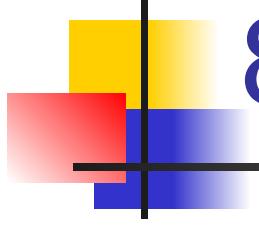
7.4.3. Đối tượng Checkbox

- Các sự kiện liên quan Checkbox, Button
 - onClick

```
<html>
  <head>
    <script language="javascript">
      function switchlight() {
        var check = window.document.frmlight.chklight;
        if (check.checked)
          document.bgColor = 'white';
        else
          document.bgColor='black';
      }
    </script>
  </head>
  <body bgcolor="white">
    <form name = frmlight>
      <input type="checkbox" name = "chklight"
onClick="switchlight();">The light switch
    </form>
  </body>
</html>
```

7.4.3. Đối tượng Checkbox





8. Trắc nghiệm

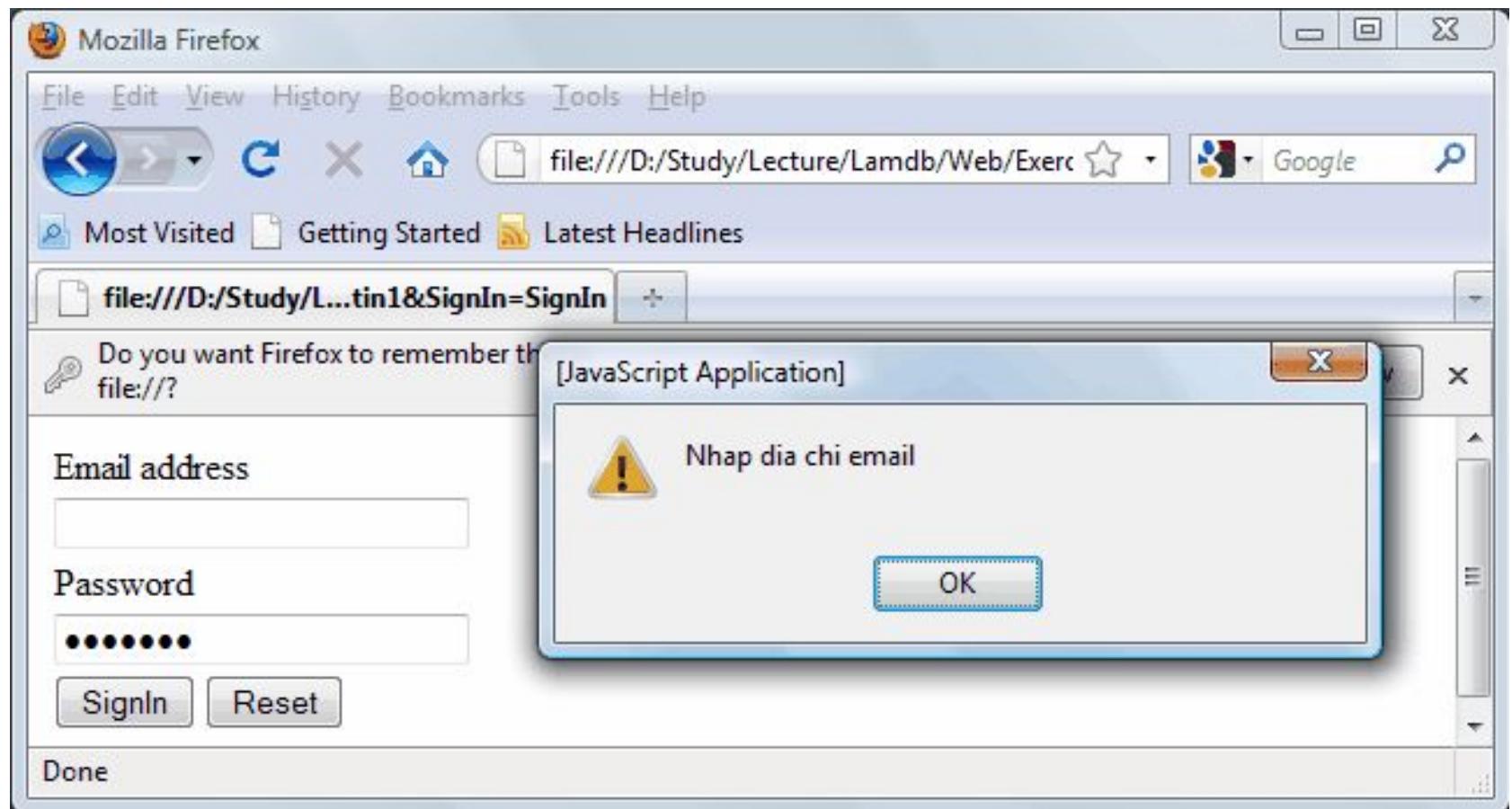
- Nguồn

www.w3schools.com

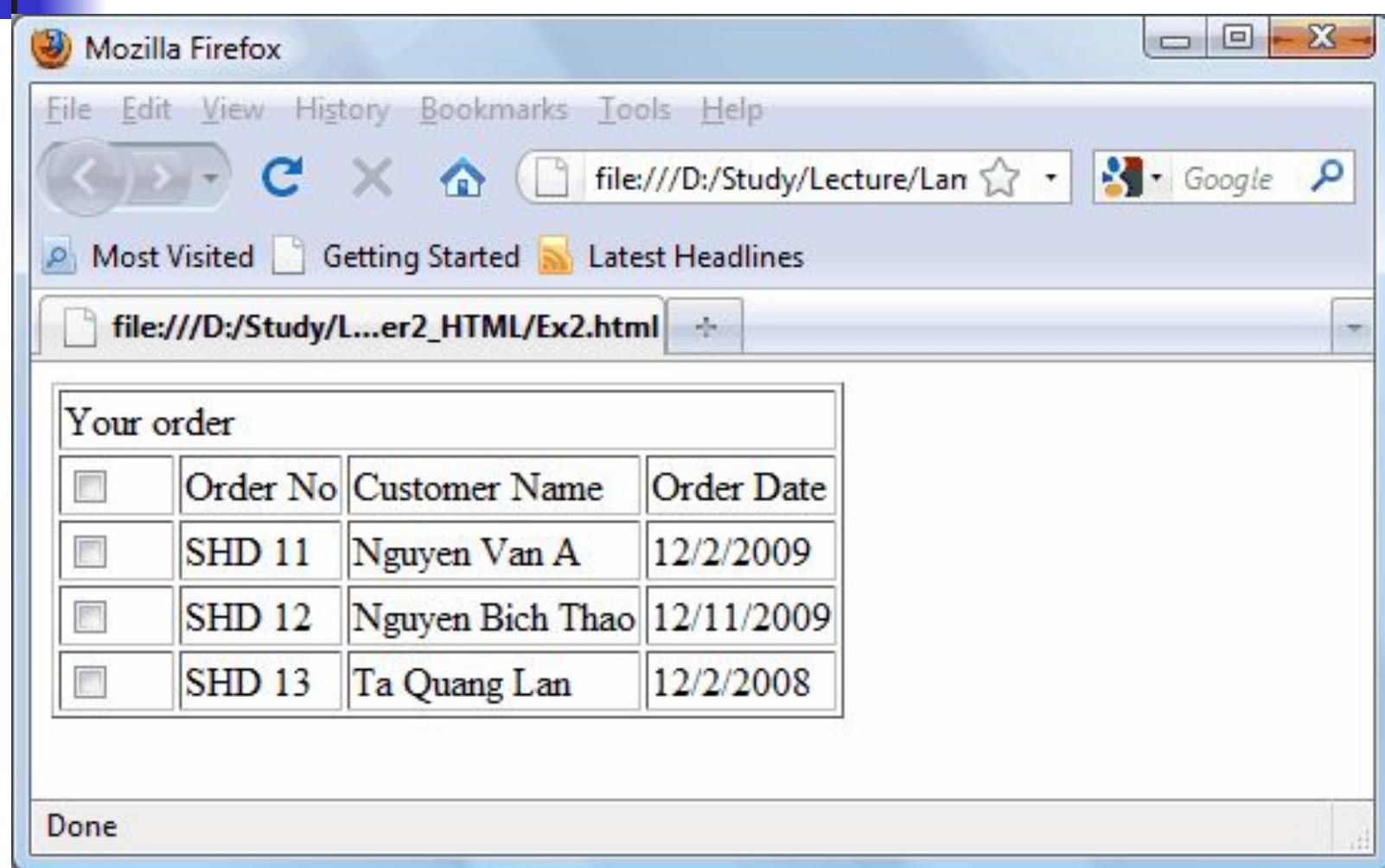
- File

Quiz.doc

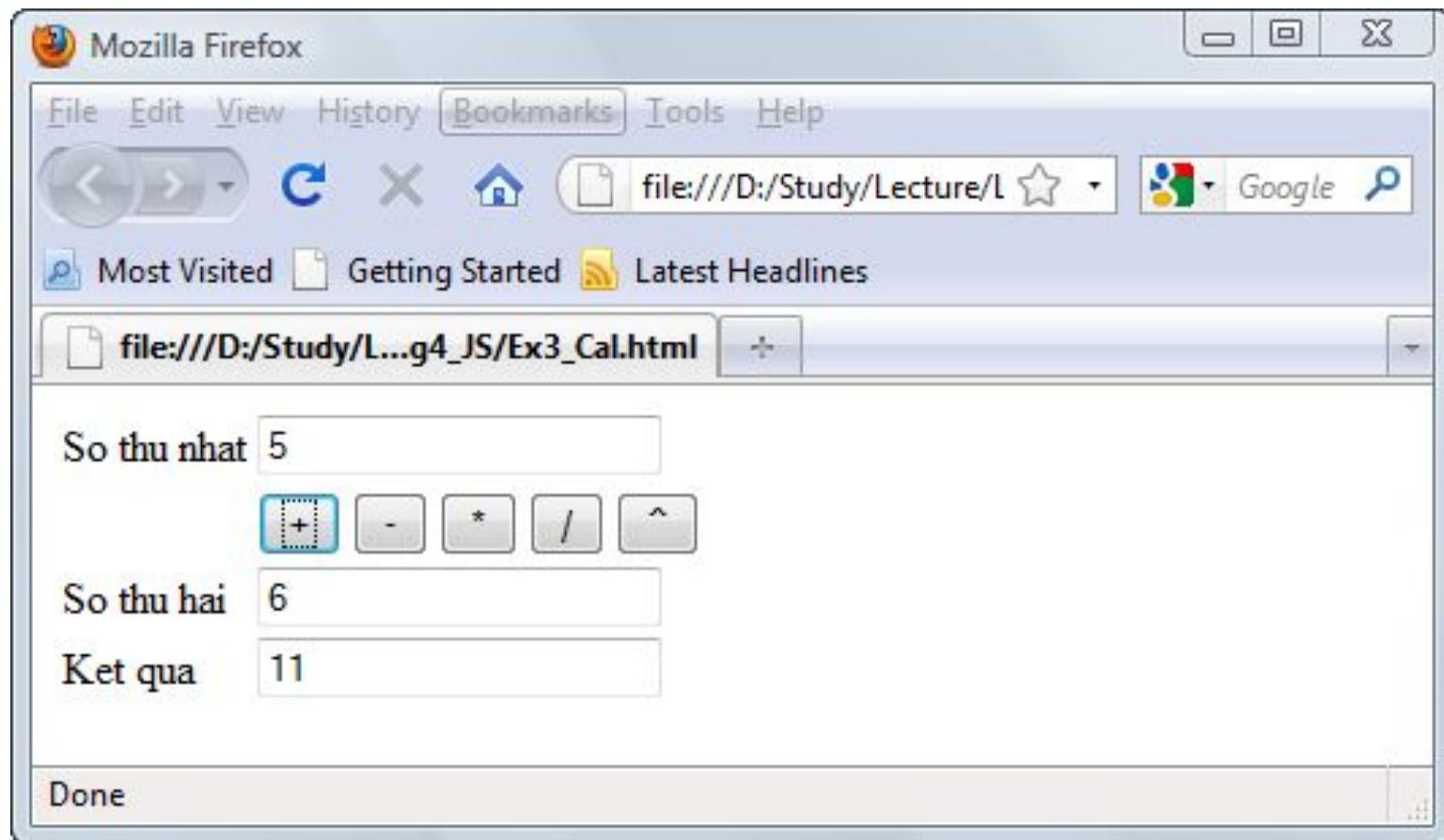
9. Thực hành



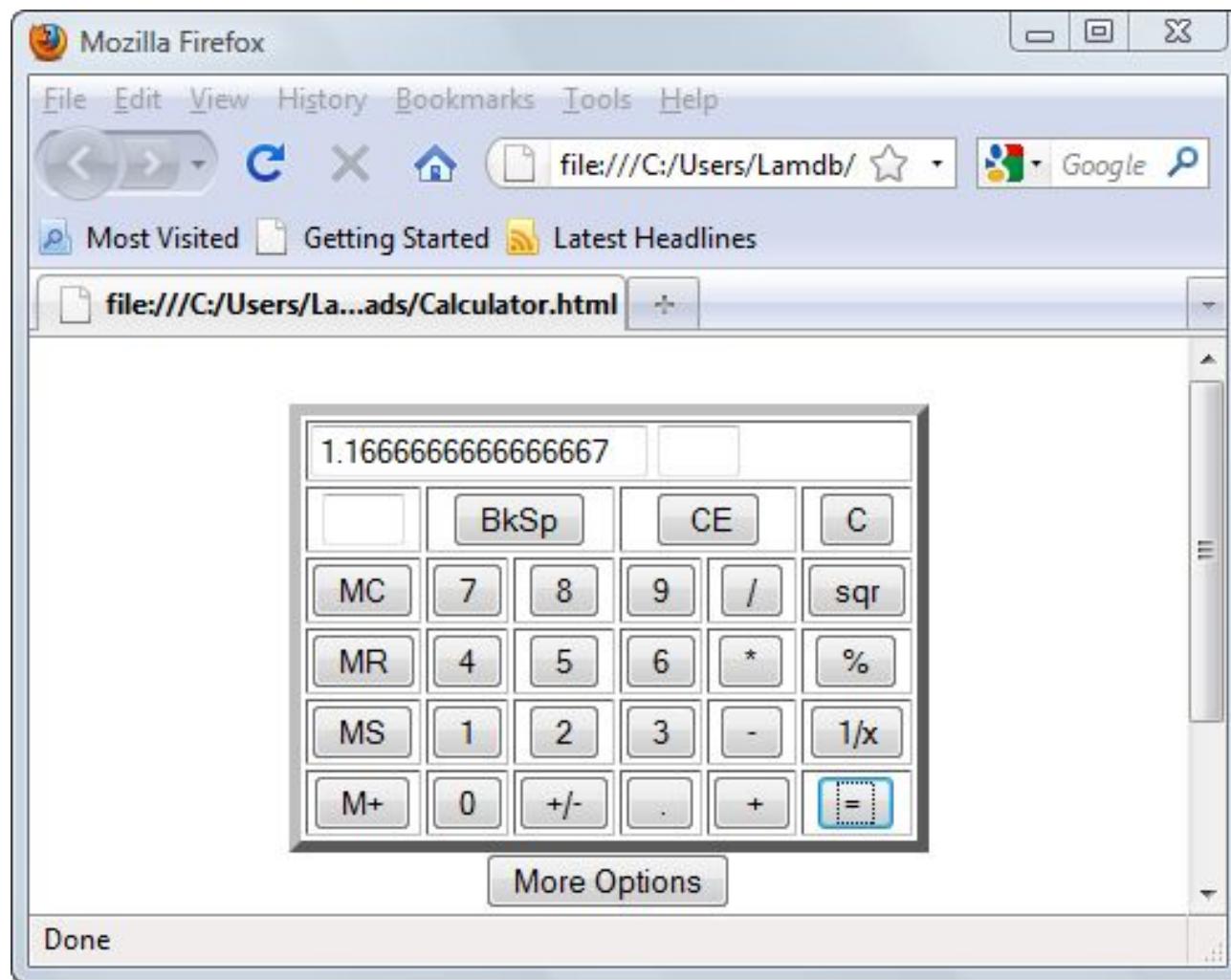
9. Thực hành



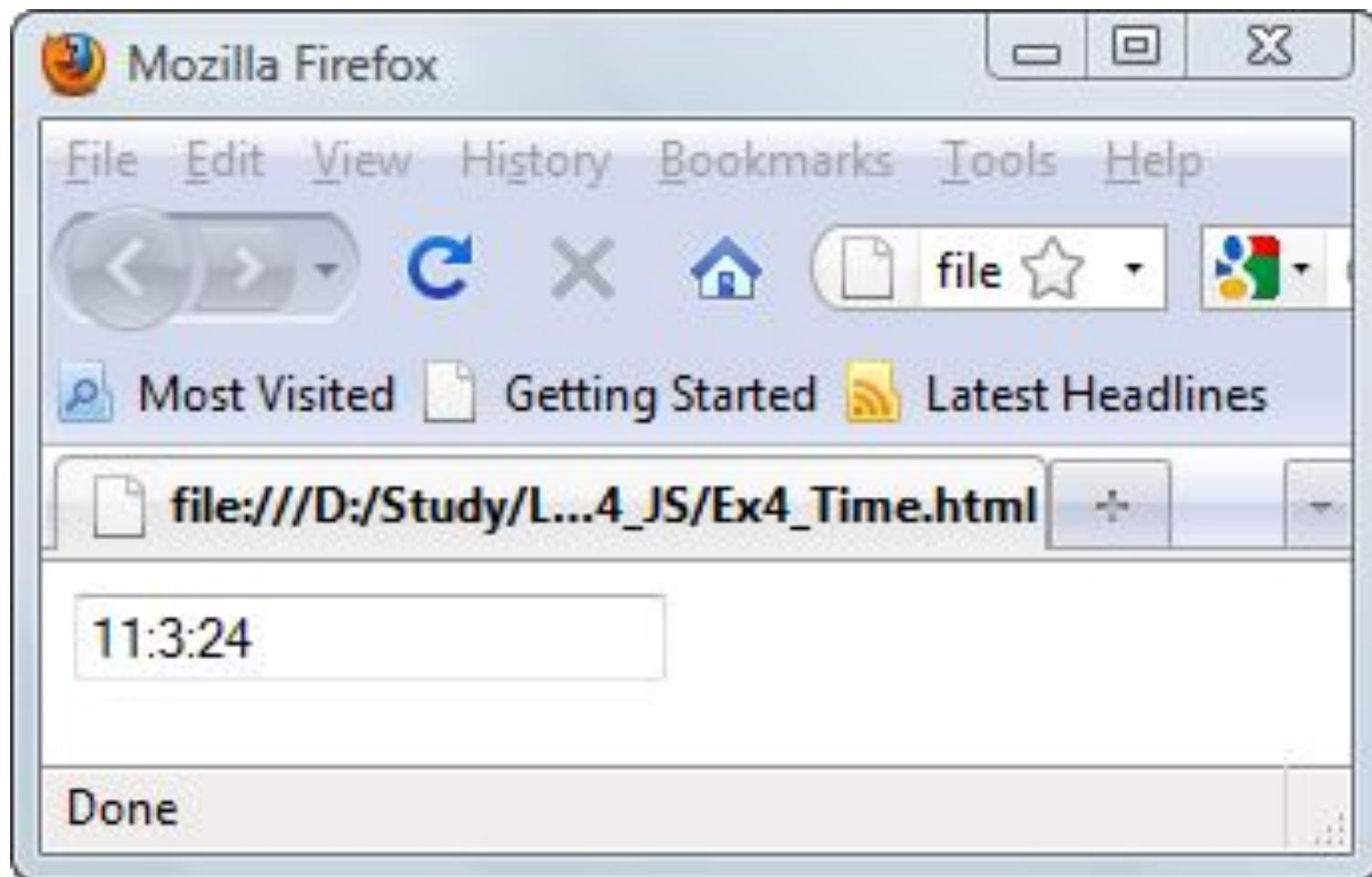
9. Thực hành



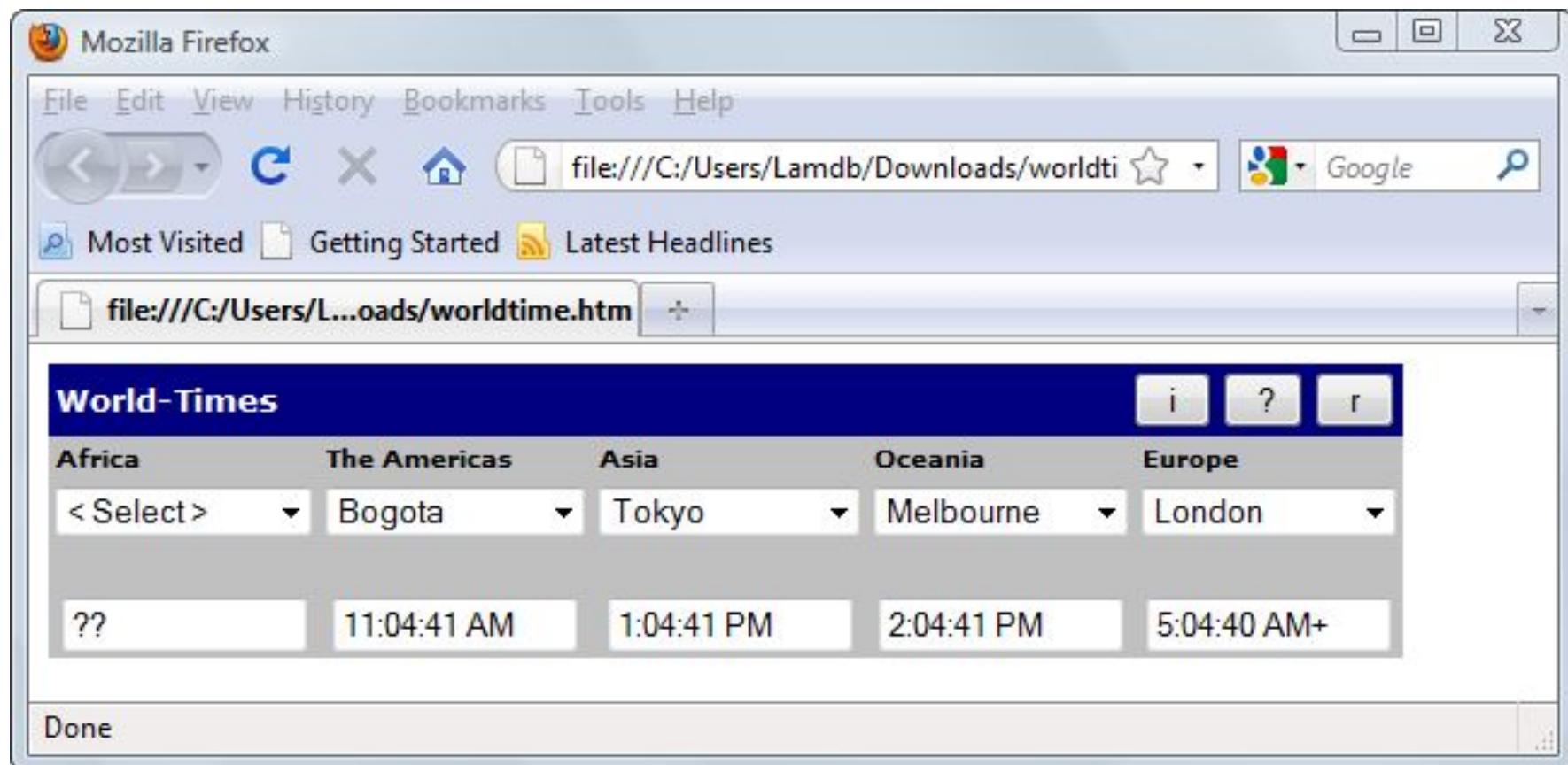
9. Thực hành

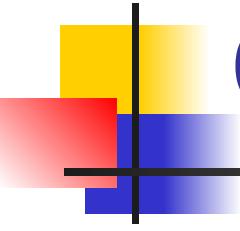


9. Thực hành



9. Thực hành





Câu hỏi

