



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

LẬP TRÌNH WEB HƯỚNG JAVA

Bài 04c: DHTML



Mục đích

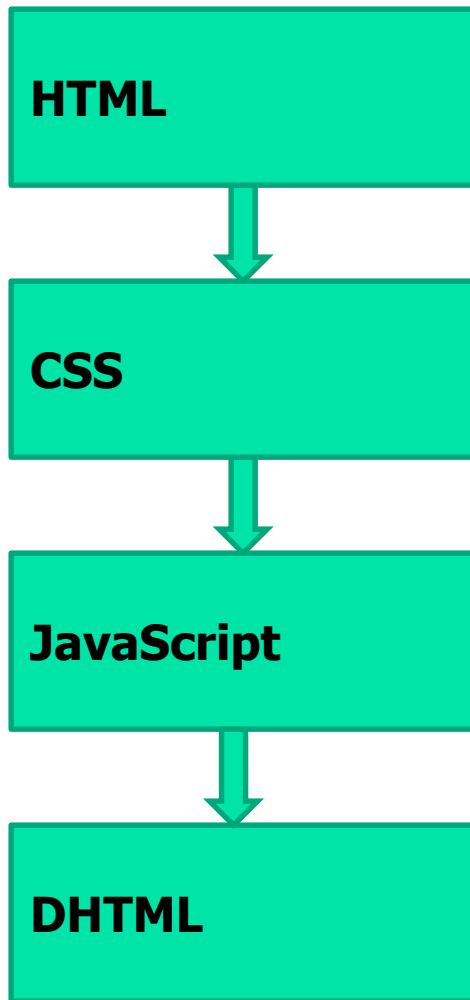
- Cung cấp cho sinh viên những kiến thức về DHTML
- Thực hiện các bài thực hành liên quan đến các kiến thức đã học: HTML, CSS, JavaScript



Nội dung

- 1. Giới thiệu về DHTML
- 2. Các đặc điểm của DHTML
- 3. Mô hình sự kiện DHTML
- 4. Thực hành

1. Giới thiệu về DHTML



HTML "tĩnh"

Style của trang

Tương tác





1. Giới thiệu về DHTML

- Dynamic HTML hay DHTML (HTML động)
Là sự kết hợp của HTML, CSS và JavaScript, HTML DOM trong một trang Web, giúp cho việc thao tác và xây dựng trang Web trở nên sinh động, dễ dàng và hiệu quả hơn

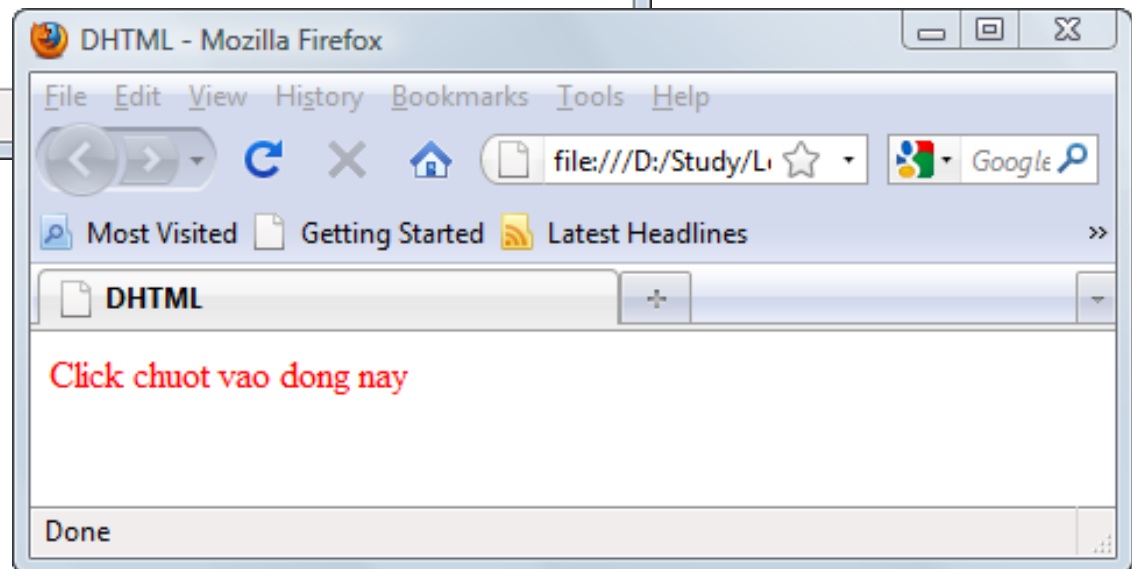
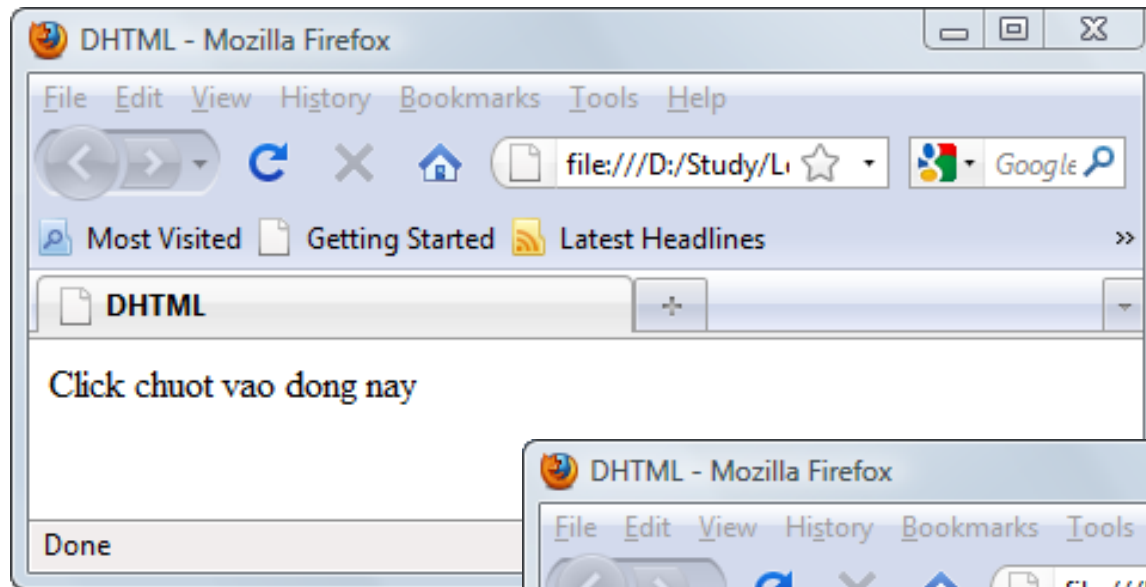
$$\text{DHTML} = \text{HTML} + \text{CSS} + \text{JavaScript} + \text{HTML DOM}$$



1. Giới thiệu về DHTML

```
<html>
<head>
    <title>DHTML</title>
</head>
<body>
    <p onclick="style.color='red'">
        Click chuột vào dòng này</p>
</body>
</html>
```

1. Giới thiệu về DHTML





2. Các đặc điểm của DHTML

- 2.1. Cấu trúc đối tượng
- 2.2. Style động
- 2.3. Định vị
- 2.4. Xử lý sự kiện



2.1. Cấu trúc đối tượng

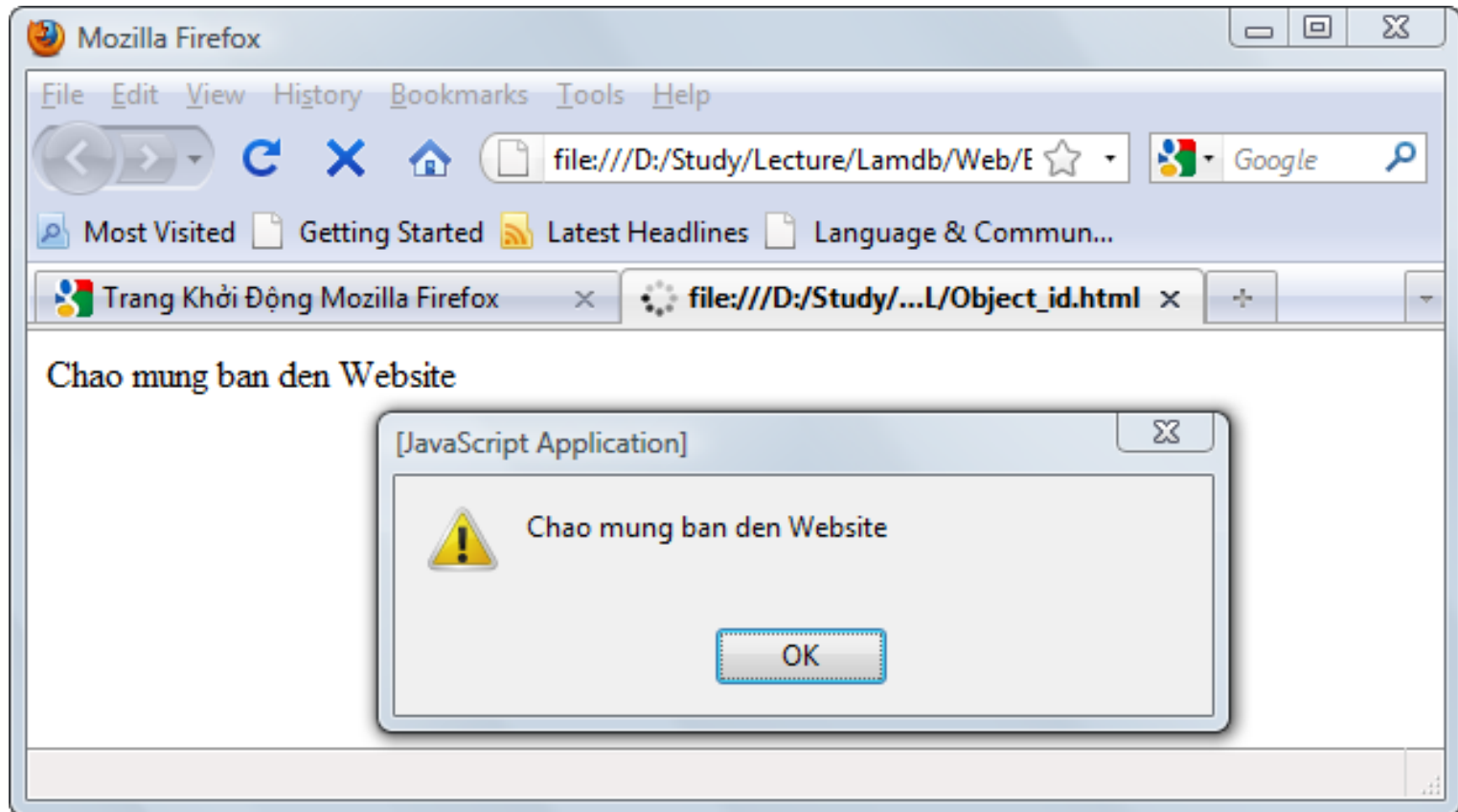
- DHTML coi mỗi phần tử là một đối tượng. Mỗi đối tượng có thể được truy cập và lập trình độc lập
- Truy cập vào đối tượng thông qua DOM
 - Thông qua tên đối tượng: ***name***
 - Thông qua id đối tượng: ***id***
method: ***document.getElementById(id)***
hoặc trực tiếp qua ***id***
 - Lấy nội dung sử dụng thuộc tính: ***innerHTML***



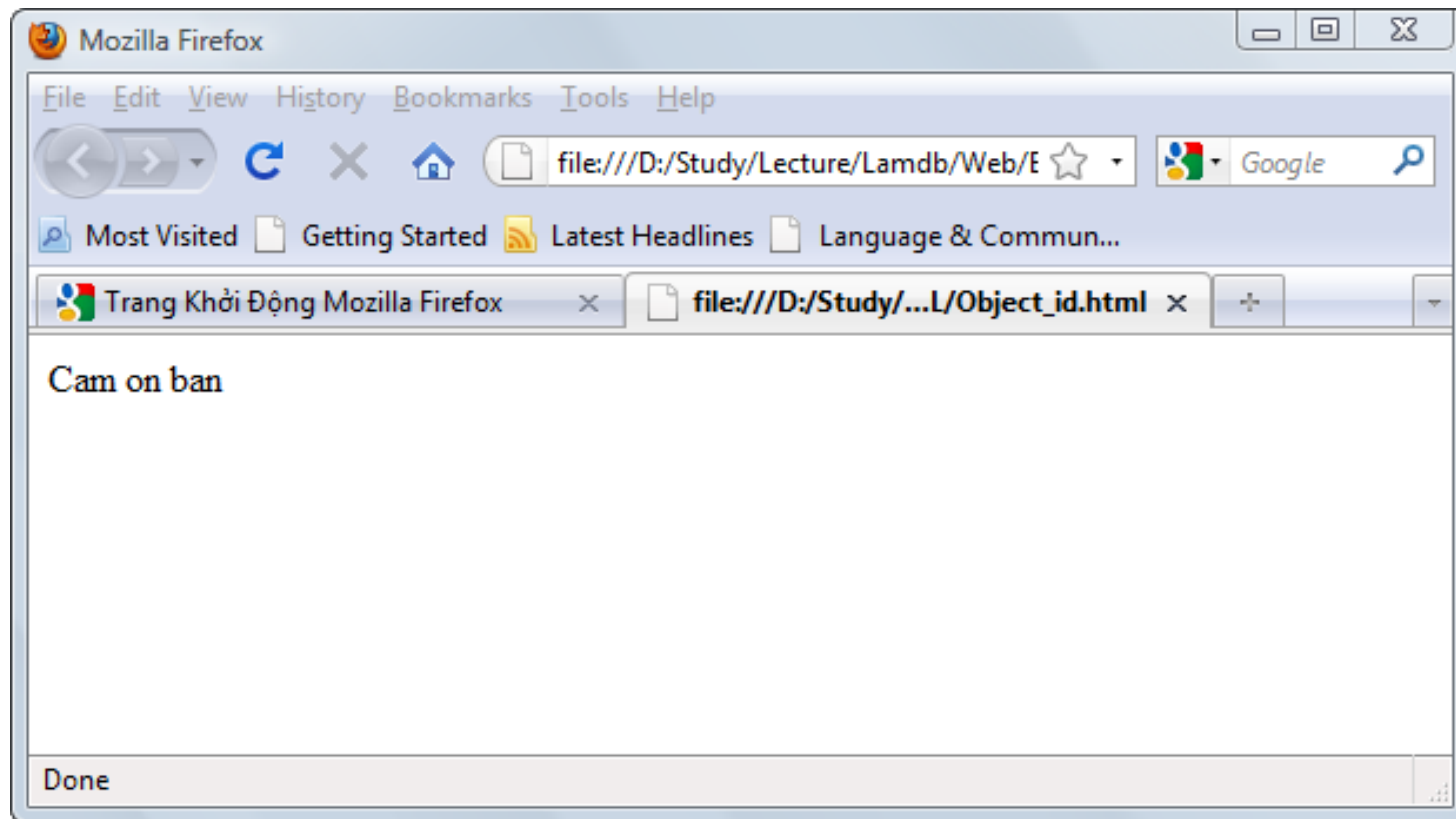
2.1. Cấu trúc đối tượng

```
<html>
<head>
  <script language="javascript">
    function bat_dau() {
      alert(pObj.innerHTML);
      pObj.innerHTML="Cam on ban";
    }
  </script>
</head>
<body onLoad="bat_dau();">
  <p id="pObj">Chao mung ban den Website</p>
</body>
</html>
```

2.1. Cấu trúc đối tượng



2.1. Cấu trúc đối tượng





2.2. Style động

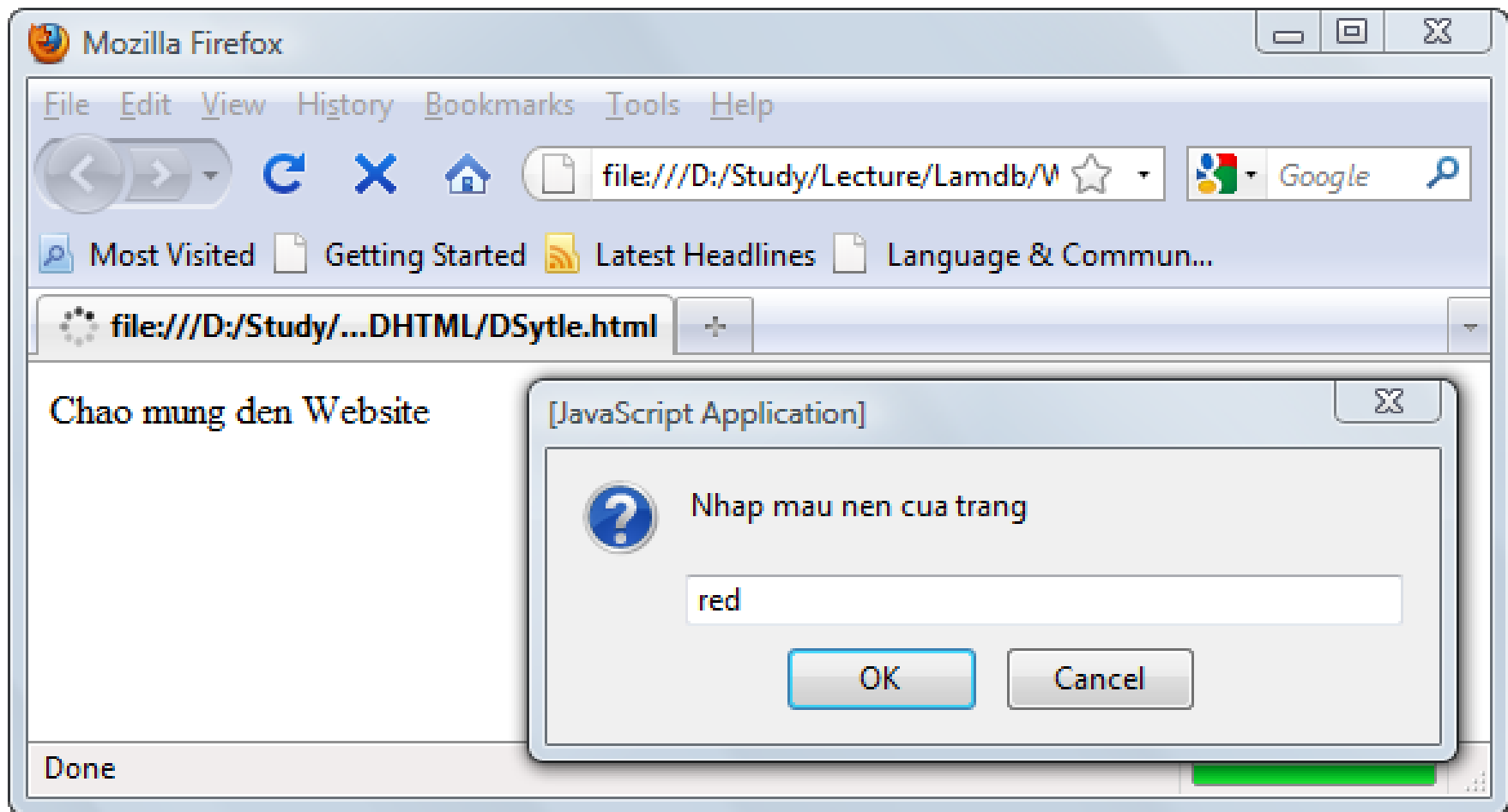
- Sử dụng CSS hay Style, song song với những tương tác của người sử dụng
- Thay đổi về Style như:
 - Màu sắc
 - Kiểu chữ, cỡ chữ, font chữ
 - Khoảng cách
 - Vị trí
 -



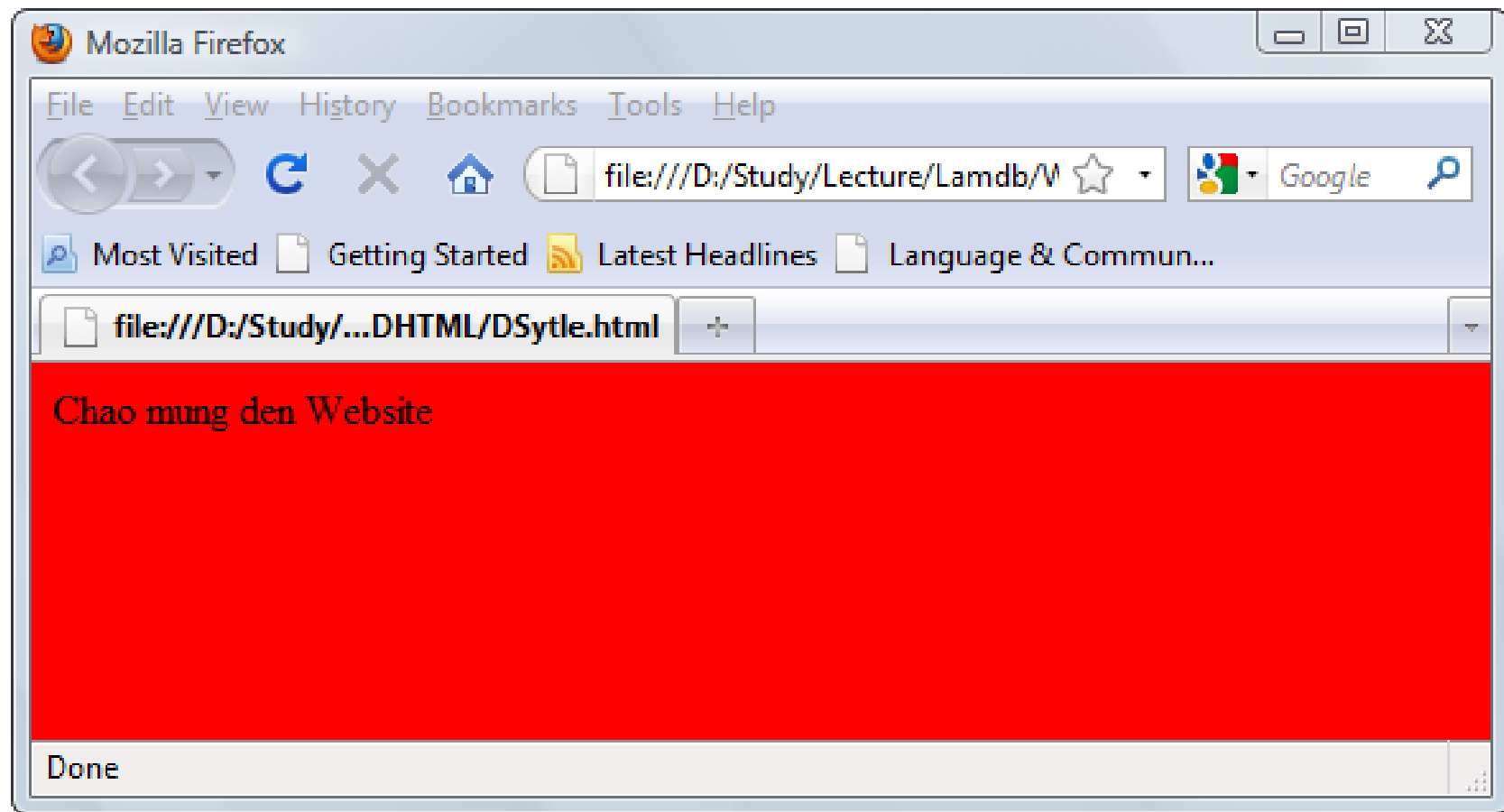
2.2. Style động

```
<html>
  <head>
    <script language="javascript">
      function bat_dau() {
        var mau=prompt("Nhap mau nen cua trang","");
        document.body.style.backgroundColor=mau;
      }
    </script>
  </head>
  <body onload="bat_dau();" >
    <p>Chao mung den Website</p>
  </body>
</html>
```

2.2. Style động



2.2. Style động





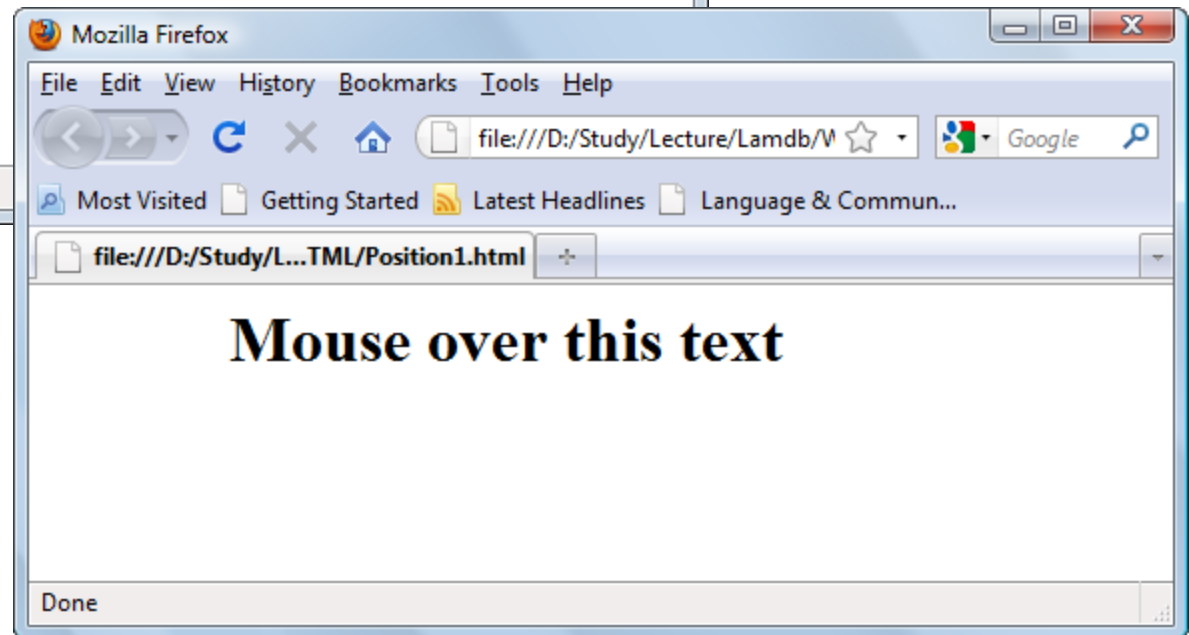
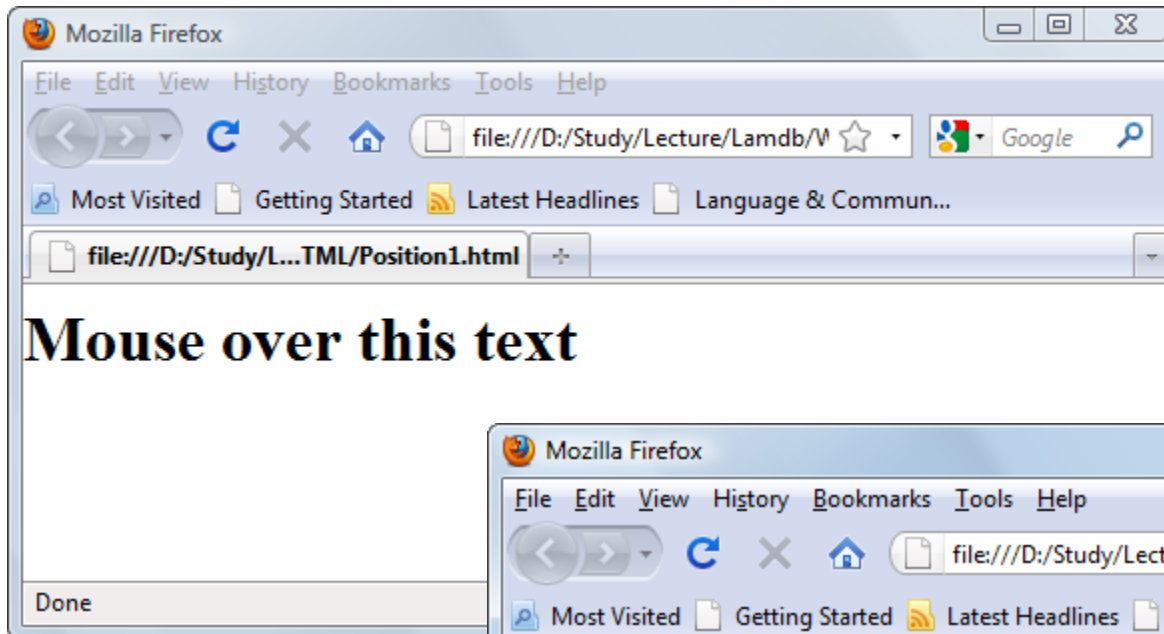
2.3. Định vị

- HTML: không cho phép điều chỉnh tọa độ của phần tử
- DHTML: cho phép điều chỉnh tọa độ của phần tử. Chúng ta có thể chỉ ra vị trí chính xác (tương đối hoặc tuyệt đối).

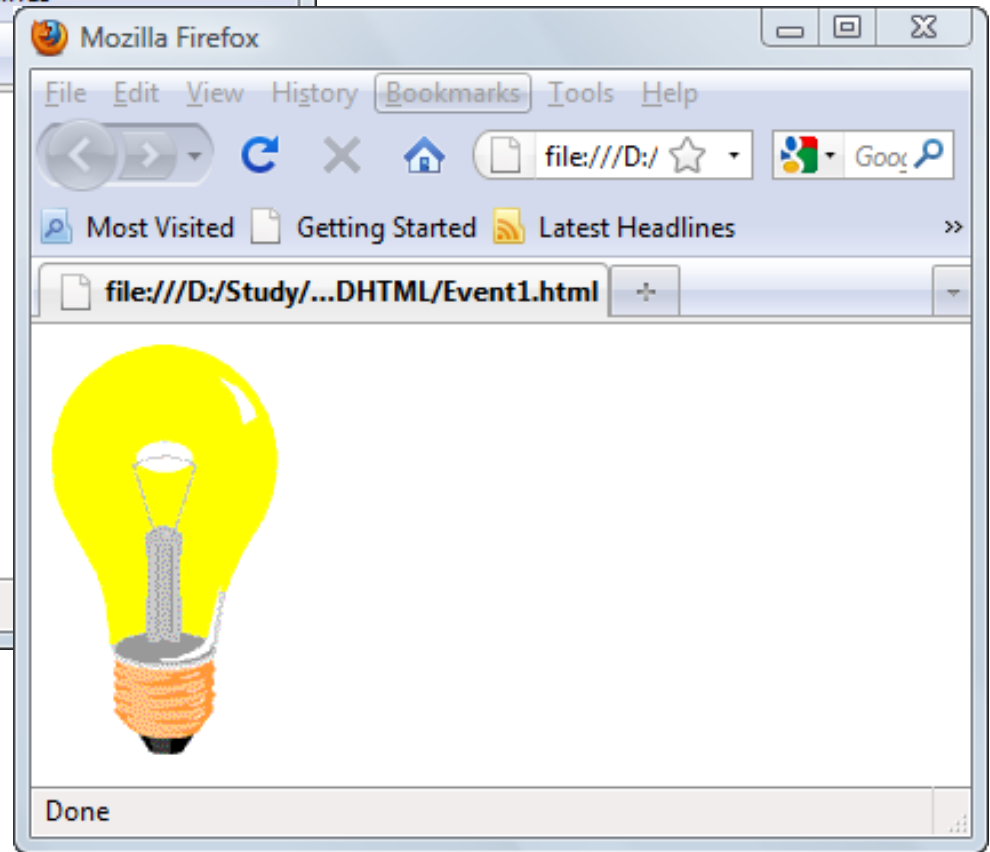
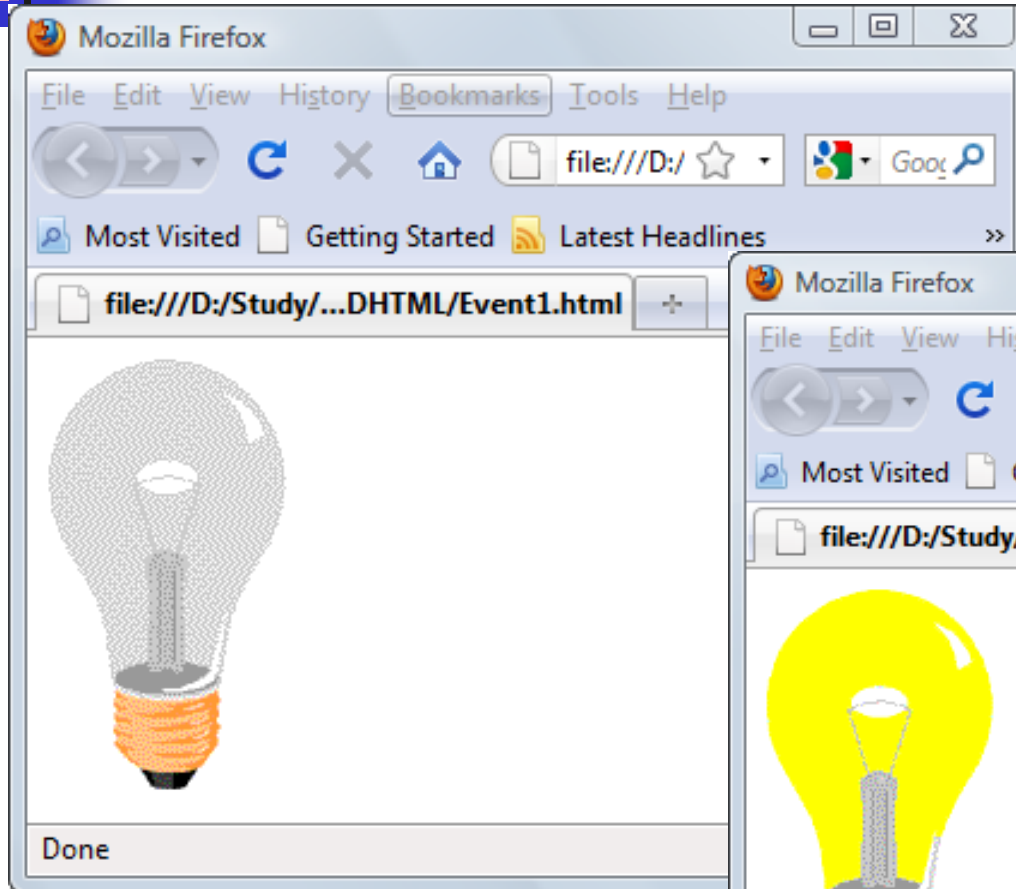


```
<html>
<head>
<script language="javascript">
    function moveleft() {
        document.getElementById('header').style.position=
                                                    "absolute";
        document.getElementById('header').style.left="0";
    }
    function moveback() {
        document.getElementById('header').style.left="100";
    }
</script>
</head>
<body>
    <h1 id="header" onmouseover="moveleft()"
        onmouseout="moveback()"> Mouse over this text</h1>
</body>
</html>
```

2.3. Định vị



2.4. Xử lý sự kiện



```
<html>
  <head>
    <script language="javascript">
      bLight=0;
      function changeimage() {
        if (bLight==0) {
          bLight=1; document.myimage.src="bulbon.gif";
        }
        else{
          bLight=0; document.myimage.src="bulboff.gif";
        }
      }
    </script>
  </head>
  <body>
    
  </body>
</html>
```



Sử dụng DHTML

- DHTML có 3 đặc điểm:
 1. Style động: cho phép chỉ ra kiểu định dạng văn bản trong trang web: màu sắc, kích thước, kiểu chữ...
 2. Định vị: cho phép chỉ ra vị trí của các phần tử HTML trong một trang theo tọa độ x, y
 3. Xử lý sự kiện: cho phép nắm bắt sự kiện do người dùng hay hệ thống tạo ra để sửa đổi kiểu hay nội dung trang web



3. Mô hình xử lý sự kiện DHTML

- 3.1. Sự kiện
- 3.2. Nổi bật sự kiện
- 3.3. Kiểm soát lỗi với sự kiện onerror



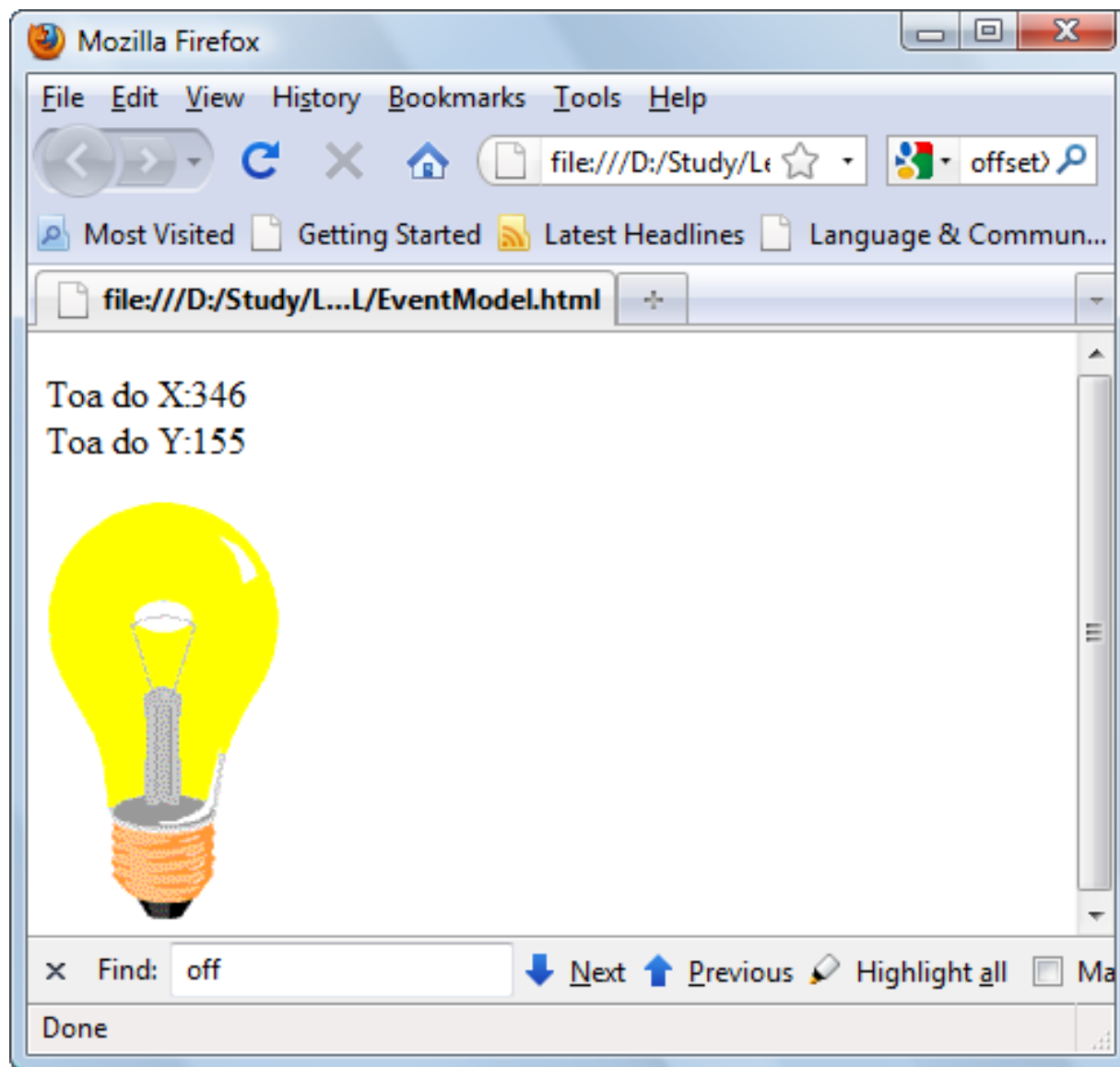
3.1. Sự kiện

- Mô hình sự kiện (Event Model) là một phần quan trọng trong mô hình đối tượng tài liệu (Document Object Model)
- Là thành phần chính tạo nên một trang Web động và có tương tác với người sử dụng
- Mô hình sự kiện xác định sự kiện nào sinh ra, thông tin cung cấp cho trình xử lý sự kiện (event handler)


```
<html>
<body>
<script language="JavaScript">
function toa_do(event1) {
    myX.innerHTML = event1.pageX ;
    myY.innerHTML = event1.pageY;
}
</script>
<p>Toa do X:<span id="myX">0</span><br>
    Toa do Y:<span id="myY">0</span></p>
<div id="myDiv" onmousemove="toa_do(event) ;">

</div>
</body>
</html>
```

3.1. Sự kiện

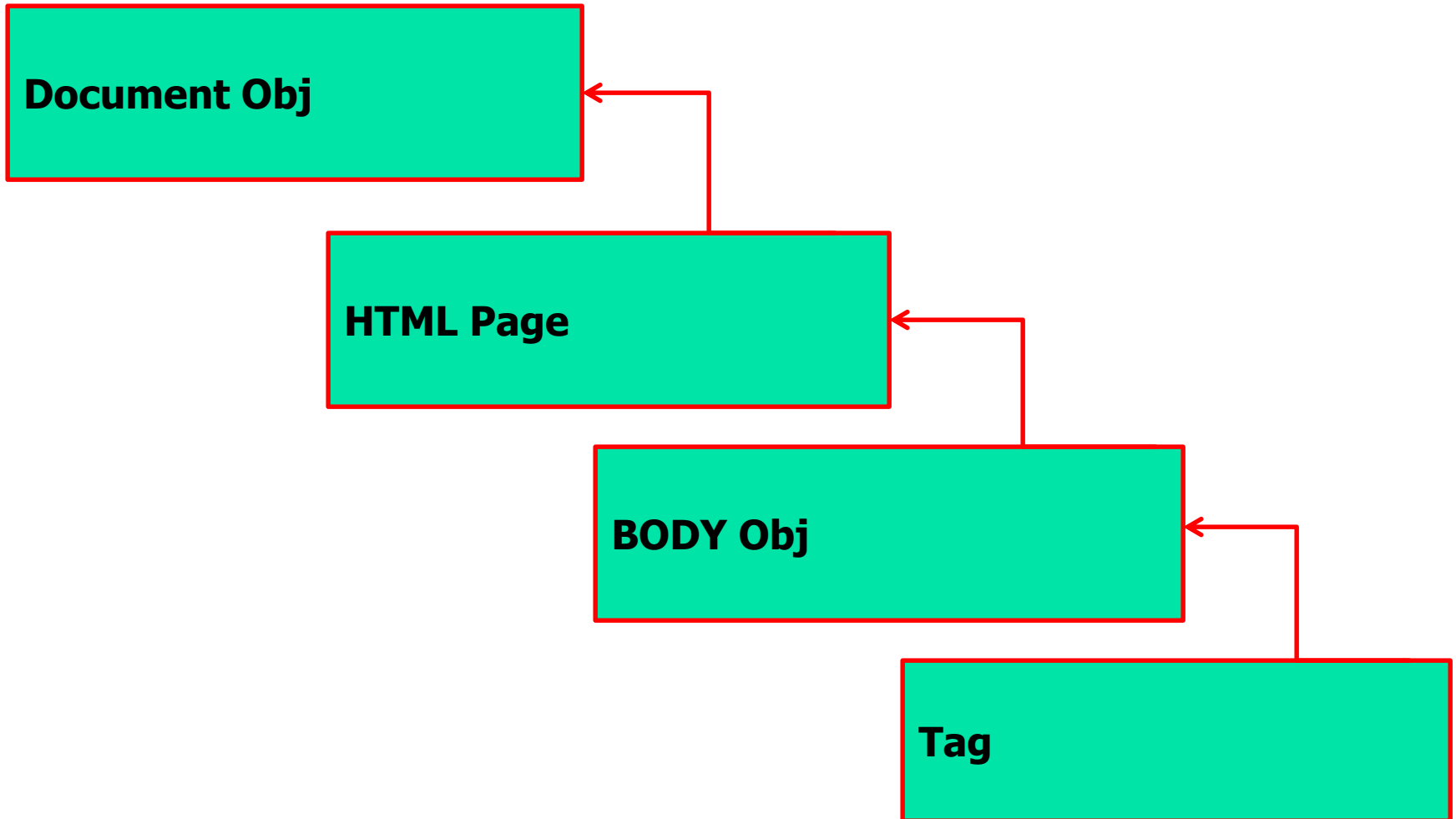




3.2. Nổi bật sự kiện

- HTML: muốn xác định hành vi cho một sự kiện chúng ta phải đưa ra đoạn mã mô tả cho đối tượng đó
- DHTML cung cấp cho chúng ta một hệ thống phân cấp đối tượng.

3.2. Nội bộ sự kiện





3.2. Nổi bọt sự kiện

■ Giải thích

- Khi một sự kiện ở cấp "dưới" xảy ra, thì sự kiện đó được chuyển ngược lên cấp phía trên
- Sự phân cấp này không phụ thuộc vào thứ tự các mục hiển thị trên tài liệu

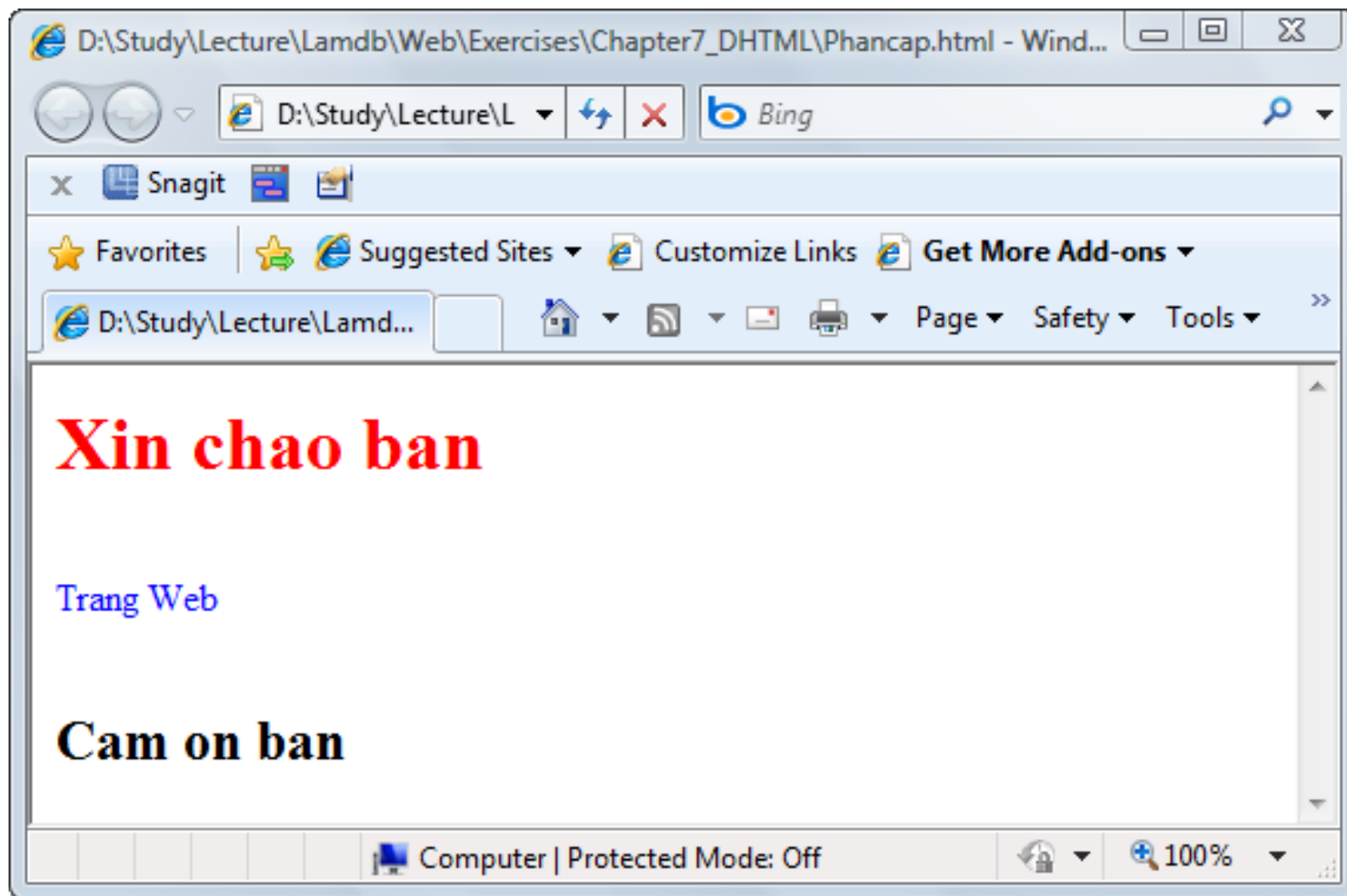
```
<body onClick="ClickHere() ;">
<h1 style="color:red">Xin chao ban</h1><br />
<p style="color:blue" onclick="AlsoClick() ;"
>Trang Web</p><br />
<h2>Cam on ban</h2>
  <script language="javascript">
    function ClickHere() {
      alert("Thanh phan duoc bam
"+window.event.srcElement.tagName);
    }
    function AlsoClick() {
      alert("Ban da bam
"+window.event.srcElement.tagName);
    }
  </script>
</body>
```



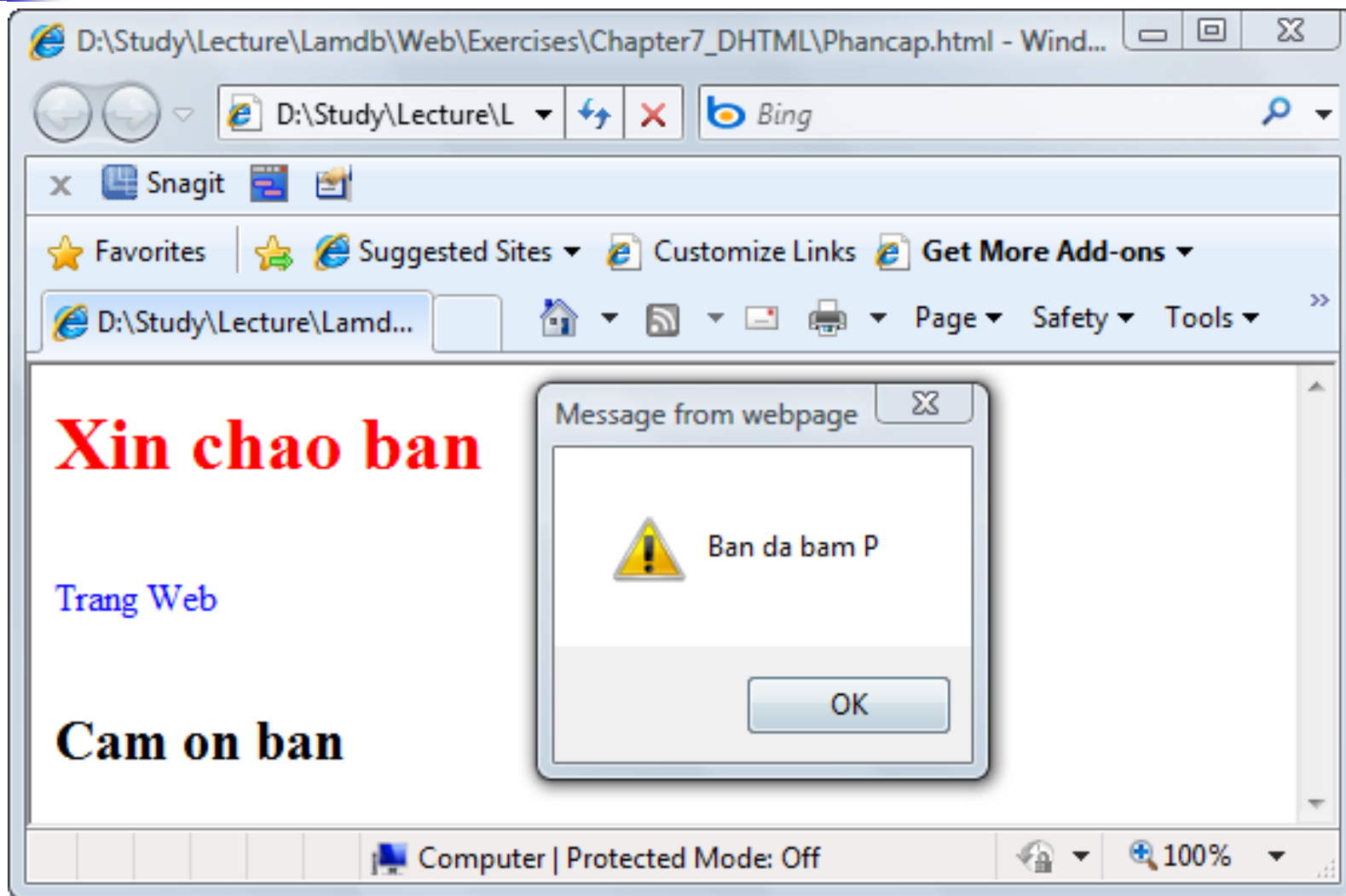
3.2. Nổi bật sự kiện

- Khi người dùng click chuột vào H1 thì sự kiện OnClick xảy ra
- H1 không có bộ xử lý sự kiện, sự kiện này sẽ chuyển lên đến BODY và gọi hàm ClickHere()
- Khi người dùng click chuột vào P thì sự kiện OnClick xảy ra
- P có bộ xử lý sự kiện và gọi AlsoClick(). Tiếp tục chuyển lên BODY và gọi ClickHere()

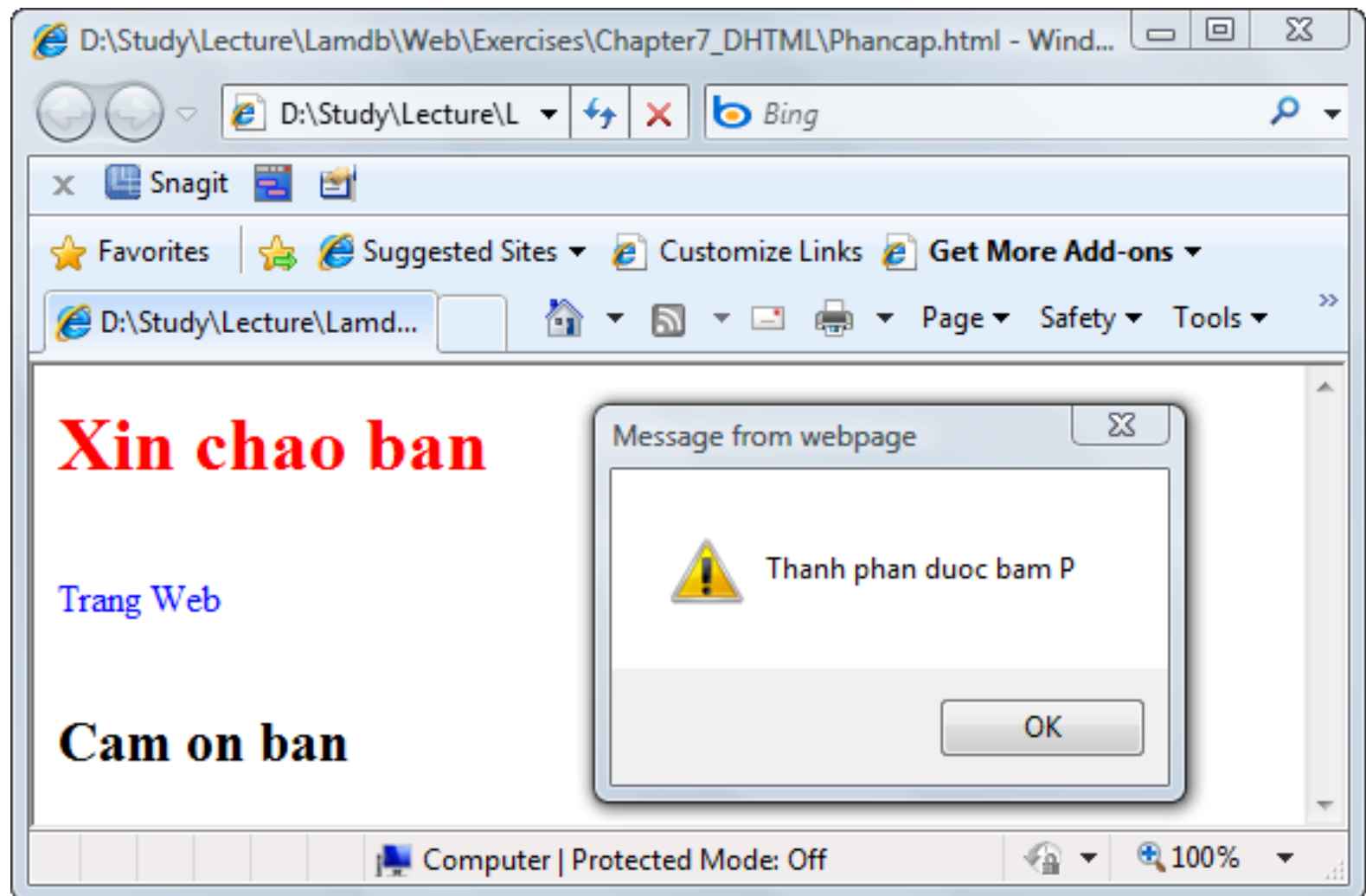
3.2. Nổi bọt sự kiện



3.2. Nổi bọt sự kiện



3.2. Nổi bọt sự kiện





3.2. Nổi bọt sự kiện

- Nếu không muốn các sự kiện chuyển lên cấp cao hơn, đặt thuộc tính ***cancelBubble*** của đối tượng là True

window.event.cancelBubble=true;

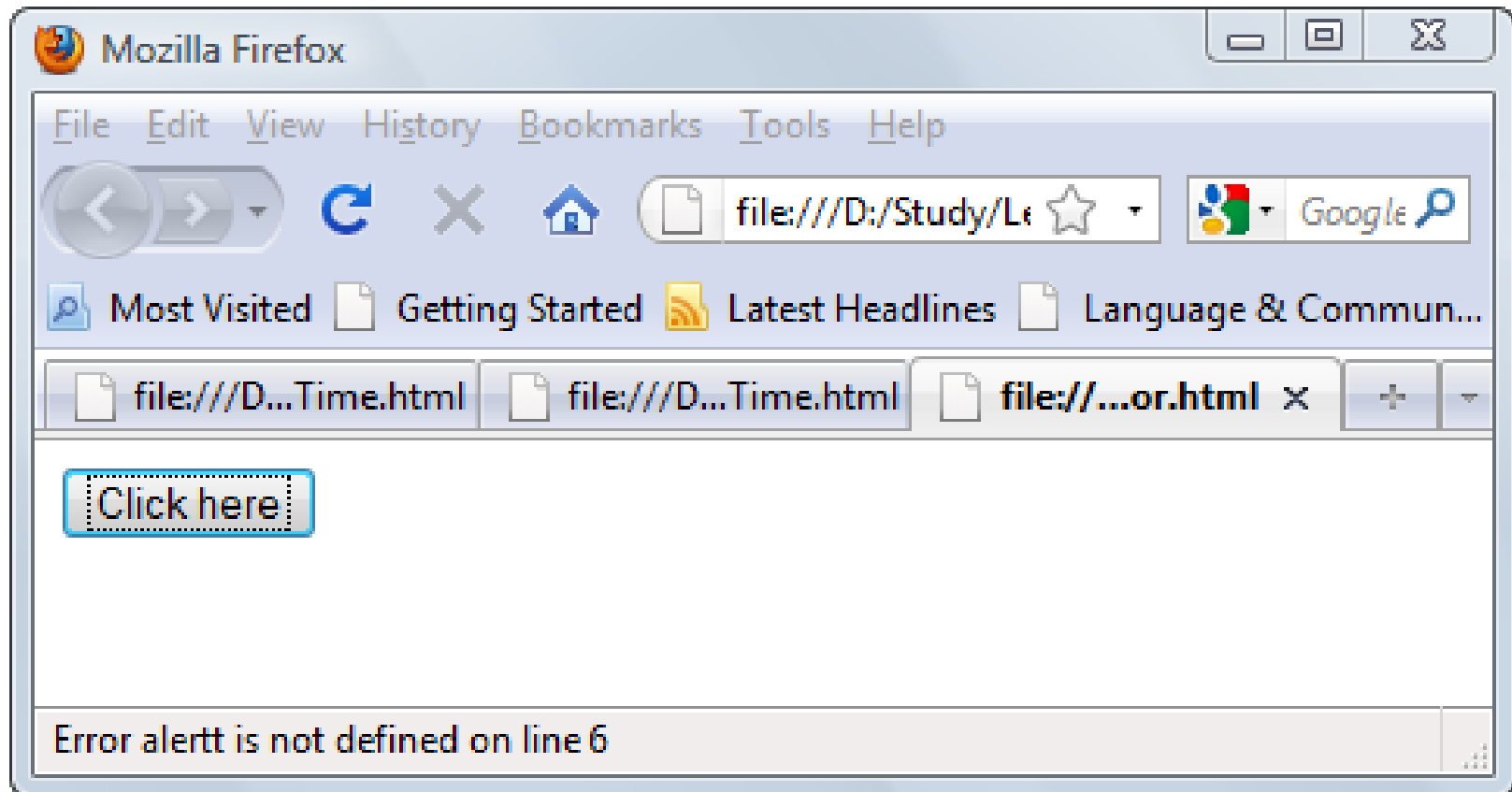


3.3. Kiểm soát lỗi với sự kiện onerror

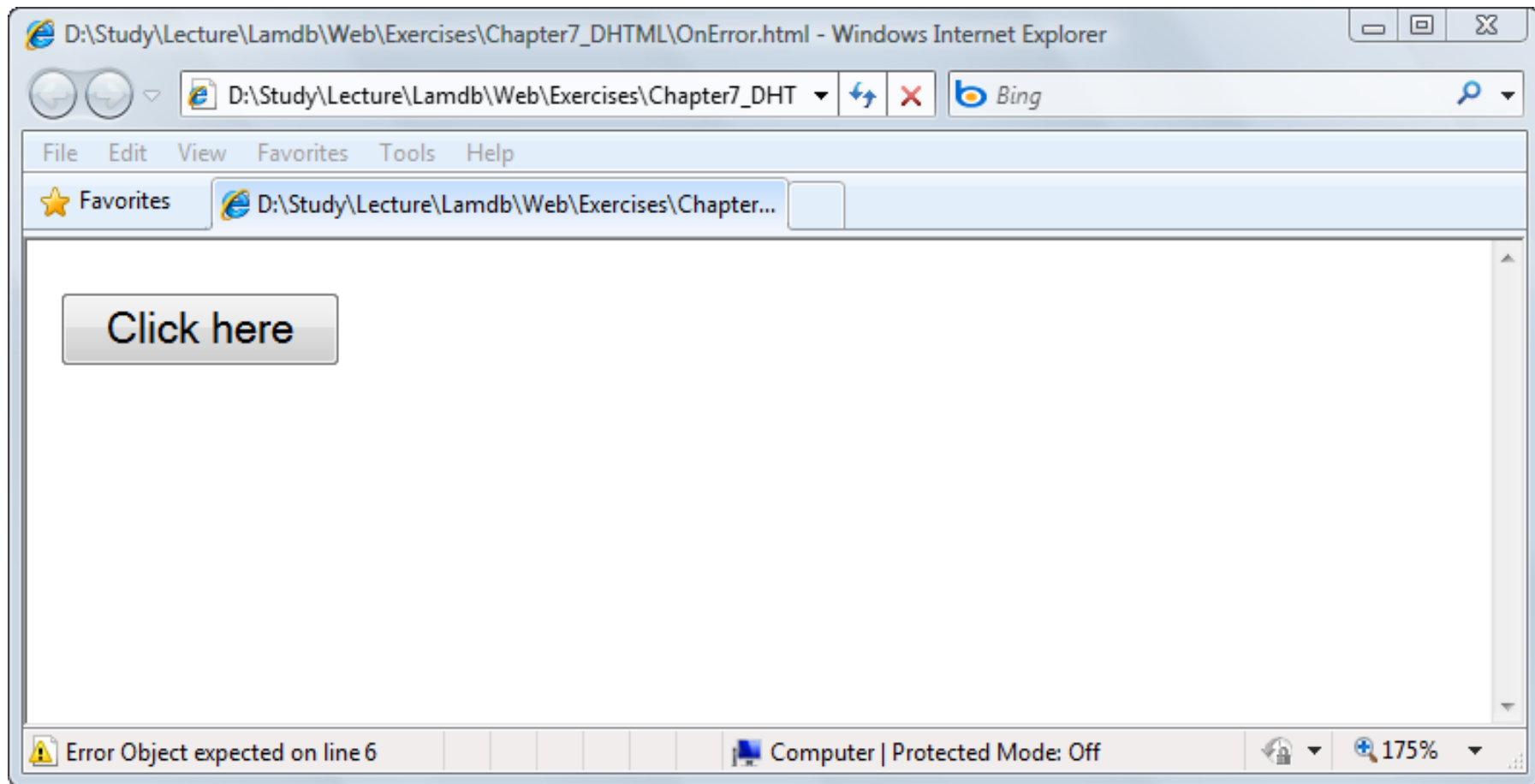
- Sự kiện onerror được thực thi khi trình duyệt phát hiện lỗi trong đoạn mã JavaScript
- Giúp cho người lập trình biết điều gì đang xảy ra

```
<html>
  <head> <script language="javascript">
    window.onerror = soat_loi;
    function hien_thi(){
      alert("xin chao ban");
    }
    function soat_loi(kieu_loi,url,dong_sai){
      window.status = "Error "+kieu_loi+" on
                        line "+dong_sai;
    }
  </script> </head>
  <body>
    <input id="button" type="button"
value="Click here" onclick="hien_thi();" />
  </body>
</html>
```

3.3. Kiểm soát lỗi với sự kiện onerror



3.3. Kiểm soát lỗi với sự kiện onerror

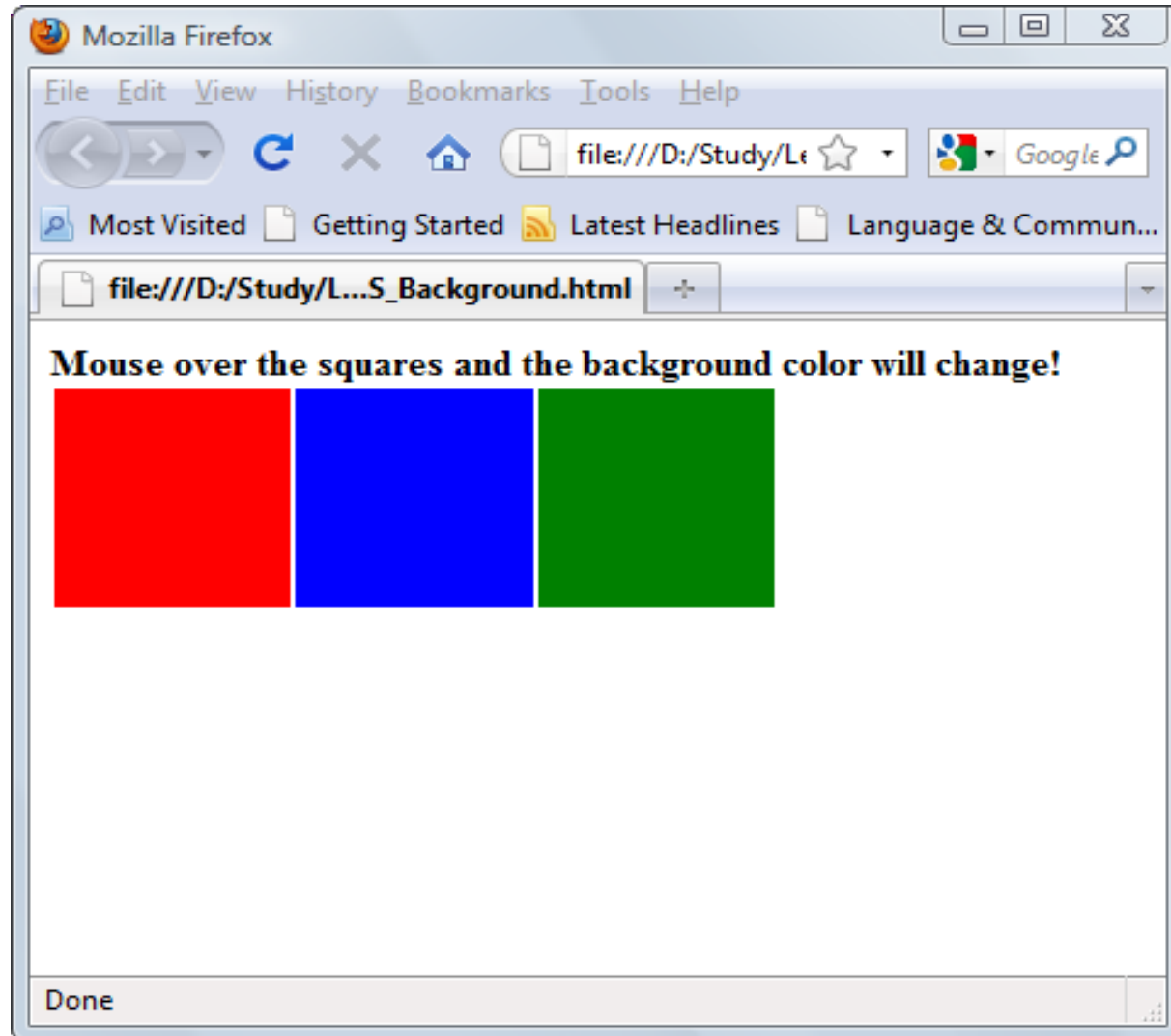




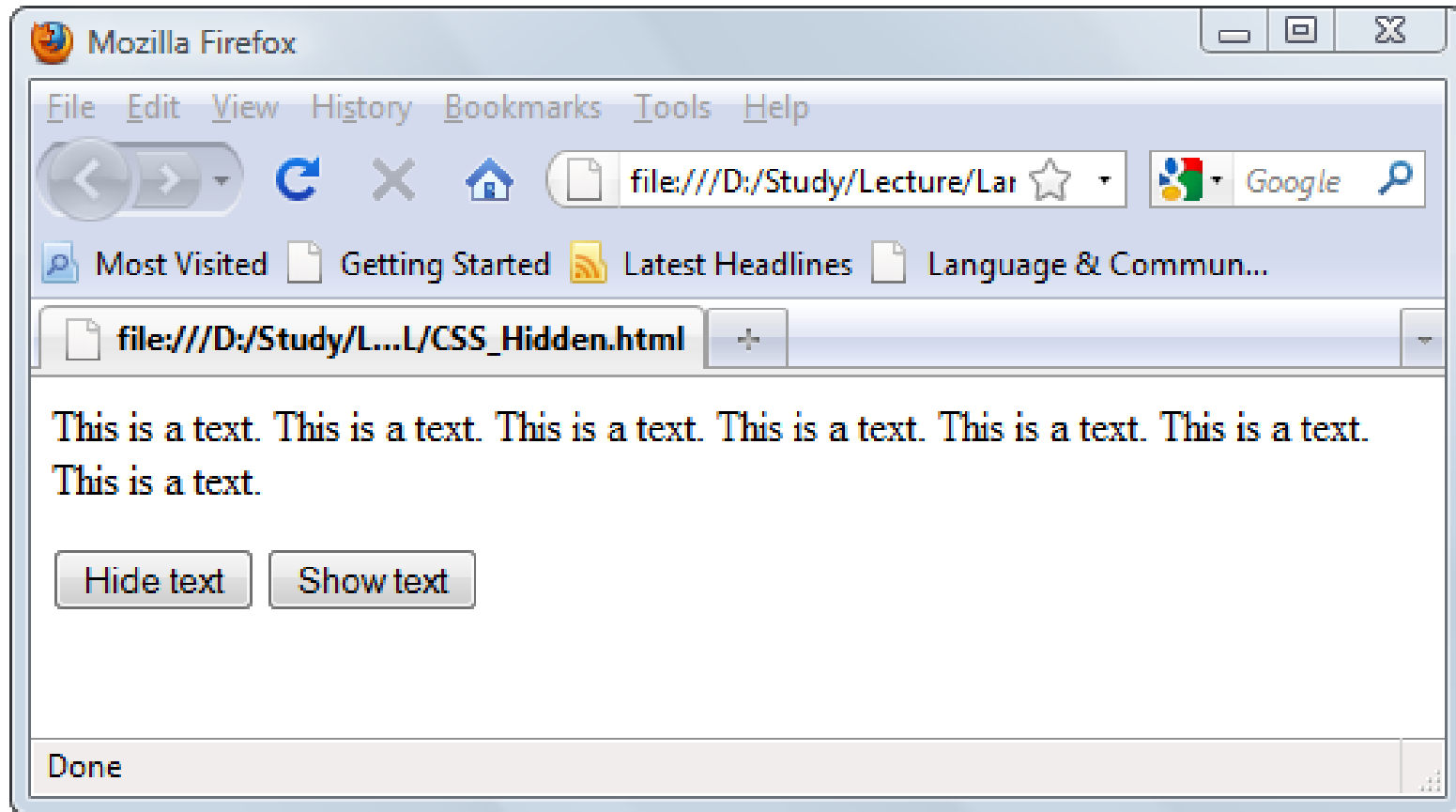
4. Thực hành

- 1. CSS
- 2. Event
- 3. Text
- 4. Image
- 5. Menu
- 6. Mouse
- 7. Link
- 8. Input Form
- 9. Window

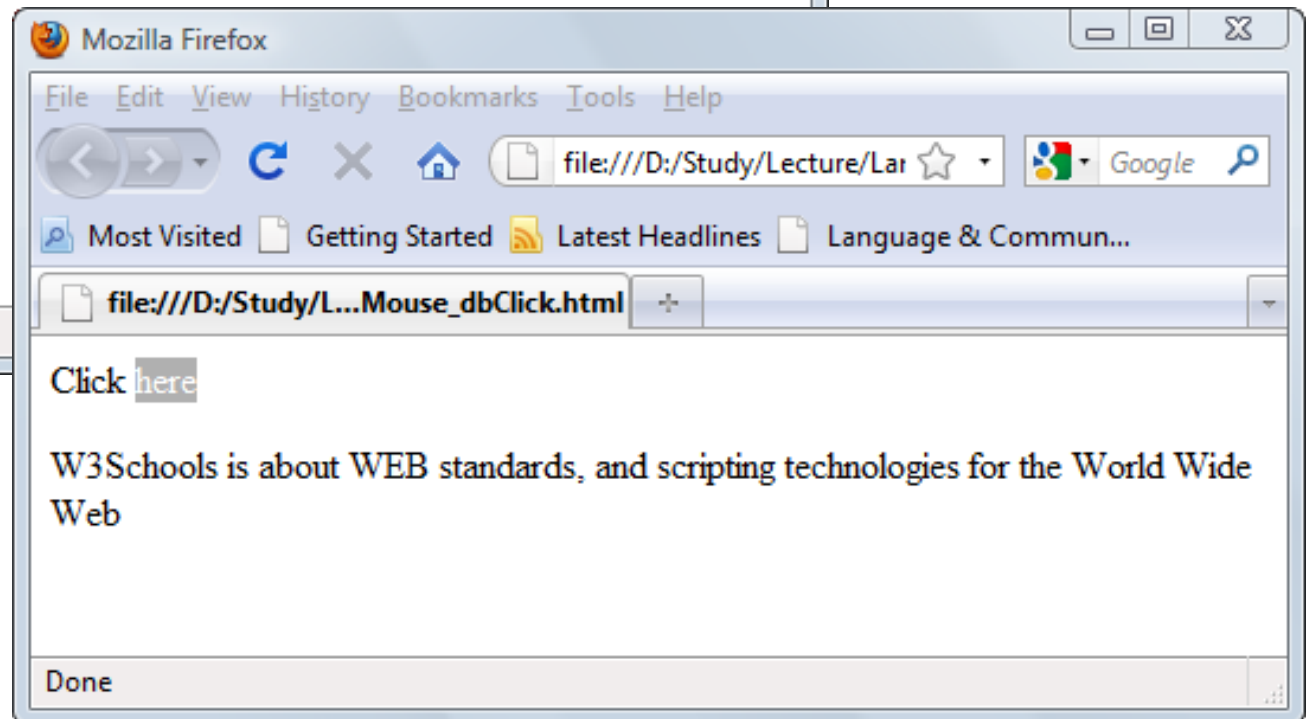
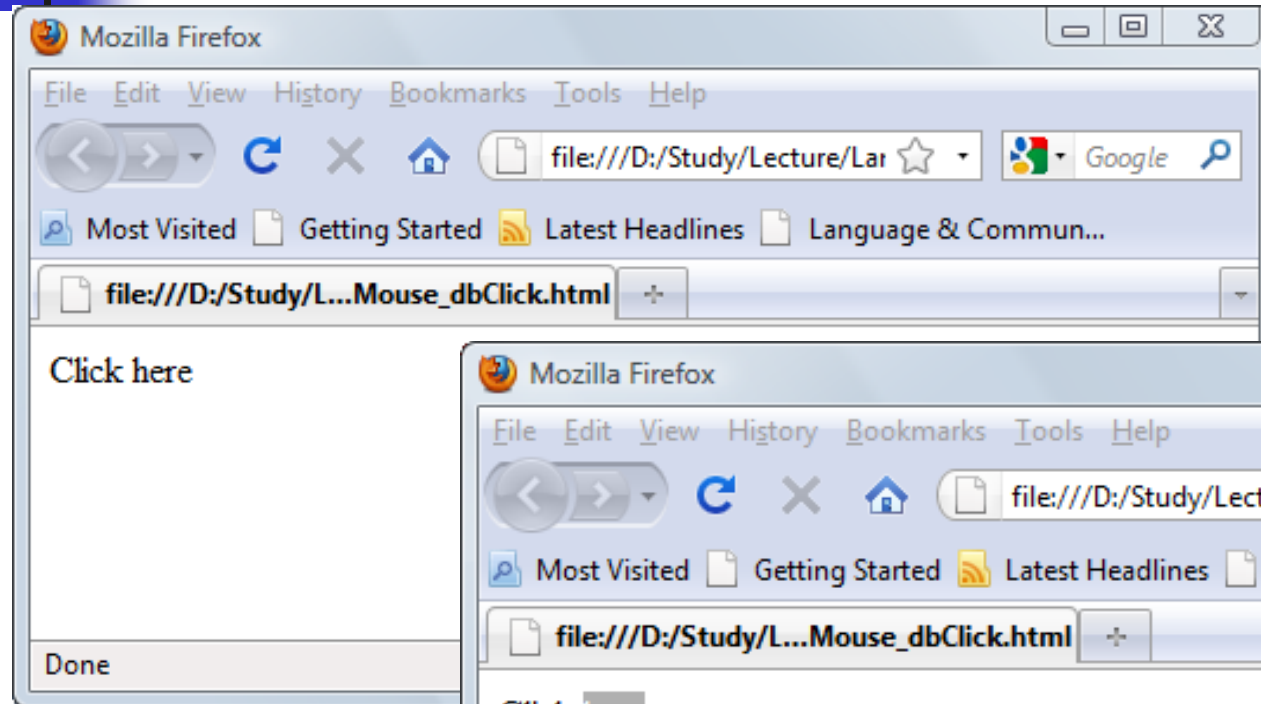
1. CSS



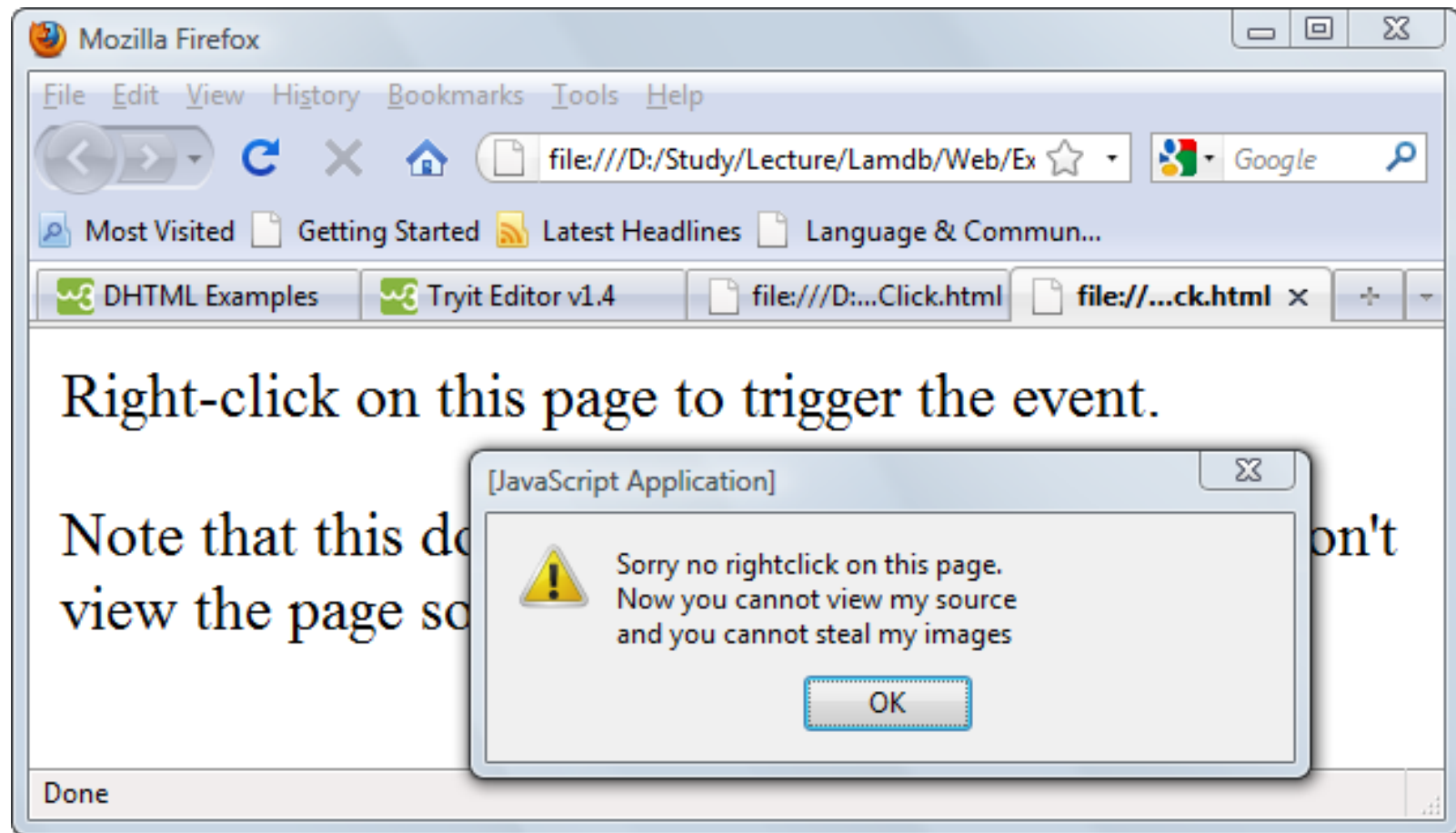
1. CSS



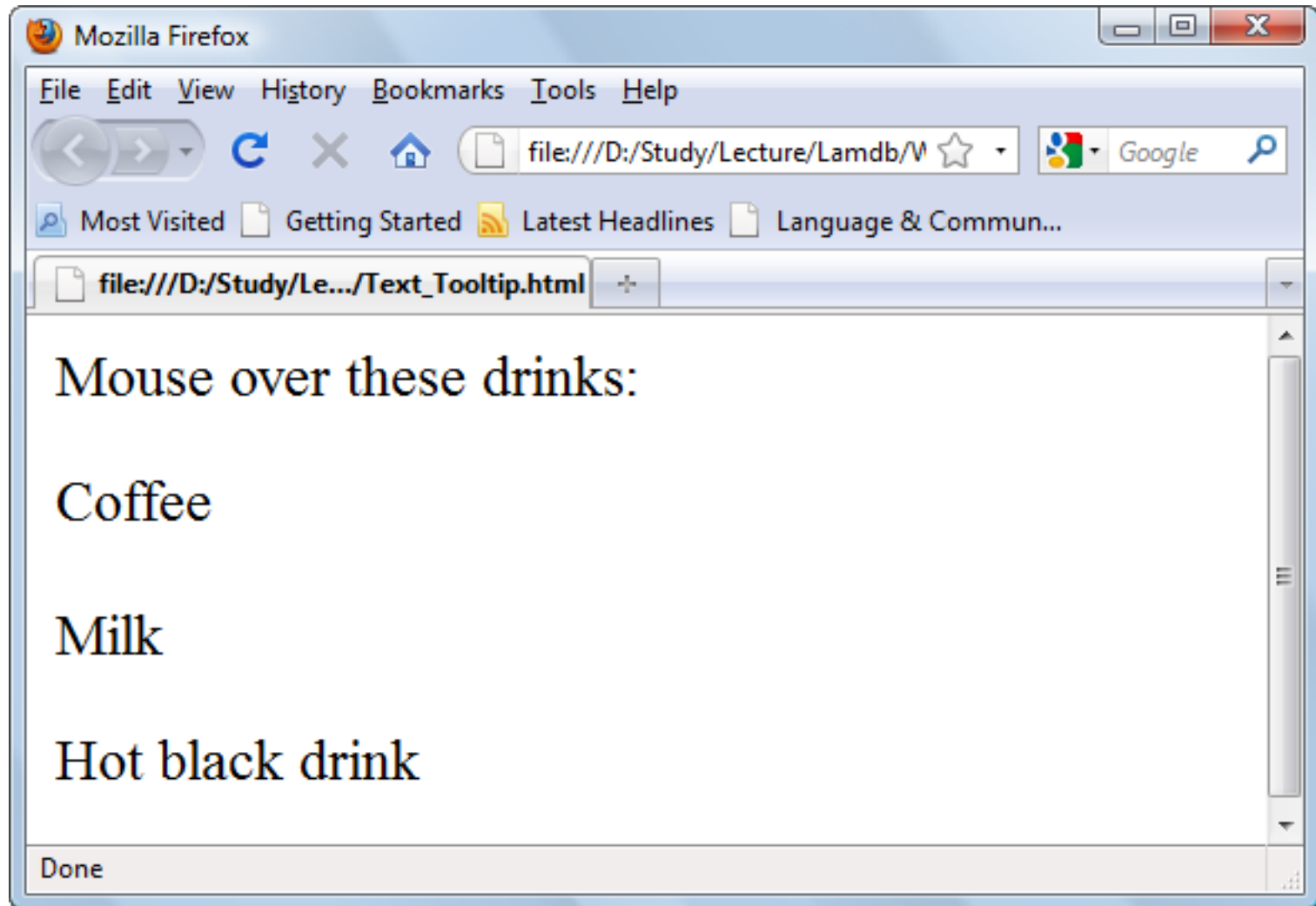
2. Event



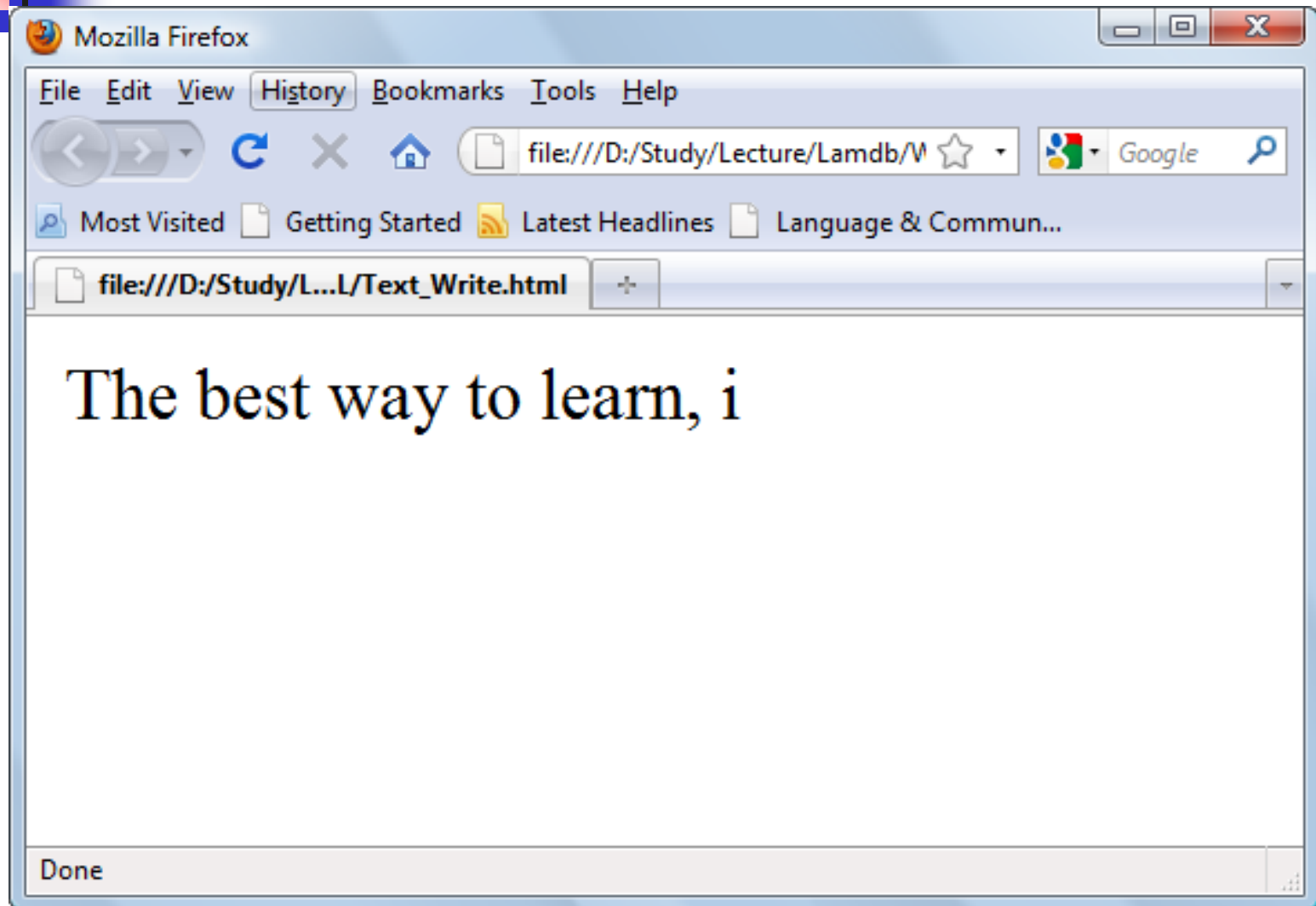
2. Event



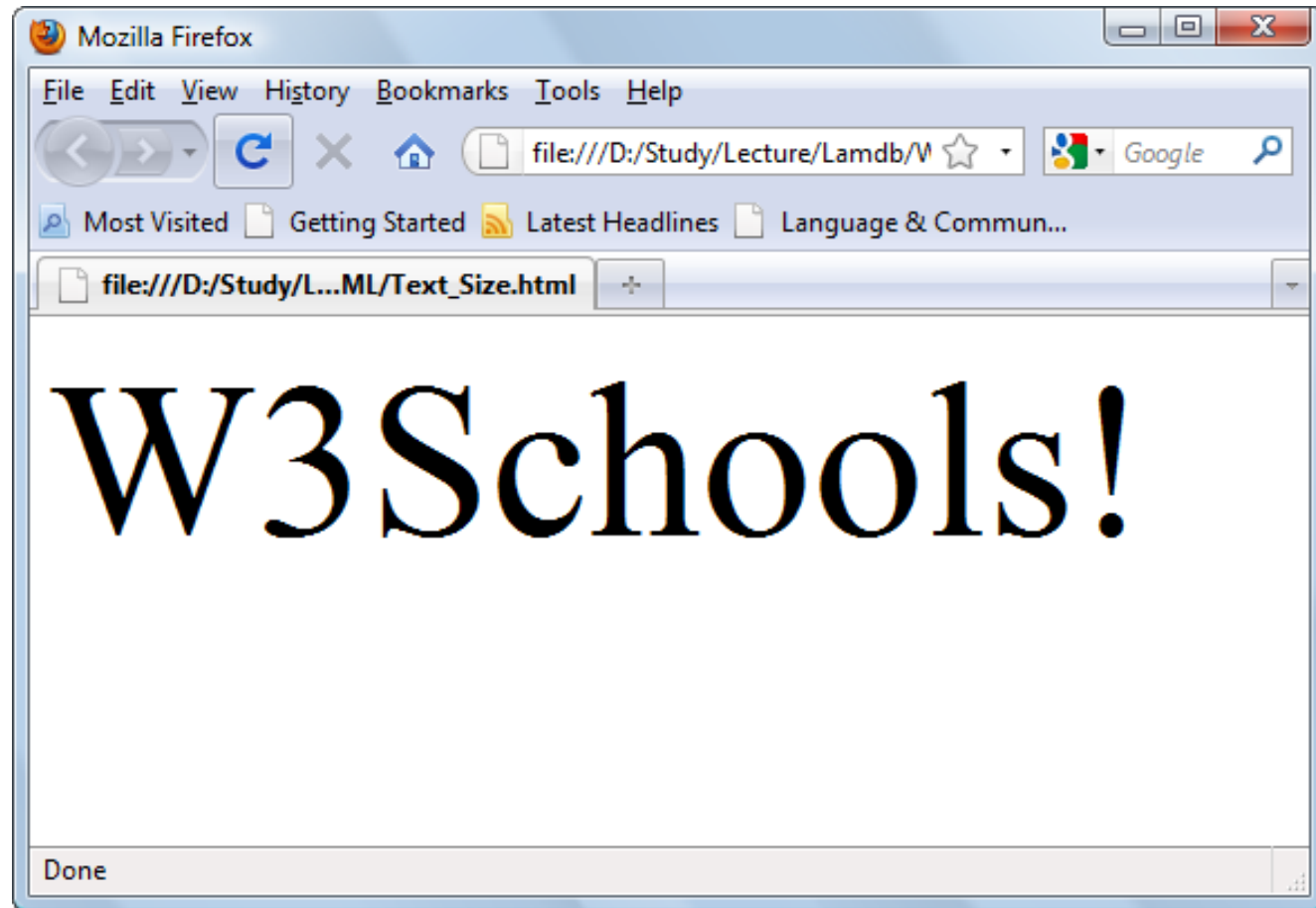
3. Text - ToolTip



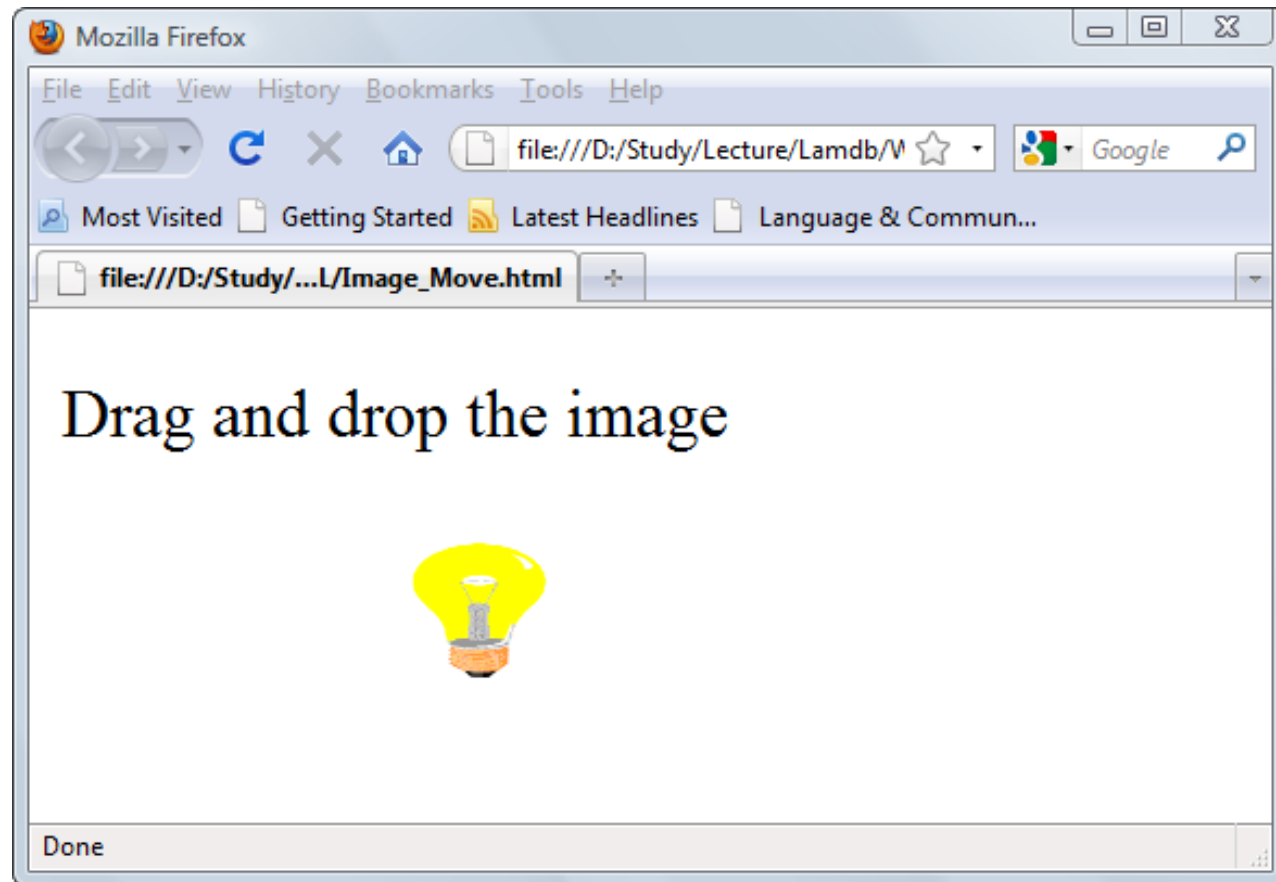
3. Text - Write Message



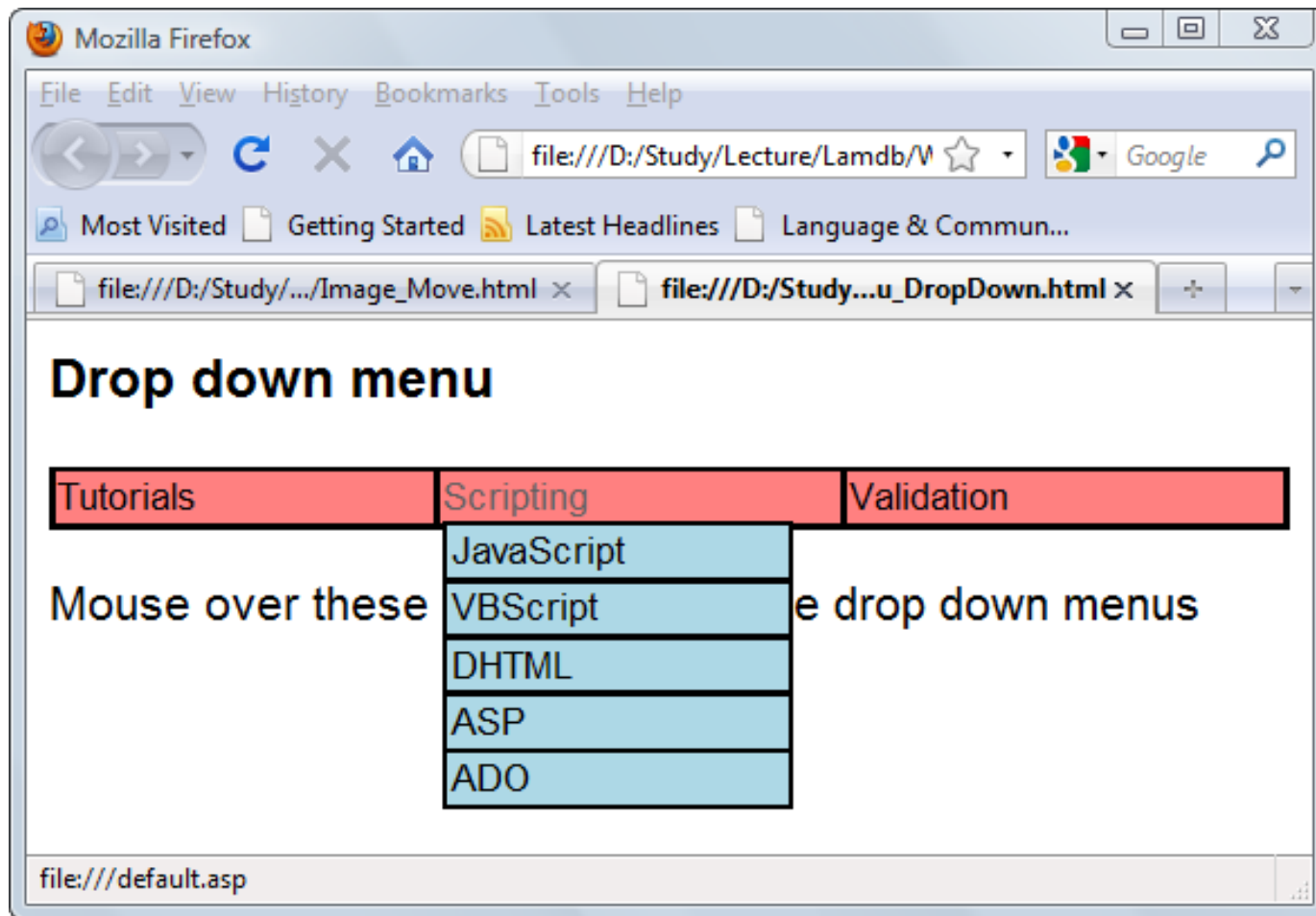
3. Text - Size



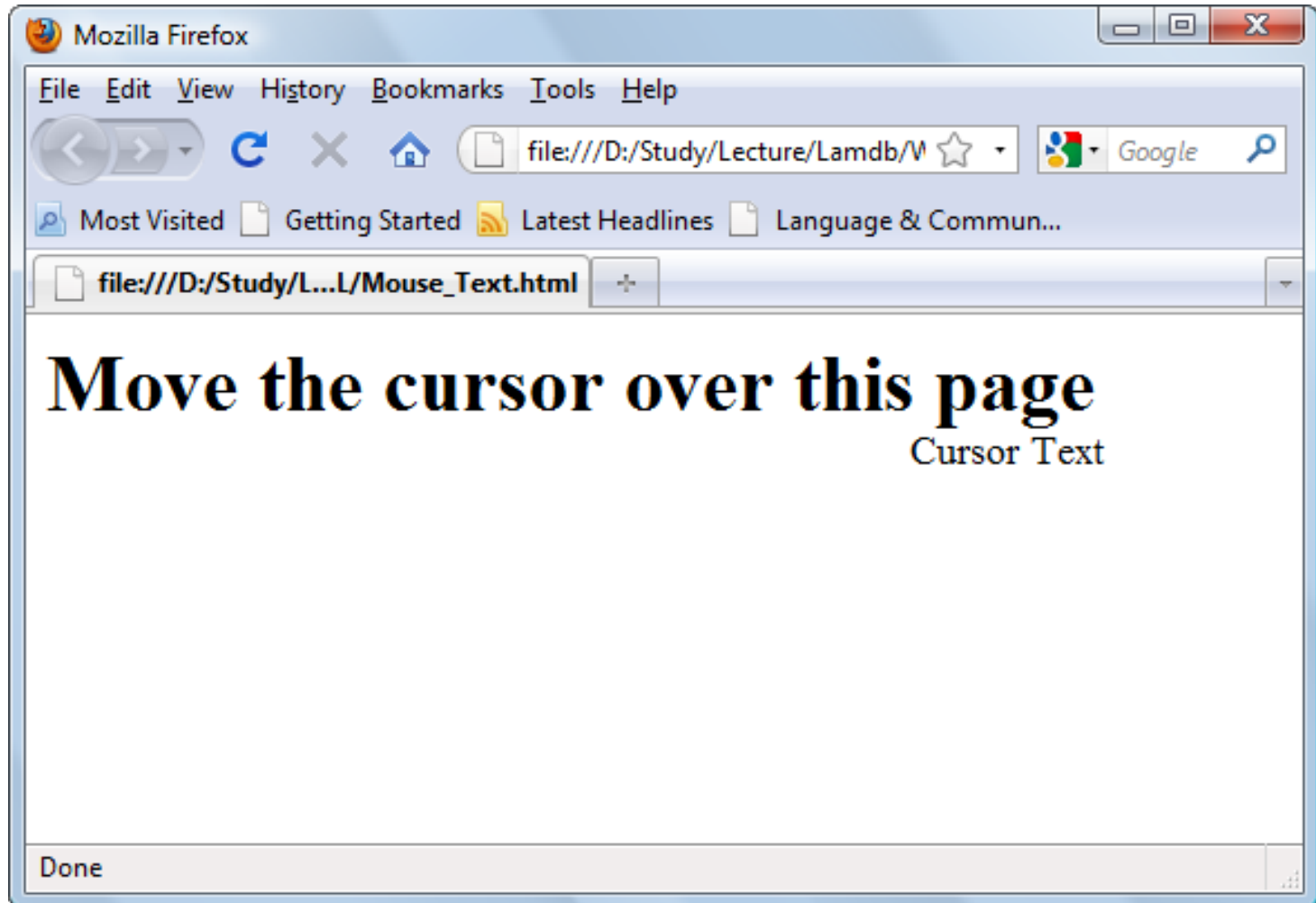
4. Image



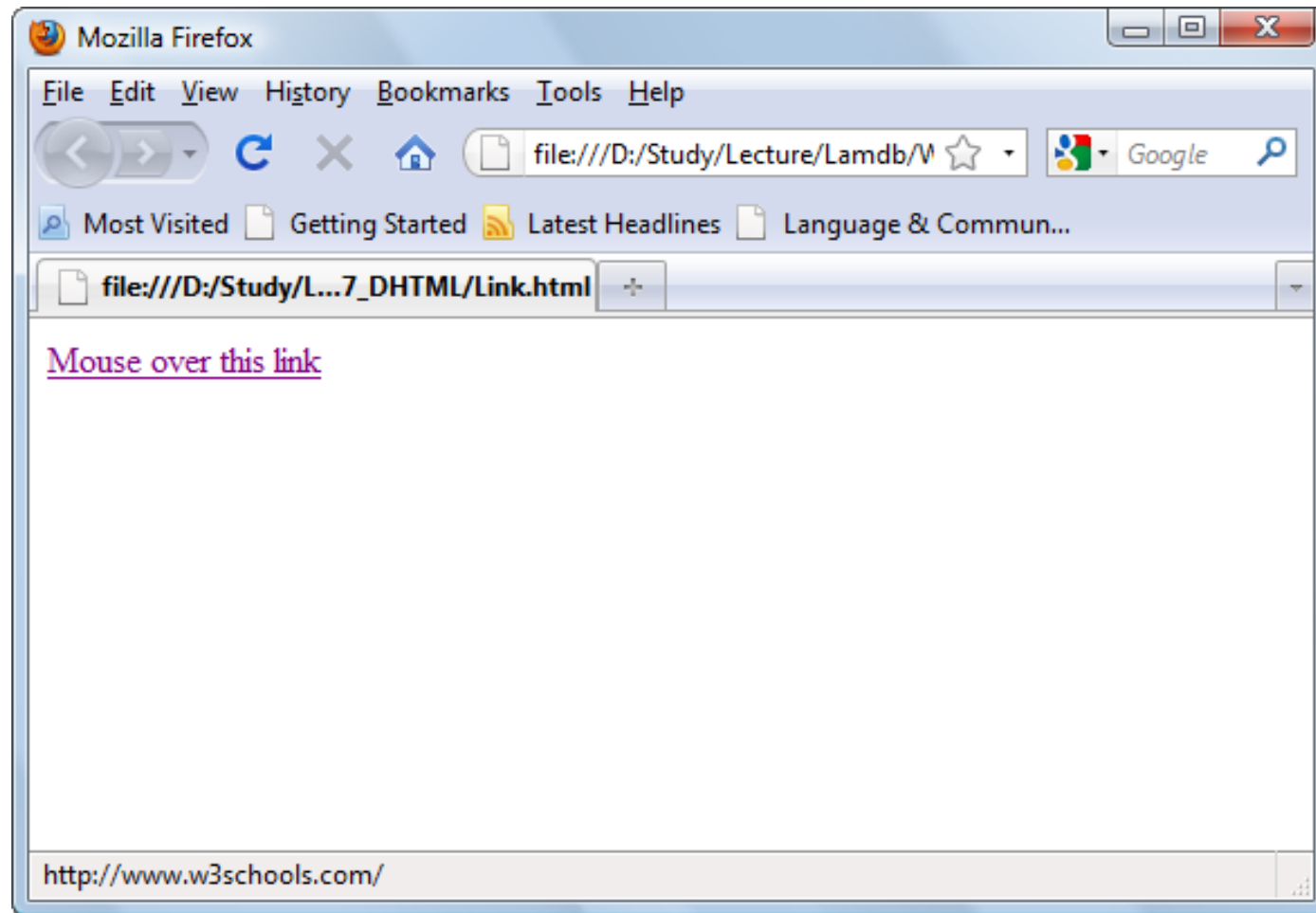
5. Menu



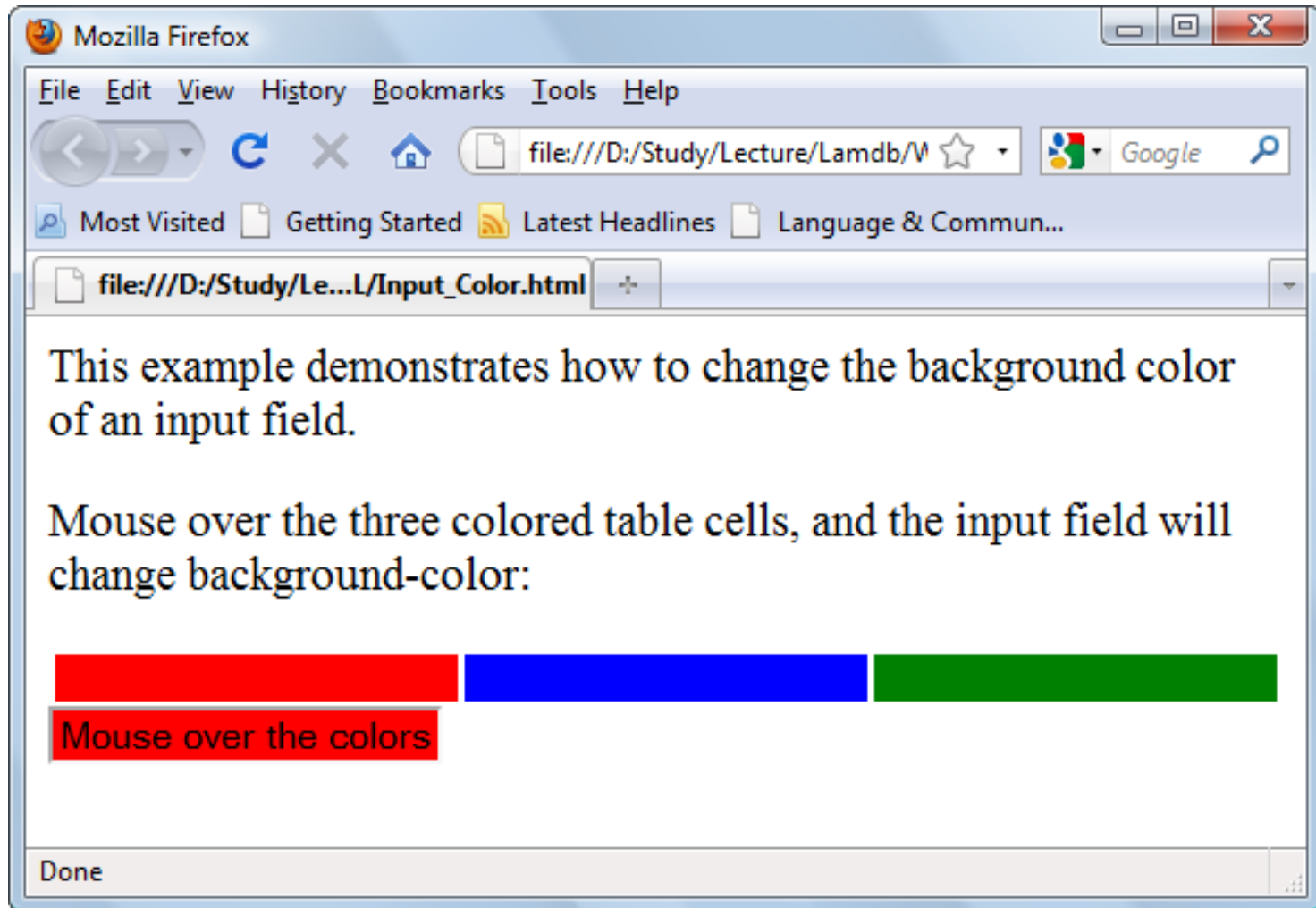
6. Mouse



7. Link



8. Input Form



9. Window

