

TÀI LIỆU

HTML, DHTML VÀ JAVASCRIPT

HA NOI 9/2008

TÀI LIỆU DÀNH CHO KHÓA HỌC

CƠ BẢN VỀ HTML, JAVASCRIPT, CSS VÀ ASP

- ❖ **Tài liệu này chứa những gì?** Tài liệu này chứa một số bài tập kèm giải thích nội dung kiến thức liên quan đến kỹ thuật lập trình Web cơ bản phía Client, bao gồm:

Phần A

Chương 0: Tạo các phần tử HTML.

Chương I: Bài tập cơ bản về JavaScript.

Chương II: Sử dụng các lớp (đối tượng) xử lý Chuỗi, Ngày tháng, Toán học.

Chương III: Xử lý sự kiện trong trang HTML với JavaScript

Chương IV: Định dạng các phần tử HTML bằng CSS

Chương V: Tạo và xử lý các tầng (Layer)

Chương VI: Nội dung động và định vị động

- ❖ **Ai nên đọc tài liệu này**

Là Sinh viên, giáo viên hoặc những người cần có kiến thức về lập trình Web.

- ❖ **Cần có những kiến thức gì trước khi đọc tài liệu này?**

Cần có kiến thức cơ bản về lập trình nói chung.

- ❖ **Giáo trình lý thuyết**

- HTML, DHTML & JavaScript của Aptech worldwide

- ❖ **Các trang web nên ghé thăm**

- www.3schools.com

- Search với từ khóa **Java Script tutorial**; **Java script Introduction**

MỤC LỤC

Chương 0: Tạo các phần tử HTML cơ bản.....	4
1. Cú pháp chung:.....	4
2. Tạo một số phần tử cơ bản.....	4
Chương I: Bài tập cơ bản về JavaScript.....	6
Chương II: Sử dụng các lớp xử lý Chuỗi, Ngày tháng, Toán học.....	9
Chương III: Xử lý sự kiện trong trang HTML với JavaScript.....	17
Chương IV: Định dạng các phần tử HTML bằng CSS.....	26
Chương V: Tạo và xử lý các tầng (Layer).....	36
Chương VI: Nội dung động và định vị động.....	42

Chương 0: Tạo các phần tử HTML cơ bản.

Mục tiêu: Kết thúc chương này, người học có thể

- Tạo các phần tử HTML cơ bản bằng cách code trực tiếp
- Dùng Notepad tạo một trang web chứa các phần tử HTML

Nội dung

Giới thiệu

Trong nhiều ứng dụng web không phải lúc nào chúng ta cũng có thể sử dụng trình soạn thảo tích hợp kiểu như Frontpage hay Dreamweaver để thiết kế giao diện, đặc biệt là khi giao diện này có liên quan đến yếu tố lập trình, khi đó người lập trình phải tạo các phần tử HTML hoàn toàn thủ công (Code chứ không dùng kéo thả). Do vậy, việc hiểu cú pháp để tạo các phần tử HTML là vô cùng quan trọng.

1. Cú pháp chung:

- ❖ **<Tên_Loại_Phần_Tử <Thuộc tính 1> = “Giá trị” <Thuộc Tính> = “Giá trị” ... >**
- ❖ **<Tên_Loại_Phần_Tử Style = “Thuộc tính: giá trị; thuộc tính : giá trị ;...; >**
- ❖ **Kết hợp cả hai cách.**

Trong đó :

Tên loại phần tử HTML	Thuộc tính
Button	Name
Text	VALUE
File	MAXLENGTH
Hidden	ReadOnly
Select	Disable
TextArea	Cols, Rows
CheckBox	Multiple
	TYPE

Phần “giá trị” có thể đặt trong cặp ngoặc kép hoặc cặp ngoặc đơn hoặc không cần !!

Nếu đặt thuộc tính theo cách 2, thì có thể tham chiếu bảng sau (gọi là theo cú pháp CSS)

2. Tạo một số phần tử cơ bản

- ❖ Tạo nút nhấn

<Input name="KiemTra" TYPE="button" VALUE="Kiểm tra dữ liệu">

<Input TYPE="Submit" VALUE="Đăng nhập">

- ❖ Tạo ô nhập

<Input name="HoVaTen" TYPE="text" VALUE="Nguyễn Văn A" size="20" MAXLENGTH="30">

<Input name="Khoa" TYPE="text" VALUE="Khoa Công nghệ thông tin" size="40" MAXLENGTH="50" readonly="true">

<Input name="Truong" TYPE="text" VALUE="Trường Đại học sư phạm kỹ thuật Hưng Yên" size="40" MAXLENGTH="50" Disabled="true">

- ❖ Tạo vùng nhập (Textarea)

<textarea name="GhiChu" cols="50" rows="5"> Nội dung ghi chú: </textarea>

❖ Tạo ô nhập Password

<Input name="MatKhau" type="password" value="123456" size="10" maxlength="20">

❖ Tạo listbox

```
<select name="MonHoc" size="5" >
    <option>Visual Basic</option>
    <option>Lập trình .NET</option>
    <option>Lập trình ASP</option>
</select>
```

❖ Tạo ComboBox (chỉ cần bỏ thuộc tính size)

```
<select name="Mon" size = 1 onChange="Call DocGiaTriListBox">
    <option value = "Visual Basic">Visual Basic</option>
    <option value = "DOT_NET">Lập trình .NET</option>
    <option value = "ASP">Lập trình ASP</option>
</select>
```

❖ Tạo hộp kiểm

<Input name="chkVB" type="checkbox" value="Visual Basic" checked> Visual Basic
<Input name="chkASP" type="checkbox" value="ASP "> Active Server Pages

❖ Tạo nút Radio

<Input name="GioiTinh" type="radio" value="Nam" checked>
<Input name="GioiTinh" type="radio" value="Nữ" >

<Input name="TinhTrang" type="radio" value="Đã lập gia đình" >
<Input name="TinhTrang" type="radio" value="Độc thân" checked >

Tên giống nhau thì sẽ thuộc về cùng một nhóm (Groups)

❖ Phần tử chọn File

<Input name="ChonFile" type="file" size="30">

❖ Tạo textbox ẩn (Hidden).

<Input name="PhanTuAn" type="hidden" value="">

Tạo các phần tử và đặt thuộc tính:

+ Tạo một textbox và đặt thuộc tính font:

```
<FONT FACE = "Times New Roman">
    <Input type = text value = "Font chữ Unicode đây !">
</FONT>
```

+ Tạo một textbox và đặt thuộc tính thông qua phong cách CSS:

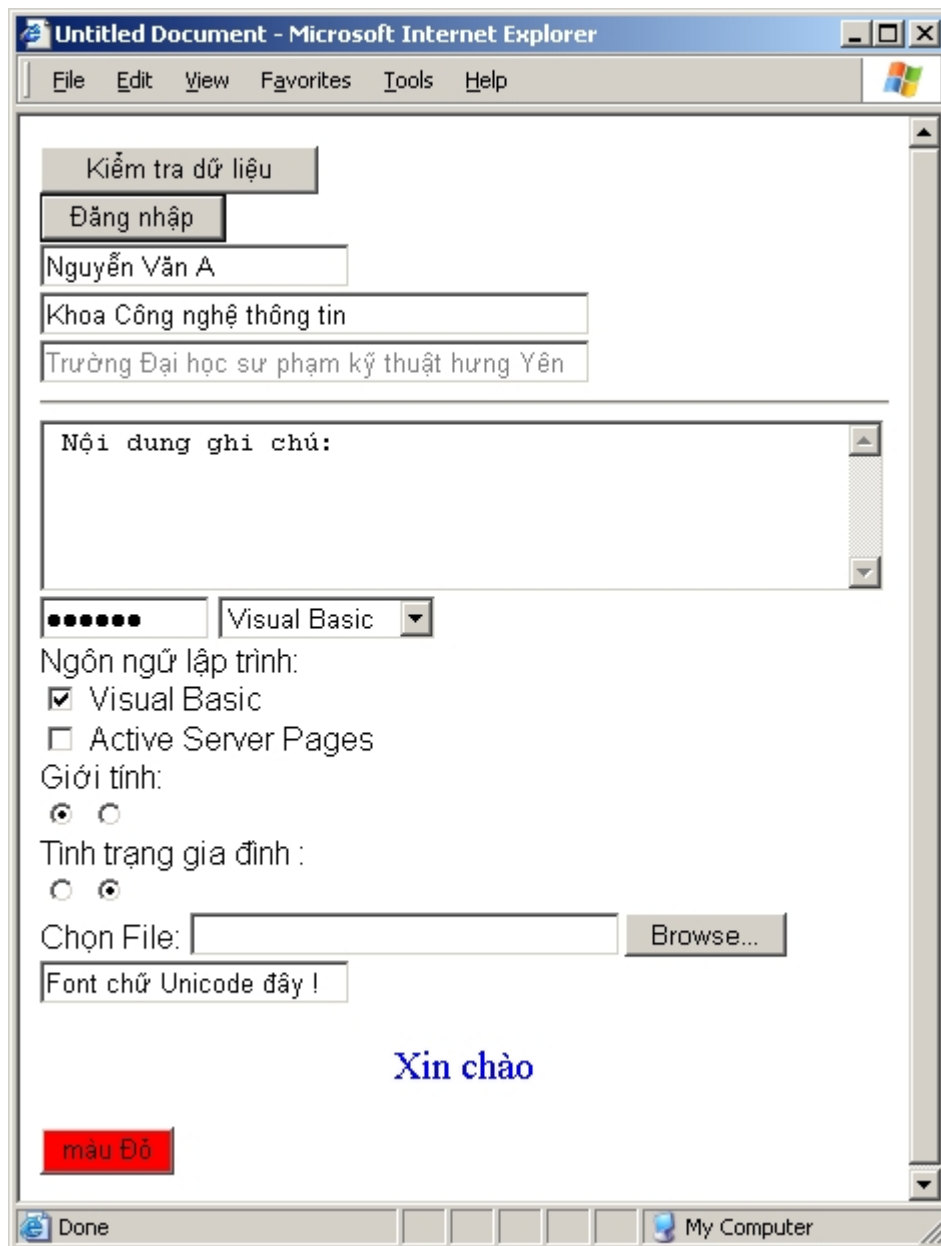
```
<Input type = text value = "Font Unicode" Style = "Font-Family:Times new
roman">
```

+ Tạo một nhãn có font chữ xanh, có hiệu ứng:

```
<P Style ="Color:Blue; font-size:20; Text-Align:center">Xin chào </p>
```

+ Tạo một nút nhấn có màu nền đỏ:

<Input type = button style="font-family:arial; background-color:red" value = "Đỏ">



Kết quả

Chương I: Bài tập cơ bản về JavaScript

Mục tiêu: Kết thúc chương này người học có thể:

- Viết các câu lệnh JavaScript và nhúng vào trang web
- Sử dụng được các đối tượng nhập xuất Prompt, **document.write**.
- Truy xuất thuộc tính của các phân tử HTML bằng câu lệnh JavaScript
- Viết lệnh xử lý một số sự kiện đơn giản.

Nội dung:

Ví dụ 1: Cho người dùng nhập vào tên và tuổi. Hãy viết lại tên và tuổi của người đó ra màn hình bằng hàm **document.write**, trong đó tên có màu đậm, tuổi được gạch chân.

Giải mẫu:

```
<HTML>
<HEAD> </HEAD>
<BODY>
<script language = "JavaScript">
    var Ten, Tuoi;           // Khai báo 2 biến để lưu tên và tuổi
    Ten = prompt("Bạn hãy nhập vào tên ", "");
    Tuoi = prompt("Bạn hãy nhập vào Tuổi : ", 20);
    document.write("Chào bạn : <B> " + Ten + "</B>");
    document.write("<BR>"); // Xuống dòng
    document.write("Tuổi của bạn là : <U> " + Tuoi + "</U>");
</script>
</BODY>
</HTML>
```

Ví dụ 2: Tạo một nút nhấn (button) có name là **welcome**, value là " **Welcome** ". Một textbox có tên là msg, value = "Welcome to".

Hướng dẫn : Sử dụng phương thức (hàm) write của đối tượng document để tạo.

Giải mẫu:

```
<HTML>
<HEAD> </HEAD>
<BODY>
<script language = "JavaScript">
    document.write("Tao Button va Text bang Script<BR>");
    document.write("<BR>");
    document.write("<input type=button name=welcome value = 'Welcome' ");
    document.write("onclick = 'alert ('Welcome to JavaScript');' > ");
    document.write("<input type = text name = msg value = 'Welcome to'>");
</script>
</BODY>
</HTML>
```

Ví dụ 3: Tạo một nút như trong ví dụ 2 và thêm chức năng sau: Khi người dùng click vào nút welcome thì hiển thị thông báo "Welcome to JavaScript !"

Hướng dẫn: Dùng thẻ để tạo nút nhấn và thêm thuộc tính onClick = "<Câu lệnh JavaScript>," (Trong đó <Câu lệnh JavaScript> có thể là một lệnh JavaScript bất kỳ, ví dụ lệnh **document.write**, **alert**, **prompt** hoặc lệnh gọi hàm v.v...)

Giải mẫu:

```
<HTML>
<HEAD> </HEAD>
<BODY>
<input type=button name= welcome value="Welcome" onclick="alert('Welcome to JavaScript');">
</BODY>
</HTML>
```

Lưu ý quan trọng: Trong JavaScript, một hằng xâu được bao bởi cặp nháy đơn hoặc nháy kép, ví dụ các xâu: 'nháy đơn', "nháy kép" là những xâu hợp lệ, tuy nhiên bạn viết : 'abc" hay "xyz' là những xâu không hợp lệ. Trong trường hợp bạn muốn in chính bản thân dấu nháy đơn hoặc nháy kép ra màn hình thì bạn đặt trước nó một ký tự \, ví dụ bạn có thể in ra màn hình dòng chữ : Women's day ra màn hình bằng hai hàm alert và document theo các cách sau đây : alert("Women's day"), **document.write**('Women\'s day'); alert("Women\'s day"); alert('Women"s day'); v.v...

Ví dụ 4: Lấy (đọc) giá trị của một phần tử HTML

Tạo 2 phần tử như trong ví dụ 2 bằng thẻ HTML, khi người dùng click chuột vào nút Welcome thì hiển thị nội dung chứa trong text có tên là **msg**.

Hướng dẫn: Để lấy giá trị của một phần tử HTML, bạn viết <Tên phần tử>.value

Ví dụ: msg.value cho ta giá trị của text tên là msg.

Giải mẫu:

```
<HTML>
<HEAD> </HEAD>
<BODY>

<input type = button name = welcome value = "Welcome" onclick = "alert(msg.value)">
<input type = text name = msg value = "Welcome to JavaScript" size = 30>

</BODY>
</HTML>
```

Ví dụ 5: Khai báo hàm trong JavaScript và cách liên kết nút nhấn với một hàm

Tạo 2 phần tử như ví dụ 2, khi người dùng nhấn nút thì gọi một hàm có tên là HienThi, hàm hiển thị có chức năng hiển thị nội dung trong text có tên là **msg** ở trên.

Hướng dẫn: Trong thẻ tạo button, bạn đặt thuộc tính onClick = "<Tên hàm>", trong trường hợp này bạn đặt onClick = "HienThi()". Điều này có nghĩa là khi người sử dụng Click chuột (OnClick = Click chuột) thì trình duyệt hãy gọi hàm HienThi(). Cũng giống như trong ngôn ngữ C, Một hàm bắt buộc phải có cặp ngoặc đơn, cho dù có tham số hay không. Ví dụ khi gọi hàm HienThi thì bạn phải viết là HienThi().

Giải mẫu:

```
<HTML>
<HEAD>
<Script Language = "JavaScript">
    function HienThi()    // Khai báo một hàm tên là HienThi
    {
        alert(msg.value);    // Lấy nội dung trong text box và hiển thị
        alert("Bạn hãy nhập vào ô text và thử lại !");
    }
</Script>
</HEAD>

<BODY>
<input type = button name = welcome value = "Welcome" onclick = "HienThi()">
<input type = text name = msg value = "Welcome to JavaScript" size = 30>
</BODY>

</HTML>
```

Lưu ý: Trong C, để khai báo một hàm thường bạn viết, ví dụ: **int HienThi()** v...v.. Tuy nhiên, với JavaScript có hơi khác tí chút, thay vào đó bạn viết **function HienThi()**

Còn các câu lệnh khác bạn viết tương tự như ngôn ngữ C đã học. Các hàm khi khai báo trong JavaScript bắt buộc phải đặt trong thẻ <Script> </Script>.

Chương II: Sử dụng các lớp xử lý Chuỗi, Ngày tháng, Toán học.

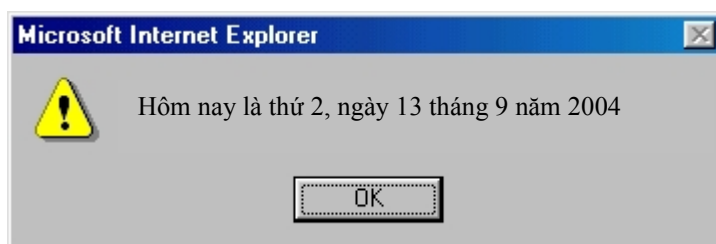
Mục tiêu: Kết thúc bài học này, người học có thể.

- Mô tả được công dụng của các lớp xử lý Chuỗi (String), xử lý ngày tháng (Date) và xử lý các hàm toán học Math.
- Sử dụng được một số phương thức, thuộc tính cơ bản của các lớp này.
- Vận dụng viết một số trang web đơn giản có sử dụng đến 3 lớp trên.

Nội dung:

Bài tập 1: Minh họa cách khai báo và sử dụng đối tượng **Date** để ngày giờ của hệ thống.

Yêu cầu: Hãy hiển thị ngày và giờ của hệ thống máy tính khi trang Web được nạp. Thông tin hiển thị ra có dạng như sau:



Hướng dẫn: Sử dụng đối tượng Date và sử dụng các hàm lấy thứ, ngày, tháng, năm để in thông tin ra màn hình. Chú ý đến các hàm tính tháng, ngày trong tuần bị hụt một đơn vị.

Bài tập 2: Minh họa sử khai báo và dùng đối tượng Date để lấy Giờ, phút, giây của hệ thống

Yêu cầu: Hiển thị Giờ và phút trong thanh tiêu đề của cửa sổ khi trang Web được nạp.

Hướng dẫn: Giá trị hiển thị trong thanh tiêu đề của trang web được lưu trong thuộc tính title của đối tượng document, do vậy để hiển thị thông tin trên thanh tiêu đề, bạn cần viết: **document.title = <Giá trị>**. Ví dụ, để hiển thị dòng chữ "Hello Every body !", bạn viết: **document.title "Hello Every body !"**

Minh họa:

```
<HTML>
<BODY>
<script language="JavaScript">
    var D = new Date();
    document.title = "Bây giờ là: " + D.getHours()+" giờ "+ D.getMinutes()+" phút.";
</script>
</BODY>
</HTML>
```

Bài 3: Vận dụng biến đổi đối tượng Date để tính tuổi của một người.

Yêu cầu : Cho người dùng nhập vào năm sinh của họ, sau đó hiển thị tuổi tương ứng.

Hướng dẫn: Sử dụng đối tượng Date để lấy năm hiện tại. Tuổi sẽ bằng năm hiện tại trừ đi năm sinh vừa nhập vào.

Minh họa mẫu:

```
<HTML>
<TITLE>Tính tuổi</TITLE>
<BODY>
<script language="JavaScript">
```

```

var D = new Date();
var NamSinh, NamHienTai;
NamHienTai = D.getYear(); // Lưu năm hiện tại vào biến
NamSinh = prompt("Bạn sinh năm bao nhiêu ? : ", "");
alert("Tuổi của bạn bây giờ là : " + (NamHienTai-NamSinh));
</script>
</BODY>
</HTML>

```

Bài 4: Tương tự như bài 3 nhưng Năm sinh nhập vào không được lớn hơn năm hiện tại.

Hướng dẫn: Sử dụng vòng lặp **do...while** để yêu cầu nhập lại nếu năm sinh > năm hiện tại.

Minh hoạ mẫu:

```

<HTML>
<TITLE>Tính tuổi</TITLE>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<BODY>
<script language="JavaScript">
var D = new Date();
var NamSinh, NamHienTai;
NamHienTai = D.getYear(); //Lưu năm hiện tại vào biến
do {
    NamSinh = prompt("Bạn sinh năm bao nhiêu : ", "");
} while (parseInt(NamSinh)>NamHienTai); //Nhập lại nếu Năm sinh>năm hiện tại

alert("Tuổi của bạn bây giờ là : " + (NamHienTai-NamSinh));
</script>
</BODY>
</HTML>

```

Bài 5: Minh hoạ cách đặt các câu lệnh JavaScript vào trong các phần tử HTML để thực thi khi người dùng click chuột và sử dụng hàm open của đối tượng window để mở trang web.

Yêu cầu: Viết đoạn Script cho người dùng nhập vào một số nguyên. Nếu người dùng nhập số 1 thì mở trang Web <http://www.vnn.vn>, nếu nhập số 2 thì mở trang <http://www.mail.yahoo.com>, nếu nhập số 3 thì mở trang <http://www.echip.com.vn>, còn nếu nhập một số khác với 1, 2 hay 3 thì mở trang <http://www.google.com>.

Hướng dẫn: Để mở một trang Web bất kỳ trong cửa sổ hiện hành bạn viết như sau:

window.open("Địa chỉ của trang cần mở").

Ví dụ : window.open(<http://www.vnn.vn>) để mở trang chủ của VNN trong cửa sổ hiện tại. Như vậy, để giải quyết yêu cầu của bài toán trên , bạn cần cho người dùng nhập vào một số và sử dụng cấu trúc **switch** để kiểm tra và mở trang web tương ứng.

Minh hoạ mẫu:

```

<HTML>
<TITLE>Mở trang web với hàm open của đối tượng window</TITLE>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<BODY>
<script language="JavaScript">
var LuaChon;
LuaChon = prompt("Bạn hãy nhập vào một số để mở trang web : ", 1);
switch (LuaChon)
{
    case "1" : window.open("http://www.vnn.vn"); break;
    case "2" : window.open("http://www.mail.yahoo.com"); break;
    case "3" : window.open("http://www.echip.com.vn"); break;
}

```

```

        default : window.open("http://www.google.com");
    }
</script>
</BODY>
</HTML>

```

Bài số 6: Minh họa việc khai báo và sử dụng biến đối tượng Array để lưu trữ danh sách và cách sử dụng các hàm của đối tượng Array như hàm **sort** và vòng lặp **for...in**

Yêu cầu: Cho người dùng nhập vào danh sách tên của một lớp, sau đó sắp xếp theo vần Alphabet rồi hiển thị danh sách đã sắp xếp đó ra màn hình, mỗi người trên một dòng.

Hướng dẫn: Sử dụng vòng lặp for để cho phép nhập danh sách họ tên và Lưu danh sách vào một mảng, sau đó sử dụng phương thức sort của đối tượng mảng để sắp xếp, tiếp theo dùng vòng lặp **for...in** để in các phần tử trong danh sách.

Minh họa mẫu:

```

<HTML>
<TITLE>Sắp xếp mảng</TITLE>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<BODY>
<script language="JavaScript">
    var SoLuong, x;
    var DS = new Array(100); // Khai báo mảng DS, có thể lưu tối đa là 100 phần tử
    SoLuong = prompt("Bạn cần nhập bao nhiêu người : ", 5);
    for (i=0; i < SoLuong; i++)
    {
        DS[i] = prompt("Nhập vào họ tên : ", "");
    }

    // Gọi hàm sort của đối tượng mảng DS để sắp xếp
    DS.sort();

    //Hiển thị kết quả sau khi sắp (sort)
    document.write("<h1>Danh sách đã sắp xếp là </h1>");

    for (x in DS)          /* Nên sử dụng cấu trúc for ... in này để duyệt mảng */
    {
        document.write( DS[x] );
        document.write("<BR>");          // Xuống dòng
    }
</script>
</BODY>
</HTML>

```

*** Nhận xét: Nếu muốn sắp theo chiều giảm dần thì sau khi sort bạn gọi hàm reverse.

Bài số 7: Minh họa việc đưa các câu lệnh JS vào trong một thẻ khi người dùng **click** chuột.

Yêu cầu: Tạo một nút nhấn (Button) có name = "DangKy", value = "Đăng ký". Khi người dùng Click vào nút này thì thông báo là **"Đăng ký dịch vụ E-Mail"**.

Hướng dẫn: Đối với các phần tử HTML, như textbox, button, checkbox, Select v.v... Các trình duyệt đều cho phép ta thực thi một hoặc nhiều câu lệnh JavaScript khi người sử dụng click chuột lên các phần tử đó. Vấn đề ở chỗ, viết các câu lệnh đó như thế nào ?

Để viết các câu lệnh JavaScript khi người dùng click chuột lên một phần tử nào đó, trong thẻ của phần tử này, ta viết như sau: **OnClick = "Các câu lệnh JavaScript"**.

"Các câu lệnh JavaScript" ở đây là bất kỳ câu lệnh JavaScript nào và chúng phải được cách nhau bởi dấu chấm phẩy. Ngoài ra, các câu lệnh phải đặt trong cặp dấu nháy kép (Hoặc nháy đơn).

Ví dụ một số cách đưa câu lệnh JavaScript cần thực thi khi người dùng click chuột

1. OnClick = "alert('Hello world');"
2. OnClick = 'document.write("Welcome to JavaScript");'
3. OnClick = "var x,y; x = 10; y = 20; alert('Tổng là : ' + (x + y)); "
4. OnClick = "var Tuoi; Tuoi = txtTuoi.value; KiemTra(Tuoi);"
5. OnClick = "KiemTra();"

Theo ví dụ trên, Khi người sử dụng Click :

- 1 : Thực hiện câu lệnh **alert('Hello world')**
- 2 : Thực hiện câu lệnh **document.write('Welcome to JavaScript');**
- 3 : Thực hiện NHIỀU câu lệnh JavaScript
- 4 : Thực hiện nhiều câu lệnh JavaScript và có lời gọi đến hàm KiemTra(Tuoi)
- 5 : Thực hiện câu lệnh gọi hàm KiemTra().

Minh hoạ mẫu:

```
<HTML>
<HEAD>
<TITLE>Minh hoạ đưa câu lệnh JavaScript vào các phần tử</TITLE>
<META http-equiv="Content-Type" content="text/html; charset=utf-8">
</HEAD>
<BODY>
  <INPUT type="button" name="DangKy" value="Đăng ký" onClick="alert('Dang ky E-Mail'); ">
</BODY>
</HTML>
```



Bài số 8: Minh hoạ cách thay đổi thuộc tính của một đối tượng thông qua việc viết các câu lệnh JavaScript.

Tạo một nút có name = ThayMauNen, value = "Thay đổi màu nền". Khi người dùng click chuột vào nút này thì thay đổi màu nền của trang Web thành màu "xanh".

Hướng dẫn: Để thay đổi màu nền của trang Web thành màu, ta cần thay đổi thuộc tính document.bgColor = "blue". (Màu đỏ là red, tím : magenta, đen: black, trắng: white, vàng: Yellow, tím nhạt: lavender). Như vậy, câu lệnh này sẽ được đặt trong phần onClick như sau:

```
<HTML>
<HEAD>
<TITLE>Thay mau nen bang click chuot</TITLE>
<META http-equiv="Content-Type" content="text/html; charset=utf-8">
</HEAD>
<BODY>
  <h2>Thay đổi màu nền sử dụng đối tượng document</h2>
  <INPUT type="button" name="ThayMauNen" value="Thay đổi màu nền"
    onClick="document.bgColor = 'blue' ">
</BODY>
</HTML>
```

Bài số 9: Minh họa việc đọc giá trị trong phần tử text và hiển thị ra màn hình

Yêu cầu: Tạo một hộp text có tên là HoTen. Một nút có tên là HienThi, value = "Hiển thị". Khi người dùng click vào nút HienThi thì hiển thị nội dung trong hộp text đó bằng hàm alert.

Hướng dẫn: Để lấy giá trị của một phần tử HTML nào đó, chúng ta viết

<Tên phần tử>.value

Trong đó: <Tên phần tử> chính là giá trị của thuộc tính name khi bạn tạo thẻ.

Ví dụ : - Hoten.value, DangKy.value, GioiTinh.value, Password.value v.v...

Minh họa :

```
<HTML>
<HEAD>
<TITLE>Đọc giá trị trong hộp text</TITLE>
</HEAD>
<BODY>
    <h2>Hãy gõ văn bản vào trong hộp text và click vào nút Hiển thị</h2>
    <INPUT type="text" name="HoTen" >
    <INPUT type="button" name="HienThi" value="Hiển thị" onClick="alert(HoTen.value); ">
</BODY>
</HTML>
```

🔗 **Lưu ý:** Khi muốn lấy giá trị của phần tử nào đó bằng JavaScript thì bạn phải đặt cho nó một cái tên, như ví dụ ở trên, để lấy giá trị trong hộp text ta đã đặt cho hộp text này tên (name) là **HoTen**.

Bài số 10: Minh họa việc thay đổi giá trị của hộp textbox

Yêu cầu: Tạo ra ba hộp text lần lượt tên là SoHang1, SoHang2, KetQua và một nút có tên là TinhTong, để thực hiện phép tính tổng. Khi người dùng nhập hai số hạng vào hộp SoHang1 và SoHang2, sau đó click vào nút TinhTong thì kết quả tổng sẽ được lưu vào trong hộp text KetQua.

Hướng dẫn: Để thay đổi giá trị một thuộc tính nào đó của phần tử HTML, bạn viết theo cách sau:

<Tên của phần tử>.<Tên thuộc tính> = <Giá trị mới>

Trong đó: Tên phần tử chính là giá trị của thuộc tính name khi bạn tạo thẻ.

Ví dụ: HoTen.value = "Đây là văn bản mới", DangKy.value = "Sign Up now", v.v...

```
<HTML>
<HEAD>
<TITLE>Thay đổi giá trị của thuộc tính</TITLE>
<META http-equiv="Content-Type" content="text/html; charset=utf-8">
</HEAD>
<BODY>
    <h2>Hãy nhập hai số và click vào nút Tính tổng</h2>
    <INPUT type="text" name="SoHang1" > +
    <INPUT type="text" name="SoHang2"> =
    <INPUT type="text" name="KetQua">
    <INPUT type="button" value="Tính tổng"
        onClick="KetQua.value = parseFloat(SoHang1.value) + parseFloat(SoHang2.value)">
</BODY>
</HTML>
```

🔗 **Lưu ý:** - Giá trị lưu trong hộp text luôn là một chuỗi, do vậy để thực hiện phép cộng được đúng, bạn cần phải chuyển giá trị sang dạng số bằng hàm parseFloat (Hoặc parseInt) như ở trên.

- Việc thay đổi này có thể áp dụng cho các phần tử khác như button, checkbox, v.v...



Bài tập 11: Minh họa việc gọi hàm khi người dùng click vào một nút

Yêu cầu: Tạo ra 4 text có tên lần lượt là : MauNen, MauChu, TieuDe, TrangThai và một nút có tên là ThayDoi, value là "Thay đổi". Khi người dùng click vào nút ThayDoi thì màu nền, màu chữ, tiêu đề của tài liệu và thanh trạng thái của cửa sổ trình duyệt sẽ được thay đổi bằng các giá trị trong text tương ứng

Hướng dẫn: Bạn hoàn toàn có thể viết nhiều câu lệnh trong thuộc tính OnClick như các ví dụ trước, tuy nhiên nếu có nhiều lệnh thì chương trình trông không được sáng sủa cho lắm. Khi đó bạn có thể nhóm các câu lệnh vào trong một hàm và trong thuộc tính OnClick bạn chỉ việc gọi hàm này ra.

Minh họa mẫu

```
<HTML>
<HEAD> <TITLE>Thay đổi thuộc tính của trang Web</TITLE> </HEAD>
<SCRIPT language="JavaScript">
    function CapNhat( )
    {
        document.title = TieuDe.value;           /* Thay đổi tiêu đề của trang Web */
        document.bgColor = MauNen.value;          /* Thay đổi màu nền của trang */
        document.fgColor = MauChu.value;          /* Thay đổi màu chữ của trang */
        window.defaultStatus = TrangThai.value; /* Thay đổi dòng trạng thái của cửa sổ */
    }
</SCRIPT>
<BODY>
    <h2>Nhập vào các giá trị và nhấn nút Thay đổi</h2>
    <INPUT type="text" name="TieuDe" value="Tiêu đề mới">
    <INPUT type="text" name="MauNen" value="Nhập màu vào đây (ví dụ blue)"> <BR>
    <INPUT type="text" name="MauChu" value="Nhập màu chữ vào đây (ví dụ white)">
    <INPUT type="text" name="TrangThai" value="Nhập dòng trạng thái vào đây "> <BR>
    <INPUT type="button" name="ThayDoi" value="Thay đổi" onClick="Ham( );">
</BODY>
</HTML>
```

Nhận xét: Ở ví dụ trên, khi người dùng click chuột lên nút **ThayDoi** thì hàm **CapNhat()** sẽ được gọi.



Đối tượng String

Bất kỳ một biến xâu hoặc một hằng xâu đều được coi là một đối tượng xâu.

Ví dụ ta có: `var s = "Hung Yen-Aptech", "JavaScript"` hay

`var x = new String("Welcome to Aptech")` thì biến `s`, `x` và hằng `"Hung Yên - Aptech"` đều là các đối tượng xâu và đều có các phương thức và thuộc tính dưới đây.

	Tên thuộc tính	ý nghĩa	Ví dụ
Thuộc tính	length	Cho biết độ dài của một xâu x	<code>var x = "abc"; alert(x.length); // 3 alert("Aptech".length); // 6</code>
Phương thức	Tên phương thức	ý nghĩa	Ví dụ
	charAt(n)	Cho ta ký tự tại vị trí: n	<code>alert(s.charAt(0)); //H</code>
	indexOf(x)	Cho ta vị trí xuất hiện của xâu s trong xâu s. Nếu không thấy thì vị trí trả về là -1.	<code>s.indexOf("Aptech") -> 9 "Hello".indexOf("e") -> 1 "Java".indexOf("C") -> -1</code>
	lastIndexOf(x)	Cho ta vị trí cuối cùng của xâu x trong xâu s	<code>s.lastIndexOf("n") -> 7 "Hello".lastIndexOf("l") -> 3</code>
	substring(n1, n2)	Lấy ra một xâu con trong xâu s, lấy từ vị trí n1 đến n2 (số ký tự lấy ra là n2-n1 ký tự)	<code>s.substring(0,3) -> "Hun" s.substring(2,4) -> "ng" "Hello".substring(2,5) -> "llo"</code>
	toLowerCase()	Chuyển xâu s thành chữ thường	<code>s.toLowerCase() -> "hung yen-apttech" "Hello".toLowerCase()->"hello"</code>
	toUpperCase()	Chuyển xâu s thành chữ HOA	<code>s.toUpperCase() -> "HUNG YEN-APTECH" "Hello".toUpperCase() -> "HELLO"</code>
	big()	In to xâu s	<code>document.write(s.big()) document.write("abc".big())</code>
	bold()	In đậm xâu s	<code>document.write(s.bold()) document.write("abc".bold())</code>
	fontcolor(m)	In xâu s với màu m. ví dụ màu: "red", "blue" m"magenta" v.v..	<code>document.write(s.fontcolor("blue")) document.write("abc".fontcolor("red"))</code>
	fontSize(n)	In xâu s với kích cỡ font là n	<code>document.write(s.fontSize(30)) document.write("Java".fontSize(20))</code>
	strike()	In xâu s với đường gạch ngang	<code>document.write(s.strike())</code>
	sub()	In xâu s ở dưới dòng hiện tại	<code>document.write(s.sub()) document.write("H"+"2".sub()+"O")//H₂O</code>
	sup()	In xâu s ở trên dòng hiện tại	<code>document.write(s.sub()) document.write("x" + "2".sup());// -> x²</code>
	anchor(A)	Tạo một điểm neo, có tên là A, phần hiển thị là s.	<code>document.write(s.anchor("TOP"))</code>
	link(A)	Tạo một liên kết đến điểm neo A, phần hiển thị là s	<code>document.write(s.link("#TOP")) document.write("Về đầu trang".link("#TOP"))</code>

Đối tượng Math

Với đối tượng Math, Khi khai báo biến thuộc đối tượng này, bạn không được viết dạng như : `var m = new Math();` Khi muốn sử dụng các thuộc tính và phương thức của đối tượng này bạn gọi trực tiếp các thuộc tính và phương thức, ví dụ: `Math.sin(3.14)`, `Math.PI`, `Math.abs(x)` v.v...

	Tên thuộc tính	ý nghĩa	Ví dụ
Thuộc tính	PI	Cho ta hằng số PI (tức 3.14159)	<code>var BanKinh = 10; alert("Diện tích hình tròn là : " + Math.PI * BanKinh*BanKinh);</code>
	E	Cho ta hằng số E (= 2.718...)	<code>alert("Hằng số E là: " + Math.E)</code>

	SQRT2	Cho ta căn bậc 2 của 2 : (=1.4142)	alert("Căn bậc 2 của 2 = " + Math.SQRT2);
	SQRT1_2	Cho ta (căn bậc 2 của 2) / 2	alert("Căn bậc 2 của 2 / 2 = " + Math.SQRT1_2);
Phương thức	Tên phương thức	ý nghĩa	Ví dụ
	abs(x)	Cho ta trị tuyệt đối của x	alert(Math.abs(-19)); //-> 19 alert(Math.abs(-1.5)); //-> 1.5
	sin(x), cos(x)	Tính sin và cos của x	alert("Sin(1.5) = " + Math.sin(1.5)); alert("Cos(0) = " + Math.cos(0));
	sqrt(x)	Tính căn bậc hai của x	alert("Căn 16 = " + Math.sqrt(16)); //4
	pow(x,y)	Tính x ^y	alert("6^2="+ Math.pow(6,2)); //->36 alert("9^0.5="+ Math.pow(9,0.5)); //3
	round(x)	Làm tròn số x. Nếu phần lẻ sau phần thập phân >= 0.5 thì bỏ phần thập phân và cộng thêm 1. Trái lại thì bỏ phần thập phân nhưng và không cộng gì	alert(Math.round(3.5)); //->4 alert(Math.round(3.6)); //->4 alert(Math.round(3.49)); //->3
	max(a,b)	Cho ta giá trị lớn nhất trong hai số a và b	var a = 10, b = 100; alert("Max(a,b) = ",Math.max(a,b)); //100 alert(Math.max(-1,2)); //->2
	min(a,b)	Cho ta giá trị nhỏ nhất trong hai số a và b	var a = 10, b = 100; alert("Min(a,b)=",Math.min(a,b)); //10 alert(Math.min(-1,2)); //->-1
	ceil(x)	Làm tròn số x, Nếu số x có phần thập phân thì phần thập phân bị cắt đi sau đó cộng thêm 1 vào x	var x = 1.1, y = 2.5, z = 4.8; alert(Math.ceil(x), Math.ceil(y),Math.ceil(z)); //->235
	floor(x)	Làm tròn số x, nếu x có phần lẻ thập phân thì bị cắt đi, chỉ lấy phần nguyên.	var x = 1.1, y = 2.5, z = 4.8; alert(Math.floor(x), Math.floor(y),Math.floor(z)); //->124

Đối tượng Date

Khai báo biến thuộc đối tượng Date như sau: var <Tên biến> = new Date();
Ví dụ ngày, giờ hiện tại là thứ hai 20/12/2004, 6h30' 20", ta có các kết quả sau:

Tên phương thức	ý nghĩa	Ví dụ
getDay()	Lấy thứ hiện tại trong tuần (Chủ nhật ứng với 0, thứ hai ứng với 1, ..., thứ 7 ứng với 6)	var D = new Date(); alert(D.getDay()); //-> 1
getDate()	Lấy ngày hiện tại	alert(D.getDate()); //->20
getMonth()	Lấy tháng hiện tại (0->tháng 1, 1-> tháng 2)	alert(D.getMonth()); //->11
getYear()	Lấy năm hiện tại	alert(D.getYear()); //->2004
getHours()	Lấy giờ hiện tại (Tính theo 24 h)	alert(D.getHours()); //->6
getMinutes()	Lấy phút hiện tại	alert(D.getMinutes()); //->30
getSeconds()	Lấy giây hiện tại	alert(D.getSeconds()); //->20
setDate(n)	Đặt ngày là n	D.setDate(10); alert("Bây giờ: "+D.getDate()); //10

các phương thức setYear(n), setHours(n) cũng làm tương tự. Chú ý, số n phải là số nguyên. và việc set đó chỉ làm thay đổi giá trị ngày, tháng năm, giờ, phút, giây ... của đối tượng Date chứ không làm thay đổi ngày giờ của hệ thống máy tính.

Chương III: Xử lý sự kiện trong trang HTML với JavaScript

Mục tiêu của chương:

- Giúp học viên nhận biết được khi nào sự kiện xảy ra
- Viết các câu lệnh JavaScript đặt vào các sự kiện khi nó xảy ra
- Vận dụng linh hoạt vào viết chương trình

Nội dung:

1. Nhắc lại khái niệm sự kiện (event)

Sự kiện là những hành động do người dùng hoặc hệ thống gây ra. Các hành động do người dùng gây ra có thể là di chuyển chuột, nhấn chuột, nhả chuột, nhấn phím, nhả phím, copy, kéo giãn cửa sổ, di chuyển cửa sổ v.v... Các sự kiện do hệ thống gây ra có thể là nạp tài liệu, đóng cửa sổ v.v...

Khi sự kiện xảy ra, nó sẽ tự động thực thi các câu lệnh JavaScript tương ứng với sự kiện đó (nếu chúng ta đã định nghĩa chương trình xử lý sự kiện tương ứng).

2. Bảng liệt kê các sự kiện và tên tương ứng

Mỗi sự kiện khi xảy ra chúng đều có một cái tên và thường bắt đầu bằng từ **on**, ví dụ như onClick, onChange.... cụ thể được mô tả như trong bảng dưới đây:

Tên Sự kiện	Chỉ áp dụng cho phần tử	Mô tả
Onabort	Image	Được kích hoạt khi người sử dụng huỷ bỏ việc tải một hình ảnh bằng cách kích vào một kết nối hoặc nút Stop
Onblur	Window, frame, all form element	Khi phần tử bị mất focus
OnClick	Button, radio button, check box, submit button, reset button, link	được kích hoạt khi người sử dụng kích trái chuột vào phần tử.
OnChange	Text field, textarea, select list	Nó được kích hoạt khi người sử dụng thay đổi giá trị của phần tử.
Onfocus	Window, frame, all form element	Nó được kích hoạt khi người sử dụng đặt focus vào một cửa sổ, khung, hay phần tử form
Onload	Document, applet, frameset, img, link, object, script, style, window	Nó được kích hoạt khi tài liệu được trình duyệt nạp xong.
Onmousedown	Button, document, link	Nó được kích hoạt khi người sử dụng ấn nút con chuột
Onmouseout	Area, layer, link	Nó được kích hoạt khi người sử dụng di chuyển con trỏ ra khỏi một phần tử.
Onmouseover	Area, layer, link	Nó được kích hoạt khi người sử dụng di chuyển con trỏ khắp một phần tử.
Onmouseup	Button, document, link	Nó kích hoạt khi người sử dụng nhả nút con chuột đã được ấn.
Onreset	Form	Khi người sử dụng click vào nút reset form
Onresize	Window, frame	Nó kích hoạt khi người sử dụng kéo giãn cửa sổ hoặc một khung.
onsubmit	Form	Nó được kích hoạt khi người sử dụng

		dụng click vào nút submit của form.
onunload	Document, frameset, image, window	Nó được kích hoạt khi người sử dụng chuyển sang (mở) một trang khác.

Vậy áp dụng tên các sự kiện đã liệt kê ở trên như thế nào ?

Nếu bạn đã biết khi nào một sự kiện xảy ra thì bạn hoàn có thể thực thi các câu lệnh JavaScript tương ứng với sự kiện đó.

Cú pháp khai báo để trình duyệt **thực thi các câu lệnh JavaScript khi một sự kiện xảy ra** như sau:

a/ Cách 1: <Tên thẻ <Tên sự kiện> = " <Một Câu lệnh JavaScript>" >

Lưu ý: **Một câu lệnh JavaScript** có thể là bất kỳ câu lệnh nào mà bạn đã học. Câu lệnh này phải được đặt trong cặp nháy kép (hoặc cặp nháy đơn).

Ví dụ 1 :

```
<Input onClick = "alert('Bạn đã click vào textbox');">
```

Ví dụ 2:

```
<Input type=button value="Thử" onMouseMove = "alert('Bạn di chuột');">
```

Ví dụ 3:

```
<P onClick = "window.status='Văn bản bị click chuột';"> Hello ! </P>
```

Ví dụ 4:

```
<Input type = submit value = "Gửi" onclick = "alert('Đã được gửi');">
```

Ta hãy đi phân tích ví dụ 1. Trong ví dụ này ta đã tạo ra một textbox và viết (khai báo) sự kiện click như sau : **onClick = "alert('Bạn đã click vào textbox');"**

Ở đây có 2 phần:

- Phần **onClick** : là tên của sự kiện click chuột (xin tham khảo ở bảng trên).
- Phần thứ 2 sau dấu =, là một câu lệnh JavaScript tương ứng sẽ được thực thi khi sự kiện click chuột xảy ra đối với textbox đó. ở đây là câu lệnh alert.

Điều này có nghĩa là, bất cứ khi nào người dùng click chuột vào textbox này thì trình duyệt sẽ tự động thực thi câu lệnh **alert('Bạn đã click vào textbox');**

✎ **Kết luận:** Nếu chúng ta muốn trình duyệt thực thi một câu lệnh nào đó khi một sự kiện xảy ra thì cần khai báo trong phần định nghĩa thẻ như sau:
<Tên sự kiện> = "<Câu lệnh JavaScript cần thực thi>"

- Tương tự trong ví dụ 2: Bất cứ khi nào người dùng di chuyển chuột trong textbox (tên sự kiện là onMouseMove) thì lệnh "alert('Bạn di chuột');" sẽ được thực thi.
- Trong ví dụ 3: Bất cứ khi nào bạn click chuột vào dòng chữ "Hello !" thì thanh trạng thái của cửa sổ sẽ có dòng chữ : "Văn bản bị click chuột"
- Trong ví dụ 4: Theo bạn, thông báo "Đã được gửi" khi nào thì xuất hiện !?

b/ Cách 2: Bạn có thể không chỉ viết một câu lệnh khi một sự kiện xảy ra đối với một phần tử nào đó mà JavaScript còn cho phép bạn thực thi nhiều câu lệnh đồng thời, với điều kiện các câu lệnh này phải được phân cách nhau bởi dấu chấm phẩy ";".

Cú pháp viết như sau:

```
<Tên thẻ <Tên sự kiện>=<Câu lệnh 1>; <Câu lệnh 2>; ...; <Câu lệnh n>" .... >
```

Ví dụ 1:

```
<input onclick="window.status='Click chuột'; alert('Bạn đã click chuột');">
```

Trong ví dụ này, ta đã tạo một textbox và khi người dùng click chuột vào textbox này thì trình duyệt sẽ thực thi 2 câu lệnh tương ứng như ta đã chỉ ra trong thẻ :

window.status='Click chuột' và alert('Bạn đã click chuột'). 2 lệnh này được phân cách nhau bởi dấu chấm phẩy.

Ví dụ 2:

```
<input name=Hoten onFocus="Hoten.value=' ' ; window.status='Họ tên đã nhận focus' ; window.document.title = 'Nội dung trong textbox đã bị xóa' ">
```

Trong ví dụ này ta cũng tạo ra một textbox và khi textbox này nhận được focus (click chuột) thì trình duyệt sẽ tự động thực thi 3 câu lệnh :

- Hoten.value=' '
- window.status='Họ tên đã nhận focus'
- window.document.title = 'Nội dung trong textbox đã bị xóa'

Nhận xét: Nếu số câu lệnh cần thực thi khi một sự kiện xảy ra là ít (Một hoặc hai câu lệnh) thì ta có thể khai báo đoạn chương trình xử lý sự kiện sử dụng theo cách 1 hoặc cách 2. Còn trong trường hợp số câu lệnh cần xử lý là lớn, thì cách nên sử dụng cách khác mà ta sẽ đề cập dưới đây.

C/ Cách 3: Gọi một hàm khi một sự kiện xảy ra.

Về bản chất cách này chính là **cách một**, có điều câu lệnh là một lời gọi hàm.

Cách này thường được sử dụng khi :

- Số lệnh cần thực thi khi một sự kiện xảy ra là lớn
- Đảm bảo cho chương trình sáng sủa và dễ đọc, dễ quản lý và bảo trì

Cú pháp khai báo hàm trong định nghĩa sự kiện như sau:

```
<Tên thẻ <tên sự kiện> = "Tên hàm cần gọi([Tham số nếu có])" .... >
```

Ví dụ:

```
1/ <input onclick ="Ham1()" >
```

```
2/ <input type = button value = Gui onclick = "GuiThongTin()">
```

```
3/ <input type = radio name = GT onclick = "KiemTra()">
```

Trong đó, Ham1(), GuiThongTin() và KiemTra() là các hàm.

Tóm lại: Tùy vào trường hợp cụ thể mà khi một sự kiện xảy ra, bạn có thể viết một lệnh, nhiều lệnh hoặc một hàm tương ứng sẽ được thực thi trong định nghĩa thẻ. Tuy nhiên, một quy tắc chung là:

Nếu đoạn chương trình xử lý sự kiện chỉ có một lệnh thì nên viết theo cách a, còn trái lại thì nên viết các lệnh trong một hàm (tức theo cách viết b).

3. Một số bài tập minh họa

Ví dụ 1: Hãy tạo một nút nhấn (button) có value = "Thử". Khi người dùng click vào nút này thì tiêu đề của cửa sổ sẽ là "Bạn đã click chuột"

Hướng dẫn: Trước hết ta cần xác định xem các lệnh nào cho phép ta thay đổi tiêu đề của cửa sổ thành "Bạn đã click chuột", tiếp theo là đặt các lệnh đó vào đâu để khi người dùng click chuột thì nó được thực thi theo như yêu cầu bài toán

- Lệnh để thay đổi tiêu đề như sau: **document.title = "Bạn đã click chuột"**
- Như ta đã biết khi người dùng click thì sự kiện **onClick** xuất hiện, do vậy câu lệnh trên sẽ được đặt tương ứng vào sự kiện onClick, như sau:

```
<HTML>
<HEAD>
<TITLE>Hay click vào nút ở dưới và quan sát tiêu đề</TITLE>
</HEAD>
<BODY>

<input type=button value="Thử" onClick="document.title='Ban da click chuot' ">

</BODY>
</HTML>
```

Ví dụ 2: Tạo một trang Web, có 2 phần tử : Phần tử button có value = "Gửi", và một phần tử textbox. Yêu cầu: khi người dùng click vào nút Gửi thì thông báo trên màn hình là : "Bạn đã click vào nút gửi" còn khi người dùng click vào textbox thì thông báo là "Bạn đã click vào textbox".

Hướng dẫn: Theo yêu cầu của bài thì dòng thông báo "Bạn đã click chuột vào nút gửi" xuất hiện chỉ khi người dùng click chuột vào nút gửi, do vậy các lệnh thực hiện hiển thị thông báo sẽ được **đặt trong sự kiện onclick của nút nhấn**. Còn dòng thông báo "Bạn đã click chuột vào text box" khi người dùng click chuột vào textbox, do vậy các lệnh thực hiện hiển thị dòng thông báo sẽ được **đặt trong sự kiện onclick của textbox** :

Minh họa:

```

<HTML>
<HEAD>
<TITLE>Hay click vào nút và textbox o duoi va quan sat tieu de</TITLE>
</HEAD>
<BODY>

<input type=button value="Gui" onClick="alert( 'Ban da click chuột vào nút') ">
<input type=text onclick = "alert('Ban da click chuột vào textbox') ">

</BODY>
</HTML>

```

Ví dụ 3: Tạo 2 nút, nút thứ nhất có value = "Xanh", nút thứ hai có value = "Đỏ". **Yêu cầu:** Khi người dùng click vào nút xanh thì màu nền của tài liệu là xanh (blue), còn khi người dùng click vào nút đỏ thì màu nền của tài liệu là: Đỏ (red).

Hướng dẫn: Thuộc tính màu nền của tài liệu được lưu trong thuộc tính bgColor của đối tượng document. Thuộc tính này có thể thay đổi được.

Minh họa:

```

<HTML>
<HEAD>
</HEAD>
<BODY>

<input type=button value=Xanh onclick="window.document.bgColor = 'blue'; ">
<input type = button value =Đỏ onclick="window.document.bgColor = 'red'; ">

</BODY>
</HTML>

```

Ví dụ 4:

Tạo một danh sách lựa chọn gồm có 4 màu: red, blue, brown và lavender. Khi người dùng chọn một màu thì màu nền của tài liệu sẽ thay đổi tương ứng.

Hướng dẫn: Để thay đổi màu nền của tài liệu ta làm tương tự như ví dụ 3

Minh họa:

```

<HTML>
<HEAD>
</HEAD>
<BODY>
<script language = JavaScript>
    function DoiMau()
    {
        document.bgColor = Mau.value;
        // Hoặc viết: window.document.bgColor = Mau.value;
    }
</script>
Bạn hãy chọn màu nền:
<Select name = Mau onchange = "DoiMau();" >
    <option value = red>   Màu đỏ   </option>
    <option value = blue>  Màu xanh </option>
    <option value = brown> Màu nâu   </option>
    <option value = lavender> Màu xanh nhạt</option>
</select>

</BODY>
</HTML>

```

Ví dụ 5 : Tạo một textarea có tên là NoiDung, một Textbox có tên là : SoKyTu. Với yêu cầu như sau: Khi người sử dụng gõ các phím vào trong textarea thì số lượng ký tự (Độ dài xâu) chứa trong textarea đó sẽ được hiển thị trong textbox. Nếu số lượng ký tự trong textarea gõ vào vượt quá 200 ký tự thì thông báo : "Bạn đã gõ quá số ký tự cho phép !".

```

<html>

```

```

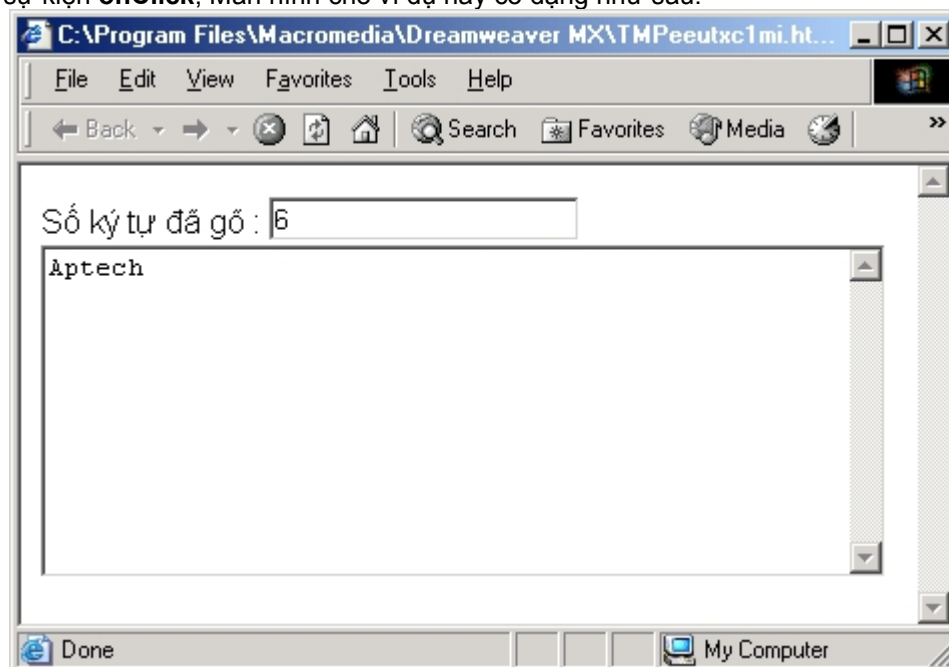
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<script language="JavaScript">
    function KiemTra()
    {
        if (NoiDung.value.length > 200) alert("Bạn đã gõ quá số ký tự cho phép !");
        SoKyTu.value = NoiDung.value.length; // Hiển thị số ký tự trong textbox SoKyTu
    }
</script>
<body style="font-family:arial">

Số ký tự đã gõ : <input type="text" name="SoKyTu"> <BR>

<textarea name="NoiDung" cols=50 rows="10" onKeyUp="KiemTra();" > </textarea>
</body>
</html>

```

Ở ví dụ trên: Hàm kiểm tra sẽ được gọi mỗi khi sự kiện nhấn phím (onKeyUp) xuất hiện hay nói cách khác là khi người dùng gõ thêm một ký tự vào trong textarea. Ở đây ta không đặt hàm kiểm tra vào trong sự kiện **onClick**; Màn hình cho ví dụ này có dạng như sau:



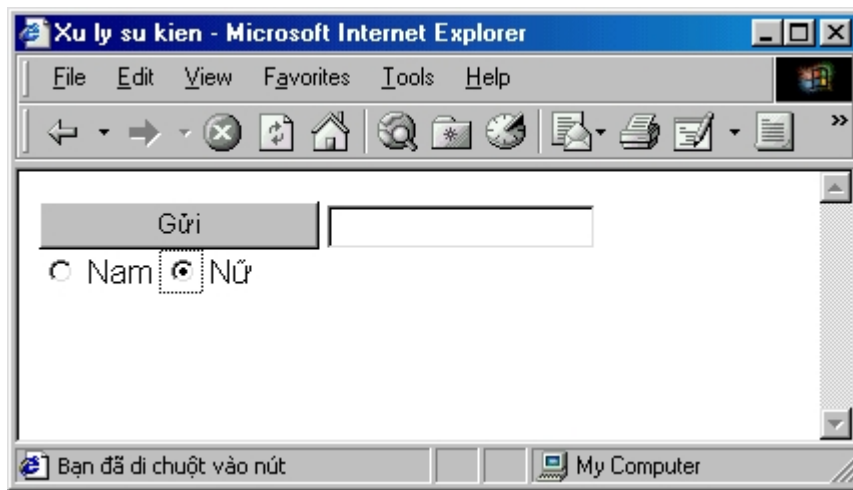
[<Xem kết quả>](#)

Ví dụ 6: Tạo một nút có value = "Gửi", textbox có name = "HoTen", 2 nút radio có tên là GioiTinh và nhãn tương ứng là Nam, nữ.

Yêu cầu: Khi người dùng di chuyển chuột vào phần tử nào thì hiển thị thông báo tương ứng dưới thanh trạng thái. Ví dụ nếu người sử dụng di chuyển chuột qua nút nhấn "Gửi" thì thanh trạng thái sẽ là "Bạn đang di chuyển chuột vào nút"...

Hướng dẫn: Khi người dùng di chuyển chuột thì sự kiện di chuyển chuột sẽ xuất hiện, sự kiện này có tên là : **onMouseMove**. Vậy ta sẽ viết lệnh trong sự kiện này.

Minh họa: màn hình



```
<html>
<head>
<title>Xu ly su kien</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<body style="font-family:arial">

<input type="button" value="Gửi" onmousemove="window.status = 'Chuột trong nút'; ">
<input onMouseMove="window.status='Chuột trong textbox';"> <BR>
<input type="radio" name="GioiTinh" onMouseMove="window.status='Trong nam';">Nam </option>
<input type="radio" name="GioiTinh" onMouseMove="window.status='Trong nữ';"> Nữ </option>

</body>
</html>
```

[<Xem kết quả>](#)

Ví dụ 7:

Tạo 3 textbox, có tên lần lượt là: SoLuong (Số lượng), DonGia (Đơn giá) và ThanhTien (Thành tiền);
Yêu cầu: Khi người dùng nhập giá trị trong DonGia thì kết quả sẽ được cập nhật ngay trong ThanhTien.

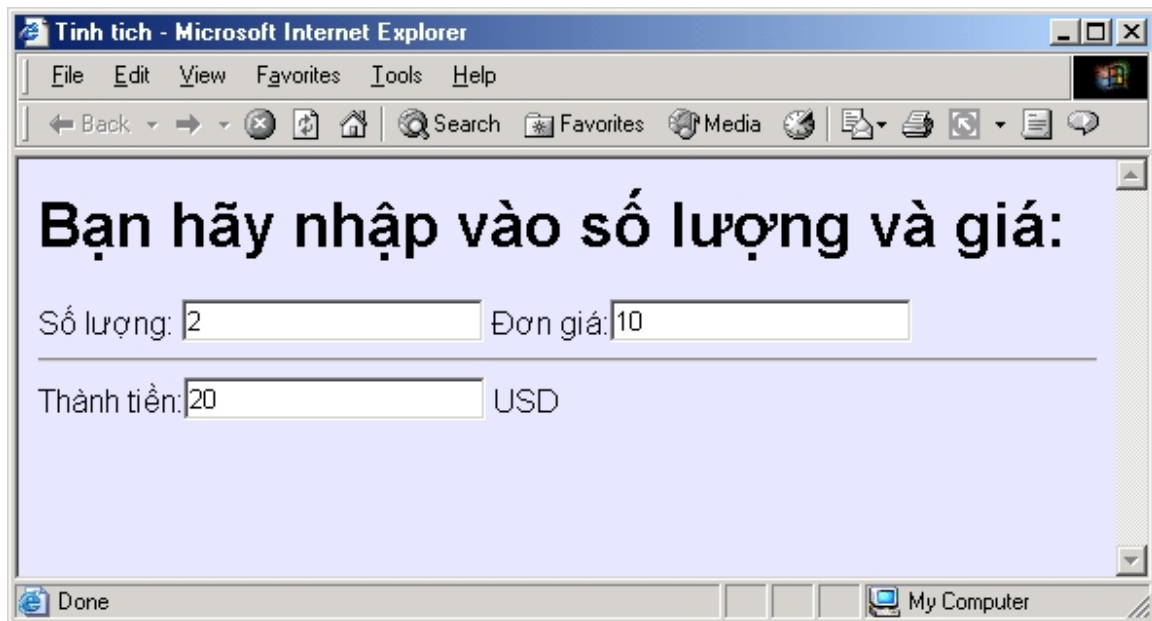
Hướng dẫn: Khi người dùng nhập giá trị trong textbox DonGia bằng cách nhấn các phím số thì sự kiện nhấn phím xuất hiện (sự kiện nhấn phím có tên là onKeyUp), do vậy ta sẽ viết các lệnh đáp ứng với sự kiện này. Các lệnh ở đây chỉ có một do vậy nên đặt ngay trong định nghĩa thẻ, như sau:

```
<html>
<head>
<title>Tinh tich</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>

<body style="font-family:arial;background-color:lavender">
<H1>Bạn hãy nhập vào số lượng và giá:</H1>

Số lượng: <input name="SoLuong">
Đơn giá:<input name="DonGia" onKeyUp="ThanhTien.value=SoLuong.value*DonGia.value"> <HR>
Thành tiền:<input name="ThanhTien"> USD

</body>
</html>
```



[<Xem kết quả>](#)

Ví dụ 8: Tương tự như ví dụ 7, nhưng viết theo cách khai báo 3 (Các lệnh viết trong hàm). Kết quả vẫn cho ta như ví dụ 7

```
<html>
<head>
<title>Tinh tich</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<Script language = JavaScript>
function TinhToan()
{
    ThanhTien.value=SoLuong.value*DonGia.value      ;

    // Hoặc bạn viết đầy đủ là:
    // window.ThanhTien.value = window.SoLuong.value*window.DonGia.value
}

</head>

<body style="font-family:arial;background-color:lavender">
<H1>Bạn hãy nhập vào số lượng và giá:</H1>

Số lượng:  <input name="SoLuong">
Đơn giá:      <input name="DonGia" onKeyUp="TinhToan();"> <HR>
Thành tiền: <input name="ThanhTien"> USD

</body>
</html>
```



Ví dụ 9: Tạo ra một trang Web đăng nhập vào trang Vinaphone để cho phép người gửi tin nhắn đến điện thoại di động.

hướng dẫn: Để đăng nhập vào một trang Web nào đó, những thông tin bắt buộc thường là UserName và password (mật khẩu). Tuy nhiên, ngoài những thông tin bắt buộc đó chúng ta còn phải gửi các thông tin phụ. Những thông tin phụ này người dùng người dùng không phải nhập. (các thông tin phụ đó dưới đây sẽ được gạch chân)

Mình họa:



```
<HTML>
<HEAD>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Đăng nhập vào trang http://sms.vinaphone.vnn.vn</title>
</HEAD>
<BODY style="font-family:arial">
<H2>Đăng nhập vào trang Web của vinaphone </H2><HR>

<form action="http://sms.vinaphone.vnn.vn/sms/servlet/UserInterface" method=post>
  User Name      Password <BR>
  <input type="text" id = username name="username" size="20">
  <input type="text" id= password name="password" size="20">
  <input type="hidden" name="id" value="0" >
  <input type="hidden" name="language" value="en">
```



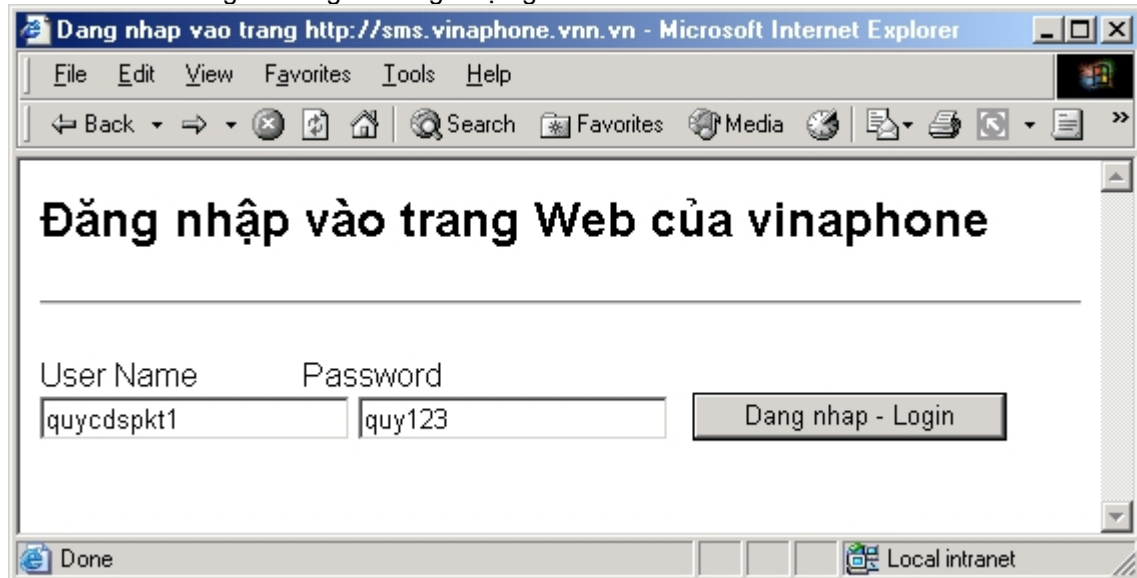
```

<input type="submit" value="Dang nhap - Login">
</p>
</form>

</BODY>
</HTML>

```

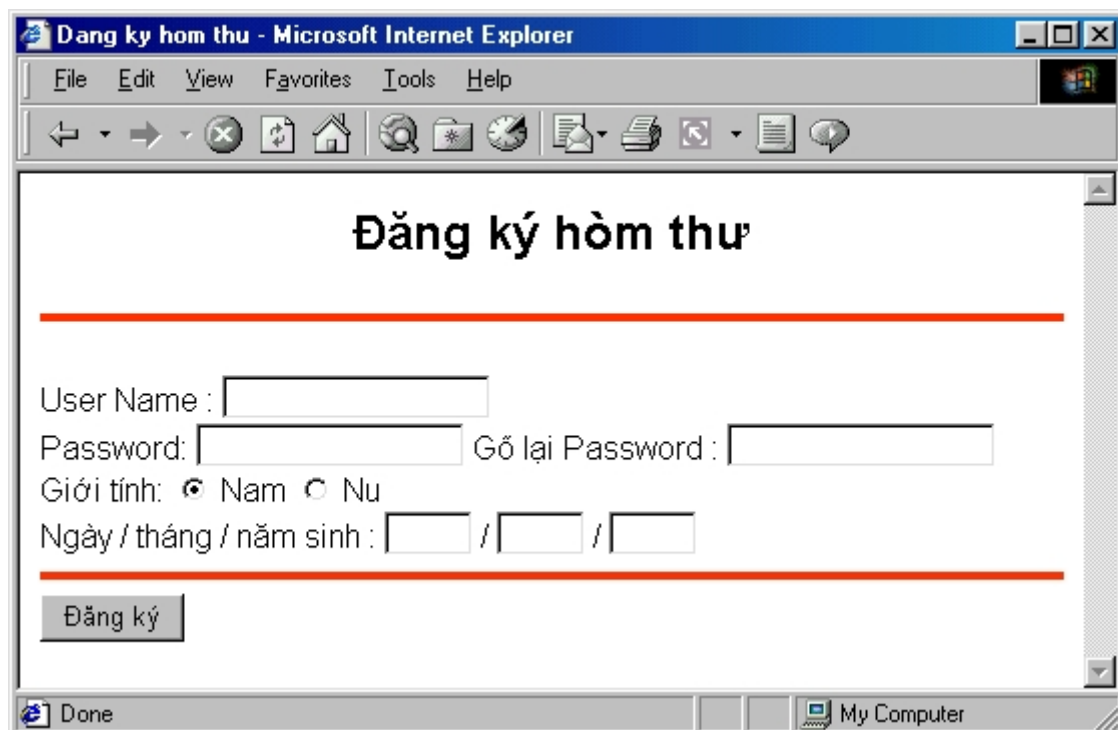
Các thẻ có thuộc tính type = "hidden" sẽ không được hiển thị trong trình duyệt, tuy nhiên khi chúng ta "Submit" thì các thông tin trong đó cũng được gửi đi.



[<Xem kết quả>](#)

Các bài tập tự giải

Bài 1: Hãy tạo ra trang Web có giao diện như sau:



Yêu cầu:

- Khi người dùng di chuyển chuột đến phần tử nào thì hiển thị dòng nhắc dưới thanh trạng thái để hướng dẫn người dùng. Ví dụ: Khi người dùng đưa chuột vào trong ô textbox User Name

thì thì hiển thị dưới thanh trạng thái là : "Nhập mã người dùng", hay khi người đưa chuột đến nút "Đăng ký" thì hiển thị dòng nhắc: "Gửi thông tin đi để đăng ký" v.v...

Hướng dẫn: Nút đăng ký nên là nút thường, tức là tạo bằng thẻ :

<input type = button value = "Đăng ký">

Bài tập 2: Có giao diện như bài 1, nhưng yêu cầu như sau:

- Khi gửi thông tin đi, cần kiểm tra xem nội dung người dùng gõ trong ô Password với textbox trong ô "Gõ lại password" có giống nhau hay không? Nếu bằng nhau thì mới gửi (Submit) đi, còn nếu không bằng thì thông báo là "Password phải giống nhau"
- Khi gửi thông tin, cần kiểm tra ngày sinh, tháng sinh phải hợp lệ. (Tức ngày phải nhỏ hơn 32, tháng phải nhỏ hơn 13)

Hướng dẫn:

Nút đăng ký nên là nút thường, tức là tạo bằng thẻ :

<input type = button value = "Đăng ký" onClick = "DangKy();">

Trong đó Hàm DangKy() sẽ kiểm tra dữ liệu hợp lệ và khi dữ liệu nhập vào đúng đắn thì gửi đi bằng cách gọi phương thức **submit** của đối tượng **document**, như sau: **document.submit();**

Bài tập 3:

Làm tương tự bài tập 7 và 8, nhưng thêm yêu cầu: Khi người dùng nhập hoặc số lượng, hoặc đơn giá thì hãy tính luôn ô textbox thành tiền.

Hướng dẫn: Viết lệnh tính thành tiền trong cả 2 sự kiện **onKeyUp** của cả hai textbox số lượng và textbox đơn giá.

***** Lưu ý:** Trước khi tính tích thì cần phải kiểm tra xem dữ liệu trong hai textbox đã có hay chưa, nếu một ô chưa nhập gì thì chưa tính.

Chương IV: Định dạng các phần tử HTML bằng CSS

Mục tiêu của chương:

- Giúp học viên hiểu rõ hơn về ý nghĩa của việc dùng kiểu (style)
- Tra cứu thành thạo các thuộc tính trong bảng đánh sách kiểu
- Vận dụng các thuộc tính (kiểu) để định dạng cho các phần tử trong trang web, nâng cao tính thẩm mỹ.

Nội dung:

I. Nhắc lại khái niệm về kiểu

Kiểu (style) thực chất là một cách định nghĩa thuộc tính cho các phần tử trong trang web theo một cách thức mới.

Việc định nghĩa các thuộc tính này cho các phần tử sẽ cho ta một "dáng vẻ mới", một "diện mạo mới" về trang web. Ngoài ra, việc định nghĩa thuộc tính theo cú pháp mới này sẽ làm tiền đề cho việc xây dựng các trang web động mà ta sẽ đề cập trong các chương tiếp theo.

Ví dụ: Trước đây, để định nghĩa một đoạn văn bản nằm trong thẻ <P> có font chữ là .vntime, chúng ta sẽ viết như sau:

<P >Dòng văn bản này có font chữ là .vntime</p>

Nhưng với cách định dạng mới theo cú pháp mà ta gọi là cú pháp CSS, thì có thể thực hiện yêu cầu trên như sau:

<P style= "font-family:.vntime">Dòng văn bản này có font chữ là .vntime </P>

II. Minh họa cách khai báo style

Trong JavaScript, bạn có thể thiết lập các thuộc tính cho một thẻ nào đó theo rất nhiều cách. Dưới đây xin giới thiệu 3 cách thiết lập thuộc tính cho thẻ, đó là :

- Thiết lập (Định nghĩa) thuộc tính ngay trong khi định nghĩa thẻ (tag), kiểu này còn gọi là định nghĩa kiểu ở mức dòng (style line)

- Thiết lập thuộc tính cho toàn bộ một loại thẻ nào đó. Cách này còn gọi là định nghĩa bộ chọn
- Định nghĩa một lớp, sau đó có thể đem ra sử dụng cho bất kỳ phần tử nào
- Định nghĩa bộ chọn ID, cho phép áp dụng tất cả các thuộc tính của bộ chọn vào tất cả các thẻ có thuộc tính ID phù hợp với mã ID trong định nghĩa.

I. Style áp dụng ở mức dòng (Inline style)

Style mức dòng (inline style) cho phép bạn có thể thay đổi hình thức (style) của một phần tử nào đó bằng cách thêm các thuộc tính định dạng trực tiếp vào ngay bên trong định nghĩa của thẻ.

Một số ví dụ minh họa

Ví dụ 1: Để tạo một dòng văn bản với thẻ <P>; có màu chữ là đỏ, bạn có thể viết như▼ sau:

<P style = "color:blue"> Màu này là màu của hoà bình <P>.

Kết quả cho ta :

Màu này là màu của hoà bình

Ví dụ 2 : Để tạo một button (Nút nhấn) có màu nền là tím (magenta), bạn viết:

<input type="button" style="background-color:magenta" value="Hello world">

Kết quả :

Hello world

Ví dụ 3: Tạo một nút nhấn (Button), trong đó, khi ngư▼ời dùng đư▼a chuột đến nút đó thì con trỏ chuột có hình bàn tay

Thuộc tính qui định chuột có kiểu hình bàn tay là "cursor:hand", do vậy cần thêm thuộc tính này trong định nghĩa thẻ

<Input type = button style = "cursor:hand" value="Chuột hình bàn tay ">

Ta đọc kết quả (Bạn di chuột vào nút này):

Ví dụ 4 : Tạo một textbox có viền màu đỏ

Thuộc tính tạo viền màu đỏ là : "background-border:red", do vậy bạn cần đặt vào trong định nghĩa thẻ text nh▼ư sau:

<input type="text" style="border-color:red" value="viền màu đỏ">

Kết quả cho ta một textbox có viền màu đỏ: viền màu đỏ

Nhận xét :

ã Trong các ví dụ ở trên, chúng ta có thể thiết lập một số thuộc tính của bất kỳ phần tử nào bằng cách đ▼ưa vào dòng style = "Tên thuộc tính:Giá trị của thuộc tính" Trong đó, Cặp tên thuộc tính : Giá trị của thuộc tính các bạn có thể tra trong [bảng các thuộc tính](#). đọc đặt trên th▼ mục của máy chủ.

ã Có thể đư▼a vào một hoặc nhiều thuộc tính trong biểu thức style="...." , khi đưa nhiều thuộc tính thì các thuộc tính cách nhau bởi dấu chấm phẩy ";"

Ví dụ 5: Tạo một nút nhấn có màu nền là tím (magenta) và màu chữ là trắng (white)

Thuộc tính qui định màu nền tím là : "background-color:magenta", Còn thuộc tính qui định màu chữ trắng là "color:white"

Như▼ vậy cần định nghĩa thẻ là : <input type ="button" style="background-

color:magenta; color:white" value="Nền tím- chữ trắng">

Kết quả : Nền tím- chữ trắng

Ví dụ 6: Thiết lập ảnh nền của trang Web là anh1.jpg, và ảnh này hiển thị ở vị trí cố định (Tức là nếu bạn có cuộn thanh cuộn của cửa sổ thì ảnh này sẽ không bị cuộn theo mà vẫn đứng yên).

Biết rằng thuộc tính đưa ảnh nền vào trang web như sau:

background-image:url(' <Đường dẫn và tên file ảnh> ')

Thuộc tính để ảnh ở vị trí cố định là :

background-attachment:fixed

Lưu ý: đặc tính ảnh cố định chỉ có trong IE, không có trong Netscape

Như vậy, yêu cầu trên có thể thực hiện như sau:

```
<body style="background-image:url('anh1.jpg'); background-attachment:fixed; color:white; font-family:arial">
```

[<Click vào đây để xem minh họa>](#) (Bạn phải đảm bảo là mở trong trình duyệt IE)

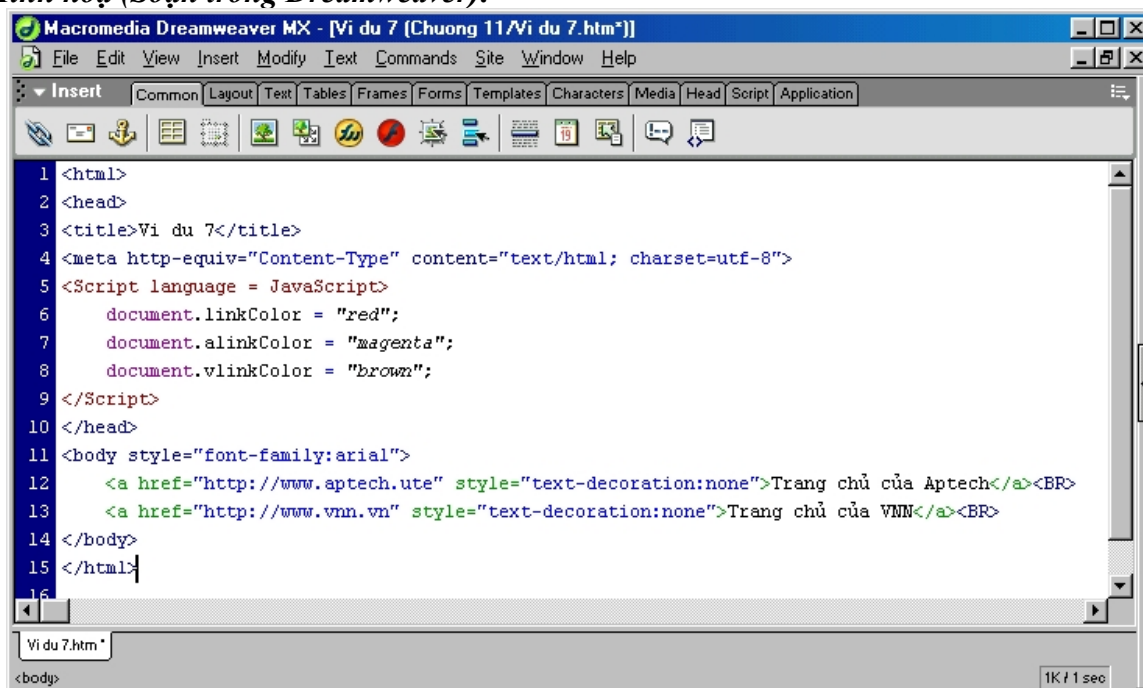
Ví dụ 7: Tạo các liên kết đến các trang <http://www.aptech.ute>, <http://www.vnn.vn>, nhưng các liên kết này không có đường gạch chân và có các thuộc tính như sau:

- Màu chữ : Đỏ (red)
- Màu nền : lavender
- Màu khi chuột click vào liên kết : tím (magenta)
- Màu khi trang đó đã được thăm : nâu (brown)

Hướng dẫn:

- Thuộc tính qui định màu chữ của liên kết là: linkColor, thuộc đối tượng document
- Thuộc tính qui định màu nền của văn bản trong một thẻ : background-color (CSS)
- Thuộc tính qui định của liên kết khi bị click chuột : alinkColor, thuộc document
- Thuộc tính qui định màu của liên kết đã được thăm: vlinkColor, thuộc document

Minh họa (Soạn trong Dreamweaver):



Định nghĩa style ở mức dòng

III. Bộ chọn HTML

Bộ chọn HTML cho phép chúng ta chỉ cần định nghĩa các thuộc tính một lần duy nhất cho một loại thẻ HTML nào đó (Ví dụ P, INPUT, H1, H2), và về sau tất cả các văn bản nằm trong loại thẻ này sẽ có cùng định dạng như nhau. Điều này rất có lợi cho ta trong thực tế, chẳng hạn khi bạn tạo một website cho một cuốn sách nào đó mà trong cuốn sách này lại có rất nhiều chương và bạn muốn màu sắc, kích cỡ, font chữ, kiểu dáng... cho tiêu đề các chương là như nhau, thì bạn chỉ cần định nghĩa một lần và về sau tiêu đề các chương sẽ có cùng định dạng giống như nhau.

Việc chúng ta tạo ra các tính chất chung cho một loại thẻ nào đó thì ta gọi là một bộ chọn (Selector) HTML

Để định nghĩa ra một bộ chọn HTML, ta có 2 cách, bạn có thể tùy ý chọn một trong 2 cách này. Cách thứ nhất là ta viết theo một dạng cú pháp mà ta gọi là cú pháp dạng CSS, dạng thứ hai là ta viết theo cú pháp của JavaScript.

Cú pháp để tạo bộ chọn theo CSS như sau:

```
<Style type = "text/CSS">
```

```
    <Tên thẻ HTML> { <Tên thuộc tính> : <Giá trị> }
```

```
    <Tên thẻ HTML> { <Tên thuộc tính> : <Giá trị> }
```

```
    <Tên thẻ HTML> { <Tên thuộc tính> : <Giá trị> }
```

```
    .....
```

```
</Style>
```

Trong đó:

- <Tên thẻ HTML> là một trong các thẻ mà bạn đã học, ví dụ nó có thể là thẻ P, INPUT, LI, UL, B, H1, H2 v.v...
- **Tên thuộc tính** : Là tên của thuộc tính của loại thẻ mà bạn muốn thay đổi . Nó có thể là color, background-color, cursor, text-align v.v... Tên các thuộc tính này các bạn có thể tra trong bảng (File [Cac the su dung trong CSS.doc](#)" nằm trong thư mục tài liệu về JavaScript của máy Server)
- **Giá trị**: Là giá trị mới mà bạn muốn đặt cho thuộc tính. Ví dụ, "red", "lavender", "hand", "center" v.v.... Để có thể đặt giá trị cho hợp lệ, bạn tham khảo trong file: [Cac the su dung trong CSS.doc](#)" trong server.

Sau khi đã định nghĩa tên của một loại thẻ với các thuộc tính của nó, thì tất cả các thẻ cùng loại về sau sẽ có tất cả các thuộc tính như đã định nghĩa. Để làm sáng tỏ điều này, chúng ta hãy lấy một số ví dụ :

Ví dụ 1: Giả sử bạn cần tạo một danh mục các đầu sách của Aptech, theo định dạng dưới đây

KNOW YOUR DESKTOP

Office 2000

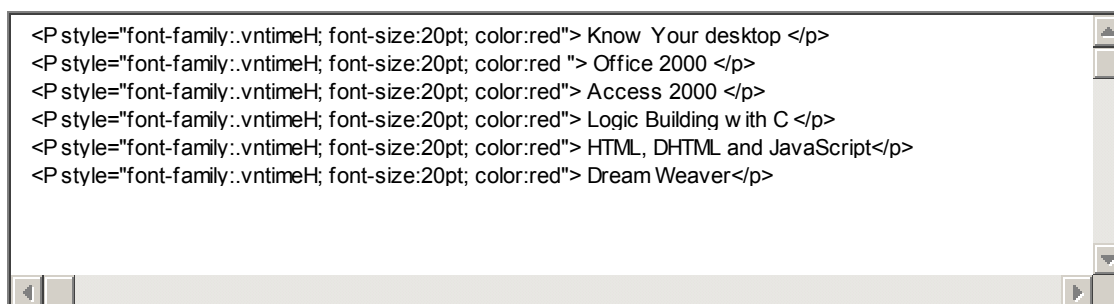
Access 2000

Logic Building with C

HTML, DHTML and JavaScript

Dream Weaver

Một cách thông thường nhất, với các kiến thức ở những phần trước, bạn hoàn toàn có thể tạo được bằng cách viết như sau :



Tuy nhiên nếu theo cách ở trên thì chúng ta dễ dàng thấy có một điểm chưa được thuận tiện, đó là mỗi khi thêm tên một loại sách chúng ta cần phải thêm trong thẻ P dòng:

<style="font-family:.vntimeH; font-size:20pt; color:red".

Trong trường hợp này, Bằng cách định ra một qui tắc (Định dạng) chung cho tất cả các thẻ P ta có thể có kết quả như mong đợi mà không phải viết lại các dòng dư thừa như ở trên. Việc định ra qui tắc chung cho thẻ P ta gọi là định nghĩa bộ chọn P.

áp dụng cú pháp định nghĩa bộ chọn theo dạng CSS để định nghĩa bộ chọn P như sau:

<style type = "text/css">

P {font-family:.vntimeH}

P {font-size:20pt}

P {color:red}

</style>

Hoặc định nghĩa một cách ngắn gọn :

<style type = "text/css">

P {font-family:.vntimeH; font-size:20pt; color:red}

</style>

PHỤ LỤC : BẢNG TRA CỨU CÁC THUỘC TÍNH CỦA CSS

- Các thuộc tính áp dụng cho Font chữ

Thuộc tính	Các giá trị hợp lệ	Ví dụ
font-family	[Tên font hoặc kiểu]	font-family: Verdana, Arial;
font-style	normal hoặc italic	font-style:italic;
font-variant	normal hoặc small-caps	font-variant:small-caps;

font-weight	normal ho bold	font-weight:bold;
font-size	[xx-large x-large large medium small x-small xx-small] [larger smaller] <i>phần trăm</i> hoặc <i>length</i>	font-size:12pt;
font	[font-style font-variant font-weight] ? font-size [/ line-height] ? font-family	font: bold 12pt Arial;

- Các thuộc tính màu và nền (Color and background properties)

Thuộc tính	Các giá trị hợp lệ	Ví dụ
color	Màu	color: red
background-color	Màu hoặc transparent	background-color: yellow
background-image	Địa chỉ (url) hoặc Không đặt gì	background-image: url(house.jpg)
background-repeat	repeat repeat-x repeat-y no-repeat	background-repeat: no-repeat
background-attachment	scroll hoặc fixed	background-attachment: fixed
background-position	[position length] {1,2} [top center bottom] [left center right]	background-position: top center
background	transparent color url repeat scroll position	background: silver url(house.jpg) repeat-y

* Lưu ý: Các giá trị in nghiêng khi sử dụng bạn phải thay bằng giá trị cụ thể, ví dụ viết phần trăm hay length thì bạn phải thay các giá trị dạng phần trăm, ví dụ 50% đối với thuộc tính phần trăm (percentage) và 10, 20 cho thuộc tính length.

- Các thuộc tính áp dụng cho Text

Thuộc tính	Giá trị hợp lệ	Ví dụ
letter-spacing	normal length	letter-spacing:5pt
text-decoration	none underline overline line-through	text-decoration:underline
vertical-align	sub super	vertical-align:sub
text-transform	capitalize uppercase lowercase none	text-transform:lowercase
text-align	left right center justify	text-align:center
text-indent	length percentage	text-indent:25px
line-height	normal number length percentage	line-height:15pt

- Các thuộc tính áp dụng cho các ô trong một bảng

Thuộc tính	Giá trị hợp lệ	Ví dụ
margin-top	length percentage auto	margin-top:5px
margin-right	length percentage auto	margin-right:5px
margin-bottom	length percentage auto	margin-bottom:1em
margin-left	length percentage auto	margin-left:5pt
margin	length percentage auto {1,4}	margin: 10px 5px 10px 5px
padding-top	length percentage	padding-top:10%
padding-right	length percentage	padding-right:15px
padding-bottom	length percentage	padding-bottom:1.2em
padding-left	length percentage	padding-left:10pt; }
padding	length percentage {1,4}	padding: 10px 10px 10px 15px
border-top-width	thin medium thick length	border-top-width:thin
border-right-width	thin medium thick length	border-right-width:medium
border-bottom-width	thin medium thick length	border-bottom-width:thick
border-left-width	thin medium thick length	border-left-width:15px
border-width	thin medium thick length {1,4}	border-width: 3px 5px 3px 5px
border-top-color	color	border-top-color:navajowhite
border-right-color	color	border-right-color:whitesmoke
border-bottom-color	color	border-bottom-color:black
border-left-color	color	border-left-color:#C0C0C0
border-color	color {1,4}	border-color: green red white blue; }

border-top-style	none solid double groove ridge inset outset	border-top-style:solid
border-right-style	none solid double groove ridge inset outset	border-right-style:double
border-bottom-style	none solid double groove ridge inset outset	border-bottom-style:groove
border-left-style	none solid double groove ridge inset outset	border-left-style:none
border-style	none solid double groove ridge inset outset	border-style:ridge; }
border-top	border-width border-style border-color	border-top: medium outset red
border-right	border-width border-style border-color	border-right: thick inset maroon
border-bottom	border-width border-style border-color	border-bottom: 10px ridge gray
border-left	border-width border-style border-color	border-left: 1px groove red
border	border-width border-style border-color	border: thin solid blue
float	none left right	float:none
clear	none left right both	clear:left

- Các thuộc tính phân loại - classification Properties

Thuộc tính	Giá trị hợp lệ	Ví dụ
display	none block inline list-item	display:none
list-style-type	disk circle square decimal lower-roman upper-roman lower-alpha upper-alpha none	list-style-type:upper-alpha
list-style-image	url none	list-style-image:url(icFile.gif)
list-style-position	inside outside	list-style-position:inside
list-style	keyword position url	list-style: square outside url(icFolder.gif)

- Các thuộc tính định vị trí cho các phần tử

Thuộc tính	Giá trị hợp lệ	Ví dụ	Có thể áp dụng cho
clip	Toạ độ của một hình chữ nhật auto	clip:rect(0px 200px 200px 0px)	tất cả các phần tử (all elements)
height	length auto	height:200px	DIV, SPAN và các ptử bị thay thế
left	length percentage auto	left:0px	Các phần tử được định vị tuyệt đối và tương đối
overflow	visible hidden scroll auto	overflow:scroll	tất cả các phần tử
position	absolute relative static	position:absolute	tất cả các phần tử
top	length percentage auto	top:0px	Các phần tử được định vị tuyệt đối và tương đối
visibility	visible hidden inherit	visibility:visible	tất cả các phần tử
width	length percentage auto	width:80%	DIV, SPAN and replaced elements
z-index	auto integer	z-index:-1	Các phần tử được định vị tuyệt đối và tương đối

- Thuộc tính liên quan đến in ấn - Printing Properties

Thuộc tính	Giá trị hợp lệ	Ví dụ
page-break-before	auto always left right	page-break-before:always
page-break-after	auto always left right	page-break-before:auto

-Pseudo Classes

Thuộc tính	Giá trị hợp lệ	Ví dụ
cursor	auto crosshair default hand move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help	{ cursor:hand; }
active, hover, link, visited	n/a	a:hover { color:red; }
first-letter, first-line	any font manipulating declaration	p:first-letter{ float:left;color:blue } .

Một số ví dụ áp dụng:

Ví dụ 1: Tạo một dòng văn bản "Welcome to CSS" có font chữ là Arial, in nghiêng và kích thước font chữ là 16 point.

```
<HTML>
<HEAD> </HEAD>
<BODY>
<P style="font-family:arial; font-style:italic; font-size:16pt">Welcome to </P>
</BODY>
</HTML>
```

[<Xem kết quả>](#)

Ví dụ 2: Tạo một textbox với màu nền là màu tím (magenta).

```
<HTML>
<HEAD> </HEAD>
<BODY>
<input type = text style = "background-color:magenta">
</BODY>
</HTML>
```

[<Xem kết quả>](#)

Ví dụ 3: Đặt 3 file ảnh nền tương ứng vào tài liệu, vào textbox và vào textarea

```
<HTML>
<HEAD> </HEAD>
<BODY style="background-image:url('anh1.jpg');background-repeat:no-repeat">
<input style = "background-image:url('anh2.jpg');background-repeat:repeat-x; width:100%">
<P>
<textarea style = "background-image:url('anh3.jpg');background-repeat:repeat-y" cols = 80 rows =
25>
</textarea>
</BODY>
</HTML>
```

[<Xem kết quả>](#)

***Ghi chú:**

-Khi ta đặt là repeat-x thì ta có một dãy ảnh được xếp liên tiếp theo chiều ngang

-Khi ta đặt là repeat-y thì ta có một dãy ảnh được xếp liên tiếp theo chiều dọc

Ví dụ 4: Tạo một dòng văn bản, chứa xâu: "Trang chu của HY-Aptech", trong đó từ "Trang chu" có màu xanh (blue) và khi người dùng di chuyển chuột đến chữ "Trang chu" thì chuột chuyển thành hình bàn tay, và khi chuột click lên chữ này thì trang <http://www.aptech.ute> sẽ được mở.

```
<HTML>
<HEAD> </HEAD>
<BODY>
<TITLE>Tạo liên kết và xử lý sự kiện</TITLE>
<font style ="cursor:hand;color:blue"
onclick="window.open('http://www.aptech.ute');">Trang chu </font> của HY-Aptech
</BODY>
</HTML> ===> Xem kết quả
```

Ghi chú: Khi muốn áp dụng các kiểu cho một số phần tử các bạn chỉ cần viết :
style="Tên_Thuộc_tính : Giá_Trị;" trong định nghĩa thẻ. Trong đó cặp "Tên_Thuộc_tính :
Giá_Trị;" có thể viết như cột ví dụ đã chỉ ra ở các bảng trên.

Nội dung trang Web của chúng ta bây giờ sẽ là:

```
<html>
<head>
<title>Su dung bo chon</title>
</head>
<style type = "text/css">
  P {font-family:.vntimeH; font-size:20pt; color:red}
</style>
<body>
<P>Know Your desktop </P>
<P>Office 2000 </P>
<P>Access 2000 </P>
<P>Logic Building with C </P>
<P>HTML, DHTML and JavaScript</P>
<P>Dream Weaver</P>

</body>
</html>
```

Chương V: Tạo và xử lý các tầng (Layer)

Mục tiêu của chương này giúp người học có thể:

- ❑ Thay đổi thuộc tính của một phần tử HTML khi có các sự kiện chuột và bàn phím xảy ra.
- ❑ Định nghĩa các tầng (Layer) trong IE và Netscape.
- ❑ Ẩn, hiện tầng bằng các lệnh JavaScript
- ❑ Tạo một menu liên kết đến các trang khác sử dụng tầng.

A. TÓM TẮT LÝ THUYẾT

1. Một số sự kiện thường dùng:

- Sự kiện Click chuột (**onClick**) được kích hoạt khi chuột được click
- Sự kiện di chuyển chuột vào bên trong phần tử (**onMouseOver**) được kích hoạt khi chuột di chuyển chuột từ ngoài vào trong phần tử.
- Sự kiện di chuột ra ngoài (**onMouseOut**) được kích hoạt khi di chuyển chuột từ bên trong phần tử ra khỏi phần tử.
- Sự kiện nhấn phím (**onKeyUp**) được kích hoạt khi người dùng nhấn một phím.

2. Thay đổi thuộc tính của các phần tử HTML

Để thay đổi thuộc tính của một phần tử HTML nào đó trong trang bằng lệnh JavaScript, bạn viết như sau:

document.all.<Giá trị ID của thẻ>.style.<Tên thuộc tính> = <Giá trị mới>

Trong đó:

- **<Giá trị ID của thẻ>** chính là giá trị mà bạn đặt cho thuộc tính ID, ví dụ:
<H2 ID = 'TieuDe'> THAY ĐỔI THUỘC TÍNH CỦA THẺ </H2> thì <giá trị ID của thẻ> trong trường hợp này sẽ là 'TieuDe'.
- **<Tên thuộc tính>** là tên thuộc tính cần thay đổi. Bạn có thể tra trong tài liệu đã được giáo viên phát: "Danh sách các thuộc tính". Ví dụ : text-align, font-size, color v.v...

- **<Giá trị mới>** là những giá trị hợp lệ cho thuộc tính cần thay đổi. (Bạn cũng tra trong tài liệu “Danh sách các thuộc tính”).

**** Lưu ý:** Nếu bạn muốn thay đổi thuộc tính của một phần tử thì bắt buộc bạn phải gán cho thẻ đó một giá trị ID duy nhất (Như ví dụ ở trên).

3. Định nghĩa tầng trong IE và Netscape

a. Định nghĩa tầng trong IE

Để định nghĩa tầng trong IE, bạn chỉ cần đặt thuộc tính position cho style.

Ví dụ: `<H2 style = “position:absolute;”>Thẻ H2 này được đặt trên một tầng </H2>`

Tuy nhiên, với cách định nghĩa ở trên thì mỗi tầng chỉ chứa được một phần tử. Nếu bạn muốn một tầng chứa được nhiều hơn một phần tử thì bạn định nghĩa tầng bằng cặp thẻ `<DIV>` hoặc `` như sau:

<DIV ID = ‘Tang1’ style = “position:absolute”>

<Đặt các phần tử khác vào đây, ví dụ : >

<H1> Phần tử này nằm trên tầng 1 </H1>

<input type = button value = “Move”>

** Trang chủ của báo E-Chip **

</DIV>

Mỗi tầng khi tạo ra, bạn nên gán thuộc tính ID cho tầng đó một giá trị duy nhất (Không trùng với ID của bất kỳ thẻ nào trong trang). Bạn hoàn toàn có thể thực hiện các thao tác đối với tầng như :

- Ẩn tầng, ví dụ : `document.all.Tang1.style.visibility = ‘hidden’`
- Hiện tầng, ví dụ : `document.all.Tang1.style.visibility = ‘visible’`
- Di chuyển tầng sang trái 5 pixel, ví dụ: `document.all.Tang1.style.pixelLeft -= 5`
- Di chuyển tầng sang phải 5 pixel, ví dụ: `document.all.Tang1.style.pixelLeft += 5`
- Di chuyển tầng lên trên 5 pixel, ví dụ: `document.all.Tang1.style.pixelTop -= 5`
- Di chuyển tầng xuống dưới 5 pixel, ví dụ: `document.all.Tang1.style.pixelTop+=5`

b. Định nghĩa tầng trong Netscape

Trong Netscape, việc định nghĩa (tạo ra) tầng có phần dễ dàng hơn với thẻ LAYER :

<LAYER name = “Tang1”>

<Đặt các phần tử khác vào đây, ví dụ : >

<H1> Phần tử này nằm trên tầng 1 </H1>

<input type = button value = “Move”>

** Trang chứa nhiều mẹo vặt về IT **

</LAYER>

- Để ẩn / hiện tầng được tạo bởi thẻ LAYER thông qua lệnh JavaScript, bạn viết :
 - `document.<Tên của tầng>.visibility = 'hidden'` (để ẩn tầng)
 - `document.<Tên của tầng>.visibility = 'show'` (để hiện tầng)

Trong đó : `<Tên của tầng>` chính là giá trị của thuộc tính name.

****Lưu ý:** Để truy cập đến một tầng, trong IE sử dụng giá trị của thuộc tính ID, trong khi đó Netscape sử dụng giá trị của thuộc tính **name**.

✓ **Ví dụ :** `document.Tang1.visibility = 'hidden'` Để ẩn tầng có tên là Tang1 tạo ra ở trên.

- Để di chuyển tầng bằng JavaScript, bạn viết:
 - `document.<Tên tầng>.left = <Giá trị>` (Di chuyển sang trái hoặc phải)

- `document.<Tên tầng>.top = <Giá trị>` (Di chuyển lên | xuống)

Trong đó: <giá trị> âm để di chuyển lùi, giá trị dương để di chuyển tiến.

- Ví dụ di chuyển tầng lên trên thêm 10 pixel
 - `document.Tang1.top -= 10`
- Ví dụ di chuyển tầng sang phải thêm 10 pixel
 - `document.Tang1.left += 10`
- Ví dụ, di chuyển tầng đến vị trí cách mép trên cửa sổ trình duyệt 100px, và cách mép trái 50 pixel.
 - `document.Tang1.top = 100`
 - `document.Tang1.left = 50`

B. BÀI TẬP MẪU

Bài số 1: Minh họa sự kiện di chuyển chuột vào phần tử.

Yêu cầu: Tạo một liên kết đến trang <http://www.w3schools.com> bằng thẻ H2. Có màu nền là xanh, màu chữ là trắng. Khi chuột di chuyển đến thì đổi màu nền thành màu đỏ.

Hướng dẫn: Để đổi màu nền thành đỏ đối với thẻ H2 (hoặc thẻ bất kỳ) bạn viết:

`document.all.LienKet.style.backgroundColor = 'red'`

Trong đó **LienKet** là giá trị của thuộc tính ID .

☺ Câu lệnh JavaScript này đặt ở đâu ?

☞ Theo như yêu cầu đầu bài là : “Khi chuột di chuyển ...”. Do vậy câu lệnh này sẽ được đặt trong sự kiện di chuyển chuột đến (có tên là **onMouseMove**)

Minh họa:

```
<HTML>
<HEAD>
<TITLE>Hiệu ứng di chuyển chuột</TITLE>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
</HEAD>

<BODY>
  <h2 ID= "LienKet" STYLE="background-color:blue; color:white"
    onMouseMove="document.all.LienKet.style.backgroundColor = 'red'">
    Trang dạy JavaScript http://www.w3schools.com
  </h2>
</BODY>
</HTML>
```

🔔 **Nhắc lại:** Các thuộc tính trong bảng tra cứu “danh sách thuộc tính” nếu có chứa dấu gạch nối (ví dụ : background-color) thì khi viết trong JavaScript, bạn phải chuyển ký tự đứng ngay sau dấu gạch nối đó thành chữ hoa và bỏ dấu gạch nối đi. (Ví dụ từ background-color → background**C**olor)

Bài số 2: Minh họa sự kiện di chuyển chuột vào và ra khỏi một phần tử.

Yêu cầu: Như bài tập 1, và thêm yêu cầu sau: Khi người dùng di chuyển chuột ra khỏi phần tử H2 đó thì đặt lại màu nền là màu xanh.

Hướng dẫn: Viết lệnh thay đổi màu nền thành xanh trong sự kiện di chuột ra ngoài.

Minh họa:

```
<HTML>
<HEAD>
<TITLE>Hiệu ứng di chuyển chuột Version 2</TITLE>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
</HEAD>
```

```

<BODY>
  <H2 ID= "LienKet" STYLE="background-color:blue; color:white"
    onMouseMove = "document.all.LienKet.style.backgroundColor = 'red' "
    onMouseOut = "document.all.LienKet.style.backgroundColor = 'blue' ">
    Trang dạy JavaScript http://www.w3schools.com
  </H2>
</BODY>
</HTML>

```

Bài số 3: Minh họa tạo một liên kết hoàn chỉnh đến một trang web

Yêu cầu: Như bài số 2 nhưng khi người dùng click chuột thì mở trang <http://www.w3schools.com>. Và chuột có hình bàn tay.

Hướng dẫn: Để mở trang web bất kỳ bạn viết: window.open('Địa chỉ URL'). Lệnh mở này được đặt trong sự kiện Click chuột (Vì theo yêu cầu: khi người dùng click chuột thì mới mở).

Minh họa:

```

<HTML>
<HEAD>
<TITLE>Hiệu ứng di chuyển chuột Version 2</TITLE>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
</HEAD>

<BODY>
  <H2 ID="LienKet" STYLE="background-color:blue; color:white; cursor:hand"
    onMouseMove = "document.all.LienKet.style.backgroundColor = 'red';"
    onMouseOut  = "document.all.LienKet.style.backgroundColor = 'blue';"
    onClick      = "window.open('http://www.w3schools.com');">
    Trang dạy JavaScript http://www.w3schools.com
  </H2>
</BODY>
</HTML>

```

Bài tập tự giải:

Bài tự giải 1: Hãy tạo một menu đặt theo chiều dọc gồm 4 mục như sau:

Giới thiệu

Tin tức

Sản phẩm

Trợ giúp

Yêu cầu: 4 mục này có màu nền là xanh, màu chữ là vàng (yellow). Khi người dùng di chuyển đến mục nào thì mục đó có màu nền là màu đỏ. Khi di chuyển chuột ra khỏi thì màu nền trở lại màu xanh.

Khi người dùng click vào thì mở ra trang tương ứng là <http://www.echip.com.vn>, <http://www.manguan.com>, <http://www.w3schools.com> và <http://www.quantrimang.com>.

Gợi ý: Làm tương tự như bài tập mẫu, nhưng tạo ra 4 thẻ H2.

Bài tập tự giải 2: Hãy tạo một menu gồm 4 mục như ở trên nhưng theo chiều ngang,
Hướng dẫn: Nếu bạn dùng thẻ H2, thì mỗi mục sẽ tự động được đặt riêng trên một dòng. Để có thể đặt nhiều mục trên cùng một dòng, bạn có thể tạo một bảng có một hàng và nhiều cột. Mỗi mục bây giờ sẽ được đặt trong một thẻ <TD>. Để thay đổi kích thước, màu nền, màu chữ v.v.. bạn cũng sử dụng STYLE: <TD style = “color: white”

Hoặc cách thứ hai là bạn sử dụng thẻ . Ví dụ:

```
<Span style = “color:white; background-color:blue; cursor:hand; font-size:16pt”  
onClick = “window.open(‘http://www.echip.com.vn’);” >
```

Trang Echip.com

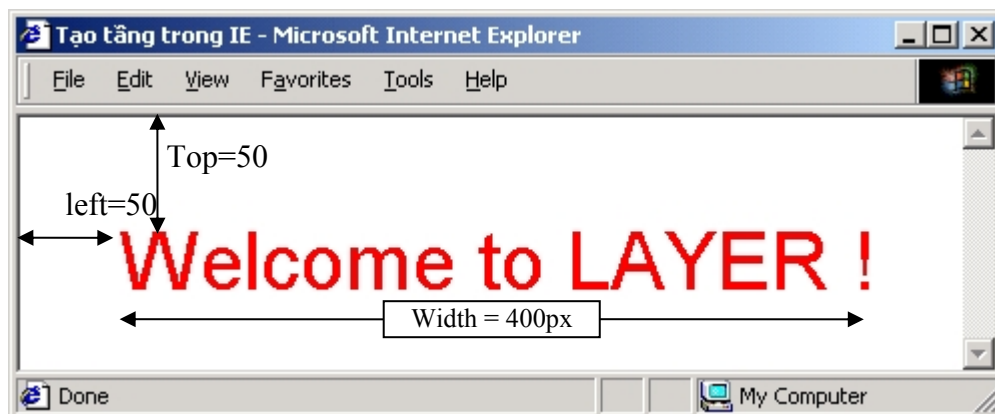
Bài số 4: Minh hoạ tạo tầng trong IE

Tạo một tầng gồm có dòng chữ “Welcome to LAYER !”, màu đỏ, kích thước 40pt, font chữ Arial. Toàn bộ dòng chữ này có độ rộng (width) là 300px.

Hướng dẫn: Việc tạo tầng và đặt các thuộc tính có thể đặt thông qua định nghĩa STYLE.

Minh hoạ:

```
<HTML>  
<HEAD>  
<TITLE>Tạo tầng trong IE</TITLE>  
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">  
</HEAD>  
  
<BODY>  
  <DIV STYLE="position:absolute; color:red; font-family:arial; font-size:30pt;  
              top: 50px; left: 50px; width:400px">  
    Welcome to LAYER !  
  </DIV>  
</body>  
</HTML>
```



Bài số 5: Minh hoạ việc ẩn và hiện tầng bằng câu lệnh JavaScript

Yêu cầu: tạo một dòng chữ “**Các liên kết**” bằng thẻ H2, màu nền là tím (magenta), màu chữ trắng (white). Và tạo một tầng có ID = LienKet, Trong tầng có một bảng gồm 2 hàng, 1 cột, nội dung của bảng chứa 2 liên kết (bạn có thể tạo bằng thẻ A HREF) đến các trang <http://www.vol.vnn.vn>, và <http://www.fateback.com>, Ban đầu, tầng này ẩn (visible : ‘hidden’). Khi người dùng di chuyển chuột đến thẻ H2 thì tầng này hiện. Còn khi người dùng click vào một trong 3 liên kết thì tầng này ẩn.

Hướng dẫn: Để ẩn hay hiện tầng bạn viết:

document.all.LienKet.style.visibility = 'visible' (hoặc 'hidden')

Minh họa:

```
<HTML>
<HEAD>
<TITLE>Tạo tầng trong IE Version 2</TITLE>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
</HEAD>

<BODY>
<H2 STYLE="color:white; background-color:magenta; width:200px"
    onMouseOver = "document.all.LienKet.style.visibility = 'visible';">
    Các liên kết
</H2>

<DIV ID="LienKet" STYLE="position:absolute; font-family:arial; visibility: hidden;
    background-color:yellow; top: 40px; left: 10px; width:200px; ">

    <TABLE BORDER="0">
        <TR>
            <TD    onClick = "document.all.LienKet.style.visibility = 'hidden';" >
                <A HREF = "HTTP://www.vol.vnn.vn">Trang đăng ký E-Mail</A>
            </TD>
        </TR>

        <TR>
            <TD onClick = "document.all.LienKet.style.visibility = 'hidden';">
                <A HREF = "HTTP://www.fateback.com">Website miễn phí</A>
            </TD>
        </TR>
    </TABLE>
</DIV>
</body>
</HTML>
```

Bài tập tự giải 3: Tạo một hệ thống menu như trang bên:

```
<HTML>
<HEAD>
<TITLE>Tạo Menu trong IE - Sử dụng TẦNG và JavaScript</TITLE>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
</HEAD>

<!-- Định nghĩa lớp -->
<STYLE TYPE="TEXT/CSS">
    .MenuBar{background-color:blue; color:white; font-family:arial; font-size:16pt;width:150px; cursor:hand}
    .Menu { background-color: yellow; color: white; position: absolute; top : 40; visibility : hidden}
</STYLE>
<SCRIPT language="JavaScript">
    function AnCacMenu() // Hàm Ẩn các Menu có tác dụng làm ẩn các tầng mỗi khi được gọi
    {
        document.all.Menu1.style.visibility = 'hidden';
        document.all.Menu2.style.visibility = 'hidden';
        document.all.Menu3.style.visibility = 'hidden';
    }
</SCRIPT>

<!-- Hàm AnCacMenu () trong sự kiện onClick của BODY sẽ được gọi khi người dùng
```

Click chuột vào bất kỳ đâu trong tài liệu (Xem lại phần Nổi bật sự kiện) -->

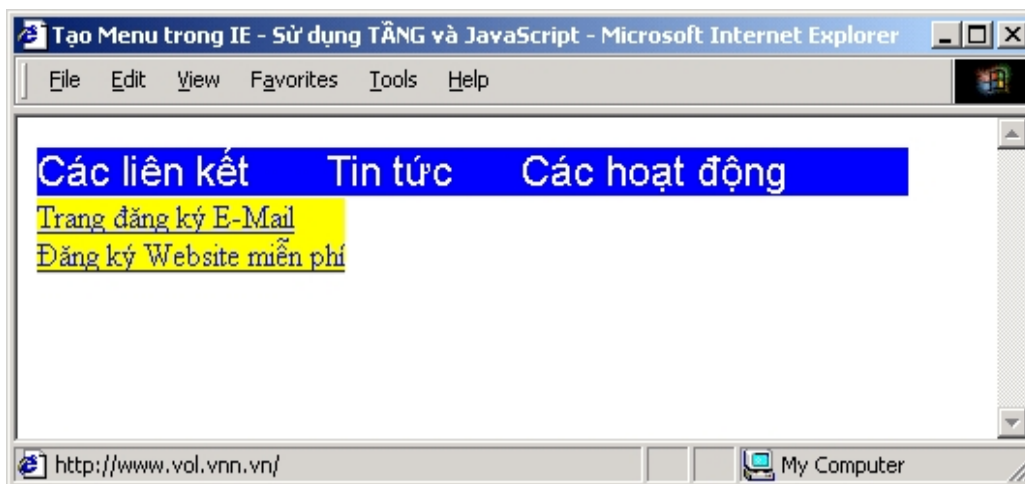
```

<BODY onClick = "AnCacMenu()">
  <!-- Tạo tầng 1 chứa các liên kết của mục "Các Liên kết" -->
  <DIV CLASS="Menu" ID="Menu1" style = "left:10">
    <A HREF = "HTTP://www.vol.vnn.vn">Trang đăng ký E-Mail</A> <BR>
    <A HREF = "HTTP://www.fateback.com">Đăng ký Website miễn phí</A>
  </DIV>

  <!-- Tạo tầng 2 chứa các liên kết của mục "Các Liên kết" -->
  <DIV CLASS="Menu" ID="Menu2" style = "left:150;">
    <A HREF = "HTTP://www.w3schools.com">Trang dạy JavaScript</A> <BR>
    <A HREF = "HTTP://www.manguon.com">Trang IT Tiếng việt</A>
  </DIV>

  <!-- Tạo tầng 3 chứa các liên kết của mục "Các Liên kết" -->
  <DIV CLASS="Menu" ID="Menu3" style = "left:250">
    <A HREF = "HTTP://www.hut.edu.vn">Đại học Bách khoa Hà Nội</A> <BR>
    <A HREF = "HTTP://www.ctu.edu.vn">Đại học cần thơ</A>
    <A HREF = "HTTP://www.vnu.edu.vn">Đại học Quốc gia</A>
  </DIV>
  <span CLASS="MenuBar" onMouseMove="AnCacMenu(); document.all.Menu1.style.visibility = 'visible'">
    Các liên kết
  </span>
  <span CLASS="MenuBar" onMouseMove="AnCacMenu(); document.all.Menu2.style.visibility = 'visible'">
    Tin tức
  </SPAN>
  <span CLASS="MenuBar" onMouseMove="AnCacMenu(); document.all.Menu3.style.visibility = 'visible'">
    Các hoạt động
  </SPAN>
</body>
</HTML>

```



Chương VI: Nội dung động và định vị động

Mục tiêu của chương này giúp người học có thể:

- ❑ Thay đổi nội dung của các thẻ trong trang web thông qua JavaScript
- ❑ Sử dụng JavaScript để di chuyển các tầng
- ❑ Vận dụng khả năng định vị động và nội dung động vào thiết kế web

C. TÓM TẮT LÝ THUYẾT

1. Nội dung động là khả năng cho phép thay đổi nội dung của trang web bằng các lệnh Script.

Để thay đổi nội dung của một phần tử nào đó **trong IE**, chúng ta có thể thay đổi một trong 4 thuộc tính của nó :

- innerText
- innerHTML
- outerText
- outerHTML

Cú pháp dùng để thay đổi giá trị của các thuộc tính như sau:

document.all.<Giá trị của thuộc tính ID>.<Tên thuộc tính> = <Giá trị mới>

Trong đó <Tên thuộc tính> là một trong số 4 thuộc tính nêu trên.

2. Định vị (Thay đổi vị trí) các tầng trong IE và Netscape

Lưu ý quan trọng : Để thay đổi vị trí của một phần tử nào đó trong IE và Netscape, trước hết bạn cần phải đặt phần tử đó trong một tầng.

2.1 Thay đổi trong IE

Bạn cần thay đổi 2 thuộc tính của tầng là pixelLeft và pixelTop.

- Thuộc tính pixelLeft qui định di chuyển sang phải / sang trái
- Thuộc tính pixelTop qui định di chuyển lên trên / xuống dưới

Cú pháp câu lệnh để thay đổi như sau:

`document.all.<Giá trị ID>.style.pixelLeft = <Vị trí cần di chuyển đến>`

`document.all.<Giá trị ID>.style.pixelTop = <Vị trí cần di chuyển đến>`

2.2 Thay đổi trong Netsape

Trong Netscape, muốn thay đổi bạn cũng chỉ việc thay đổi 2 thuộc tính là top và left, như sau:

`document.<Tên tầng>.top = <Vị trí cần di chuyển>`

`document.<Tên tầng>.left = <Vị trí cần di chuyển>`

Trong đó, Tên tầng chính là giá trị của thuộc tính **name** do bạn đặt.

B. BÀI TẬP MẪU

Bài số 1: Minh hoạ thay đổi thuộc tính innerText

Yêu cầu: Tạo một dòng chữ "I am having fun" thành "This is great fun" khi người dùng click chuột.

Hướng dẫn: Bạn có thể dùng bất kỳ thẻ gì (thẻ H, thẻ P v.v...) để tạo dòng chữ ở trên. Do đầu bài yêu cầu là chỉ thay đổi dòng văn bản, do vậy chúng ta sẽ thay đổi thuộc tính innerText, dòng lệnh này sẽ đặt trong sự kiện onClick.

Minh hoạ :

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<body>
    <h2 id="ID1" onClick="document.all.ID1.innerText='This is great fun'">I am having fun</h2>
</body>
</html>
```

Bài số 2: Minh hoạ thay đổi thuộc tính innerHTML

Yêu cầu: Tạo một dòng chữ "Click here". Khi người dùng click chuột vào thì dòng chữ đó biến thành nút có nhãn là "Having fun"

Hướng dẫn:

- Bạn có thể sử dụng bất kỳ thẻ nào để hiển thị dòng chữ "Click here"
- Trong sự kiện click chuột (onMouseMove) bạn viết lệnh thay nội dung trong thẻ bằng nội dung mới (Nội dung này là thẻ tạo nút).

Minh hoạ:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<body>
    <h2 id="ID1" onClick="document.all.ID1.innerHTML='<input type=button value = OK>'">
        Click here
    </h2>
</body>
</html>
```

Bài số 3: Minh hoạ thay thế thuộc tính outerText.

Yêu cầu: Tạo một nút có nhãn là "Open". Khi người dùng click vào nút này thì mở trang <http://www.echip.com.vn>, trong một cửa sổ mới và dòng nút đó sẽ thay bằng dòng chữ "Trang này đã mở".

Hướng dẫn:

- Để mở một trang web trong một cửa sổ mới, bạn viết: window.open("Địa chỉ URL của trang cần mở", "_Blank")
- Thay thế nút bằng một dòng chữ thông qua thay đổi thuộc tính outerText của nút.

Minh hoạ:

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<SCRIPT language="JavaScript">
    function ThayDoi()
    {
        window.open("http://www.echip.com.vn","_blank");
        document.all.Nut1.outerText="Trang này đã thăm";
    }
</SCRIPT>
<body>
    <INPUT id="Nut1" type="button" value="Open" onClick="ThayDoi();">
</body>
</html>

```

Bài số 4: Minh họa thay đổi thuộc tính outerHTML

Yêu cầu: Tạo một dòng chữ "Click here" có màu xanh, kích cỡ H1. Khi người dùng click vào dòng chữ này thì thay bằng một liên kết đến trang <http://www.vn.vn>

Hướng dẫn: Do ở đây khi người dùng click vào dòng chữ thì thay thế bằng một liên kết nên thuộc tính cần thay đổi sẽ là outerHTML. Câu lệnh thay đổi sẽ được đặt trong sự kiện Click chuột.

Minh họa:

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<SCRIPT language="JavaScript">
    function ChenLienKet()
    {
        document.all.LienKet.outerHTML='<A HREF = "http://www.vnn.vn">Trang Aptech</A>' ;
    }
</SCRIPT>
<body>
    <h1 id="LienKet" style="color:blue" onClick="ChenLienKet()">Click Here</h1>
</body>
</html>

```

Bài số 5: Minh họa việc định vị động trong IE

Yêu cầu: Tạo một nút có nhãn là "Sign Up". Khi chuột di chuyển trong nút này thì hiển thị dòng nhắc là "Đăng ký hòm thư mới" có màu nền là vàng tại vị trí của con chuột. Khi chuột di chuyển ra ngoài thì dòng nhắc ẩn đi.

Hướng dẫn: Bạn tạo ra một tầng chứa dòng chữ "Đăng ký hòm thư mới" có màu nền là vàng. Khi chuột di chuyển đến (onMouseMove) thì đặt thuộc tính visibility là 'visible' để cho hiện tầng đó và khi di chuyển chuột ra ngoài (onMouseOut) thì đặt lại thuộc tính visibility là 'hidden' để ẩn tầng. Lưu ý, vị trí của chuột được lưu trong thuộc tính event.clientX và event.clientY. Bạn sẽ gán vị trí này của chuột cho 2 thuộc tính pixelLeft và pixelTop để định vị tầng.

Minh họa:

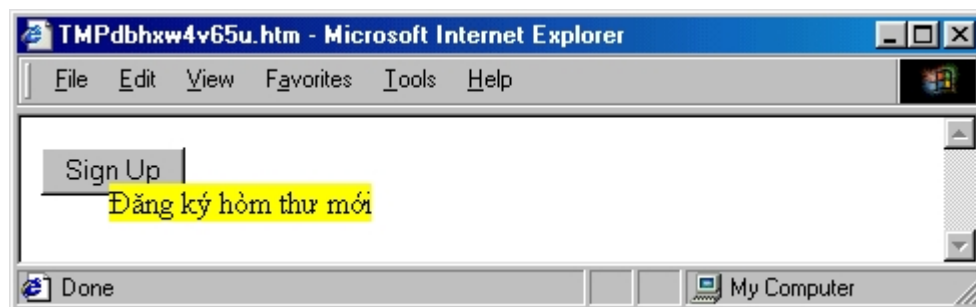
```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<SCRIPT language="JavaScript">
    function HienThi()
    {
        document.all.Tang1.style.pixelLeft = event.clientX;
        document.all.Tang1.style.pixelTop = event.clientY;
        document.all.Tang1.style.visibility = 'visible';
    }

    function AnTang() // Ẩn Tầng Tang1
    {
        document.all.Tang1.style.visibility = 'hidden';
    }
</SCRIPT>
<BODY>
    <INPUT type="button" value="Sign Up"
        onMouseMove="HienThi();"
        onMouseOut="AnTang()">

    <DIV id= Tang1 style="position:absolute; visibility: hidden; background-color:yellow">
        Đăng ký hòm thư mới
    </DIV>
</BODY>
</html>

```



Kết quả khi chuột di chuyển vào trong nút

Bài số 6: Minh họa nội dung trong Netscape

Yêu cầu: Tạo một tầng có tên là Tang1, một nút nhấn có nhãn là "Nạp trang Web" và một hộp text có tên là DiaChi. Khi người dùng nhập địa chỉ vào trong hộp text và nhấn nút "Nạp trang web" thì nội dung của trang đó sẽ được nạp vào tầng **Tang1**.

Hướng dẫn:

- Dùng thẻ LAYER để tạo tầng và đặt tên cho nó là Tang1
- Viết trong sự kiện onClick của nút "Nạp trang web" câu lệnh nạp tài liệu vào tầng. Cú pháp nạp tài liệu vào một tầng trong Netscape như sau:

document.<Tên tầng>.src = "Địa chỉ trang cần nạp"

Trong trường hợp này sẽ là :

document.Tang1.src = document.form1.DiaChi.value

Minh họa:

```

<html>
<head>
<TITLE>Hiện thị tooltip</TITLE>

```

```

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<body>
  <LAYER name="Tang1">
    <h1> Tầng này dùng để hiển thị trang Web bạn gõ trong hộpText !
  </LAYER>

  <FORM name="Form1">
    <INPUT type="text" name="DiaChi">
    <INPUT type="button" onClick="document.Tang1.src = document.form1.DiaChi.value">
  </FORM>

</body>
</html>

```

C. BÀI TẬP TỰ GIẢI

Bài số 1:

Tạo một thẻ H1, màu chữ xanh dùng để hiển thị thời gian của hệ thống (gồm giờ:phút:giây). Khi người dùng click vào nút này thì yêu cầu người dùng nhập vào một xâu, sau đó thay nội dung trong thẻ H1 bằng xâu nhập vào này (Theo 4 cách: thay thuộc tính innerText, innerHTML, outerText, outerHTML).

Bài số 2:

Tạo một thẻ H1, màu chữ xanh dùng để hiển thị thời gian của hệ thống (gồm giờ:phút:giây).

Gợi ý: Viết hàm **CapNhat** để lấy giờ:phút:Giây trong máy tính sau đó gán cho thuộc tính innerText của thẻ H1. Sử dụng hàm setInterval("CapNhat();", 1000) để liên tục cập nhật thời gian theo từng giây.

Bài số 3: Tạo 2 tầng trong IE, mỗi tầng chứa một bức ảnh. Tầng thứ nhất chạy từ trái sang phải màn hình, tầng thứ hai chạy từ trên xuống dưới màn hình.

Gợi ý: Sử dụng 2 hàm setInterval để gọi 2 hàm di chuyển 2 tầng.

Bài số 4: Tạo một tầng chứa 3 liên kết. Chuột click tại vị trí nào thì tầng sẽ được đặt tại vị trí đó. **Gợi ý:** Viết lệnh trong sự kiện onClick của thẻ BODY.

Bài số 5: Tạo một form đăng ký E-Mail tương tự như của Yahoo (Bạn chỉ cần tạo một số phần tử, không cần tạo hết). Mỗi khi người dùng di chuyển chuột đến một phần tử nào đó thì hiển thị một lời chú thích bằng Tiếng việt. (Xem Bài số 5)

Bài số 6: Tạo 3 tầng (Trong Netscape), mỗi tầng nạp một trang tương ứng như sau: <http://www.vnn.vn>, <http://www.echip.com.vn> và <http://www.manguon.com>.

Bài số 7: Tạo một tầng chứa một bức ảnh, một nút có nhãn là "Di chuyển". Khi người dùng click vào nút này thì bức ảnh sẽ di chuyển chéo từ góc trên bên phải xuống góc dưới bên trái của màn hình. **Gợi ý:** Tăng dần pixelTop, giảm pixelLeft.

Bài số 8: Hãy làm hết các bài tập trong sách giáo khoa của cuốn giáo trình HTML, DHTML & JavaScript.