



Xử lý sự kiện trong trang HTML với JavaScript

Bởi:

Khoa CNTT ĐHSP KT Hưng Yên

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:

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, thả chuột, nhấn phím, thả 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).

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:

<i>Tên Sự kiện</i>	<i>Chỉ áp dụng cho phân tử</i>	<i>Mô tả</i>
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 phân 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 phân tử.
onchange	Text field, textarea, select list	Nó được kích hoạt khi người sử dụng thay đổi giá trị của phân tử.
onfocus	Window, frame, all form element	Nó được kích hoạt khi người sử dụng đặt focus vào một cửa sổ, khung, hay phân 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 phân tử.
onmouseover	Area, layer, link	Nó được kích hoạt khi người sử dụng di chuyển con trỏ khắp một phân 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 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 ?

Xử lý sự kiện trong trang HTML với JavaScript

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:

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 !?

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 xoá' ">
```

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 xoá'

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.

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 qui 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).

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="Thu" onClick="document.title='Ban da click chuột' ">
```

```
</BODY>
```

```
</HTML>
```

Vi 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 ở dưới và quan sát tiêu đề</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

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

Xử lý sự kiện trong trang HTML với JavaScript

```
<input type=text onclick = "alert('Ban da click chuot vao 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 =Do 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>
```

Xử lý sự kiện trong trang HTML với JavaScript

```
</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 onclick = "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>
```


Xử lý sự kiện trong trang HTML với JavaScript

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

</head>

<script language="JavaScript">

functionKiemTra()

{

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">

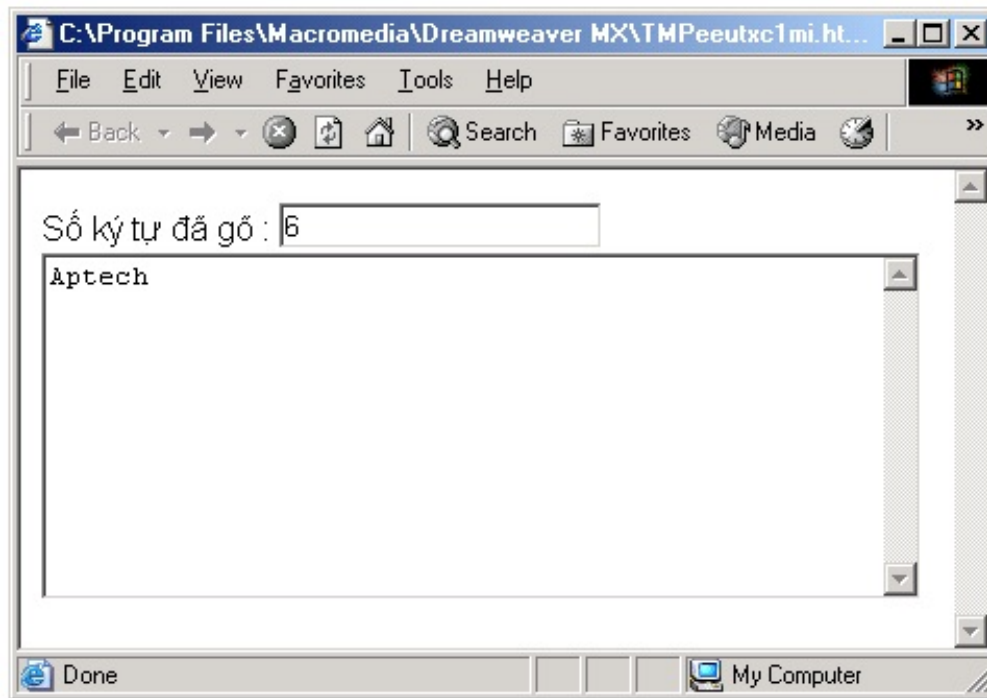
Số ký tự đã gõ : <input type="text" name="SoKyTu"> <BR>

<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:



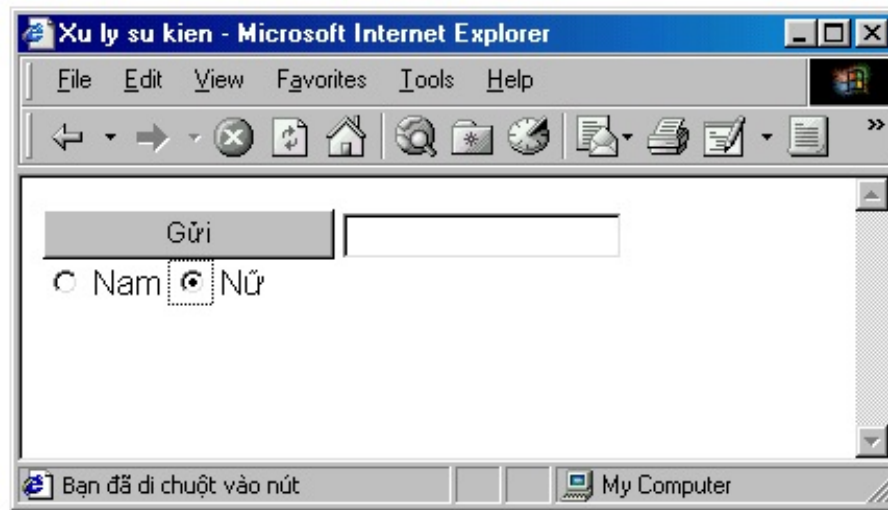
[<Xem kết quả>](#)

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:

Xử lý sự kiện trong trang HTML với JavaScript

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>

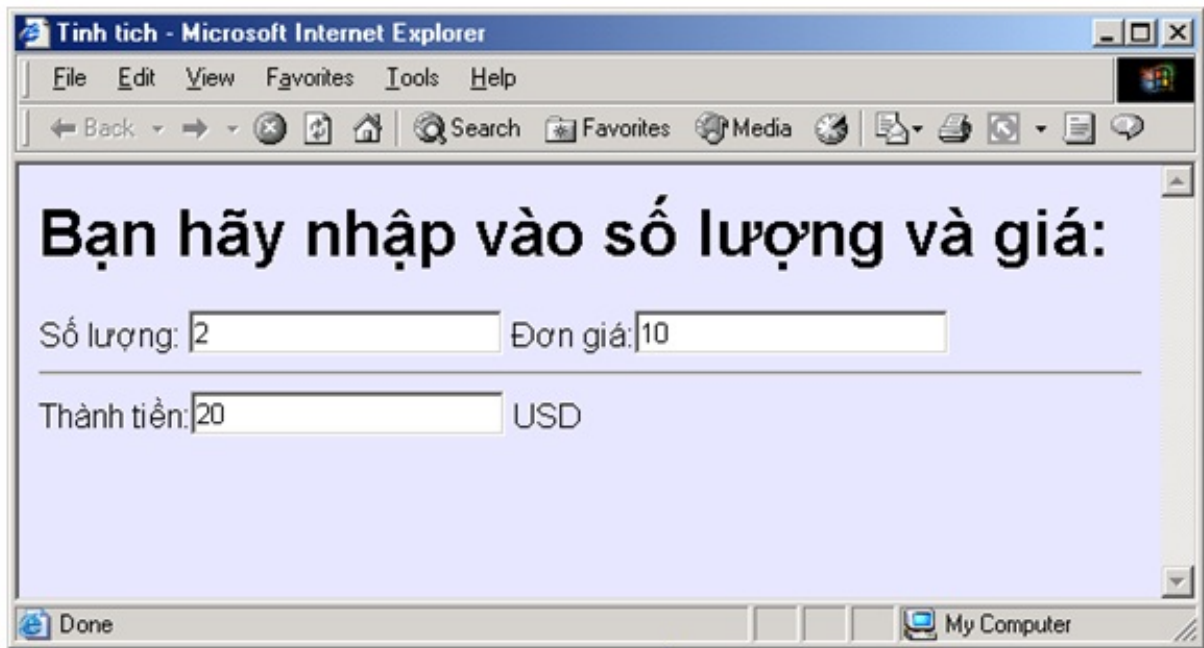
Số lượng: <input name="SoLuong">

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
```

```
}
```

Xử lý sự kiện trong trang HTML với JavaScript

```
</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>
```

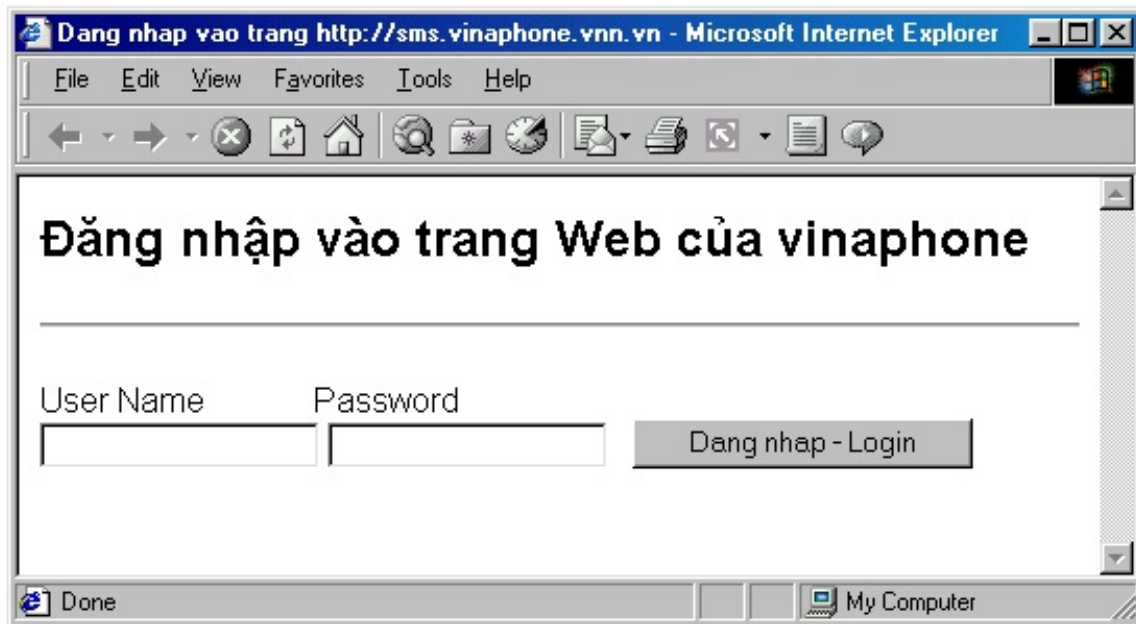
```
</html>
```



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)

Minh 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**

<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="Đăng nhập - Login">

Xử lý sự kiện trong trang HTML với JavaScript

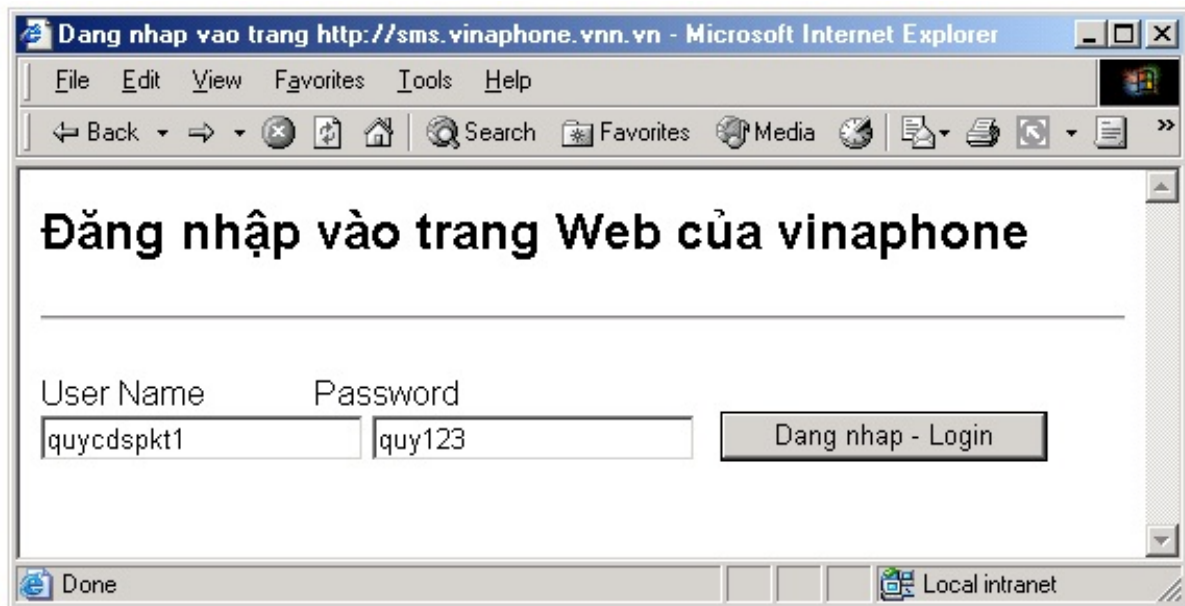
</p>

</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:

The screenshot shows a Microsoft Internet Explorer window titled "Đăng ký hòm thư - Microsoft Internet Explorer". The browser's address bar is empty. The main content area displays a registration form with the following elements:

- A red horizontal line at the top of the form area.
- The title "Đăng ký hòm thư" in a large, bold font.
- A "User Name" label followed by a text input field.
- A "Password:" label followed by a text input field, and a "Gõ lại Password:" label followed by another text input field.
- A "Giới tính:" label with two radio buttons: "Nam" (selected) and "Nu".
- A "Ngày / tháng / năm sinh:" label followed by three separate text input fields for day, month, and year.
- A second red horizontal line below the input fields.
- A "Đăng ký" button at the bottom left of the form area.

The browser's status bar at the bottom shows "Done" and "My Computer".

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 hiển thị dòng nhắc là "Nhập mã người dùng", hay khi người dùng 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:

Xử lý sự kiện trong trang HTML với JavaScript

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.