

THIẾT KẾ WEB

XỬ LÝ SỰ KIỆN VÀ BIỂU THỨC CHÍNH QUY

BỘ MÔN KỸ THUẬT VÀ MẠNG MÁY TÍNH

BÀI 7: XỬ LÝ SỰ KIỆN VÀ BIỂU THỨC CHÍNH QUY

- Xử lý sự kiện
- Biểu thức chính qui

PHẦN 1: XỬ LÝ SỰ KIỆN

➤ Khái niệm sự kiện

- ✓ Sự kiện xuất hiện khi người sử dụng tương tác với trang web. Hầu hết các sự kiện được tạo ra khi có thao tác với chuột hay bàn phím.
- ✓ Mỗi sự kiện được kết hợp với một đối tượng Sự kiện (Event). Đối tượng Event cung cấp thông tin về:
 - ✓ Kiểu sự kiện
 - ✓ Vị trí của con trỏ tại thời điểm xảy ra sự kiện

➤ Khái niệm sự kiện (tiếp)

- ✓ Quá trình điều khiển sự kiện được gọi là xử lý sự kiện, xử lý sự kiện được thực thi ngay khi sự kiện được kích hoạt bởi các trình xử lý sự kiện.
- ✓ Trình xử lý sự kiện có thể là đoạn mã javascript viết trực tiếp hoặc một hàm viết sẵn.

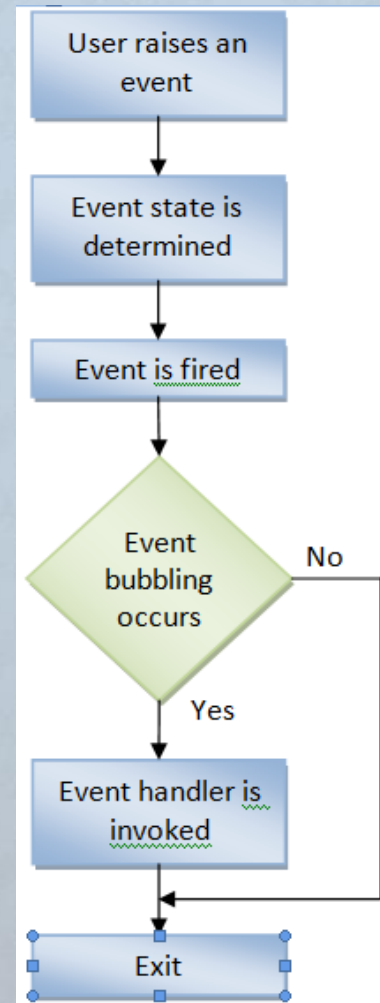
➤ Chu trình sống của sự kiện

- ✓ Chu trình sống của sự kiện thông thường bao gồm các bước:
 - ✓ Hành động của người sử dụng hoặc một điều kiện tương ứng khi sự kiện xảy ra
 - ✓ Đối tượng Event được cập nhật ngay lập tức để phản ánh sự kiện
 - ✓ Phát sinh sự kiện
 - ✓ Trình xử lý sự kiện tương ứng được gọi
 - ✓ Trình xử lý sự kiện thực hiện các hành động và trả về kết quả

➤ Chu trình sống của sự kiện

- ✓ Nổi bật sự kiện: khi một sự kiện xuất hiện ở thành phần bên trong của một thẻ thì sự kiện đó tiếp tục xuất hiện ở thẻ bao bên ngoài nó. Ví dụ: nếu sự kiện onclick (sự kiện kích chuột) được đặt trong thẻ form và đồng thời trong thẻ button của form thì: khi sự kiện onclick kích hoạt trên button thì nó cũng tiếp tục kích hoạt trên form, khi đó trình xử lý sự kiện onclick trên button và trên form cùng được thực thi

➤ Chu trình sống của sự kiện



➤ Điều khiển sự kiện

- Cú pháp

- Trình điều khiển sự kiện như là thuộc tính của thẻ HTML:

`<TAG eventHandler="JavaScript Code">`

- Trình điều khiển sự kiện như là thuộc tính của đối tượng:

`object.eventhandler = function;`

➤ Các sự kiện bàn phím

- ✓ Sự kiện bàn phím xuất hiện khi phím trên bàn phím được nhấn hoặc thả ra.
- ✓ Một số sự kiện bàn phím:
 - **onKeyDown**: Xuất hiện khi nhấn phím xuống.
 - **onKeyUp**: Xuất hiện khi nhả phím ra.
 - **onKeyPress**: Xuất hiện sau khi phím đã được nhả ra

➤ Ví dụ sự kiện bàn phím

```
<!DOCTYPE html>
<html>
<body>

<p>A function is triggered when the user is pressing a key
in the input field.</p>

<input type="text" onkeypress="myFunction()">

<script>
function myFunction() {
    alert("You pressed a key inside the input field");
}
</script>

</body>
</html>
```


➤ Các sự kiện chuột

- ✓ Sự kiện chuột (Mouse) xuất hiện khi người xử dụng thao tác chuột trên thành phần của trang web
- ✓ Một vài sự kiện:
 - **onMouseDown**: Xuất hiện khi phím chuột được nhấn xuống.
 - **onMousUp**: Xuất hiện khi phím chuột được thả ra.
 - **onClick**: xuất hiện sau khi hoàn thành nhấn chuột .
 - **onDbIcClick**: xuất hiện khi nhấn đúp chuột
 - **onMouseOver** : xuất hiện khi di chuyển chuột qua vị trí của một thành phần web
 - **onMouseOut**: xuất hiện khi di chuyển chuột dời khỏi vị trí của một thành phần web.

➤ Ví dụ sự kiện chuột

```
<!DOCTYPE html>
<html>
<body>

<div onmouseover="mOver(this)" onmouseout="mOut(this)"
style="background-
color:#D94A38;width:120px;height:20px;padding:40px;">
Mouse Over Me</div>

<script>
function mOver(obj) {
    obj.innerHTML = "Thank You"
}

function mOut(obj) {
    obj.innerHTML = "Mouse Over Me"
}
</script>

</body>
</html>
```

Mouse Over Me

Thank You

➤ Các sự kiện focus và selection

- Sự kiện focus được kích hoạt khi một thành phần của form trở thành thành phần hoạt động hiện tại.
- Sự kiện selection xuất hiện khi một thành phần được lựa chọn
 - **onFocus**: xuất hiện khi một thành phần trở thành thành phần hoạt động hiện tại
 - **onBlur**: Xuất hiện khi thành phần hoạt động hiện tại không còn là thành phần hoạt động hiện tại nữa
 - **onSelectStart**: xuất hiện khi hành động lựa chọn bắt đầu.
 - **onSelect**: Xuất hiện khi thay đổi thành phần được lựa chọn.

➤ Ví dụ sự kiện focus

```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction(x) {
    x.style.background = "yellow";
}
</script>
</head>
<body>
```

```
Enter your name: <input type="text"
onfocus="myFunction(this)">
```

```
<p>When the input field gets focus, a function is
triggered which changes the background-color.</p>
```

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

Enter your name:

When the input field gets focus, a function is triggered which changes background-color.

Enter your name:

When the input field gets focus, a function is triggered which changes background-color.

➤ Sự kiện onchange: xuất hiện khi có sự thay đổi dữ liệu

```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction() {
    var x = document.getElementById("fname");
    x.value = x.value.toUpperCase();
}
</script>
</head>
<body>
```

```
Enter your name: <input type="text" id="fname"
onchange="myFunction()">
<p>When you leave the input field, a function is triggered
which transforms the input text to upper case.</p>
```

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

Enter your name:

When you leave the input field, a function is triggered which transform input text to upper case.

Enter your name:

When you leave the input field, a function is triggered which transform input text to upper case.

PHẦN 2: BIỂU THỨC CHÍNH QUI

➤ Khái niệm Biểu thức chính quy

- Biểu thức chính quy là một chuỗi các ký tự tạo thành một mẫu tìm kiếm.
- Biểu thức chính quy có thể là một ký tự đơn hoặc một mẫu phức tạp hơn.
- Biểu thức chính quy dùng mô tả những gì bạn đang tìm kiếm và có thể được sử dụng để thực hiện tất cả các hoạt động tìm kiếm văn bản và thay thế văn bản

➤ Cách Tạo một biểu thức chính quy

- Cách 1: biểu thức chính qui được viết trong cặp `//`.
`Var re=/exp/;`
- Cách 2: sử dụng đối tượng `RegExp`
`var re = new RegExp("exp");`

➤ Sử dụng các kí tự đặc biệt

TT	Biểu thức chính quy	Mô tả
1	.	Khớp (match) với bất kỳ ký tự nào
2	^regex	Biểu thức chính quy phải khớp tại điểm bắt đầu
3	regex\$	Biểu thức chính quy phải khớp ở cuối dòng.
4	[abc]	Thiết lập định nghĩa, có thể khớp với a hoặc b hoặc c.
5	[abc][vz]	Thiết lập định nghĩa, có thể khớp với a hoặc b hoặc c theo sau là v hay z.
6	[^abc]	Khi dấu ^ xuất hiện như là nhân vật đầu tiên trong dấu ngoặc vuông, nó phủ nhận mô hình. Điều này có thể khớp với bất kỳ ký tự nào ngoại trừ a hoặc b hoặc c.
7	[a-d1-7]	Phạm vi: phù hợp với một chuỗi giữa a và điểm d và con số từ 1 đến 7.
8	X Z	Tìm X hoặc Z.
9	XZ	Tìm X và theo sau là Z.
10	\$	Kiểm tra kết thúc dòng.

➤ Sử dụng các kí tự đặc biệt

11	\d	Số bất kỳ, viết ngắn gọn cho [0-9]
12	\D	Ký tự không phải là số, viết ngắn gọn cho [^0-9]
13	\s	Ký tự khoảng trắng, viết ngắn gọn cho [\t\n\x0b\r\f]
14	\S	Ký tự không phải khoảng trắng, viết ngắn gọn cho [^\s]
15	\w	Ký tự chữ, viết ngắn gọn cho [a-zA-Z_0-9]
16	\W	Ký tự không phải chữ, viết ngắn gọn cho [^\w]
17	\S+	Một số ký tự không phải khoảng trắng (Một hoặc nhiều)
18	\b	Ký tự thuộc a-z hoặc A-Z hoặc 0-9 hoặc _, viết ngắn gọn cho [a-zA-Z0-9_].
19	*	Xuất hiện 0 hoặc nhiều lần, viết ngắn gọn cho {0,}
20	+	Xuất hiện 1 hoặc nhiều lần, viết ngắn gọn cho {1,}
21	?	Xuất hiện 0 hoặc 1 lần, ? viết ngắn gọn cho {0,1}.
22	{X}	Xuất hiện X lần, {}
23	{X,Y}	Xuất hiện trong khoảng X tới Y lần.
24	*?	* có nghĩa là xuất hiện 0 hoặc nhiều lần, thêm ? phía sau nghĩa là tìm kiếm khớp nhỏ nhất.

➤ Làm việc với biểu thức chính quy

Phương thức	Mô tả
<code>exec</code>	Một phương thức của RegExp dùng để tìm kiếm chuỗi phù hợp với mẫu so khớp. Nó trả về một mảng chứa kết quả tìm kiếm.
<code>test</code>	Một phương thức của RegExp dùng để kiểm tra mẫu có khớp với chuỗi hay không. Nó trả về giá trị true hoặc false.
<code>match</code>	Một phương thức của chuỗi dùng để tìm kiếm chuỗi phù hợp với mẫu so khớp. Nó trả về một mảng chứa kết quả tìm kiếm hoặc null nếu không tìm thấy.
<code>search</code>	Một phương thức của chuỗi dùng để tìm kiếm chuỗi phù hợp với mẫu so khớp và trả về vị trí của chuỗi đó hoặc -1 nếu không tìm thấy.
<code>replace</code>	Một phương thức của chuỗi dùng để tìm kiếm một chuỗi theo mẫu so khớp và thay thế chuỗi con được khớp với một chuỗi thay thế.
<code>split</code>	Một phương thức của chuỗi dùng một biểu mẫu chính quy hoặc một chuỗi bất biến để ngắt chuỗi đó thành một mảng các chuỗi con.

➤ Ví dụ biểu thức chính quy

- Kiểm tra độ mạnh mật khẩu:

re=/^(?=.*[A-Z].*[A-Z])(?=.*[!@#\$%&*])(?=.*[0-9].*[0-9])(?=.*[a-z].*[a-z].*[a-z]).{8}\$

- Kiểm tra email:

/[A-Z0-9._%+-]+@[A-Z0-9-]+.+. [A-Z]{2,4}

Xin cảm ơn