

49

Event (Sự kiện)

Event (sự kiện)

- Sự kiện của windows (cửa sổ): load, unload
- Sự kiện của người dùng (user): dùng chuột và bàn phím

50

Event	Purpose	Applies To
onload	Document has finished loading (if used in a frameset, all frames have finished loading).	<body> <frameset>
onunload	Document is unloaded, or removed, from a window or frameset.	<body> <frameset>
onclick	Button on mouse (or other pointing device) has been clicked over the element.	Most elements
ondblclick	Button on mouse (or other pointing device) has been double-clicked over the element.	Most elements
onmousedown	Button on mouse (or other pointing device) has been depressed (but not released) over the element.	Most elements
onmouseup	Button on mouse (or other pointing device) has been released over the element.	Most elements
onmouseover	Button on mouse (or other pointing device) has been moved onto the element.	Most elements
onmousemove	Button on mouse (or other pointing device) has been moved while over the element.	Most elements
onmouseout	Button on mouse (or other pointing device) has been moved off the element.	Most elements
onkeypress	A key is pressed and released over the element.	Most elements
onkeydown	A key is held down over an element.	Most elements
onkeyup	A key is released over an element.	Most elements

Event (sự kiện - tt)

51

Event	Purpose	Applies To
onfocus	Element receives focus either by mouse (or other pointing device) clicking it, tabbing order giving focus to that element, or code giving focus to the element.	<a> <area> <button> <input> <label> <select> <textarea>
onblur	Element loses focus.	<a> <area> <button> <input> <label> <select> <textarea>
onsubmit	A form is submitted.	<form>
onreset	A form is reset.	<form>
onselect	User selects some text in a text field.	<input> <textarea>
onchange	A control loses input focus and its value has been changed since gaining focus.	<input> <select> <textarea>

Event (sự kiện)

52

► Gán function vào sự kiện: `event = "functionname()"`

Ví dụ:

- `<input type="submit" onClick="exampleFunction()" />`
- `document.getElementById("myBtn").onclick = displayDate;`
- `<h1 onclick="this.innerHTML = 'Oops!'">Click on this text!</h1>`

EventListener

53

- ▶ Đăng kí sự kiện cho một element.
- ▶ Cùng một sự kiện có thể có nhiều function khác nhau (→ không viết lại function)
- ▶ Cú pháp:

element.addEventListener(event, function, useCapture);

- ▶ Ví dụ:

```
element.addEventListener("click", myFunction);  
  
function myFunction() {  
    alert ("Hello World!");  
}
```

EventListener

54

- ▶ Gán nhiều function cho 1 sự kiện:

```
element.addEventListener("click", myFunction);  
element.addEventListener("click", mySecondFunction);
```

- ▶ Gán nhiều sự kiện cho một element

```
element.addEventListener("mouseover", myFunction);  
element.addEventListener("click", mySecondFunction);  
element.addEventListener("mouseout", myThirdFunction);
```

55

DOM (Document Object Model)

56

Document Object Model (DOM)

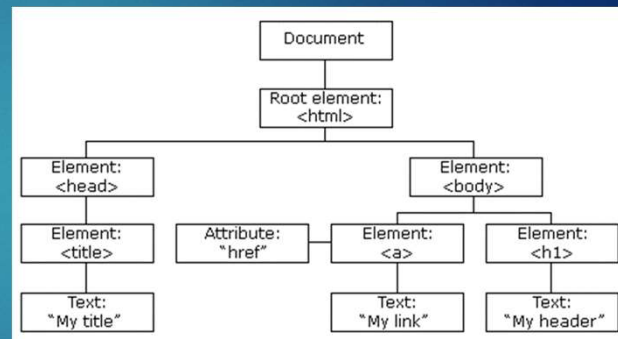
- ▶ DOM là nền tảng và giao diện ngôn ngữ trung lập cho phép các script truy cập và cập nhật nội dung, cấu trúc và style của tài liệu.
- ▶ HTML DOM là mô hình đối tượng chuẩn và giao diện lập trình cho HTML, định nghĩa:
 - ▶ Các thẻ/thành phần của HTML như là **objects**
 - ▶ Các **properties** cho tất cả các thẻ/thành phần
 - ▶ Các **methods** để truy cập đến các thẻ/thành phần
 - ▶ Các **events** cho các thẻ/thành phần

DOM

57

► Javascript có thể:

- Thay đổi tất cả các *thẻ* trong trang web
- Thay đổi tất cả các *thuộc tính* của các thẻ
- Thay đổi CSS
- **Add/remove** các thẻ và thuộc tính
- Thay đổi sự kiện
- Tạo sự kiện mới.



DOM

58

► Phân cấp các element trong tài liệu HTML

- Đối tượng Document cấp cao nhất, đại diện cho trang hiện hành
- Các đối tượng con của Document được tham chiếu như một tập hợp có tên trùng với tên thẻ.

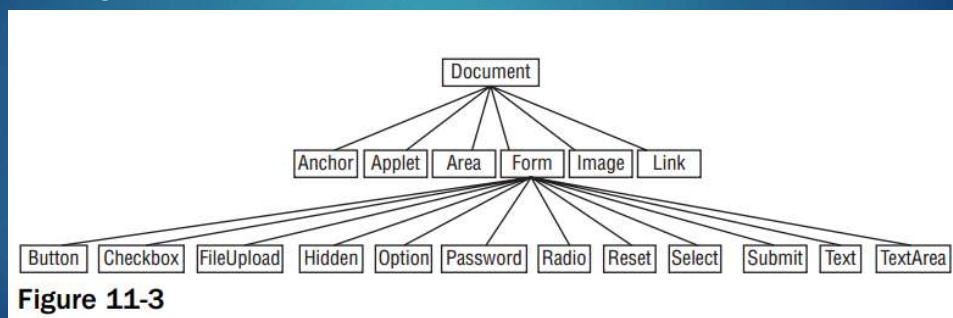


Figure 11-3

DOM

59

- ▶ Form: tập hợp các form có trong tài liệu
- ▶ Image: Tập hợp các hình ảnh
- ▶ Link: Tập các hyperlink
- ▶ Anchor: Tập hợp các liên kết của thẻ a có xác định id, hoặc name
- ...

DOM

60

- ▶ Các tập hợp được xem như là mảng (array)
- ▶ Truy cập đến 01 đối tượng: đi từ cấp cao đến cấp thấp, dùng dấu . liên kết.

```
document.form[0].txtname.value  
document.frmLogin.txtuser.value
```

Objects, Methods, and Properties

61

- ▶ Mỗi đối tượng có thuộc tính và phương thức
 - ▶ Thuộc tính: các mô tả về đối tượng
 - ▶ Phương thức: thực hiện một action
- ▶ Giới thiệu về đối tượng document, form, image

Đối tượng document

62

- ▶ Các thuộc tính

VD:

`document.title = "Page1"`

Property Name	Purpose	Read/Write
<code>alinkColor</code>	Specifies link colors. (Like the deprecated <code>alink</code> attribute on the <code><body></code> element.)	Read/write
<code>bgcolor</code>	Specifies background color. (Like the deprecated <code>bgcolor</code> attribute on the <code><body></code> element.)	Read/write
<code>fgcolor</code>	Foreground/text color. (Like the deprecated <code>text</code> attribute of the <code><body></code> element.)	Read/write
<code>lastModified</code>	The date the document was last modified. (This is usually sent by the web server in things known as HTTP headers that you do not see.)	Read only
<code>linkColor</code>	Specifies link colors. (Like the deprecated <code>link</code> attribute of the <code><body></code> element.)	Read/write
<code>referrer</code>	The URL of the XHTML page that users came from if they clicked a link. Empty if there is no referrer.	Read only
<code>title</code>	The title of the page in the <code><title></code> element.	Read only (until IE 5 and Netscape 6 and later versions)
<code>vlinkColor</code>	The <code>vlink</code> attribute of the <code><body></code> element (deprecated).	Read/write

Đối tượng document

63

- Các phương thức:

VD: `document.write("this is a document ")`

Method or property	Description
<code>getElementById(id)</code>	Returns the DOM node representing the XHTML element whose <code>id</code> attribute matches <code>id</code> .
<code>write(string)</code>	Writes the string to the XHTML document as XHTML code.
<code>writeln(string)</code>	Writes the string to the XHTML document as XHTML code and adds a newline character at the end.
<code>cookie</code>	A string containing the values of all the cookies stored on the user's computer for the current document. See Section 11.9, Using Cookies.
<code>lastModified</code>	The date and time that this document was last modified.

Đối tượng form

64

- Có nhiều form trên cùng tài liệu: search, login, register...
- Truy cập đến tập hợp form như array hoặc dùng tên form
`document.forms[0].action`
`document.frmlogin.action`

Property Name	Purpose	Read/Write
<code>action</code>	The <code>action</code> attribute of the <code><form></code> element	Read/write
<code>length</code>	Gives the number of form controls in the form	Read only
<code>method</code>	The <code>method</code> attribute of the <code><form></code> element	Read/write
<code>name</code>	The <code>name</code> attribute of the <code><form></code> element	Read only
<code>target</code>	The <code>target</code> attribute of the <code><form></code> element	Read/write

Đối tượng form

65

► Các phương thức:

Method Name	Purpose
<code>reset()</code>	Resets all form elements to their default values
<code>submit()</code>	Submits the form

Các đối tượng trong form

66

- Text fields
- Checkboxes,
- Radio button
- Button
- Select boxes

Property	Applies to	Purpose	Read/Write
<code>checked</code>	Checkboxes and radio buttons	Returns <code>true</code> when checked or <code>false</code> when not	Read/write
<code>disabled</code>	All except hidden elements	Returns <code>true</code> when disabled and user cannot interact with it (supported in IE4 and Netscape 6 and later versions only)	Read/write
<code>form</code>	All elements	Returns a reference to the form it is part of	Read only
<code>length</code>	Select boxes	Number of options in the <code><select></code> element	Read only
<code>name</code>	All elements	Accesses the name attribute of the element	Read only
<code>selectedIndex</code>	Select boxes	Returns the index number of the currently selected item	Read/write
<code>type</code>	All	Returns type of form control	Read only
<code>value</code>	All	Accesses the value attribute of the element or content of a text input	Read/write

Các đối tượng trong form

67

► Các phương thức

Property Name	Applies to	Read/Write
<code>blur()</code>	All except hidden	Takes focus away from currently active element to next in tabbing order
<code>click()</code>	All except text	Simulates the user's clicking the mouse over the element
<code>focus()</code>	All except hidden	Gives focus to the element
<code>select()</code>	Text elements except hidden	Selects the text in the element

Các đối tượng trên form

68

► Ví dụ: viết trang html và javascript có form nhập họ tên, khi click vào button sẽ hiển thị tên đã nhập

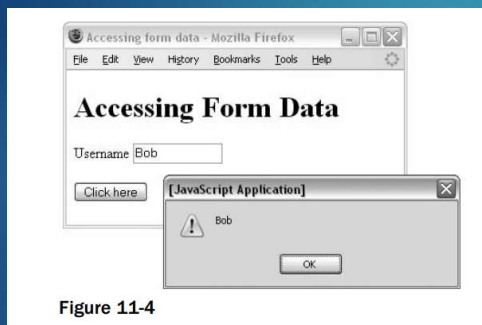


Figure 11-4

Đối tượng image

69

- ▶ Tập hợp image trong tài liệu được xem như 1 array. Truy cập đến các phần tử như array hoặc tên cụ thể:

`document.images[0].src` hoặc `document.imgHome.src`

Property	Purpose	Read/write
<code>border</code>	The <code>border</code> attribute of the <code></code> element	Read/write
<code>complete</code>	Indicates whether an image has loaded successfully	Read only
<code>height</code>	The <code>height</code> attribute of the <code></code> element	Read/write
<code>hspace</code>	The <code>hspace</code> attribute of the <code></code> element	Read/write
<code>lowsrc</code>	The <code>lowsrc</code> attribute of the <code></code> element (indicating a lower resolution version of the image)	Read/write
<code>name</code>	The <code>name</code> attribute of the <code></code> element	Read/write
<code>src</code>	The <code>src</code> attribute of the <code></code> element	Read/write

Đối tượng image

70

- ▶ VD: Viết trang html + javascript hiển thị ảnh **màu đỏ** (click here), khi rê chuột lên ảnh thì ảnh sẽ đổi sang **màu xanh**.



Figure 11-5

71

Quản lý các element

72

Tìm một element

Method

`document.getElementById(id)`

`document.getElementsByTagName(name)`

`document.getElementsByClassName(name)`

Description

Tìm dựa vào tên ID

Tìm dựa vào tên thẻ

Tìm dựa vào tên class

Tìm một element trong một Collections

```
var x = document.forms["frm1"];
```

Tìm một element

73

```
<p id="intro">Hello World!</p>

<p>This example demonstrates the <b>getElementById</b> method!</p>

<p id="demo"></p>

<script>
var myElement = document.getElementById("intro");
document.getElementById("demo").innerHTML =
"The text from the intro paragraph is " + myElement.innerHTML;
</script>
```

Hello World!

This example demonstrates the **getElementById** method!

The text from the intro paragraph is Hello World!

Thay đổi một element

74

Method	Description
<i>element.innerHTML = new html content</i>	Thay đổi nội dung
<i>element.attribute = new value</i>	Thay đổi thuộc tính
<i>element.setAttribute(attribute, value)</i>	Thay đổi thuộc tính
<i>element.style.property = new style</i>	Thay đổi style

Thay đổi một element

75

Đổi nội dung:

```
document.getElementById("p1").innerHTML = "New text!";
```

```
var element = document.getElementById("id01");
element.innerHTML = "New Heading";
```

Đổi style

```
document.getElementById("p2").style.color = "blue";
```

Thêm hoặc xóa bớt element

76

Method	Description
<code>document.createElement(<i>element</i>)</code>	Tạo mới một element
<code>document.removeChild(<i>element</i>)</code>	Remove một element
<code>document.appendChild(<i>element</i>)</code>	Chèn thêm một element
<code>document.replaceChild(<i>element</i>)</code>	Thay thế một element
<code>document.write(<i>text</i>)</code>	In ra văn bản