

LẬP TRÌNH WEB VỚI HTML5, CSS3, jQuery và Bootstrap

Bài 6. *Xử lý web page với ngôn ngữ JavaScript*

Ngành Lập trình & CSDL

<http://www.t3h.vn>



Nội dung

- 1. Giới thiệu tổng quan về ngôn ngữ lập trình JavaScript**
2. Một số hộp thoại thường dùng: Alert box, Confirm box, Prompt box
3. Sử dụng JavaScript trong trang HTML
4. Ngôn ngữ lập trình JavaScript



Nội dung

5. Các đối tượng cơ bản

6. Các đối tượng trong JavaScript

7. Các sự kiện trên trang HTML

8. Các xử lý cơ bản của DHTML



1. Giới thiệu về JavaScript

- ❑ Của hãng Netscape Communications
- ❑ Là một ngôn ngữ kịch bản (scripting language) dùng để tương tác với các trang HTML dựa trên đối tượng (object-based scripting language)
- ❑ Chủ yếu dùng cho kỹ thuật lập trình ở phía client
- ❑ Code của JavaScript thường được nhúng (embedded) trực tiếp hoặc tích hợp (integrated) vào trang web



1 Giới thiệu về JavaScript

❑ Có một số đặc điểm sau:

- Là một ngôn ngữ thông dịch(interpreted language), nghĩa là các script thi hành không cần biên dịch trước (precompile).
Trình duyệt dịch script, phân tích và thi hành ngay tức thời
- Có phân biệt chữ HOA và thường



1 Giới thiệu về JavaScript

❑ Các trình duyệt hỗ trợ JavaScript:

- Netscape Navigator (bắt đầu từ phiên bản 2.0)
- Microsoft Internet Explorer (bắt đầu từ phiên bản 3.0)
- Những trình duyệt khác có hỗ trợ JavaScript (như Opera, ...)
- Gần như tất cả các trình duyệt hiện nay đều hỗ trợ JavaScript



Nội dung

1. Giới thiệu tổng quan về ngôn ngữ lập trình JavaScript
2. Một số hộp thoại thường dùng: Alert box, Confirm box, Prompt box
3. Sử dụng JavaScript trong trang HTML
4. Ngôn ngữ lập trình JavaScript



2. Một số hộp thoại thường dùng

- ❑ Hộp thoại thông báo – Alert box
- ❑ Hộp thoại xác nhận – Confirm box
- ❑ Hộp thoại nhận giá trị - Prompt box



2. Một số hộp thoại thường dùng

❑ Hộp thoại thông báo – Alert box

- Xuất câu thông báo trong một cửa sổ

alert("Câu thông báo")



2. Một số hộp thoại thường dùng

❑ Hộp thoại xác nhận – Confirm box

- Có dạng hàm (true/false), giúp người dùng quyết định đồng ý hoặc từ chối một yêu cầu

confirm("Câu hỏi xác nhận")

❑ Hộp thoại xác nhận – Confirm box

- Có dạng hàm (true/false), giúp người dùng quyết định đồng ý hoặc từ chối một yêu cầu

confirm("Câu hỏi xác nhận")



2. Một số hộp thoại thường dùng

❑ Hộp thoại nhận giá trị – Confirm box

- Có dạng hàm, yêu cầu nhập vào một giá trị

Biến = prompt(*“Câu hướng dẫn”, “giá trị mặc định”*)



3. Sử dụng JavaScript trong trang HTML

- ☐ Chèn JavaScript vào sự kiện
- ☐ Chèn JavaScript vào Head section
- ☐ Chèn JavaScript vào Body section
- ☐ Sử dụng tập tin thư viện – External script
- ☐ Chú thích trong JavaScript



3. Sử dụng JavaScript trong trang HTML

❑ Chèn JavaScript vào sự kiện

- HTML cho phép chèn JavaScript vào các sự kiện của tag
- Code JavaScript sẽ được thi hành khi phát sinh sự kiện

```
<html>
  <head><meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Ví dụ về script</title>
  </head>
  <body > Nội dung trang web
    <input onclick="alert('chào mừng đến với script')" type="button"
value="Chọn"/>
  </body>
</html>
```



3. Sử dụng JavaScript trong trang HTML

❑ Chèn JavaScript vào head section

- Đặt đoạn lệnh JavaScript trong cặp tag <script>

```
<html>
  <head><meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Ví dụ về script</title>
    <script language="javascript" type="text/javascript">
      //Đoạn JavaScript
    </script>
  </head>
  <body> Nội dung trang web </body>
</html>
```



3. Sử dụng JavaScript trong trang HTML

❑ Chèn JavaScript vào body section

- Đặt cặp tag <script> trong cặp tag <body>

```
<html>
  <head> <title>Ví dụ về script</title></head>
  <body>
    <script language="javascript" type="text/javascript">
      //Đoạn JavaScript
    </script>
    Nội dung trang web
    <script language="javascript" type="text/javascript">
      //Đoạn JavaScript
    </script>
    ...
  </body></html>
```



3. Sử dụng JavaScript trong trang HTML

❑ Sử dụng tập tin thư viện – External script

- Các đoạn code dùng chung sẽ được viết trong một tập tin và lưu với phần mở rộng là .js
- Sử dụng thuộc tính **src** (source) của tag <script> để khai báo tập tin *.js

```
<html>
  <head><title>Ví dụ về script</title>
  <script language="javascript" src="<tên tập tin>.js"></script>
  </head>
  <body> Nội dung trang web  </body>
</html>
```




3 Sử dụng JavaScript trong trang HTML

❑ Chú thích trong JavaScript

- Chú thích trong JavaScript
- Chú thích cho một dòng, ta dùng ký hiệu **//**
- Chú thích từ 2 dòng trở lên, bắt đầu bằng ký hiệu **/*** và kết thúc bằng ***/**

Ví dụ:

```
<script language="javascript">
```

```
    //In dòng tiêu đề
```

```
    document.write("<h1>Đây là dòng tiêu đề</h1>");
```

```
    /* Hai dòng lệnh sau in hai đoạn văn bản */
```

```
    document.write("<p>Đây là đoạn văn thứ nhất </p>");
```

```
    document.write("<p>Đây là đoạn văn thứ hai </p>");
```

```
</script>
```





Nội dung

1. Giới thiệu tổng quan về ngôn ngữ lập trình JavaScript
2. Một số hộp thoại thường dùng: Alert box, Confirm box, Prompt box
3. Sử dụng JavaScript trong trang HTML
4. Ngôn ngữ lập trình JavaScript



4 Ngôn ngữ lập trình JavaScript

- ☐ Biến - Variable
- ☐ Các toán tử - Operators
- ☐ Một số hàm cơ bản trong JavaScript
- ☐ Hàm - Function
- ☐ Cấu trúc điều khiển



4 Ngôn ngữ lập trình JavaScript

❑ Biến - Variable

- Khai báo:

```
var <tên biến>;  
hoặc var <tên biến> = <giá trị>;
```

- <tên biến> phải bắt đầu bằng ký tự (a -> z) hoặc dấu gạch dưới (_)
- <tên biến> có phân biệt chữ HOA/thường



4 Ngôn ngữ lập trình JavaScript

❑ Biến - Variable

- Khai báo bên ngoài hàm gọi là biến toàn cục (global variable)
- Khai báo trong hàm gọi là biến cục bộ (local variable) và chỉ được phép sử dụng trong hàm khai báo nó



4 Ngôn ngữ lập trình JavaScript

❑ Biến - Variable

- Giá trị của biến (kiểu dữ liệu trong JS):
 - Số - Number : một con số (như 23 hoặc 3.1415)
 - Luận lý – Boolean : true hoặc false
 - Chuỗi – String : một chuỗi các ký tự (như “abc”)
 - null : khi không xác định giá trị
 - undefined : khi không xác định giá trị



4 Ngôn ngữ lập trình JavaScript

❑ Biến - Variable

- Phân biệt giá trị null và undefined:
 - undefined: biến được khai báo nhưng chưa được gán giá trị

Ví dụ: `var TestVar;`
 `alert(TestVar);` ➔ `undefined`

- null: biến được khai báo và được gán = null

Ví dụ: `var TestVar = null;`
 `alert(TestVar);` ➔ `null`



4 Ngôn ngữ lập trình JavaScript

❑ Biến - Variable

- Kiểm tra kiểu dữ liệu: dùng toán tử typeof

hoặc

```
typeof <giá trị>  
typeof (<giá trị>)
```

Ví dụ:

```
document.write(typeof 10);
```

→ “number”

```
document.write(typeof ('JS'));
```

→ “string”

```
document.write(typeof new Date());
```

→ “object”

```
document.write(typeof myVar);
```

→ “undefined”



4 Ngôn ngữ lập trình JavaScript

❑ Các toán tử - operators

- Toán tử số học

Toán tử	Ý nghĩa	Ví dụ
+	Cộng (cũng được sử dụng để nối chuỗi)	
-	Trừ	
*	Nhân	
/	Chia	
%	Chia lấy phần dư	7%3 → 1
++	Tăng thêm 1	x = 3; x++ → 4
--	Trừ đi 1	x = 3; x-- → 2



4 Ngôn ngữ lập trình JavaScript

❑ Các toán tử - operators

- Toán tử gán

Toán tử	Ví dụ	Biểu thức tương ứng
=	<code>x = 4</code>	
<code>+=</code>	<code>x += 4</code>	<code>x = x + 4</code>
<code>-=</code>	<code>x -= 4</code>	<code>x = x - 4</code>
<code>*=</code>	<code>x *= 4</code>	<code>x = x * 4</code>
<code>/=</code>	<code>x /= 4</code>	<code>x = x / 4</code>
<code>%=</code>	<code>x %= 4</code>	<code>x = x % 4</code>



4 Ngôn ngữ lập trình JavaScript

❑ Các toán tử - operators

- Toán so sánh

Toán tử	Ý nghĩa	Ví dụ
==	Trả về true nếu 2 toán hạng bằng nhau. Nếu khác kiểu, JavaScript sẽ chuyển sang kiểu thích hợp rồi mới so sánh	<code>x = 3</code> <code>y = "3"</code> <code>x == y → true</code>
!=	Trả về true nếu 2 toán hạng không bằng nhau. Cách thức so sánh cũng giống toán tử ==	<code>x != y → false</code>
===	Trả về true nếu 2 toán hạng cùng kiểu và cùng giá trị.	<code>x === y → false</code>
!==	Trả về true nếu 2 toán hạng không bằng nhau. Cách thức so sánh cũng giống toán tử ===	<code>x !== y → true</code>
<, >, <=, >=		



4 Ngôn ngữ lập trình JavaScript

❑ Các toán tử - operators

- Toán luận lý

Toán tử	Cách sử dụng	Ý nghĩa
&& (And)	<biểu thức 1> && <biểu thức 2>	Trà về true , nếu cả 2 biểu thức trả về true
(Or)	<biểu thức 1> <biểu thức 2>	Trà về true , nếu 1 trong 2 biểu thức trả về true
! (Not)	!<biểu thức>	Trà về false , nếu biểu thức là true và ngược lại



4 Ngôn ngữ lập trình JavaScript

❑ Các toán tử - operators

- Toán tử điều kiện

(<điều kiện>) ? <giá trị 1> : <giá trị 2>

❑ Một số hàm cơ bản trong JavaScript

- **eval("chuỗi")**: đánh giá biểu thức "chuỗi" và thi hành như là code JavaScript

❑ Một số hàm cơ bản trong JavaScript

- **isNaN(<giá trị>)**: kiểm tra <giá trị> không phải là số hay không (true/false). Nếu <giá trị> là chuỗi "", trả về true



4 Ngôn ngữ lập trình JavaScript

❑ Một số hàm cơ bản trong JavaScript

- **Number(<đối tượng>)**: chuyển đổi một đối tượng sang một con số, nếu không chuyển được, trả về NaN

❑ Một số hàm cơ bản trong JavaScript

- **parseInt("chuỗi" [,<ơ số>])**: phân tích "chuỗi" và trả về số nguyên, <ơ số> xác định hệ thống số. Chỉ phân tích số đầu tiên và nếu không chuyển được, trả về NaN

❑ Một số hàm cơ bản trong JavaScript

- **parseFloat("chuỗi")**: tương tự hàm parseInt nhưng trả về số thực



4 Ngôn ngữ lập trình JavaScript

❑ Hàm - Function

- **Khai báo:** Hàm có thể được khai báo trong tag <head> hoặc tag <body> hoặc trong tập tin *.js

```
function <tên hàm>( [ tham số 1, tham số 2, ... ] )  
{  
    /* khối lệnh xử lý */  
    [ return <giá trị trả về > ]  
}
```

Ví dụ: **function** dt_hinhvuong(canh)
 { **return** canh*canh; }



4 Ngôn ngữ lập trình JavaScript

❑ Hàm - Function

- Sử dụng:

<biến> = <tên hàm>([<giá trị>, ...])
hoặc **<tên hàm>([<giá trị>, ...])**

Ví dụ:

```
<input type="button" value="Tính diện tích"  
onClick = "alert(dt_hinhvuong(3))">
```




4 Ngôn ngữ lập trình JavaScript

❑ Cấu trúc điều khiển

- if ... else

```
if (<điều kiện>
    // Một lệnh xử lý;

Hoặc

if (<điều kiện>
{
    /* Khối lệnh xử lý; */
}
```



4 Ngôn ngữ lập trình JavaScript

❑ Cấu trúc điều khiển

- if ... else

Hoặc

```
if (<điều kiện>
{
    /* Khối lệnh xử lý 1; */
}
else
{
    /* Khối lệnh xử lý 2; */
}
```



4 Ngôn ngữ lập trình JavaScript

□ Cấu trúc điều khiển

- switch

```
switch (<biểu thức>)  
{  
    case <giá trị 1>:  
        /* Khối lệnh xử lý 1 ; */  
        break;  
    case <giá trị 2>:  
        /* Khối lệnh xử lý 2 ; */  
        break;  
    ...  
    [default:  
        /* Khối lệnh xử lý n ; */  
    ]  
}
```



4 Ngôn ngữ lập trình JavaScript

❑ Cấu trúc điều khiển

- Vòng lặp for

```
for (<giá trị bắt đầu> ; <điều kiện lặp> ; <bước nhảy>)  
{  
    /* Khối lệnh xử lý; */  
    [break;]  
}
```



4 Ngôn ngữ lập trình JavaScript

❑ Cấu trúc điều khiển

- Vòng lặp while và do ... while

```
while (<điều kiện>)  
{  
    /* Khối lệnh xử lý; */  
    [break;]  
}  
  
Hoặc  
  
do  
{  
    /* Khối lệnh xử lý; */  
    [break;]  
} while (<điều kiện>);
```



5 Các đối tượng cơ bản

❑ String

❑ Date

❑ Math



5 Các đối tượng cơ bản

❑ String

- Chuỗi là một đoạn text có thể có một hoặc nhiều ký tự và thông thường lưu trữ nó vào một biến, biến này gọi là biến có kiểu dữ liệu là String (chuỗi).
- Tất cả các chuỗi đều phải được bao quanh bằng cặp dấu nháy đơn ' hoặc nháy đôi “.
- Ví dụ:

`var chuoi = “Học lập trình javascript”;`

hoặc `var dtChuoi = new String(“Học lập trình javascript”);`



5 Các đối tượng cơ bản

❑ String

- Thuộc tính **length** giúp xác định số lượng ký tự của chuỗi đang có.
- Một số phương thức thường dùng:
 - **charAt(<vị trí>)**: trả về ký tự tại vị trí đưa vào.
 - **concat(<chuỗi 1>[, <chuỗi 2>,...])**: nối chuỗi hiện tại với các chuỗi con truyền vào.
 - **indexOf(<chuỗi con>)**: trả về vị trí đầu tiên tìm được chuỗi con trong chuỗi cha, trả về -1 nếu không tìm thấy.
 - **lastIndexOf(<chuỗi con>)**: trả về vị trí cuối cùng tìm được chuỗi con trong chuỗi cha, trả về -1 nếu không tìm thấy.
 - **replace(<giá trị cần thay thế>,<giá trị mới>)**: trả về chuỗi mới đã được thay các giá trị cần thay thế bằng giá trị mới.



5 Các đối tượng cơ bản

❑ String

- Một số phương thức thường dùng:
 - **split(<ký tự tách>)**: trả về mảng chuỗi con được tách theo ký tự truyền vào
 - **substr(<vị trí cắt>,<độ dài>)**: trả về chuỗi con cắt được từ vị trí với độ dài tương ứng.
 - **toLowerCase()**: trả về chuỗi viết thường
 - **toUpperCase()**: trả về chuỗi viết hoa
 - **trim(<ký tự>)**: trả về chuỗi đã loại bỏ ký tự truyền vào ở hai đầu chuỗi gốc.



5 Các đối tượng cơ bản

❑ String

- Vị trí của ký tự đầu tiên trong chuỗi luôn bắt đầu bằng **chỉ số 0**
- `search(<regExp>)`: tìm một “chuỗi” và trả về chỉ số tìm được.
Nếu không tìm thấy, trả về -1
 - `<regExp>`: là một biểu thức có qui tắc, chứa **/chuỗi tìm/** và option **i** ; cho phép tìm theo chữ HOA/thường
- `replace(<regExp>,”chuỗi thế”)`: tìm và thay thế.
 - `<regExp>`: chứa **/chuỗi tìm/** , option **i** ; cho phép tìm theo chữ HOA/thường, và option **g** ; cho phép thay thế toàn bộ



5 Các đối tượng cơ bản

❑ Date

- Đối tượng date quản lý kiểu dữ liệu thời gian
- Khai báo:
 - `new Date()`
 - `new Date(milliseconds)`
 - `new Date(dateString)`
 - `new Date(year, month, day, hours, minutes, seconds, milliseconds)`
- Lưu ý: <tháng> được tính từ **0** là tháng 1



5 Các đối tượng cơ bản

□ Date

- Một số phương thức thường dùng:
 - **getDate()**: trả về ngày theo dạng số từ 1->31
 - **getDay()**: trả về ngày trong tuần theo dạng số từ 0->6
 - **getFullYear()**: trả về năm đầy đủ 4 chữ số
 - **getHours()**: trả về số giờ của ngày
 - **getMinutes()**: trả về số phút của ngày
 - **getMonth()**: trả về số tháng từ 0->11
 - **getSeconds()**: trả về số giây của ngày
 - **getTime()**: lấy tổng thời gian từ 1-1-1970 đến hiện tại tính bằng milliseconds
- Để cài các thông số tương tự dùng phương thức **set** thay cho **get**



5 Các đối tượng cơ bản

❑ Math

- Đối tượng Math quản lý các phương thức xử lý liên quan đến toán học.
- Một số phương thức thường dùng:
 - **random()**: trả về số bất kỳ từ 0->1
 - **round(<số thập phân>)**: trả về số nguyên sau khi làm tròn
 - **pow(<số>,<số mũ>)**: trả về kết quả bình phương theo số mũ
 - **sqrt(<số>)**: trả về số sau khi tính căn bậc hai.
 - **max(<danh sách giá trị>)**: trả về giá trị lớn nhất trong danh sách.
 - **min(<danh sách giá trị>)**: trả về giá trị nhỏ nhất trong danh sách.



5 Các đối tượng cơ bản

❑ Math

- Ví dụ:

Ví dụ: phát sinh một số nguyên ngẫu nhiên trong khoảng từ 10 đến 100

```
var so = parseInt(Math.random()*91 +10);
```

Ví dụ:

Math.round(3.4) → 3

Math.round(3.5) → 4



6 Các đối tượng trong JavaScript

❑ Mô hình BOM

❑ Mô hình DOM



6 Các đối tượng trong JavaScript

❑ Mô hình BOM

- BOM là chữ viết tắt của **B**rowser **O**bject **M**odel, hay còn gọi là các đối tượng liên quan đến trình duyệt browser.
- Mỗi browser sẽ có những đối tượng khác nhau nên nó không có một chuẩn chung nào cả.



6 Các đối tượng trong JavaScript

❑ Mô hình BOM

- Một số loại BOM:

- window
- screen
- location
- history
- navigator
- popup
- timing
- cookies



6 Các đối tượng trong JavaScript

❑ Mô hình BOM

- BOM window

- Đối tượng Window được hỗ trợ hầu hết các trình duyệt và mỗi tab trình duyệt sẽ có một đối tượng Window riêng và không liên quan tới nhau.
- Đối tượng Window có rất nhiều thuộc tính và phương thức riêng. Ví dụ đối tượng **document** là một thuộc tính của window (window.document), vì vậy nên ta có thể ví Window là một đối tượng toàn cục và ở đâu trong website cũng có thể sử dụng được.



6 Các đối tượng trong JavaScript

❑ Mô hình BOM

- BOM window
 - Một số thuộc tính cơ bản:

Thuộc tính	Ý nghĩa
innerHeight	Chiều cao window
innerWidth	Chiều rộng window



6 Các đối tượng trong JavaScript

❑ Mô hình BOM

- BOM window
 - Một số phương thức cơ bản:

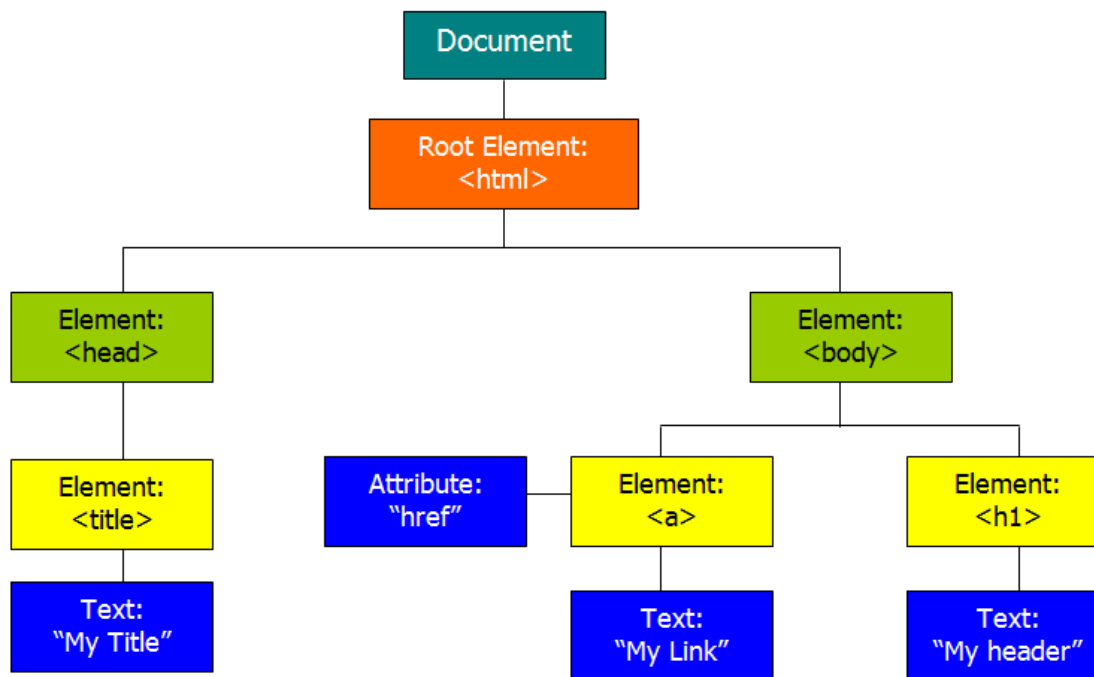
Phương thức	Tham số	Ý nghĩa
Open	url, name, options	Mở một window mới
Close		Đóng cửa sổ đang mở
moveTo	top, left	Di chuyển window đến vị trí truyền vào.
resizeTo	width, height	Thay đổi kích thước window



6 Các đối tượng trong JavaScript

❑ Mô hình DOM

- DOM là viết tắt của chữ **D**ocument **O**bject **M**odel, dịch tạm ra là mô hình các đối tượng trong tài liệu HTML
- Sơ đồ:





6 Các đối tượng trong JavaScript

❑ Mô hình DOM

- DOM Document
 - Một số thuộc tính cơ bản:

Thuộc tính	Ý nghĩa
innerHTML	Gán hoặc lấy nội dung của một element
attribute	Gán giá trị cho thuộc tính value của element
style.property	Gán hoặc lấy giá trị của các thuộc tính trong style của element Property: là tên của thuộc tính có trong style. Ví dụ: color



6 Các đối tượng trong JavaScript

❑ Mô hình DOM

- DOM Document
 - Một số phương thức cơ bản:

Phương thức	Ý nghĩa
getElementById ("ID")	Truy xuất tới element theo ID.
getElementsByName("Name")	Truy xuất danh sách element có cùng giá trị của thuộc tính Name
getElementsByTagName("TagName")	Truy xuất tới danh sách element có cùng tên tag
getElementsByClassName("ClassName")	Truy xuất danh sách element theo tên Class
setAttribute(ten,giatri)	Cài đặt thuộc tính mới cho element



Nội dung

- 5. Các đối tượng cơ bản
- 6. Các đối tượng trong JavaScript
- 7. Các sự kiện trên trang HTML**
- 8. Các xử lý cơ bản của DHTML



7 Các sự kiện trên trang HTML

- ☐ Sự kiện của window
- ☐ Sự kiện của các điều khiển trên form
- ☐ Sự kiện phím
- ☐ Sự kiện chuột



7 Các sự kiện trên trang HTML

❑ Sự kiện của window

- **Onload**: sự kiện này xảy ra khi người dùng vào website.
- Chỉ có trong thẻ body và img
- Cú pháp:

onload="functionName"

- Ví dụ: `<body onload="alert('Trang đang được tải...')">`



7 Các sự kiện trên trang HTML

❑ Sự kiện của window

- **Onunload**: sự kiện này xảy ra khi người dùng rời khỏi website.
- Chỉ có trong thẻ body
- Cú pháp:

onunload="functionName"

- Ví dụ: <body onunload="alert('Cảm ơn bạn đã ghé thăm website')">



7 Các sự kiện trên trang HTML

❑ Sự kiện của các điều khiển trên form

- **onchange**: sự kiện này xảy ra khi người dùng thay đổi giá trị nhập liệu của các thẻ input.

- Cú pháp:

onchange="functionName"

- Ví dụ: `<input type="text" onchange="alert('Giá trị đã được thay đổi')">`



7 Các sự kiện trên trang HTML

❑ Sự kiện của các điều khiển trên form

- **onsubmit**: sự kiện này xảy ra thực hiện thao tác gửi dữ liệu lên server.
- Chỉ có trong thẻ form
- Cú pháp: **onsubmit="functionName"**
- Ví dụ: `<form onsubmit="alert('Đang gửi dữ liệu lên server')">`



7 Các sự kiện trên trang HTML

❑ Sự kiện của các điều khiển trên form

- **onselect**: sự kiện này xảy ra khi người dùng chọn nội dung bất kỳ trong thẻ input hoặc textarea.
- Cú pháp: **onselect="functionName"**
- Ví dụ: `<input type="text" onselect="alert('Bạn đang chọn nội dung')">`



7 Các sự kiện trên trang HTML

❑ Sự kiện của các điều khiển trên form

- **onfocus**: sự kiện này xảy ra khi người dùng di chuyển con trỏ văn bản vào ô nhập liệu.
- Chỉ có trong các thẻ input, select, textarea.
- Cú pháp: **onfocus="functionName"**
- Ví dụ: `<input type="text" onfocus="alert('Đã di chuyển con trỏ vào input')">`



7 Các sự kiện trên trang HTML

❑ Sự kiện của các điều khiển trên form

- **onblur**: sự kiện này xảy ra khi người dùng di chuyển con trỏ văn bản ra khỏi ô nhập liệu.
- Chỉ có trong các thẻ input, select, textarea.
- Cú pháp: **onblur="functionName"**
- Ví dụ: `<input type="text" onblur="alert('Đã di chuyển con trỏ ra khỏi input')">`



7 Các sự kiện trên trang HTML

❑ Sự kiện phím

- **onkeydown**: sự kiện này xảy ra khi người dùng nhấn phím bất kỳ.
- Cú pháp: **onkeydown="functionName"**
- Ví dụ: `<input type="text" onkeydown="alert('Đang nhập phím')">`



7 Các sự kiện trên trang HTML

❑ Sự kiện phím

- **onkeyup**: sự kiện này xảy ra khi người dùng thả phím đang nhấn.
- Cú pháp: **onkeyup="functionName"**
- Ví dụ: `<input type="text" onkeyup="alert(Đã nhả phím')">`



7 Các sự kiện trên trang HTML

❑ Sự kiện phím

- **onkeypress**: sự kiện này xảy ra khi người dùng nhấn và thả phím bất kỳ.
- Cú pháp: **onkeypress="functionName"**
- Ví dụ: `<input type="text" onkeypress="alert(Đã tác động phím')">`



7 Các sự kiện trên trang HTML

❑ Sự kiện chuột

- **onclick**: sự kiện này xảy ra khi người dùng nhấp chuột vào element.
- **ondblclick**: sự kiện này xảy ra khi người dùng nhấp đúp chuột vào element
- **onmousedown**: sự kiện này xảy ra khi người dùng nhấp và giữ chuột vào element



7 Các sự kiện trên trang HTML

❑ Sự kiện chuột

- **onmouseup**: sự kiện này xảy ra khi người dùng thả chuột
- **onmousemove**: sự kiện này xảy ra khi người dùng rê chuột trên element
- **onmouseover**: sự kiện này xảy ra khi người dùng rê chuột vào element
- **onmouseout**: sự kiện này xảy ra khi người dùng rê chuột ra khỏi element



Nội dung

- 5. Các đối tượng cơ bản
- 6. Các đối tượng trong JavaScript
- 7. Các sự kiện trên trang HTML
- 8. Các xử lý cơ bản của DHTML**



8 Các xử lý cơ bản của DHTML

❑ Một thuộc tính đơn

❑ Một class



8 Các xử lý cơ bản của DHTML

❑ Một thuộc tính đơn

- Dùng javascript để thay đổi một số thuộc tính css của một element cụ thể
- Cú pháp: **this.style.<tên thuộc tính> = <giá trị>**

❑ Một class

- Dùng javascript để thay đổi class css của một element.
- Cú pháp: **this.className = 'Tên lớp'**

