INTERNET VÀ CÔNG NGHỆ WEB

Phần 3:JavaScritp

Mục Tiêu Của Bài Học

- Mục Tiêu
  - Cung cấp kiến thức về ngôn ngữ lập trình Javascript
  - Cung cấp kiến thức về các đối tượng trong javascript
  - Sử dụng Javascript trên Website
- Sau khi hoàn thành bài học, sinh viên có thể
  - Nắm bắt ngôn ngữ lập trình Javascript
  - Sử dụng được Javascript

Nội Dung Bài Học

Bài 1 : Ngôn ngữ lập trình JavaScript

I. **Giới thiệu tổng quan về JavaScript**

II. **Một số hộp thoại cơ bản – Popup Boxes**

III. **Sử dụng JavaScript trong trang HTML**

IV. **Ngôn ngữ lập trình JavaScript**

Bài 2: Các đối tượng trong JavaScript

I. **Các đối tượng cơ bản**

II. **Các đối tượng trong JavaScript**

III. **Các sự kiện trên trang HTML**

Bài 1 : Ngôn ngữ lập trình JavaScript

Bài 1 : Ngôn ngữ lập trình JavaScript

I. **Giới thiệu tổng quan về JavaScript**

II. **Một số hộp thoại cơ bản – Popup Boxes**

III. **Sử dụng JavaScript trong trang HTML**

IV. **Ngôn ngữ lập trình JavaScript**

1. Giới thiệu tổng quan 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 tổng quan 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
  - Lập trình theo cấu trúc (Structured progarming)
  - Có phân biệt chữ HOA và thường

1. Giới thiệu tổng quan 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, …)

2. Một số hộp thoại cơ bản

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

Hộp thông báo – Alert box

- Xuất câu thông báo trong một cửa sổ

alert(*"Câu thông báo"*)

Ví dụ:

Một số hộp thoại cơ bản

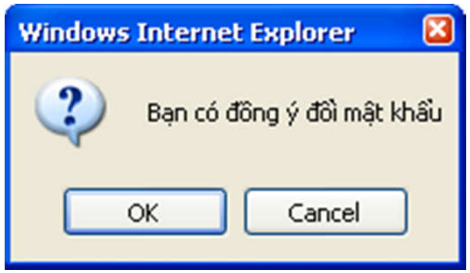
- Hộp thông báo – Alert box
- Hộp xác nhận – Confirm box
- Hộp nhận giá trị - Prompt box

Hộp xác nhận – Confirm box

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

<biến> = confirm(*"Câu hỏi"*)

Ví dụ:

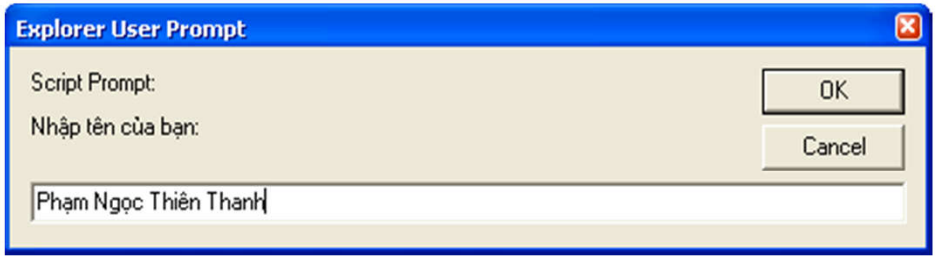


Hộp nhận giá trị – Prompt box

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

<biến> = prompt(*"Câu hướng dẫn", "giá trị mặc định"*)

Ví dụ:



3. Sử dụng JavaScript trong trang HTML

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

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">
              //Đọan JavaScript
        </script>
  </head>
 <body>
        Nội dung trang web
 </body>
</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">
                //Đọan JavaScript
        </script>
        Nội dung trang web
        <script language="javascript" type="text/javascript">
                //Đọan JavaScript
        </script>
        …
</body></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>
```

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

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 đọan văn bản */
        document.write("<p>Đây là đọan văn thứ nhất </p>");
        document.write("<p>Đây là đọan văn thứ hai </p>");
 </script>
```

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

Biến – Variable

- Khai báo:

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

Biến – Variable

- Khai báo:
  - 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

Biến – Variable

- Giá trị của biến:
  - 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

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
```

Biến – Variable

- Kiểm tra kiểu dữ liệu: dùng toán tử **typeof**

typeof *<giá trị>*

hoặc 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"
```

Các toán tử - operators

- Toán tử số học – Arithmetic operators

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

Các toán tử - operators

- Toán tử gán – Assignment operators

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

Các toán tử - operators

- Toán tử so sánh – Comparision operators

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	x = 3 y = "3" x == y ➔ true
!=	Trả về **true** nếu 2 toán hạng không bằng nhau. Cách thức so sánh cũng giống toán tử ==	x != y ➔ false
===	Trả về **true** nếu 2 toán hạng cùng kiểu và cùng giá trị.	x === y ➔ false
!==	Trả về **true** nếu 2 toán hạng không bằng nhau. Cách thức so sánh cũng giống toán tử ===	x !== y ➔ true
<, >, <=, >=		

Các toán tử - operators

- Toán tử luận lý – Logical operators

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

Các toán tử - operators

- Toán tử điều kiện – Conditonal operator

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

Ví dụ:

tinhtrang = (tuoi>=18) ? “trưởng thành” : “vị thành niên”

Một số hàm cơ bản trong JavaScript

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

Ví dụ:

```
str="if (3>2) document.write('đúng'); else document.write('sai');"
var ktra = eval(str);                          //➔ "đúng"
document.write("<br>" + eval("2+2"));          //➔ 4
```

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**

Ví dụ:

```
document.write(isNaN(5-2)+ "<br />");        ➔ false
document.write(isNaN(0)+ "<br />");          ➔ false
document.write(isNaN("Hello")+ "<br />");    ➔ true
```

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**

Ví dụ:

```
document.write(Number("227") + "<br>");          ➔ 227
document.write(Number(true)  + "<br>"); ➔ 1
document.write(Number(false) + "<br>"); ➔ 0
document.write(Number("227 233"));               ➔ NaN
```

Một số hàm cơ bản trong JavaScript

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

Ví dụ:

```
document.write(parseInt("10") + "<br />");          ➔ 10
document.write(parseInt("10.33") + "<br />");       ➔ 10
document.write(parseInt("34 45 66") + "<br />");    ➔ 34
document.write(parseInt(" 60 ") + "<br />");        ➔ 60
document.write(parseInt("40 tuổi") + "<br />");     ➔ 40
document.write(parseInt("Cô ấy 40") + "<br />");    ➔ NaN
```

Một số hàm cơ bản trong JavaScript

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

Ví dụ:

```
document.write(parseFloat("10.33") + "<br />");          ➔ 10.33
document.write(parseFloat("34 45 66") + "<br />");       ➔ 34
document.write(parseFloat(" 60 ") + "<br />");           ➔ 60
document.write(parseFloat("40 tuổi") + "<br />");        ➔ 40
document.write(parseFloat("Cô ấy 40") + "<br />");       ➔ NaN
```

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;       }
```

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))">
```

Cấu trúc điều khiển

- if … else

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

Cấu trúc điều khiển

- if … else

Hoặc

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

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 ; */        ]
}
```

Cấu trúc điều khiển

- Vòng lặp for

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

Cấu trúc điều khiển

- Vòng lặp for

Ví dụ: in bảng cửu chương 2

```
var i;
for ( i=1; i<=10; i++)
{
        document.write( "2 * " + i + " = " + 2*i + "<br>")
}
```

Ví dụ: in ngược bảng cửu chương 2

```
for (var i=1, j=10; i<=10; i++, j--)
{
        document.write( "2 * " + j + " = " + 2*j + "<br>")
}
```

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>)
```

Cấu trúc điều khiển

- Vòng lặp while và do … while

Ví dụ:

```
var i = 0;
var n = 0;
while (i < 5)
{   i++ ;
      if (i == 3)
           continue;
   n += i;
   document.write("i : " + i + " ; " + "n : " + n + "<br>");
}
```

```
i : 1 ; n : 1
i : 2 ; n : 3
i : 4 ; n : 7
i : 5 ; n : 12
```

Thảo luận

Bài 2: Các đối tượng trong JavaScript

Bài 2: Các đối tượng trong JavaScript

I. **Các đối tượng cơ bản**

II. **Các đối tượng trong JavaScript**

III. **Các sự kiện trên trang HTML**

1. Các đối tượng cơ bản

- String
- Date
- Math

1. Các đối tượng cơ bản

- JavaScript là ngôn ngữ lập trình dựa trên đối tượng (Object-based language)
- Các đối tượng trong JavaScript (Math, String, …) giúp người lập trình xử lý cắt chuỗi, sử dụng các hàm toán học, …
- JavaScript sẽ dựa vào giá trị của biến để xác định biến đó thuộc đối tượng nào

String

- Xử lý chuỗi văn bản

var <tên biến> **= new String;**

hoặc **var** <tên biến> **= new String(***"chuỗi khởi tạo"***);**

Ví dụ:

var chuoi = new **String**("Lập trình web cơ bản");

- Thuộc tính:
  - length : trả về tổng số ký tự của chuỗi

String

- Phương thức:

  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

String

- Phương thức:

Ví dụ: tìm chữ "Tâm" trong chuỗi "Chữ tâm kia mới bằng ba chữ tài"

```
var chuoi = new String("Chữ tâm kia mới bằng ba chữ tài");
n = chuoi.search(/Tâm/i)
alert(n);
```

➔ 4

String

- Phương thức:
  - 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

Ví dụ: thay thế toàn bộ chữ “Mẹ” thành “Má” trong 2 câu đầu của bài hát “Mẹ dấu yêu”

```
var chuoi =new String("Mẹ là làn gió mát, đưa con giấc ngủ ngoan; Mẹ là dòng suối trong, cho con luôn tìm về");
str = chuoi.replace(/mẹ/gi,"Má")
```

Date

- Dùng để xử lý dữ liệu kiểu thời gian

```
        var <tên biến> = new Date();
hoặc    var <tên biến> = new Date(năm, tháng, ngày);
```

  - <tháng> được tính từ 0 ; tháng 1

Date

- Phương thức
  - getDay(): trả về thứ tự ngày trong tuần (0 → 6), 0 bắt đầu là Chủ nhật
  - getMonth(): trả về số tháng trong năm (0 → 11), 0 bắt đầu tháng 1

Ví dụ: in thứ tự ngày trong tuần của ngày sinh 25/05/2008

```
var ngaysinh = new Date(2008,4,25)
thu = ngaysinh.getDay()
alert(thu);                    → 0
```

Math

- Các xử lý liên quan đến toán học
- Không cần khai báo và khởi tạo
- Thuộc tính:

Tên	Ý nghĩa
PI	Trả về hằng số pi
SQRT1_2	Trả về căn bậc 2 của ½
SQRT2	Trả về căn bậc 2 của 2

Math

- Phương thức

Tên	Ý nghĩa
abs(x)	Trả về giá trị tuyệt đối của x
ceil(x)	Trả về số nguyên gần nhất lớn hơn hoặc bằng x (1)
floor(x)	Trả về số nguyên gần nhất nhỏ hơn hoặc bằng x (2)
max(a, b)	Trả về số lớn hơn giữa a và b
min(a, b)	Trả về số nhỏ hơn giữa a và b
pow(x, y)	Trả về x lũy thừa y (x^y)
random()	Trả về con số ngẫu nhiên lớn hơn 0 và nhỏ hơn 1 (như 0.62535) (3)
round(x)	Làm tròn đến số nguyên gần nhất của x dựa theo phần thập phân là 0.5 (4)
sqrt(x)	Trả về căn bậc 2 của x

Math

- Phương thức

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

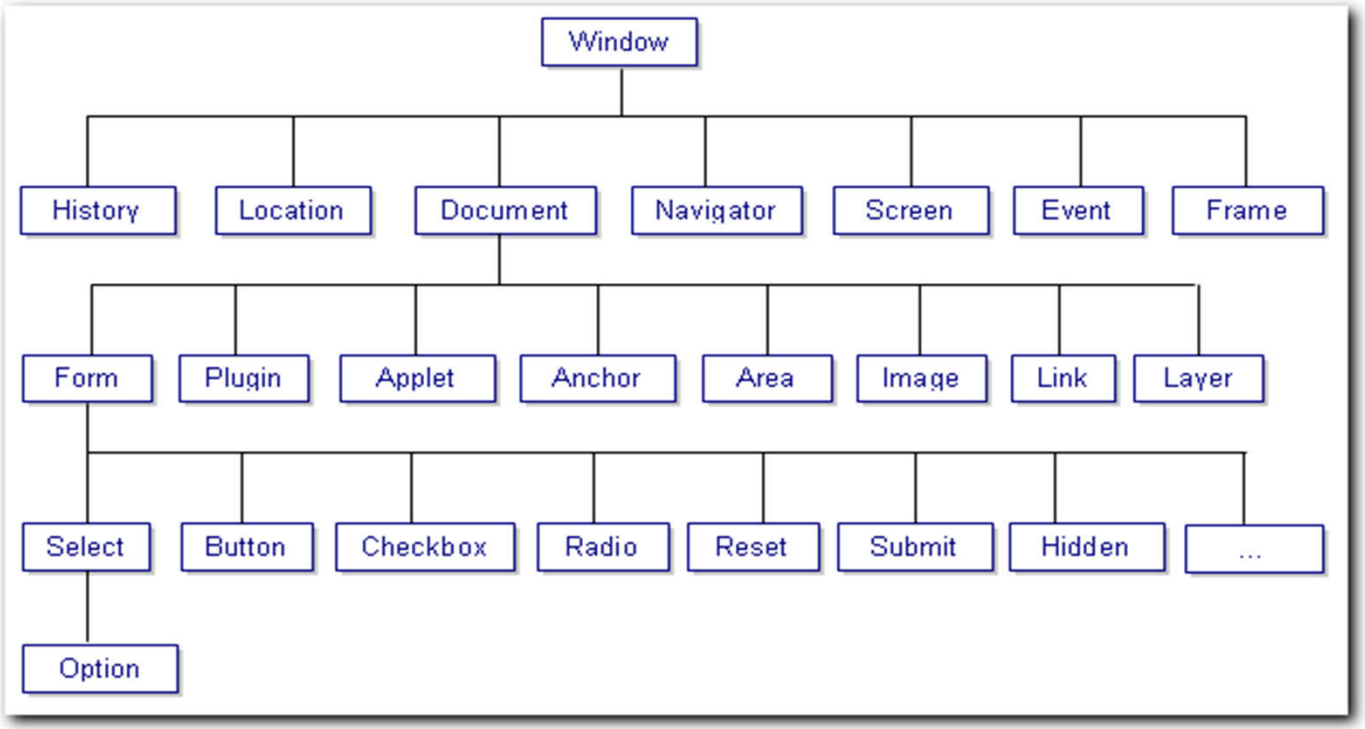
Math.round(3.5) ➔ 4

2. Các đối tượng trong JavaScript

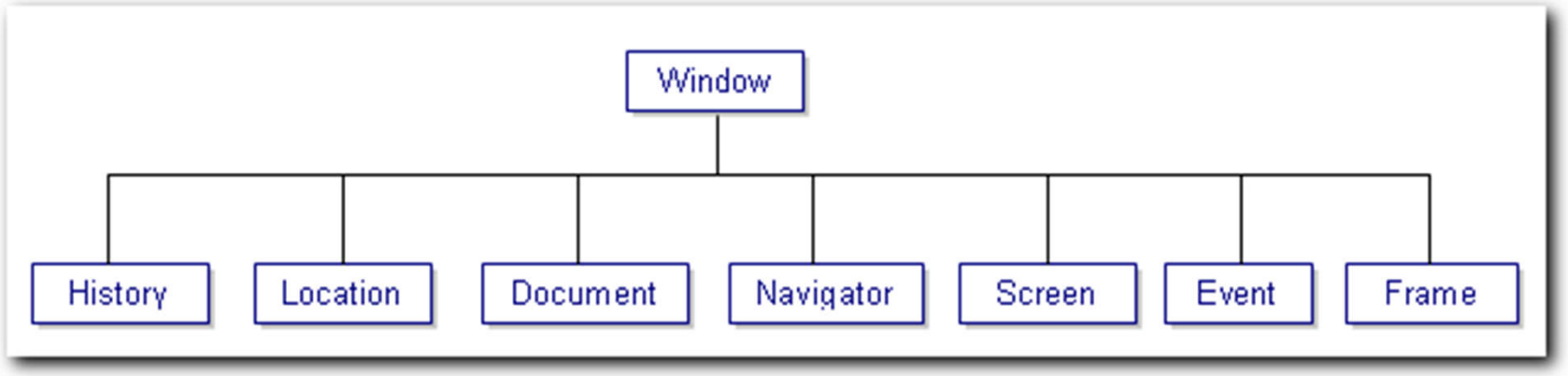
- Mô hình BOM - Browser Object Model
- Mô hình DOM - Document Object Model
- Tham chiếu đến một đối tượng trong DOM

2. Các đối tượng trong JavaScript

- Sắp xếp theo hệ thống phân cấp (hierarchy) và bắt đầu là đối tượng window



Mô hình BOM - Browser Object Model



Mô hình BOM - Browser Object Model

- **window**: đại diện cho cửa sổ trình duyệt
  - Thuộc tính status, history, location
  - Phương thức close, open, setInterval, setTimeout, …
  - Sự kiện:

Tên	Ý nghĩa
Onblur	Xảy ra khi window mất focus **
Onerror	Xảy ra khi có lỗi
Onfocus	Xảy ra khi window nhận focus
Onload	Xảy ra khi mở đã xong trang web
Onresize	Xảy ra khi window bị thay đổi kích thước

Quay trở lại

Mô hình BOM - Browser Object Model

- **navigator**: cung cấp thông tin về trình duyệt và hệ thống tại máy client
  - Thuộc tính:

Tên	Ý nghĩa
appName	Tên trình duyệt
appVersion	Trả về phiên bản của trình duyệt, tuy nhiên đó không phải là con số chính xác. Chẳng hạn IE từ 4 → 7 sẽ thể hiện là 4.0, Netscape 6.0 thể hiện 5.0, ...
platform	Hệ điều hành hiện tại mà cửa sổ trình duyệt đang mở

Mô hình BOM - Browser Object Model

- **location**: chứa thông tin về URL hiện hành, thường sử dụng để di chuyển đến một trang web khác
  - Thuộc tính: protocol, hostname, port, …
  - Phương thức:

Tên	Ý nghĩa
reload()	Tải lại trang hiện hành, tương tự nút Refresh của cửa sổ trình duyệt.
replace("url")	Thay thế trang hiện hành bằng trang mới có địa chỉ là <url>

Mô hình BOM - Browser Object Model

- **location**: chứa thông tin về URL hiện hành, thường sử dụng để di chuyển đến một trang web khác

Ví dụ: khi nhấn nút Đọc Báo, thế trang hiện hành bằng trang web của Báo Thanh Niên

```
<form>
<input type="button" value="Đọc Báo"
onClick="location.replace('http://www.thanhnien.com.vn')">
</form>
```

Mô hình BOM - Browser Object Model

- **event:** được hỗ trợ từ IE 5.0 và Netscape 6.0 trở lên, dùng để lưu vết các sự kiện xảy ra trên trang web như nhấn chuột, di chuyển chuột, …

Mô hình BOM - Browser Object Model

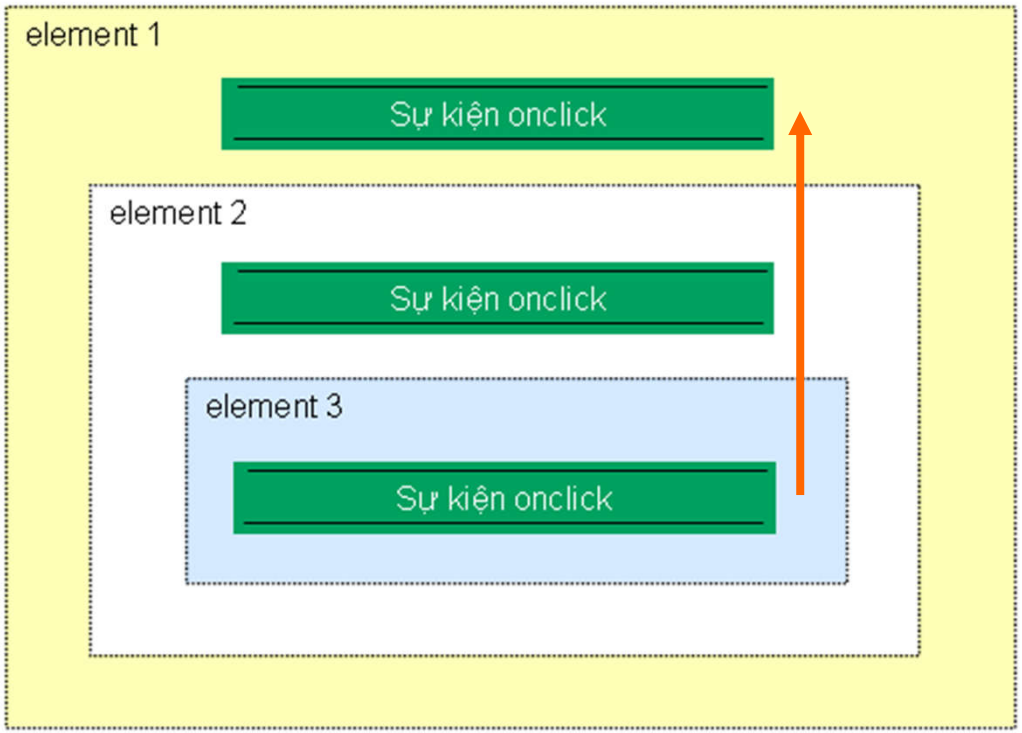
- **event:**
  - Thuộc tính:

Áp dụng cho IE 5.0+

Tên	Ý nghĩa
altKey, ctrlKey, shiftKey	Trả về true/false, chỉ ra các phím Alt / Ctrl / Shift có được nhấn không
keyCode	Trả về giá trị Unicode của phím được nhấn
button	Trả về số nguyên, chỉ ra phím chuột nào được nhấn (1: trái; 2: phải; 4: giữa)
cancelBubble	Đặt **true** nếu muốn ngăn các sự kiện cha xảy ra. Ví dụ: khi click chuột vào sự kiện của một điều khiển trong form, bạn không muốn các xử lý trong sự kiện Onclick của form và trang xảy ra thì phải đặt giá trị là **true**
clientX, clientY	Trả về tọa độ x, y của chuột (1)
type	Trả về tên của sự kiện (như click, mouseup, ...)

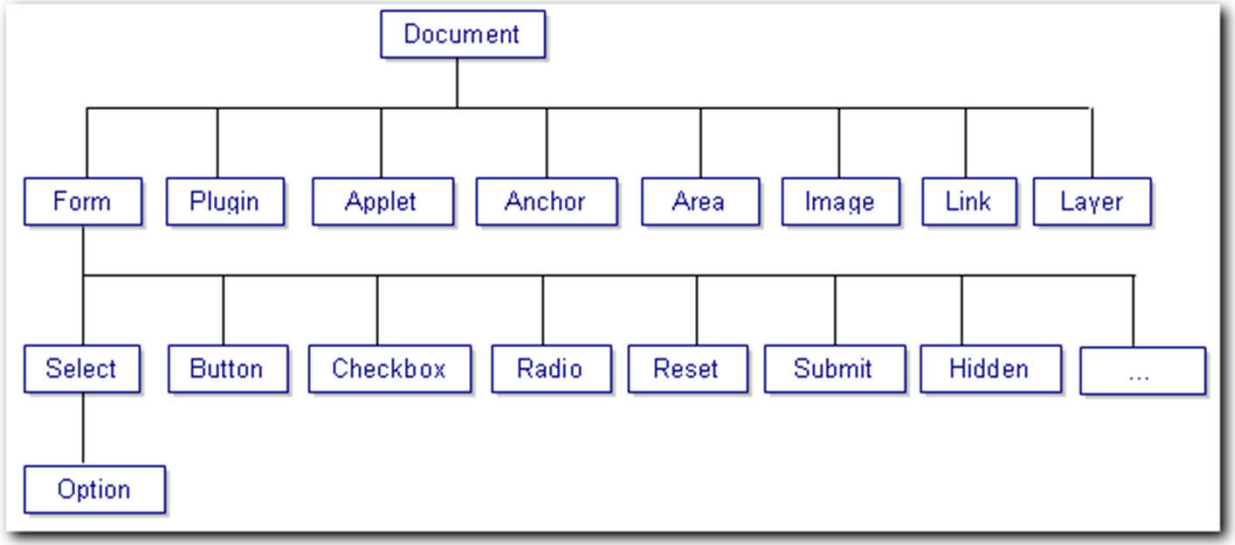
Mô hình BOM - Browser Object Model

- **event:**
  - Thứ tự xảy ra trên cùng một kiểu sự kiện của các element trong một trang web



Mô hình DOM – Document Object Model

- Phản ánh cấu trúc của một tài liệu HTML
- Được phép thêm, xóa hoặc cập nhật các đối tượng trong DOM



Mô hình DOM – Document Object Model

- **document**: đại diện cho toàn bộ trang HTML
  - Thuộc tính tập hợp: thường dùng để xác định một phần tử hoặc duyệt các phần tử trong một tập hợp
  - Thuộc tính: title, URL

Mô hình DOM – Document Object Model

- **document**: đại diện cho toàn bộ trang HTML

Ví dụ: duyệt và in tên các tag <a> ra màn hình

```
<body><a name="first"> anchor đầu tiên</a><br />
<a name="second">anchor thứ hai</a><br />
<a name="third"> anchor thứ ba</a><br />
<script type="text/javascript">
for (var i=0; i<=document.anchors.length -1 ; i++)
      document.write(document.anchors[i].name + "<br>")
</script> </body>
```

anchor đầu tiên
anchor thứ hai
anchor thứ ba
first
second
third

Mô hình DOM – Document Object Model

- **document**: đại diện cho toàn bộ trang HTML
  - Phương thức:

Tên	Ý nghĩa
write(<danh sách biểu thức>)	Ghi các biểu thức chuỗi hoặc code JavaScript vào tài liệu HTML
getElementById("ID")	Trả về một tham chiếu đến đối tượng thông qua thuộc tính id. Tham số **ID** chính là giá trị của thuộc tính id trong một tag (1)
getElementsByName("Name")	Trả về tập hợp các đối tượng có cùng giá trị thuộc tính name (2)
getElementsByTagName("TagName")	Trả về tập hợp các đối tượng có cùng tên tag (3)

Minh họa

Minh họa

Mô hình DOM – Document Object Model

- **document**: đại diện cho toàn bộ trang HTML
  - Sự kiện:

Tên	Ý nghĩa
Sự kiện chuột	
onclick	Xảy ra khi nhấn chuột
ondbclick	Xảy ra khi double-click
onmousedown	Xảy ra khi nhấn phím chuột
onmouseup	Xảy ra khi thả phím chuột
onmousemove	Xảy ra khi di chuyển chuột đến đối tượng
onmouseover	Xảy ra khi rê chuột lên trên đối tượng
onmouseout	Xảy ra khi rê chuột ra khỏi đối tượng
Sự kiện phím	
onkeydown	Khi nhấn phím bất kỳ
onkeypress	Khi nhấn phím có hiển thị dữ liệu (như phím số, ký tự hoặc Esc)
onkeyup	Khi thả phím trạng thái (Shift, Enter, Ctrl, ...)

Minh họa

Minh họa

Quay trở lại

Tham chiếu đến một đối tượng trong DOM

- Tham chiếu một đối tượng của document

document.<tên t.tính tập hợp>**[**"tên đ.tượng"**|**<chỉ số>**].<t.tính>**

hoặc

document.<tên t.tính tập hợp>**.**<tên đ.tượng>**.<thuộc tính>**

Ví dụ:

document.images['hinh1'].src

document.images[0].name

document.images.hinh1.src

Tham chiếu đến một đối tượng trong DOM

- Tham chiếu một đối tượng của form

document.forms["tên form"**|**<chỉ số>**].**<đ.tượng>**.<thuộc tính>**

Ví dụ:

```
document.forms['nhap'].ks.checked
```

Tham chiếu đến một đối tượng trong DOM

- Điều khiển Drop-down List:
  - Truy xuất một mục bất kỳ

document.forms["tên form"|<chỉ số>**].**<đ.tượng>**.options[**<chỉ số>**].<t.tính>**

hoặc

document.forms.<tên form>.<đ.tượng>**.options[**<chỉ số**>].<thuộc tính>**

Ví dụ:

```
var nd = document.forms['nhap'].thuc_an.options[1].innerText
```

hoặc

```
var nd = document.forms.nhap.thuc_an.options[1].innerText
```

Tham chiếu đến một đối tượng trong DOM

- Điều khiển Drop-down List:
  - Duyệt và kiểm tra từng mục

```
<biến> = document.getElementById('id của tag <select>');
        for (i=0; i< <biến>.length; i++ )
        {           if (<biến>.options[i].selected)      //mục thứ i được chọn
                    {
                                /* Khối lệnh xử lý */
                    }
        }
```

Tham chiếu đến một đối tượng trong DOM

- Điều khiển Drop-down List:
  - Thêm một mục

document.forms['tên form'**].**<đ.tượng>**.options[**<chỉ số>**]**
= new Option(<nội dung>, <giá trị>**);**

Tham chiếu đến một đối tượng trong DOM

- Điều khiển Drop-down List:
  - Xóa một mục

document.forms['tên form'**].**<đối tượng>**.options[**<chỉ số>**]= null;**

hoặc

<biến> **= document.getElementById(**'id của tag <select>'**);**

<biến>**.remove(**<chỉ số>**);**

Tham chiếu đến một đối tượng trong DOM

- Điều khiển Drop-down List:
  - Xóa tất cả các mục

document.forms['tên form'**].**<đối tượng>**.options.length= 0;**

3. Các sự kiện trên trang HTML

- Sự kiện của window – Window Events
- Sự kiện của các điều khiển trên form
- Sự kiện phím – Keyboard Events
- Sự kiện chuột – Mouse Events

3. Các sự kiện trên trang HTML

- Sự kiện của window – Window Events

  (xem lại mục đối tượng Window trong Mô hình BOM)

3. Các sự kiện trên trang HTML

- Sự kiện của các điều khiển trên form
  - onchange: khi thay đổi nội dung của điều khiển
  - onfocus: khi điều khiển nhận được focus
  - onblur: khi điều khiển mất đi focus

3. Các sự kiện trên trang HTML

- Sự kiện phím – Keyboard Events
  - Các sự kiện: onkeydown, onkeypress, onkeyup
  - Thường áp dụng cho đối tượng document, form và các điều khiển trên form

Ví dụ:

Chỉ cho phép nhập số, nếu nhập ký tự thì vô hiệu hóa phím nhấn

3. Các sự kiện trên trang HTML

- Sự kiện chuột – Mouse Events

  (xem lại mục đối tượng Document trong mô hình DOM)

Thảo luận