



Chương 4. Javascript & DOM

ThS. Trần Anh Dũng
Email: dungta@uit.edu.vn

Nội dung



Giới thiệu về Javascript

Nhúng Javascript vào trang web

Kiểu dữ liệu & Cú pháp Javascript

Xử lý sự kiện

DOM HTML với Javascript

Giới thiệu

- ❖ HTML nguyên thủy không có khả năng xử lý các tương tác của người dùng.
- ❖ Mọi tương tác với trang web cần phải xử lý ở server → chi phí về thời gian, thông lượng quá mức cần thiết.
- ❖ Các trình duyệt có khả năng hỗ trợ **thông dịch ngôn ngữ kịch bản**.

Giới thiệu

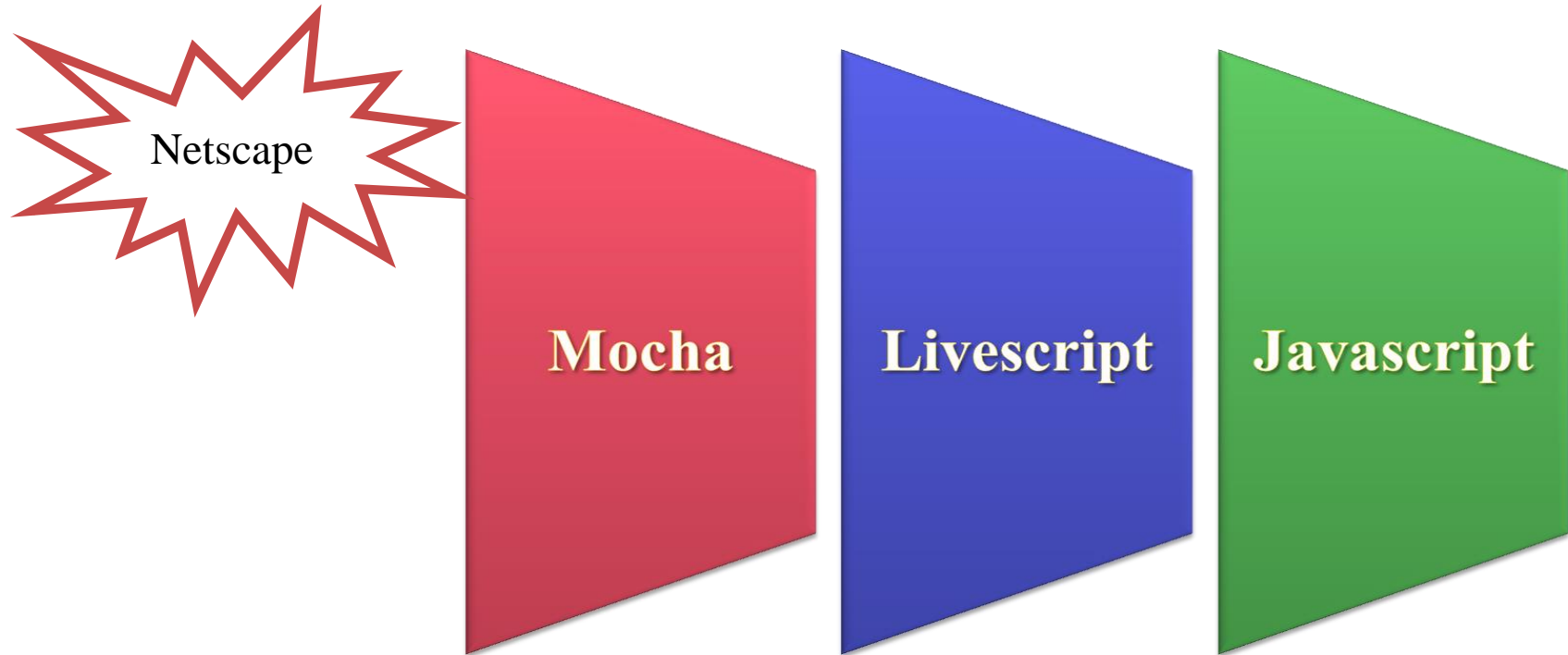
❖ Client-Side Script:

- Script được thực thi tại **Client-Side (Trình duyệt)**: Thực hiện các tương tác với người dùng: tạo menu chuyển động, kiểm tra dữ liệu nhập,...

❖ Server-Side Script:

- Script được xử lý tại **Server-Side**, nhằm tạo các trang web có khả năng phát sinh nội dung động. Một số xử lý chính: kết nối CSDL, truy cập hệ thống file trên server, phát sinh nội dung html trả về phía người dùng,...

Lịch sử

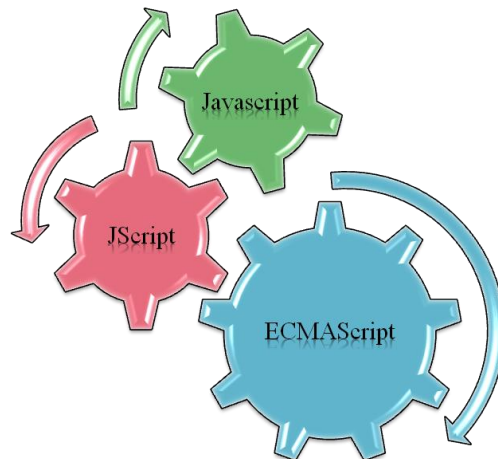


- Javascript được phát triển bởi Brendan Eich tại hãng truyền thông Netscape với tên đầu tiên là Mocha, sau đó đổi tên thành Livescript, và cuối cùng là Javascript.
- Cùng thời điểm Netscape bắt đầu sử dụng công nghệ Java trên trình duyệt Netscape, Livescript được đổi tên thành Javascript.

Lịch sử

- ❖ Javascript được giới thiệu và triển khai trên trình duyệt Netscape 2.0b3 vào tháng 12/1995
- ❖ ECMAScript là phiên bản chuẩn hóa của Javascript.
- ❖ Phiên bản mới nhất của Javascript?

https://www.w3schools.com/js/js_versions.asp



Giới thiệu về Javascript

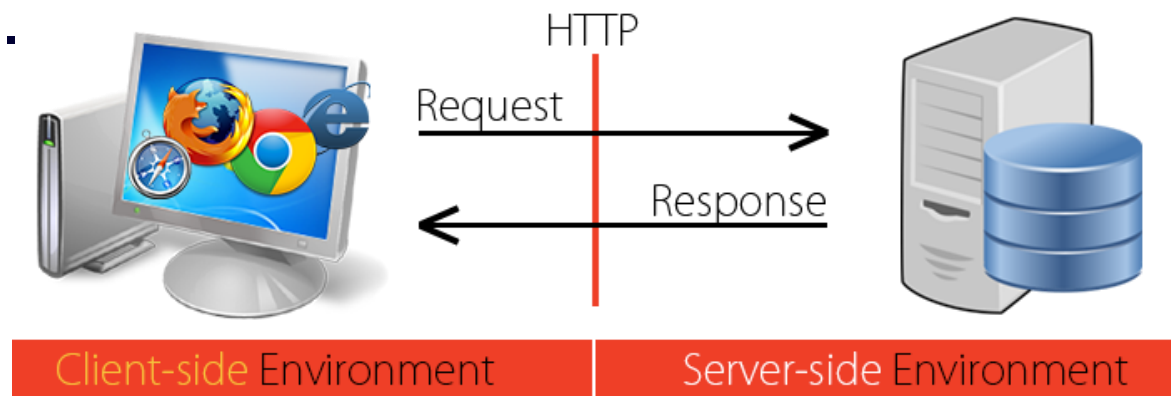
- ❖ Là ngôn ngữ **Client-side script** hoạt động trên trình duyệt của người dùng (client).
- ❖ Chia sẻ xử lý trong ứng dụng web, giảm các xử lý không cần thiết trên server.
- ❖ Giúp tạo các hiệu ứng, tương tác cho trang web.
- ❖ **Khả năng thường được dùng của JavaScript**
 - Nhận và kiểm tra tính hợp lệ của dữ liệu
 - Tính toán dữ liệu tạm thời
 - Tạo các hiệu ứng hoạt ảnh, xử lý các sự kiện
 - Bị quy định không được phép truy cập vào các tài nguyên mức hệ thống,...

Giới thiệu về Javascript

- ❖ JavaScript là ngôn ngữ dưới dạng script có thể gắn với các file HTML và dùng **thông dịch (interpreter)**
- ❖ JavaScript không phải là ngôn ngữ hướng đối tượng như C++, Java,...
- ❖ JavaScript là ngôn ngữ dựa trên đối tượng: math, document, windows,...
- ❖ Thiết kế độc lập với hệ điều hành.

Giới thiệu về Javascript

- ❖ Khi trình duyệt (Client browser) truy cập trang web có chứa các đoạn mã xử lý tại server-side. Server (run-time engine) sẽ thực hiện các lệnh Server-side scripts và trả về nội dung HTML cho trình duyệt.
- ❖ Nội dung html trả về chủ yếu bao gồm: mã html, client-script.



Nhúng Javascript vào trang web

- ❖ Định nghĩa Javascript trực tiếp trong trang HTML:

```
<script type="text/javascript">  
    <!--  
    // Lệnh Javascript  
    -->  
</script>
```

- ❖ Nhúng sử dụng script cài đặt từ 1 file .js khác:

```
<script src="xxx.js"></script>
```

Nhúng Javascript vào trang web

```
<html>
  <head>
    <script type="text/javascript">
      some statements
    </script>
  </head>
  <body>
    <script type="text/javascript">
      some statements
    </script>
    <script src="Tên_file_script.js">method()</script>
    <script type="text/javascript">
      // gọi thực hiện các phương thức được định nghĩa
      // trong "Tên_file_script.js"
    </script>
  </body>
</html>
```

Nhúng Javascript vào trang web

- ❖ Đặt giữa tag `<head>` và `</head>`: Script sẽ thực thi ngay khi trang web được mở.
- ❖ Đặt giữa tag `<body>` và `</body>`: Script trong phần body được thực thi khi trang web đang mở (sau khi thực thi các đoạn script có trong phần `<head>`).
- ❖ Số lượng đoạn client-script chèn vào trang không hạn chế.

Nhúng Javascript vào trang web

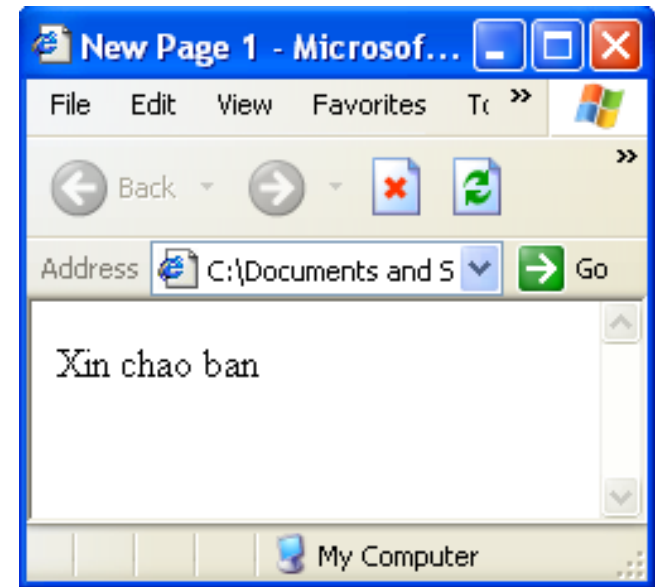
❖ Ví dụ:

```
<html>

<head>
<title>New Page 1</title>
</head>

<body>
<script language="JavaScript">
    document.write("Xin chao ban");
</script>
</body>

</html>
```



Nhúng Javascript vào trang web

❖ Ví dụ:

```
<html>
```

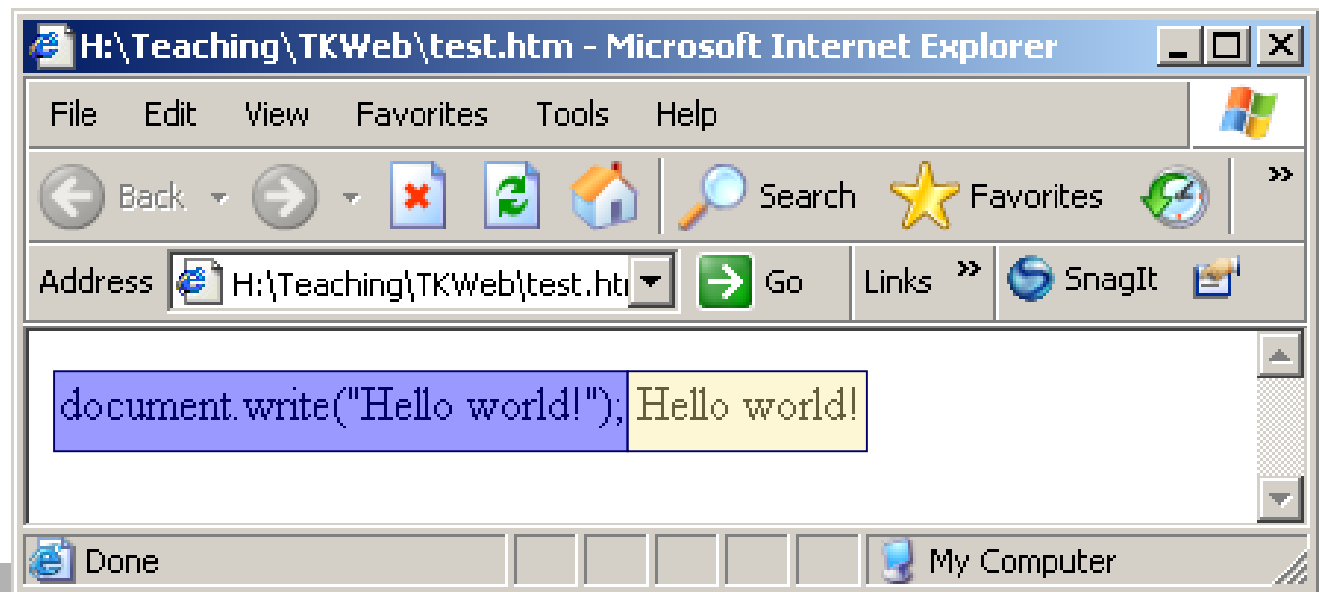
```
<body>
```

```
document.write("Hello world!");
```

```
<script type="text/javascript">  
    document.write("Hello world!");  
</script>
```

```
</body>
```

```
</html>
```



Biến trong Javascript

❖ Cách đặt tên biến:

- Bắt đầu bằng một chữ cái hoặc dấu _
- A..Z, a..z, 0..9, _ : **Phân biệt HOA, Thường**

❖ Khai báo biến:

- Sử dụng từ khóa **var**
- Ví dụ: **var count=10, amount;**
- Không cần khai báo biến trước khi sử dụng, biến thật sự tồn tại khi bắt đầu sử dụng lần đầu tiên

Kiểu dữ liệu trong Javascript

Kiểu dữ liệu	Ví dụ	Mô tả
Object	<code>var listBooks = new Array(10) ;</code>	Trước khi sử dụng, phải cấp phát bằng từ khóa new
String	<code>"The cow jumped over the moon."</code> <code>"40"</code>	Chứa được chuỗi unicode Chuỗi rỗng <code>""</code>
Number	<code>0.066218</code> <code>12</code>	Theo chuẩn IEEE 754
boolean	<code>true/false</code>	
undefined	<code>var myVariable ;</code>	<code>myVariable = undefined</code>
null	<code>connection.Close();</code>	<code>connection = null</code>

1 Biến trong javascript có thể lưu bất kỳ kiểu dữ liệu nào.

Đổi kiểu dữ liệu

- ❖ Biến tự đổi kiểu dữ liệu khi giá trị mà nó lưu trữ thay đổi. Ví dụ:
 - `var x = 10;` // x kiểu Number
 - `x = "hello world !";` // x kiểu String
- ❖ Có thể cộng 2 biến khác kiểu dữ liệu
- ❖ Ví dụ:
 - `var x;`
 - `x = "12" + 34.5;`
- ❖ Hàm `parseInt(...)`, `parseFloat(...)`: Đổi kiểu dữ liệu từ chuỗi số sang số.

Special Characters

<code>\b</code>	backspace
<code>\f</code>	form feed
<code>\n</code>	new line (xuống dòng và đưa con trỏ về đầu dòng đó)
<code>\r</code>	Carriage return (đưa con trỏ về đầu dòng hiện tại)
<code>\t</code>	Tab

Toán tử số học

Toán tử	Mô tả	Ví dụ
+	Phép cộng	$A = 5 + 8$
-	Phép trừ	$A = 8 - 5$
*, /	Phép nhân, chia	$A = 20 / 5$
%	Phép chia lấy số dư	$10 \% 3 = 1$
++	Tăng lên một đơn vị. Toán tử này nhận một toán hạng. Giá trị của toán hạng sẽ tăng lên 1 đơn vị. Giá trị được trả về sẽ tùy thuộc vào toán tử ++ nằm sau hay nằm trước toán hạng.	++x sẽ trả về giá trị của x sau khi tăng. x++ sẽ trả về giá trị của x trước khi tăng.
--	Giảm một đơn vị. Toán tử này nhận một toán hạng. Giá trị được trả về tùy thuộc vào toán tử -- nằm trước hay nằm toán hạng.	--x sẽ trả về giá trị của x sau khi giảm. x-- sẽ trả về giá trị của x trước khi giảm. .
-	Lấy số đối. Nó sẽ trả về số đối của toán hạng	Nếu a là 5, thì $-a = -5$.

Toán tử số học

Toán tử	Mô tả	Ví dụ
<code>==</code>	Bằng. Trả về giá trị true nếu các toán hạng bằng nhau.	<code>a == b</code>
<code>!=</code>	Không bằng. Trả về giá trị true nếu các toán hạng không bằng nhau.	<code>Var2 != 5</code>
<code>></code>	Lớn hơn. Trả về giá trị true nếu toán hạng trái lớn hơn toán hạng phải.	<code>Var1 > var2</code>
<code>>=</code>	Lớn hơn hoặc bằng. Trả về giá trị true nếu toán hạng trái lớn hơn hoặc bằng toán hạng phải.	<code>Var1 >= 5</code> <code>Var1 >= var2</code>
<code><</code>	Nhỏ hơn. Trả về giá trị true nếu toán hạng trái nhỏ hơn toán hạng phải.	<code>Var2 < var1</code>
<code><=</code>	Nhỏ hơn hoặc bằng. Trả về giá trị true nếu toán hạng trái nhỏ hơn hoặc bằng toán hạng phải.	<code>Var2 <= 4</code> <code>Var2 <= var1</code>

Toán tử logic

Toán tử	Giá trị	Mô tả
And (&&)	expr1 && expr2	Trả về giá trị của expr1 nếu nó là true. Nếu không thì nó trả về giá trị của expr2.
Or ()	expr1 expr2	Trả về giá trị của expr1 nếu nó là true. Nếu không thì nó sẽ trả về giá trị của expr2.
Not (!)	!expr	Trả về giá trị false nếu biểu thức đúng và trả về giá trị true nếu biểu thức sai.

Toán tử khác

❖ Toán tử điều kiện

- `(condition) ? trueVal : falseVal`

❖ `typeof`

- Toán tử `typeof` trả về chuỗi cho biết tên kiểu dữ liệu của toán hạng

Kiểu dữ liệu mảng

- ❖ `arrayObjectName = new Array([element0, element1, ..., elementN])`
- ❖ `arrayObjectName = new Array(3)`
- ❖ `MyArray = new Array(5,5);`
 - `MyArray[0, 0]`

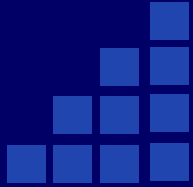
```
var cars = new Array("Saab", "Volvo", "BMW");
```

```
cars[0] = "Opel";
```

Phương thức mảng

Phương thức	Mô tả
Join	Kết hợp các phần tử của mảng thành một chuỗi
Pop	Trả về phần tử cuối cùng của mảng, sau khi xoá nó từ mảng.
Push	Thêm một hoặc nhiều phần tử vào cuối mảng. Trả lại phần tử cuối cùng thêm vào.
Reverse	Đảo ngược các phần tử của mảng: phần tử đầu tiên của mảng trở thành phần tử cuối cùng và phần tử cuối cùng trở thành phần tử đầu tiên.
Shift	Xoá phần tử đầu tiên của mảng và trả về phần tử đó.
Sort	Sắp xếp các phần tử của mảng.

Phương thức mảng



Phương thức	Ví dụ
array1.concat([item1[, item2[,...,...,... [, itemN]]]])	<pre>var a, b, c, d; a = new Array(1,2,3); b = "JScript"; c = new Array(42, "VBScript"); d = a.concat(b, c); //Returns the array [1, 2, 3, "JScript", 42, "VBScript"]</pre>
arrayObj.join(separator)	<pre>var a, b; a = new Array(0,1,2,3,4); b = a.join("-"); // b = "0-1-2-3-4"</pre>
arrayObj.pop()	<pre>var n = a.pop(); // n = 4</pre>
arrayObj.push([item1 [item2 [. ... [itemN]]])	<pre>a.push(5); // a = [0,1,2,3,5]</pre>
arrayObj.reverse()	<pre>a.reverse(); // a = [5,3,2,1,0]</pre>
arrayobj.sort(sortFunction)	<pre>function numerical_sorter(n1,n2) { return (n1 - n2); } var l = a.sort(numerical_sorter); // l = [0,1,2,3,5].</pre>

Phương thức mảng

- ❖ The JavaScript method `toString()` converts an array to a string of (comma separated) array values

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

<h2>JavaScript Array Methods</h2>

<h2>toString()</h2>

<p>The toString() method returns an array as a comma separated string:
</p>

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

<script>
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits.toString();
</script>

</body>
</html>
```

Phương thức mảng

❖ Sorting an Array

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

<h2>JavaScript Array Sort</h2>

<p>The sort() method sorts an array alphabetically.</p>

<button onclick="myFunction()">Try it</button>

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

<script>
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits;

function myFunction() {
  fruits.sort();
  document.getElementById("demo").innerHTML = fruits;
}
</script>

</body>
</html>
```

JavaScript Array Sort

The sort() method sorts an array alphabetically.

Try it

Apple,Banana,Mango,Orange

Phương thức mảng

❖ Numeric Sort

JavaScript Array Sort

Click the button to sort the array in ascending order.

Try it

1,5,10,25,40,100

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

<h2>JavaScript Array Sort</h2>

<p>Click the button to sort the array in ascending order.</p>

<button onclick="myFunction()">Try it</button>

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

<script>
  var points = [40, 100, 1, 5, 25, 10];
  document.getElementById("demo").innerHTML = points;
  function sorter(a, b){return a - b}
  function myFunction() {
    points.sort(sorter);
    document.getElementById("demo").innerHTML = points;
  }
</script>

</body>
</html>
```

Hàm trong Javascript

❖ Dạng thức khai báo chung:

```
function Tên_hàm(thamso1, thamso2,..)
{
    .....
}
```

❖ Hàm có giá trị trả về:

```
function Tên_hàm(thamso1, thamso2,..)
{
    .....
    return (value);
}
```

Hàm trong Javascript

❖ Ví dụ:

```
function Sum(x, y)  
{  
    tong = x + y;  
    return tong;  
}
```

❖ Gọi hàm:

```
var x = Sum(10, 20);
```

Một số qui tắc chung

- ❖ Khối lệnh được bao trong dấu { }
- ❖ Mỗi lệnh nên kết thúc bằng dấu ;
- ❖ Cách ghi chú thích:
 - // Chú thích 1 dòng
 - /* Chú thích nhiều dòng */

Một số hàm thông dụng

❖ Hàm eval

- Hàm eval được dùng để đánh giá một chuỗi và không cần tham chiếu đến bất kỳ một đối tượng cụ thể nào.
- `eval(string)`

❖ Hàm isNaN

- Hàm isNaN được dùng để kiểm tra xem đối số truyền vào có phải là một số hay không.
- `isNaN(testValue)`

Lệnh điều kiện if

```
if (condition) {  
    statement[s] if true  
}  
else {  
    statement[s] if false  
}
```

Ví dụ:

```
var x = 5, y = 6, z;  
if (x == 5) {  
    if (y == 6) z = 17;  
}  
else  
    z = 20;
```

Lệnh rẽ nhánh switch

```
switch (expression)
{
    case label :
        statementlist
    case label :
        statementlist
    ...
    default :
        statement list
}
```

Ví dụ :

```
var diem = "G";
switch (diem) {
    case "Y":
        document.write("Yếu");
        break;
    case "TB":
        document.write("Trung bình");
        break;
    case "K":
        document.write("Khá");
        break;
    case "G" :
        document.write("Giỏi");
        break;
    default:
        document.write("Xuất sắc")
}
```

Vòng lặp for

```
for ([initial expression]; [condition];  
    [update expression]) {  
    statement[s] inside loop  
}
```

Ví dụ:

```
var myarray = new Array();  
for (i = 0; i < 10; i++)  
{  
    myarray[i] = i;  
}
```

Vòng lặp for – Ví dụ 1

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

<h2>JavaScript Arrays</h2>
<p>The best way to loop through an array is using a standard for loop:
</p>
<p id="demo"></p>

<script>
var fruits, text, fLen, i;
fruits = ["Banana", "Orange", "Apple", "Mango"];
fLen = fruits.length;

text = "<ul>";
for (i = 0; i < fLen; i++) {
    text += "<li>" + fruits[i] + "</li>";
}
text += "</ul>";

document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

JavaScript Arrays

The best way to loop through an array is using a standard for loop:

- Banana
- Orange
- Apple
- Mango

Vòng lặp for – Ví dụ 2

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

<h2>JavaScript For Loop</h2>

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

<script>
var cars = ["BMW", "Volvo", "Saab", "Ford", "Fiat", "Audi"];

```

JavaScript For Loop

BMW
Volvo
Saab
Ford
Fiat
Audi



```
</script>

</body>
</html>

```

Vòng lặp while

```
while (expression)
{
    statements
}
```

Ví dụ:

```
var i = 9, total = 0;
while (i < 10)
{
    total += i * 3 + 5;
    i = i + 5;
}
```

Vòng lặp do...while

```
do  
  {  
    statement  
  } while (expression);
```

Ví dụ:

```
var i = 9, total = 0;  
do  
{  
    total += i * 3 + 5;  
    i = i + 5;  
} while (i > 10);
```

Các đối tượng trong Javascript

- ❖ String
- ❖ Number
- ❖ Array
- ❖ Date
- ❖ Math
- ❖ Window
- ❖ Document
- ❖ History
- ❖ Location
- ❖ Navigator

Đối tượng String

❖ Thuộc tính:

- length: Chiều dài của chuỗi
- constructor: Dùng để kiểm tra kiểu của biến
- Nối kết các chuỗi bằng toán tử “+”

```
<script language="javascript">
  var sTenBien = new String();
  sTenBien = 256;
  if (sTenBien.constructor == String)
  {
    document.writeln("sTenBien la mot chuoi co chieu dai la:" + sTenBien.length);
  }
  else{
    document.writeln("sTenBien la mot so co gia tri la: " + sTenBien);
  }
</script>
```

Đối tượng String

- ❖ `s1 = "foo"` //creates a string literal value
- ❖ `s2 = new String("foo")` //creates a String object
- ❖ Sự khác biệt:
 - `s1 = "2 + 2"` //creates a string literal value
 - `s2 = new String("2 + 2")` //creates a String object
 - `eval(s1)` Kết quả = ?
 - `eval(s2)` Kết quả = ?
- ❖ Phương thức:
 - `stringObj.anchor(anchorString)` – tạo Bookmark
 - `strVariable.link(URL)`
 - `string1.concat([string2[, string3[,... [, stringn]]])`

Đối tượng String

Javascript – String . Phương thức

- anchor
- big
- blink
- bold
- charAt
- charCodeAt
- concat
- fixed
- fontcolor
- fontsize
- fromCharCode
- indexOf
- italics
- lastIndexOf
- link
- localeCompare
- match
- replace
- search
- slice
- small
- split
- strike
- sub
- substr
- substring
- sup
- toLocaleLowerCase
- toLocaleUpperCase
- toLowerCase
- toUpperCase
- toString
- valueOf

Đối tượng String

Phương thức	Ví dụ
strObj.charAt(index)	<pre>var str = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"; var s1 = str.charAt(1); // s1 = "B" var s2 = str.charAt(26); // s2 = null</pre>
strObj.charCodeAt(index)	<pre>var n = str.charCodeAt(1); //Get the Unicode char at pos 1.</pre>
strObj.concat([string2[, string3[,... [, stringn]]])	<pre>var str1 = "ABCDEFGHJKLM" var str2 = "NOPQRSTUVWXYZ"; var s = str1.concat(str2); // s = "ABC...NOP...XYZ"</pre>
strObj.fromCharCode([code1[, code2[,.... [, codeN]]])	<p>Tạo chuỗi từ mã ngoài Unicode.</p> <pre>var test = String.fromCharCode(112, 108, 97, 105, 110);</pre>
strObj.indexOf(subString[, startIndex])	<pre>var str1 = "BABEBIBOBUBABEBIBOBU" var s = str1.indexOf("BEB", 0); // s = 2</pre>
strObj.lastIndexOf(substring[, startindex])	<pre>var str1 = "BABEBIBOBUBABEBIBOBU" var s = str1.lastIndexOf("BEB", 0); // s = 12</pre>

Đối tượng String

Phương thức	Ví dụ
strObj.match (rgExp) rgExp = /pattern/[flags] flags, may be combined: g (global search for all occurrences of pattern) i (ignore case) m (multiline search)	<pre>var s = "The rain in Spain falls mainly in the plain"; var re = /The/gi; //Create regular expression pattern. var r = s.match(re); // r = {"The", "the"}.</pre>
strObj.replace (rgExp, replaceText)	Tìm và thay thế.
stringObj.search (rgExp)	Có xuất hiện, cho ra vị trí. Ngược lại -1.
stringObj.slice (start, [end])	Lấy chuỗi con.
stringObj.split ([separator[, limit]]) separator: string or Regular Expression	<pre>var s = "The rain in Spain falls mainly in the plain."; var ss = s.split(" ", 3); // ss = {"The"; "rain"; "in"}</pre>
string.anchor (anchorString)	<pre>var string = "Information about Fish"; var str_anchor = string.anchor("fish_infor"); window.document.writeln(str_anchor); // Information about fish</pre>

Đối tượng Date

❖ `dateObjectName = new Date([parameters])`

❖ Các constructor:

- `var dateObj = new Date()`
- `var dateObj = new Date(dateVal)`
- `var dateObj = new Date(year, month, date[,hours[,minutes[,seconds[,ms]]]])`
- `var dateObj = new Date(dateString)`

❖ Ví dụ:

- `var my_date=new Date("October 12, 1988 13:14:00")`
- `var my_date=new Date("October 12, 1988")`
- `var my_date=new Date(88,09,12,13,14,00)`
- `var my_date=new Date(88,09,12)`
- `today = new Date().`

Date – Phương thức get

Method	Description
getFullYear()	Get the year as a four digit number (yyyy)
getMonth()	Get the month as a number (0-11)
getDate()	Get the day as a number (1-31)
getHours()	Get the hour (0-23)
getMinutes()	Get the minute (0-59)
getSeconds()	Get the second (0-59)
getMilliseconds()	Get the millisecond (0-999)
getTime()	Get the time (milliseconds since January 1, 1970)
getDay()	Get the weekday as a number (0-6)
Date.now()	Get the time. ECMAScript 5.

Date – Phương thức get

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

<h2>JavaScript getDay()</h2>

<p>The getDay() method returns the weekday as a number:</p>

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

<script>
var d = new Date();
document.getElementById("demo").innerHTML = d.getDay();
</script>

</body>
</html>
```


Date – Phương thức get

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

<h2>JavaScript getMonth()</h2>

<p>The getMonth() method returns the month of a date as a
number from 0 to 11.</p>
<p>To get the correct month, you must add 1:</p>

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

<script>
var d = new Date();
document.getElementById("demo").innerHTML = d.getMonth() + 1;
</script>

</body>
</html>
```

Date – Bài tập

**Viết chương trình xuất ra trình duyệt
hôm nay là thứ mấy trong tuần?**

“Today is Thursday”

Date – Phương thức set

Method	Description
<code>setDate()</code>	Set the day as a number (1-31)
<code>setFullYear()</code>	Set the year (optionally month and day)
<code>setHours()</code>	Set the hour (0-23)
<code>setMilliseconds()</code>	Set the milliseconds (0-999)
<code>setMinutes()</code>	Set the minutes (0-59)
<code>setMonth()</code>	Set the month (0-11)
<code>setSeconds()</code>	Set the seconds (0-59)
<code>setTime()</code>	Set the time (milliseconds since January 1, 1970)

Date – Phương thức set

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

<h2>JavaScript setFullYear()</h2>

<p>The setFullYear() method sets the year of a
date object:</p>

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

<script>
var d = new Date();
d.setFullYear(2020);
document.getElementById("demo").innerHTML = d;
</script>

</body>
</html>
```

JavaScript setFullYear()

The setFullYear() method sets the year of a date object:

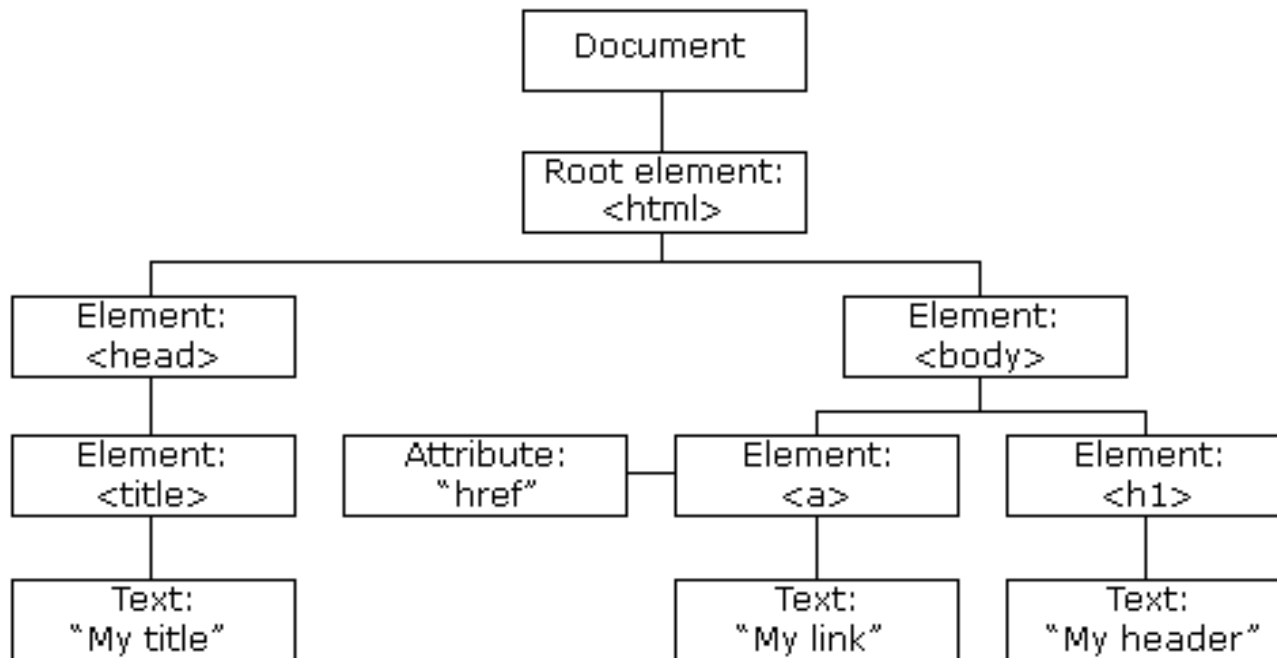
Sun Aug 30 2020 21:46:27 GMT+0700 (Giờ Đông Dương)

JavaScript Math Object

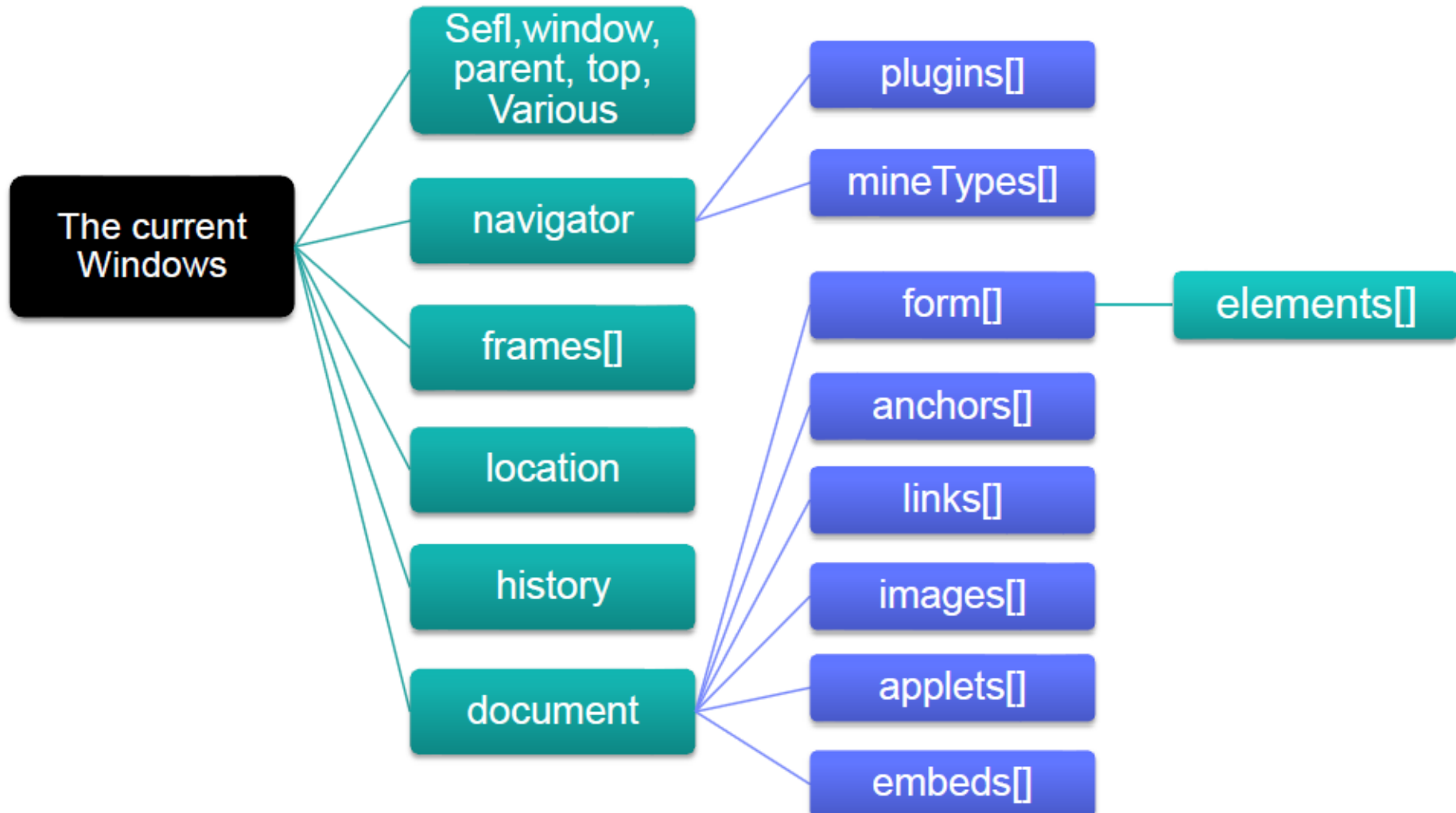
- ❖ `Math.abs(x)`
- ❖ `Math.pow(x, y)`
- ❖ `Math.sqrt(x)`
- ❖ `Math.round(x)`
- ❖ `Math.ceil(x)`
- ❖ `Math.floor(x)`
- ❖ `Math.min([number1[, number2[. . . [, numberN]]]])`
- ❖ `Math.max([number1[, number2[. . . [, numberN]]]])`
- ❖ `Math.random()`

JavaScript HTML DOM

- ❖ With the HTML DOM, JavaScript can access and change all the elements of an HTML document
- ❖ When a web page is loaded, the browser creates a Document Object Model of the page



JavaScript HTML DOM



JavaScript HTML DOM

- Mỗi đối tượng DOM đều có danh sách thuộc tính (**Properties**) và danh sách các phương thức (**Method**) tương ứng.

- Cách sử dụng:

objectName.**propertyName** = value

objectName.**methodName**()

- Ví dụ:

document.**bgColor** = "blue";

window.**focus**();

JavaScript HTML DOM – Example

```
<html>
<body>

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

<script>
document.getElementById("demo").innerHTML = "Hello World!";
</script>

</body>
</html>
```

Window object

- ❖ Là thể hiện của đối tượng cửa sổ trình duyệt
- ❖ Tồn tại khi mở 1 tài liệu HTML
- ❖ Sử dụng để truy cập thông tin của các đối tượng trên cửa sổ trình duyệt (Tên trình duyệt, phiên bản trình duyệt, thanh tiêu đề, thanh trạng thái,...)
- ❖ Điều khiển các sự kiện xảy ra trong window
- ❖ Nếu tài liệu định nghĩa nhiều frame, browser tạo 1 window object cha và các window object con cho từng frame

Window object

❖ Properties

- document
- event
- history
- location
- name
- navigator
- screen
- status

● Methods

- alert
- confirm
- prompt
- blur
- close
- focus
- open

Window object

❖ alert()

- `window.alert("How are you doing?");`

❖ prompt()

- `window.prompt("Please provide your name","optional text");`

❖ confirm()

- `window.confirm("Are you sure you want to do this?");`

❖ blur() - Move This Window to Back

❖ focus() - Bring Main Window to Front

❖ open()

- `window.open("URL", "window name", "attributes");`
- `newWin = window.open("", "", "width=300,height=250");`

Window object

❖ moveBy():

- window.moveBy(xDistance,yDistance);

<form>

<input type="Button" value="Move Window"
onclick="window.moveBy(50,50);">

<input type="Button" value="Close Window"
onclick="window.close();">

</form>

❖ moveTo():

- window.moveTo(xDistance,yDistance);

❖ resizeBy():

- window.resizeBy(xSize,ySize);

❖ resizeTo():

- window.resizeTo(xWidth,yLength);

Window object

Các chương trình xử lý sự kiện

onLoad – Xuất hiện khi cửa sổ kết thúc việc tải.

onUnload – Xuất hiện khi cửa sổ được loại bỏ.

```
<body onLoad="alert('This page has finished loading!')"
```

```
window.onload=functionName;
```

```
<body onUnload="alert('You're leaving so soon?')"
```

❖ onResize

❖ onError

❖ onBlur

❖ onFocus

Window object – Example

❖ Ví dụ:

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

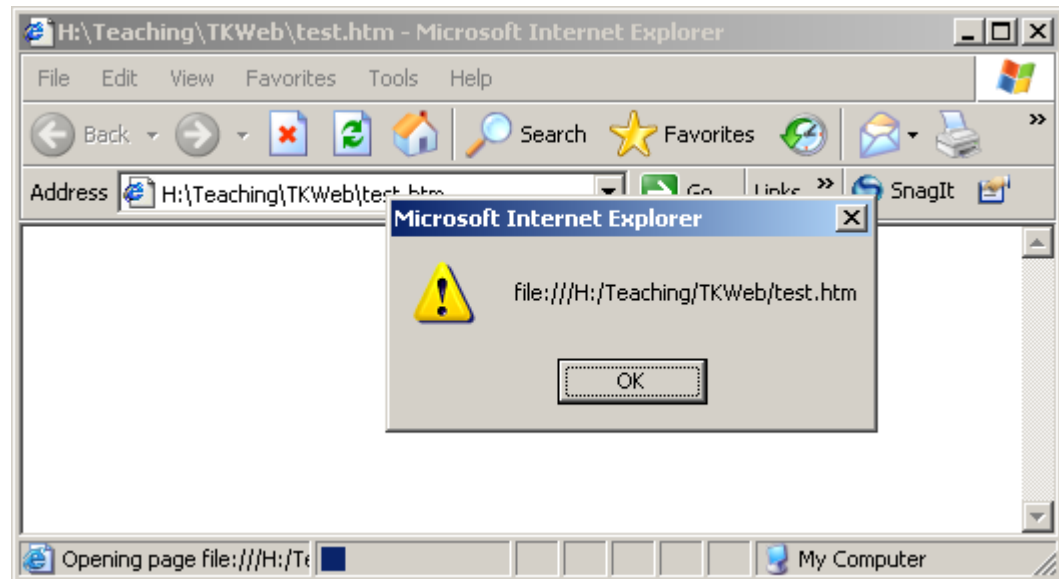
```
var curURL = window.location;
```

```
window.alert(curURL);
```

```
</script>
```

```
</body>
```

```
</html>
```



Document object

- ❖ Document Object biểu diễn cho tài liệu HTML
- ❖ Dùng để lấy thông tin về tài liệu, các thành phần (phần tử) HTML và xử lý sự kiện

□ Properties

- **aLinkColor**
- **bgColor**
- **Body**
- **fgColor**
- **linkColor**
- **Location**
- **Title**
- **URL**
- **vlinkColor**
- ...

□ Methods

- **clear**
- **close**
- **open**
- **write**
- **writeln**
- **document.getElementById(id)**
- **document.getElementsByTagName(name)**
- **document.getElementsByClassName(name)**
- ...

History Object

- ❖ History Object cung cấp danh sách các URL đã được duyệt bởi người dùng
- ❖ Methods:
 - Back
 - Forward
 - go
- ❖ `History.go(-1)` – `history.back()`
- ❖ `History.go(+1)` – `history.forward()`

Location Object

❖ Chứa thông tin về URL hiện tại

❖ Properties

- hostname
- href
- pathname
- ...

❖ Methods

- Reload
- ...

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
```

```
<body>
```

```
  <script type="text/javascript">
```

```
    document.write("appName: " + navigator.appName);
```

```
    document.write("<br/>appVersion : " + navigator.appVersion);
```

```
    document.write("<br/>appCodeName : " + navigator.appCodeName);
```

```
    document.write("<br/>cookieEnabled : " + navigator.cookieEnabled);
```

```
    document.write("<br/>hostname : " + location.hostname);
```

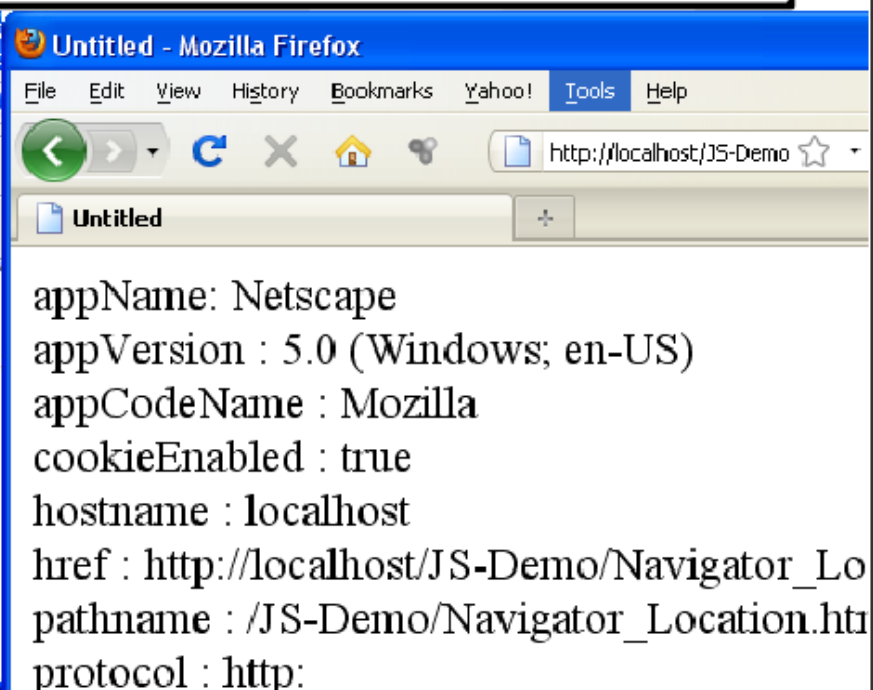
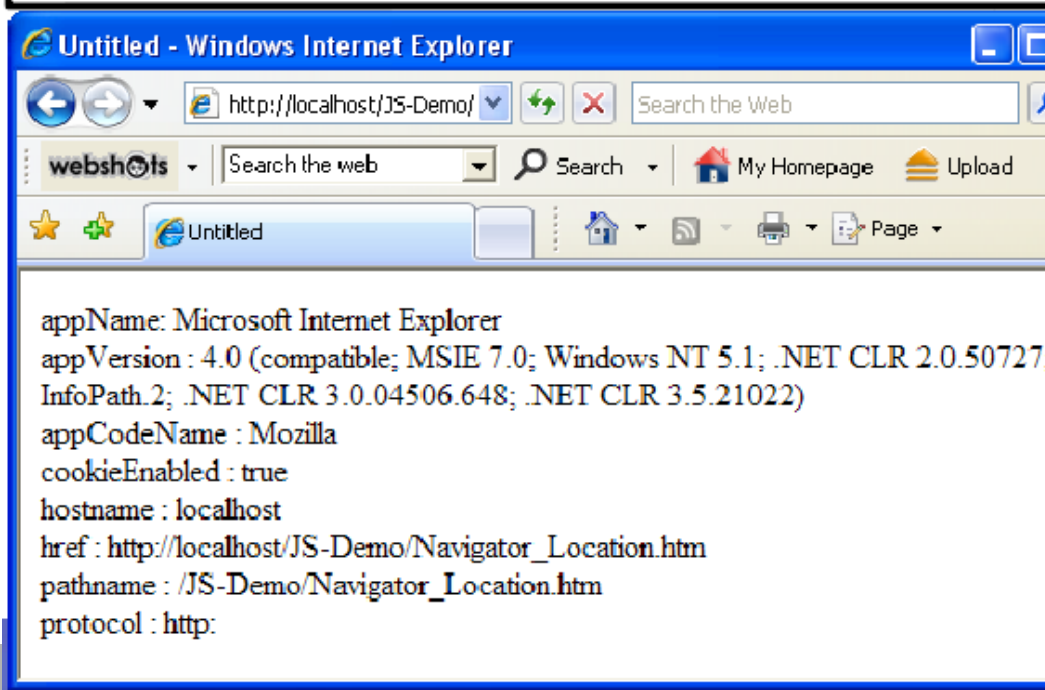
```
    document.write("<br/>href : " + location.href);
```

```
    document.write("<br/>pathname : " + location.pathname);
```

```
    document.write("<br/>protocol : " + location.protocol);
```

```
  </script>
```

```
</body>
```



Navigator Object

❖ Cung cấp thông tin về trình duyệt Browser

Thuộc tính

- `appName`
- `appVersion`
- `appCodeName`
- `cookieEnabled`
- `online`
- `platform`
- ...



Phương thức

- `javaEnabled()`
- ...

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
```

```
<body>
```

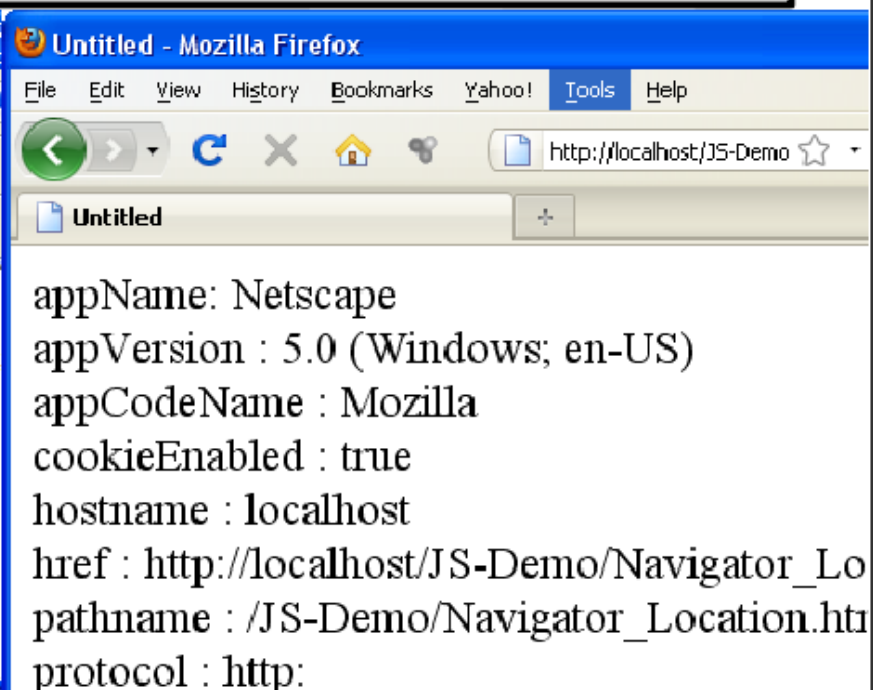
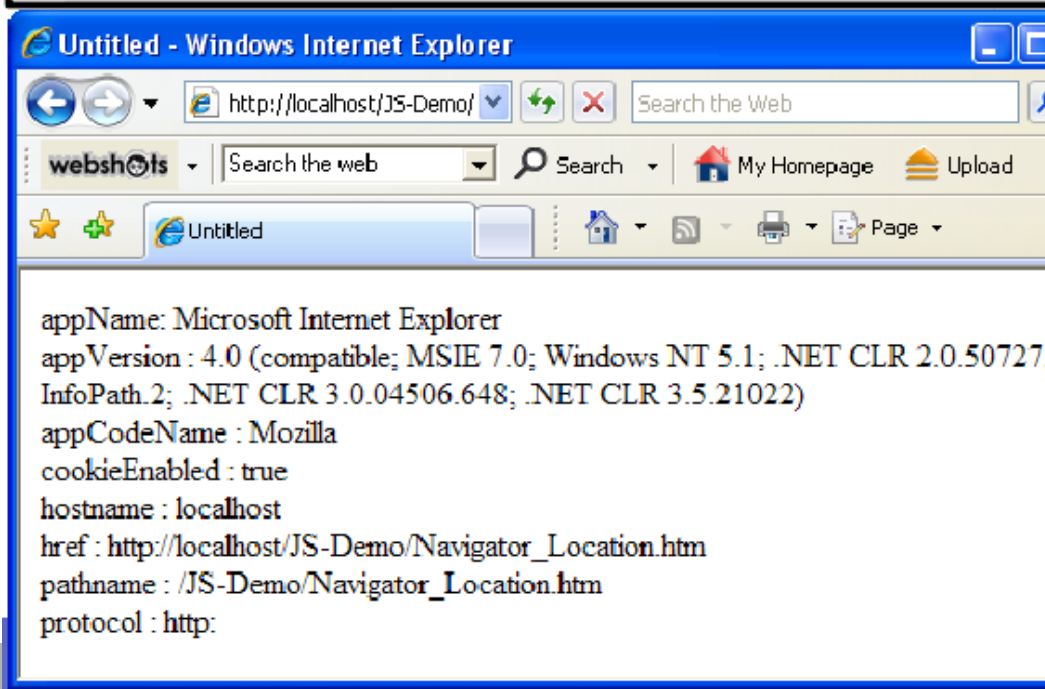
```
  <script type="text/javascript">
```

```
    document.write("appName: " + navigator.appName);  
    document.write("<br/>appVersion : " + navigator.appVersion);  
    document.write("<br/>appCodeName : " + navigator.appCodeName);  
    document.write("<br/>cookieEnabled : " + navigator.cookieEnabled);
```

```
    document.write("<br/>hostname : " + location.hostname);  
    document.write("<br/>href : " + location.href);  
    document.write("<br/>pathname : " + location.pathname);  
    document.write("<br/>protocol : " + location.protocol);
```

```
  </script>
```

```
</body>
```



HTML Objects

- ❖ Mỗi form trong một document sẽ tạo ra một đối tượng form
- ❖ Một document có thể có nhiều form, được lưu trong một forms array (bắt đầu từ form[0])
- ❖ Truy cập đến form
 - document.forms[0]
 - document.formName
- ❖ Truy cập đến các thành phần của form
 - document.forms[0].item[0].value
 - document.formName.InputName.value

Xử lý sự kiện – Event object

- ❖ Events là các sự kiện xảy ra trên trang Web
- ❖ Do người dùng hoặc do hệ thống tạo ra
- ❖ Mỗi sự kiện sẽ liên quan đến một event object
- ❖ Cung cấp thông tin về event
 - Loại event
 - Vị trí con trỏ tại thời điểm xảy ra sự kiện

Các sự kiện thông dụng

❖ Các sự kiện được hỗ trợ bởi hầu hết các đối tượng

- onClick
- onFocus
- onChange
- onBlur
- onMouseOver
- onMouseOut
- onMouseDown
- onMouseUp
- onLoad
- onSubmit
- onResize
- ...

Xử lý sự kiện cho các thẻ HTML

❖ Cú pháp 1:

- `<TAG eventHandler = "JavaScript Code">`

❖ Ví dụ:

```
<body>
```

```
    <INPUT TYPE="button" NAME="Button1"
```

```
    VALUE="OpenSesame!"
```

```
    onClick="window.open('mydoc.html');">
```

```
</body>
```

❖ Lưu ý: Dấu “...” và ‘...’

Xử lý sự kiện bằng function

```
<head>
  <script language="Javascript">
    function GreetingMessage ()
    {
      window.alert("Welcome to my world");
    }
  </script>
</head>

<body onload="GreetingMessage ()">
</body>
```

Xử lý sự kiện bằng thuộc tính

- ❖ Gán tên hàm xử lý cho 1 object event
 - `object.eventhandler = function_name;`

❖ Ví dụ:

```
<head>
  <script language="Javascript">
    function GreetingMessage()
    {
        window.alert("Welcome to my world");
    }

    window.onload = GreetingMessage();
  </script>
</head>

<body>

</body>
```

Ví dụ: onclick Event

```
<SCRIPT LANGUAGE="JavaScript">
```

```
function compute(frm)
```

```
{
```

```
    var x = frm.expr.value;
```

```
    result.innerHTML = x*x;
```

```
}
```

```
</SCRIPT>
```

```
<FORM name="frm">
```

```
  X = <INPUT TYPE="text" NAME="expr" SIZE=15>
```

```
  <BR><BR>
```

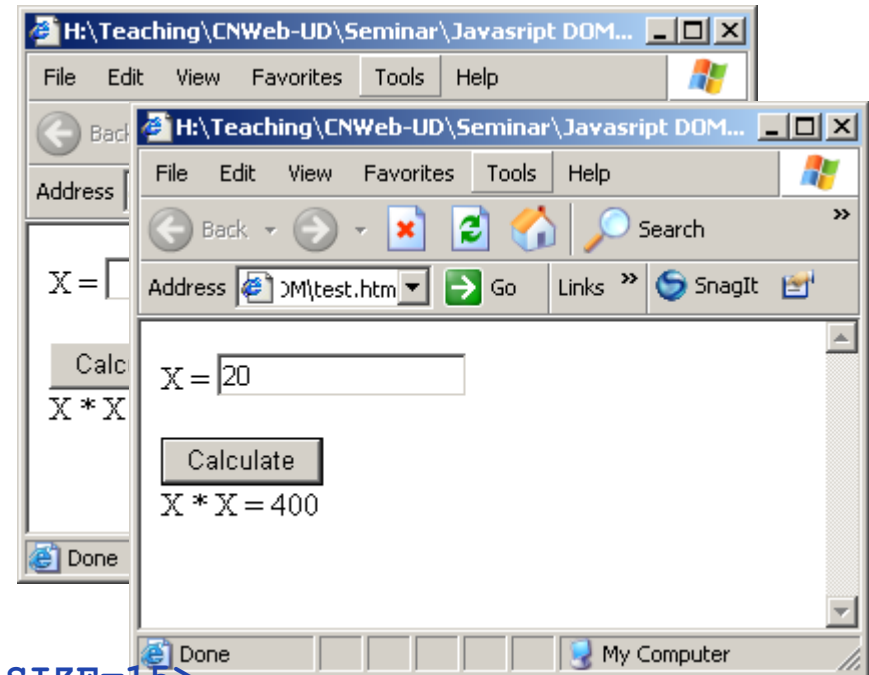
```
  <INPUT TYPE="button" VALUE="Calculate"
```

```
    ONCLICK="compute(this.form)">
```

```
  <BR>
```

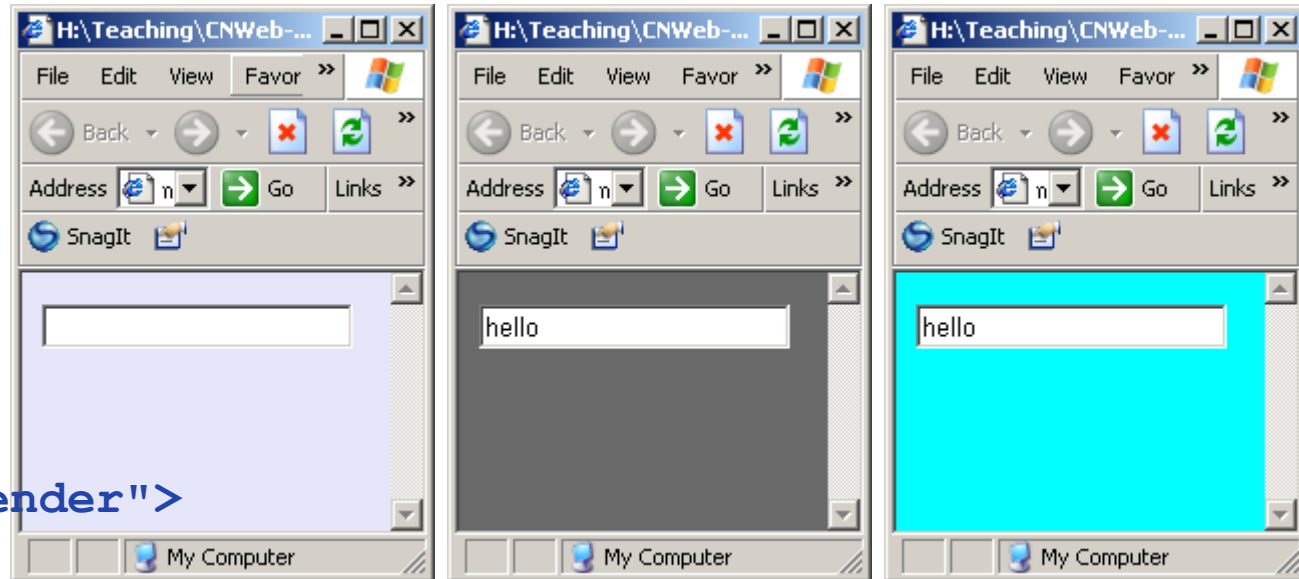
```
  X * X = <SPAN ID="result"></SPAN>
```

```
</FORM>
```



Ví dụ: onFocus - onBlur

- ❖ Xảy ra khi một thành phần HTML bị focus (onFocus) và mất focus (onBlur)
- ❖ Ví dụ:



```
<BODY BGCOLOR="lavender">
```

```
<FORM>
```

```
<INPUT type="text" name="myTextbox"
```

```
onblur=" (document.bgColor='aqua') "
```

```
onfocus=" (document.bgColor='dimgray') ">
```

```
</FORM>
```

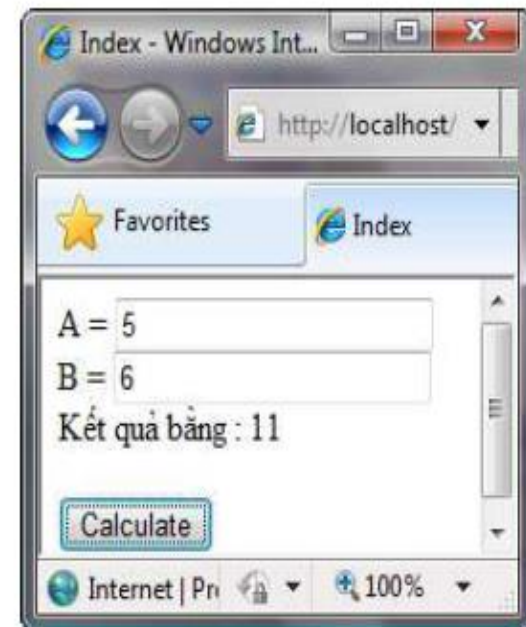
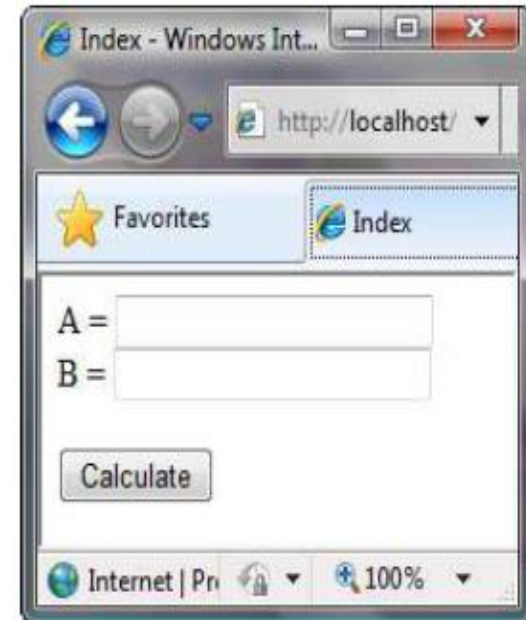
```
</BODY>
```

Các sự kiện của Form field

	Blur	Click	Change	Focus	Load	Mouse over	Select	Submit	Unload
Button		x							
Checkbox		x							
Document					x				x
Form								x	
Link		x				x			
Radio		x							
Reset		x							
Selection	x		x	x					
Submit		x							
Text	x		x	x			x		
Textarea	x		x	x			x		

Ví dụ: Tính tổng 2 số:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Index</title>
<script type="text/javascript" language="javascript">
    function btnTinhTong()
    {
        var a = parseInt(document.getElementsByName("txtSoA")[0].value);
        var b = parseInt(document.getElementsByName("txtSoB")[0].value);
        var tong = a + b;
        document.getElementById("divKetQua").innerHTML = "Kết quả bằng : " + tong;
    }
</script>
</head>
<body>
<form id="form1" name="form1" method="get" action="">
    A = <input type="text" name="txtSoA" id="txtSoA" /> <br/>
    B = <input type="text" name="txtSoB" id="txtSoB" /> <br/>
    <div id="divKetQua" name="divKetQua" > </div> <br/>
    <input type="button" onclick="btnTinhTong();" value="Calculate" />
</form>
</body>
</html>
```



Ví dụ - Form validation

...

```
<form method="get" name="fields" action="mydoc.htm"  
onsubmit="javascript: return checkAll();">
```

```
  <p>Field 1: <input type="text" name="f1">
```

```
  <br>Field 2: <input type="text" name="f2">
```

```
  <br>Field 3: <input type="text" name="f3">
```

```
  <br>Field 4: <input type="text" name="f4">
```

```
  </p>
```

```
  <input type="reset">
```

```
  <input type="submit" value="Submit">
```

```
</form>
```

...

Ví dụ - Form validation

```
<script language="javascript">
function checkAll() {
    for (i = 0; i < document.fields.elements.length-2; i++){
        var f = document.fields.elements[i];
        if (f.value == ""){
            alert("Please enter a value for Field " + (i + 1));
            f.style.borderColor="#FF0000";
            f.focus();
            return false;
        }
    }
    return true;
}
</script>
```

Ví dụ - Form validation (Phone Number)

...

```
<form onsubmit = "javascript: return validPhone();"
      action="mydoc.htm" method="get" name="phone">
  <p>Please enter your phone number:
  (<input type="text" name="area" size="3" maxlength="3">)
  <input type="text" name="pre" size="3" maxlength="3"> -
  <input type="text" name="last" size="4" maxlength="4">
  </p>
  <input type="reset">
  <input type="submit" value="Submit">
</form>
```

...

Ví dụ - Form validation (Phone Number)

```
<script language="javascript">
function validPhone() {
    var phNum = document.phone.area.value +
        document.phone.pre.value + document.phone.last.value;
    for (i = 0; i < phNum.length; i++) { // Check for numbers only
        if (phNum.charAt(i) < "0" || phNum.charAt(i) > "9")
        {
            alert("Please enter only numbers.");
            return false;
        }
    }
    if (phNum.length < 10) { // Check for 10 digits
        alert("Please enter your 10-digit phone number.");
        return false;
    }
    return true;
}
</script>
```

Mở một cửa sổ bằng java script

```
<html>
<head>
<script>
var newwindow;
function showWindow()
{
    newwindow = window.open("", "new window", "toolbar=no,width=300,height=25,top=30");
    newwindow.document.open();
    newwindow.document.write("<html><title>New Window</title><body><em>This is a test</em></body></html>");
    newwindow.document.close();
}
function closeWindow() { newwindow.close(); }
</script>
</head>
<body>
    <input type="button" value="Open" onClick="return showWindow();"/>
    <input type="button" value="Close" onClick="return closeWindow();"/>
</body>
</html>
```

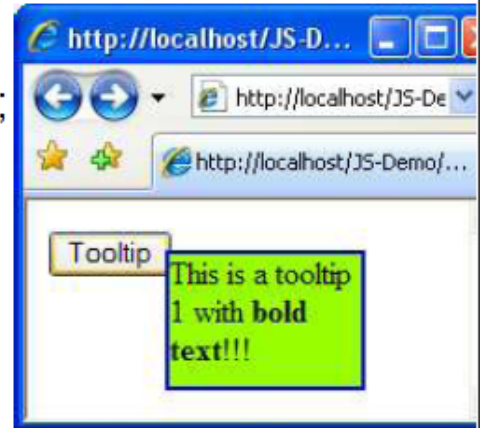
Ví dụ: Java script và CSS

```
<html>
<head>
<script language="javascript">
    function setStyle( object, styleText )
    {
        if( object.style.setAttribute )
            object.style.setAttribute("cssText", styleText );
        else
            object.setAttribute("style", styleText );
    }

    function showTooltip()
    {
        setStyle (document.getElementById("divTooltip"), "display:visible;");
    }
    function hideTooltip()
    {
        setStyle (document.getElementById("divTooltip"), "display:none;");
    }
</script>
</head>
<body>
    <input type="button" value="Tooltip" onMouseOver="showTooltip(this);" onMouseOut="hideTooltip(this);" />
    <div id="divTooltip" class="tooltip">
        This is a tooltip !!!
    </div>
</body>
</html>
```

Ví dụ: Tooltip với java script

```
<html> <head>
<script language="javascript">
    function setStyle( object, styleText ) {
        if( object.style.setAttribute )
            object.style.setAttribute("cssText", styleText );
        else
            object.setAttribute("style", styleText );
    }
    function showTooltip(e) {
        var evt = e || window.event;
        var mouseX = evt.clientX + document.body.scrollLeft;
        var mouseY = evt.clientY + document.body.scrollTop;;
        setStyle (document.getElementById("divTooltip"), "display:visible;left:" + mouseX
    }
    function hideTooltip() { setStyle (document.getElementById("divTooltip"), "display:none;");
</script>
<style> div.tooltip { position:absolute; height:70px; width:100px; background-color:#9F0; border:#0
</head>
<body>
    <input id="btn1" type="button" value="Tooltip" onmouseout="hideTooltip();" />
    <div id="divTooltip" class="tooltip" style="display:none;">
        This is a tooltip 1 with <b>bold text</b>!!!
    </div>
    <script language="javascript">
        document.getElementById("btn1").onmousemove = showTooltip; // need to assign
    </script>
</body> </html>
```



Bài tập 1

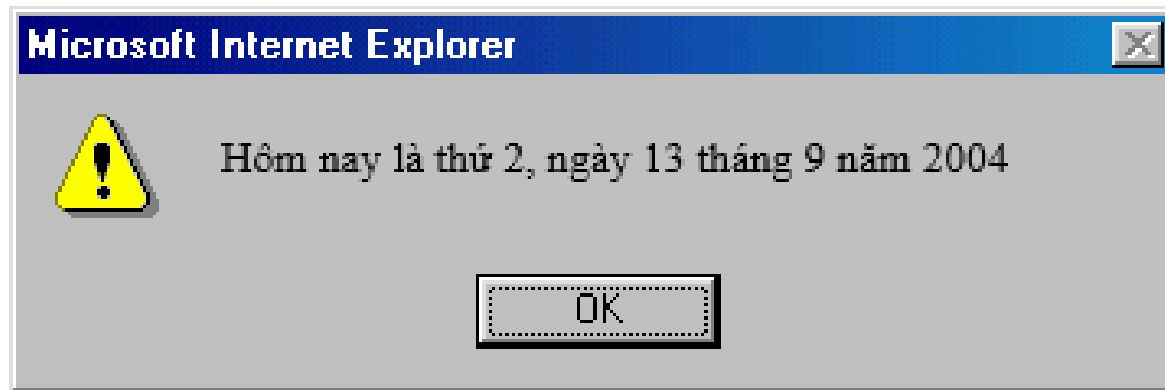
- ❖ Sử dụng Javascript, xây dựng trang web cho phép người dùng nhập vào tên và tuổi. Hãy viết lại tên và tuổi của người đó ra màn hình bằng hàm `document.write`, trong đó tên có màu đậm, tuổi được gạch chân.

Bài tập 1

```
<HTML>
  <HEAD> </HEAD>
  <BODY>
    <script language = "JavaScript">
      var Ten, Tuoi; // Khai báo 2 biến để lưu tên và tuổi
      Ten = prompt("Bạn hãy nhập vào tên ", "");
      Tuoi = prompt("Bạn hãy nhập vào Tuổi : ", 20);
      document.write("Chào bạn : <B> " + Ten + "</B>");
      document.write("<BR>"); // Xuống dòng
      document.write("Tuổi của bạn là : <U> " + Tuoi + "</U>");
    </script>
  </BODY>
</HTML>
```


Bài tập 2

- ❖ Sử dụng đối tượng Date trong JavaScript để hiển thị ngày giờ của hệ thống. Hãy hiển thị ngày và giờ của hệ thống máy tính khi trang Web được nạp. Thông tin hiển thị ra có dạng như sau:



Bài tập 3

- ❖ Sử dụng đối tượng Date để tính tuổi của một người. Cho phép người dùng nhập vào năm sinh của họ, sau đó hiển thị tuổi tương ứng.

Bài tập 3

```
<HTML>
<TITLE>Tính tuổi</TITLE>
<BODY>
<script language="JavaScript">
var D = new Date();
var NamSinh, NamHienTai;
NamHienTai = D.getFullYear(); // Lưu năm hiện tại vào biến
NamSinh = prompt("Bạn sinh năm bao nhiêu ? : ","");
alert("Tuổi của bạn bây giờ là : " + (NamHienTai-NamSinh));
</script>
</BODY>
</HTML>
```

Bài tập 4

- ❖ Sử dụng Javascript, xây dựng trang web cho phép người dùng nhập một số nguyên vào textbox.
- ❖ Khi người dùng click vào button “Kiểm tra” sẽ hiển thị kết quả số nguyên vừa nhập có phải là số nguyên tố hay không?

Q & A

